
Homework 3 – Due Thursday, April 2

Reminders: Collaboration is permitted on homeworks, but you must (a) list your collaborators on the problem set and (b) **understand and write your solutions yourself**.

Clarity is an important component of the grade on your problem sets. Poorly explained solutions will not receive full marks, even if they are correct.

Problems to be handed in:

1. (**Squares**) We say b is a square root of a modulo N if $b^2 \equiv a \pmod{N}$. The set of elements with square roots modulo p is called the set of *quadratic residues* mod N , denoted QR_N .
 - (a) Suppose that p is prime. Show that there is an integer $f(p)$ such that $a \in \mathbb{Z}_p^*$ has a square root modulo p if and only if $a^{f(p)} \equiv 1 \pmod{p}$. (Hint: Use the isomorphism between $\langle \mathbb{Z}_p^*, \times \rangle$ and $\langle \mathbb{Z}_{p-1}, + \rangle$.)
 - (b) Show that if p is prime, every quadratic residue has exactly 2 square roots modulo p .
 - (c) Suppose that p is prime and $p \equiv 3 \pmod{4}$. Give a polynomial time algorithm for finding a square root of a , assuming that one exists. (Hint: there is a square root of the form $a^{g(p)}$ for some function g .)
 - (d) (Finding higher order roots) KL, exercise 7.18. Note that the question is specifically about powers x that are relatively prime to $p - 1$.
2. (**Miscellaneous:**)
 - (a) (Computing $\phi(N)$ is as hard as factoring) KL, Exercise 7.13.
 - (b) (For $e = 3$, computing d is as hard as factoring) KL, Exercise 7.14.
 - (c) In the book, the “Common modulus attack I and II” attacks (page 361) warn of the dangers of using one modulus N for a group of people and issuing a different key pair (e_i, d_i) for each person. In particular, the attack we saw in class works if any two public exponents are relatively prime, i.e. $\gcd(e_i, e_j) = 1$.

Suppose that a company decides to avoid this easy attack by ensuring that every pair e_i, e_j shares a non-trivial factor, but so there is still no non-trivial factor common to all of the exponents e_i . Describe an efficient algorithm that takes the values N, e_1, e_2, \dots, e_k and ciphertexts $[m^{e_1} \pmod{N}], \dots, [m^{e_k} \pmod{N}]$ and returns m . Here k is the total number of members of the group.
 - (d) Consider the case above, but where there is a non-trivial factor $f > 1$ that is common to all the exponents. Show that given any one of the secret keys d_i (in addition to N, e_1, e_2, \dots, e_k and ciphertexts $[m^{e_1} \pmod{N}], \dots, [m^{e_k} \pmod{N}]$), it is possible to compute m . Give a complete description of your algorithm and prove its correctness. You may use Euclid’s algorithm as a black box, but you should not use the fact that one can factor N from (e_i, d_i) without giving a proof.

3. (Chinese Remainder Theorem)

- (a) (Completing the CRT proof) KL, Exercise 7.9
- (b) Suppose N has k distinct *odd* prime factors. Prove that every quadratic residue x in \mathbb{Z}_N^* has at least 2^k distinct square roots.
- (c) ~~(Plain RSA is weak for $e = 3$): Let N_1, N_2, N_3 be integers such that e is relatively prime to $\phi(N_1), \phi(N_2)$ and $\phi(N_3)$. Let x be an integer smaller than N_1, N_2, N_3 . Give a polynomial-time algorithm that computes x given N_1, N_2, N_3 and $[x^3 \bmod N_1], [x^3 \bmod N_2], [x^3 \bmod N_3]$.~~
- (d) ~~Does a similar attack work for $e = 5$? What needs to change?~~

4. (**Extra credit: coding**) Suppose you wish to send a message $m \in \{0, 1\}^k$ over a network. The problem is that the network is lossy, and some packets will get dropped. You would like to encode your message into n packets of size at most t bits each, such that the message m can be recovered in polynomial time from any subset of $s = \lceil \frac{k}{t-1} \rceil$ or more packets. Because the packets are relatively large, it is ok to assume that n is less than the number of primes between 2^{t-1} and $2^t - 1$.

For concreteness, suppose packets have size $t = 1025$ bits. Then you could encode a message of length $k = 2^{23}$ bits (1 MB) into $n = 2^{14} \approx 16,000$ packets so that any half of the packets suffices to recover the message.

Give an algorithm to do this based on the Chinese Remainder Theorem. Both the encoding and decoding should be polynomial time. You may assume that the decoder knows all the details of the encoding ahead of time (but not the message, of course).