DM534 — Øvelser Uge ??

Introduktion til Datalogi, Efterår 2021

Jonas Vistrup og Rolf Fagerberg

---

1 I

1.1

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

(a) $1517^{43} \bmod 13$

(b) $43^{13} \bmod 1517$

(c) $13^{43} \bmod 1517$

SVAR: b

1.2

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

$$12, 56, 121, 212$$

SVAR: 212 is the multiplicative inverse of 49 modulo 221.

1.3

Which of the following is a valid RSA key (ignoring the fact that the numbers are not large enough for security)?

(a) $\text{PK} = (91, 37); \text{SK} = (91, 23)$.

SVAR: Invalid, since $N = 91 = 13 \cdot 7$, but $37 \cdot 23 \mod (12 \cdot 6) = 59 \neq 1$, so e and d do not fulfill the multiplicative inverse criterion.

(b) $\text{PK} = (143, 77); \text{SK} = (143, 53)$.

SVAR: Valid.

(c) $\text{PK} = (231, 59); \text{SK} = (231, 47)$.

SVAR: Invalid, since $231 = 3 \cdot 7 \cdot 11$, so N is not a product of two primes.

(d) $\text{PK} = (107, 25); \text{SK} = (107, 30)$.

SVAR: Invalid, since 107 is prime, so N is not a product of two primes.

1.4

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

SVAR: 6, since 13 is the sixth prime.

1.5

We consider the RSA system and use the notation and algorithms from the slides. Let $N = 1517$ and $e = 13$.

(a) Encrypt the message $m = 43$. For the modular exponentiation, use the algorithm from the slides. Show all steps of your calculation. 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.]

(b) If you had created the keys yourself, you would know that $p = 37$ and $q = 41$. From this information and $e = 13$, find the secret key d. Use the Extended Euclidean Algorithm from the slides. Show all steps in your computation.

SVAR a:

During the recursion in the fast modular exponentiation algorithm, the value of k changes as follows:

$$13 \to 12 \to 6 \to 3 \to 2 \to 1$$

So there are two odd cases and three even cases (since we do not count the last case of $k = 1$).

Here are the full details of the recursion:

$$\begin{aligned}
43^{13} \mod 1517 &= 43 \cdot z_1 \mod 1517 \\
z_1 = 43^{12} \mod 1517 &= c_1 \cdot c_1 \mod 1517 \\
c_1 = 43^{6} \mod 1517 &= c_2 \cdot c_2 \mod 1517 \\
c_2 = 43^{3} \mod 1517 &= 43 \cdot z_2 \mod 1517 \\
z_2 = 43^{2} \mod 1517 &= c_3 \cdot c_3 \mod 1517 \\
c_3 = 43 \mod 1517 &= 43 \\
z_2 = 43 \cdot 43 \mod 1517 &= 332 \\
c_2 = 43 \cdot 332 \mod 1517 &= 623 \\
c_1 = 623 \cdot 623 \mod 1517 &= 1294 \\
z_1 = 1294 \cdot 1294 \mod 1517 &= 1185 \\
43^{13} \mod 1517 &= 43 \cdot 1185 \mod 1517 = 894
\end{aligned}$$

SVAR b:

In the Extended Euclidean Algorithm from the slides we have $a\ (= e) = 13$ and $b\ (= N' = (p-1)(q-1)) = (37-1)(41-1) = 36 * 40 = 1440$. We are to find $s\ (= d)$.

Following the Extended Euclidean Algorithm, we get the following:

$$
\begin{aligned}
d_0 &= 1440 \\
d_1 &= 13 \quad (1440 = 110 \cdot 13 + 10) \\
d_2 &= 10 \quad (13 = 1 \cdot 10 + 3) \\
d_3 &= 3 \quad (10 = 3 \cdot 3 + 1) \\
d_4 &= 1 \quad (3 \cdot 1 + 0) \\
d_5 &= 0 \quad \text{STOP, return } d_4 \text{ (that is, return 1)}
\end{aligned}
$$

Now work backwards:

$$
\begin{aligned}
1 &= 10 - 3 \cdot 3 \\
&= 10 - 3 \cdot (13 - 10) = -3 \cdot 13 + 4 \cdot 10 \\
&= -3 \cdot 13 + 4 \cdot (1440 - 110 \cdot 13) = 4 \cdot 1440 - 443 \cdot 13
\end{aligned}
$$

So the algorithm returns $s = -443$ and $t = 4$. For these s and t we have $sa + tb = \gcd(a, b) = 1$.

In other words, $-443 \cdot 13 + 4 \cdot 1440 = 1$. From this follows $-443 \cdot 13 \bmod 1440 = -443 \cdot 13 + 4 \cdot 1440 \bmod 1440 = 1$, where the first equality holds because adding a multiple of 1440 does not change the remainder modulo 1440. So -443 is a multiplicative inverse to 13 (modulo 1440) and is our first candidate for s.

However, in RSA we require d to be positive, hence we change d by adding 1440, making -443 become $-443+1440 = 997$. We have $997 \cdot 13 \bmod 1440 = (-443+1440) \cdot 13 \bmod 1440 = -443 \cdot 13 + 1440 \cdot 13 \bmod 1440 = -443 \cdot 13 \bmod 1440 = 1$, where the next to last equality holds because adding a multiple of 1440 does not change the remainder modulo 1440. In other words, the value 997 is also a multiplicative inverse to 13 (modulo 1440).

Hence, we choose $d = 997$, as this is positive and fulfills the criterion $d \cdot e \bmod N' = 1$ from RSA.

1.6

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

SVAR: Hash function: for size reduction of the message (the document) to be encrypted, since we in a single RSA encryption only can encode messages up to a certain size in bits (namely up to at most $\log_2(N)$ bits in length, as the message must be interpretable as an integer less than N). Cryptographically secure: so it's improbable that somebody is able to change the message without changing the hash (that is, it should be improbably to find another message with the same hash as the original message).

