4-7反馈

马骏

majun@nju.edu.cn

12. Find integers n, E, and X such that

$$X^E \equiv X \pmod{n}.$$

Is this a potential problem in the RSA cryptosystem?

- E.g.
  - n=6,E=3,X=2
  - X=1
  - …
- 进一步分析
  - $X^E \equiv X\ (mod\ n)$
  - $\Rightarrow X^{E-1} \equiv 1 (mod\ n)$
  - $\Rightarrow ord(X) | E-1$
  - 而$ord(X) | \phi(n)$
  - $\therefore \gcd\big(E-1, \phi(n)\big) = 1$时没有影响

31.7-2

Prove that if Alice's public exponen[illegible]and an adversary obtains Alice's secret exponent d, where $0 < d < \phi(n)$, then the adversary can factor Alice's modulus n in time polynomial in the number of bits in n. (Although you are not asked to prove it, you may be interested to know that this result remains true even if the condition $e = 3$ is removed. See Miller [255].)

- 已知$e = 3, 0 < d < \phi(n)$,多项式时间内确定p, q s.t. $pq = n$
- 基本思路:
  - $ed = 1 \ mod \ \phi(n) \Rightarrow ed = k(\phi(n)) + 1 = k(p-1)(q-1) + 1$
  - 又$\because e = 3, 0 < d < \phi(n)$
  - $\therefore 0 < 3d = k(\phi(n)) + 1 < 3\phi(n)$
  - $\therefore k$只可能等于1或2，分别针对k=1,2两种情况进一步处理：
    - $\because ed = k(p-1)(q-1) + 1$①
    - 又$\because pq = n$,即$q = n/p$②
    - ②带入①得到：$ed = k(p-1)\left(\frac{n}{p} - 1\right) + 1$ 关于p的一元二次方程

31-2 *Analysis of bit operations in Euclid's algorithm*

a. Consider the ordinary "paper and pencil" algorithm for long division: dividing a by b, which yields a quotient q and remainder r. Show that this method requires $O((1 + \lg q) \lg b)$ bit operations.

b. Define $\mu(a, b) = (1 + \lg a)(1 + \lg b)$. Show that the number of bit operations performed by EUCLID in reducing the problem of computing $\gcd(a, b)$ to that of computing $\gcd(b, a \bmod b)$ is at most $c(\mu(a, b) - \mu(b, a \bmod b))$ for some sufficiently large constant $c > 0$.

c. Show that EUCLID(a, b) requires $O(\mu(a, b))$ bit operations in general and $O(\beta^2)$ bit operations when applied to two β-bit inputs.

31-2 Analysis of bit operations in Euclid's algorithm

a. Consider the ordinary "paper and pencil" algorithm for long division: dividing a by b, which yields a quotient q and remainder r. Show that this method requires $O((1+\lg q)\lg b)$ bit operations.

$$37\,\overline{)\,1260257}$$

```
            1110  r.  11
1101)  10111001
        1101
        10100
         1101
          1110
          1101
            11
```

a 考虑 a,b 的二进制表示, 他们的长度是 $\lg a,\lg b$. 在长除法的一次迭代中要做[redacted]与减法, 这一步的位操作次数是 $O(\lg b)$.

注意到每计算出商的一位, 至多需要长除法的一次迭代. 加上[redacted], 长除法至多有 $O(\lg q+1)$ 次迭代. 所以总位操作次数是 $O((\lg q+1)\lg b)$

b. Define $\mu(a, b) = (1 + \lg a)(1 + \lg b)$. Show that the number of bit operations performed by EUCLID in reducing the problem of computing $\gcd(a, b)$ to that of computing $\gcd(b, a \bmod b)$ is at most $c(\mu(a, b) - \mu(b, a \bmod b))$ for some sufficiently large constant $c > 0$.

- 由a)可得$\gcd(a, b) \Rightarrow \gcd(b, a\ mod\ b)$的复杂度为$O((1 + \lg q) \lg b)$，即存在常数c使得操作总数$M(a, b) \le c\big((1 + \lg q) \lg b\big)$
- 又$\because \lg q \le \lg a - \lg b$, $\lg r = \lg(a\ mod\ b) < \lg b$
- $\therefore \lg q + \lg r < \lg a$, $1 + \lg q \le \lg a - \lg r$
- 而$c\big(\mu(a, b) - \mu(b, a\ mod\ b)\big) = c\big((1 + \lg a)(1 + \lg b) -$

c. Show that EUCLID(a,b) requires $O(\mu(a,b))$ bit operations in general and $O(\beta^2)$ bit operations when applied to two β-bit inputs.

