(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第4836208号
(P4836208)
(45)発行日　平成23年12月14日(2011.12.14)　(24)登録日　平成23年10月7日(2011.10.7)

(51)Int.Cl.　F I
G09C 1/00 (2006.01)　G09C 1/00 650A
G09C 1/00 660D

請求項の数 11　(全 22 頁)

(21)出願番号	特願2008-526767 (P2008-526767)
(86)(22)出願日	平成19年7月24日 (2007.7.24)
(86)国際出願番号	PCT/JP2007/064474
(87)国際公開番号	WO2008/013154
(87)国際公開日	平成20年1月31日 (2008.1.31)
審査請求日	平成22年6月9日 (2010.6.9)
(31)優先権主張番号	特願2006-200946 (P2006-200946)
(32)優先日	平成18年7月24日 (2006.7.24)
(33)優先権主張国	日本国(JP)
(31)優先権主張番号	特願2007-10072 (P2007-10072)
(32)優先日	平成19年1月19日 (2007.1.19)
(33)優先権主張国	日本国(JP)

(73)特許権者　504147243
国立大学法人　岡山大学
岡山県岡山市北区津島中一丁目１番１号
(74)代理人　100080160
弁理士　松尾　憲一郎
(72)発明者　野上　保之
岡山県岡山市津島中三丁目１番１号　岡山大学大学院自然科学研究科内
(72)発明者　森川　良孝
岡山県岡山市津島中三丁目１番１号　岡山大学大学院自然科学研究科内
審査官　石田　信行

最終頁に続く

(54)【発明の名称】暗号化／復号化プログラム、暗号化／復号化装置及び拡大体の乗算装置

(57)【特許請求の範囲】
【請求項１】
素数ｐを標数とし、拡大次数ｍの拡大体F_p^mの２つの元Ａ＝{a_0,a_1,a_2,・・・, a_{m-1}}、Ｂ＝{b_0,b_1,b_2,・・・, b_{m-1}}を、平文データと暗号化鍵、または暗号データと復号化鍵として、
電子計算機で、前記平文データと前記暗号化鍵とを乗算して暗号データの元Ｃ＝{c_0,c_1,c_2,・・・, c_{m-1}}を生成させる、または前記暗号データと前記復号化鍵とを乗算して平文データの元Ｃ＝{c_0,c_1,c_2,・・・, c_{m-1}}を生成させる暗号化／復号化プログラムにおいて、
km+1が素数であって、F_{km+1}でｐが原始元となる正整数ｋを特定する第１のステップと、
前記の２つの元Ａ，Ｂを、前記正整数ｋを用いて素数ｐを標数とする拡大次数kmの拡大体F_p^{km}における２つの元として乗算を行う第２のステップと、
この乗算の結果を用いて部分体である前記拡大次数ｍの拡大体F_p^mの元における乗算の結果を求める第３のステップと
を有することを特徴とする暗号化／復号化プログラム。
【請求項２】
０≦ｉ≦m-1、０≦ｊ≦k-1とし、
<ｘ>がｘのmod(km+1)をとるものとして、
前記第２のステップは、

$0 \leqq t \leqq k-1$で、$<p^{mt}>=K[t]$となる各$K[t]$をそれぞれ求めるステップと、
$0 \leqq i \leqq km$で、それぞれ$0=q[<i>]$とするステップと、
$0 \leqq i \leqq m-1$で、$a_i b_i \bmod p = q[<p^i>]$をそれぞれ求めるステップと、
$0 \leqq i < j \leqq m-1$で、$(a_i-a_j)(b_i-b_j) \bmod p = M$をそれぞれ求めて、$0 \leqq t \leqq k-1$で$q[<p^i+(p^jK[t])>]$にMをそれぞれ足し込むステップと
を有することを特徴とする請求項１記載の暗号化／復号化プログラム。
【請求項３】
前記第３のステップは、
$0 \leqq i \leqq m-1$で、かつ$1 \leqq t \leqq k-1$で、$q[<p^i>]$に$q[<(p^iK[t])>]$をそれぞれ足し込むステップと、
$0 \leqq i \leqq m-1$で、$kq[<0>]-q[<p^i>]=c_i$をそれぞれ求めるステップと
を有することを特徴とする請求項２記載の暗号化／復号化プログラム。
【請求項４】
$k=2k'$の場合に、
$F_{2k'm+1}$でｐが原始元あるいは位数が$k'm$かつ$k'm$が奇数となる正整数k'とし、
$0 \leqq i \leqq m-1$、$0 \leqq j \leqq 2k'-1$とし、
$<x>$がｘの$\bmod(2k'm+1)$をとるものとして、
前記第２のステップは、
$0 \leqq t \leqq k'-1$で、$<p^{mt}>=K[t]$となる各$K[t]$をそれぞれ求めるステップと、
$0 \leqq i \leqq 2k'm$で、それぞれ$0=q[<i>]$とするステップと、
$0 \leqq i \leqq m-1$で、$a_i b_i \bmod p = q[<p^i>]$をそれぞれ求めるステップと、
$0 \leqq i < j \leqq m-1$で、$(a_i-a_j)(b_i-b_j) \bmod p = M$をそれぞれ求めて、$0 \leqq t \leqq k'-1$で$q[<p^i+(p^jK[t])>]$にMをそれぞれ足し込むとともに、$q[<p^i-(p^jK[t])>]$にMをそれぞれ足し込むステップと
を有することを特徴とする請求項１記載の暗号化／復号化プログラム。
【請求項５】
前記第３のステップは、
$0 \leqq i \leqq m-1$で、かつ$1 \leqq t \leqq k'-1$で、$q[<p^i>]$に$q[<(p^iK[t])>]$をそれぞれ足し込むとともに、$q[<p^i>]$に$q[<-(p^iK[t])>]$をそれぞれ足し込むステップと、
$0 \leqq i \leqq m-1$で、$-q[<p^i>]=c_i$をそれぞれ求めるステップと
を有することを特徴とする請求項４記載の暗号化／復号化プログラム。
【請求項６】
素数ｐを標数とし、拡大次数ｍの拡大体$F_p{}^m$の２つの元$A=\{a_0,a_1,a_2,\cdots,a_{m-1}\}$、$B=\{b_0,b_1,b_2,\cdots,b_{m-1}\}$を、平文データと暗号化鍵、または暗号データと復号化鍵として、
前記平文データと前記暗号化鍵とを乗算させて元$C=\{c_0,c_1,c_2,\cdots,c_{m-1}\}$を生成することにより暗号化する、または前記暗号データと前記復号化鍵とを乗算させて元$C=\{c_0,c_1,c_2,\cdots,c_{m-1}\}$を生成することにより復号化する演算器を備えた暗号化／復号化装置において、
前記元をそれぞれ記憶する第１の記憶部と、
前記拡大次数ｍを記憶する第２の記憶部と、
$km+1$が素数であって、F_{km+1}でｐが原始元となる正整数ｋを前記演算器による演算に基づいて特定して記憶する第３の記憶部と、
前記の２つの元Ａ，Ｂを、前記正整数ｋを用いて素数ｐを標数とする拡大次数ｋｍの拡大体$F_p{}^{km}$における２つの元として前記演算器で乗算した結果を記憶する第４の記憶部と、
前記拡大次数ｋｍの拡大体$F_p{}^{km}$の元の乗算結果を用いて前記演算器で所定の演算を行って、部分体である前記拡大次数ｍの拡大体$F_p{}^m$の元における乗算の結果を求めて記憶する第５の記憶部と
を有することを特徴とする暗号化／復号化装置。
【請求項７】

