Assignments 6 Solutions

1. (Trappe page 160: 10) Suppose two users Alice and Bob have the same RSA modulus n and suppose that their encryption exponents e_A and e_B are relatively prime. Charles wants to send the message m to Alice and Bob, so he encrypts to get $c_A \equiv m^{e_A}$ and $c_B \equiv m^{e_B}$ (mod n). Show how Eve can find m if she intercepts c_A and c_B.

   Sol:

   $\gcd(e_A, e_B) = 1$ implies that $\exists a, b$ such that $a \cdot e_A + b \cdot e_B = 1$
   (you can always find out a, b using the extended Euclidean algorithm)
   Since $c_A \equiv m^{e_A}$ and $c_B \equiv m^{e_B}$ (mod n), Eve can calculate the following

   $c_A^{a} + c_B^{b} \equiv m^{a \cdot e_A + b \cdot e_B} \equiv m^{1} \equiv m \pmod{n}$ and retrieve m easily.

   Despite the fact that Alice and Bob do not hold their individual secret during this scheme, this is a type of common modulus attack for RSA when a message is transferred twice using different encryption exponents.

2. (Trappe page 160: 11) Suppose Alice uses the RSA method as follows. She starts with a message consisting of several letters, and assigns a=1, b=2, ..., z=26. She then encrypts each letter separately. For example, if her message is cat, she calculates 3^e (mod n), 1^e (mod n), and 20^e (mod n). Then she sends the encrypted message to Bob. Explain how Eve can find the message without factoring n. In particular, suppose n=8881 and e =13. Eve intercepts the message

   4461 794 2015 2015 3603

   Find the message without factoring 8881

   Sol:

   It is very easy to find out a^e (mod n) where e = 13, n=8881, and $a \in \{1, 2,..., 26\}$. They are tabulated as follows:

a	1	2	3	4	5	6	7	8	9	10	11	12	13
a^e (mod n)	1	8192	4624	4028	794	2343	231	4461	4809	3556	476	2015	513
a	14	15	16	17	18	19	20	21	22	23	24	25	26
a^e (mod n)	699	3603	8078	2825	8093	2547	1072	2424	633	413	5982	8766	1783

   Therefore, the corresponding plaintext can be obtained through a simple table lookup. The plaintext is “hello”.

3. (Trappe page 175: 3)

   (a) Let α be a primitive root mod p. Show that

$$L_\alpha(\beta_1 \beta_2) \equiv L_\alpha(\beta_1) + L_\alpha(\beta_2) \pmod{p-1}$$

(Hint : You need the proposition in Section 3.7.)

Sol:

Because α is a primitive,

$\exists$ unique $L_\alpha(\beta_1)$ such that $\beta_1 \equiv \alpha^{L_\alpha(\beta_1)} \pmod{p}$, also … (1)

$\exists$ unique $L_\alpha(\beta_2)$ such that $\beta_2 \equiv \alpha^{L_\alpha(\beta_2)} \pmod{p}$ and … (2)

$\exists$ unique $L_\alpha(\beta_1\beta_2)$ such that $\beta_1\beta_2 \equiv \alpha^{L_\alpha(\beta_1\beta_2)} \pmod{p}$ … (3)

multiply both sides of equation (1) and (2), we get

$\beta_1\beta_2 \equiv \alpha^{L_\alpha(\beta_1)} \alpha^{L_\alpha(\beta_2)} \pmod{p}$ … (4)

equate (3) and (4) we get

$$\beta_1\beta_2 \equiv \alpha^{L_\alpha(\beta_1\beta_2)} \equiv \alpha^{L_\alpha(\beta_1)} \alpha^{L_\alpha(\beta_2)} \pmod{p}$$

from the proposition in section 3.7, we get the following

$$L_\alpha(\beta_1 \beta_2) \equiv L_\alpha(\beta_1) + L_\alpha(\beta_2) \pmod{p-1}$$

(b) More generally, let α be arbitrary. Show that

$$L_\alpha(\beta_1 \beta_2) \equiv L_\alpha(\beta_1) + L_\alpha(\beta_2) \quad (\text{ mod } \mathrm{ord}_p(\alpha)),$$

where $\mathrm{ord}_p(\alpha)$ is defined in Exercise 3.9.

