修士論文報告書
暗号理論と素因数分解

CIPHER

1991 年 3 月 16 日

目 次

概 要

暗号はかつて、軍事や外交などの使用のみに限定されていた。

しかし、近年コンピュータによるネットワークの発達がめざましくなるに伴い、通信の機密を保持する必要が生じてきたため、ネットワークにおいても暗号が使われるようになってきた。しかし、ネットワークにおいてはこれまでのようにごく限られたユーザだけではなく、不特定多数のユーザが存在する。よってこれまでのように、任意の２人のユーザがそれぞれ暗号化・復号化鍵を保持していたのでは、ネットワークのすべてのユーザが互いに通信をするには莫大な数の鍵が必要になってしまい、とても実用的とはいえない。

そんなおり、1976 年にスタンフォード大学の Diffe,Hellman は公開鍵暗号系の概念を発表した。従来の暗号系 (秘密鍵暗号系) においては暗号化法と復号化法とは、ともに秘密にしなければならなかった。一方、公開鍵暗号系においては対となる暗号化法と復号化法とは異なっており、暗号化法は公開し、復号化法だけを秘密にするという方式をとっている。ゆえに、公開鍵暗号系では事前にコンタクトをとったり、予備的な情報を交換することなく通信を始めることが可能である。

この公開鍵暗号系を実現する方法として注目されているのが、RSA 公開鍵暗号系である。RSA 公開鍵暗号系では、その安全性の根拠として素因数分解の困難性を使用している。

そこで、この報告書では公開鍵暗号系の概略と、RSA 公開鍵暗号系の安全性の根拠となっている素因数分解法・素数判定法を説明する。

第1章　整数論

1.1　倍数と約数

a, b を整数とする。このとき $ax = b$ を満たす整数 x が存在するとき a は b の約数、b は a の倍数であるという。このような関係にある a, b を $a|b$ と書く。そうでないとき $a \not| b$ と書く。また、$p^\alpha | n$ で $p^{\alpha+1} \not| n$ のとき $p^\alpha || n$ と書く。

a, b の共通の約数のうち最大のものを a と b の最大公約数といい、$\gcd(a, b)$ または簡単に (a, b) と書く。

2 つの整数 a, b に対して $b \neq 0$ ならば

$$a = q\,b + r, \quad 0 \leq r < b$$

となる q, r は唯一つ定まる。q を商、r を余りという。

Theorem 1.1.1 (Euclid の互除法)

2 整数 $a > b > 0$ について、次のような列をつくる。

$$\begin{array}{rcll} a & = & bq_1 + r_2, & 0 < r_2 < b \\ b & = & r_2 q_2 + r_3, & 0 < r_3 < r_2 \\ r_2 & = & r_3 q_3 + r_4, & 0 < r_4 < r_3 \\ & \cdots\cdots & & \cdots\cdots \\ r_{n-2} & = & r_{n-1} q_{n-1} + r_n, & 0 < r_n < r_{n-1} \\ r_{n-1} & = & r_n q_n. & \end{array}$$

$b, r_2, r_3, \cdots$ は正の整数の減少数列となり、上の操作は有限回で終わる。このとき $(a, b) = r_n$ となる。またこのとき、Euclid の互除法より $au + bv = r_n = (a, b)$ となる u, v が計算できる。

1.2　素数、素因数分解

正の整数 n について、$n = 1 \times n$ であるから 1 と n の 2 つの約数は必ず存在する。$1, n$ 以外の正の約数を真の約数という。真の約数を持たない数を素数といい、真の約数を持つ数を合成数という。

Theorem 1.2.1 (素因数分解の一意性)

任意の整数 $n \geq 2$ は、素数 (必ずしも異なる必要はない) の積として

$$n = p_1 p_2 \cdots p_r$$

と表される。かつ、この分解は順序を無視すれば一意的である。

(注) $i \neq j$ のとき $p_i \neq p_j$ (素数) として

$$n = p_1^{e_1} p_2^{e_2} \cdots p_r^{e_r}$$

と表すこともある。

1.3 合同式

整数 a, b と n $(\neq 0)$ が与えられたとき、

$$^{\exists}k \in \boldsymbol{Z};\ a - b = kn$$

であるならば、すなわち $a-b$ が n で割りきれる (記号で $n|a-b$ と表す) ならば a と b は n を法として合同といい、

$$a \equiv b \bmod n$$

と書く。このとき次のことが成り立つ。

Proposition 1.3.1

(1) $a \equiv b \bmod n,\ b \equiv c \bmod n$ ならば $a \equiv c \bmod n$

(2) $a \equiv b \bmod n,\ c \equiv d \bmod n$ ならば $a + c \equiv b + d \bmod n,\ ac \equiv bd \bmod n$

Theorem 1.3.1 $b, c, m \in \boldsymbol{Z}$ とする。 $(c, m) = 1$ ならば

$$cx \equiv b \bmod m$$

を満たす整数 x が存在し、x は m を法として唯一つである。

Theorem 1.3.2 (中国剰余定理)
$m_1, m_2, \cdots, m_r$ $(k \geq 2)$、$i \neq j$ $(1 \leq i, j \leq r)$ ならば $(m_i, m_j) = 1$ として、$b_1, b_2, \cdots, b_r$ を任意の整数とする。このとき連立合同式

$$\begin{cases} x \equiv b_1 \bmod m_1 \\ x \equiv b_2 \bmod m_2 \\ \quad \cdots\cdots \\ x \equiv b_r \bmod m_r \end{cases}$$

は $m = m_1 m_2 \cdots m_r$ を法として唯一つの解を持つ。

Theorem 1.3.3 (Fermat の定理)
p を素数とする。このとき任意の整数 a について、

$$a^p \equiv a \bmod p$$

特に、$(a, p) = 1$ ならば上の式の両辺を a で割って

$$a^{p-1} \equiv 1 \bmod p$$

が成立する。

Euler の関数 $\varphi(n)$ とは $0 < a < n$ である整数 a のうち、$(a, n) = 1$ となる a の個数である。つまり

$$\varphi(n) = \#\{a\,;\ 0 < a < n,\ (a, n) = 1\}$$

である。Euler の関数については次のことが成り立つ。

Theorem 1.3.4 (Euler の関数)

(1) p が素数ならば $\varphi(p) = p - 1$

(2) p が素数で $e \geq 1$ のとき $\varphi(p^e) = p^e(1 - \frac{1}{p})$

(3) $n = ab,\ (a, b) = 1$ ならば $\varphi(n) = \varphi(a)\varphi(b)$

(4) $n = p_1^{e_1} \cdots p_r^{e_r}$ を n の素因数分解とすると $\varphi(n) = n \prod_{i=1}^{r} \left(1 - \frac{1}{p_i}\right)$

Theorem 1.3.5 (Euler の定理)
m が正の整数で、$(a, m) = 1$ ならば

$$a^{\varphi(m)} \equiv 1 \bmod m$$

(注) Fermat の定理は Euler の定理の特別な場合である。

1.4 平方剰余

p を素数、$(a, p) = 1$ とする。このとき $x^2 \equiv a \bmod p$ となる x が存在するとき a は p の平方剰余、存在しないとき平方非剰余といい、

$$\left(\frac{a}{p}\right) = \begin{cases} 1 & (a \text{ が } p \text{ の平方剰余のとき}) \\ -1 & (a \text{ が } p \text{ の平方非剰余のとき}) \end{cases}$$

と書く。$\left(\frac{a}{p}\right)$ のことを Legendre 記号という。また $(a, p) \neq 1$ のとき

$$\left(\frac{a}{p}\right) = 0$$

と書く。

Legendre 記号については次のことが成り立つ。

Theorem 1.4.1 (Legendre 記号)

(1) $\left(\frac{1}{p}\right) = 1$

(2) $a \equiv b \bmod p \Longrightarrow \left(\frac{a}{p}\right) = \left(\frac{b}{p}\right)$

(3) $\left(\dfrac{ab}{p}\right) = \left(\dfrac{a}{p}\right)\left(\dfrac{b}{p}\right)$

p を奇素数としたとき

(4) $\left(\dfrac{-1}{p}\right) = \begin{cases} 1 & p \equiv 1 \bmod 4 \\ -1 & p \equiv 3 \bmod 4 \end{cases}$ (**第一補充法則**)

(5) $\left(\dfrac{a}{p}\right) \equiv a^{(p-1)/2} \bmod p$ (**Euler の規準**)

(6) $\left(\dfrac{2}{p}\right) = \begin{cases} 1 & p \equiv 1 \text{ or } 7 \bmod 8 \\ -1 & p \equiv 3 \text{ or } 5 \bmod 8 \end{cases}$ (**第二補充法則**)

$q\,(\neq p)$ を奇素数としたとき

(7) $\left(\dfrac{q}{p}\right) = \begin{cases} \left(\dfrac{p}{q}\right) & p \equiv 1 \text{ or } q \equiv 1 \bmod 4 \\ -\left(\dfrac{p}{q}\right) & p \equiv q \equiv 3 \bmod 4 \end{cases}$ (**平方剰余の相互法則**)

次に Legendre 記号を拡張して Jacobi 記号を定義する。
奇数 $m = p_1^{e_1} \cdots p_r^{e_r},\ (a, m) = 1$ に対して

$$\left(\frac{a}{m}\right) = \left(\frac{a}{p_1}\right)^{e_1}\left(\frac{a}{p_2}\right)^{e_2}\cdots\left(\frac{a}{p_r}\right)^{e_r}$$

と定義する。これを Jacobi 記号という。

(注) $\left(\frac{a}{m}\right) = 1$ であることは a が m の平方剰余であることを意味しているわけではない。

Theorem 1.4.2 (Jacobi 記号)

(1) $\left(\dfrac{1}{m}\right) = 1$

(2) $a \equiv b \bmod m \Longrightarrow \left(\dfrac{a}{m}\right) = \left(\dfrac{b}{m}\right)$

(3) $\left(\dfrac{ab}{m}\right) = \left(\dfrac{a}{m}\right)\left(\dfrac{b}{m}\right)$

(4) $\left(\dfrac{-1}{m}\right) = \begin{cases} 1 & m \equiv 1 \bmod 4 \\ -1 & m \equiv 3 \bmod 4 \end{cases}$

(5) $\left(\dfrac{2}{m}\right) = \begin{cases} 1 & m \equiv 1 \text{ or } 7 \bmod 8 \\ -1 & m \equiv 3 \text{ or } 5 \bmod 8 \end{cases}$

奇数 $n,\ (n, m) = 1$ に対して

(6) $$\left(\frac{n}{m}\right) = \begin{cases} \left(\frac{m}{n}\right) & m \equiv 1 \text{ or } n \equiv 1 \bmod 4 \\ -\left(\frac{m}{n}\right) & m \equiv n \equiv 3 \bmod 4 \end{cases}$$

(注) (4), (5), (6) を使うことで Jacobi 記号は $\log m$ に比例した時間で計算できる。
詳しい証明等については、参考文献 [1],[10],[12],[15] を参照のこと。

1.5 繰り返し 2 乗法

繰り返し 2 乗法とは、効率的に $b^m \bmod n$ を計算するための方法である。このときのアルゴリズムは次のようになる。

まず、m を 2 進数展開する。つまり

$$m = m_0 + 2m_1 + 4m_2 + \cdots + 2^{k-1}m_{k-1} \quad (m_j = 0 \text{ or } 1)$$

とする。次に、順次

$$b^2\,,\ (b^2)^2 = b^4\,,\ (b^4)^2 = b^8\,,\cdots,\ (b^{2^{k-2}})^2 = b^{2^{k-1}}$$

を計算する。すると

$$\begin{aligned} b^m &= b^{m_0+2m_1+4m_2+\cdots+2^{k-1}m_{k-1}} \\ &= b^{m_0}\cdot b^{2m_1}\cdot b^{4m_2}\cdots b^{2^{k-1}m_{k-1}} \\ &= b^{m_0}\cdot (b^2)^{m_1}\cdot (b^4)^{m_2}\cdots (b^{2^{k-1}})^{m_{k-1}} \end{aligned}$$

となり、$b^2\,,\ (b^2)^2 = b^4\,,\ (b^4)^2 = b^8\,,\cdots,\ (b^{2^{k-2}})^2 = b^{2^{k-1}}$ はすでに計算してあるので、$b^m \bmod n$ の値は簡単に求めることができる。

第2章　公開鍵暗号系

2.1　公開鍵暗号系

1976 年にスタンフォード大学の Diffie,Hellman は公開鍵暗号の概念を発表した。従来の暗号系においては暗号化法 (鍵) と復号化法 (鍵) とは、ともに秘密にしなければならなかった。

一方、公開鍵暗号においては対となる暗号化法 (鍵) と復号化法 (鍵) とは異なっており、暗号化法 (鍵) は公開し、復号化法 (鍵) だけを秘密にするという方式をとっている。

公開鍵暗号は、次のような特徴を持っている。

(1)　鍵の配送が簡単である。
(2)　秘密にする鍵の数が少ない。
(3)　安全な認証が可能である (デジタル署名)。

従来の暗号系では暗号化法 (鍵) と復号化法 (鍵) の計算量はほぼ同じくらいであったため暗号化法 (鍵) を知られてしまうと、復号化法 (鍵) を計算することが可能であった。しかし、公開鍵暗号では一方向性関数の性質をうまく使って、暗号化法 (鍵) を公開しても復号化法 (鍵) を計算するのは困難であるという事実を使っている。