０≦ｉ≦m-1、０≦ｊ≦k-1とし、
<ｘ>がｘのmod(km+1)をとるものとして、
前記第４の記憶部は、
０≦ｔ≦k-1で、<p^{mt}>＝K[t]となる各K[t]をそれぞれ求めて記憶する記憶部と、
０≦ｉ≦kmで、それぞれ０＝q[<i>]として記憶する記憶部と、
０≦ｉ≦m-1で、$a_i b_i$modp＝q[<p^i>]をそれぞれ求めて記憶する記憶部と、
０≦ｉ＜ｊ≦m-1で、$(a_i-a_j)(b_i-b_j)$modp＝Ｍをそれぞれ求めて記憶する記憶部と、
０≦ｔ≦k-1でq[<$p^i+(p^jK[t])$>]にＭをそれぞれ足し込んで記憶する記憶部と
を有することを特徴とする請求項６記載の暗号化／復号化装置。
【請求項８】
前記第５の記憶部は、
０≦ｉ≦m-1で、かつ１≦ｔ≦k-1で、q[<p^i>]にq[<$(p^iK[t])$>]をそれぞれ足し込んで記憶する記憶部と、
０≦ｉ≦m-1で、kq[<0>]-q[<p^i>]＝c_iをそれぞれ求めて記憶する記憶部と
を有することを特徴とする請求項７記載の暗号化／復号化装置。
【請求項９】
k=2k'の場合に、
$F_{2k'm+1}$でｐが原始元あるいは位数がk'mかつk'mが奇数となる正整数k'として、この正整数k'を記憶する記憶部を有し、
０≦ｉ≦m-1、０≦ｊ≦2k'-1とし、
<ｘ>がｘのmod(2k'm+1)をとるものとして、
前記第４の記憶部は、
０≦ｔ≦k'-1で、<p^{mt}>=K[t]となる各K[t]をそれぞれ求めて記憶する記憶部と、
０≦ｉ≦2k'mで、それぞれ０＝q[<i>]として記憶する記憶部と、
０≦ｉ≦m-1で、$a_i b_i$modp＝q[<p^i>]をそれぞれ求めて記憶する記憶部と、
０≦ｉ＜ｊ≦m-1で、$(a_i-a_j)(b_i-b_j)$modp＝Ｍをそれぞれ求めて記憶する記憶部と、
０≦ｔ≦k'-1でq[<$p^i+(p^jK[t])$>]にＭをそれぞれ足し込し込んで記憶するとともに、q[<$p^i-(p^jK[t])$>]にＭをそれぞれ足し込んで記憶する記憶部と
を有することを特徴とする請求項６記載の暗号化／復号化装置。
【請求項１０】
前記第５の記憶部は、
０≦ｉ≦m-1で、かつ１≦ｔ≦k'-1で、q[<p^i>]にq[<$(p^iK[t])$>]をそれぞれ足し込むとともに、q[<p^i>]にq[<$-(p^iK[t])$>]をそれぞれ足し込んで記憶する記憶部と、
０≦ｉ≦m-1で、-q[<p^i>]＝c_iをそれぞれ求めて記憶する記憶部と
を有することを特徴とする請求項９記載の暗号化／復号化装置。
【請求項１１】
素数ｐを標数とし、拡大次数ｍの拡大体$F_p{}^m$の２つの元Ａ＝{$a_0,a_1,a_2,\cdots,a_{m-1}$}、Ｂ＝{$b_0,b_1,b_2,\cdots,b_{m-1}$}を乗算して元Ｃ＝{$c_0,c_1,c_2,\cdots,c_{m-1}$}を生成する演算器を備えた拡大体の乗算装置において、
前記元をそれぞれ記憶する第１の記憶部と、
前記拡大次数ｍを記憶する第２の記憶部と、
km+1が素数であって、F_{km+1}でｐが原始元となる正整数ｋを前記演算器による演算に基づいて特定して記憶する第３の記憶部と、
前記の２つの元Ａ，Ｂを、前記正整数ｋを用いて素数ｐを標数とする拡大次数kmの拡大体$F_p{}^{km}$における２つの元として前記演算器で乗算した結果を記憶する第４の記憶部と、
前記拡大次数kmの拡大体$F_p{}^{km}$の元の乗算結果を用いて前記演算器で所定の演算を行って、部分体である前記拡大次数ｍの拡大体$F_p{}^m$の元における乗算の結果を求めて記憶する第５の記憶部と
を有することを特徴とする拡大体の乗算装置。
【発明の詳細な説明】

【技術分野】
【０００１】
本発明は、暗号化／復号化プログラム、暗号化／復号化装置、及び暗号化または復号化における乗算処理を実行する拡大体の乗算装置に関する。
【背景技術】
【０００２】
従来、インターネットなどの電気通信回線を利用した情報通信において、送受信されるデータの秘匿性を保持するために、データの暗号化が行われている。すなわち、平文データを送信する送信者は、平文データに暗号化処理を施して暗号データを生成し、この暗号データを受信者に送信している。一方、暗号データを受信した受信者は、暗号データに復号化処理を施して平文データを生成することにより、平文データを受け取ることができる。
【０００３】
このような暗号化及び復号化の方法として、昨今では公開鍵暗号と呼ばれる方法が用いられている。公開鍵暗号では、公開鍵と秘密鍵があらかじめ設定され、所定の平文データを送信する送信者に公開鍵を開示しており、送信者は公開鍵を用いて平文データを暗号化することにより暗号データを生成し、受信者に送信している。受信者は、暗号データを受信すると、秘密鍵を用いて暗号データを復号化して平文データを生成している。
【０００４】
ここで、平文データを暗号化する場合には、平文データを暗号化に用いる公開鍵の鍵長に合わせた所定のデータ長ごとに分断して複数の単位長平文データを生成し、各単位長平文データと公開鍵を用いて乗算処理を行うことにより単位長暗号データを生成し、所定数の単位長暗号データで構成された暗号データを生成している。
【０００５】
一方、暗号データを復号化する場合には、暗号データを復号化に用いる秘密鍵の鍵長に合わせた所定のデータ長ごとに分断して複数の単位長暗号データを生成し、各単位長暗号データと秘密鍵を用いて乗算処理を行うことにより単位長平文データを生成し、所定数の単位長平文データで構成された平文データを生成している。
【０００６】
このような公開鍵暗号には様々な方式が提案されており、Rivest Shamir Adleman（ＲＳＡ）暗号、楕円曲線暗号、ElGamal暗号などが知られている。
【０００７】
昨今、パーソナルコンピュータなどの電子計算機の性能向上にともなって、各種の暗号方法で暗号化された暗号データが解読されるおそれが高まっている。すなわち、暗号データは、秘密鍵さえ分かれば解読できることから、電子計算機によって手当たり次第に秘密鍵を生成して復号化を試みることにより、時間はかかるものの解読されるおそれがあった。
【０００８】
そこで、各暗号方法では、安全性を高めるために公開鍵及び秘密鍵の鍵長を長くすることが行われている。すなわち、公開鍵及び秘密鍵の鍵長を長くすると、解読に用いる秘密鍵の数が飛躍的に増大し、現実的な時間内での解読を不可能とすることができるからである。そのため、現在では、ＲＳＡ暗号では１０２４ビット以上、楕円曲線暗号では１６０ビット以上、ElGamal暗号では１０２４ビット以上の鍵長が求められている。
【０００９】
しかしながら、公開鍵及び秘密鍵の鍵長を長くした場合には、公開鍵を用いた乗算処理による暗号化、及び秘密鍵を用いた乗算処理による復号化に多大な時間を要することとなっていた。この場合、処理時間の短縮のためには、乗算処理の演算速度が高速な演算手段が必要となることにより、安全性を高めるためのコストが増大することとなっていた。
【００１０】
一方で、現実的には、電気通信回線中を流れているデータにおいては重要度が様々に異

なっており、利用価値が高いために高度の秘匿性が必要なデータから、利用価値が低いために秘匿性を必要としないデータまでが存在している。

【００１１】

したがって、何れのデータに対しても同等の安全性で暗号化する必要はなく、高い安全性が要求されるデータには長い鍵長の公開鍵及び秘密鍵を用い、高い安全性が要求されないデータには短い鍵長の公開鍵及び秘密鍵が用いられている。

【００１２】

また、パーソナルコンピュータなどのような高速な演算処理の実行が可能な装置では、できるだけ長い鍵長の公開鍵及び秘密鍵による暗号化が利用されており、携帯電話機やＩＣカードなどのような演算処理能力の乏しい装置では、比較的短い鍵長の公開鍵及び秘密鍵による暗号化が利用され、保証され得る安全性の範囲内でデータの送受信が行われている。

【００１３】

特に、パーソナルコンピュータなどの電子計算機では、複数種類の鍵長の公開鍵及び秘密鍵が利用可能となっており、送信するデータに応じた安全性、あるいは送信先の装置における乗算処理の処理能力などに応じて鍵長を変えて暗号化することも行われており、鍵長に汎用性を持たせることが提案されている（例えば、特許文献１参照。）。

【特許文献１】特開２００１－０５１８３２号公報

【発明の開示】

【発明が解決しようとする課題】

【００１４】

しかしながら、通常、鍵長の決定にともなって、乗算処理などの演算処理に必要となる拡大体が、所定の拡大体に特定されてしまうために、鍵長に汎用性を持たせた場合には、鍵長ごとの拡大体をあらかじめ準備しておかなければならなかった。したがって、鍵長の汎用性を高めようとすればするほど、鍵長に対応した拡大体を記憶しておくためのより大きな記憶領域が必要となっていた。

【００１５】

したがって、現実的には、準備できる記憶領域の大きさの制限から、鍵長は３～５種類程度しか準備されておらず、必ずしも十分な汎用性が提供できてはいなかった。そのため、準備された鍵長の公開鍵及び秘密鍵では安全性が十分に保証できない状態となった場合に、安全性をさらに高めることはできなかった。

【００１６】

しかも、従来の暗号化または復号化における乗算処理では、多項式積の計算の後に多項式剰余算の計算が必要であって、多項式積の計算が終了するまでは多項式剰余算の計算を開始することができなかった。したがって、このような乗算処理を演算回路を構成して実行する場合には、演算回路の並列化が困難であって、処理速度の高速化を図りにくいという問題もあった。

【００１７】

本発明者らは、このような現状に鑑み、任意な鍵長を選択可能としながら高速に乗算処理を実行可能とすることにより、利便性の高い暗号化及び復号化のシステムを構築可能とするために研究を行って、本発明を成すに至ったものである。

【課題を解決するための手段】

【００１８】

本発明の暗号化／復号化プログラムでは、素数ｐを標数とし、拡大次数ｍの拡大体F_p^mの２つの元Ａ＝$\{a_0,a_1,a_2,\cdots,a_{m-1}\}$、Ｂ＝$\{b_0,b_1,b_2,\cdots,b_{m-1}\}$を、平文データと暗号化鍵、または暗号データと復号化鍵として、電子計算機で、平文データと暗号化鍵とを乗算して暗号データの元Ｃ＝$\{c_0,c_1,c_2,\cdots,c_{m-1}\}$を生成させる、または暗号データと復号化鍵とを乗算して平文データの元Ｃ＝$\{c_0,c_1,c_2,\cdots,c_{m-1}\}$を生成させる暗号化／復号化プログラムにおいて、km+1が素数であって、F_{km+1}でｐが原始元となる正整数ｋを特定する第１のステップと、２つの元Ａ，Ｂを、正整数ｋを用いて素数ｐを標数