Sol:

First of all, α cannot be really arbitrary.

α must be chosen such that

$\exists$ unique $L_\alpha(\beta_1)$ such that $\beta_1 \equiv \alpha^{L_\alpha(\beta_1)} \pmod{p}$, … (5)

$\exists$ unique $L_\alpha(\beta_2)$ such that $\beta_2 \equiv \alpha^{L_\alpha(\beta_2)} \pmod{p}$, and … (6)

$\exists$ unique $L_\alpha(\beta_1\beta_2)$ such that $\beta_1\beta_2 \equiv \alpha^{L_\alpha(\beta_1\beta_2)} \pmod{p}$ … (7)

where $L_\alpha(\beta_1)$, $L_\alpha(\beta_2)$, and $L_\alpha(\beta_1\beta_2)$ are defined within 1 and $\mathrm{ord}_p(\alpha)$

multiply both sides of equation (5) and (6), we get

$\beta_1\beta_2 \equiv \alpha^{L_\alpha(\beta_1)} \alpha^{L_\alpha(\beta_2)} \pmod{p}$ … (8)

equate (7) and (8) we get

$$\beta_1\beta_2 \equiv \alpha^{L_\alpha(\beta_1\beta_2)} \equiv \alpha^{L_\alpha(\beta_1)} \alpha^{L_\alpha(\beta_2)} \pmod{p}$$

also from the definition of $\mathrm{ord}_p(\alpha)$ we know $\alpha^{\mathrm{ord}_p(\alpha)} \equiv 1 \pmod{p}$, we get

$$\beta_1\beta_2 \equiv \alpha^{L_\alpha(\beta_1\beta_2) \bmod \mathrm{ord}_p(\alpha)} \equiv \alpha^{L_\alpha(\beta_1) \bmod \mathrm{ord}_p(\alpha)} \alpha^{L_\alpha(\beta_2) \bmod \mathrm{ord}_p(\alpha)} \pmod{p}$$

from the proposition in section 3.7, we get the following

$$L_\alpha(\beta_1 \beta_2) \equiv L_\alpha(\beta_1) + L_\alpha(\beta_2) \quad (\text{mod } \mathrm{ord}_p(\alpha)) \pmod{p-1}$$

because $\mathrm{ord}_p(\alpha) \mid p-1$, the above equation is equivalent to

$$L_\alpha(\beta_1 \beta_2) \equiv L_\alpha(\beta_1) + L_\alpha(\beta_2) \quad (\text{mod } \mathrm{ord}_p(\alpha))$$

4. (Trappe page 175: 4)

(a) Suppose you have a random 500-digit prime p. Suppose some people want to store passwords, written as numbers. If x is the password, then the number $2^x \pmod{p}$ is stored in a file. When y is given as a password, the number 2^y (mod p) is compared with the entry for the user in the file. Suppose someone gains access to the file. Why is it hard to deduce the passwords?

Sol:

If the $\mathrm{ord}_p(2)$ is large (preferably being p-1), then given 2^x (mod p), it will be difficult to figure out the complete value x because it is an instance of the discrete log problem with 2 as the base and p a 500-digit prime number. Furthermore, if one can solve x given 2^x (mod p), then he can solve z given α^z (mod p) by calculating $\mathrm{dlog}_2(\alpha^z) \cdot \mathrm{dlog}_2(\alpha)^{-1}$.

(b) Suppose p is instead chosen to be a five-digit prime. Why would the system in part (a) not be secure?

Sol:

Solving x from 2^x (mod p) is easy if p is a five-digit prime. One can just tabulate all possible (x, 2^x (mod p)) pairs and match the second terms to find the corresponding x.