2.1.1　一方向性関数

公開鍵暗号は、暗号化の方法しか知らないのであれば、復号化の方法を見つけるのは事実上不可能であるという性質を使っている。言い換えると暗号化鍵 K_E が分かっていれば、暗号化 $f : P \to C$ はたやすく実行できるが復号化 $f^{-1} : C \to P$ を実行するのは事実上不可能であるということだ。このような関数 f を一方向性関数 (one way function) と呼ぶ。しかし、一方向性関数は数学的に厳密に定義されるものではなく経験的なものであり、現在一方向性関数と思われている関数がコンピュータなどのテクノロジーの進歩、計算スピードをアップする新しいアルゴリズムの出現などによっては、数十年後にはその地位をなくしているかも知れない。

一方向性関数 f とは次の性質を満たすものである。

(a)　f に対して任意の元 x が与えられたとき、$y = f(x)$ を計算するのは容易である。

(b)　$y = f(x)$ なる y が与えられたとき $x = f^{-1}(y)$ を計算して、x を求めるのは (現実的には) 不可能である。

2.1.2 一方向性落とし戸関数

一方向性関数はそのままでは暗号として使用できない。正規の受信者であろうとなかろうと f^{-1} を計算することができないので、復号化ができないからである。そこで、一方向性関数に対して、落とし戸 (trapdoor) と呼ばれる性質をつけ加えて一方向性落とし戸関数を作り出す。

一方向性落とし戸関数とは次のような性質を満たす関数である。

(a) f に対して任意の元 x が与えられたとき、$y = f(x)$ を計算するのは容易である。

(b) $y = f(x)$ なる y が与えられたとき $x = f^{-1}(y)$ を計算して、x を求めるのは「ある情報」を知っていれば容易であるが「ある情報」を知っていなければ (現実的には) 不可能である。

正規の受信者はこの「ある情報」、つまり復号化鍵を知ることにより f^{-1} を簡単に計算することができ、復号化を達成することができる。

2.2 認証

公開鍵暗号系においては暗号化の方法が公開されているので、そのメッセージが確かにその人から送られたという確認が絶対的に必要となる。この確認のことを認証と呼ぶ。

もし普通の手紙等であれば手書きのサイン、印章などで確認ができる。しかし通信においてはそのような手段がとれないため、通信を認証するためのまったく別な手段を考えなければならない。公開鍵暗号系では、認証のための次のような手段が提案されている。

A,B を公開鍵暗号系の 2 人のユーザとする。f_A を暗号系のユーザが Alice にメッセージを送るときの暗号化の変換、f_B をシステムのユーザが Bob にメッセージを送るときの暗号化の変換とする。

P を Alice の署名とする (手書きの署名と区別してこれはデジタル署名と呼ばれ、ID 番号、メッセージを送った時間などからなる)。Alice が Bob にメッセージ M を送ろうとしているとする。公開鍵暗号系においては暗号化の方法が公開されているので

$$f_B(M)\,,\ f_B(P)$$

を Bob に送っただけでは充分とは言えない。Alice は Bob 以外にも署名 P を送っているわけである。ゆえに第三者でも P を偽造することができるからである。そこで Alice は $f_B(M)$ と $f_B(P)$ ではなく、メッセージの終わりに $f_B f_A^{-1}(P)$ をつけ加えて

$$f_B(M)\,,\ f_B f_A^{-1}(P)$$

を送る。ただし M にはこの文章が Alice から送られたということをつけ加える。Bob はこのメッセージを受け取ったら f_B^{-1} を使って復号化する。すると

$$f_B^{-1} f_B(M) = M\,,\ f_B^{-1} f_B f_A^{-1}(P) = f_A^{-1}(P)$$

より M と $f_A^{-1}(P)$ を得ることができる。M にはこの文章が Alice から送られたということが書かれているので、Bob は f_A を使って P を得る。Alice 以外には f_A^{-1} を知り得ないので、確かにそのメッセージは Alice から送られたものであるということが確認される。

2.3 コード化

公開鍵暗号系においては、メッセージを数値に変換することが必要となる。このような方法はいろいろと考えられるが、例えば次のようにすることが考えられる。

(i) いま、n 文字からなるメッセージはアルファベット A–Z とスペース␣の 27 文字からなるとする。このとき、A–Z に対しては 0–25、スペース␣に対しては 26 を対応させる。そして、n 文字からなるメッセージを n 桁からなる 27 進数と見て、それを 10 進数表示にする。

例

$$\text{cipher} \Longleftrightarrow 2 \cdot 27^5 + 8 \cdot 27^4 + 15 \cdot 27^3 + 7 \cdot 27^2 + 4 \cdot 27 + 17 = 33249815$$

(ii) 日本語のメッセージのときは、JIS (日本工業規格) が制定している漢字体系のうち区点コード (10 進数) を使うことが考えられる。

例 区点コードより、暗 (1637)、号 (2570) であるから

$$\text{暗号} \Longleftrightarrow 16372570$$

数値を日本語にもどすときは、4 桁ずつ区切ってそれぞれに対応した漢字を見つければよい。

2.4 RSA 公開鍵暗号

Rivest,Shamir,Adleman は公開鍵暗号を実現するための方法を考案した。この暗号は彼ら 3 人の名前にちなんで RSA 公開鍵暗号またはたんに RSA 暗号と呼ばれている。RSA 暗号の安全性は巨大な整数の素因数分解の困難さに基づいている。RSA 暗号は公開鍵暗号の中でも、もっとも有力であると思われている。

2.4.1 RSA 公開鍵暗号

RSA 暗号のユーザ A は巨大な素数 p_A, q_A (10 進数表示で 100 桁ほど) を選ぶ。そして $n_A = p_A q_A$ とする。n_A の素因数分解は分かっているので $\varphi(n_A) = (p_A - 1)(q_A - 1) = n_A + 1 - p_A - q_A$ はすぐに計算できる。次にユーザは $1 < e_A < \varphi(n_A)$, $(\varphi(n_A), e_A) = 1$ となる整数 e_A をランダムに選び、$d_A \equiv e_A^{-1} \bmod \varphi(n_A)$ を計算する。そして $K_E = (n_A, e_A)$ を暗号化鍵として公開し、$K_D = (n_A, d_A)$ は復号化鍵として秘密にしておく。

平文を P, 暗号文を C とする。このとき暗号化 f、復号化 f^{-1} のアルゴリズムはそれぞれ

$$C = f(P) = P^{e_A} \bmod n_A$$

$$P = f^{-1}(C) = C^{d_A} \bmod n_A$$

となる。

この 2 つの写像は互いに逆写像になる。なぜなら

$$f^{-1}f(P) = (P^{e_A})^{d_A} = P^{e_A d_A} \tag{2.1}$$

しかし、e_A, d_A の選び方より $e_A d_A \equiv 1 \bmod \varphi(n_A)$ なので (2.1) は Euler の定理より

$$f^{-1}f(P) = (P^{e_A})^{d_A} = P^{e_A d_A} \equiv P \bmod n_A$$

となる。(**Theorem1.3.5** 参照)

例　2 つの素数 $p = 4129, q = 2531$ を選ぶ。このとき $n = 4129 \times 2531 = 10450499$, $\varphi(n) = (p-1)(q-1) = 4128 \times 2530 = 10443840$ となる。$e = 494311$ と選んでくると $(e, \varphi(n)) = (494311, 10443840) = 1, d \equiv e^{-1} \equiv 1163671 \bmod 10443840$ である。

このとき peace という文字を送る。peace を 27 進数と見て数値化すると、

$$\text{peace} \Longleftrightarrow 15 \times 27^4 + 4 \times 27^3 + 0 \times 27^2 + 2 \times 27 + 4 = 8050405$$

となる。このとき送信すべき整数は

$$9456594 \equiv 8050405^{494311} \bmod 10450499$$

である。これを復号化するには $d = 1163671$ 乗して

$$8050405 \equiv 9456594^{1163671} \bmod 10450499$$

となる。

もし、n_A の素因数分解ができれば、$\varphi(n_A)$ はすぐに計算できるので公開された e_A より d_A はすぐに計算できる。しかし、n_A が 200 桁の場合には素因数分解は現実的には不可能である。よって素因数分解の困難さより、RSA 暗号の安全性が保障される。

2.5　ナップザック暗号

公開鍵暗号の具体的アルゴリズムとして、Merkle,Hellman は 1978 年にナップザック暗号を発表した。この暗号は発明者にちなんで、MH 型ナップザック暗号と呼ばれている。この暗号を説明する前にナップザック問題についてふれておく。

2.5.1　ナップザック問題

k 個の正の整数からなるベクトル $(v_1, v_2, \cdots, v_k)$ と正の整数 V が与えられたとき、

$$\sum_{i=1}^{k} \varepsilon_i v_i = V \ , \ \varepsilon_i \in \{0\,,\,1\} \qquad (1 \le i \le k)$$

となるベクトル $n = (\varepsilon_1, \varepsilon_2, \cdots, \varepsilon_k)$ が存在するかどうか判定し (判定問題)、存在するのであればすべての解を求めよ (探索問題)。

このような一般的ナップザック問題は k のサイズが大きくなると解くことが非常に難しい問題であることが知られていて、NP 完全問題と呼ばれるものの一つである。

ナップザック問題の特別のケースが超増加ナップザック問題である。超増加ナップザック問題では、ベクトルの各要素 v_i , $(i = 1, 2, \cdots, k)$ が

$$v_i > \sum_{j=1}^{i-1} v_j \qquad (i = 2, \cdots, k)$$

という条件を満たす超増加数列になっている。

この条件を満たしたナップザック問題では、

$$\varepsilon_k = 1 \Longleftrightarrow V \geq v_k$$

という同値関係が成立し、$i = k-1, k-2, \cdots, 1$ に対し、

$$(V - \sum_{j=i+1}^{k} \varepsilon_j v_j) \geq v_i \Longleftrightarrow \varepsilon_i = 1$$

が成立する。よって、超増加ナップザック問題の場合は簡単に解くことができる。

2.5.2 MH 型ナップザック暗号

まず、各ユーザは超増加となるベクトル $(v_1, v_2, \cdots, v_k)$ と $m \geq \sum_{i=1}^{k} v_i$ を満たす m を選ぶ。次に $(a, m) = 1$, $0 < a < m$ を満たす a を選ぶ。そして、

$$b \equiv a^{-1} \bmod m$$

を計算し、$w_i \equiv a v_i \bmod m$ で定義されるベクトル $(w_1, \cdots, w_k)$ を計算する。

ユーザは $(v_1, v_2, \cdots, v_k), m, a, b$ を秘密にし、暗号化鍵 $K_E = (w_1, \cdots, w_k)$ を公開する。このとき復号化鍵は $K_D = (b, m)$ となる。

いま、k-bit からなる平文メッセージ $P = (\varepsilon_1, \varepsilon_2, \cdots \varepsilon_k)$ を $K_E = (w_1, \cdots, w_k)$ を使って送ろう。それには $C = \sum_{i=1}^{k} \varepsilon_i w_i$ を計算して、その整数を送る。

暗号メッセージを読むには、

$$b w_i \equiv b a v_i \equiv v_i \mod m$$

であることに注意すれば

$$bC \equiv \sum_{i=1}^{k} \varepsilon_i b w_i \equiv \sum_{i=1}^{k} \varepsilon_i v_i \mod m$$

である。ゆえに $0 < bC \bmod m \leq m$ であるように選べば $0 < \sum_{i=1}^{k} \varepsilon_i v_i \leq \sum_{i=1}^{k} v_i \leq m$ より

$$bC = \sum_{i=1}^{k} \varepsilon_i v_i$$

となる。そうすれば、超増加ナップザック問題を解くことで一意的な解 $(\varepsilon_1, \cdots, \varepsilon_k)$ が見つかる。$(w_1, \cdots, w_k)$ しか知らない人は一般的ナップザック問題に直面することになる。

しかし、ナップザック暗号系に対する安全性については当初から疑問がもたれていた。それは、変換によって超増加ナップザック問題ではなくなったとはいえ、$(w_1, \cdots, w_k)$ は非常に特殊なベクトルだからだ。実際 1982 年に Shamir は k の多項式時間でこのタイプのナップザック問題を解くアルゴリズムを提案した。(参考文献 [13])

ゆえに、Markle-Hellman のオリジナルのナップザック暗号は安全であるとはいえない。安全性を高めるため変換を繰り返して行う反復 HM 型ナップザック暗号も提案されたがこれも Adleman と Brickell によって解読されている。

2.6 鍵配送システム

2.6.1 離散的対数問題

RSA 暗号においては巨大な整数の素因数分解の困難さを、ナップザック暗号においてはナップザック問題の解法の困難さを利用した。

このような一方向性の性質を持つもう一つのプロセスが大きな有限体における離散的対数問題である。実数においては対数を計算するのは大きな数でも簡単であるが、大きな有限体上での離散的対数問題は極めて難しい。離散的対数問題は次のように書き表せる。

Definition 2.6.1 G : 有限体、$b \in G,\ y \in \langle b \rangle$ とする。このときベース b に対する y の離散的対数 ($\log_b y$ とかく) とは $b^x = y$ となる x のことである。

一方、実数体上においても有限体上においても指数べきを計算することは繰り返し 2 乗法を用いることによって簡単に計算することができる。

2.6.2 Diffe-Hellman の鍵配送システム

鍵配送法は公開鍵を使って任意の二人のユーザが秘密鍵を共有する方法である。秘密鍵を取り決める場合、秘密鍵をそのまま送ることは危険が伴う。しかし、鍵配送法を使うと安全でない通信路を使ってもユーザ同士が共有の秘密鍵を作り出すことができる。鍵配送法は秘密通信機能と認証機能がないので公開鍵暗号とは異なる。