とする拡大次数kmの拡大体F_p^{km}における２つの元として乗算を行う第２のステップと、この乗算の結果を用いて部分体である拡大次数ｍの拡大体F_p^{m}の元における乗算の結果を求める第３のステップとを有することとした。

【００１９】

さらに、本発明の暗号化／復号化プログラムでは、

０≦ｉ≦m-1、０≦ｊ≦k-1とし、

<ｘ>がｘのmod(km+1)をとるものとして、

第２のステップが、

０≦ｔ≦k-1で、$<p^{mt}>=K[t]$となる各K[t]をそれぞれ求めるステップと、

０≦ｉ≦kmで、それぞれ０＝q[<i>]とするステップと、

０≦ｉ≦m-1で、$a_ib_i \bmod p = q[<p^i>]$をそれぞれ求めるステップと、

０≦ｉ＜ｊ≦m-1で、$(a_i-a_j)(b_i-b_j) \bmod p = M$をそれぞれ求めて、０≦ｔ≦k-1で$q[<p^i+(p^jK[t])>]$にＭをそれぞれ足し込むステップと

を有することにも特徴を有するものであり、

第３のステップが、

０≦ｉ≦m-1で、かつ１≦ｔ≦k-1で、$q[<p^i>]$に$q[<(p^iK[t])>]$をそれぞれ足し込むステップと、

０≦ｉ≦m-1で、$kq[<0>]-q[<p^i>]=c_i$をそれぞれ求めるステップと

を有することにも特徴を有するものである。

【００２０】

さらに、本発明の暗号化／復号化プログラムでは、

k=2k'の場合に、

$F_{2k'm+1}$でｐが原始元あるいは位数がk'mかつk'mが奇数となる正整数k'とし、

０≦ｉ≦m-1、０≦ｊ≦2k'-1とし、

<ｘ>がｘのmod(2k'm+1)をとるものとして、

第２のステップが、

０≦ｔ≦k'-1で、$<p^{mt}>=K[t]$となる各K[t]をそれぞれ求めるステップと、

０≦ｉ≦2k'mで、それぞれ０＝q[<i>]とするステップと、

０≦ｉ≦m-1で、$a_ib_i \bmod p = q[<p^i>]$をそれぞれ求めるステップと、

０≦ｉ＜ｊ≦m-1で、$(a_i-a_j)(b_i-b_j) \bmod p = M$をそれぞれ求めて、０≦ｔ≦k'-1で$q[<p^i+(p^jK[t])>]$にＭをそれぞれ足し込むとともに、$q[<p^i-(p^jK[t])>]$にＭをそれぞれ足し込むステップと

を有することにも特徴を有し、

第３のステップが、

０≦ｉ≦m-1で、かつ１≦ｔ≦k'-1で、$q[<p^i>]$に$q[<(p^iK[t])>]$をそれぞれ足し込むとともに、$q[<p^i>]$に$q[<-(p^iK[t])>]$をそれぞれ足し込むステップと、

０≦ｉ≦m-1で、$-q[<p^i>]=c_i$をそれぞれ求めるステップと

を有することにも特徴を有するものである。

【００２１】

また、本発明の暗号化／復号化装置では、素数ｐを標数とし、拡大次数ｍの拡大体F_p^{m}の２つの元$A=\{a_0,a_1,a_2,\cdots,a_{m-1}\}$、$B=\{b_0,b_1,b_2,\cdots,b_{m-1}\}$を、平文データと暗号化鍵、または暗号データと復号化鍵として、平文データと暗号化鍵とを乗算させて元$C=\{c_0,c_1,c_2,\cdots,c_{m-1}\}$を生成することにより暗号化する、または暗号データと復号化鍵とを乗算させて元$C=\{c_0,c_1,c_2,\cdots,c_{m-1}\}$を生成することにより復号化する演算器を備えた暗号化／復号化装置において、元をそれぞれ記憶する第１の記憶部と、拡大次数ｍを記憶する第２の記憶部と、km+1が素数であって、F_{km+1}でｐが原始元となる正整数ｋを演算器による演算に基づいて特定して記憶する第３の記憶部と、２つの元Ａ，Ｂを、正整数ｋを用いて素数ｐを標数とする拡大次数kmの拡大体F_p^{km}における２つの元として演算器で乗算した結果を記憶する第４の記憶部と、拡大次数kmの拡大体F_p^{km}の元の乗算結果を用いて演算器で所定の演算を行って、部分体である拡大次数ｍの拡大体F_p^{m}の元

における乗算の結果を求めて記憶する第５の記憶部とを有することとした。

【００２２】

さらに、本発明の暗号化／復号化装置では、

０≦ｉ≦m-1、０≦ｊ≦k-1とし、

<ｘ>がｘのmod(km+1)をとるものとして、

第４の記憶部が、

０≦ｔ≦k-1で、$<p^{mt}>=K[t]$となる各K[t]をそれぞれ求めて記憶する記憶部と、

０≦ｉ≦kmで、それぞれ０＝q[<i>]として記憶する記憶部と、

０≦ｉ≦m-1で、$a_ib_i \bmod p=q[<p^i>]$をそれぞれ求めて記憶する記憶部と、

０≦ｉ＜ｊ≦m-1で、$(a_i-a_j)(b_i-b_j) \bmod p=M$をそれぞれ求めて記憶する記憶部と、

０≦ｔ≦k-1でq[$<p^i+(p^jK[t])>$]にMをそれぞれ足し込んで記憶する記憶部と

を有することにも特徴を有し、

第５の記憶部が、

０≦ｉ≦m-1で、かつ１≦ｔ≦k-1で、q[$<p^i>$]にq[$<(p^iK[t])>$]をそれぞれ足し込んで記憶する記憶部と、

０≦ｉ≦m-1で、$kq[<0>]-q[<p^i>]=c_i$をそれぞれ求めて記憶する記憶部と

を有することにも特徴を有するものである。

【００２３】

さらに、本発明の暗号化／復号化装置では、

k=2k'の場合に、

$F_{2k'm+1}$でｐが原始元あるいは位数がk'mかつk'mが奇数となる正整数k'として、この正整数k'を記憶する記憶部を有し、

０≦ｉ≦m-1、０≦ｊ≦2k'-1とし、

<ｘ>がｘのmod(2k'm+1)をとるものとして、

第４の記憶部が、

０≦ｔ≦k'-1で、$<p^{mt}>=K[t]$となる各K[t]をそれぞれ求めて記憶する記憶部と、

０≦ｉ≦2k'mで、それぞれ０＝q[<i>]として記憶する記憶部と、

０≦ｉ≦m-1で、$a_ib_i \bmod p=q[<p^i>]$をそれぞれ求めて記憶する記憶部と、

０≦ｉ＜ｊ≦m-1で、$(a_i-a_j)(b_i-b_j) \bmod p=M$をそれぞれ求めて記憶する記憶部と、

０≦ｔ≦k'-1でq[$<p^i+(p^jK[t])>$]にMをそれぞれ足し込し込んで記憶するとともに、q[$<p^i-(p^jK[t])>$]にMをそれぞれ足し込んで記憶する記憶部と

を有することにも特徴を有し、

第５の記憶部が、

０≦ｉ≦m-1で、かつ１≦ｔ≦k'-1で、q[$<p^i>$]にq[$<(p^iK[t])>$]をそれぞれ足し込むとともに、q[$<p^i>$]にq[$<-(p^iK[t])>$]をそれぞれ足し込んで記憶する記憶部と、

０≦ｉ≦m-1で、$-q[<p^i>]=c_i$をそれぞれ求めて記憶する記憶部と

を有することにも特徴を有するものである。

【００２４】

また、本発明の乗算装置では、素数ｐを標数とし、拡大次数ｍの拡大体F_p^mの２つの元$A=\{a_0,a_1,a_2,\cdots,a_{m-1}\}$、$B=\{b_0,b_1,b_2,\cdots,b_{m-1}\}$を乗算して元$C=\{c_0,c_1,c_2,\cdots,c_{m-1}\}$を生成する演算器を備えた拡大体の乗算装置において、元をそれぞれ記憶する第１の記憶部と、拡大次数ｍを記憶する第２の記憶部と、km+1が素数であって、F_{km+1}でｐが原始元となる正整数ｋを演算器による演算に基づいて特定して記憶する第３の記憶部と、２つの元Ａ，Ｂを、正整数ｋを用いて素数ｐを標数とする拡大次数kmの拡大体F_p^{km}における２つの元として演算器で乗算した結果を記憶する第４の記憶部と、拡大次数kmの拡大体F_p^{km}の元の乗算結果を用いて演算器で所定の演算を行って、部分体である拡大次数ｍの拡大体F_p^mの元における乗算の結果を求めて記憶する第５の記憶部とを有することとした。

【発明の効果】

【００２５】

本発明では、素数ｐを標数とし、拡大次数ｍの拡大体F_p^mの２つの元Ａ＝{a_0,a_1,a_2,・・・，a_{m-1}}、Ｂ＝{b_0,b_1,b_2,・・・，b_{m-1}}を乗算して元Ｃ＝{c_0,c_1,c_2,・・・，c_{m-1}}を生成する際に、km+1が素数であって、F_{km+1}でｐが原始元となる正整数ｋを用いて、素数ｐを標数とする拡大次数kmの拡大体F_p^{km}を定義体として想定して演算を行うことにより、拡大次数ｍを任意の整数とすることができる。

