1. Satoshi assumes that he would care about the value of the bitcoin that he currently has and may continue to earn. Satoshi also assumes that his attempts to undermine the bitcoin network would cause a massive devaluation of bitcoin, which is reasonable. If malicious attacks created by this person are discovered (which they likely would be), then everyone would come to the consensus that bitcoin is worthless and the attacker then loses any monetary value he had with his bitcoin.

2. Given that the selfish miner has a 0.45 chance of finding a block, he has less than a 0.1 percentage of ever catching up from a 340 block deficit.

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

4. (a) We can still have a valid signature message pair, but it may not be the block that we had intended to be stored at position i. So technically the storage could duplicate multiple blocks and we would be unable to tell as we would not have a sense of order or location of the blocks.

   (b) We cannot be sure that they are the same, it could be the case that the storage provider just duplicated one block multiple times instead of actually storing each separate block. Therefore, you could be getting a different block than the ith block but the signature would still match the block.

5. (a) Verify: To verify, one needs to pull all blocks from storage and hash them and then compare the hash to the local hash. If they are equivalent, then it is properly verified. If they do not match, then there is an issue with verification. Read: In this case the read is to pull the required block from the storage, then verify to make sure the data is correct. Then read the block if the verification checks out. Write: To write, one needs to write to block i, then pull all records from the storage and concatenate them, then store the hash of the concatenation locally.

   (b) Since both reading and writing require one to recompute the hash of the concatenated blocks, the read and write operations both belong to the complexity class $\Theta(n)$. Note that the read itself is constant, but becomes linear with the verification step.

6. (a) Write: Each time we write a value into our storage, we need to write it such that the intermediary Merkle hashes are stored as well. We then store the new Merkle root hash. We also need to update all hashes in the Merkle tree. Verify: The verification process here is to get the sibling in the Merkle Tree and compute the hash together and then retrieve all the other hashes in the tree that we need to calculate the Merkle Root, if the root matches the locally stored match, we are verified. Read: To read we need to pull a file, then run compute the Merkle Root hash using the sibling and other stored intermediary hashes then check to make sure it matches the local Merkle root.

(b) Given that this is a Merkle Tree, the read and write operations belong to the complexity class $log(n)$. Note that the read is constant if you do not include the verification step.

7. (a) There are $\frac{24*60}{10} = 144$ 10 minute chunks in a day, which means 144 blocks found on average in a day. Since $\alpha = 0.15$, we would expect that $0.15 * 144 = 21.6$ blocks are expected to be found in a day.

   (b) Given time in minutes t, this would meant that $E(t, \alpha) = \frac{t}{10} * \alpha$.

8. The expected value for the number of blocks found in a day is: $\frac{144*\alpha*(1-\alpha)*L}{600}$

9. When selfish mining, you hold onto your entire block chain until the other network is behind by 1 block. In this case, the expected number of blocks orphaned is then then expected number of blocks you get per day selfish mining - 1. The expected value of the number of blocks you find while selfish mining is: $v = \alpha * (2 + \frac{\alpha}{1-2*\alpha})$ The proportion that you would find per cycle, is $\frac{v}{v+1-\alpha} * 144 - 1$.

10. Given bitcoin's anonymous nature, to what extent can someone be prosecuted for bitcoin related crimes?