1.7

With RSA, why would you never use the value 2 as one of of the two primes p and q?

SVAR: Because it would result in a even N, making everyone see that you used 2 as one of the primes. From there the second prime is easily found by dividing N by 2.

1.8

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

SVAR: RSA is working with modular arithmetic, hence decryption always returns a number between 0 and the modulus minus one (i.e., $N - 1$). Thus, if larger numbers (or negative numbers) were encrypted, then decryption cannot lead to the orignal message.

1.9

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. You should use the Extended Euclidean Algorithm.

(b) Try encrypting 423. Use the algorithm for fast modular exponentiation. 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.]

SVAR a: $d = 593$. [For a similar example with all steps of the calculation shown, see question 1.5.]

SVAR b:

The encrypted value is $423^e \bmod N = 423^{17} \bmod 1517 = 562$.

During the recursion in the fast modular exponentiation algorithm, the value of k changes as follows:

$$17 \to 16 \to 8 \to 4 \to 2 \to 1$$

So the answer is: 1 odd, 4 even (since we do not count the last case of $k = 1$).

Here are the full details of the recursion:

$$
\begin{aligned}
423^{17} \mod 1517 &= 423 \cdot z \mod 1517 \\
z &= 423^{16} \mod 1517 = c_1 \cdot c_1 \mod 1517 \\
c_1 &= 423^{8} \mod 1517 = c_2 \cdot c_2 \mod 1517 \\
c_2 &= 423^{4} \mod 1517 = c_3 \cdot c_3 \mod 1517 \\
c_3 &= 423^{2} \mod 1517 = c_4 \cdot c_4 \mod 1517 \\
c_4 &= 423^{1} \mod 1517 = 423 \\
c_3 &= c_4 \cdot c_4 \mod 1517 = 423 \cdot 423 \mod 1517 = 1440 \\
c_2 &= c_3 \cdot c_3 \mod 1517 = 1440 \cdot 1440 \mod 1517 = 1378 \\
c_1 &= c_2 \cdot c_2 \mod 1517 = 1378 \cdot 1378 \mod 1517 = 1117 \\
z &= c_1 \cdot c_1 \mod 1517 = 1117 \cdot 1117 \mod 1517 = 715 \\
423^{17} \mod 1517 &= 423 \cdot z \mod 1517 = 423 \cdot 715 \mod 1517 = 562
\end{aligned}
$$

SVAR c:

The decrypted value is $562^d \bmod N = 562^{593} \bmod 1517 = 423$.

So the result is correct (we get the original message back). During the recursion in the fast modular exponentiation algorithm, the value of k changes as follows:

$$593 \to 592 \to 296 \to 148 \to 74 \to 37 \to 36 \to 18 \to 9 \to 8 \to 4 \to 2 \to 1$$

So the answer is: 3 odd, 9 even (since we do not count the last case of $k = 1$).

2 II

2.1

Why in RSA is it necessary that $\gcd(e, (p-1)(q-1)) = 1$? Find an example (that is, find values e, p and q) where this greatest common divisor is not equal to 1.

SVAR:

This condition is used in the proof of correctness of RSA (see slides).

To see that the condition is actually necessary for RSA to work, consider the case $e = 9$ and $(p, q) = (13, 5)$.

Here, $\gcd(e, (p-1)(q-1)) = \gcd(9, 12 \cdot 4) = 3$. In this case, we can see that d does not exist: With $N' = (p-1)(q-1)$, assume that we have d and k such that $ed = 1 + kN'$. Then, as $e = 9 = 3 \cdot 3$ and $N' = 48 = 3 \cdot 16$, we would have $ed = (3 \cdot 3)d = 1 + kN' = 1 + k(3 \cdot 16)$, i.e., $1 = 3 \cdot t$ for an integer t, which is a contradiction.

We also observe that in this case, both 2 and 32 will be encrypted to 57, as ($2^9 \bmod 65 = 32^9 \bmod 65 = 57$). But decoding can at most return one of these value. So RSA does not work.

2.2

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.

SVAR: The following numbers are square roots of 1 modulo 143: 1, 12, 131 og 142.

2.3

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, using calculations in Python (raising to a power and first 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 used in the slides, 4, 5, 7, 10, and 13 are other examples). What happens differently if you try $a = 3$? Can you explain the latter?

SVAR:

We have a prime number (11), for which the algorithm never fails (fails here means returning "composite") for any a, a composite number (15), which is discovered already in the first line (the Fermat test) for a lot of a's, and a Carmichael number (561), which is composite, but not for any a (except potentially those with $\gcd(a, n) \neq 1$) is discovered in the first line of the Miller-Rabin test (the Fermat test). However, considering the entire Miller-Rabin test, Carmichael numbers (and all other composite numbers) will still be discovered for at least 3/4 of the possible values of a.

SVAR:

As $n = 561 = 3 \cdot 11 \cdot 17$, we have $\gcd(a, n) = 3 \neq 1$ for $a = 3$. Here the first line (the Fermat test) actually will fail (discover that n is composite), even if this n is a Carmichael number. This shows that the phrase "$\gcd(a, n) \neq 1$" cannot be removed from the definition of Carmichael numbers (if we in that definition want to say that they fail the Fermat test for almost all a's). That the first line (the Fermat test) in the above case will fail can be seen from the fact that if $a^{n-1} \bmod n = 3^{560} \bmod 561$ should be equal to one, we would have $3 \cdot 3^{559} = 3 \cdot 11 \cdot 17 \cdot k + 1$, that is, $3 \cdot t = 1$, which is impossible since k and t are integers.