平成21年度 サイエンス・パートナーシップ・プロジェクト

暗号の基礎から実用まで

第3部：公開鍵暗号の基礎

高知大学 理学部

塩 田 研 一

2009年9月7日

3　公開鍵暗号の基礎

3.1　法演算

RSA 暗号では「法演算」という計算が用いられます。これは「法」と呼ばれる自然数 n をひとつ決めて、「n で割った余りが同じ数」は「同じ数」だと思って四則演算（加減乗除）の計算をしましょう、というものです。普通の計算とは違うので $=$ の代わりに $\equiv$ を用いて式を書きます。

例えば法が 7 なら

$$5+6 \equiv 4 \pmod 7$$
$$2-6 \equiv 3 \pmod 7$$
$$3\times 4 \equiv 5 \pmod 7$$
$$3\div 5 \equiv 2 \pmod 7$$

となります。（割り算はちょっと難しいかもしれませんが、

$$2\times 5 \equiv 10 \equiv 3$$

の両辺を 5 で割る、と考えます。）

練習問題

$$1+1 \equiv ? \pmod 2$$
$$0-1 \equiv ? \pmod 2$$
$$2+3 \equiv ? \pmod 5$$
$$2-3 \equiv ? \pmod 5$$
$$2\times 3 \equiv ? \pmod 5$$

3.2　フェルマの小定理

問題　サンプルプログラム「フェルマの小定理」を実行して、整数の n 乗を n で割った余りについて法則をみつけよ。

> **フェルマの小定理**　n が素数ならば、すべての整数 x に対して
>
> $$x^n \equiv x \pmod n$$
>
> が成り立つ。

という法則はみつかりましたか。

3.3　法べき乗

練習問題　$3^{100} \equiv ? \pmod 7$

解　フェルマの小定理から

$$3^7 \equiv 3 \pmod 7$$

が成り立つので、

$$3^{100} \equiv 3^7 \times 3^{93} \equiv 3 \times 3^{93} \equiv 3^{94} \pmod 7$$

同様にして

$$3^{94} \equiv 3^{88} \equiv 3^{82} \equiv \cdots \equiv 3^4 \equiv 81 \equiv 4 \pmod 7$$

よって答えは 4 です。

この解を一般化すると次の法則が得られます：

> **法則**　n を素数とすると、任意の整数 x に対して
>
> $$x \equiv x^n \equiv x^{2n-1} \equiv x^{3n-2} \equiv \cdots \pmod n$$
>
> が成り立つ。

ではこの n に色々な素数を入れてみましょう。

$$x \equiv x^5 \equiv x^9 \equiv x^{13} \equiv x^{17} \equiv \cdots \pmod 5$$
$$x \equiv x^7 \equiv x^{13} \equiv x^{19} \equiv x^{25} \equiv \cdots \pmod 7$$
$$x \equiv x^{11} \equiv x^{21} \equiv x^{31} \equiv x^{41} \equiv \cdots \pmod{11}$$
$$x \equiv x^{13} \equiv x^{25} \equiv x^{37} \equiv x^{49} \equiv \cdots \pmod{13}$$
$$x \equiv x^{17} \equiv x^{33} \equiv x^{49} \equiv x^{65} \equiv \cdots \pmod{17}$$
$$x \equiv x^{19} \equiv x^{37} \equiv x^{55} \equiv x^{73} \equiv \cdots \pmod{19}$$

これらの中でべき指数が合成数のところを書き出してみます。

$$x \equiv x^9 \pmod 5$$
$$x \equiv x^{25} \pmod 7$$
$$x \equiv x^{21} \pmod{11}$$
$$x \equiv x^{25} \pmod{13}$$
$$x \equiv x^{49} \pmod{13}$$
$$x \equiv x^{33} \pmod{17}$$
$$x \equiv x^{49} \pmod{17}$$
$$x \equiv x^{65} \pmod{17}$$

$$x \equiv x^{55} \ (\bmod 19)$$

更にべき指数を分解すると、

$$x \equiv (x^3)^3 \ (\bmod 5)$$
$$x \equiv (x^5)^5 \ (\bmod 7)$$
$$x \equiv (x^3)^7 \ (\bmod 11)$$
$$x \equiv (x^5)^5 \ (\bmod 13)$$
$$x \equiv (x^7)^7 \ (\bmod 13)$$
$$x \equiv (x^3)^{11} \ (\bmod 17)$$
$$x \equiv (x^7)^7 \ (\bmod 17)$$
$$x \equiv (x^5)^{13} \ (\bmod 17)$$
$$x \equiv (x^5)^{11} \ (\bmod 19)$$