鍵配送システムの具体的方法は 1976 年 Diffe-Hellman によって提案された。この方法は大きな有限体上での離散的対数問題が難しいことを利用している。その方法は次のようになる。

大きな有限体 F_q に含まれる共有鍵をつくるとする。このとき q は公開しておく。よって誰でも鍵がどのような有限体にあるのか知ることはできる。また g を F_q の生成元とする。このとき Aïda と Bernado の二人が共有鍵をつくるには次のようにする。

Aïda は 1 から $q-1$ までのランダムな整数 a を選び秘密にし、g^a を公開する。Bernado も同様に 1 から $q-1$ までのランダムな整数 b を選び秘密にし、g^b を公開する。このとき二人が使う共有鍵は g^{ab} とする。二人のユーザはこの鍵を計算することができる。例えば Aïda は Bernado が公開

した g^b を a 乗すればよい。しかし第三者は g^a と g^b しか知り得ない。ゆえに離散的対数問題を解くことなしには共有鍵を計算することはできない。

公開鍵暗号を用いても秘密鍵の共有はできるが、暗号化鍵と復号化鍵の生成に時間がかかる。公開鍵配送法を用いると、認証機能はないが、前もって生成する鍵は任意でよいので、簡単に秘密鍵の共有ができる。つまり、鍵配送システムにおいては一方向性関数のみをうまく使っているわけである。(逆にいうと、落とし戸を仕掛けることができないわけである。)

第3章　素数判定法

3.1　素数判定法と素因数分解法

素因数分解 (factoring) の問題は次のように述べることができる。

- 整数 n が与えられたとき、$n = pq$ となるような 1 より大きな整数 p, q を求めよ。

これには 2 つの大きな問題がある。1 つは n が素数であるか判定する問題 (素数判定法) であり、もう 1 つは p と q を求める問題 (素因数分解法) である。素数判定法と素因数分解法はまったく別物である。素数判定法とはある与えられた数 n が素数か否か (つまり合成数か) を判定するものである。

さらに、素数判定法は決定論的 (確定的) 素数判定法と確率的素数判定法がある。確定的素数判定法は、素数であれば確実に素数であると判定できるもので、確率的素数判定法とは素数である確率が極めて高いと判定するものである。

これに対して素因数分解法とは素数判定法で合成数と判定された数の実際の素因数を見つけるものである。

現在は 100 桁ほどの数でも素数判定だけならば大型計算機によってほんの数分で判定できる。しかし素数判定法を用いて合成数であるとわかったとしても、素因数分解法となると (特殊な例をのぞいては)100 桁の場合は極めて困難である。

これまでの研究においては多項式時間で素因数分解を行うアルゴリズムは発見されておらず、おそらく素因数分解の問題は NP 問題であるだろうと予測されている。

3.2　試行割算法

試行割算法 (Trial Division) は最も簡単な素数判定法かつ素因数分解法である。これは $d = 2, 3, 4, 5, \cdots$ で順番に n を割っていくだけの方法である。そのとき次のいずれかになる。

(1)　$d > \sqrt{n}$ かつ $^{\forall}d' < d$ について $d' \not| n$　このとき n は素数

(2)　$d < n$ かつ $d | n$　このとき d は n の自明でない素因数

n が 12,13 桁までならばこの方法が最も効果的だ。しかしこのままでは効率が悪いので次のように改良することができる。

例えば、n が奇数のときは奇数の d についてのみ考えればよい。また n が $n \neq 0 \bmod 3$ ならば 3 の倍数については考えなくてよい。

この改良を続けると d として考えればよいのは $d < \sqrt{n}$ となる素数 d についてのみということが言える。しかしこれは事前に素数の集合が分かっていなければ使えない。

3.3 Fermat 法

与えられた n が素数かどうか判定する簡単な方法は、よく知られた Fermat の定理を使うものである。(**Theorem1.3.3** 参照)

つまり、素数かどうか判定したい n に対して $(a,n)=1$ となる任意の a について

$$a^{n-1} \equiv 1 \bmod n \tag{3.1}$$

となるかどうか計算すればよい。もし (3.1) が成り立たない a が見つかれば n は合成数と判断してよい。

また、もし $(a,n)>1$ となったら n が合成数と分かるのと同時に n の素因子が見つかったことになる。

- **擬似素数**

n を合成数とする。$(a,n)=1$ である a について (3.1) が成り立つとき、n を a を底とする擬似素数という (擬似素数は無限個あることが証明されている)。

例　$5^{216} \equiv 1 \bmod 217$ であるが、$217 = 7 \times 31$ である。つまり 217 は 5 を底とする擬似素数である。

- **Carmichael 数**

逆に $(a,n)=1$ となる任意の a について (3.1) が成り立てば n が素数であると言えるであろうか。答は No である。

例えば

$$n = 561 = 3 \cdot 11 \cdot 17$$

は $(a,561)=1$ である任意の a に対して (3.1) が成り立つ。

なぜなら、Fermat の定理よりそれぞれ $(a,3)=1,\ (a,11)=1,\ (a,17)=1$ であるような a に対して

$$a^{560} \equiv (a^{3-1})^{280} \equiv 1^{280} \equiv 1 \bmod 3$$
$$a^{560} \equiv (a^{11-1})^{56} \equiv 1^{56} \equiv 1 \bmod 11$$
$$a^{560} \equiv (a^{17-1})^{35} \equiv 1^{35} \equiv 1 \bmod 17$$

ゆえに、中国剰余定理を使うと (**Theorem1.3.2** 参照) $(a,\ 3\cdot 11\cdot 17) = (a,561) = 1$ である a について

$$a^{560} \equiv 1 \bmod 561$$

このような数を Carmichael 数と呼ぶ。(561 は最小の Carmichael 数である。近年コンピュータにより、Carmichael 数が数千個発見され、無限個存在するのであろうと予測されているが、まだ証明はされていない。) ゆえに Carmichael 数の場合には幸運に $(a,n)>1$ となる a に巡り会わない限り n が合成数とは分からない。よって Fermat 法だけでは n を素数と判断することはできない。

3.4 Solovay–Strassen 法

Fermat の定理では Carmichael 数のようなものが存在するため素数判定法の改良を行う。それは Solovay–Strassen 法と呼ばれるもので Euler の規準 (**Theorem1.4.1** (5) 参照) を使う。

p は奇素数 とする。そのとき

$$b^{(p-1)/2} \equiv \left(\frac{b}{p}\right) \bmod p \tag{3.2}$$

証明 もし、$p|b$ であれば両辺ともに 0 となる。ゆえに $p\not|b$ を仮定。Fermat の定理より $b^{p-1} \equiv 1 \bmod p$ だから $b^{(p-1)/2} \equiv \pm 1 \bmod p$ となる。

g を F_p^* の生成元とし、$b = g^j$ とする。すると

$$j : \text{偶数} \Longleftrightarrow \left(\frac{b}{p}\right) = 1$$

である。そして

$$b^{(p-1)/2} = g^{j(p-1)/2} = 1 \Longleftrightarrow (p-1) \,\Big|\, j(p-1)/2 \Longleftrightarrow j : \text{偶数}$$

両辺ともに ± 1 にしかならないので j : 奇数 のとき両辺ともに -1 がいえる。

(証明終)

次に Euler の擬似素数を定義する。

- **Euler の擬似素数**

n を奇数合成数としたとき $(b, n) = 1$ となる b が (3.2) を満たせば n を b を底とする Euler の擬似素数という。

もし、n が合成数ならば $(b, n) = 1$ となる b の少なくとも 50 % について (3.2) は成り立たない。よって Carmichael 数のようなものは存在しない。

いまから n が奇数合成数ならば $(b, n) = 1$ である b の少なくとも 50%について (3.2) が成立しないことを次のようにして示す。

Proposition 3.4.1 b_1 について (3.2) が成立、b_2 について (3.2) が不成立であるとする。つまり

$$b_1^{(n-1)/2} \equiv \left(\frac{b_1}{n}\right) \bmod n \tag{3.3}$$

$$b_2^{(n-1)/2} \not\equiv \left(\frac{b_2}{n}\right) \bmod n \tag{3.4}$$

このとき次のことが成り立つ。

$$(b_1 b_2)^{(n-1)/2} \not\equiv \left(\frac{b_1 b_2}{n}\right) \bmod n$$

証明 もし

$$(b_1 b_2)^{(n-1)/2} \equiv \left(\frac{b_1 b_2}{n}\right) \bmod n$$

とすると (3.3) で両辺を割って (Jacobi 記号の乗法性を使って)

$$b_2^{(n-1)/2} \equiv \left(\frac{b_2}{n}\right) \bmod n$$

よって矛盾。 **(証明終)**

いま $\{b_1, b_2, \cdots, b_s\}$ を (3.2) が成り立つ数の集合とする。もし $b^{(n-1)/2} \not\equiv \left(\frac{b}{n}\right)$ となる b が存在すれば $\{bb_1, bb_2, \cdots, bb_s\}$ については (3.2) が成り立たない。ゆえに $b^{(n-1)/2} \not\equiv \left(\frac{b}{n}\right)$ となる b が 1 つでも存在すれば少なくとも 50%について (3.2) は成り立たない。よってこのような b が存在することを示す。

Proposition 3.4.2 $p^2|n$ となる素数 p が存在すれば、$b^{(n-1)/2} \not\equiv \pm 1 \bmod n$ となる b が存在する。

証明 $b = 1 + n/p$ とすると $\left(\frac{b}{n}\right) = 1$ である。なぜなら $p^2|n$ であるから $n = n'p^2$ とすると

$$b = 1 + n/p = 1 + n'p \equiv 1 \bmod p,\ b = 1 + n/p = 1 + n'p \equiv 1 \bmod n'$$

である。ゆえに

$$\left(\frac{b}{n}\right) = \left(\frac{b}{n'}\right)\left(\frac{b}{p}\right)^2 = 1$$

次に、$(1 + n/p)^j \equiv 1 \bmod n$ とする。左辺は

$$1 + j\frac{n}{p} + \binom{j}{2}\frac{n^2}{p^2} + \cdots \equiv 1 + \frac{jn}{p} \bmod n$$

であるのでこれが $\bmod n$ で 1 となるには $p|j$ でなければならない。しかし、
$p \not| (n-1)/2$ である。なぜなら $^\exists k \in \boldsymbol{Z};\ kp = (n-1)/2$ とすれば、

$$2kp = n - 1 = n'p^2 - 1 \Longrightarrow n'p^2 - 2kp = 1 \Longrightarrow p(n'p - 2k) = 1$$

よって矛盾 **(証明終)**

Proposition 3.4.3 $n = p_1 p_2 \cdots p_l$ ($p_1, p_2, \cdots, p_l$: 異なる素数) とする。このとき $\left(\frac{b}{p_1}\right) = -1,\ b \equiv 1 \bmod n/p_1$ を満たす b が存在し、この b について (3.2) は成り立たない。

証明

$$\left(\frac{b}{n}\right) = \left(\frac{b}{p_1}\right)\left(\frac{b}{n/p_1}\right) = (-1) \times 1 = -1$$

また $b = 1 + k(n/p_1)$ とすると

$$b^{(n-1)/2} = (1 + k(n/p_1))^{(n-1)/2} \equiv 1 \bmod n/p_1$$

よって

$$b^{(n-1)/2} \not\equiv \left(\frac{b}{n}\right) \bmod n/p_1$$

このとき

$$b^{(n-1)/2} \not\equiv \left(\frac{b}{n}\right) \bmod n$$

このような b は b' を $\bmod p_1$ での非剰余としたとき

$$b \equiv b' \bmod p_1,\ b \equiv 1 \bmod n/p_1$$

の解として中国剰余定理 (**Theorem1.3.2**) を使い構成することができる。

(証明終)

3.4.1 アルゴリズム

n を素数判定したい整数とする。そのとき Solovay–Strassen 法は次のようになる。

(1) $0 < b < n$ であるような b をランダムに選んでくる。

(2) $b^{(p-1)/2} \equiv \left(\frac{b}{p}\right) \bmod p$ であるかどうかテストする。

(3) もし成り立てば (1) にいく。
もし成り立たなければ、n は合成数であると判定してテストを終わる。

もし n がランダムに選んだ k 個の b についてテストを通過すれば n が合成数であるにもかかわらず誤って素数であると判定される確率は $1/2^k$ 以下である。

3.5 Miller–Rabin 法

Miller–Rabin は Solovay–Strassen 法よりも高い確率で素数判定をする方法を提案した。それは強擬似素数の考えに基づいている。

- **強擬似素数**

n を奇数合成数として $n = 2^s t$ (ただし t は奇数) とする。$(b, n) = 1$ である b について

$$\begin{array}{c} \text{(i)} \quad b^t \equiv 1 \bmod n \\ \text{または} \quad \text{(ii)} \quad 0 \leq {}^{\exists} r < s\ ;\ b^{2^r t} \equiv -1 \bmod n \end{array} \tag{3.5}$$

このとき n は b を底とした強擬似素数と呼ばれる。
もし

$$(b^t)^{2^r} \not\equiv -1 \bmod n \quad \text{and} \quad (b^t)^{2^{r+1}} \equiv 1 \bmod n$$

となったら n は合成数とわかる。なぜなら ± 1 のみが素数を法としたときの 1 の平方根となるからである。

今から、n が奇数合成数であるとき $0 < b < n$ であるせいぜい 25%の b について n は強擬似素数になることを証明する。

Proposition 3.5.1 nがbを底とするEulerの擬似素数ならば、nはbを底とする擬似素数

Proposition 3.5.2 nがbを底とする強擬似素数ならば、nはbを底とするEulerの擬似素数

Proposition 3.5.3 nが奇数合成数ならば、$0 < b < n$であるせいぜい25 %のbについてnは強擬似素数

Proposition 3.5.1, 3.5.2 の証明については参考文献 [7] 参照のこと。**Proposition3.5.3** を示す前に次の **Lemma** を示しておく。

