University of Stavanger

Faculty of Science and Technology

Exam in subject:	**DAT650 Blockchain technology**
Date:	**Prøveeksamen, November 2019**
Duration:	**4 hours**
Allowed remedies:	**None**
The exam consists of:	**9 exercises on 6 pages**
Contact during exam:	**Leander Jehl, tlf. (518) 32062**
Remarks:	None
Attachments:	

Question 1: Data structures (7%)

(a) (4%) Sketch a Merkle tree with 8 data elements. What data needs to be included in an inclusion proof for the third data element?

> **Solution:** Need hashes h_4, $h_{1,2}$, and $h_{5,6,7,8}$. Additional position information, left, right, left.

(b) (3%) Let t be a transaction included in the bitcoin blockchain. What can we say about, when t was issued?

> **Solution:** The transaction is included in a block, that contains a timestamp. The transaction must be issued, shortly before that block timestamp.

Question 2: Unspend transaction output (UTXO) (9%)

(a) (6%) Which checks need to be done in bitcoin to validate a block?

> **Solution:**
>
> - Transactions in the block are valid
> - Hash of block meets difficulty. (solves pow)
> - Timestamp within reasonable interval

(b) (3%) Alice owns 3 bitcoin outputs, worth 2, 1, and 0.5 bitcoin. How can Alice create a transaction that sends 2.5 bitcoin to Bob. Remember that Alice needs to pay fees to the miners (e.g. 0.001 bitcoin).

Solution: Alice creates a transaction that has the 2 and 1 bitcoin outputs as input. Alice creates two outputs for that transaction. One output for Bob with 2.5 bitcoins and one with 0.499 bitcoins for herself.

Question 3: Proof of work (20%)

(a) (10%) We has the following definition for a proof of work function:

Def: *For an integer d, the proof-of-work (PoW) function with difficulty d takes a data item and returns a nonce (random bits) and a hash value:*

$$(h_{PoW}, nonce) = f_{PoW}(Data)$$

The proof of work is valid*, if a)* h_{PoW} *is the hash of the data, concatenated with the nonce*

$$h_{PoW} \stackrel{?}{=} H(Data||nonce))$$

and b) the first d bits of h_{PoW} *are 0.*

What are the short-commings of this definition and how can it be adjusted?

Solution: The above definition does not allow fine tuning the difficulty. Only allows to make PoW twice or half as difficult.

A better definition for a difficulty is to take d a hexadecimal number and require $h_{PoW} < d$.

(b) (5%) In Bitcoin, that has an average block delay of 10 minutes, what is the probability that a block is found in the next second?

Solution: $\frac{1}{600}$. For different seconds, wether a block is found, is an independent random variable. That is because for every nonce tried, the probability of success is the same.

Question 4: Fees and forks (10%)

(a) (2%) In bitcoin, which transactions require a high fee and which a low fee?

Solution: Transactions that take a lot of bytes, e.g. because they consume many inputs and create many outputs, are more expensive than transactions with few inputs and outputs.

(b) (2%) Name one reason why bitcoin provides a block reward to miners, additional too fees.

Solution: For example:

- to bring bitcoin into circulation
- to ensure small fees
- to ensure miners do not fight over fees

(c) (2%) Why do forks occur in bitcoin?

Solution: If two miners find a block concurrently, a fork occurs.

(d) (2%) Why do forks pose a problem?

Solution: During a fork, the blockchain is undecided on the system state. When one fork is chosen, transactions on the smaller fork will be rolled back.

(e) (2%) Are the following changes hard, soft or hard and soft forks?

1. Replacing SHA256 used for proof of work in Bitcoin with SHA3.
2. Allowing transactions to be signed using both ECDSA and RSA signatures, (opposed to only ECDSA currently used).

Solution: 1 is hard and soft fork. 2 is soft fork.

Question 5: Attacks (20%)

(a) (4%) What are the two rules that an honest miner should follow in Bitcoin?

Solution: An honest miner should try to extend the longest chain and publish a block as soon as it is found.

(b) (6%) Name one way in which a miner could deviate from these rules. Give an example where this could be beneficial for a miner.

Solution: A miner could deviate by trying to extend a different chain than the longest chain (i.e. stubborn mining). For example assume that the miner found one block in a fork, but the other branch was extended first and became the longest chain. Thus the miner looses his block reward unless he manages to extend his block.

