RSA暗号

鍵の生成

1. 相異なる2つの素数 p, q を求め、その積を n とする
2. $\varphi(n) = (p-1)(q-1)$ と互いに素な正整数を e とする。
3. $ed \equiv 1 \pmod{\varphi(n)}$ をみたす d をとる。
4. n と e を公開, d, p, q は非公開とする。

暗号化

5. 平文 x (数値化されたもの)を用意。
6. $y \equiv x^e \pmod{n}$ を計算し、y を暗号文とする。

復号

7. d を実際に計算する。
8. $z \equiv y^d \pmod{n}$ とすると $x = z$ である。

命題5.1 手順8で $x = z$ となる

∵ $x \leq n$ は仮定する。

$$z \equiv y^d \equiv (x^e)^d = x^{ed} \equiv x \pmod{n}$$

Thm.4.14. $g \equiv 1 \pmod{\varphi(n)}$
$\Rightarrow a^g \equiv a \pmod{n}$. //

注意5.2

1. 大きな素数の効率良いアルゴリズムがある。

2. 手順2で e を求めるには、$p-1, q-1$ の最小公倍数 l を求め、
l と素な l より小さな整数 e を選べばよい。
なお、$\gcd(a,b)\cdot \mathrm{lcm}(a,b)=a\cdot b$ より、l は高速に求められる。

3. 手順3.7の d をみたす方法は、1次合同式のところでみた。
(ユークリッドの互除法の発展版)

4. 手順6.8の計算は、

$$x^k=\begin{cases} x & (k=1) \\ (x^{\frac{k}{2}})^2 & (k\geq 2,\ 偶数) \\ (x^{\frac{k-1}{2}})^2\cdot x & (k\geq 3,\ 奇数) \end{cases}$$ とすれば、

$\log_2 k$ 回の反復ですむので、e、d の桁数に比例する回数で行える。

注意5.3.

RSAの安全性について。
復号鍵は d なので、d が求まればよいが、そのためには
$\varphi(n)$ がわからないといけない。このためには、n の素因数分解 $n=pq$ が必要。
⟶ この難しさがRSA暗号の根拠になる。

今だと $e^{c\cdot(\log n\cdot \log(\log n))^{\frac{1}{2}}}$ くらいかかる

⟶ 100桁、1ステップ $1\mu s$ で10億年くらい。

例. $p=7$, $q=11$ とすると　$n=pq=77$.

$\varphi(n)=(7-1)(11-1)=60$ である.

$\mathrm{lcm}(6,10)=30$ より. $e=7$ としてみる.

ここで $7d\equiv 1 \mod \varphi(n)=60$ を求めると.

$$d\equiv 7^{59}\equiv 43 \pmod{60}$$ となる.

さて. $x=3$ とすると.

$$x^e=3^7=2187\equiv 31 \pmod{n}$$ より $y=31$.

復号すれば.

$$x\equiv 31^{43} \pmod{77}\equiv 3$$ となる.

ここで

$$31^{43}=31\cdot(31^{21})^2$$

$$31^{21}=31\cdot(31^{10})^2$$

$$31^{10}=(31^5)^2.$$

$$31^5=31\cdot(31^2)^2.$$ と計算すればよい.