暗号の数理要綱 #10

4.5　素因数分解

4.5.1　フェルマー法

素因数分解アルゴリズムの中でも，古くから知られているものの一つが **フェルマー法** である。この方法は，RSA で使われるような 100〜200 桁 以上の非常に大きな合成数 n であっても，$\sqrt{n}$ に非常に近い約数を持てば，短時間でそれを発見することができるという方法である。

その基礎となるのが，非常に大きな自然数 n に対して，$[\sqrt{n}]$ を短時間で計算するアルゴリズムである。ここで，実数 x に対して，$[x]$ は **ガウス記号** と言って，x 以下の整数の中で最大のものを表す。x 自身が整数なら，$[x] = x$ である。

4.5.2　$[\sqrt{n}]$ の計算アルゴリズム

与えられた自然数 n に対して，$r_1 = \left[\dfrac{1+n}{2}\right]$ とおき，数列 $r_1, r_2, \cdots, r_k, \cdots$ を

$$r_{k+1} = \left[\frac{r_k^2 + n}{2r_k}\right]$$

により帰納的に定義する。

補題 4.7　n と a を正の実数とすると

$$\left[\frac{a^2+n}{2a}\right] \geqq [\sqrt{n}]$$

が成立する。

証明:　任意の正の実数 a, n に対して

$$(a-\sqrt{n})^2 = a^2 + n - 2a\sqrt{n} \geqq 0$$

であるから，

$$a^2 + n \geqq 2a\sqrt{n}$$

が成り立つ。a は正の実数なので，この両辺を $2a$ で割ることにより，

$$\frac{a^2+n}{2a} \geqq \sqrt{n}$$

を得る。両辺のガウス記号をとっても，明らかに，この不等号はそのまま成立する。□

今の場合，n は自然数であり，従って正の実数なので，$a = 1$ とおいて 補題 4.7 を適用すると，

$$r_1 = \left[\frac{1+n}{2}\right] \geqq [\sqrt{n}] \geqq 1$$

であることがわかる。

また，a を r_k とおいて 補題 4.7 を適用することにより，任意の $k \geqq 1$ に関して，

$$r_k \geqq [\sqrt{n}]$$

であることがわかる。

補題 4.8　$r_k > [\sqrt{n}]$ ならば $r_k > r_{k+1}$ である。

証明:　一般に，整数 p と実数 q に対して，$p > [q]$ ならば $p > q$ である。なぜなら，p は整数なので，$p = [p]$ であり，$[p] > [q]$ であるから，$p > q$ となる。

今の場合，r_k は整数であり，$r_k > [\sqrt{n}]$ であると仮定しているので，$r_k > \sqrt{n}$ が成り立つ。$\sqrt{n} > 0$ なので，$r_k > \sqrt{n}$ の両辺を 2 乗すると $r_k^2 > n$ を得る。この両辺に r_k^2 を加えると $2r_k^2 > n + r_k^2$ を得る。r_k は正の数なので，この両辺を $2r_k$ で割ると，

$$r_k > \frac{r_k^2 + n}{2r_k}$$

を得る。r_{k+1} の定義より，

$$r_k > \frac{r_k^2 + n}{2r_k} \geqq \left[\frac{r_k^2 + n}{2r_k}\right] = r_{k+1}$$

であるので，$r_k > r_{k+1}$ が証明された。□

これらの 補題 より，数列 $\{r_1, r_2, \cdots, r_k, \cdots\}$ は，常に $r_k \geqq \sqrt{n}$ であり，$r_k = \sqrt{n}$ となるまでは，必ず真に減少して行く数列であることがわかった。従って，いつかは必ず $r_k = \sqrt{n}$ に行き着くことになる。すなわち，

> ある k があって，$r_k = \sqrt{n}$ となる。

ということがわかった。

$\sqrt{n}$ を得るためのアルゴリズムはわかったが，このアルゴリズムのスピードが問題である。n は 100 桁くらいの数を想定しているので，r_k が等差数列的な速度で減少するならば，何億年計算し続けても $\sqrt{n}$ には到達しないことも十分あり得る。

