Institut for Matematik og Datalogi
Syddansk Universitet, Odense

8. december 2022
RF/JFB

Opgaver DM573 uge 50

Husk at læse de relevante sider i slides og noter før du/I forsøger at løse en opgave.

I: Løses i løbet af øvelsestimerne i uge 50

1. Suppose in RSA the public key is $PK = (1517, 13)$. Which of the following is the RSA encryption of the message 43?

   (a) $1517^{43} \pmod{13}$
   (b) $43^{13} \pmod{1517}$
   (c) $13^{43} \pmod{1517}$

   For the correct answer among those above, we want to use the algorithm for fast modular exponentiation (page 29 on the slides) to calculate the encrypted message. How many times during the recursive execution is the "**if** k is odd" case encountered, and how many times is the "**if** k is even" case encountered? [Do not include the base cases $k = 0$ and $k = 1$ in the counts.]

2. Is one of the following the multiplicative inverse of 49 modulo 221?

   12, 56, 121, 212

3. Which of the following sets of public key (PK) and secret key (SK) is a valid set of RSA keys (ignoring the fact that the numbers are not large enough for security)?

   (a) $PK = (91, 37)$; $SK = (91, 23)$
   (b) $PK = (143, 77)$; $SK = (143, 53)$

(c) $PK = (231, 59)$; $SK = (231, 47)$

(d) $PK = (107, 25)$; $SK = (107, 30)$

4. In the Sieve of Eratosthenes (page 53 on the slides), how many lists (including the current) have been created at the point where the number 13 is the first element in a list?

5. Consider an RSA system with Alice's public key $N = 1517$ and $e = 17$. Note that $1517 = 37 \cdot 41$.

   (a) Find Alice's secret key d. Use the Extended Euclidean Algorithm from pages 46–47 of the slides from the lecture. [Note: On page 46, e and d are called a and s. On top of page 47, d also is called x. On page 47, m is not the message (as earlier in the slides), but is the number $(p-1)(q-1)$. See also the note *Algoritmiske aspekter af RSA* for an exposition with less variation in the notation.]

   (b) Try encrypting 423. Use the algorithm for fast modular exponentiation (page 29 on the slides). How many times during the recursive execution is the "**if** k is odd" case encountered, and how many times is the "**if** k is even" case encountered? [Do not include the base cases $k = 0$ and $k = 1$ in the counts.]

   (c) Decrypt the number obtained above, using fast modular exponentiation. Is the result correct? How many times during the recursive execution is the "**if** k is odd" case encountered, and how many times is the "**if** k is even" case encountered? [Do not include the base cases $k = 0$ and $k = 1$ in the counts.]

6. Why is a cryptographically secure hash function used in connection with RSA digital signatures?

7. In RSA, why must the message being encrypted be a non-negative integer strictly less than the modulus?

8. We consider the RSA system and use the notation and algorithms from the slides. Let $N_A = 1517$ and $e_A = 13$.

   - Encrypt the message $m = 43$. For the modular exponentiation, use the algorithm from the slides, page 29. Show all steps in your computation.
   - If you had created the keys yourself, you would know that $p = 37$ and $q = 41$. From this information and $e_A = 13$, find the secret

key d_A. Use the Extended Euclidean Algorithm from pages 46–47 of the RSA slides used in lectures. Show all steps in your computation. [Note: On page 46, e and d are called a and s. On top of page 47, d also is called x. On page 47, m is not the message (as earlier in the slides), but is the number $(p-1)(q-1)$. See also the note *Algoritmiske aspekter af RSA* for an exposition with less variation in the notation.]

9. Find four different square roots of 1 modulo 143, i.e., numbers which multiplied by themselves modulo 143 give 1 (and which are at least 0 and less than 143). You may consider writing a simple program for finding them.

10. Try executing the Miller-Rabin primality test on 11, 15, and 561. First, what type of numbers are they (note: 561 is known from the slides)? For each, run the Miller-Rabin test for at least one value a of your own choice, preferably for more. Use e.g. a simple Java program importing the class `BigInteger` from the Java library for executing the exponentiations (first raising to a power and then using modulus should work in reasonable time for numbers this size, hence there is no need to use the fast modular exponentiation algorithm in this exercise). Which calculations showed that the composite numbers were not prime—the first line (the Fermat test) or later lines?

    With 561, be sure to try an a relatively prime to 561 (most are, 2 is a simple example). What happens differently if you try $a = 3$? Can you explain the latter?

II: Løses hjemme

Ingen (der er ikke flere øvelsestimer i kurset).