Problem Set 3

This problem set is due on *Thursday, October 4, 2001* at the beginning of class. Late homeworks will *not* be accepted. You are to work on this problem set in groups of three or four people.

Mark the top of each sheet with your names, 6.857, the problem set number and question, and the date. **Type up your solutions** and be clear. Each problem should begin on a new sheet of paper. That is, you will turn in each problem on a separate pile of paper.

Problem 3-1. Inverses

In this problem, we want to develop a practical understanding of the best way to compute modular inverses. In general, there are two ways to compute $a^{-1} \bmod p$ where p is a prime and a is a random element of Z_p:

1. Use the equation $a^{-1} \bmod p = a^{p-2} \bmod p$.
2. Use Euclid's extended algorithm (see CLRS chapter 31).

Program both of these approaches and give the average times found for many such random a's. (You can let the p be the same.) Which is better?

Summarize your results and submit your code.

Problem 3-2. El Gamal

Explain how to generalize the El Gamal signature scheme to work over 2×2 invertible matrices modulo p, where p is prime. (All elements we work with are then represented by such 2×2 invertible matrices instead of by individual elements modulo p.) The group of such matrices is typically denoted $\mathrm{GL}(2, p)$.

Hint: Let $f(p)$ denote the number of invertible 2×2 matrices modulo p. Argue that $f(p) = (p^2 - 1)(p^2 - p)$. Note that all elements have an order that divides $f(p)$ since $f(p)$ is the size of the group. But the group may not have a generator, so you might need an element of a large order instead.

Problem 3-3. Blind signatures

Consider the following well-known signature scheme presented below. (Assume that $H()$ is a collision-resistant hash function.)

Key Generation	**Sign**(m)
Find two primes, p, q such that $q \mid p-1$.	Pick $t, u \in Z_q$
Find elements $g, h \in Z_p^*$ of order q.	Let $a = g^t h^u \bmod p$
(p, q, g, h) are global parameters	Let $c = H(m, a)$
Pick $r, s \in Z_q$	Let $R = t + cr \bmod q$
Public key is $(y = g^{-r} h^{-s})$	Let $S = u + cs \bmod q$
Secret key is (r, s)	Output (a, R, S)

In order to verify this signature, first compute $c = H(m, a)$ and then check whether $a = g^R h^S y^c \bmod p$. This works because

$$\begin{aligned} g^R h^S y^c &= g^{t+cr} h^{u+cs} y^c \\ &= g^{t+cr} h^{u+cs} (g^{-r} h^{-s})^c \\ &= g^t h^u = a \end{aligned}$$

Verification Ben Bitdiddle wants to modify this scheme into a *blind* signature scheme. In this setup, there is an Authority that holds a secret key and has the power to sign messages. A user would like to have the Authority sign a message m in such a way that the Authority learns nothing about either m or sign(m). Nonetheless, any outsider can verify that sign(m) is a signature of m made by the Authority. One could imagine this kind of application in a notary public that notorizes only sealed envelopes, in a voting scheme, or in an electronic cash systems.

In the original signature scheme, the signer knows m and computes $c = H(m, a)$. Instead of this, Ben envisions the following:

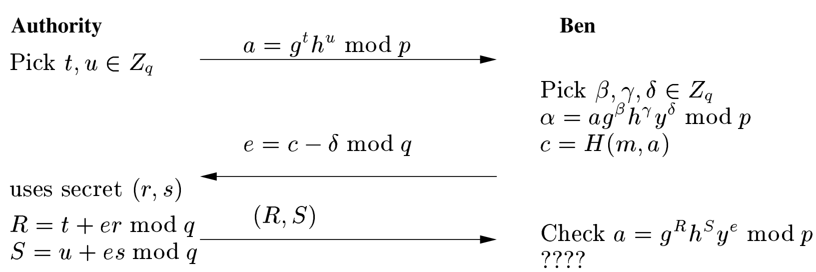

Figure 1: Ben's Blind Signature Scheme

1. The Authority picks $t, u \in Z_q$, and sends Ben the value $a = g^t h^u \bmod p$.
2. Ben picks $\beta, \gamma, \delta \in Z_q$ and blinds a into $\alpha = a g^\beta h^\gamma y^\delta \bmod p$. Ben also computes $c = H(m, \alpha)$ and sends $e = c - \delta \bmod q$ to the Authority.
3. Using the secret key (r, s), the Authority computes $R = t + er \bmod q$, $S = u + es \bmod q$ and sends (R, S) to Ben.
4. Ben makes sure that $a = g^R h^S y^e$ as before in the original scheme. Ben derives a signature for m given R, S.

(a) Help Ben complete this signature scheme by describing exactly what Ben should publish in step (4). At this point, Ben knows $(a, \alpha, \beta, \gamma, \delta, c, e, R, S)$. However, since the signature should be blind, it cannot simply be (a, R, S) as before. Be sure to explain what a verifier needs to check in order to make sure that the signature is correct.

(b) Argue that the Authority learns nothing about either m or sign(m).

(c) Argue that the Authority has no way of linking the values that she sees with either m or sign(m). In other words, suppose the Authority cleverly chooses the values (a, c, R, S) and remembers them. Suppose later she sees a signature $m, (\alpha, \rho, \sigma)$. Is there a way for her to use her database of remembered values to learn anything about who submitted the message or when it was signed?