Question 1 ***A Tour of Tor*** ()

As a reminder, when connecting to a normal website through Tor, your computer first queries the Tor "consensus" to get a list of all Tor nodes, and using this information it connects to the first Tor node and, from there, creates a circuit through the Tor network, eventually ending at an exit node.

Q1.1 (4 min) Consider the scenario where you are in a censored country and the censor choses not to block Tor, the censor is the adversary, and no Tor relays exist within this country. How many Tor relays must your traffic pass through, including the exit node, to prevent the censor from blocking your traffic.

- ◯ One
- ◯ Two
- ◯ Three
- ◯ Four
- ◯ Tor doesn't stop this adversary

Q1.2 (4 min) Consider the scenario where you are the only user of Tor on a network that keeps detailed logs of all IPs contacted. You use Tor to email a threat. The network operator is made aware of this threat and that it was sent through Tor and probably originated on the operator's network. How many Tor relays must your traffic pass through, including the exit node, to guarantee the network operator can't identify you as the one who sent the threat?

- ◯ One
- ◯ Two
- ◯ Three
- ◯ Four
- ◯ Tor doesn't stop this adversary

Q1.3 (4 min) Consider the scenario where there is a single hostile Tor node but you don't know that node's identitity, and that node can be an exit node. You want to keep confidential from this node what HTTP sites you are visiting through Tor. How many Tor relays must your traffic pass through, including the exit node, to guarantee this adversary can't know what sites you visit?

- ◯ One
- ◯ Two
- ◯ Three
- ◯ Four
- ◯ Tor doesn't stop this adversary

Q1.4 (4 min) Consider the scenario where there are mulitple independent hostile Tor nodes but you don't know their identities, and these nodes can be exit nodes. You want to keep confidential from all these nodes what HTTP sites you are visiting through Tor. How many Tor relays must your traffic pass through, including the exit node, to guarantee that every independent hostile node can't know what sites you visit?

- ◯ One
- ◯ Two
- ◯ Three
- ◯ Four
- ◯ Tor doesn't stop this adversary

Q1.5 (4 min) Consider the scenario where there are multiple colluding hostile Tor nodes but you don't know those nodes identities, and these nodes can be exit nodes. You want to keep confidential from all these nodes what HTTP sites you are visiting through Tor. How many Tor relays must your traffic pass through, including the exit node, to guarantee that the colluding system of hostile nodes can't know what sites you visit?

- ◯ One
- ◯ Two
- ◯ Three
- ◯ Four
- ◯ Tor doesn't stop this adversary

Q1.6 (4 min) Consider the scenario where there is a single hostile Tor node but you don't know that node's identity, and that node can be an exit node. You want to have data integrity for the HTTP sites you are visiting through Tor. How many Tor relays must your traffic pass through, including the exit node, to guarantee this adversary can't manipulate the data you receive from the sites you visit?

- ◯ One
- ◯ Two
- ◯ Three
- ◯ Four
- ◯ Tor doesn't stop this adversary

Question 2 ***Bitcoin*** **()**

Assume a simplified Bitcoin model, where each block contains the following fields:

- `minerID`: The public key of the node who mined this block. Recall that the person who mined a block is given a mining reward in Bitcoin. Assume that a miner can redeem this award by simply referencing the block ie. the initial award is *not* stored as a transaction.
- `prevHash`: The hash of the previous block
- `transactions`: The list of transactions. Recall each transaction contains references to its origin transactions, a list of recipients, and is signed using the private key of the coins' owner.
- `nonce`: A value such that the hash of the current block contains the correct number of zeros

Assume that the hash of a block is computed as:

```
Hash(minerID || prevHash || transactions || nonce)
```

Bob wants to save on computing power by omitting certain fields in a block from being part of the hash. For each modified block hashing scheme below, select all the things an adversary with a single standard CPU can do.

Assume that if the adversary can come up with a modified blockchain of the same length, the rest of the network will accept it. Furthermore, assume the adversary has not made any transactions thus far. **Any option that could result in an invalid state should not be selected.**

Q2.1 (4 points) Each block hash is computed as `Hash(prevHash || transactions || nonce)`

☐ (A) Modify a block to gain Bitcoin

☐ (B) Given some amount of pre-computation, can consistently win proof of work

☐ (C) Modify some transaction amounts

☐ (D) Can remove any transaction in an arbitrary block by *only* modifying that block

☐ (E) None of the above

☐ (F) ——

Q2.2 (4 points) Each block hash is computed as `Hash(minerID || transactions || nonce)`

☐ (G) Modify a block to gain Bitcoin

☐ (H) Given some amount of pre-computation, can consistently win proof of work

☐ (I) Modify some transaction amounts

☐ (J) Can remove any transaction in an arbitrary block by *only* modifying that block

☐ (K) None of the above

☐ (L) ——

Q2.3 (4 points) Each block hash is computed as `Hash(minerID || prevHash || nonce)`

- ☐ (A) Modify a block to gain Bitcoin
- ☐ (B) Given some amount of pre-computation, can consistently win proof of work
- ☐ (C) Modify some transaction amounts
- ☐ (D) Can remove any transaction in an arbitrary block by *only* modifying that block
- ☐ (E) None of the above
- ☐ (F) ——

This is the end of Q2. Leave the remaining subparts of Q2 blank on Gradescope, if there are any. You have reached the end of the exam.