(c) (10%) What is a delivery denial attack? How could it be avoided? Explain the underlying trade-off between efficiency and security related to this.

Solution: In a delivery denial attack a node in the bitcoin network forwards an inventory, but on request does not forward the block. In this scenario, the node requesting the block will wait for a long timeout, before requesting the block from a different node.

This attack could be avoided by requesting the block from multiple nodes. The design involves a tradeoff. Requesting the block only once is more efficient (saves bandwidth). Requesting the block multiple times in parallel may improve security but creates unnecessary network traffic.

Question 6: Alternative PoW (4%)

(a) (4%) What is necessary to perform a 51% attack in a Proof of stake system?

Solution: Proof of stake assigns voting power based on money or stake that participants have locked. Thus to perform a 51% attack requires 51% of the staked money.

Question 7: Scaling blockchain (30%)

(a) (4%) Name a reason to increase block size in bitcoin. What problems does an increased blocksize cause?

Solution: An increased blocksize means that more transactions can be committed every 10 minutes. However, a larger block may take longer to be propagated through the network. This larger network delay will result in a higher fork probability.

(b) (4%) Give an example of a blockchain, including forks where the GHOST rule would lead to selecting a different chain than the longest chain rule.

(c) (2%) Name one advantage of an inclusive blockchain, i.e. uncles in Ethereum.

Solution: Uncles reduce the need to form mining pools, since a fraction of the block reward is received even on a loosing branch.

(d) (4%) Give an example how the risk from performing a selfish mining attach is reduced through uncles in Ethereum.

Solution: In selfish mining a miner does not publish his block immediately to the network. If another block is found, the selfish miner publishes his block, but he now risks that his block ends up on the loosing branch. In Ethereum, even if the block from the selfish miner ends on the loosing branch, he will receive part of the block reward, as uncle reward.

(e) (10%) Bitcoin-NG defines both key-blocks and micro-blocks. Fees from Micro-blocks are divided 40/60 between the creator of the last and next key block. What attack would be possible if the creator of the next key block would only get 40% of the fees, e.g. a 60/40 divide?

Solution: A miner could, instead of extending the micro-blocks, extend the last key block. This allows him publish his own micro-blocks, including the transactions. He thus gets 60% instead of 40% from the fees.

(f) (6%) Sharding is proposed to significantly improve the performance of blockchains. Name two problems of difficulties that blockchain applying sharding has to solve.

Solution: Give two of the following three:

- Need to decide, which transactions should be processed in which shard.
- Need to counteract that an adversary takes over 51% of one shard.
- Need to enable cross shard transactions.

Question 8: System models and BFT (14%)

(a) (2%) What is the difference between a permissioned and an unpermissioned/permissionless system.

Solution: In a permissioned systems a membership table exists.

(b) (2%) What is the equivalent of a 51% attack in a permissioned system?

Solution: The equivalent to a 51% attack is when an attacker controls a majority of the identities registered in the system.

(c) (10%) In Bitcoin a block counts as confirmed is 6 other blocks have been added on top.

- In the BFT algorithm that we discussed in class: How does a block get confirmed?
- What is the main difference between the guarantees that are given for a confirmed block in Bitcoin and in BFT?

Question 9: Ethereum (17%)

(a) (5%) Transactions in Ethereum can execute certain functions on a smart contract. How is the cost of this execution determined? Mention the gas price.

Solution: Each individual bytecode in the ethereum virtual machine costs a specific amount of gas. Thus the execution of a function costs a specific amount of gas, based on the bytecodes executed. The transaction specifies the gas price, i.e. the amount of ether or money payed for the gas.

(b) (5%) It is possible to test and profile, how much gas a function costs to execute. Why can the actual execution cost still be different?

Solution: If you iterate over a list, new elements can be added to the list without you knowing.

(c) (2%) What vulnerability can be prevented if you use the Open-zeppelin safe math library?

Solution: Integer overflow.

(d) (5%) Explain re-entrance vulnerability? When sending money from one contract to another using call.

Solution: The call executes the default function or of the receiving contract. This receiving contract can again call the sending function on the sending contract. Checks that guard the Call will still be valid, unless state has been changed before invoking the call function.