Tor, Bitcoin

Question 1 ***True/false***

Q1.1 Assume you've set up a 3-relay Tor circuit to access some websites over HTTPS. A malicious adversary takes control of the entry relay, but the other two are honest and uncompromised.

TRUE or FALSE: The adversary can now learn your identity.

● TRUE ○ FALSE

Solution: True.

Q1.2 Assume you've set up a 3-relay Tor circuit to access some websites over HTTPS. A malicious adversary takes control of the middle relay, but the other two are honest and uncompromised.

TRUE or FALSE: The adversary can now learn which website you are visiting.

○ TRUE ● FALSE

Solution: False, the exit relay protects against this.

Q1.3 Assume you've set up a 3-relay Tor circuit to access some websites over HTTPS. A malicious adversary takes control of the exit relay, but the other two are honest and uncompromised.

TRUE or FALSE: The adversary can now learn your identity.

○ TRUE ● FALSE

Solution: False, the entry and middle relays protect against this.

Q1.4 TRUE or FALSE: In Bitcoin, once your transaction is successfully added to a block that lives on the longest chain, you can be guaranteed that it will never be lost.

○ TRUE ● FALSE

Solution: False. The blockchain could fork and not include your transaction.

Q1.5 TRUE or FALSE: If you had 70% of the hashing power in Bitcoin, you would be able to add spoofed transactions to the blockchain

○ TRUE ● FALSE

Solution: False, transactions require a valid signature. Having a large amount of hashing power doesn't allow you to forge signatures. What we would worry about now is *double-spending* attacks.

Q1.6 TRUE or FALSE: In Bitcoin, including the hash of the previous block is only useful for ensuring immutability and append-only properties

○ TRUE ● FALSE

Solution: False. Proof of work also relies on this. We need some public, un-predictable value to be part of the hashing problem for each block. This ensures that the hashing problem "restarts" on each new block added to the blockchain so an adversary must race the rest of the network - they can't simply precompute a valid block.

Note that are other ways you could enforce this property but in Bitcoin the fact that you include the hash of the previous block accomplishes it.