CS381 Exercise 7

Name: Zhang Yupeng

Student ID: 5130309468

1. PayTV systems require that only the paid customers can watch the program, which of the 5 security services can be used to achieve this goal?

Solution:

1. Authentication is required to verify the identities of customers who have paid.
2. Access control is required to make sure only paid customers can watch.
3. Data integrity is required to make sure the service is at good and legal condition.

2. Determine the complexity (number of arithmetic operations) of:

1. **computing** $gcd(a, b)$**;**

   Solution:

   Assume that $a > b >= 1$, construct the number seires: $u_0 = a, u_1 = b, u_k = u_{k-2}$.

   Obviously, if the algorithm will do n mod operations, $u_n = gcd(a, b), u_{n+1} = 0$. Compare $\{u_n\}$ and Fibonacci series $\{F_n\}, F_0 = 1 \leq u_n, F_1 = 1 \leq u_{n-1}$, and u_k, so $u_k \leq u_{k+1} + u_{k+2}$, thus $u_k \leq F_{n-k}$, thus $a = u_0 \geq F_n, b = u_0 \geq F_{n-1}$.

   Therefore, if $b < F_{n-1}$, the number of mod operations needed is less than n. According to the properties of Fibonacci series, $F_{n-1} > \frac{1.618n}{\sqrt{5}}$, so the time complexity is $O(\log b)$.

2. **computing RSA encryption** $C = M^e$ **mod** n

   Solution:

   Considering the square-multiplication algorithm for RSA encryption, every multiplication operation needs $O(k^2)$ complexity where k is the text length and the algorithm needs $\log e$ multiplications. Thus, the time complexity is $O(k_2 \log e)$.

3. Limitation of raw RSA signature: Only when M has redundancy structure, can the signature be securely verified. Why?

Solution:

If M doesn't have redundancy structure, it is susceptible to existential forgeries.

Let (e, N) be the public signature verification key of RSA, then one can randomly choose a signature σ and compute the message $m = \sigma^e \pmod{N}$.

Applying a redundancy structure to messages, for example, hashing and padding prior to signing, the forged signatures would be useless so that the signature can be securely verified.

4. For RSA, it requires $|p - q|$ should not be small. Task: design an attack if $|p - q|$ is smaller than 10000.

Solution:

If $|p - q|$ is too small, we can use Fermat factoring method to calculate p and q quickly.

The Fermat factoring method works as follows: for $a = \lceil \sqrt{n} \rceil, \lceil \sqrt{n} \rceil + 1, \lceil \sqrt{n} \rceil + 2, \ldots$, it checks whether n/a^2 is a perfect square; if so, it has factored nn.

We can analyze the running time of Fermat's method. Let $\epsilon = (p/\sqrt{n}) - 1$, so that $p = \sqrt{n}(1 + \epsilon)$ and $q = \sqrt{n}/(1 + \epsilon) = \sqrt{n}(1 - \epsilon + \epsilon^2 - \cdots)$. Fermat's method succeeds when $a = (p + q)/2 = \sqrt{n}(1 + \epsilon^2/2 - \cdots)$. In other words, it requires $\approx \sqrt{n}\epsilon^2/2$ iterations.

So, since $|p - q| \approx 2\sqrt{n}\epsilon$, if $|p - q| < 10000$, we can get that:

$$\begin{cases} 2\sqrt{n}\epsilon < 10000 \\ \sqrt{n}\epsilon^2/2 = t \end{cases}$$

Then, we get get the conclusion that:

$$t < \frac{10^8\sqrt{N}}{2}$$

So, p, q can be calculated in resonable time, then the RSA could be broken in limited time.

5. Show that in RSA, knowing $\phi(n)$ is equivalent to knowing the factorization of n.

Solution:

From the definition of the totient function, we have the relation:

$$\phi(n) = (p - 1)(q - 1) = pq - p - q + 1 = (n + 1) - (p + q)$$

It then easily follows that:

$$(n + 1) - \phi(n) = p + q$$

$$(n+1) - \phi(n) - p = q$$

Since $n = pq$, so $p^2 - (n + 1 - \phi(n))p + n = 0$, this is a quadratic equation in p, with:

$$\begin{aligned} a &= 1 \\ b &= -(n + 1 - \phi(n)) \\ c &= n \end{aligned}$$

$$p = \frac{-b \pm \sqrt{|b|^2 - 4ac}}{2a} = \frac{(n + 1 - \phi(n)) \pm \sqrt{|n + 1 - \phi(n)|^2 - 4n}}{2}$$

In conclusion, knowledge of $\phi(n)$ allows one to factor n in time $O(1)$. The other answers are equivalent.