April 18, 2016

Instructions. We will break into groups to discuss the following questions. Please think of as many solutions as you can. Be original! Maybe you will come up with something no one has thought of yet. Be prepared to talk about your solutions with the rest of the section.

Question 1  Hashchaining in practice  (15 min)

Bitcoin relies on hash chaining to prevent malicious actors from modifying events which occurred in the past. Recall from lecture this works by starting from some root $x_0$, and sending $x_{i+1} = H(x_i, \text{message})$.

(a) This is the first time a client connects to the bitcoin network. It asks for the most recent value in the hash chain, and receive two chains $C_1$ and $C_2$. How should the client decide which chain is the real chain?

(b) In practice, many transactions occur simultaneously (or nearly). What problems does this present to having a hash chain of transactions?

(c) Bonus problem: what might we be able to do to solve these problems?
Question 2  

Project 2 Rollback Prevention  

In Project 2, you were not required to prevent rollback attacks where the server reverted the state of a file to a previous value. In this problem you will design a scheme to prevent partial rollback attacks using the idea of a hash chain. (A partial rollback is one where the server rolls back the contents of one file, but not another.)

In this question you will design a scheme which prevents partial rollback attacks using a hash chain. As in Project 2, clients in your scheme can cooperate to work together against a malicious server, but may not keep state on the client.

(a) The simplest scheme has each user write to a shared hash chain that exists on the server. After client operation, the user who performed that operation adds a new node to the chain and uploads it to the storage server. What should each node contain?

(b) Can we create a more secure scheme if we allow clients to maintain state? What additional guarantees can we provide?
Question 3  Namecoin (20 min)

Another use of Bitcoin can be to create a decentralized domain name server. Users will still “mine” coins as in Bitcoin, and maintain a hashchain.

(Why might we want this? There are many reasons. The main idea is that it doesn’t rely on someone trusted third party to manage access to our domains. We don’t actually own our domain names with DNS – the third party does – and it can transfer the domain to someone else (either intentionally or not) or be a single point of failure.

(a) Namecoin has two general transaction types: register and name update. A register transaction is a statement of the form “I would like to own asdf.bit”. An update transaction is a statement of the form “Please point asdf.bit (which I own) to 1.2.3.4”. What information should be included in the contents of a register transaction? a update transaction?

(b) I want to try to make sure no one in the world can register domain names about cs161 (so that they don’t say anything bad). How could I try to go about doing this given the previous design? I can’t enumerate all possible domains that might say something bad. HINT: There is a race condition.

(c) How would you fix this issue, so that I can register any domain I want and no one else can try to stop me? HINT: You might want to use three transactions.