CS 591 Cybersecurity, Final

Prof. Jared Saia, University of New Mexico

Due May 12th at 5pm

You can use any textbook in completing this exam and you may refer to your class notes (and the notes on Jonathan Katz's web page) and the papers we discussed in class. However, do not discuss the exam with anyone and do not try to look up answers on the Internet. When you are finished, please either email the finished exam to me or put it in my mailbox or under my office door.

1. Consider the following function $f(G, C) = G$ where G is a graph and C is a set of vertices that form a maximum clique in G (in the case where there is more than one maximum clique, C is the first in some canonical ordering of cliques determined by the indices of vertices they contain). Assuming $P \neq NP$, is f a 1-way function? Justify your answer. You will need to make use of the definition of 1-way functions, which we discussed in class. *Solution: f is not a 1-way function. In particular, we know that if $P \neq NP$, it is hard to compute the inverse of f on an infinite number of instances in the range of f. However, this does not imply that it is hard to find the maximum sized clique in a* random *graph. Hardness to compute the problem on a random input instance is critical in order for f to be 1-way.*

2. Following are three question about the exponential information gathering algorithm discussed in class.

   - What is the number of nodes at the level i layer of each information gathering tree when the algorithm ends?
   - Consider two information gathering trees T_1 and T_2 which belong to two good players and assume the sender is a bad player. What is the minimum number of children of the root node that will have the same reduced value in both T_1 and T_2?

- What if the sender is a good player. What is the minimum number of children of the root node that will have the same reduced value in both T_1 and T_2?

*Solution: If $i = 0$, it is 1, otherwise it is $n - i$ * the number of nodes at the $i - i$ level. In other words, $(n-i)*(n-i+1)*...*(n-1)*1$. If the root node corresponds to a bad player, then by the lemma discussed in class, the reduced values of* all *the children nodes of the root will be the same in T_1 and T_2. If the root node is a good player, then only the children nodes associated with good players are guaranteed to have the same reduced value. The number of children nodes associated with good players will be $n - t - 1$, so this is the minimum number of nodes that share the same reduced value.*

3. The ϵ-approximation cake-cutting algorithm we discusses in class assumed the existence of a trusted third party to pose questions to the players, collect the answers to these questions and then calculate how to cut the cake and assign pieces to the players. Describe how you could implement this algorithm if there is not such a trusted third party. Assume that a 1/10 fraction of the players in the protocol are bad. These bad players may give different answers to different players, not answer questions, not obey the rules of the protocol, etc., but they are computationally bounded. Your goal is to guarantee that at the end of the protocol, all the good players agree on how to cut the cake and assign the pieces and that this cut and assignment of pieces is the same as would have been decided by the trusted third party.

   Solution: Each player simulates the protocol, asking other players for which piece the prefer at the appropriate time. Whenever a question is asked by the trusted third party in the original protocol, the queried player simply sends its answer to all other players in our new protocol. All players then use Byzantine agreement to come to consensus on what the queried player actually answered. Note that the protocol is deterministic so everyone knows who the queried player will be at each step. If the queried player is good, they will give the same answer to everyone. If they are bad, they may give different answers to different players, but after running the agreement protocol, the players will at least come to consensus on some answer (note that the answers are not just yes or no, but as discussed in class, we can solve agreement on muti-valued inputs simply by running single bit Byzantine agreement multiple times). Finally at the end, all the good players will agree

on how the cake should be cut and who should get each piece for the following reasons: 1) all players will have the same inputs as to the sequence of questions asked and 2) the algorithm that decides the cut and division of pieces is a deterministic one.

4. Imagine there are n players connected in a ring topology. What is the maximum number of faulty players that can be tolerated to perform Byzantine agreement in such a network. Hint: You will need to mimic the "hexagon proof" we went over in class. First look at the situation for $n = 4$ and then see if you can generalize to arbitrary n. *Solution: t can not even be one. To see this, assume there is some algorithm to solve Byzantine agreement on such a network. First set up a $2n$ node ring network, G', that looks like $s, 1, 2, 3, ..., n, s', 1', 2', ...n', s$. Next, consider the case where the players all execute the algorithm honestly communicating but where s has input 0 and s' has input 1. Note the following three facts. 1) From the point of view of players $1, 2, ..., n$ the action of s and s' represents possible adversarial behavior. Thus, all of these players must output the same value. 2) From the point of view of players $s, 1, ..., n-1$, the actions of players n and n' represent possible malicious behavior of player n in the original network. Thus, all of these players must output a 0. 3) A symmetric argument with respect to s' and 1 shows that players $2, 3, ..., s'$ must all output a 1. However, this is a contradiction so the original algorithm cannot exist.*

5. In a special game-theory episode of "Fear Factor", Alice, Bob and Eve are required to eat pieces of a very disgusting cake. In this new situation, a player will always prefer an empty piece of the cake to a non-empty piece. However, different players may prefer different non-empty pieces. For example, Alice may have less of an aversion than Bob to a piece with maggots and Bob may have less of an aversion than Eve to a piece with pig ovaries. Give a variant of the cake-cutting algorithm discussed in class that ensures that this cake is split up fairly. Please show that your algorithm works correctly (up to ϵ small crumbs of cake).

   Solution: This is just the chore division problem. One can use the same algorithm described in the "Rental Harmony" paper we read in class. The key observation is that even though for this new problem the labels on the vertices will not be a Sperner labeling, it's still the case that some triangle has a 1,2 and 3 vertex. See the rental harmony paper for details on how to prove this.