Lemma 3.5.1 $d = (k, m)$とする。このとき巡回群$\{g, g^2, \cdots, g^m = 1\}$の中で$x^k = 1$を満たす元が$d$個存在する。

証明

$$\begin{aligned} g^j\text{が } x^k = 1 \text{ を満たす} &\iff g^{jk} = 1 \\ &\iff m|jk \\ &\iff \tfrac{m}{d}|j\tfrac{k}{d} \\ &\iff \tfrac{m}{d}|j \end{aligned}$$

よってそのようなjは$\frac{m}{d}, \frac{2m}{d}, \cdots, \frac{dm}{d}$の$d$個である。 **(証明終)**

Lemma 3.5.2 pを奇素数とし、$p - 1 = 2^{s'}t'$ (t': 奇数) と書く。
すると$x^{2^r t} \equiv -1 \bmod p$ (ただしt:奇数) を満たす$x \in (Z/p\boldsymbol{Z})^*$の数は

$$\begin{cases} 0 & r \geq s' \text{のとき} \\ 2^r \gcd(t, t') & r < s' \text{のとき} \end{cases}$$

証明 gを$(\boldsymbol{Z}/p\boldsymbol{Z})^*$の生成元、$x = g^j \quad (0 \leq j < p - 1)$とする。すると$g^{(p-1)/2} \equiv -1 \bmod p$は$p - 1 = 2^{s'}t'$であることから$g^{2^{s'-1}t'} \equiv -1 \bmod p$とかくことができる。また、

$$x^{2^r t} \equiv g^{j2^r t} \equiv -1 \bmod n$$

より **Lemma 3.5.2** の合同式は

$$2^r tj \equiv 2^{s'-1}t' \bmod 2^{s'}t' \quad (j \text{ は未知数})$$

と同値となる。もし$r > s' - 1$であれば解なし、$r \leq s' - 1$であれば$\gcd(2^{s'}t', 2^r t) = 2^r d$ (ただし$d = \gcd(t, t')$) で割って

$$\frac{t}{d}j \equiv 2^{s'-1-r}\frac{t'}{d} \bmod 2^{s'-r}\frac{t'}{d}$$

この合同式は$\bmod 2^{s'-r}\frac{t'}{d}$で唯一つの解を持ち (**Theorem 1.3.1**)、$\bmod 2^{s'}t'$では$2^r d$個の解を持つ。

(証明終)

Proposition3.5.3 の証明

Case(i) $p^\alpha || n, \alpha \geq 2$ とする。このとき

$$\#\{b\ ; 0 < b < n, b^{n-1} \equiv 1 \bmod n\} \leq \frac{n-1}{4}$$

を示す。つまり $0 < b < n$ である b のせいぜい 25% しか擬似素数にならない。

$b^{n-1} \equiv 1 \bmod n$ とする。すると $b^{n-1} \equiv 1 \bmod p^2$ である。よって、b が $\bmod p^2$ でこのようになる条件を見つける。

$(\boldsymbol{Z}/p^2\boldsymbol{Z})^*$ は位数 $p(p-1)$ の巡回群だから

$$^\exists g\ ; (\boldsymbol{Z}/p^2\boldsymbol{Z})^* = \{g, g^2, \cdots, g^{p(p-1)}\}$$

Lemma 3.5.1 より

$$\#\{b\ ; 0 < b < n, b^{n-1} \equiv 1 \bmod p^2\} = d \quad (\text{ ただし } d = \gcd(p(p-1), n-1)\ )$$

しかし $p|n$ だから $p \not| n-1$ となり $p \not| d$。ゆえに d は最大でも $p-1$ でしかない。

よって $b^{n-1} \equiv 1 \bmod p^2$ となるような 0 から n までの範囲にある p^2 で割りきれない b の割合は

$$\frac{p-1}{p^2-1} = \frac{1}{p+1} \leq \frac{1}{4}$$

以下である。

$b^{n-1} \equiv 1 \bmod n$ となるような 0 から n までの範囲にある b の割合はこれ以下であるので $0 < b < n$ である b のせいぜい 1/4 について n は擬似素数であるとわかる。**Proposition 3.5.1,3.5.2** より、$0 < b < n$ となる b のせいぜい 25% が強擬似素数であることがわかる。

Case(ii) $n = pq$ (p, q : 異なる素数) とし、

$$p - 1 = 2^{s'}t',\ q - 1 = 2^{s''}t'' \quad (t', t'' : \text{奇数},\ s' \leq s'')$$

とする。このとき n が強擬似素数となる $b \in (\boldsymbol{Z}/n\boldsymbol{Z})^*$ について

$$\begin{array}{rl} & (1) \quad b^t \equiv 1 \bmod p \text{ かつ } b^t \equiv 1 \bmod q \\ \text{または} & (2) \quad 0 \leq {}^\exists r < s\ ;\ b^{2^r t} \equiv -1 \bmod p \text{ かつ } b^{2^r t} \equiv -1 \bmod q \end{array}$$

Lemma 3.5.1 より

$$\#\{b\ ; b^t \equiv 1 \bmod p \text{ かつ } b^t \equiv 1 \bmod q\} = \gcd(t, t')\gcd(t, t'') \leq t't''$$

Lemma 3.5.2 より各 $r < \min(s', s'') = s'$ に対して

$$\begin{aligned} \#\{b\ ; b^{2^r t} \equiv -1 \bmod n\} &= \#\{b\ ; b^{2^r t} \equiv -1 \bmod p \text{ かつ } b^{2^r t} \equiv -1 \bmod q\} \\ &= 2^r \gcd(t, t')2^r \gcd(t, t'') < 4^r t't'' \ . \end{aligned}$$

$n-1>\varphi(n)=2^{s'+s''}t't''$ であるから、n が強擬似素数となる b $(0<b<n)$ の割合はせいぜい

$$\frac{t't''+t't''+4t't''+\cdots+4^{s'-1}t't''}{2^{s'+s''}t't''}=2^{-(s'+s'')}\left(1+\frac{4^{s'}-1}{4-1}\right)\ . \tag{3.6}$$

もし $s''>s'$ ならばこれはせいぜい

$$2^{-2s'-1}\left(\frac{2}{3}+\frac{4^{s'}}{3}\right)\leq 2^{-3}\frac{2}{3}+\frac{1}{6}=\frac{1}{4}$$

である。

一方、もし $s'=s''$ ならば $\gcd(t,t')<t'$ or $\gcd(t,t'')<t''$ となることを示す。なぜなら $t'|t,t''|t$ であるなら

$$n-1=2^st=pq-1\equiv q-1 \bmod t'$$

よって、$t'|(q-1)=2^{s''}t''$ つまり $t'|t''$。同様にして $t''|t'$。 よって $t'=t''$ つまり $p=q$ となり矛盾

ゆえに $\gcd(t,t')<t'$ とすると t,t' : 奇数 としているので $\gcd(t,t')\leq\frac{1}{3}t'$ がいえる。ゆえにこのケースでは (3.6) の評価において $t't''$ を $\frac{1}{3}t't''$ で置き直すことができる。よってその評価は

$$\frac{1}{3}2^{-2s'}\left(\frac{2}{3}+\frac{4^{s'}}{3}\right)\leq\frac{1}{18}+\frac{1}{9}=\frac{1}{6}<\frac{1}{4}$$

Case(iii) n を異なる 3 つ以上の素数の積とする。つまり

$$n=p_1p_2\cdots p_k\ (k\geq 3),\quad p_j-1=2^{s_j}t_j\quad (t_j:\text{奇数})$$

とする。このとき一般性を失うことなく $s_1=\min(s_1,s_2,\cdots,s_k)$ とできる。よってその評価は $k\geq 3$ であることより

$$\begin{aligned}2^{-s_1-s_2\cdots-s_k}\left(1+\frac{2^{ks_1}-1}{2^k-1}\right)&\leq 2^{-ks_1}\left(\frac{2^k-2}{2^k-1}+\frac{2^{ks_1}}{2^k-1}\right)\\&=2^{-ks_1}\frac{2^k-2}{2^k-1}+\frac{1}{2^k-1}\leq 2^{-k}\frac{2^k-2}{2^k-1}+\frac{1}{2^k-1}=2^{1-k}\leq\frac{1}{4}\end{aligned}$$

となる。 **(証明終)**

3.5.1 アルゴリズム

n を素数判定したい整数とする。そのとき Miller–Rabin 法は次のようになる。

(1) $n-1=2^st$ (t:奇数) とする

(2) $0<b<n$ であるような b をランダムに選んでくる。

(3) $b^t\equiv 1 \bmod n$ であるかどうかテストする。

(4) もし成り立てば (2) にいく。
もし成り立たなければ、$r = 0, 1, 2, \cdots$ として

$$(b^t)^{2^r} \equiv -1 \bmod n$$

となる r $(0 \leq r < s)$ が見つかるかテストする。もし見つかれば (2) にいく。

(5) (4) のような r が見つからず

$$(b^t)^{2^r} \not\equiv -1 \bmod n \quad \text{and} \quad (b^t)^{2^{r+1}} \equiv 1 \bmod n$$

となったら n は合成数であると判定してテストを終わる。

もし n がランダムに選んだ k 個の b についてテストを通過すれば n が合成数であるにもかかわらず誤って素数であると判定される確率は $1/4^k$ 以下である。

3.6 Adleman–Rumely の素数判定法

Adleman–Rumely は 1980 年に、決定論的素数判定法を発表した。この方法を用いると、100 桁以内の整数も 1 分以内に素数か否か判定される。200 桁の数でも 10 分以内に判定される (参考文献 [15])。この方法を説明する前に数学的な準備をする。

3.6.1 ガウス和

q を奇素数とし、g を $\bmod q$ での原始根とする。このとき $(a, q) = 1$ であるならば

$$a \equiv g^x \bmod q$$

となる x が $\bmod(q-1)$ で唯一決まる。この x を a の指数といい、 $x = \operatorname{ind}_q a = \operatorname{ind} a$ と表す。
次に p を $q-1$ の素因子とし、

$$\zeta_p = e^{2\pi i/p}$$

とする。ζ_p を 1 の原始 p 乗根という。$p|q-1$ なので $\zeta_p^{q-1} = 1$ である。よって、

$$\chi(a) = \chi_{p,q}(a) = \zeta_p^x = \zeta_p^{\operatorname{ind} a}$$

は定まる。$a \equiv a' \bmod q$ ならば $a \equiv a' \equiv g^x \bmod q$ だから $\chi(a) = \chi(a')$ である。次に $(b, q) = 1$ で $y = \operatorname{ind} b$ ならば

$$ab \equiv g^x g^y = g^{x+y} \bmod q$$

よって

$$\chi(ab) = \zeta_p^{x+y} = \zeta_p^x \zeta_p^y = \chi(a)\chi(b)$$

となる。また

$$\chi^n(a) = \Big(\chi(a)\Big)^n = \zeta_p^{n \operatorname{ind} a} \ , \ \overline{\chi}(a) = \overline{\chi(a)} = \zeta_p^{-\operatorname{ind} a}$$

とする。さらに ζ_q を 1 の原始 q 乗根とする。つまり $\zeta_q = e^{2\pi i/q}$ である。このとき

$$\tau(\chi) = \tau(\chi_{p,q}) = \sum_{a=1}^{q-1} \chi(a)\zeta_q^a = \sum_{a=1}^{q-1} \zeta_p^{\mathrm{ind}\, a}\zeta_q^a$$

をガウスの和 (Gauss sum) という。このとき次の公式が成り立つ。

$$\tau(\chi)\tau(\overline{\chi}) = \chi(-1)q$$

3.6.2 Adleman-Rumely 法の説明

具体的に説明するため、$t = 2\cdot 3\cdot 5\cdot 7\cdot 11\cdot 13\cdot 17 \simeq 500000$ とする。$q-1$ が t の約数となる素数 q はたくさんある。つまり、q は

$$q - 1 = 2^{e_1}3^{e_2}5^{e_3}7^{e_4}11^{e_5}13^{e_6}17^{e_7}\ ,\ e_i = 0 \text{ または } 1$$

となる素数である。q の候補としては $q = 11, 43, 23, \cdots$ などが上げられる。このような q の積を s とする。つまり

$$s = \prod_{q-1|t, q:\text{素数}} q \simeq 2.55 \times 10^{83}$$

となる。$s < n < s^2 \simeq 6.50 \times 10^{166}$ とする。

実際の値は

$$\begin{cases} t &= 510510 \\ s &= 255326063724039312904523743080416423277471162753367- \\ & \quad 341938296020524531511934351748454 \\ s^2 &= 651913988168121839665072110527472775853386745265482- \\ & \quad 357490239441279542798626307377271930779298183008865- \\ & \quad 064977443106977700829139792844823489956968771502578- \\ & \quad 27046891390116 \end{cases}$$

n を素因数分解したい整数とする。まず、n が s, t と共通約数を持つどうかは、すぐにわかる。よって

$$(n, st) = 1 \tag{3.7}$$

とする。

$p|q-1\,,\ q-1|t$ となる素数 p, q の組に対し、$\tau(\chi)$ を考える。n が素数であるならば、$n|\binom{n}{i}$ $(1 \le i \le n-1)$ である。よって $\tau(\chi)$ を n 乗すると

$$\tau(\chi)^n = (\sum \chi(a)\zeta_q^a)^n \equiv \sum \chi^n(a)\zeta_q^{na} \bmod n$$