こんな法則が見えてきましたね。

法則　n は素数、e と d は

$$e \times d \equiv 1 \ (\bmod (n-1))$$

を満たす自然数とする。このとき

$$y \equiv x^e \ (\bmod n) \ \Leftrightarrow \ x \equiv y^d \ (\bmod n)$$

が成り立つ。

証明　$e \times d \equiv 1 \ (\bmod (n-1))$ より、先の法則から

$$x \equiv x^{ed} \equiv (x^e)^d \ (\bmod n)$$

が成り立つ。これに $y \equiv x^e \ (\bmod n)$ を代入すると

$$x \equiv y^d \ (\bmod n)$$

が得られる。逆に

$$y \equiv y^{ed} \equiv (y^d)^e \ (\bmod n)$$

も成り立つので、これに $x \equiv y^d \ (\bmod n)$ を代入すると

$$y \equiv x^e \ (\bmod n)$$

が得られる。つまり、

$$x \equiv y^d \ (\bmod n)$$

と

$$y \equiv x^e \ (\bmod n)$$

は同値になる。

3.4　法べき乗暗号

素数の法 n と指数 e を決めて数 x ($0 \leqq x < n$) を

$$y = f(x) = \mathrm{MOD}(x^e, n)$$

という数 y に変換することを考えましょう。ここで $\mathrm{MOD}(x, n)$ は x を n で割った余りを表す記号です。

e が 1 でなければ y は x と全く違う数になりますから関数 $f(x)$ が暗号化関数として使えそうですね。

上述の法則から

$$e \times d \equiv 1 \ (\bmod (n-1))$$

を満たす d を見つけて

$$g(y) = \mathrm{MOD}(y^d, n)$$

という変換式を使えば $x = g(y)$ となって $g(y)$ が復号化関数になります。(e は $n-1$ と互いに素であることが必要です。)

e は「暗号化指数」、d は「復号化指数」と呼びます。

サンプルプログラム「法べき乗暗号鍵生成」「法べき乗暗号暗号化」「法べき乗暗号復号化」を用いて暗号化・復号化の実験をしてみましょう。

3.5　公開鍵暗号方式

昔から使われてきた暗号は、暗号の送り手（送信者）と受け手（受信者）が、お互いだけが知っている「暗号の鍵」を持っていました。

しかし、インターネットが発達して不特定多数の相手と通信する必要のある現代では、このような暗号では困ったことがあります：

- 膨大な個数の「鍵」を作る必要がある
- 通信相手とその「鍵」を打ち合わせる手立てがない

そこで考え出されたのが「公開鍵暗号方式」です。これは、暗号文をつくる「錠前」(= 暗号化鍵）と、暗号文を元の文章に戻す「鍵」(= 復号化鍵）を別々に構える方式です。


「錠前」は全世界に公開し、自分への通信はこの「錠前」を使って暗号化してもらいます。それを自分だけが持っている「鍵」を使って読むのです。

3.6 法べき乗暗号は公開鍵暗号ではない

法べき乗暗号は公開鍵暗号と言えるでしょうか。

暗号化関数には n と e という 2 つの数が使われています。復号化関数にはさらに d という数が必要です。しかし、e と $n-1$ を「てんびん」プログラムに入力すると

$$eu + (n-1)v = 1$$

という数 u, v が高速に計算できます。よく見るとこの式は

$$eu \equiv 1 \ (\bmod (n-1))$$

を意味しているので、この u が d になります。つまり、暗号化関数（錠前）の n と e を見せてしまうと復号化関数（鍵）の d もわかってしまいます。これでは公開鍵暗号とは言えません。

3.7 RSA 暗号

RSA 暗号は 1977 年に初めて発明された公開鍵暗号で、法べき乗暗号をもうひと捻りして設計されています。

RSA 暗号の暗号化関数 $f(x)$、復号化関数 $g(y)$ は法べき乗暗号と同じ形の式です：

$$\begin{cases} f(x) = \mathrm{MOD}(x^e, n) \\ g(y) = \mathrm{MOD}(y^d, n) \end{cases}$$

次の 3 点が法べき乗暗号とは違っています。

(1) 法べき乗暗号では法 n を素数にしていましたが、RSA 暗号では 2 つの素数 p, q の積を法 n にします。

(2) 暗号化指数 e は $(p-1)(q-1)$ と互いに素な数とします。

(3) e と $(p-1)(q-1)$ を「てんびん」プログラムに入力して

$$eu + (p-1)(q-1)v = 1$$

を満たす数 u, v ($u > 0$) を求め、この u を復号化指数 d とします。

(3) の式から

$$\begin{cases} ed \equiv 1 \ (\bmod (p-1)) \\ ed \equiv 1 \ (\bmod (q-1)) \end{cases}$$

が成り立ちます。すると、法べき乗の法則を導いた計算と同じようにして、$f(x)$ と $g(y)$ が互いに逆変換になっていることがわかります。すなわち $f(x)$ を暗号化関数に、$g(y)$ を復号化関数に用いることができるのです。

サンプルプログラム「RSA 暗号鍵生成」「RSA 暗号化」「RSA 復号化」を用いて暗号化・復号化の実験をしてみましょう。

3.8 RSA 暗号は公開鍵暗号である

RSA 暗号の復号化指数 d を計算するには n の素因数 p, q が必要になります。前回の話から素因数分解は「工夫しても高速にならない計算」ですので、暗号化関数の n, e からは事実上計算することができません。

したがって RSA 暗号は公開鍵暗号である、と言えるのです。

3.9 まとめ

- RSA 暗号の正規ユーザが使う計算
  - 鍵生成：素数判定、てんびんクイズ
  - 変換式：法べき乗... いずれも高速
- RSA 暗号への攻撃者が使う計算
  - 素因数分解... 天文学的な時間が掛かる

今日の後半はこの RSA 暗号を使ったメールツールの実習です。