CS 591 Cybersecurity, HW1

Prof. Jared Saia, University of New Mexico

Due March 2nd

1. Recall that for any positive integer n, Z_n^* is by definition the set of numbers between 1 and $n-1$ which are relatively prime with n along with a binary operation which is multiplication mod n. Show that Z_n^* is a group.
   Note: You may find it useful to make use of the EXTENDED-EUCLID algorithm. EXTENDED-EUCLID takes as input two positive integers x and y and returns integers d, a, b such that $d = gcd(x, y) = ax + by$

   Now show that for any group G and any element $a \in G$, $\langle a \rangle$ which is the set of elements generated by a is itself a group under the group operator of G i.e. $\langle a \rangle$ is a subgroup of G.

   Solution: This is fairly straightforward. Come see me if you want the details

2. Show that for any group G, every element in G has a *unique* inverse. Now give an algorithm that for any finite Abelian group G, and any element $a \in G$, computes the inverse of a efficiently (that is in time polynomial in $\log |G|$). You may assume the algorithm is given $|G|$ as input. *Solution: Let* $|G| = n$ *then* $a^n = 1$ *and so* a^{n-1} *(which can be computed efficiently) is the inverse of* a*.*

3. (adapted from Goldreich Ex. 2, p.92) Prove that the existence of 1-way trap door permutations implies that $P \neq NP$.
   Hint: For any polynomial-time computable function f which maps from $\{0,1\}^k$ to $\{0,1\}^k$, define the set $L_f \in NP$ such that if $L_f \in P$, then there is a polynomial-time algorithm (in k) for inverting f. (remember that a permutation is just a special type of function).

   Solution: Assume that 1-way trap door permutations exist and let f *be one such permutation. Let* L_f *be the set of all strings of the form*

(i, y) such that the i-th bit of $f^{-1}(y)$ is 1. L_f is clearly in NP since $f^{-1}(y)$ is a certificate that (i, y) is in L_f. Now assume that L_f is in P. This means that there is a polynomial time algorithm, A, that can recognize strings in L_f. We must show that we can use this algorithm to invert f. Given some value y that we want to find the inverse of, we can simply do a binary search using A as follows. For all i between 1 and k, we ask A if the tuple (i, y) is in L_f. The answers we get back from A give us all of the bits in $f^{-1}(y)$. Thus we can invert f in polynomial time. But this contradicts our assumption that f was 1-way. This implies that L_f must not be in P. This implies that $P \neq NP$. Just FYI, note that we can say something even stronger from this proof: if 1-way permutations exist, $BPP \neq P$.

4. Professor Plum postulates that parity is a perfectly fine hard core bit. More precisely, he postulates that if f is any 1-way permutation over bit strings, that the parity of x is a hard core bit for $f(x)$. Is Plum correct? Please prove your answer either way.

   Solution: Parity is not a hard core bit as we now show. Consider some 1-way permutation $f(x)$. Consider some arbitrary string xb which is the concatenation of x which is $k-1$ bits long and b which is a single bit. Let $p(xb)$ be the parity of the string xb. Define $f'(xb) = f(x)p(xb)$. We first show that f' is a permutation. Consider two arbitrary strings y and y' where $y \neq y'$, we will show that $f'(y) \neq f'(y')$. let $y = xb$ and $y' = x'b'$. If $x \neq x'$ then clearly $f'(y) \neq f'(y')$. If $x = x'$, then $b \neq b'$ and so $p(y) \neq p(y')$ and so again $f'(y) \neq f'(y')$ since the last two bits are different. It's not hard to show that f' is also 1-way if f is 1-way. To show this, note that if there exists an adversary which can guess f'^{-1} with probability q, you can easily construct an adversary which can guess f^{-1} with probability $2q$. But clearly for this 1-way permutation, the parity of the input, z is not a hardcore bit for $f'(z)$.

5. Fill in the details of the proof sketched in class that El Gamal is not semantically secure if Z_p^* is the group used for p prime. In particular, go back to the definition of semantic security and show that you can construct an adversary which ensures that the probability of the "bad" event is non-negligible.