となる。$\chi^n(n)\chi^n(a) = \chi^n(na)$ となるので

$$\tau(\chi)^n \equiv \chi^{-n}(n)\sum \chi^n(na)\zeta_q^{na} \bmod n$$

となる。a が 1 より $q-1$ まで動けば na も $\bmod q$ で考えれば 1 より $q-1$ まで動く。$a \equiv a' \bmod q$ ならば $\chi(a) = \chi(a')$, $\zeta_q^a = \zeta_q^{a'}$ だから

$$\tau(\chi)^n \equiv \chi^{-n}(n)\sum \chi^n(a)\zeta_q^a = \chi^{-n}(n)\tau(\chi^n) \bmod n$$

となる。もし、$p|q-1$, $q-1|t$ である p, q のいずれか 1 つでも上の合同式が成り立たなければ、n は合成数とわかる。

よって、

$$\tau(\chi)^n \equiv \chi^{-n}(n)\tau(\chi^n) \bmod n \tag{3.8}$$

が成り立っているとする。すると (3.8) より

$$\tau(\chi)^{n^i} \equiv \chi^{-in^i}(n)\tau(\chi^{n^i}) \bmod n \tag{3.9}$$

が帰納法により証明される。

(3.9) において $i = p-1$ とすると、$n^{p-1} \equiv 1 \bmod p, \chi^p = 1$ を使い

$$\tau(\chi)^{n^{p-1}} \equiv \chi(n)\tau(\chi) \bmod n$$

が分かる。$\tau(\chi)\tau(\overline{\chi}) = \pm q$ は n と互いに素だから両辺に $\tau(\overline{\chi})$ を掛け $\pm q$ で割っても、合同式は成り立つ。よって

$$\tau(\chi)^{n^{p-1}-1} \equiv \chi(n) \bmod n \tag{3.10}$$

が得られる。(3.10) は (3.7) と (3.8) が成り立てば n が合成数であっても成立する。n の素因子を r とする。n を r で置き換えても r は (3.7) を満たし、素数であるから (3.8) も満たす。よって

$$\tau(\chi)^{r^{p-1}-1} \equiv \chi(r) \bmod r \tag{3.11}$$

が成り立つ。r は n の約数であるから、(3.10) より

$$\tau(\chi)^{n^{p-1}-1} \equiv \chi(n) \bmod r \tag{3.12}$$

が得られる。$\chi(n)$ は 1 の p 乗根だから、(3.12) の両辺を p 乗すれば $\tau(\chi)$ の $\bmod r$ での位数は $p(n^{p-1}-1)$ の約数であることがわかる。$n^{p-1}-1 = p^h u$, $p \not| u$ とおいたとき、もし、$\chi(n) \neq 1$ ならば $\tau(\chi)$ の $\bmod r$ での位数は $p^{h+1}u$ の約数であるが $p^h u$ の約数ではない。$p|q-1$ なる p と q の組に対して χ は決まるが $p|t$ に対して

$$p|q-1\ ,\ q-1|t\ ,\ \chi(n) \neq 1 \tag{3.13}$$

となる q が 1 つはあると仮定する。するとそのような p, q の組に対して、$\tau(\chi)$ の mod$r$ での位数は p できっかり $h+1$ 回割れる。(3.11) より位数は $p(r^{p-1}-1)$ の約数だから、$p^{h+1}|p(r^{p-1}-1)$ つまり $p^h|(r^{p-1}-1)$ となり、$r^{p-1}-1=p^h r'$ とおけば

$$\frac{r^{p-1}-1}{n^{p-1}-1}=\frac{r'}{u},\quad p\not| u$$

となる。$ux\equiv 1 \bmod p$ なる x は存在するので、$r'x=a,\ ux=b$ とおけば

$$\frac{r^{p-1}-1}{n^{p-1}-1}=\frac{a}{b},\quad b\equiv 1 \bmod p \tag{3.14}$$

となる。(3.13) は (3.14) が成り立つために、つまり $r^{p-1}-1$ が p^h で割り切れる保証のため使われている。よって、(3.14) が成り立たなくても

$$n^{p-1}\not\equiv 1 \bmod p^2 \tag{3.15}$$

が成り立てば、(3.14) は成立する。なぜなら Fermat の定理より $r^{p-1}\equiv 1 \bmod p$ となり、$h\leq 1$ より $p^h|r^{p-1}-1$ がいえるからである。

(注)　(3.13),(3.15) ともに成立しないときでも改良をすることによって判定ができるようになる。

(3.14) が成り立つ a は r と p に依存するので、$a=a(p,r)$ と表すことにする。$b\equiv 1 \bmod p$ より、(3.11) を使うと

$$\chi(r)=\chi(r)^b\equiv\tau(\chi)^{b\,(r^{p-1}-1)} \bmod r$$

となる。(3.14) と (3.12) を使えば、

$$\chi(r)=\chi(r)^{a\,(n^{p-1}-1)}\equiv\chi(n)^a \bmod r$$

となる。$\chi(r),\chi(n)^a$ はともに 1 の p 乗根であるので mod$r$ で等しいときは、真に等しい。つまり

$$\chi(r)=\chi(n)^a \tag{3.16}$$

である。$a=a(p,r)$ は p と r に依存するが

$$i\equiv a(p,r) \bmod p\quad (p=2,3,\cdots,17)$$

を満たす i は mod$t$ で唯一つ決まる。$\chi(n)^p=1$ であるから、p に依存しない i を用いて

$$\chi(r)=\chi(n)^i=\chi(n^i)$$

が得られる。よって、χ の定義より

$$\zeta_p^{\mathrm{ind}\,a}=\zeta_p^{\mathrm{ind}\,n^i}$$

となる。つまり

$$\operatorname{ind} r \equiv \operatorname{ind} n^i \bmod p \tag{3.17}$$

がいえた。$q|s$ なる素数 q を固定したとき、$p|q-1$ なるすべての p に対して (3.17) がいえるので

$$\operatorname{ind} r \equiv \operatorname{ind} n^i \bmod q-1 \tag{3.18}$$

となる。mod$q$ での原始根の意味を考えれば

$$r \equiv n^i \bmod q \tag{3.19}$$

が得られる。$q|s$ なるすべての素数 q で (3.19) がいえるのだから

$$r \equiv n^i \bmod s \tag{3.20}$$

となる。この式を次のように使う。

もし、n が素数でないとし、$r \leq \sqrt{n}$ なる n の素因数 r があったならば、$r^2 \leq n < s^2$ より $r < s$ となる。また i は mod$t$ で決まるので

$$r_i \equiv n^i \bmod s, \quad (0 < r_i < s,\ 0 < i < t) \tag{3.21}$$

とおけば、(3.20) より $r = r_i$ となる。つまり n が合成数ならばその素因子の 1 つは (3.21) により必ず見つけられる。よって、(3.21) により得られるすべての r_i が n の約数でなければ、n が素数であることが分かる。

3.6.3 アルゴリズム

改良をくわえて、n が素数かどうか判定するアルゴリズムは次のようになる。

(1) $(n, st) \neq 1$ であれば、n の約数が見つかったことになり、(5) にいく。

(2) $p|q-1$, $q-1|t$ となる素数 p, q の組み合わせに対して、ガウスの和 $\tau(\chi) = \tau(\chi_{p,q})$ を考える。

$$\tau(\chi)^n \equiv \chi^{-n}(n)\tau(\chi^n) \bmod n \tag{3.22}$$

が成立しなければ (5) にいく。

(3) 以下の条件 (i),(ii),(iii) によって ν と r_i を決める。

(i) 『$^{\exists}q\ ;\ \chi(n) \neq 1$』
が成り立てば n の素因数を r としたとき

$$\frac{r^{p-1}-1}{n^{p-1}-1} = \frac{a}{b}\ ,\ b \equiv 1 \bmod p \tag{3.23}$$

となる a , b が mod$p$ で定まる。

(ii)　　『$n^{p-1} \not\equiv 1 \bmod p^2$』
が成り立てば $r^{p-1} \equiv 1 \bmod p$ なので (3.23) のような a, b が $\bmod p$ で定まる。

(iii)　　『$n^{p-1} - 1 = p^h u\ ,\ p \not| u\ ,\ h > 1$ とおく。 $^{\exists} j\ ;\ \tau(\chi)^{p^i u} \equiv \zeta_p^j \bmod n$ となる最小の $i\,(> 0)$ を $\omega(\chi_{p,q})$ と書く。$^{\forall} q$ に対する $\omega(\chi)$ の最大値を $\omega = \omega_q$ とおく。$\omega = \omega(\chi)$ なる χ に対して

$$\omega > 1\ ,\ \tau(\chi)^{p^\omega u} \equiv 1 \bmod n$$

ならば、$^{\forall} i$ に対して $\tau(\chi)^{p^{\omega-1}u} - \zeta_p^i$ のどれかの 1 の根の係数は n と互いに素である。』
(iii) が成り立たないときは n の約数が見つかり、(5) にいく。
(iii) が成り立つときは r を n の素因数として

$$r^{p-1} \equiv 1 \bmod p^\omega$$

が成立する。ゆえに

$$\frac{r^{p-1} - 1}{p^\omega u} = \frac{a}{b}\ ,\ b \equiv 1 \bmod p \tag{3.24}$$

となる a, b が $\bmod p$ で決まる。

(i) または (ii) が成立するときは、$p^\omega u = n^{p-1} - 1$ と ω を定める。すると、(i)∼(iii) のどれかが成立するとき

$$\tau(\chi)^{p^\omega u} \equiv \eta(\chi) \bmod n\ ,\ \eta(\chi) = 1 \text{ の } p \text{ 乗根}$$

となる $\eta(\chi)$ が定まる。なお、(i) または (ii) が成立するときは $\eta(\chi) = \chi(n)$ である。

以上の議論より、n の素因子 r に対して、

$$\chi(r) = \eta(\chi)^a$$

が得られる。従って $^{\forall}\chi$ に対して、

$$\chi(\nu) = \eta(\chi)$$

となる自然数 ν が存在する。

(4)　　$\nu^i \equiv r_i \bmod s\ ,\ 0 < r_i < s\ ,\ 0 \le i < t$ なる $^{\forall} r_i$ に対して $r_i \not| n$ が成立すれば、n は素数と判定される。成立しなければ、n 約数 r_i が見つかり、(5) へいく。

(5)　　n は合成数であると判定される。

詳細については、参考文献 [8],[15] を参照のこと。

第4章　素因数分解法

4.1　Fermat の素因数分解法

n を素因数分解する整数とし、$n = pq$ (ただし $p > q$) とする。
もし、$p \simeq q$ のときは $p - q \simeq 0$ なので次のようなことが言える。
まず

$$t = \frac{p+q}{2}\ ,\ s = \frac{p-q}{2}$$

とすると

$$n = pq = t^2 - s^2 = (t-s)(t+s)$$

と分解され $s \simeq 0$ なので $t \simeq \sqrt{n}$ となる。

$$t^2 - n = s^2 \tag{4.1}$$

より $t = [\sqrt{n}] + 1, [\sqrt{n}] + 2, \cdots$ と 1 ずつ増やしていき n を引いたものが完全平方になっているかを調べればよい。もしそのような t, s がみつかれば t, s の決め方より

$$p = \frac{t+s}{2}\ ,\ q = \frac{t-s}{2}$$

となるので p, q が求められる。

また (4.1) を見つけるほかの方法として拡張 Fermat 法と言われるものがある。(4.1) が成り立つためにはある値 E について

$$s^2 \equiv t^2 - n \bmod E$$

が成り立たなければならない。よってまず E を適当に決め $n \equiv a \bmod E$ がわかれば $t = 0, 1, 2, \cdots, E-1$ に対して $t^2 - a \bmod E$ を計算する。また $s = 0, 1, 2, \cdots, E-1$ に対して $s^2 \bmod E$ も計算する。そうすれば $t^2 - a \equiv s^2 \bmod E$ となる s が存在する t についてのみ調べればよいことになり、調べればよい t の範囲が狭まる。

しかしこの方法は n の約数が $\sqrt{n}$ の近くにないと時間がかかる。つまり Fermat の素因数分解法は n が特殊な形の素因数分解を持つときのみに使える方法であって、汎用性はない。

4.2　Pollard の $p-1$ 法

Pollard の $p-1$ 法は Fermat の定理にもとづいている。(**Theorem1.3.3 参照**)

いま p を n の素因子で $p-1|E$ とする。このとき $(a,n)=1$ であるような a を選んでくると $(a,p)=1$ となり、 $p|a^E-1$ であるから

$$p|\,(a^E-1,n) \qquad (\text{ ただし } (a,n)=1\ )$$

よって (a^E-1,n) を計算することで n の約数を見つけられる。p は事前にはわかっていないので、アルゴリズムにおいて $p-1$ のすべての素数べき約数は M によっておさえられるとする。つまり $p-1=p_1^{e_1}p_2^{e_2}\cdots p_l^{e_l}$ と素因数分解したとき $p_i^{e_i}<M$ とする。$q_i<M$ となるすべての素数のついて $q_i^{e_i}<M\leq q_i^{e_{i+1}}$ となるような e_i を求め

$$E=\prod_{q_i^{e_i}<M\leq q_i^{e_{i+1}}} q_i^{e_i}$$

とする。すると $p-1|E$ とすることができる。

E は非常に大きくなるので計算することは困難になる。そこで $a^E \bmod n$ を計算するかわりに

$$a\leftarrow a^{q_i^{e_i}} \bmod n$$

として繰り返し計算すればよい。よって、もし $p-1$ が小さな素数の積として表せるのであれば、$p-1$ 法は非常に有力な方法である。

4.3 Lenstra の楕円曲線法

Pollard の $p-1$ 法は $F_p=GF(p)$ の乗法群における単位元を作り出す試みであるとみることができる。Lenstra による楕円曲線法も Pollard の $p-1$ 法とよく似ていて、乗法群のかわりに $\bmod n$ での楕円曲線を使う。 Lenstra による楕円曲線法は次のようになる。

