CS487 - Introduction to Cryptography Due on September 11, 2007

Homework 1

Instructor: Rafael Pass

You may collaborate with other students on the homework but you must submit your own individually written solution and you must identify your collaborators. If you make use of any other external source, you must acknowledge it. You are not allowed to submit a problem solution which you cannot explain orally to the course staff.

Problem 1 *Modular Arithmetic and Probability*

1. Prove that $(a \bmod n) + (b \bmod n) = (a + b) \bmod n$
2. Prove that $(a \bmod n) \cdot (b \bmod n) = (a \cdot b) \bmod n$
3. Fact: If Bob is given a randomly chosen Skittle, he will eat it 85% of the time.

   Fact: 20% of all Skittles are yellow.

   Assume Bob eats each individual Skittle with an independent probability solely based on its color. If Bob is fed only yellow Skittles, give tight upper and lower bounds on the percentage that he eats.

Problem 2 *Perfect Secrecy*

Alice claims to use a perfectly-secret encryption scheme to encrypt messages to Bob. However, Eve shows Alice that she can recover 90% of the bits of Alice's key even after just seeing one encrypted message from Alice. Explain how this is possible by constructing a perfectly-secret encryption scheme which has these properties.

Problem 3 *Voting*

There are n professors (warlords) in a room. Together they want to determine whether a majority supports the current Dean (kingpin); individually no one wants to announce in public whether they champion the candidate or prefer to oust her. Indeed, the consequences of making such information public could be dire for those in the minority.

We let professor i's vote $v_i = \{0, 1\}$ represent "no" or "yes." The professors exchange messages as follows:

1. Describe how to implement line 1. Prove your answer is correct.
2. Prove that if all professors follow the protocol, then S will be the tally of their votes.
3. In this problem, you will be asked to come up with a meaningful definition of *perfect secrecy* for this protocol. Intuitively, we do not want any set of colluding parties to learn the votes of any other honest party. Note that this is not possible when all the honest parties have voted in the same way (explain why).

Algorithm 1: Voting Protocol: Professor i proceeds as follows on input v_i

1 Uniformly pick $x_{i,1}, \ldots, x_{i,n} \in \{0, 1, \ldots, 2n-1\}$ s.t. $\sum_{j=1}^{n} x_{i,j} = v_i \bmod 2n$.
2 Send $x_{i,j}$ to professor j.
3 Wait to receive a messages $x_{j,i}$ from each other professor j.
4 Compute $s_i = \sum_{j=1}^{n} x_{j,i} \bmod 2n$. Send s_i to everyone, and wait to receive back a message s_j from professor j.
5 Compute $S = \sum_{j=1}^{n} s_j \bmod 2n$ and output S as the tally of the votes.

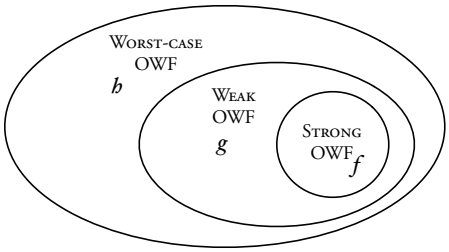
As long as there is at least one honest party that votes yes, and one that votes no, provide a definition of *perfect secrecy* that states that if all parties follow the protocol, then the vote of each individual professor i remains perfectly secure, even if a group C of less than $n-1$ professors collude to try and learn i's vote.

4. Prove that the scheme meets this definition. (Hint: use part 1.)

5. Does the protocol still work if one of the professors deviates from the protocol instructions? Explain either way.

Problem 4 *One-Way Functions*

Three notions of one-way functions were discussed in class: (1) worst-case OWF, (2) weak OWF, and (3) strong OWF. In this problem you will show that (1) is strictly weaker than (2), and (2) is strictly weaker than (3), i.e., that these three notions are related as depicted in the diagram below.



1. Prove that (3) $\Rightarrow$ (2) $\Rightarrow$ (1), i.e. a strong OWF is also a weak OWF and a weak OWF is also a worst-cast OWF.

2. Assume f is a strong OWF. Construct a function g that is a weak OWF, but not a strong OWF. To show that the constructed function g is a weak OWF, you need to prove that if there exists a non-uniform PPT algorithm A that inverts g (with "high" probability), then there exists a non-uniform PPT algorithm A' that inverts f with non-negligible probability.

3. Assume g is a weak OWF. Construct a function h that is a worst-case OWF, but not a weak OWF.