【００２６】

すなわち、任意の拡大次数ｍを用いても、km+1が素数であって、F_{km+1}でｐが原始元となる正整数ｋを用いた拡大次数kmの拡大体によって、演算に必要な拡大体を極めて容易に準備できるので、拡大次数ｍごとに対応した拡大体を準備する必要がなく、拡大次数ｍを無制限に選択することができる。

【００２７】

この拡大次数ｍは、暗号化及び復号化における鍵長であって、鍵長を任意に選択することができるので、必要に応じて暗号データの安全性を任意に選択可能とすることができる。特に、安全性を高めたければ拡大次数ｍをできるだけ大きくするだけでよく、暗号データの解読行為に対して極めて容易に対応できる。

【００２８】

しかも、拡大次数kmの拡大体を定義体として用いることにより、乗算を、所定の元の加算または減算と、乗算との繰り返しに分解して演算することができ、並列処理化が容易であって、高速演算を可能とすることができる。

【００２９】

さらに、ｋ＝2k'となるk'を用いた場合には、所定の元の加算または減算と、乗算との繰り返し回数を削減できることにより、演算のさらなる高速化を図ることができる。

【図面の簡単な説明】

【００３０】

【図１】本発明に係る暗号化／復号化装置のブロック図である。

【図２】本発明に係る暗号化／復号化プログラムにおけるフローチャートである。

【図３】本発明に係る暗号化／復号化プログラムにおけるフローチャートである。

【図４】他の実施形態のフローチャートである。

【図５】既約多項式を求めるフローチャートである。

【符号の説明】

【００３１】

10　演算器
20　不揮発性記憶部
30　揮発性記憶部
40　データ入出力部
50　データバス

【発明を実施するための最良の形態】

【００３２】

本発明の暗号化／復号化プログラム、暗号化／復号化装置、及び拡大体の乗算装置では、素数ｐを標数とする拡大体を定義体とし、拡大次数ｍの拡大体F_p^mの２つの元Ａ＝{a_0,a_1,a_2,・・・，a_{m-1}}、Ｂ＝{b_0,b_1,b_2,・・・，b_{m-1}}を乗算して元Ｃ＝{c_0,c_1,c_2,・・・，c_{m-1}}を生成する際に、拡大次数ｍに対応し、km+1が素数であって、F_{km+1}でｐが原始元となる正整数ｋを用いて素数ｐを標数とする拡大次数kmの拡大体F_p^{km}を新たな定義体として乗算を行い、この乗算の結果を用いて部分体である拡大次数ｍの拡大体F_p^mの元における乗算の結果を求めている。

【００３３】

すなわち、任意の拡大次数ｍに対応して、km+1が素数であって、F_{km+1}でｐが原始元となる正整数ｋを用いることにより、素数ｐを標数とする拡大次数ｍの拡大体F_p^mを部分体とする拡大次数kmの拡大体F_p^{km}を定義体として用いることができ、拡大次数ｍ及び標数ｐを任意としても２つの元Ａ、Ｂの乗算を行うことができる。

【００３４】
ここで、拡大次数ｍは、暗号化及び復号化における鍵長に当たり、任意の拡大次数ｍが選択できるということは、鍵長を任意に選択できることを示しており、必要に応じて適宜の鍵長として暗号データを生成することができる。
【００３５】
したがって、状況に応じて鍵長を選択することにより、暗号データの安全性と、暗号化及び復号化の演算の負荷をバランスさせながら適宜の鍵長とした暗号データを生成できる。
【００３６】
例えば、標数ｐ＝2^{500}－863の500ビットの素数を用いた場合には、下表に示すように、拡大次数ｍ、パラメータとなる正整数ｋ、鍵長の組み合わせを選択できる。
【表１】

拡大次数m	2	3	4	5	・・・
パラメータｋ	1	2	1	2	・・・
鍵長（ビット）	1000	1500	2000	2500	・・・

【００３７】
ここで、鍵長が短くなる組み合わせを用いれば、暗号化及び復号化の演算速度を向上させることができ、鍵長が長くなる組み合わせを用いれば、解読困難な暗号化データを生成して安全性を向上させることができる。
【００３８】
特に、拡大次数ｍは、任意な正整数とすることができ、いくらでも大きくすることもできる。また、標数ｐも任意の素数とすることができる。
【００３９】
なお、暗号化に必要となる鍵は、Diffie－Hellman鍵交換アルゴリズムを用いて受け渡すことにより、安全に受け渡すことができる。
【００４０】
すなわち、アリスとボブが鍵交換を行う場合、まず、アリスからボブに鍵データとしてｇ，$A=g^a \in F_p^m$を送信している。
【００４１】
ボブは、Ａを受信すると、$C=A^b=g^{ab} \in F_p^m$を計算して、$B=g^b \in F_p^m$を計算し、$B=g^b$をアリスに送信している。
【００４２】
アリスは、Ｂを受信して$C=B^a=g^{ab} \in F_p^m$を計算することにより、アリスとボブとを接続するネットワーク上に鍵g^{ab}を流すことなく共有することができる。
【００４３】
なお、ＩＣカードにおける認証用データに用いる鍵の場合には、ＩＣカードの作成時にあらかじめＩＣカードに鍵データを記憶させておくことにより、ネットワーク上に鍵を流すことなく共有することができる。
【００４４】
あるいは、一般的な公開鍵暗号の場合であれば、ボブがアリスから所要のデータを受け取る際に、ボブは予め公開鍵を公開しておく。
【００４５】
例えば、Elgamal暗号の場合であれば、ボブは秘密鍵ｓを用いて、ｇ，$B=g^s \in F_p^m$を計算して公開しておく。
【００４６】
メッセージＭを送信するアリスは、ボブの公開鍵データｇ，Ｂをダウンロードして、メ

ッセージＭに対して、

$C_1 = M \times B^t$、　$C_2 = g^t$

を計算することにより暗号化を行っている。ここで、ｔは乱数である。

【００４７】

また、C_1の演算を行うに当たり、後述する本発明の暗号化／復号化プログラム、暗号化／復号化装置、あるいは拡大体の乗算装置を用いている。

【００４８】

アリスは、C_1、C_2をボブに送信し、ボブは、受信したC_1、C_2を用いて、$C_1/C_2{}^s$を演算することによりメッセージＭの復号化を行っている。すなわち、

$C_1/C_2{}^s = MB^t/g^{ts} = Mg^{ts}/g^{ts}$。

【００４９】

この復号化においても、後述する本発明の暗号化／復号化プログラム、暗号化／復号化装置、あるいは拡大体の乗算装置を用いている。

【００５０】

このようにメッセージＭの復号化は、ｓを秘密にもつボブのみが行える計算であり、対称鍵暗号のようにある一つのパスワードを互いに共有する必要がなく、かつ安全性が確保される長さの鍵を使用することで高い信頼性を有することができる。

【００５１】

図１は、本実施形態の暗号化／復号化装置のブロック図である。なお、この暗号化／復号化装置は拡大体の乗算装置でもある。

【００５２】

暗号化／復号化装置は、ＣＰＵ（中央演算装置）などの演算器10と、この演算器10が実行するプログラムなどを記憶した不揮発性記憶部20と、プログラムの実行にともなって必要となるデータを一時的に格納する揮発性記憶部30とを備えている。不揮発性記憶部20は、いわゆるＲＯＭ（Read Only Memory）あるいはハードディスクなどの不揮発性の記憶手段で構成し、揮発性記憶部30はいわゆるＲＡＭ（Random Access Memory）で構成している。

【００５３】

特に、揮発性記憶部30は、複数のレジスタを備えており、それぞれのレジスタで所要のデータを記憶している。

【００５４】

不揮発性記憶部20には、本実施形態の暗号化／復号化プログラムを記憶しており、この暗号化／復号化プログラムを起動させることにより、必要に応じて暗号化／復号化プログラムを揮発性記憶部30に展開して、暗号化あるいは復号化の演算を行っている。

【００５５】

さらに、暗号化／復号化装置にはデータ入出力部40を設けており、このデータ入力部40を介して任意の拡大次数ｍの値を入力可能とし、入力された拡大次数ｍの値に基づいて暗号化を実行可能としている。あるいは、拡大次数ｍの値だけでなく、標数ｐの値を変更可能とすることもできる。

【００５６】

また、データ入出力部40では、公開鍵となる鍵データの出力や、暗号化されたデータの入出力などを行っている。図１中、50はデータバスである。

【００５７】

演算器10では、必要に応じて、揮発性記憶部30に記憶されている所要の鍵データ、標数ｐの値、拡大次数ｍなどを不揮発性記憶部20に記憶させてもよい。

【００５８】

暗号化／復号化装置あるいは拡大体の乗算装置は、このように暗号化／復号化プログラムを実行する電子計算機などの装置に限定するものではなく、暗号化／復号化プログラムに相当する演算を実行する演算回路を備えた演算用デバイスとして、演算処理の高速化を図ることもできる。

