Department of Electrical Engineering and Computer Science

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

6.824 Fall 2001

Quiz

You have three hours to complete this quiz.

Write your name on this cover sheet AND at the bottom of each page of this booklet.

Some questions may be much harder than others. Read them all through first and attack them in the order that allows you to make the most progress. If you find a question ambiguous, be sure to write down any assumptions you make. Be neat. If we can't understand your answer, we can't give you credit.

THIS IS AN OPEN BOOK, OPEN NOTES QUIZ.

Please do not write in the boxes below.

1-4 (xx/28)	5-6 (xx/14)	7-7 (xx/7)	8-9 (xx/14)	10-11 (xx/14)	12-14 (xx/23)	Total (xx/100)

Name:

I Bayou

Mike Bloomberg's transition office must handle seat reservations for the upcoming mayoral inauguration ceremony. Three of his aides, A, B, and C, will be responsible for reserving seats. They would like to use a distributed database, but unfortunately the office is not wired for a network. Moreover, they do not want to use wireless networking because they've heard that 802.11b encryption was totally cracked.

The transition office decides to build a solution based on Bayou. They buy three laptops for each of the three aides. Each laptop keeps a Bayou replica database of seat assignments. A's laptop is designated the "primary" replica. From time to time, one aide will bring his laptop into another aide's office, and they will run anti-entropy over their serial ports. The laptops do not have synchronized clocks.

Recall that a Bayou write consists of three components: an update, a dependency check, and a merge procedure. For this application, all updates are insertion operations that insert a $\langle$seat, name$\rangle$ pair into the database. For example:

$$\{\text{insert}, \langle \text{seat 1}, \text{Charlotte Bloomberg}\rangle\}$$

The dependency check accompanying an update simply verifies that no record for that seat has yet been inserted in the database. Finally, all database writes contain a merge procedure to be run when the dependency check procedure detects a conflict. In this application, the merge procedure just looks for the next available seat:

```
mergeproc (str name)
{
  const int start_seat = 2;
  int s = start_seat;
  do {
    if (seat s is not reserved) {
      reserve seat s for name;
      return;
    }
    if (++s > MAX_SEAT)
      s = 1;
  } while (s != start_seat);
  /* there are no seats left */
}
```

Name:

1. [7 points]: Why does each Bayou write require an update and a dependency check in addition to the merge procedure? Doesn't the merge procedure suffice to apply an update to the database? Circle the best answer:

A. Yes, merge procedures suffice. Dependency checks are just a performance optimization.

B. Yes, merge procedures suffice. Dependency checks just save space by letting you discard the merge procedure as soon as a write becomes stable.

C. Yes, merge procedures suffice. However, a dependency check may reference records in a database the user does not have permission to modify. Since merge procedures contain arbitrary code, it's easiest just to prevent them from opening any database they should not modify.

D. No, you need both updates and merge procedures. In the absence of an update conflict, this gives you more flexibility (e.g., you can reserve some particular seat).

E. No, dependency checks must be separate from merge procedures. You need to run every uncommitted write's dependency check after running any write's merge procedure.

Answer: Answer **A** is correct. See the second paragraph of section 4.3.

2. [7 points]: Once the laptops are configured, Bloomberg is the first to make a reservation. He asks aide A to reserve one seat for his mother (Charlotte Bloomberg) and one seat for Diana Taylor. Aide A types a few keys and says that he has reserved seat 1 for Mrs. Bloomberg. He types a bit more and says he just got seat 2 for Ms. Taylor.

A few minutes later, President George W. Bush calls up aide B and asks for a seat at the inauguration. Aide B replies no problem, that he has reserved seat 1 for the president.

After hanging up, B brings his laptop to A's office and they run anti-entropy with each other. Both laptops then show the same seat assignments. What are the possible seat assignments they might show? Assume that only the actions described above have occurred.

Answer: Bloomberg in 1, Taylor in 2, Bush in 3. The reason is that laptop A is the primary, and immediately commits. Bush is moved to seat 3 when laptop B talks to A.

3. [7 points]: Suppose instead that Bloomberg had reserved the first two seats with aide C? What possible seat assignments might result in this case?

Answer: Three possibilities: Bloomberg Taylor Bush, or Bloomberg Bush Taylor, or Bush Bloomberg Taylor. Bush could occur anywhere since B and C don't have synchronized clocks. However, Bloomberg must appear before Taylor since both updates originated from the same laptop. Eventually both B and C will do anti-entropy with the primary on A, but that won't change the order, since B and C have already talked to each other and have identical state.

4. [7 points]: Rudolph Giuliani suddenly realizes that he will be relegated to the audience for the second half of the ceremony. He calls up aide B, right after B's anti-entropy session with A, and demands to be given seat 1. Aide B replies, "I'm sorry, my laptop won't let me do that."

Giuliani won't take no for an answer. Indignant, he replies, "You are very uncooperative, but I'm sure I can get aide C to assign me that seat."

Who is right: Aide B? Giuliani? Both? Neither? Who can end up in seat 1? Explain.

Assume only the actions from Problem 2 have occurred.

Answer: Both are correct. Aide B can only assign seat 4. Aide C has an empty database, and thus can assign seat 1. However, as soon as aide C talks to the primary, Giuliani will be moved to seat 4; thus only Bloomberg can end up in seat 1.

