Sheet 2, WS 2018/19
Cryptocurrencies and Blockchain
Published 10.12.2018
Deadline 11.01.2019

Katinka Wolter
Larissa Zech
Inst. für Informatik
Freie Universität Berlin

Assignment 1. Bitcoin Various Topics

Please explain your answers to the following questions and quantify your answers as far as possible.

1. **Transaction validation:**
   Consider the steps involved in processing Bitcoin transactions. Which of these steps are computationally expensive? If youre an entity validating many transactions (say, a miner) what data structure might you build to help speed up verification?

2. **Green addresses:**
   One problem with green addresses is that there is no punishment against double-spending within the Bitcoin system itself. To solve this, you decide to design an altcoin called GreenCoin that has built-in support for green addresses. Any attempt at double spending from addresses (or transaction outputs) that have been designated as green must incur a financial penalty in a way that can be enforced by miners. Propose a possible design for GreenCoin.

Extra question: more forking

1. The most prominent Bitcoin hard fork was a transient one caused by the version 0.8 bug.
   How many blocks were abandoned when the fork was resolved?

2. The most prominent Bitcoin soft fork was the addition of pay-to-script-hash. How many blocks were orphaned because of it?

3. Bitcoin clients go into safe mode when they detect that the chain has forked. What heuristic(s) could you use to detect this?

Assignment 2. Transaction fees

Consider the following definitions for the priority of a transaction:

$$\mathrm{P}riority = \sum(\mathrm{i}nputAge * \mathrm{i}nputValue)/(\mathrm{t}ransSize)$$

where the approximate transaction size is (in bytes)

$$transSize = 148N_{\mathrm{inputs}} + 34N_{outputs} + 10.$$

Let the fees be computed following the given simple rule:

```
No fee if
        tx less than 1000 bytes in size,
        all outputs are 0.01 BTC or larger, and
        priority is large enough

Otherwise fee is 0.0001 BTC per 1000 bytes
```

1. Alice has a large number of coins each of small value v, which she would like to combine into one coin. She constructs a transaction to do this, but finds that the transaction fee shed have to spend equals the sum of her coin values. Based on this information (and the default transaction fee policy specified above), estimate v

2. Can Alice somehow consolidate her coins without incurring any transaction fee under the default policy?

3. Compared to a fee structure that doesnt factor the age of the inputs into the transaction fee, what effect might the current default fee structure have on the behavior of users and services?

Assignment 3. Multi-signature wallet

1. BitCorp has just noticed that Mallory has compromised one of their servers holding their Bitcoin private keys. Luckily, they are using a 2-of-3 multi-signature wallet, so Mallory has learnt only one of the three sets of keys. The other two sets of keys are on different servers that Mallory cannot access. How do they re-secure their wallet and effectively revoke the information that Mallory has learned?

2. If BitCorp uses a 2-out-of-2 instead of a 2-out-3 wallet, what steps can they take in advance so that they can recover even in the event of one of their servers getting broken into (and Mallory not just learning but also potentially deleting the key material on that server)?

Assignment 4. Node in a blockchain

Please consider the attachment assignment3.zip. Please implement the requested functions and demonstrate that your validation of transactions is correct. Please comply with the following specifications:

1. Solutions can be submitted in groups of up to 3 students. Please hand in only one submission per group.
2. Only .pdf and .java files are accepted.
3. Name your group members in all .pdf files.
4. Do not make any changes to the framework. Especially, do not change the naming of the .java-files. If you feel like you really need to change the framework, point that out in your .pdf submission and give detailed description on why it is necessary for your solution!
5. Please hand in your validation results together with your code.
6. If your implementation fails during compiling or execution, point that out in your .pdf submission! Do not hand in code that is not compiling without any comments!