Massachusetts Institute of Technology
6.857: Network and Computer Security
Professor Ronald L. Rivest

Handout 12
October 9, 1997

Problem Set 5

This problem set is due on *Thursday, October 16, 1997* at the end of class. Late homeworks will *not* be accepted.

Mark the top of each sheet with your name, 6.857, the problem set number, and the date. Type up your solutions, and be clear. Points may be deducted if your TA has problems understanding your solution.

If you collaborate with other students, you MUST write up solutions on your own and acknowledge the people you work with.

Problem 5-1. Short commitments

With the commitment scheme described in class based on the Chaum-van Heijst-Pfitzmann hash function, the commitment typically has a length equal to the length of the modulus p, which may be 1000 bits or more.

For improved bandwidth utilization, consider the modified commitment scheme that only uses the low-order 160 bits of the previous commitment.

Is this "secure"? Argue why or why not. Make explicit any computational assumptions that you need to support your argument.

Problem 5-2. Zero-knowledge with a more general verifier

In class we presented a protocol for "proof of knowledge of discrete logarithm", and argued that is a zero-knowledge against an "honest verifier" (i.e. one who just computes his challenge by following the protocol specified and actually flipping a coin).

This protocol is actually zero-knowledge in general, when the verifier is "dishonest" and computes his challenge, not by flipping a coin, but by *computing* his challenge based on the first message s received from the prover, other information known to the verifier, or other coin flips.

Prove the special case of the above statement when the verifier (unknown to the prover) computes his challenge c as the xor of the two lowest bits of s. That is, prove that the verifier can generate simulated transcripts of the protocol with the same probability distribution as would be generated by actual executions of the protocol.

Recall protocol:

Prover knows a secret x such that $y = g^x \pmod p$ is public. Prover wishes to prove knowledge of x, without revealing x. Here p is a large public prime, and g is a generator of Z_p^*, such that taking logarithms to the base g modulo p is believed to be hard.

1. Prover picks an element k at random $k \in Z_{p-1}$. Prover sends verifier $s = g^k \pmod{p}$.
2. Verifier flips a coin to obtain a challenge $c \in \{0, 1\}$. (For the homework problem, the verifier computes c as the exclusive-or of the low-order two bits of s.)
3. The prover sends the verifier the value $r = k + cx \pmod{p-1}$. (In other words, if $c = 0$, the prover sends the verifier the value $r = k$, and if $c = 1$, the prover sends the verifier the value $r = k + x \pmod{p-1}$. The verifier checks that $sy^c = g^r \leftarrow_{mod} p$).)

The round consisting of the above three steps is repeated t times. The verifier accepts if and only if each check succeeds.