【００５９】
以下において、暗号化あるいは復号化における２つの元の乗算について説明する。
【００６０】
まず、第１実施形態として、以下の条件下における元Ａと元Ｂの積の演算について説明する。ここで、この積の演算が暗号化である場合には、元Ａと元Ｂのいずれか一方が暗号化鍵であって、他方が暗号化鍵の鍵長ごとに分断された単位長平文データであり、積の演算が復号化である場合には、元Ａと元Ｂのいずれか一方が復号化鍵であって、他方が復号化鍵の鍵長ごとに分断された単位長暗号データである。
【００６１】
・素数ｐを標数とする拡大体を定義体とする。
・拡大次数ｍに対し、km+1が素数であって、F_{km+1}でｐが原始元となる適当な正整数ｋ。
・０≦ｉ≦ｍ－１、０≦ｊ≦ｋ－１。
・<ｘ>がｘのmod(km+1)をとるものとする。
・ωを１の原始km+1乗根とする。
・$\{a_0,a_1,a_2,\cdots,a_{m-1}\}$を拡大次数ｍの拡大体$F_p^m$の元Ａ
【数１】

$$A=\sum_{i=0}^{m-1}a_i\left(\sum_{j=0}^{k-1}\omega^{\langle p^{i+mj}\rangle}\right)=\sum_{i=0}^{m-1}a_i\gamma^{p^i},\gamma=\sum_{j=0}^{k-1}\omega^{\langle p^{mj}\rangle}$$

・$\{b_0,b_1,b_2,\cdots,b_{m-1}\}$を拡大次数ｍの拡大体$F_p^m$の元Ｂ
【数２】

$$B=\sum_{i=0}^{m-1}b_i\left(\sum_{j=0}^{k-1}\omega^{\langle p^{i+mj}\rangle}\right)=\sum_{i=0}^{m-1}b_i\gamma^{p^i}$$

【００６２】
なお、任意の正整数ｍに対して、km+1が素数となる正整数ｋが少なくともｋ＜ｍの範囲に存在していることは一般的に証明済みの事実であり、正整数ｋを特定する専用プログラムを用いて容易に見つけ出すことができる。
【００６３】
通常、元Ａと元Ｂの積Ｃ＝ＡＢは、ＣＶＭＡ（Cyclic Vector Multiplication Algorithm）で計算されるが、部分体における演算の閉性から元Ａと元Ｂの積Ｃも部分体F_p^mの元なることから、積Ｃのベクトル表現においても、基底ベクトルω'とω”のベクトル係数が等しくなる。
【００６４】
ここで、表記の便宜上、
【数３】

$$\omega'=\omega^{\langle p^i\rangle}$$

及び、
【数４】

$$\omega''=\omega^{\langle p^{i+mj}\rangle}$$

である。
【００６５】
したがって、求める必要があるのは０≦ｉ≦m-1でのω'の各係数c_iのみであり、

【数５】

$$C=\sum_{i=0}^{m-1}c_i\left(\sum_{j=0}^{k-1}\omega^{\left\langle p^{i+mj}\right\rangle}\right)=\sum_{i=0}^{m-1}c_i\gamma^{p^i}$$

として、係数c_iは、図２に示すフローチャートに基づく暗号化／復号化プログラムにより、パーソナルコンピュータなどの電子計算機での演算によって求めることができる。

【００６６】

ただし、この場合、直接、8p|m(p-1)となる拡大体を構成することはできない。そこで、8p|m(p-1)となる拡大体を構成する場合には、拡大次数ｍの偶数因数部分m_Eと奇数因数部分m_Oとに分けて（ｍ＝m_E・m_O）、それぞれを逐次拡大体に分けて拡大することにより、8p|m(p-1)となる拡大体にも対応可能とすることができる。なお、場合によっては後述する第２実施形態を組み合わせて用いることが必要となることもある。

【００６７】

暗号化の場合には、まず、電子計算機では拡大次数ｍを設定する（ステップＳ１）。なお、復号化の場合には、電子計算機は、秘密鍵または公開鍵の鍵長から拡大次数ｍを特定することもできる。

【００６８】

次いで、電子計算機は、設定された拡大次数ｍに基づいて、km+1が素数であって、F_{km+1}でｐが原始元となる適当な正整数ｋを専用プログラムを用いて特定する（ステップＳ２）。なお、前述したように、ｋ＜ｍの範囲でkm+1が素数となる正整数ｋは少なくとも１つは特定でき、電子計算機は、比較的短時間で正整数ｋの特定処理を終了することができる。

【００６９】

次いで、電子計算機は、０≦ｔ≦k-1で、$<p^{mt}>=K[t]$となる各K[t]をそれぞれ求め（ステップＳ３）、さらに、０≦ｉ≦kmで、それぞれ０＝q[<i>]とする（ステップＳ４）。

【００７０】

次いで、電子計算機は、０≦ｉ≦m-1で、$a_ib_i$modp＝q[$<p^i>$]をそれぞれ求め（ステップＳ５）、その後、０≦ｉ＜ｊ≦m-1で、$(a_i-a_j)(b_i-b_j)$modp＝Ｍをそれぞれ求めて、０≦ｔ≦k-1でq[$<p^i+(p^jK[t])>$]へのＭの足し込みをそれぞれ行う（ステップＳ６）。

【００７１】

その後、電子計算機は、拡大次数ｍの拡大体$F_p{}^m$ではなく、拡大次数kmの拡大体を定義体として用いていることによる処理として、０≦ｉ≦m-1で、かつ１≦ｔ≦k-1で、q[$<p^i>$]へのq[$<(p^iK[t])>$]の足し込みをそれぞれ行って（ステップＳ７）、最後に、０≦ｉ≦m-1で、kq[<0>]-q[$<p^i>$]＝c_iを求めている（ステップＳ８）。

【００７２】

このように、c_iは、拡大次数kmの拡大体$F_p{}^{km}$における所定の元の加算または減算と、乗算との繰り返しに分解して演算を行うことができ、並列処理化が容易であって、高速演算を可能とすることができる。

【００７３】

特に、演算の負荷が小さいことにより、携帯電話機やＩＣカードなどの演算機能の乏しい装置でも容易に乗算処理を実行でき、しかも公開鍵及び秘密鍵の鍵長の変更を可能とすることができる。

【００７４】

さらに、乗算処理における演算は、演算処理回路として半導体基板上に形成して乗算処理用半導体デバイスとすることもでき、この乗算処理用半導体デバイスを備えた暗号化／復号化装置として、暗号化及び復号化のさらなる高速化を図ることができ、しかも、並列処理化が可能であって、一層の高速化を図ることができる。また、拡大体の乗算装置として乗算処理用半導体デバイスを用いることもできる。

【００７５】

すなわち、乗算処理用半導体デバイスでは、半導体基板上に、所要の演算を実行可能とした演算器を形成するとともに、各記憶部として以下のレジスタ及びレジスタ群を形成している。

・標数ｐを記憶するレジスタ。
・拡大次数ｍを記憶するレジスタ。
・拡大次数ｍに基づく拡大体F_p^mの元Ａを記憶するレジスタ群。
・拡大次数ｍに基づく拡大体F_p^mの元Ｂを記憶するレジスタ群。
・設定された拡大次数ｍに対応した正整数ｋを特定して記憶するレジスタ。
・０≦ｔ≦k-1で、$<p^{mt}>=K[t]$となる各K[t]をそれぞれ求めて記憶するレジスタ群。
・０≦ｉ≦km-1で、それぞれ０＝q[<i>]として記憶するレジスタ群。
・０≦ｉ≦m-1で、$a_ib_i \bmod p = q[<p^i>]$をそれぞれ求めて記憶するレジスタ群。
・０≦ｉ＜ｊ≦m-1で、$(a_i-a_j)(b_i-b_j) \bmod p = M$をそれぞれ求めて記憶するレジスタ群。
・０≦ｔ≦k-1で$q[<p^i+(p^jK[t])>]$にＭをそれぞれ足し込んで記憶するレジスタ群。
・０≦ｉ≦m-1で、かつ１≦ｔ≦k-1で、$q[<p^i>]$に$q[<(p^iK[t])>]$をそれぞれ足し込んで記憶するレジスタ群。
・０≦ｉ≦m-1で、$kq[<0>]-q[<p^i>]=c_i$をそれぞれ求めて記憶するレジスタ群。

【００７６】

これらのレジスタ及びレジスタ群を設けることにより、各演算は並列化が可能であるので、並列処理を行うことにより乗算処理をさらに高速化することができる。したがって、高速な暗号化及び復号化を可能とする暗号化／復号化装置、あるいは拡大体の乗算装置を提供できる。

【００７７】

この乗算処理用半導体デバイスを所要の実装ボードに装着することにより、高速な乗算処理を可能とする実装ボードを提供でき、この実装ボードを用いて携帯電話機や電子計算機などを構成することにより、高速な暗号化及び復号化が可能であって、しかも公開鍵及び秘密鍵の鍵長の長さを自在に可変とした携帯電話機や電子計算機などを提供することができる。

【００７８】

以下において、第２実施形態として、以下の条件下における元Ａと元Ｂの積の演算について説明する。ここで、第１実施形態との違いは、ｋ＝2k'となるk'を用いて演算を行うことである。すなわち、

