第 4-6 讲: 加密算法

姓名: 林凡琪　　**学号:** 211240042

评分: ______　　**评阅:** ______

2023 年 4 月 5 日

请独立完成作业，不得抄袭。
若得到他人帮助, 请致谢。
若参考了其它资料，请给出引用。
鼓励讨论，但需独立书写解题过程。

1　作业 (必做部分)

题目 1 (TJ 7-7(a,b))

解答:
(a)$n = 3551, E = 629, x = 31$
$x^E \mod n = 31^{629} \mod 3551 = 2791$
(b)$n = 2257, E = 46, x = 23$
$x^E \mod n = 23^{47} \mod 2257 = 769$

题目 2 (TJ 7-9(b))

解答:

$$y^D \mod n = 34^{81} \mod 5893 = 2014$$

题目 3 (TJ 7-12)

解答:
极端情况举例:

$$n = 5 \times 11 = 55$$
$$m = 4 \times 10 = 40$$
$$E = 21$$
$$D = 21$$

所以,

$$\forall X, X^E \equiv X \mod n$$

$$X^E \equiv X \mod n \Rightarrow X\left(X^{E-1}-1\right) = 0 \mod n$$

因为 $\gcd(X,n)$ and $\gcd\left(X^{E-1}-1,n\right)$ 可能有任意一个不是 1. 计算 $\gcd(X,n)$ and $\gcd\left(X^{E-1}-1,n\right)$, 我们能得到 n 的因子.

题目 4 (TC 31.7-1)

解答:
$\phi(n) = (p-1)\cdot(q-1) = 280.$
$d = e^{-1} \mod \phi(n) = 187.$
$P(M) = M^e \mod n = 254.$
$S(C) = C^d \mod n = 254^{187} \mod n = 100.$

题目 5 (TC 31.7-2)

解答:

$$ed \equiv 1 \mod \phi(n)$$

$$ed - 1 = 3d - 1 = k\phi(n)$$

如果 $p, q < n/4$, 那么

$$\phi(n) = n - (p+q) + 1 > n - n/2 + 1 = n/2 + 1 > n/2.$$

$kn/2 < 3d - 1 < 3d < 3n$, then $k < 6$, then we can solve $3d - 1 = n - p - n/p + 1$.

题目 6 (TC Problem 31-3)

解答:
a. 为了解决 FIB(n), 我们需要计算 FIB($n-1$) 和 FIB($n-2$)。因此，我们有递归式

$$T(n) = T(n-1) + T(n-2) + \Theta(1).$$

我们可以得到斐波那契数列的上界为 $O(2^n)$，但这不是紧密的上界。
斐波那契递推式定义为

$$F(n) = F(n-1) + F(n-2).$$

这个函数的特征方程将是

$$x^2 = x + 1 \; x^2 - x - 1 \quad = 0.$$

我们可以通过二次公式得到根：$x = \frac{1\pm\sqrt{5}}{2}$。
我们知道线性递归函数的解为

$$F(n) = \alpha_1^n + \alpha_2^n \quad = \left(\frac{1+\sqrt{5}}{2}\right)^n + \left(\frac{1-\sqrt{5}}{2}\right)^n,$$

其中 α_1 和 α_2 是特征方程的根。
由于 $T(n)$ 和 $F(n)$ 都表示同一件事情，它们在渐近意义下是相同的。
因此，

$$T(n) = \left(\frac{1+\sqrt{5}}{2}\right)^n + \left(\frac{1-\sqrt{5}}{2}\right)^n \quad = \left(\frac{1+\sqrt{5}}{2}\right)^n \approx O(1.618)^n.$$

b.

```
FIBONACCI(n)
let fib[0..n] be a new array
fib[0] = 1
fib[1] = 1
for i = 2 to n
    fib[i] = fib[i - 1] + fib[i - 2]
return fib[n]
```

c. 假设所有整数乘法和加法都可以在 $O(1)$ 的时间内完成。首先，我们要证明

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

通过归纳，

$$\begin{pmatrix}0 & 1\,1 & 1\end{pmatrix}^{k+1} = \begin{pmatrix}0 & 1\,1 & 1\end{pmatrix}\begin{pmatrix}0 & 1\,1 & 1\end{pmatrix}^k \quad = \begin{pmatrix}0 & 1\,1 & 1\end{pmatrix}\begin{pmatrix}F_{k-1} & F_k\,F_k & F_{k+1}\end{pmatrix}^k = \Big(F_k \quad F_{k+1}\,F_{k-1}+F_k \quad F_k + F$$

我们证明我们可以在 $O(\lg n)$ 时间内计算给定矩阵的 $n-1$ 次幂，右下角的元素是 F_n。

我们应该注意，通过 8 次乘法和 4 次加法，我们可以将任何两个 2×2 矩阵相乘，这意味着矩阵乘法可以在常数时间内完成。因此，我们只需要限制算法中这些操作的数量。

运行算法 MATRIX-POW$(A, n-1)$ 需要 $O(\lg n)$ 时间，因为我们在每个步骤中将 n 的值减半，并且在每个步骤中，我们执行恒定数量的计算。

递归式为

$$T(n) = T(n/2) + \Theta(1).$$

```
MATRIX-POW(A, n)
if n % 2 == 1
    return A * MATRIX-POW(A^2, (n - 1) / 2)
return MATRIX-POW(A^2, n / 2)
```

d. 首先，我们应该注意到 $\beta = O(\log n)$。对于第 (a) 部分，我们朴素地添加一个每次都在指数级增长的 β 位数，因此递归变为

$$\begin{aligned} T(n) &= T(n-1) + T(n-2) + \Theta(\beta) \\ &= T(n-1) + T(n-2) + \Theta(\log n), \end{aligned}$$

因为 $\Theta(\log n)$ 可以被指数时间吸收，所以它的解与 $T(n) = O\left(\frac{1+\sqrt{5}}{2}\right)^n$ 相同,。
对于第 (b) 部分，记忆化版本的递归变为

$$M(n) = M(n-1) + \Theta(\beta).$$

这有一个解 $\sum_{i=2}^{n} \beta = \Theta(n\beta) = \Theta(2^{\beta} \cdot \beta)$ or $\Theta(n \log n)$.
对于第（c）部分，我们执行恒定数量的加法和乘法。递归变为

$$P(n) = P(n/2) + \Theta(\beta^2),$$

它的解为 $\Theta(\log n \cdot \beta^2) = \Theta(\beta^3)$ or $\Theta(\log^3 n)$。

---

2 作业 (选做部分)

题目 1 (TC Problem 31-4)

解答:

---

3 Open Topics

Open Topics 1 (中国剩余定理)
向同学介绍中国剩余定理及其应用。

Open Topics 2 (椭圆曲线加密 (Elliptic Curve Cryptography, ECC))
椭圆曲线加密是基于椭圆曲线数学理论实现的一种非对称加密算法。相比 RSA，ECC 优势是可以使用更短的密钥，来实现与 RSA 相当或更高的安全。

(参考资料-1：https://medium.com/dev-genius/introduction-to-elliptic-curve-cryptography-567e47b0e49e) (参考资料-2：https://en.wikipedia.org/wiki/Elliptic-curve_cryptography) (参考资料-3：https://www.jianshu.com/p/e41bc1eb1d81)

4 反馈