- 由b)可得$M(a,b) \le c\big(\mu(a,b) - \mu(b, a\ mod\ b)\big) = c\big(\mu(a_0,b_0) - \mu(a_1,b_1)\big)$
  - $a_0 = a, b_0 = b; a_{i+1} = b_i, b_{i+1} = a_i\ mod\ b_i$
- 总开销$T(a,b) = M(a_0,b_0) + T(a_1,b_1)$

$$
\begin{aligned}
&\le c\big(\mu(a_0,b_0) - \mu(a_1,b_1)\big) + T(a_1,b_1) \\
&\le c\big(\mu(a_0,b_0) - \mu(a_1,b_1)\big) + c\big(\mu(a_1,b_1) - \mu(a_2,b_2)\big) + T(a_2,b_2) \\
&\le c\big(\mu(a_0,b_0) - \mu(a_2,b_2)\big) + T(a_2,b_2) \\
&\le \cdots \\
&= c\big((\mu(a_0,b_0) - \mu(a',0)\big) = O(\mu(a,b))
\end{aligned}
$$

31-3 *Three algorithms for Fibonacci numbers*

This problem compares the efficiency of three methods for computing the nth Fibonacci number F_n, given n. Assume that the cost of adding, subtracting, or multiplying two numbers is $O(1)$, independent of the size of the numbers.

a. Show that the running time of the straightforward recursive method for computing F_n based on recurrence (3.22) is exponential in n. (See, for example, the FIB procedure on page 775.)

b. Show how to compute F_n in $O(n)$ time using memoization.

c. Show how to compute F_n in $O(\lg n)$ time using only integer addition and multiplication. (*Hint:* Consider the matrix

$$\begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}$$

and its powers.)

a. Show that the running time of the straightforward recursive method for computing F_n based on recurrence (3.22) is exponential in n. (See, for example, the FIB procedure on page 775.)

```
FIB(n)
if n ≤ 1
        return n
else x = FIB(n − 1)
        y = FIB(n − 2)
        return x + y
```

$$T(n) = T(n-1) + T(n-2) + 1$$

假设$T(n) \le c \cdot 2^n$

Base:

$T(0) = 0 \le c$

$T(1) = 1 \le c \cdot 2$

H:

$\forall k < n, T(k) \le c \cdot 2^k$成立

I:

$T(n) = T(n-1) + T(n-2) + 1$

$\le 2T(n-1) \le 2c \cdot 2^{n-1} = c \cdot 2^n$

$\therefore T(n) = O(2^n)$

假设$T(n) \ge c \cdot 2^{n/2} + k$

Base:

$T(0) = 0 \ge c + k$

$T(1) = 1 \ge c \cdot \sqrt{2} + k$

H:

$\forall k < n, T(k) \ge c \cdot 2^k$成立

I:

$$
\begin{aligned}
T(n) &= T(n-1) + T(n-2) + 1 + k \\
&\ge 2T(n-2) + k \\
&\ge 2c \cdot 2^{\frac{n-2}{2}} + k \\
&= c \cdot 2^{\frac{n}{2}} + k
\end{aligned}
$$

$\therefore T(n) = \Omega(2^{n/2}) = \Omega((\sqrt{2})^n)$

b. Show how to compute F_n in $O(n)$ time using memoization.

DP

c. Show how to compute F_n in $O(\lg n)$ time using only integer addition and multiplication. (*Hint:* Consider the matrix

$$\begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}$$

and its powers.)