もし、p に近い位数 g を持った群 C_f をランダムに選んでくれば F_p ではなく C_f において Pollard の $p-1$ 法と同じような方法を取ることができる。もし g のすべての素因子が M 以下であれば n の約数を見つけることができる。そうでなければ、異なった C_f を使って n の約数が見つかるまで繰り返す。

4.3.1 楕円曲線

K を体 , $\mathrm{char}\,K\neq 2,3$, $f(x)=x^3+ax+b\quad (a,b\in K)$ は重解をもたない 3 次方程式とする。このとき K 上の楕円曲線 C_f とは

$$y^2=f(x)=x^3+ax+b \tag{4.2}$$

を満たす $(x,y)\quad$ (ただし $x,y\in K$) の集合と無限遠点 ∞ との和集合である。

このとき C_f 上で次のようにして 演算を定義する。

Definition 4.3.1　$P_1 = (x_1, y_1)$, $P_2 = (x_2, y_2)$ を ∞ と異なる楕円曲線上の点とする。このとき $P_3 = (x_3, y_3) = P_1 + P_2$ を

$$(x_3, y_3) = \Big(\lambda^2 - x_1 - x_2 \ , \ \lambda(x_1 - x_3) - y_1\Big) \tag{4.3}$$

ただし、

$$\lambda = \begin{cases} (3x_1^2 + a)/(2y_1) & P_1 = P_2 \text{のとき} \\ (y_1 - y_2)/(x_1 - x_2) & P_1 \neq P_2 \text{のとき} \end{cases}$$

また λ の分母が 0 になるときは $P_1 + P_2 = \infty$ とし、さらに

$$P + \infty = \infty + P = P \, , \, \infty + \infty = \infty$$

と定義する。

このとき C_f は演算 $+$ についてアーベル群となる。このとき ∞ が単位元である。Lenstra のアルゴリズムにおいては、$K = F_p$ とする (ただし p は n の素因子)。そして、$p-1$ 法で使った F_p の代わりに (4.2),(4.3) で定義される群 C_f を使う。p は事前には解っていないから、すべての演算は環 $\bmod n$ で行う。

4.3.2　アルゴリズム

(1)　x_0, y_0 と $a \in [0, n)$ をランダムに選んでくる。すると $b = y_0 - (x_0^3 + ax_0) \bmod n$ とすることで楕円曲線 C_f が定義できる。そして

$$P \leftarrow P_0 = (x_0, y_0)$$

とする。

(2)　素数 $q = 2, \cdots, M$ に対して、Pollard の $p-1$ 法で求めたような E を計算する。そして、a, b によって定義される群 C_f において

$$EP \equiv \underbrace{P + P + \cdots + P}_{E \text{ 個}} \bmod n$$

を計算する。このときも $p-1$ 法のときのように

$$P \leftarrow q^e P \bmod n$$

として、繰り返し計算をする。(ただし e は Pollard の $p-1$ 法で選んできたようなもの)

もし、$P = \infty$ になれば (つまり $\bmod n$ で λ の分母を計算したとき 0 になれば) n の素因子が見つかる。

4.4　2 次合同式法

現在、汎用的でもっとも強力な素因数分解アルゴリズムは次のような簡単な合同式を利用している。

もし、

$$x^2 \equiv y^2 \bmod n$$

となる x, y を見つけることができれば

$$x^2 - y^2 = (x+y)(x-y) \equiv 0 \bmod n$$

となるので $(x \pm y, n)$ を計算することで n の真の約数を見つけることが出来るであろうという方法である。この方法を 2 次合同式法と呼ぶ。このような x, y を系統的に見つける方法として以下で述べる連分数法・2 次ふるい法・複数多項式 2 次ふるい法などが考え出されている。これらの方法を述べる前に上の合同式に関する次の **Proposition** を示す。

Proposition 4.4.1

$$x^2 \equiv y^2 \bmod n$$

を満たす整数 x, y $(0 < y < x < n)$ に対して、$(x-y, n)$ または $(x+y, n)$ を計算して n の約数が見つかる確率は $1/2$ 以上である。

証明　$(x, n) > 1$ のとき $p | x-n$ となる素数 p が存在する。このとき $y^2 - x^2 \equiv y^2 \bmod p$ だから $p | y$ である。よって、このとき $p \leq (x-y, n) < n$ となり $(x-y, n)$ を計算することで n の約数が見つかる。$(y, n) > 1$ のときも同様。

$(x, n) = (y, n) = 1$ のとき n の素因数分解を

$$n = p_1 \cdots p_l \ (i \neq j \text{ ならば } p_i \neq p_j)$$

とする。このとき各 p_j に対して、

$$x^2 - y^2 = (x+y)(x-y) \equiv 0 \bmod p_j$$

となる。もし $x - y \equiv 0 \bmod p_j$ かつ $x + y \equiv 0 \bmod p_j$ とすると $2x \equiv 0 \bmod p_j$ となり仮定に反するので、$1/2$ の確率で $x - y \equiv 0 \bmod p_j$ または $x + y \equiv 0 \bmod p_j$ のいずれか一方のみが成立する。

$$x - y \equiv 0 \bmod p_1 \cdots p_l \text{ または } x + y \equiv 0 \bmod p_1 \cdots p_l$$

以外は $(x-y, n)$ または $(x+y, n)$ を計算して n の約数が見つかるから、求める確率は

$$\frac{2^l - 2}{2^l} \geq \frac{1}{2}$$

以上である。　**(証明終)**

4.5 2次ふるい法

4.5.1 factor base

2次ふるい法を説明する前に、factor baseの定義をする。

Definition 4.5.1 factor baseとは異なった素数の集合 $B = \{p_1, p_2, \cdots, p_l\}$ ($p_1 = -1$ とすることが多い) のことである。もし $y^2 \bmod n$ が B に含まれる素数の積として書けるとき y は B-numberとよばれる。

では、2次ふるい法の説明をする。いま

$$y_i^2 \bmod n = \prod_{j=1}^{l} p_j^{\alpha_{ij}}$$

と書けたとき、各 $y_i^2 \bmod n$ $(i = 1, 2, \cdots, h)$ について $\{\alpha_{i1}, \alpha_{i2}, \cdots, \alpha_{il}\}$ を対応させる。合計するとすべてのコンポーネントが偶数となる $\{\alpha_{i1}, \alpha_{i2}, \cdots, \alpha_{il}\}$ $(i = 1, 2, \cdots, h)$ が見つかったとすると

$$\prod_{i=1}^{h} y_i^2 = \prod_{j=1}^{l} p_j^{\sum_i a_{ij}} \quad (\text{ただし各}\textstyle\sum_i a_{ij}\text{は偶数})$$

となる。ゆえに

$$y = \prod_{i=1}^{h} y_i \bmod n\ ,\ x = \prod_{j=1}^{l} p_j^{\gamma_j} \bmod n\ (\text{ただし}\gamma_j = \tfrac{1}{2}\textstyle\sum_i \alpha_{ij})$$

とすると $x^2 \equiv y^2 \bmod n$ となる。

たまたま $x \equiv \pm y \bmod n$ となるときもあるが、そのときは合計すると偶数となる y_i を取り直すか、集合 B の要素を取り直せばよい。

4.5.2 アルゴリズム

2次ふるい法では $x^2 \equiv y^2 \bmod n$ となる x, y を見つけるために次のような関数を使う。

$$f(x) = x^2 - n$$

このとき $x = [\sqrt{n}] \pm 1, [\sqrt{n}] \pm 2, \cdots$ と選んでくると $x^2 \simeq n$ となるので $f(x)$ の値は小さくなる。ゆえに小さな素数で素因数分解できる可能性が高い。また

$$f(x) = x^2 - n \equiv x^2 \bmod n$$

だからいくつかの $f(x)$ を計算して、その積が平方となるような組み合わせを見つければ、$x^2 \equiv y^2 \bmod n$ となる x, y を見つけることができる。

4.5.3 評価

まず、以下で使うことを述べておく。

Fact 1. (Stiring formula)

$$\log(n!) \approx n\log n - n$$

これは

$$\lim_{n\to\infty} \frac{\log(n!)}{n\log n - n} = 1$$

を意味している。

Fact 2. 与えられた正整数 n と正数 u について、

$$\sum_{j=1}^{n} \alpha_j \leq u$$

となる n 項からなる非負整数 α_j の総数は二項係数 $\binom{[u]+n}{n}$ である。ここで $[u]$ は Gauss 記号と呼ばれるもので、u を越えない最大の整数。

Fact 1,2 の証明については参考文献 [7] を参照のこと。

さて、上でのアルゴリズムでの重要なステップはランダムに取ってきた x を越えない数が y (ただし $y < x$) を越えない素数の積として書ける確率を評価することだ。

このため $u = \log x/\log y$ とする。つまり x が r-bit 、y が s-bit とすると u はおよそ r/s となる。

次に $\pi(y)$ を y 以下である素数の数とする。素数定理によって $\pi(y) \approx \frac{y}{\log y}$ である。また $u \ll \pi(y)$ とする。実際には次のようなサイズであると仮定している

$$\begin{array}{rcl} y & \simeq & 10^6 \quad (\text{ よって} \pi(y) \simeq 7\cdot 10^4, \log y \simeq 14\ ) \\ u & \simeq & 8 \\ x & \simeq & 10^{48} \end{array}$$

$\Psi(x,y)$ を y 以下の素数で素因数分解できる x 以下の整数の数とする。つまり

$$\prod_j p_j^{\alpha_j} \leq x \quad (\text{ただし } p_j : \text{素数} < y,\ \alpha_j : \text{非負整数})$$

として書けるような整数の数だ。

すると $\prod_j p_j^{\alpha_j} \leq x$ に対する $(\alpha_1, \alpha_2, \cdots, \alpha_{\pi(y)})$ と y 以下の素数で素因数分解できる x 以下の整数との間に $1:1$ の対応がつく。

ゆえに、$\Psi(x,y)$ は log をとってくることで次の不等式の整数解 α_j の数と等しい。

$$\sum_{j=1}^{\pi(y)} \alpha_i \log p_j \leq \log x \tag{4.4}$$

ここで $\log p_j$ の大部分は $\log y$ と極めて近いという事実を使う。そこで不等式 (4.4) の $\log p_j$ を $\log y$ で置きなおす。でてきた不等式の両辺を $\log y$ で割り、$\log x/\log y$ を u で置き直して、$\Psi(x,y)$ は不等式

$$\sum_{j=1}^{\pi(y)} \alpha_j \leq u$$

の解の個数に等しいといえる。

次に $\pi(y)$ を y で置き直す。$\pi(y)$ を y で置き直すことによって余分な項を取り込んでしまうが、その項はキャンセルすることがわかる。

よって $\Psi(x,y)$ は

$$\sum_{j=1}^{y} \alpha_j \leq u$$

の y 項からなる非負整数の解の個数と等しくなる。しかし **Fact 2.** によって $\Psi(x,y)$ はおよそ

$$\binom{[u]+y}{y}$$

となる。

では、いまから $\log\left(\frac{\Psi(x,y)}{x}\right)$ の評価をする。これは 1 から x までのランダムに選んできた整数が y 以下の素数の積として書ける確率の log である。

u の決め方より $\log x = u\log y$ だから $\Psi(x,y)$ と **Fact 1.** より

$$\begin{aligned}\log\left(\frac{\Psi(x,y)}{x}\right) &\approx \log\left(\frac{([u]+y)!}{[u]!y!}\right) - u\log y \\ &\approx ([u]+y)\log([u]+y) - ([u]+y) \\ &\qquad -([u]\log[u]-[u]) - (y\log y - y) - u\log y\end{aligned}$$

さらにもう少し近似をする。最初に、$[u]$ を u で置き換える。次に $u \ll y$ としているので $\log(u+y)$ を $\log y$ で置き直す。すると次のことがいえる。

$$\log\left(\frac{\Psi(x,y)}{x}\right) \approx -u\log u$$

つまり

$$\frac{\Psi(x,y)}{x} \approx u^{-u}$$

4.6 連分数法

4.6.1 連分数展開

実数 x の連分数展開とは次のようにして実行することができる。

$$a_0 = [x],\ x_0 = x - a_0,\ a_1 = [1/x_0],\ x_1 = 1/x_0 - a_1$$

とする。

次に $i > 1$ について

$$a_i = [1/x_{i-1}]\ ,\ x_i = 1/x_{i-1} - a_i$$

とする。

もし $1/x_{i-1}$ が整数となれば $x_i = 0$ として連分数展開を終了する。

このとき $a_0, a_1, \cdots, a_i$ の作り方より、各 i について

$$x = a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{a_3 + \ddots \cfrac{1}{a_i + x_i}}}}$$

しかし、これを簡単に

$$x = a_0 + \frac{1}{a_1+}\frac{1}{a_2+}\frac{1}{a_3+}\cdots\frac{1}{a_i + x_i}$$

と書くことにする。

また、このとき x の i 番目近似分数 b_i/c_i とは

$$\frac{b_i}{c_i} = a_0 + \frac{1}{a_1+}\frac{1}{a_2+}\frac{1}{a_3+}\cdots\frac{1}{a_{i-1}}\frac{1}{a_i}$$

