Michael Parisi (mjp9zn)
CS 4501 – Cryptocurrency Cabal
Fall 2015

Problem 1. In Section 6, Satoshi writes: *"The incentive may help encourage nodes to stay honest. If a greedy attacker is able to assemble more CPU power than all the honest nodes, he would have to choose between using it to defraud people by stealing back his payments, or using it to generate new coins. He ought to find it more profitable to play by the rules, such rules that favour him with more new coins than everyone else combined, than to undermine the system and the validity of his own wealth."*

What are the assumptions necessary to support Satoshi's claim that it is more profitable for a greedy attacker with a majority of the mining power "to play by the rules"? (other than the assumption that the greedy attacker is a "he")

The main assumption necessary for this claim is that undermining the system and the validity of transactions will decrease participation in the Bitcoin network and as well as decrease a coin's value.

Problem 2. At the end of Section 11, Satoshi presents a table for $p < 0.001$, where it is listed "q=0.45 z=340". What does this mean in plain English, expressed in a short sentence?

If the change of an attacker finding the next block is 45%, and it is already 340 blocks behind, there is a less than .1% chance of him catching up to the current block.

Problem 3. For that same table, what would the z values be for $p < 0.05$ (instead of $p < 0.001$)?

```
for p < 0.05
d=0.1           z=4
d=0.15          z=4
d=0.2           z=6
d=0.25          z=7
d=0.3           z=11
d=0.35          z=18
d=0.4           z=37
d=0.45          z=138
```

Problem 4. One naive approach would be to just sign each record data with a private key, and verify the signature on reading it.

(a) There is a problem with this scheme unless the bytes of data[*i*] indicate that it is in fact from *i*-th index. What is this problem?

(b) Suppose we perform a write at position *i*, both data and signature. Later on, when we read it back, if the signature matches the data, can we be sure that it is indeed the data item we wrote? Explain.

a. Since you have signed each record data, the cloud storage service provider could return any piece of stored data with your signature instead of the specific data requested at index i.

b. If the signature matches the data, you can be sure that that data was unmodified, but you cannot be sure that it was the correct data for that index.

Problem 5. Another approach would be hashing the concatenation of all records in the database. This hash is a small item that can be stored locally.

(a) What is the write/read/verify procedure for this system?

(b) How does the cost of reading and writing to the database scale with *n* (the number of records)?

a. Write: when you write a new file to the cloud storage, you would need to recover the previous concatenation of all records in the database, add the one for the file you intend to write, and recompute the hash.

Read: when you read a file from the cloud, you may just retrieve that file. Verifying it afterwards is the tough part.

Verify: to verify any data in storage, you would have to retrieve all the others as well, so that you can concatenate it with the data you want to verify, and compare the whole hash to the one you stored.

b. The cost to write is order n, since you must retrieve all the other records to compute the new hash. The cost of reading is also order n, but because you need all the other records to verify the one you want to read.

Problem 6. Instead of using the concatenating all the records linear, they were organized into a Merkle tree.

(a) What is the write/read/verify procedure for this system?

(b) How does the cost of reading and writing to the database scale with *n* (the number of records)?

a. Write: to write a file to the cloud, you add it to the tree, recompute the Merkle root by requesting and hashing the files along the path of the new item to the root.

Read: to read from the cloud, you request the file and verify.

Verify: to verify the received file, you must request the other files along its path to the Merkle root. From there, you compute the hashes of the item and its sibling all the way up to the root, where you compare it to the one you stored.

b. The cost of writing is order logn, since you only need the hashes along the new item's path to the root, which is of length logn. Reading is also logn, since to verify a file, you need the same information and to preform essentially the same operations as writing, before comparing it with the stored root.

Problem 7.
(a) If a mining pool has 15% ($\alpha = 0.15$) of the total network hashing power, how many blocks is it expected to find in a day?

(b) Obtain a general formula for expected number of blocks a mining pool with α fraction of the total hashing power should find in *t* minutes.

a. 1 day * 24 hours / day * 60 minutes / hour * 1 block / 10 minutes * 0.15 = 24*60/10*(.15) = **21.6 blocks**

b. $f(\alpha , t) = (t/10)*\alpha$

Problem 8. Assuming all of the miners are honest, what is the expected number of orphaned blocks per day for an honest mining pool with hashing power α and latency *L* (as simplified above).

(1 block / 600 sec) * (L sec / 1 block) = L/600 = chance of a found block being an orphan = (L / 600) orphans / 1 block ➔ α * (1 block / 10 min) * (60 min / hour) * (24 hour / day) * ((L / 600) orphans / block) = α*6*24*(L/600) = **(6* α*L)/25**

Problem 9. How does this change if the mining pool is mining selfishly? (For this question, assume that the selfish mining pool learns of a block announced by the rest of the network as soon as it is announced, so will immediately announce any withheld blocks at that time. That is, you may still assume the simplified latency *L* model, but that the selfish mining pool has a spy in the other supernode with a low-latency direct connection to the mining pool.)

The number of orphaned blocks would increase, as the first to find a block would not publish it immediately, which would therefore give the other miners more than L time to find a second block.

Problem 10B. In upcoming classes, we will have visits from a law professor (who also works for the State Department on international cyberlaw and promulgating the open Internet) and an FBI agent who works on criminals using bitcoin for ransom. Come up with at least one good question that you would like one of them to answer.

To the law professor: In your educated opinion, do you believe that Bitcoin or other crypto-currencies are integral in the open internet, or is the relationship more like the inverse of that? Also, is a Bitcoin considered a good or a currency?

To the FBI agent: How do you conquer the maze of transactions when attempting to identify someone through the Bitcoin network?