将 Fibonacci 递推公式写成矩阵形式，有

$$\begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}^k = \begin{pmatrix} F_{k-1} & F_k \\ F_k & F_{k+1} \end{pmatrix}.$$

故而问题转化为求矩阵的幂，使用快速幂算法可以在 $\Theta(\lg n)$ 的时间内求得结果.

d. Assume now that adding two β-bit numbers takes $\Theta(\beta)$ time and that multiplying two β-bit numbers takes $\Theta(\beta^2)$ time. What is the running time of these three methods under this more reasonable cost measure for the elementary arithmetic operations?

```
FIB(n)
if n ≤ 1
      return n
else x = FIB(n − 1)
      y = FIB(n − 2)
      return x + y
```

每次调用固定开销：一次加法

Let $n = 2\hat{}\beta$，粗略估计

$$T(n) = O(2^n) * \Theta(\beta) = O(2^n \lg n)$$

详细分析，

$$T(n) = T(n-1) + T(n-2) + c\lg n$$

$$\le 2T(n-1) + c\lg n$$

$$\le \sum_{i=1,\dots,n} 2^{n-i} \cdot c\lg i$$

DP

每次调用固定开销：一次加法

$$M(n) < c\lceil \lg n \rceil$$

$$T(n) = \sum_{i=2,\dots,n} M(i)$$

$$\le \sum_{i=2,\dots,n} c\lceil \lg i \rceil = c\lg n!$$

$$= O(n\lg n)$$

$$\begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}^k = \begin{pmatrix} F_{k-1} & F_k \\ F_k & F_{k+1} \end{pmatrix}$$

每次矩阵乘法开销：

- 8次乘法
- 4次加法

Let $n = 2\hat{}\beta$

$$T(n) = T\left(\frac{n}{2}\right) + \left(8\Theta(\lg^2 n) + 4\Theta(\lg n)\right)$$

$$= T\left(\frac{n}{2}\right) + \Theta(\lg^2 n) = c\sum_{i=0\dots\beta} \lg^2 2^i$$

$$= c\sum_{i=0\dots\beta} i^2 \cdot \lg^2 2 = c\lg^2 2 \sum_{i=0\dots\beta} i^2$$

$$= \Theta(\beta^3) = \Theta(\lg^3 n)$$

31.7-3 ★

Prove that RSA is multiplicative in the sense that

$$P_A(M_1)P_A(M_2) \equiv P_A(M_1M_2) \pmod{n} .$$

Use this fact to prove that if an adversary had a procedure that could efficiently decrypt 1 percent of messages from $\mathbb{Z}_n$ encrypted with P_A, then he could employ a probabilistic algorithm to decrypt every message encrypted with P_A with high probability.

$$P_A(M_1)P_A(M_2) = \left(M_1^e \bmod n\right)\left(M_2^e \bmod n\right) = M_1^e M_2^e \bmod n = (M_1M_2)^e \bmod n$$
$$= P_A(M_1M_2)$$

31.7-3 ★

Prove that RSA is multiplicative in the sense that

$$P_A(M_1)P_A(M_2) \equiv P_A(M_1M_2) \pmod{n} .$$

Use this fact to prove that if an adversary had a procedure that could efficiently decrypt 1 percent of messages from $\mathbb{Z}_n$ encrypted with P_A, then he could employ a probabilistic algorithm to decrypt every message encrypted with P_A with high probability.

假设已知$P_A(M_2), P_A(M_1M_2)$以及M_2, M_1M_2求M_1?

$$M_1 = M_1M_2M_2^{-1}$$

但目前我们仅已知: $M \in \mathbb{S} \subset \mathbb{M}$及其$P(M) \subset \mathbb{P}_{\mathbb{S}}, |\mathbb{S}|/|\mathbb{M}| = 0.01$

如果$\boldsymbol{M_1M_2} \in \mathbb{S}$**?**

判定一个元素$\boldsymbol{M} \in \mathbb{S}$，是容易的！

我们目标：构造$\boldsymbol{M_1M_2} \in \mathbb{S}$

Ok, so the attacker has a way of calculating M from $P_A(M)$, if $P_A(M)$ is in a set S that covers about 1 per cent of the residue classes modulo n.

The attacker, facing the task of calculating M_1, given $P_A(M_1)$ can then generate a few hundred random $(M_2, P_A(M_2))$ pairs. S/he can then check, whether $P_A(M_1)P_A(M_2)$ is in the set S for any M_2. If that happens, the attacker will know

$$M_1M_2 = P_A^{-1}(P_A(M_1)P_A(M_2))$$

AND s/he will know M_2, so figuring out M_1 is then easy.

