Exercise sheet 5, 14 December 2017

For this exercise sheet you should not use your computer for more functions than a pocket calculator offers you (though with more digits) – unless explicitly stated. In particular the last exercises have number too big for your calculator. There are many exercises with standard algebra and number theory content. Skip these if you are confident in how to solve them – or keep them as a self test if you still need to learn the material.

1. Compute $11^9 \bmod 35$ in two different ways: First compute 11^9 and then reduce modulo 35 and then compute it reducing modulo 35 whenever useful. Observe the time the computation takes you.

   For the exponentiaton with reduction you should use the *square-and-multiply method*.

2. State all elements in $(\mathbb{Z}/12)^\times$.

3. State all elements in $(\mathbb{Z}/21)^\times$.

4. Execute the RSA key generation where $p = 239, q = 433$, and $e = 23441$.

5. RSA-encrypt the message 23 to a user with public key $(e, n) = (17, 11584115749)$. Document how you compute the exponentiation.

6. Bob uses public key $(n, e) = (443507, 11)$ and secret key $d = 241187$. He receives ciphertext $c = 64649$. Decrypt the ciphertext.

7. Use the Chinese Remainder Theorem (see below) to find the smallest positive integer x satisfying the following system of congruences, should such a solution exist.

$$\begin{aligned} x &\equiv 0 \bmod 3 \\ x &\equiv 1 \bmod 5 \\ x &\equiv 2 \bmod 8 \end{aligned}$$

8. Users A, B, and C are friends of S. They have public keys $(e_A, n_A) = (3, 58483), (e_B, n_B) = (3, 50629)$, and $(e_C, n_C) = (3, 54253)$. You know that S sends the same message to all of them and you observe the ciphertexts $c_A = 52106, c_B = 7516$, and $c_C = 4649$. What was the message?

9. The m-RSA system is a multiplicative variant of RSA. To generate her keys, Alice picks integers $a, b \in \mathbb{Z}$, computes $M = a \cdot b - 1$, picks two more random integers $a', b' \in \mathbb{Z}$, and computes $e = a' \cdot M + a$, $d = b' \cdot M + b$, and $n = (e \cdot d - 1)/M$. Her public key is (n, e) and her secret key is (n, d).

   If Bob wants to encrypt a message m to Alice, he looks up her public key (n, e) and computes $c \equiv e \cdot m \bmod n$.

   Alice decrypts ciphertext c by computing $m' \equiv c \cdot d \bmod n$.

(a) Show that n is an integer.

(b) Show that $m' \equiv m \bmod n$.

(c) Break the system.

10. Read about the ROBOT attack `https://robotattack.org/`.

Reminder on how the Chinese Remainder Theorem works:

Theorem 1 (Chinese Remainder Theorem)
Let $r_1, \ldots, r_k \in \mathbb{Z}$ and let $0 \neq n_1, \cdots, n_k \in \mathbb{N}$ such that the n_i are pairwise coprime. The system of equivalences

$$\begin{aligned} X &\equiv r_1 \bmod n_1, \\ X &\equiv r_2 \bmod n_2, \\ &\vdots \\ X &\equiv r_k \bmod n_k, \end{aligned}$$

has a solution X which is unique up to multiples of $N = n_1 \cdot n_2 \cdots n_k$. The set of all solutions is given by $\{X + aN | a \in \mathbb{Z}\} = X + N\mathbb{Z}$.

If the n_i are not all coprime the system might not have a solution at all. E.g. the system $X \equiv 1 \bmod 8$ and $X \equiv 2 \bmod 6$ does not have a solution since the first congruence implies that X is odd while the second one implies that X is even. If the system has a solution then it is unique only modulo $\operatorname{lcm}(n_1, n_2, \ldots, n_k)$. E.g. the system $X \equiv 4 \bmod 8$ and $X \equiv 2 \bmod 6$ has solutions and the solutions are unique modulo 24. Replace $X \equiv 2 \bmod 6$ by $X \equiv 2 \bmod 3$; the system still carries the same information but has coprime moduli and we obtain $X = 8a + 4 \equiv 2a + 1 \overset{!}{\equiv} 2 \bmod 3$, thus $a \equiv 2 \bmod 3$ and $X = 8(3b + 2) + 4 = 24b + 20$. The smallest positive solution is thus 20.

We now present a constructive algorithm to find this solution, making heavy use of the extended Euclidean algorithm presented in the previous section. Let $N_i = N/n_i$. Since all n_i are coprime, we have $\gcd(n_i, N_i) = 1$ and we can compute u_i and v_i with

$$u_i n_i + v_i N_i = 1.$$

Let $e_i = v_i N_i$, then this equation becomes $u_i n_i + e_i = 1$ or $e_i \equiv 1 \bmod n_i$. Furthermore, since all $n_j | N_i$ for $j \neq i$ we also have $e_i = v_i N_i \equiv 0 \bmod n_j$ for $j \neq i$.

Using these values e_i a solution to the system of equivalences is given by

$$X \equiv \sum_{i=1}^{k} r_i e_i \bmod N,$$

since X satisfies $X \equiv r_i \bmod n_i$ for each $1 \leq i \leq k$.

Example 2 *Consider the system of integer equivalences*

$$\begin{aligned} X &\equiv 1 \bmod 3, \\ X &\equiv 2 \bmod 5, \\ X &\equiv 5 \bmod 7. \end{aligned}$$

The moduli are coprime and we have $N = 105$. *For* $n_1 = 3, N_1 = 35$ *we get* $v_1 = 2$ *by just observing that* $2 \cdot 35 = 70 \equiv 1 \bmod 3$. *So* $e_1 = 70$. *Next we compute* $N_2 = 21$ *and see* $v_2 = 1$ *since* $21 \equiv 1 \bmod 5$. *This gives* $e_2 = 21$. *Finally,* $N_3 = 15$ *and* $v_3 = 1$ *so that* $e_3 = 15$.
The result is $X = 70+2\cdot 21+5\cdot 15 = 187$ *which indeed satisfies all* 3 *congruences. To obtain the smallest positive result we reduce* 187 *modulo* N *to obtain* 82.

For easier reference we phrase this approach as an algorithm.

Algorithm 3 (Chinese remainder computation)
IN: *system of* k *equivalences as* $(r_1, n_1), (r_2, n_2), \ldots (r_k, n_k)$ *with pairwise coprime* n_i
OUT: *smallest positive solution to system*

1. $N \leftarrow \prod_{i=1}^{k} n_i$
2. $X \leftarrow 0$
3. `for` $i = 1$ `to` k
   (a) $M \leftarrow N \operatorname{div} n_i$
   (b) $v \leftarrow (M^{-1} \bmod n_i)$ *(use XGCD)*
   (c) $e \leftarrow vM$
   (d) $X \leftarrow X + r_i e$
4. $X \leftarrow X \bmod N$