   Solution: This is straightforward. Please come see me if you want the details.

6. Prove that if El Gamal is semantically secure, then the DDH assump-

tion is true (note: this is opposite from the direction done in class).
Solution: The proof is only sketched here. Assume we are given a PPT alg A which attacks the DDH assumption. Then we construct a PPT alg A' which attacks semantic security of El Gamal as follows. A' chooses to guess between $m_0 = 1$ and $m_1 = g^w$ for w some random number between 0 and the order of the group. A' gets back $\mathcal{E}_{pk}(m_b)$ for b a random bit. Note that this value will either be $< g^r, g^{xr} >$ in the case where $b = 0$ or $< g^r, g^{xr+w} >$ in the case where $b = 1$ (note that g^{xr+w} is essentially g to some random power since w is random). A' feeds the values g and the two values from the encryption into A. If A says that these three values satisfy DDH then A' guesses that $b = 0$ otherwise A' guesses that $b = 1$.

7. In the following problem, we will say that a probability, p, is 1/2-negligible if $|p - 1/2|$ is negligible.

   Assume Alice, Bob and Carol each have a private salary (a,b and c respectively) and they all want to know the set of their salaries while retaining as much privacy as possible. More precisely, note that if Alice (respectively Bob or Carol) learns $\{a, b, c\}$, she can identify her own salary in this set but she should have 1/2-negligible probability of guessing which of the remaining salaries map to the other players.

   Assume that Alice, Bob and Carol are honest but curious. In other words, they will exactly follow any protocol you give them but at the end of the protocol, they will carefully study the information they have gathered to try to guess the salaries of everyone else. Further assume that there are private channels between any pair of the three parties e.g.. Alice and Bob can communicate secretly without Carol overhearing anything.

   Design a protocol for Alice, Bob and Carol which has the desired properties described above. *Prove that your protocol has the desired property.* You may assume that El Gamal is semantically secure.

   Solution: Alice generates public and secret keys (pk, sk) using El Gamal(or any semantically secure PKE). She then sends Bob and Carol pk. Bob then sends Carol $\mathcal{E}_{pk}(b)$. Carol sends Alice $\mathcal{E}_{pk}(b)$ and $\mathcal{E}_{pk}(c)$ in a random order. Finally Alice decrypts what Carol sends her in order to recover $\{b, c\}$, she then sends $\{a, b, c\}$ to both Bob and Carol. Correctness of this protocol is immediate from the correctness of El Gamal.
   We now show that each player doesn't obtain too much information. Bob receives no information except for pk and $\{a, b, c\}$ at the end so he

can not break the privacy constraint. Alice receives $\{b, c\}$ but they are unordered so the probability of guessing which salary belongs to whom is $1/2$-negligible.

Carol is the only tricky part. She receives the message $\mathcal{E}_{pk}(b)$ and the information $\{a, b, c\}$. Assume Carol had a PPT algorithm A which takes as input $\{a, b, c\}$ and $\mathcal{E}_{pk}(x)$ and determines whether x is a or b with probability q where q is not $1/2$-negligible. Then there exists a PPT algorithm A' which breaks the semantic security of El Gamal as follows. Assume A' has hard coded into it the values a, b and c (surely there exists some A' which has these values hardcoded into it). A' chooses to guess between the strings $m_0 = a$ and $m_1 = b$. When given $\mathcal{E}_{pk}(m_r)$ for some random bit r, it feeds this value along with the value $\{a, b, c\}$ into A. A' then outputs the value 1 iff A outputs b. A' has probability q of success and so A' breaks the semantic security of El Gamal. This contradicts our assumption so the algorithm A must not exist.

8. (CHALLENGE) Can you generalize the above result to n players? In other words, can you design a protocol where each player has a $1/n$-negligible probability of guessing which salary belongs to any particular player other than herself? Now can you design a protocol which takes only $O(logn)$ rounds?

   Solution: This is a straightforward extension of the above. Come see me if you want details.