If the choices for M_2 were truly random, the probabilities of failure with each M_2 are independent from each other, and all about 0.99. So with, say, 200 trials, the probability of failure is $0.99^{200} \approx e^{-2}$ or about 13 per cent. Make four hundred attempts, if that is not good enough.

http://math.stackexchange.com/questions/170201/rsa-probabilistic-decryption-problem

```
PSEUDOPRIME(n)
if MODULAR-EXPONENTIATION(2, n − 1, n) ≢ 1 (mod n)
    return COMPOSITE        // definitely
else return PRIME           // we hope!
```

We say that n is a ***base-a pseudoprime*** if n is composite and

$$a^{n-1} \equiv 1 \pmod{n} . \tag{31.40}$$

- **If n is a prime, then for $\forall a \in \mathbb{Z}_n^*, a^{n-1} \equiv 1\ (mod\ n)$?**
- **If for $\forall a \in \mathbb{Z}_n^*, a^{n-1} \equiv 1\ (mod\ n)$, then n is a prime?**

e.g. 561=3*11*17

Carmichael Numbers:
composite number satisfying $\forall a \in \mathbb{Z}_n^*, a^{n-1} \equiv 1\ (mod\ n)$

31.8-2 ★

It is possible to strengthen Euler's theorem slightly to the form

$$a^{\lambda(n)} \equiv 1 \pmod{n} \text{ for all } a \in \mathbb{Z}_n^* ,$$

where $n = p_1^{e_1} \cdots p_r^{e_r}$ and $\lambda(n)$ is defined by

$$\lambda(n) = \operatorname{lcm}(\phi(p_1^{e_1}), \ldots, \phi(p_r^{e_r})) . \quad (31.42)$$

A composite number n is a Carmichael number if $\lambda(n) \mid n - 1$. The smallest Carmichael number is $561 = 3 \cdot 11 \cdot 17$; here, $\lambda(n) = \operatorname{lcm}(2, 10, 16) = 80$, which divides 560. Prove that Carmichael numbers must be both "square-free" (not divisible by the square of any prime) and the product of at least three primes. (For this reason, they are not very common.)

$$\phi(p_i^{e_i}) = p_i^{e_i}(1 - 1/p_i)$$

$$\phi(n) = n \prod_{p \text{ is prime and } p|n} (1 - 1/p) = n \prod_{p_i, 1 \le i \le r} (1 - 1/p_i) = \prod_{p_i, 1 \le i \le r} p_i^{e_i}(1 - 1/p_i) = \prod_{p_i, 1 \le i \le r} \phi(p_i^{e_i})$$

$\forall i = 1 \sim r, \phi(p_i^{e_i}) | \phi(n)$, 即$\phi(n)$为$\phi(p_1^{e_2}), \phi(p_2^{e_2}), \ldots \phi(p_r^{e_r})$的公倍数，
所以$\lambda(n) | \phi(n)$

Prove that Carmichael numbers must be both “square-free” (not divisible by the square of any prime)

- 基本想法？
  - 反证法
    - 假设存在一个Carmichael Number N，使得N包含一个因子p^e(p为一个素数, $e \geq 2$)
    - $\because \lambda(p^e)|N-1$, 且$\lambda(p^e) = \phi(p^e) = p^e\left(1-\frac{1}{p}\right)$
    - $\therefore p^e\left(1-\frac{1}{p}\right)|N-1$，即$p^{e-1}(p-1)|N-1$
    - $\therefore p^{e-1}|N-1$，即$N-1 = \mathrm{k}\cdot p^{e-1}$
    - $\therefore N-1 \equiv \mathrm{k}\cdot p^{e-1}\ (mod\ p^{e-1})$，即$N-1 \equiv 0\ (mod\ p^{e-1})$
    - 又$\because p^e|N$
    - $\therefore -1 \equiv 0\ (mod\ p^{e-1})$, 矛盾

Prove that Carmichael numbers must be the product of at least three primes.

- 基本想法?
  - 反证法

Proof. Because a Carmichael number is without square factor and is not prime it has at least two prime factors. Let us assume that $n = pq$ with $p < q$. Then $q-1$ divides $pq - 1 = p(q-1) + p - 1$ so $q - 1$ divides $p - 1$. Absurd. $\square$