・素数ｐを標数とする拡大体を定義体とする。
・拡大次数ｍに対し、2k'm+1が素数であって、$F_{2k'm+1}$でｐが原始元あるいは位数がk'mかつk'mが奇数となる適当な正整数k'。
・０≦ｉ≦ｍ－１、０≦ｊ≦2k'－１。
・<ｘ>がｘのmod(2k'm+1)をとるものとする。
・ωを１の原始2k'm+1乗根として、τ＝ω＋ω^{-1}とする。
・$\{a_0, a_1, a_2, \cdots, a_{m-1}\}$を拡大次数ｍの拡大体$F_p^m$の元Ａ

【数６】

$$A=\sum_{i=0}^{m-1} a_i\left(\sum_{j=0}^{k'-1}\tau^{\langle p^{i+mj}\rangle}\right)=\sum_{i=0}^{m-1} a_i\gamma^{p^i},\ \gamma=\sum_{j=0}^{k'-1}\tau^{\langle p^{mj}\rangle}$$

・$\{b_0, b_1, b_2, \cdots, b_{m-1}\}$を拡大次数ｍの拡大体$F_p^m$の元Ｂ

【数７】

$$B=\sum_{i=0}^{m-1} b_i\left(\sum_{j=0}^{k'-1}\tau^{\langle p^{i+mj}\rangle}\right)=\sum_{i=0}^{m-1} b_i\gamma^{p^i}$$

【００７９】

部分体における演算の閉性から元Ａと元Ｂの積Ｃも部分体F_p^mの元なることから、積Ｃのベクトル表現においても、基底ベクトルτ'とτ”のベクトル係数が等しくなる。

【００８０】

ここで、表記の便宜上、

【数８】

$$\tau' = \tau^{\langle p^i \rangle}$$

及び、

【数９】

$$\tau'' = \tau^{\langle p^{i+mj} \rangle}$$

である。

【００８１】

したがって、求める必要があるのは０≦ｉ≦m-1でのτ'の係数c_iのみであり、

【数１０】

$$C = \sum_{i=0}^{m-1} c_i \left(\sum_{j=0}^{k-1} \tau^{\langle p^{i+mj} \rangle} \right) = \sum_{i=0}^{m-1} c_i \gamma^{p^i}$$

として、係数c_iは、図３に示すフローチャートに基づくプログラムにより、パーソナルコンピュータなどの電子計算機での演算によって求めることができる。

【００８２】

ただし、この場合には、4p|m(p-1)となる拡大体を構成することはできない。そこで、4p|m(p-1)となる拡大体を構成する場合には、拡大次数ｍの偶数因数部分m_Eと奇数因数部分m_Oとに分けて（ｍ＝m_E・m_O）、それぞれを逐次拡大体に分けて拡大することにより、4p|m(p-1)となる拡大体にも対応可能とすることができる。なお、場合によっては前述した第１実施形態を組み合わせて用いることが必要となることもある。

【００８３】

暗号化の場合には、まず、電子計算機では拡大次数ｍを設定する（ステップＴ１）。なお、復号化の場合には、電子計算機は、秘密鍵または公開鍵の鍵長から拡大次数ｍを特定する。

【００８４】

次いで、電子計算機は、設定された拡大次数ｍに基づいて、2k'm+1が素数であって、$F_{2^{k'm+1}}$でｐが原始元あるいは位数がk'mかつk'mが奇数となる適当な正整数k'を専用プログラムを用いて特定する（ステップＴ２）。なお、前述したように、2k'＜ｍの範囲で2k'm+1が素数となる正整数k'は少なくとも１つは特定でき、電子計算機は、比較的短時間で正整数k'の特定処理を終了することができる。

【００８５】

次いで、電子計算機は、０≦ｔ≦k'-1で、$<p^{mt}>$＝K[t]となる各K[t]をそれぞれ求め（ステップＴ３）、さらに、０≦ｉ≦2k'mで、それぞれ０＝q[<i>]とする（ステップＴ４）。

【００８６】

次いで、電子計算機は、０≦ｉ≦m-1で、$a_i b_i \bmod p = q[<p^i>]$をそれぞれ求め（ステップＴ５）、その後、０≦ｉ＜ｊ≦m-1で、$(a_i - a_j)(b_i - b_j) \bmod p = M$をそれぞれ求めて、０≦ｔ≦k'-1でq[$<p^i+(p^jK[t])>$]へのＭの足し込みをそれぞれ行うとともに、q[$<p^i-(p^jK[t])>$]へのＭの足し込みをそれぞれ行う（ステップＴ６）。

【００８７】

その後、電子計算機は、拡大次数ｍの拡大体F_p^mではなく、拡大次数k'mの拡大体を定義体として用いていることによる処理として、０≦ｉ≦m-1で、かつ１≦ｔ≦k-1で、q[$<p^i>$]へのq[$<(p^iK[t])>$]の足し込みをそれぞれ行うとともに、q[$<p^i>$]へのq[$<-(p^iK[t])>$]の

足し込みをそれぞれ行って（ステップＴ７）、電子計算機は、最後に、０≦ｉ≦m-1で、-q[<p^i>]＝c_iを求めている（ステップＴ８）。

【００８８】

このように、c_iは、所定の元の加算または減算と、乗算との繰り返しに分解して演算を行うことができ、並列処理化が容易であって、高速演算を可能とすることができる。

【００８９】

また、図３に示したフローチャートの乗算処理の演算も、演算処理回路として半導体基板上に形成して乗算処理用半導体デバイスとすることもでき、この乗算処理用半導体デバイスを備えた暗号化／復号化装置として、暗号化及び復号化のさらなる高速化を図ることができ、しかも、並列処理化が可能であって、一層の高速化を図ることができる。また、拡大体の乗算装置として乗算処理用半導体デバイスを用いることもできる。

【００９０】

この場合、乗算処理用半導体デバイスでは、半導体基板上に、所要の演算を実行可能とした演算器を形成するとともに、各記憶部として以下のレジスタ及びレジスタ群を形成している。

・標数ｐを記憶するレジスタ。
・拡大次数ｍを記憶するレジスタ。
・拡大次数ｍに基づく拡大体F_p^mの元Ａを記憶するレジスタ群。
・拡大次数ｍに基づく拡大体F_p^mの元Ｂを記憶するレジスタ群。
・設定された拡大次数ｍに対応した正整数k'を特定して記憶するレジスタ。
・０≦ｔ≦k'-1で、<p^{mt}>＝K[t]となる各K[t]をそれぞれ求めて記憶するレジスタ群。
・０≦ｉ≦2k'm-1で、それぞれ０＝q[<i>]として記憶するレジスタ群。
・０≦ｉ≦m-1で、$a_ib_i$modp＝q[<p^i>]をそれぞれ求めて記憶するレジスタ群。
・０≦ｉ＜ｊ≦m-1で、$(a_i-a_j)(b_i-b_j)$modp＝Ｍをそれぞれ求めて記憶するレジスタ群。
・０≦ｔ≦k'-1でq[<p^i+(p^jK[t])>]にＭをそれぞれ足し込し込んで記憶するとともに、q[<p^i-(p^jK[t])>]にＭをそれぞれ足し込んで記憶するレジスタ群。
・０≦ｉ≦m-1で、かつ１≦ｔ≦k'-1で、q[<p^i>]にq[<(p^iK[t])>]をそれぞれ足し込んで記憶するとともに、さらに、q[<p^i>]にq[<-(p^iK[t])>]をそれぞれ足し込んで記憶するレジスタ群。
・０≦ｉ≦m-1で、-q[<p^i>]＝c_iをそれぞれ求めて記憶するレジスタ群。

【００９１】

これらのレジスタ及びレジスタ群を設けることにより、第１の実施形態の場合と同様に、各演算は並列化が可能であるので、並列処理を行うことにより乗算処理をさらに高速化することができる。したがって、高速な暗号化及び復号化を可能とする暗号化／復号化装置、あるいは拡大体の乗算装置を提供できる。

【００９２】

さらに、km+1を素数とし、Ｓをmod km+1における位数ｋの部分巡回乗法群として、km次のＡＯＰ（All One Polynominal）（$x^{km+1}-1$）／（ｘ－１）の零点をωとして、

【数１１】

$$\gamma=\sum_{a\in S}\omega^a$$

で与えられるγを用い、次の集合を考える。

【数１２】

$$\left\{\gamma,\gamma^p,\cdots,\gamma^{p^{m-1}}\right\}$$

【００９３】

［数１２］の集合が拡大体F_p^mにおいて正規基底を成すことは、km/eがｍと互いに素となることと必要十分である。ここで、ｅはｐのmod km+1での位数である。前述した第１実

施形態及び第２実施形態は、この正規基底を用いた乗算に適用可能となっている。

【００９４】

すなわち、

・ｅをF_{km+1}における元ｐの位数とする。

・ｅ'＝ｋｍ／ｅ

・１の原始ｋ乗根$\varepsilon \in F_{km+1}$とする。

・$\{a_0, a_1, a_2, \cdots, a_{m-1}\}$を拡大次数ｍの拡大体$F_p{}^m$の元Ａ

【数１３】

$$A = \sum_{i=0}^{m-1} a_i \gamma^{p^i}$$

・$\{b_0, b_1, b_2, \cdots, b_{m-1}\}$を拡大次数ｍの拡大体$F_p{}^m$の元Ｂ

【数１４】

$$B = \sum_{i=0}^{m-1} b_i \gamma^{p^i}$$

【００９５】

このとき、部分体における演算の閉性から元Ａと元Ｂの積Ｃは、

【数１５】

$$C = AB = \sum_{i=0}^{m-1} c_i \gamma^{p^i}$$

として、係数c_iは、図４に示すフローチャートに基づくプログラムにより、パーソナルコンピュータなどの電子計算機での演算によって求めることができる。

【００９６】

ここで、暗号化の場合には、まず、電子計算機では拡大次数ｍを設定する（ステップＵ１）。なお、復号化の場合には、電子計算機は、秘密鍵または公開鍵の鍵長から拡大次数ｍを特定する。