のことである。

連分数展開は x が有理数のときにかぎり有限回で終わることが証明されている。

このとき次のことが成り立つ。

Proposition 4.6.1

(a) $\dfrac{b_0}{c_0} = \dfrac{a_0}{1}\ ;\ \dfrac{b_1}{c_1} = \dfrac{a_0a_1+1}{a_1}\ ;\ \dfrac{b_i}{c_i} = \dfrac{a_ib_{i-1}+b_{i-2}}{a_ic_{i-1}+c_{i-2}} \quad (i \geq 2)$

(b) $b_i = a_ib_{i-1} + b_{i-2}$ かつ $c_i = a_ic_{i-1} + c_{i-2}$ ならば $(b_i, c_i) = 1$

(c) $b_ic_{i-1} - b_{i-1}c_i = (-1)^{i-1} \quad (i \geq 1)$

(証明)

(a) 帰納法による。$i = 0, 1, 2$ のとき b_i, c_i の定義より成立。よって i 番目まで成り立つとすると、

$$\begin{aligned}
\frac{b_{i+1}}{c_{i+1}} &= a_0 + \frac{1}{a_1+}\frac{1}{a_2+}\frac{1}{a_3+}\cdots\frac{1}{a_i + \frac{1}{a_{i+1}}} \\
&= \frac{(a_i + \frac{1}{a_{i+1}})b_{i-1} + b_{i-2}}{(a_i + \frac{1}{a_{i+1}})c_{i-1} + c_{i-2}} \\
&= \frac{a_{i+1}(a_ib_{i-1} + b_{i-2}) + b_{i-1}}{a_{i+1}(a_ic_{i-1} + c_{i-2}) + c_{i-1}} \\
&= \frac{a_{i+1}b_i + b_{i-1}}{a_{i+1}c_i + c_{i-1}} = \frac{b_{i+1}}{c_{i+1}}
\end{aligned}$$

(c) 帰納法による。

$$\begin{aligned} b_{i+1}c_i - b_ic_{i+1} &= (a_{i+1}b_i + b_{i-1})c_i - b_i(a_{i+1}c_i + c_{i-1}) \\ &= b_{i-1}c_i - b_ic_{i-1} \\ &= -(-1)^{i-1} = (-1)^i \end{aligned}$$

(b) b_i, c_i の共通約数は (c) より $(-1)^{i-1} = \pm 1$ を割り切る。よって共通約数は 1。

(証明終)

Propotion 4.6.1 の (c) の両辺を c_ic_{i-1} で割ると

$$\frac{b_i}{c_i} - \frac{b_{i-1}}{c_{i-1}} = \frac{(-1)^{i-1}}{c_ic_{i-1}}$$

$\{c_i\}$ は増加数列になるので b_i/c_i は振動しながら収束する。このときの極限は x である。なぜなら

$$x = a_0 + \frac{1}{a_1+}\frac{1}{a_2+}\frac{1}{a_3+}\cdots\frac{1}{a_i+}\frac{1}{\frac{1}{x_i}}$$

だから **Proposition4.6.1** より

$$x = \frac{b_{i+1}}{c_{i+1}} = \frac{b_i/x_i + b_{i-1}}{c_i/x_i + c_{i-1}} = \frac{b_i + x_ib_{i-1}}{c_i + x_ic_{i-1}}$$

で

$$\begin{aligned} \frac{b_{i-1}}{c_{i-1}} &< x < \frac{b_i}{c_i} \quad (i \equiv 1 \bmod 2 \text{ のとき}) \\ \frac{b_i}{c_i} &< x < \frac{b_{i-1}}{c_{i-1}} \quad (i \equiv 0 \bmod 2 \text{ のとき}) \end{aligned} \tag{4.5}$$

となる。次に **Proposition4.6.2** を示す。

Proposition 4.6.2 $x > 1$ （ ただし x は実数 ）の連分数展開を b_i/c_i とする。このとき $\forall i$ に対して

$$|\, b_i^2 - x^2c_i^2 \,| < 2x$$

が成り立つ。

(証明) (4.5) と

$$\left|\frac{b_{i+1}}{c_{i+1}} - \frac{b_i}{c_i}\right| = \frac{1}{c_ic_{i+1}}$$

より

$$|\, b_i^2 - x^2c_i^2 \,| = c_i^2\left|x - \frac{b_i}{c_i}\right|\left|x + \frac{b_i}{c_i}\right| = c_i^2\frac{1}{c_ic_{i+1}}\left(x + (x + \frac{1}{c_ic_{i+1}})\right)$$

ゆえに

$$\begin{aligned} |\, b_i^2 - x^2c_i^2 \,| - 2x &< 2x\left(-1 + \frac{c_i}{c_{i+1}} + \frac{1}{2xc_{i+1}^2}\right) \\ &< 2x\left(-1 + \frac{c_i}{c_{i+1}} + \frac{1}{c_{i+1}}\right) \\ &< 2x\left(-1 + \frac{c_{i+1}}{c_{i+1}}\right) = 0 \end{aligned}$$

(証明終)

4.6.2 アルゴリズム

Proposition4.6.2 において $x = \sqrt{n}$ とすれば

$$b_i^2 - nc_i^2 \equiv b_i^2 \bmod n < 2\sqrt{n}$$

となる。つまり b_i^2 を計算してもあまり大きくならないことが事前に保証されているわけである。

ゆえに b_i^2 は小さな素数で素因数分解できる可能性が高い。つまり、B-number である可能性が高い。後は、factor base の考えを使えばよい。

4.7 複数多項式２次ふるい法

Silverman は 2 次ふるい法を改良した複数多項式 2 次ふるい法を発表した。Silverman は $f(x)$ として

$$f(x) = ax^2 + bx + c,\ (a = d^2,\ b^2 - 4ac = n) \tag{4.6}$$

を用いている。

(4.6) の両辺を $4a = (2d)^2$ 倍すると

$$\begin{aligned}(2d)^2 f(x) &= 4a^2x^2 + 4abx + 4ac \\ &= \Big((2ax)^2 + 2(2ax)b + b^2\Big) - (b^2 - 4ac) \\ &= (2ax + b)^2 - n\end{aligned} \tag{4.7}$$

となる。$(2d, n) = 1$ となる d であれば、

$$(2d)e \equiv 1 \bmod n$$

となる e を見つけてくることができる。ゆえに (4.7) の両辺に e^2 をかけて

$$f(x) = e^2(2ax + b)^2 - e^2 n \equiv \Big(e(2ax + b)\Big)^2 \bmod n$$

となる。よっていくつかの $f(x)$ を計算してその積が平方数となるような組み合わせを見つければよい。

ここで重要なのは異なった $f(x)$ を計算して得られた数を組み合わせてよいということである。

このとき $|x - (-b \pm \sqrt{n})/2a| < M$ とすれば

$$|f(x)| < aM^2 + M\sqrt{n}$$

となる。

4.7.1 具体的な計算

まず、n が奇数であるから、$b^2 - 4ac = n$ となるためには b が奇数でなければならない。このとき $b^2 \equiv 1 \bmod 4$ であるから、$n \equiv 1 \bmod 4$ でなければならない。よって、$n \equiv 3 \bmod 4$ のときは少し修正して

$$kn \equiv 1 \bmod 4 \tag{4.8}$$

となる小さな k を求める。この kn を、これから n の代わりに用いるわけである。まず、a としては

$$a \simeq \frac{\sqrt{kn}}{\sqrt{2}M} \tag{4.9}$$

とする。さらに $a = d^2$ となり

$$d \equiv 3 \bmod 4, \quad d : \text{素数} \tag{4.10}$$

$$\left(\frac{kn}{d}\right) = 1 \tag{4.11}$$

となる d をさがす。つまり $\sqrt[4]{kn}/(\sqrt[4]{2}\sqrt{M})$ の近くで (4.10),(4.11) を満たす d を求め、$a = d^2$ とおけばよい。d は、おそらく素数であろうと思われる数を選べばよく、(4.11) と関係して Euler 規準

$$(kn)^{(d-1)/2} \equiv 1 \bmod d \tag{4.12}$$

が成り立ちさえすればよい。まず、$d \equiv 3 \bmod 4$ を用いて

$$h_0 \equiv (kn)^{(d-3)/4} \bmod d \tag{4.13}$$

なる h_0 を計算する。この h_0 を用いて

$$h_1 \equiv knh_0 \bmod d \tag{4.14}$$

なる h_1 を計算する。このとき

$$h_1^2 \equiv kn \cdot kn \cdot h_0^2 \equiv kn \cdot (kn)^{(d-1)/2} \bmod d$$

となるが、(4.12) を用いれば

$$h_1^2 \equiv kn \bmod d \tag{4.15}$$

となる。$kn - h_1^2$ が d で割れるので、$(d, h_1^2) = (d, kn) = 1$ となる。

$$2h_1 x \equiv (kn - h_1^2)/d \bmod d \tag{4.16}$$

なる解を h_2 とし、

$$b \equiv h_1 + h_2 d \bmod a \tag{4.17}$$

と b を定めると、

$$b^2 \equiv h_1^2 + 2h_1h_2d + d^2 \bmod a$$

となり $d^2 = a$, (4.16) より

$$b^2 \equiv h_1^2 + (kn - h_1^2) = kn \bmod a$$

となる。

d は (4.10) より奇数なので、a も奇数である。よって、もし、b が偶数ならば、b の代わりに $b-a$ を用いればよいから、$b=$ 奇数 に選べる。このとき (4.8) を用いれば、b^2-kn は 4 で割れる。つまり、

$$b^2 - kn = 4ac \tag{4.18}$$

なるを c 定めることができる。前節の計算通りにすれば、

$$f(x) = ax^2 + bx + c \tag{4.19}$$

とおくとき

$$(2d)^2 f(x) = (2ax+b)^2 - kn \tag{4.20}$$

となる。いくつかの $f(x)$ の積が平方数になるようにさせるためには、$f(x)$ が小さな素数で多く割れることが望ましい。$f(x)$ が 2 で割れるためには、(4.20) より $kn \equiv 1 \bmod 8$ が必要である。$f(x)$ を割る奇素数を p とおくと、(4.20) より

$$kn \equiv (2ax+b)^2 \bmod p \tag{4.21}$$

となる。よって、このような p に対して kn は平方剰余である。よって、kn が平方剰余となる小さな素数がたくさんあることが望ましい。このように k を選び、kn が平方剰余となる奇素数 $p_1, \cdots, p_r$ をあらかじめ求めておく。これ以後は 2 次ふるい法のところで説明したようにふるいにかけて $2, p_1, \cdots, p_r$ だけで割れる $f(x)$ をたくさん求める。あとは掃き出し法を用いるわけである。

æ

4.7.2 計算例

複数多項式 2 次ふるい法で 43429 を素因数分解してみる。

まず、M を決めて、a, b, c, d の値を定める。

M の値を 2 から 99 まで動かして、a, b, c, d の候補を調べてみると 3 種類しかないことが確かめられる。そこで、その 3 種類について x に値を代入して $f(x)$ の値を計算した。さらに、factor base を $\{-1, 3, 5, 7, 11, 13\}$ と取ってきたとき、素因数分解できたものについてはその素因数分解を実行した。

表 1　　$M = 13,\ a = 49,\ b = -41,\ c = -213,\ d = 7$ のとき

x の値	$f(x)$ の値	$f(x)$ の素因数分解 in factor base	mod43429 での $f(x)$ の平方根
-12	7335	$\times$	3189
-11	6167	$\times$	3182
-10	5097	$\times$	3175
-9	4125	$3 \cdot 5^3 \cdot 11$	3168
-8	3251	$\times$	3161
-7	2475	$3^2 \cdot 5^2 \cdot 11$	3154
-6	1797	$\times$	3147
-5	1217	$\times$	3140
-4	735	$3 \cdot 5 \cdot 7^2$	3133
-3	351	$3^3 \cdot 13$	3126
-2	65	$5 \cdot 13$	3119
-1	-123	$\times$	3112
0	-213	$\times$	3105
1	-205	$\times$	3098
2	-99	$(-1) \cdot 3^2 \cdot 11$	3091
3	105	$3 \cdot 5 \cdot 7$	3084
4	407	$\times$	3077
5	807	$\times$	3070
6	1305	$\times$	3063
7	1901	$\times$	3056
8	2595	$\times$	3049
9	3387	$\times$	3042
10	4277	$\times$	3035
11	5265	$3^4 \cdot 5 \cdot 13$	3028
12	6351	$\times$	3021

ここで $4125, 2475, 735$ を取ってくる。すると次のような表ができる。

$f(x)$ の値	factor base の巾					
	-1	3	5	7	11	13
$3168^2 = 4125$		1	3		1	
$3154^2 = 2475$		2	2		1	
$3133^2 = 735$		1	1	2		

するとベクトルの和が偶数となるようにできる。よって、

$$\begin{aligned}(3168 \cdot 3154 \cdot 3133)^2 &= 43196^2 \bmod 43429\\ (3^2 \cdot 5^3 \cdot 7 \cdot 11)^2 &= 43196^2 \bmod 43429\end{aligned}$$

このときは自明な合同式でしかない。

そこで、65,5265 をくわえる。すると次のようになる。

$f(x)$ の値	factor base の巾					
	-1	3	5	7	11	13
$3168^2 = 4125$		1	3		1	
$3154^2 = 2475$		2	2		1	
$3133^2 = 735$		1	1	2		
$3119^2 = 65$			1			1
$3028^2 = 5265$		4	1			1

