CSCI 0220 Discrete Structures and Probability C. Klivans

Challenge Problem 1

Any Thursday

Occasionally, we will write a problem that is too challenging or involved for a regular homework, but is still extremely interesting and relevant to the course. We have decided to release them as challenge problems, which you may do (or not do) at your leisure. If you feel particularly confident in your understanding, you may turn in your answers to a challenge problem at any time. They will be picked up along with the current homework assignment at 2 pm the following Thursday.

The collaboration policy still applies for challenge problems: you may discuss them with other students, but please do not write up solutions together if you are going to turn them in. Handins which we deem to be exemplary will earn an additional free late pass on any *following* homework. Challenge problems otherwise do not affect your grade whatsoever, including not turning one in.

You don't need to attach a cover sheet to a challenge problem solution. TA Hours are for homework assignments only; if you wish to discuss the challenge problem with a TA, go to clinic.

The Fourth-Grader Problem

We've shown you how RSA can be used to encrypt things, but what other things can you do with it? In this problem, we are going to give an example of an area of cryptography called *secure multi-party computation.*

Consider a class of n fourth graders in a classroom. Each student has a single crush (which is not his or her self), and the entire class has decided to find out if their crushes are requited. More formally, if we call the set of students $S = \{1, 2, \ldots, n\}$, we define the crush relation c to be some irreflexive function from $S \to S$. Student i wants to determine if $(c(i), i) \in c$.

But crushes are serious business in fourth grade! The algorithm must ensure security – that is, no student can find out any other student's crush. We will relax this restriction if two students like each other; then, both should learn the other's crush – themselves!

The teacher doesn't want a bunch of fourth graders dating each other, and has thus told them they are not allowed to talk. The teacher can't see everything, though, so they are able to communicate by passing a single piece of paper around the class.

To start, every student i chooses two RSA key pairs, which we will call the exposed key and the hidden key. Note that we are using *hidden* and *exposed* to differentiate them from the words "public" and "private"; each key (hidden and exposed) is made up of both a public key and a private key.

(a) Declare the variables each student starts with (after generating their two RSA key pairs).

(b) In the first round of communication (i.e. the first time the class passes the paper around), every student writes their (unique) name, and their exposed public key on the piece of paper. State the variables which student i has just released.

(c) Every student is now able to send any *other* student a message which only the recipient can read. Each student wants their crush to receive their hidden public key. Completely describe the next step of the algorithm.

(d) Assume Alice (i) and Bob (j) like each other. What does Alice do to get Bob's hidden public key?

(e) Assume Alice likes Bob, but Bob does not like Alice. What happens when Alice tries to get Bob's hidden public key?

Each student chooses some long, recognizable message, like what Alice chose:

> Twas bryllyg, and ye slythy toves
> Did gyre and gymble in ye wabe:
> All mimsy were ye borogoves;
> And ye mome raths outgrabe.

Each student i announces their chosen message, x_i, to the group in plain text (i.e. without encrypting it). No two students chooses the same x (which, since they announce one at a time, is easy to enforce).

(f) Alice likes Bob. What should Alice post so that Bob will be able to read her message *if and only if* Bob likes Alice back? Complete the algorithm, and briefly justify its security.

(g) (Extra Challenge:) Identify why RSA is not an ideal encryption scheme for this algorithm, and what property an ideal encryption scheme would have. (*Hint*: think about what happens when you decrypt the hidden public key.)

If you found this problem interesting, consider taking CS151 (Cryptography) in the spring!