RSA(Rivest–Shamir–Adleman)

RSA 算法加/解密速度较慢, 常用于传输对称算法, 再由对称算法加/解密数据

算法模型: $m/d/e$ 为三个极大数, 已知 $m/n/e(0 \leqslant m < n)$, 求 d 的值

$$(m^d)^e \equiv m \quad (mod \quad n)$$

原理: m 代表需要加密的内容, n/e 代表公钥, d 代表私钥 (n 也包含在私钥中)

密钥生成步骤:

1. 随机生成两个不同素数 p/q, 两个数相同量级
2. 计算 $n = pq$, 即 RSA bits
3. 计算 $\lambda(n)$, $\lambda(n) = lcm(\lambda(p), \lambda(q))$, 又 p/q 为素数, 所以 $\lambda(n) = lcm(p-1, q-1)$
4. 选取整数 e, 并且满足 $1 < e < \lambda(n)$ 和 $gcd(e, \lambda(n)) = 1$
5. 获得 d 值, 通过 $d \cdot e \equiv 1(mod \quad \lambda(n))$

加密步骤:

初始内容为 M, 通过 padding schemes(即在 begin/middle/end 位置添加 bit 内容), 转化为 m

$$c \equiv m^e \quad (mod \quad n)$$

解密步骤:

$$c^d \equiv (m^e)^d \equiv m \quad (mod \quad n)$$

示例:

生成密钥:

1. 选择 p/q 值

   $p = 61$,$q = 53$
2. 计算 $n = pq$

   $n = 61 \times 53 = 3233$
3. 计算 $\lambda(n)$(根据 Carmichael function)

   $\lambda(3233) = lcm(60, 52) = 780$

4. 选择 e, 满足 $1 < e < 780$, 并且与 $\lambda(n)$ 互质

e=17

5. 计算 d

$$d \times 17 \equiv 1 \pmod{780} \qquad \Longrightarrow \qquad d = 413$$

加密:

假设 $m = 65$

$c = m^e \ mod \ n = 65^{17} \ mod \ 3233 = 2790$

解密:

$m = c^d \ mod \ n = 2790^{413} \ mod \ 3233 = 65$