するとベクトルの和が偶数となるようにできる。よって、

$$\begin{aligned}(3168 \cdot 3154 \cdot 3133 \cdot 3119 \cdot 3028)^2 &= 18074^2 \bmod 43429\\ (3^4 \cdot 5^4 \cdot 7 \cdot 11 \cdot 13)^2 &= 37411^2 \bmod 43429\end{aligned}$$

このとき

$$(37411 - 18074, 43429) = 317\,, \qquad (37411 + 18074, 43429) = 137$$

となり、約数がみつかった。さらに、多項式の係数を取り換えてみる。

表 2　　$M = 2,\ a = 121,\ b = -65,\ c = -81,\ d = 11$ のとき

x の値	$f(x)$ の値	$f(x)$ の素因数分解 in factor base	mod43429 での $f(x)$ の平方根
-12	18123	$\times$	2109
-11	15275	$\times$	2098
-10	12669	$\times$	2087
-9	10305	$\times$	2076
-8	8183	$\times$	2065
-7	6303	$\times$	2054
-6	4665	$\times$	2043
-5	3269	$\times$	2032
-4	2115	$\times$	2021
-3	1203	$\times$	2010
-2	533	$\times$	1999
-1	105	$3 \cdot 5 \cdot 7$	1988
0	-81	$(-1) \cdot 3^4$	1977
1	-25	$(-1) \cdot 5^2$	1966
2	273	$3 \cdot 7 \cdot 13$	1955
3	813	$\times$	1944
4	1595	$\times$	1933
5	2619	$\times$	1922
5	2619	$\times$	1922
6	3885	$\times$	1911
7	5393	$\times$	1900
8	7143	$\times$	1889
9	9135	$\times$	1878
10	11369	$\times$	1867
11	13845	$\times$	1856
12	16563	$\times$	1845

表 3　　$M = 17,\ a = 9,\ b = 7,\ c = -1205,\ d = 3$ のとき

x の値	$f(x)$ の値	$f(x)$ の素因数分解 in factor base	mod43429 での平方根
-12	7	7	7273
-11	-193	$\times$	7270
-10	-375	$(-1)\cdot 3\cdot 5^3$	7267
-9	-539	$(-1)\cdot 7^2\cdot 11$	7264
-8	-685	$\times$	7261
-7	-813	$\times$	7258
-6	-923	$\times$	7255
-5	-1015	$\times$	7252
-4	-1089	$(-1)\cdot 3^2\cdot 11^2$	7249
-3	-1145	$\times$	7246
-2	-1183	$(-1)\cdot 7\cdot 13^2$	7243
-1	-1203	$\times$	7240
0	-1205	$\times$	7237
1	-1189	$\times$	7234
2	-1155	$(-1)\cdot 3\cdot 5\cdot 7\cdot 11$	7231
3	-1103	$\times$	7228
4	-1033	$\times$	7225
5	-945	$(-1)\cdot 3^3\cdot 5\cdot 7$	7222
6	-839	$\times$	7219
7	-715	$(-1)\cdot 5\cdot 11\cdot 13$	7216
8	-573	$\times$	7213
9	-413	$\times$	7210
10	-235	$\times$	7207
11	-39	$(-1)\cdot 3\cdot 13$	7204
12	175	$5^2\cdot 7$	7201

では、3 つの表から 3 桁の数のみでふるい法を実行してみよう。
$65, 105, -81, -25, 273$ を取ってくる。すると次のような表ができる。

$f(x)$ の値	factor base の巾					
	-1	3	5	7	11	13
$3119^2 = 65$			1			1
$3084^2 = 105$		1	1	1		
$1977^2 = -81$	1	4				
$1966^2 = -25$	1		2			
$1955^2 = 273$		1		1		1

するとベクトルの和が偶数となるようにできる。よって、

$$\begin{aligned}(3119 \cdot 3084 \cdot 1977 \cdot 1966 \cdot 1955)^2 &= 17996^2 \bmod 43429 \\ \Big((-1) \cdot 3^3 \cdot 5^2 \cdot 7 \cdot 13\Big)^2 &= 25433^2 \bmod 43429\end{aligned}$$

このとき $(25433 \pm 17996, 43429)$ を計算しても自明な約数しか求められない。
$65, -99, 105, 273, -539$ を取ってくる。すると次のような表ができる。

$f(x)$ の値	factor base の巾					
	-1	3	5	7	11	13
$3119^2 = 65$			1			1
$3091^2 = -99$	1	2			1	
$3084^2 = 105$		1	1	1		
$1955^2 = 273$		1		1		1
$7264^2 = -539$	1			2	1	

するとベクトルの和が偶数となるようにできる。よって、

$$\begin{aligned}(3119 \cdot 3091 \cdot 3084 \cdot 1955 \cdot 7264)^2 &= 8142^2 \bmod 43429 \\ \Big((-1) \cdot 3^2 \cdot 5 \cdot 7^2 \cdot 11 \cdot 13\Big)^2 &= 32117^2 \bmod 43429\end{aligned}$$

このとき

$$(32117 - 8142, 43429) = 137\,, \qquad (32117 + 8142, 43429) = 317$$

となり、約数がみつかった。

第5章　新しいアルゴリズム

5.1　3次多項式2次ふるい法

今から、3次多項式を用いる新しい素因数分解アルゴリズムを述べる。

n を素因数分解したい数とする。このときまず小さな整数 k を決める。k は小さな素数の積となるようにする。次に、$a = [\sqrt[3]{kn}] + 1, s = [(a^3 - kn)/a], f = [\sqrt{s}]$ とする。

このとき $e_1 + e_2 + e_3 = 0, |e_i| \leq M$ ならば

$$\left|(a+e_1)(a+f+e_2)(a-f+e_3) - kn\right| < 2(1+M)\sqrt{kn} + 4M^2\sqrt[3]{kn}$$

であることがわかる。

$$(a+e_1)(a+f+e_2)(a-f+e_3) \equiv (a+e_1)(a+f+e_2)(a-f+e_3) - kn \bmod n \tag{5.1}$$

であるが、右辺は比較的小さくなる。よって素因数分解できる可能性も高い。また、$(a+e_1), (a+f+e_2), (a-f+e_3)$ も小さいので素因数分解できるであろう。そこで、(5.1) の右辺と、左辺の積のうち2つないしは1つが小さい素数の集合 B の範囲で素因数分解できるものを多く集める。すると、(5.1) の右辺と、左辺の積のうち2つが判っているものについては残りの1つの積の形式的な素因数分解が可能になる。

素因数分解を繰り返していくと、左辺の積のうち1つしか素因数分解できていないものも素因数分解できるようになるであろう。

後は、形式的に素因数分解できた数と3次多項式を組み合わせて、指数が偶数になるようなものを見つけてくればよい。

このアルゴリズムの特徴は

(1)　3乗根をとることで取り扱う数が小さくできる。

(2)　指数に負の数を認めることで、小さい数での (形式的な) 素因数分解が可能である。

ということである。

5.2　計算例

$k = 24, n = 43429, M = 10, B = \{-1, 2, 3, 5, 7, 11\}$ とする。(以下合同式はすべて $\bmod 43429$ であるとする)

まず、$(a+e_1),(a+f+e_2),(a-f+e_3),(a+e_1)(a+f+e_2)(a-f+e_3)-kn$ をそれぞれ計算する。自明な素因数分解を上げると

$$80=2^4\cdot 5,\quad 84=2^2\cdot 3\cdot 7,\quad 90=2\cdot 3^2\cdot 5,$$
$$98=2\cdot 7^2,\quad 100=2^2\cdot 5^2,\quad 120=2^3\cdot 3\cdot 5.$$

3 次式を使って有効な関係式をだすと

$$\begin{aligned}
80\cdot 106\cdot 120 &\equiv -2^3\cdot 3^2\cdot 7^3 &\cdots\cdots(1)\\
84\cdot 100\cdot 122 &\equiv -2^3\cdot 3^7 &\cdots\cdots(2)\\
86\cdot 98\cdot 122 &\equiv -2^8\cdot 5\cdot 11 &\cdots\cdots(3)\\
86\cdot 102\cdot 118 &\equiv -2^5\cdot 3^2\cdot 5^2 &\cdots\cdots(4)\\
86\cdot 106\cdot 114 &\equiv -2^{10}\cdot 3 &\cdots\cdots(5)\\
90\cdot 102\cdot 114 &\equiv 2^7\cdot 3\cdot 11 &\cdots\cdots(6)\\
98\cdot 102\cdot 106 &\equiv 2^7\cdot 3^3\cdot 5 &\cdots\cdots(7)\\
94^2\cdot 118 &\equiv 2^5\cdot 11 &\cdots\cdots(8)
\end{aligned}$$

このことを使って素因数分解の分かっていないものの形式的な素因数分解をすると

$$\begin{aligned}
\text{(8) より}\quad 118 &\equiv 2^5\cdot 11\cdot 94^{-2}\\
\text{(2) より}\quad 122 &\equiv -2^{-1}\cdot 3^6\cdot 5^{-2}\cdot 7^{-1}\\
\text{(3) より}\quad 86 &\equiv 2^8\cdot 3^{-6}\cdot 5^3\cdot 7^{-1}\cdot 11\\
\text{(4) より}\quad 102 &\equiv -2^{-8}\cdot 3^8\cdot 5^{-1}\cdot 7\cdot 11^{-2}\cdot 94^2\\
\text{(6) より}\quad 114 &\equiv -2^{14}\cdot 3^{-9}\cdot 7^{-1}\cdot 11^3\cdot 94^{-2}\\
\text{(5) より}\quad 106 &\equiv 2^{-12}\cdot 3^{16}\cdot 5^{-3}\cdot 7^2\cdot 11^{-4}\cdot 94^2
\end{aligned}$$

これらを、まだ使っていない関係式に代入して

$$\begin{aligned}
\text{(1) より}\quad 80\cdot 106\cdot 120 &\equiv -2^3\cdot 3^2\cdot 7^3\\
2^{-5}\cdot 3^{17}\cdot 5^{-1}\cdot 11^{-4}\cdot 94^2 &\equiv -2^3\cdot 3^2\cdot 7^3\\
3^{15}\cdot 94^2 &\equiv -2^8\cdot 5\cdot 7\cdot 11^4 \qquad (5.2)\\
\text{(7) より}\quad 98\cdot 102\cdot 106 &\equiv 2^7\cdot 3^3\cdot 5\\
-2^{-19}\cdot 3^{24}\cdot 5^{-4}\cdot 7^5\cdot 11^{-6}\cdot 94^4 &\equiv 2^7\cdot 3^3\cdot 5\\
3^{21}\cdot 7^5\cdot 94^4 &\equiv -2^{26}\cdot 5^5\cdot 11^6 \qquad (5.3)
\end{aligned}$$

(5.2),(5.3) の両辺をそれぞれ掛け合わせて

$$\begin{aligned}
3^{36}\cdot 7^5\cdot 94^6 &\equiv 2^{34}\cdot 5^6\cdot 7\cdot 11^{10}\\
3^{36}\cdot 7^4\cdot 94^6 &\equiv 2^{34}\cdot 5^6\cdot 11^{10}\\
(3^{18}\cdot 7^2\cdot 94^3)^2 &\equiv (2^{17}\cdot 5^3\cdot 11^5)^2\\
590^2 &\equiv 11139^2
\end{aligned}$$

となる。よって

$$(11139-590,43429)=137\,,\qquad (11139+590,43429)=317$$

より約数が見つかる。

しかし、実際にどの程度有効なのか (有効でないのか?) については今後の研究課題である。

関連図書

[1] 阿部英一,“代数学”, 培風館,(1977)

[2] R.P.Brent,“Parallel Algorithms for Integer Factorization”, London Math. Society Lecture Note 154(1990),26–37

[3] D.W.Davies and W.L.Price,“Security for Computer Networks”, John Wiley & Sons,(1984) (上園忠弘監訳,“ネットワーク・セキュリティ”, 日経マグロウヒル社,(1985))

[4] D.E.R.Denning,“Cryptography and Data Security”,Addison-Wesley,(1982) (上園忠弘・小嶋格・奥島昌子訳,“暗号とデータセキュリティ”, 培風館,(1988))

[5] 池山信一, 小山謙二,“現代暗号理論”, 電子通信学会編,(1986)

[6] 伊理正夫編,“数と式と文の処理”, 岩波書店,(1981)

[7] N. Koblitz,“A Course in Number Theory and Cryptogaphy”, Springer-Verlag,(1987)

[8] H.W.Lenstra, Jr.,“Primality Testing Algorithm”,Springer Lecture Note in Math. No.901(1981),243–257

[9] R.Lidl,“Some Mathmatical Aspects of Recent Advances in Cryptology”, London Math. Society Lecture Note 154(1990),1–8

[10] 松坂和夫,“代数系入門”, 岩波書店,(1976)

[11] 宮川洋・原島博・今井秀樹,“情報と符号の理論”, 岩波書店,(1982)

[12] 永尾汎,“群論の基礎”, 朝倉書店,(1967)

[13] A.Shamir,“A polynomial–time algorithm for breaking the Markle–Hellman cryptosystem”,IEEE trans. Inf. Theory,**IT–30**,5, 699–704,(1984)

[14] R.D.Silverman,“The Multipe Polynomial Quadratic Sieve”,Math. Comp. 48,329–339,(1987)

[15] 和田秀男,“コンピュータと素因子分解”, 遊星社,(1987)

[16] S.S.Wagstaff,Jr. and J.W.Smith,“Method of Factoring Large Integers”, Springer Lecture Notes in Math. No.1240(1984–85),281–303