【００９７】

次いで、電子計算機は、設定された拡大次数ｍに基づいて、km+1が素数であって、ｅ'（＝ｋｍ／ｅ）とｍとが互いに素となる適当な正整数ｋを専用プログラムを用いて特定する（ステップＵ２）。正整数ｋの特定にともなって、１の原始ｋ乗根のεを特定する（ステップＵ３）。

【００９８】

次いで、電子計算機は、１＝K[０]，０＝r[０]とする（ステップＵ４）。

【００９９】

次いで、電子計算機は、０≦ｉ≦ｍで、０＝q[<i>]とする（ステップＵ５）。

【０１００】

次いで、電子計算機は、１≦ｔ≦k-1で、<K[t-1]ε>=K[t]とする（ステップＵ６）。

【０１０１】

次いで、電子計算機は、０≦ｔ≦m-1であって、０≦ｔ≦k-1で、ｉ＋１＝r[<p^iK[t]>]とする（ステップＵ７）。

【０１０２】

次いで、電子計算機は、０≦ｔ≦m-1で、$a_i b_i$modp＝q[i+1]をそれぞれ求める（ステップＵ８）。

【０１０３】

次いで、電子計算機は、０≦ｉ＜ｊ≦m-1で、$(a_i - a_j)(b_i - b_j)$modp＝Ｍをそれぞれ求めて、０≦ｔ≦k-1でq[r[<p^i+(p^jK[t])>]]へのＭの足し込みをそれぞれ行う（ステップＵ９

）。

【０１０４】

最後に、電子計算機は、０≦ｉ≦m-1で、kq[<0>]-q[i+1]＝c_iを求めている（ステップＵ１０）。

【０１０５】

なお、ｋが偶数の場合には、ステップＵ９においてkq[<0>]＝０とすることができる。

【０１０６】

このように、より多くの正規基底に対して適用可能とすることができる。しかも、ｋの値をより小さくすることができるので、演算回数の削減による演算処理の高速化を図ることもできる。

【０１０７】

なお、前述したように、8p|m(p-1)あるいは4p|m(p-1)などのように、ｍがｐの倍数である場合には拡大体を構成できないことがあるが、拡大次数ｍを偶数因数部分m_Eと奇数因数部分m_Oとに分けて、それぞれを逐次拡大体に分けて拡大することにより、奇標数ｐに対しての拡大体の構成を可能とすることができ、上記の制限を排除することができる。

【０１０８】

すなわち、例えば、F_p^{4p}を構成する場合を考える。この場合には、最初にF_p^4の拡大体を構成し、次いで、この拡大体をｐ次逐次拡大することにより、奇標数ｐが４と互いに素であることからF_p^{4p}で表される拡大体を構成できる。すなわち、任意の奇標数及び任意次数の拡大体を構成することが可能である。

【０１０９】

本発明は数学的に見た場合には、既約多項式として法多項式を準備することなく拡大体における乗算や除算を行えるものであり、見方を変えれば本発明の考え方を利用することにより、任意次数の既約多項式を生成することができることを示している。

【０１１０】

ここで、例えばF_p^4の(1,2,1,2）というベクトル表現を持つ元はF_p^2の元である。真部分体に含まれないような元を真性元と呼ぶこととして、F_p^mの真性元をαとした場合、そのαの最小多項式、すなわちαを零点にもつF_p上の最小次数の多項式$M_\alpha(x)$が求まれば、それはF_p上のｍ次既約多項式となっている。

【０１１１】

なお、$M_\alpha(x)$は、次式で表される。

【数１６】

$$M_\alpha(x)=(x-\alpha)(x-\alpha^p)\cdots(x-\alpha^{p^{m-1}})=x^m+a_{m-1}x^{m-1}+\cdots+a_1+a,$$

$$a_0,a_1,a_2,\cdots,a_{m-1}\in F_p$$

【０１１２】

ここで、次式の計算式を考えることにする。

【数１７】

$$a_{m-1}=-Tr(\alpha)$$

【数１８】

$$a_i = (m-i)^{-1}\left\{-Tr(\alpha^{m-i}) + \sum_{j=1}^{m-1-i}(-1)a_{m-j}Tr(\alpha^{m-i-j})\right\}$$

$$Tr(x) = x + x^p + x^{p^2} + \cdots + x^{p^{m-1}}$$

【０１１３】

上式で、ｉ＝m-2,m-3,・・・,2,1,0の順に計算されることにより既約多項式が求められることとなる。

【０１１４】

すなわち、拡大体$F_p{}^m$の任意の真性元αの最小多項式は、図５に示すフローチャートに基づくプログラムにより、下のようにして求めることができる。

【０１１５】

まず、電子計算機では、真性元αのベクトル表現（$x_0,x_1,x_2,\cdots,x_{m-1}$）を設定する（ステップＷ１）。ここで、真性元αの最小多項式$M_\alpha(x)$は次式となるものとする。

【数１９】

$$M_\alpha(x) = x^m + \sum_{i=0}^{m-1} a_i x^i$$

【０１１６】

次いで、電子計算機は、１≦ｉ≦mで、T[i]＝Tr(α^i)とし（ステップＷ２）、a[m-1]=-T[1]とする（ステップＷ３）。

【０１１７】

次いで、電子計算機は、m-2≧ｉ≧０で、０＝Ｍとし、０≦ｊ≦m-1-iで、-a[m-j]Tr(α^{m-i-j})のＭへの足し込みをそれぞれ行う（ステップＷ４）。

【０１１８】

次いで、電子計算機は、a[i]=(m-i)$^{-1}${M-Tr(α^{m-i})}を求めている（ステップＷ５）。

【０１１９】

これにより、ｐ＞ｍを満たす任意の既約多項式を求めることができる。なお、前述した処理は、素体F_pではない部分体に関する最小多項式の特定に用いることもできる。

【０１２０】

このように任意の既約多項式を求めることができることによって、異なる定義体間の基底変換を可能とすることができる。

【０１２１】

基底変換を行うためには変換行列を特定する必要があるが、ここで変換行列を生成したい拡大体$F'_p{}^m$があり、これと同型の拡大体として$F_p{}^m$を考える。なお、本明細書においては、数学において通常使用されている「＾」記号付きの表現が利用できないため、以下においては「＾」の代わりに「’」を代用している。

【０１２２】

第１実施形態の方法を用いた基底変換行列の生成方法を考える。このとき、パラメータｋによって$(x^{km+1}-1)/(x-1)$という法多項式を考え、その零点ωを用いて次式で表されるγを考え、このγを用いて次式の正規基底を考える。ここで、ωは$F_p{}^{km}$に真性元として存在する。

【数２０】

$$\gamma = \sum_{i=0}^{k-1} \omega^{p^{im}}$$

【数２１】

$$\left\{\gamma, \gamma^{p}, \gamma^{p^2} \cdots, \gamma^{p^{m-1}}\right\}$$

【０１２３】

このγに相当する元γ’を拡大体$F'_p{}^m$の中から見つけることができれば拡大体$F'_p{}^m$における正規基底を特定し、基底間での対応関係から変換行列を特定することができる。この変換行列の逆行列を求めれば基底変換を可能とすることができる。

【０１２４】

ここで、元γ’を求めるためにはωに相当するω’を拡大体$F'_p{}^{km}$において特定すればよく、ω’を拡大体$F'_p{}^{km}$において特定するためには拡大体$F'_p{}^m$をｋ次逐次拡大して$F_p{}^{km}$と同型な$F'_p{}^{km}$を構成する。

【０１２５】

ここで、ｋとｍが互いに素になるようにして、

（１）k'k+1が素数である。

（２）ｐの$F_{k'k+1}$における位数がk'kである（すなわち原始元である）

ことを満たすk'を求めることができると、$(x^{k'k+1}-1)/(x-1)$を法として、$F'_p{}^m$をｋ次逐次拡大して$F'_p{}^{mk}$を構成することができる。ここで、「k'」の「'」は「＾」の代用ではない。

【０１２６】

この逐次拡大体$F'_p{}^{km}$から次式を満たす元B'を探し、ω'を求める。

【数２２】

$$B'^{\left(p^{km}-1\right)/(km+1)} \neq 1, \omega' = B'^{\left(p^{km}-1\right)/(km+1)}$$

【０１２７】

ここで、ω'は位数km+1の元となるので、$F_p{}^{km}$の位数km+1の元ωに対応することとなる。

【０１２８】

したがって、次式で与えられる元γ’により以下の集合を考えると、この集合を$F_p{}^m$の基底と対応づけることができる。これを用いて変換行列及びその逆行列を求めることができ、基底変換を可能とすることができる。

