HW 8 CMSC 456. Morally DUE Nov 4
SOLUTIONS
NOTE- THE HW IS THREE PAGES LONG

1. (0 points) READ the syllabus- Content and Policy. What is your name? What is the day and time of the FINAL?

2. (30 points) Recall the following key exchange protocol:

   (a) Alice generates rand prime p of length L, rand $S \times S$ matrix A over $\mathbb{Z}_p$. You can assume A is invertible.

   (b) Alice sends $(p, A, HAHA)$. All public. (HAHA is just our way of taunting Eve and telling her that even though she knows p and A, she can't find the shared secret. Actually, in this case we are wrong about that.)

   (c) Alice generates rand row $\vec{y} \in \mathbb{Z}_p^S$. Sends $\vec{y}A$.

   (d) Bob generate rand column $\vec{x} \in \mathbb{Z}_p^S$, Sends $A\vec{x}$.

   (e) Alice computes $\vec{y}(A\vec{x}) = \vec{y}A\vec{x}$.

   (f) Bob computes $(\vec{y}A)\vec{x} = \vec{y}A\vec{x}$.

   (g) Alice and Bob have shared secret $\vec{y}A\vec{x}$

   Eve only sees $(p, A, HAHA, \vec{y}A, A\vec{x})$. Give an attack that Eve can use to recover $\vec{y}A\vec{x}$.

   SOLUTION TO PROBLEM TWO

   Eve computes the inverse of the matrix A which we call A^{-1}.

   She then computes $A^{-1}A\vec{x} = \vec{x}$.

   Eve knows x and $\vec{y}A$ so she can compute $\vec{y}A\vec{x}$.

   END OF SOLUTION TO PROBLEM TWO

GOTO NEXT PAGE

3. (40 points) Alice and Bob never did like working in mod p or any mod. So they decide to do the following version of Diffie-Hellman.

   i. Security parameters are S, T.
   ii. Alice picks a random $g \in \{2, \ldots, S\}$ and broadcasts g.
   iii. Alice picks a random $a \in \{2, \ldots, T\}$ and broadcasts g^a.
   iv. Bob picks a random $b \in \{2, \ldots, T\}$ and broadcasts g^b.
   v. Alice computes $(g^b)^a = g^{ab}$.
   vi. Bob computes $(g^a)^b = g^{ab}$.
   vii. The shared secret key is g^{ab}.

   We assume that $+, -, \times, \div$ take 1 step each (this is not realistic if S, T are large but this is a homework problem, not the NSA).

   And NOW for the questions:

   (a) (10 points) Show that computing g^a can be done in $O(\log_2(T))$ steps.

   (b) (20 points) Give an algorithm that will, given a $g \in \{2, \ldots, S\}$ and number $x \in \{1, \ldots, Z\}$ (1) if $x = g^y$ for some $y \in \mathbb{N}$ then output y, (2) if $x \neq g^y$ for any $y \in \mathbb{N}$ then output OH, NO SUCH y. The algorithm has to be in time $(\log \log Z)^{O(1)}$. (S may play a role in the base of the log but we ignore this.) You can't just say *take the logarithm base* g, you have to do it using only the basic operations $+, -, \times, \div$.

   (c) (5 points) Eve only sees (g, g^a, g^b). Show how she can efficiently find g^{ab} using Part (b). What is the runtime?

   (d) (5 points) From the above we see that doing Diffie Hellman over the naturals is insecure. Give one more reason why using it is a bad idea.

   SOLUTION TO PROBLEM THREE

   (a) Show that computing g^a can be done in $O(\log_2(T))$ steps.
   ANSWER: use repeated squaring.

(b) Give an algorithm that will, given a $g \in \{2, \ldots, S\}$ and number $x \in \{1, \ldots, Z\}$ (1) if $x = g^y$ for some $y \in \mathbb{N}$ then output y, (2) if $x \neq g^y$ for any $y \in \mathbb{N}$ then output OH, NO SUCH y. The algorithm has to be in time $O(\log \log Z)$. (S may play a role in the base of the log but we ignore this.) You can't just say *take the logarithm base* g, you have to do it using only the basic operations $+, -, \times, \div$.

ANSWER: Compute g^2, g^{2^2}, g^{2^3}, $\ldots$ until you either hit x or exceed it.

If you hit it then you are DONE. AND as for time—since $x \leq Z$, since i is such that $g^{2^i} = x$ we have $i \leq O(\log \log Z)$.

If you exceed it then you have an i such that

$$g^{2^i} < x < g^{2^{i+1}}.$$

Do binary search on this interval to either find y such that $x = g^y$ OR find that there is no such y. The binary search is on an interval of size $2^{i+1} - 2^i = 2^i$, so it takes i steps. NOT SO FAST- each step is squaring the previous one, so takes ONE step. Oh, we're fine. So really $O(\log \log Z)$ steps.

(c) Eve only sees (g, g^a, g^b). Show how she can efficiently find g^{ab} using Part (b). What is the runtime?

ANSWER: Since $g \leq S$ and $a \leq T$, $g^a \leq S^T$. Hence, the algorithm in part 2 takes $O(\log \log S^T)$ steps which is $O(\log(T \log S)) = O(\log T + \log \log S)$.

(d) From the above we see that doing Diffie Hellman over the naturals is insecure. Give one more reason why using it is a bad idea.

ANSWER: As S and T get larger the numbers g^a, g^b can get very large. So DH over $\mathbb{N}$ would use too much space.

END OF SOLUTION TO PROBLEM THREE

GOTO NEXT PAGE FOR NEXT PROBLEM

4. (30 points) Alice and Bob are bridge partners. And they cheat! Here is their scheme:

   - If the first card is placed horizontally then the person placing it has 0 or 1 Ace.
   - If the first card is placed vertically then the person placing it has 2 or 3 or 4 Aces.

   In this problem we will both (1) help Alice and Bob and (2) help the bridge community.

   (a) (15 points) Alice and Bob will be playing 20 games and are worried that their cheating may be discovered. Show how they can use a 1-time pad to make their cheating harder to discover.

   (b) (15 points) Change something about how Bridge is played so that Alice and Bob cannot use their method to cheat.

   SOLUTION TO PROBLEM THREE

   Solution Omitted.