Problem Set 2

Collin Berman
cmb5nh

October 9, 2015

Problem 1. Satoshi is assuming the only non-honest way the attacker could use his power is by stealing back his payments. We know that there are in fact other mining strategies, such as witholding blocks, that can be more profitable.

I addition, Satoshi is assuming that the attacker has to choose between the two given options. Perhaps the attacker could generate new coins while he steals back his payments.

Problem 2. If the attacker controls 45% of the total processing power, the recipient of a new transaction needs to wait 340 confirmations before being 99.9% certain the attacker can't change the transaction.

Problem 3.
```
P < 0.05
q=0.10  z=3
q=0.15  z=3
q=0.20  z=5
q=0.25  z=6
q=0.30  z=10
q=0.35  z=17
q=0.40  z=36
q=0.45  z=137
```

Problem 4.

a. All a valid signature tells us is that the data is in fact a record we stored on the server; it need not be from index i.

b. No. The data could be the record stored at position i from before we performed the write.

Problem 5. I'm assuming read doesn't verify

a. To verify, we must download all the data, concatenate it, and check the hash. To write, we must download all the data, verify it, concatenate the data with our replacement block, update our local hash, then push the replacement block to the server. To read we just request a block.

b. Reading is always constant. But write and verify both require have the whole data in order to concatenate for the hash. So time and memory are $\Theta(n)$.

Problem 6. We will store the root hash locally.

a. To read a record, simply request it. To verify the record, ask for the siblings of the node's ancestors in the merkle tree, then compute the root hash and compare with the one we have stored. To write, ask for the siblings of the node's ancestors in the merkle tree, then compute the new root hash given the replacement block and store it locally. Then send the replacement block.

b. Reading is constant, but write and verify are $\Theta(\log n)$.

Problem 7. a.

$$\frac{1 \text{ block}}{10 \text{ minutes}} \times \frac{60 \text{ minutes}}{\text{hour}} \times \frac{24 \text{ hours}}{\text{day}} \times 15\% = 21.6 \frac{\text{blocks}}{\text{day}}$$

b.

$$\frac{1 \text{ block}}{10 \text{ minutes}} \times (t \text{ minutes}) \times \alpha = \frac{\alpha t}{10} \text{ blocks}$$

Problem 8. The honest mining pool has L seconds to mine an orphan block whenever the rest of the network finds another block. The pool will mine $\frac{\alpha L}{600}$ blocks in each of these intervals, and will have $144(1-\alpha)$ chances to do so each day. Therefore we expect the pool to mine

$$\frac{6}{25} L(1-\alpha)$$

orphan blocks a day.

Problem 9. The mining pool will mine an orphan block when it has mined a single private block and the other supernode also mines a block. Then the selfish pool will publish its single block, and continue to mine on it. The other supernode will mine on the block it published. Note that no matter who wins, one of the blocks will be orphaned. I'm not sure how to analyze how often this happens.

Problem 10. b. The FBI has been calling for cryptographic backdoors to help them fight criminals "going dark." Has anyone in the government been calling for similar backdoors in the Bitcoin protocol?