【数２３】

$$\gamma' = \sum_{i=0}^{k-1} \omega'^{p^{im}}$$

【数２４】

$$\left\{\gamma', \gamma'^{p}, \gamma'^{p^2} \cdots, \gamma'^{p^{m-1}}\right\}$$

【０１２９】

このように基底変換が可能なことから、既約多項式の因数分解を可能とすることができる。

【０１３０】

すなわち、まず、F_p上のｍ次既約多項式f(x)を考え、その零点θ'による多項式基底｛

１，θ'，θ'^{2}，・・・，θ'$^{m-1}$｝を用いて$F'_p{}^m$を構成することができる。ここで、「'」は「＾」による表示の代わりに用いている。

【０１３１】

そして、$F'_p{}^m$の元θ'を基底変換行列によって$F_p{}^m$の対応する元θに写像すれば、F_p上のｍ次既約多項式f(x)の１つの解を$F_p{}^m$上で求めたこととなる。

【０１３２】

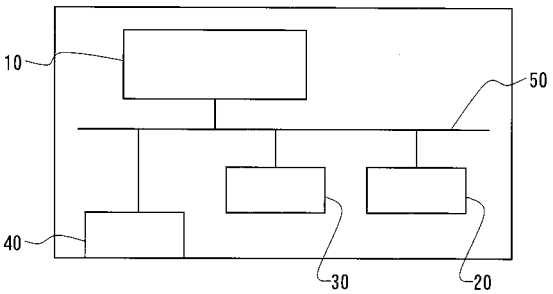
したがって、共役元まで考えれば、f(x)を$F_p{}^m$上で因数分解できることとなるので、任意の拡大体上で因数分解できることとなる。

【産業上の利用可能性】

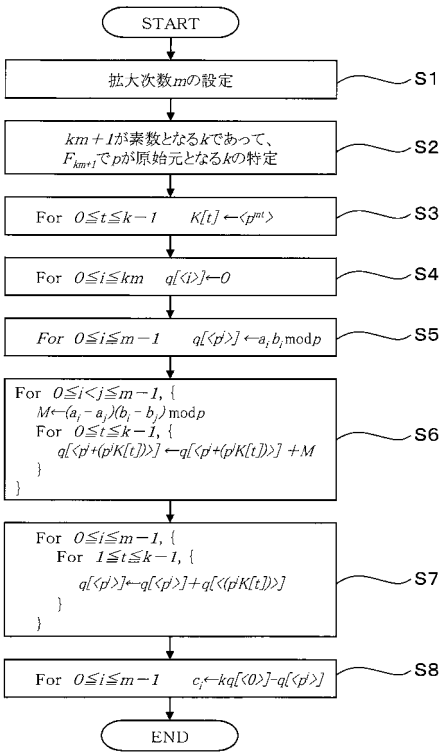
【０１３３】

パーソナルコンピュータや携帯電話、あるいはＩＣチップ付きのクレジットカードのように、他の機器とデータの送受信が行われる機器において、鍵長を任意に変更可能として、要求される安全性と、暗号化または復号化の処理負荷を調整ながら、暗号化データの送受信を可能とする。

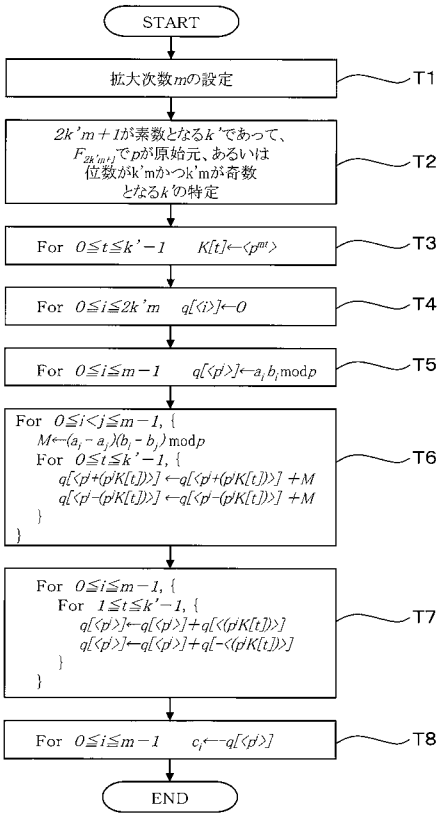
【図１】



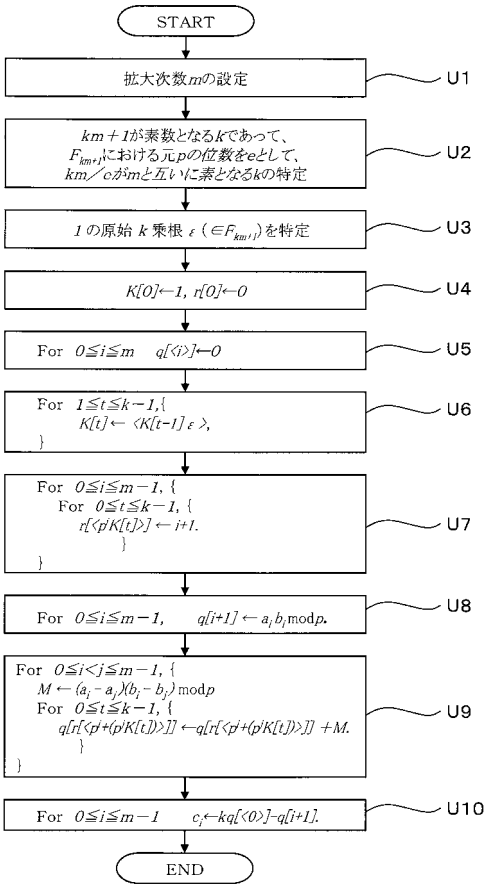
【図２】



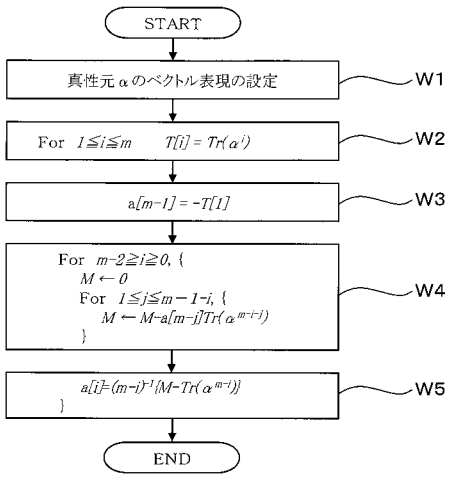
【図３】


START
拡大次数mの設定
T1
2k'm＋1が素数となるk'であって、
$F_{2k'm+1}$でpが原始元、あるいは
位数がk'mかつk'mが奇数
となるk'の特定
T2
For 0≦t≦k'−1 K[t]←<p^mt>
T3
For 0≦i≦2k'm q[<i>]←0
T4
For 0≦i≦m−1 q[<p^i>]←a_i b_i mod p
T5
For 0≦i<j≦m−1, {
M←(a_j − a_i)(b_j − b_i) mod p
For 0≦t≦k'−1, {
q[<p^i+(p^iK[t])>] ←q[<p^i+(p^iK[t])>] +M
q[<p^j−(p^jK[t])>] ←q[<p^j−(p^jK[t])>] +M
}
}
T6
For 0≦i≦m−1, {
For 1≦t≦k'−1, {
q[<p^i>]←q[<p^i>]+q[<(p^iK[t])>]
q[<p^j>]←q[<p^j>]+q[−<(p^jK[t])>]
}
}
T7
For 0≦i≦m−1 c_i←−q[<p^i>]
T8
END

【図４】


START
拡大次数mの設定
U1
km＋1が素数となるkであって、
F_{km+1}における元pの位数をeとして、
km／eがmと互いに素となるkの特定
U2
1の原始 k 乗根 ε（∈F_{km+1}）を特定
U3
K[0]←1, r[0]←0
U4
For 0≦i≦m q[<i>]←0
U5
For 1≦t≦k−1,{
K[t]← <K[t−1] ε >,
}
U6
For 0≦i≦m−1, {
For 0≦t≦k−1, {
r[<p^iK[t]>] ← i+1.
}
}
U7
For 0≦i≦m−1, q[i+1] ← a_i b_i mod p.
U8
For 0≦i<j≦m−1, {
M ← (a_j − a_i)(b_j − b_i) mod p
For 0≦t≦k−1, {
q[r[<p^i+(p^iK[t])>]] ←q[r[<p^i+(p^iK[t])>]] +M.
}
}
U9
For 0≦i≦m−1 c_i←kq[<0>]−q[i+1].
U10
END

【図５】


START
真性元αのベクトル表現の設定
W1
For 1≦i≦m T[i] = Tr(α^i)
W2
a[m−1] = −T[1]
W3
For m−2≧i≧0, {
M ← 0
For 1≦j≦m−1−i, {
M ← M−a[m−j]Tr(α^(m−i−j))
}
W4
a[i]=(m−i)^−1{M−Tr(α^(m−i))}
}
W5
END

フロントページの続き

（出願人による申告）平成１７年度、総務省戦略的情報通信研究開発推進制度「公開鍵暗号系での利用を目的とした汎用拡大体の構成に関する研究開発」に係る委託研究、産業技術力強化法第１９条の適用を受けるもの

早期審査対象出願

(56)参考文献　特開２００７－２７１７１５（ＪＰ，Ａ）
特開２００５－２８４１１１（ＪＰ，Ａ）
特開２００３－８４６６６（ＪＰ，Ａ）
特開２００２－３０４１２０（ＪＰ，Ａ）
特開２００１－５１８３２（ＪＰ，Ａ）
特開２００１－３４１６７（ＪＰ，Ａ）

(58)調査した分野(Int.Cl.，ＤＢ名)
G09C　1/00
G06F　7/72