そこで次に，数列 $\{r_k\}$ がどのくらいのスピードで $\sqrt{n}$ に近づくのか，という問題を考えてみよう。

そのために，r_k と $\sqrt{n}$ の差，すなわち，

$$d_k = r_k - \sqrt{n}$$

という数を考える。これが 0 に近づくスピードを見てみよう。

注意：　なお，計算を簡略化するために，このあたりでは，ガウス記号を全て省いて計算する。ガウス記号のあるなしは ± 1 の差でしかないので，スピードの評価においては，本質的な影響はない。ガウス記号を含めて，厳密に評価することも，もちろん可能である。

d_k と d_{k+1} の比を計算してみると，

$$
\begin{aligned}
\frac{d_{k+1}}{d_k} &= \frac{r_{k+1}-\sqrt{n}}{r_k-\sqrt{n}} = \frac{\dfrac{r_k^2+n}{2r_k}-\sqrt{n}}{r_k-\sqrt{n}} \\
&= \frac{\dfrac{r_k^2+n-2r_k\sqrt{n}}{2r_k}}{r_k-\sqrt{n}} \\
&= \frac{\dfrac{(r_k-\sqrt{n})^2}{2r_k}}{r_k-\sqrt{n}} \\
&= \frac{r_k-\sqrt{n}}{2r_k} \\
&= \frac{1-\dfrac{\sqrt{n}}{r_k}}{2} < \frac{1}{2}
\end{aligned}
$$

となり，$\{d_k\}$ は 1 ステップごとに 1/2-倍以下になる，ということがわかった。

例えば，n が 200 桁くらいの数ならば，2 進数では 660 桁程度の数なので，$\sqrt{n}$ は 2 進数では 330 桁程度の数である。r_1 が n と同程度の数であったとしても，r_{330} あたりまでには $\sqrt{n}$ に到達できる，ということになる。

4.5.3　フェルマー法による因数分解

さて，ここでの目的は n を因数分解することであるが，$\sqrt{n}$ に近い数で n の約数を探してみる。

Step 1: $[\sqrt{n}]$ を計算する。もし $[\sqrt{n}]^2 = n$ ならば，n は平方数であり，$[\sqrt{n}]$ が n の約数となるので，目的は達成された。

Step 2: $[\sqrt{n}]^2 \neq n$ とする。

$$t = [\sqrt{n}] + 1\ , \qquad t^2 - n = r_1$$

とおき，$[\sqrt{r_1}]$ を計算する。もし $[\sqrt{r_1}]^2 = r_1$ なら，r_1 は平方数である。すなわち，$s = [\sqrt{r_1}]$ とおくと，$s^2 = r_1$ なので，

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

となって n は因数分解された。

Step 3: $[\sqrt{r_1}]^2 \neq r_1$ とすると，

$$t = [\sqrt{n}] + 2\ , \qquad t^2 - n = r_2$$

とおき，$[\sqrt{r_2}]$ を計算する。もし $[\sqrt{r_2}]^2 = r_2$ なら，r_2 は平方数である。すなわち，$s = [\sqrt{r_2}]$ とおくと，$s^2 = r_2$ なので，

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

となって n は因数分解された。

Step 4 以降:　あとは

$$t = [\sqrt{n}] + k\ , \qquad t^2 - n = r_k$$

として，この操作を時間の許す限り続ける。

n が合成数ならば，原理的にはこの方法で必ず因数分解ができることを説明しよう。

まず n は奇数であると仮定して良い。n が偶数なら，2 で割って行くことにより，2^m という因数を出してしまうことができ，残ったものは奇数となるからである。

$a \geqq b$ であって $n = ab$ であるとする。ここで a, b は素数でなくても良いが，n が奇数なので，a, b は共に奇数である。

$$t_0 = \frac{a+b}{2}\ , \qquad s_0 = \frac{a-b}{2}$$

とおくと，a, b が奇数であるから，$a+b, a-b$ は共に偶数となり，t_0, s_0 は共に整数である。

$$a = t_0 + s_0\ , \qquad b = t_0 - s_0$$

であるから，

$$n = (t_0 + s_0)(t_0 - s_0) = t_0^2 - s_0^2$$

となる。すなわち n は必ず平方数の差として表される。上のアルゴリズムにおいては，t は $[\sqrt{n}]$ から始めて 1 ずつその値を増やして試しているので，必ず求める t_0 にぶつかることになる。

このアルゴリズムを **フェルマー法** と言う。もし n が $\sqrt{n}$ に非常に近い約数を持てば，たとえ n 自身が非常に大きな数であっても，この方法によって，短時間で約数を発見できる可能性がある。

このことは，RSA 暗号において，$n = pq$ となる素数の選び方に対する一つの警告を与える。もし，p と q が非常に近い値なら，n は $\sqrt{n}$ に近い約数を持つことになり，この極めて古典的な因数分解法であるフェルマー法によって簡単に破られてしまうことになる。従って，p と q を選ぶ時は，桁数がある程度離れたものを採用しなければならない。