II SSL

Consider the SSL handshake protocol between a client (C) and a server (S) as depicted below:

A. C $\Rightarrow$ S : ClientHello (client_version, random, session_id, cipher_suites)

B. S $\Rightarrow$ C: ServerHello (server_version, random, session_id, cipher_suites)

C. S $\Rightarrow$ C: ServerCertificate (certificate_list)

D. S $\Rightarrow$ C: ServerHelloDone

E. C $\Rightarrow$ S: ClientKeyExchange($E(\text{pre\_master\_secret})_{\text{ServerPubKey}}$)

F. C $\Rightarrow$ S: ChangeCipherSpec (cipher_suite)

G. C $\Rightarrow$ S: [Finished(MAC(master_secret, all handshake messages)]

H. S $\Rightarrow$ C: ChangeCipherSpec (cipher_suite)

I. S $\Rightarrow$ C: [Finished, MAC(master_secret, all handshake messages)]

J. C $\Leftrightarrow$ S: [Data(plaintext)]

Notes:

- Messages between [and] are encrypted and authenticated.
- Step **C** transfers a list of the server's certificates, signed by certificate authorities. The client verifies one with the public key of the authority.
- Step **E** typically uses RSA to encrypt the pre_master_secret with the server's public key.
- The client and server calculate the cipher session keys after steps **F** and **H**; these session keys are used to encrypt and authenticate Data messages.
- Steps **G** and **I** allow the server and client to verify that none of the previous messages were tampered with.

The purpose of the protocol is to prove the server's identity to the client, and to set up an encrypted and authenticated communication channel between client and server.

5. [7 points]: Suppose a bad person has just subverted DNS so that it returns the IP address of the bad person's server to you when you look up www.amazon.com. The bad person's server has a copy of www.amazon.com's certificate and public key (which are public knowledge). Amazon and the certificate authority have successfully kept their private keys secret. The bad person wants to learn your credit card number by pretending to be Amazon. You ask your browser to view `http://www.amazon.com`, but it actually connects to the bad person's server. That server performs the server side of the SSL protocol, sending www.amazon.com's correct certificate in step **C**. Describe in a few sentences why this won't lead to the bad person being able to decrypt the Data messages your browser sends and thus reconstruct the original plaintext.

Answer: Your browser will send the pre-master-secret encrypted with Amazon's correct key (in step **E**). Thus the bad person cannot generate the session keys, and cannot decrypt the encrypted Data messages.

6. [7 points]: Suppose now that the bad person has their own completely valid certificate, obtained from the certificate authority in the ordinary way. Once again they intercept your connection to www.amazon.com. This time they provide you with their own certificate in step **C** instead of Amazon's. What is the earliest step (if any) in the protocol at which you or your browser have enough information to be virtually certain that you are talking to an imposter? Briefly explain.

Answer: Step **C**. The DNS name in the certificate won't match the server name in the URL the user (you) asked to browse; the browser checks this.

III Taos

Consider the statement labeled (3) in Section 2.2 of the Taos paper, which reads:

$$(K_{ws}|K_{bob}) \textbf{ says } C_{bob} \Rightarrow (K_{ws} \textbf{ for } K_{bob})$$

7. [7 points]: The statement corresponds to a signed certificate. Which key signs it? The possibilities are K_{vax4}, K_{ws}, K_{bob}, and C_{bob}.

Answer: K_{ws}. K_{vax4} is only known to the boot code. K_{bob} is only known to Bob; the workstation discards it after generating statement (2). C_{bob} is the channel key; we're trying to make a certificate that says the channel speaks for Bob, so there's no point in C_{bob} signing the certificate.

IV Freenet

The FBI wants to know who is fetching a particular document stored on Freenet. The FBI performs a Freenet fetch operation for the document, and notes the node reported as the source in the reply. They then ask the owner of that node to give the FBI a copy of all future traffic into and out of that node; the node's owner agrees. The FBI records all the fetch requests that arrive at that node for the document in which they're interested; the record includes the source IP address of the packets containing each request.

8. [7 points]: Is there a specific circumstance in which the FBI's record would contain the identity (IP address) of the real requester? If yes, describe it briefly.

Answer: Yes. If the original requestor happened to send the request first to the node the FBI is watching, the original requestor's IP address will appear in the FBI's log.

9. [7 points]: List three reasons why a host successfully requesting the document could remain anonymous, despite the FBI snooping.

Answer: First: except for the case in the previous answer, fetch requests don't contain the originator's identity. Second: perhaps the document was originally inserted on multiple hosts, and thus is available on a node other than the one watched by the FBI. Third: the document may be available at other nodes because it is cached along the path of other fetches, including the fetch performed by the FBI.

V CFS–R/W

Ben thinks CFS is cool but he would like to have a read/write file system. He believes he can make CFS R/W with only a few changes. His plan is to simply reinsert the whole file system each time a file is modified, using the same public key for the root-block. You may wish to glance at Section 3.1 of the CFS paper to remind yourself what a CFS file system looks like.

For each of these questions, assume that the file system starts out with just one directory (the root directory), with 100 files in that directory, and with each file containing 4 8192-byte blocks. Assume each file's contents are different (i.e. the files are filled with random bytes). Ben has turned off DHash caching and replication, so there is only one copy of each block.

10. [7 points]: Suppose Ben changes one byte of one file in his file system. After he does this, roughly how many new blocks will be stored in DHash, that were not stored before? Which blocks are these (i.e. what role do they play in the file system)?

Answer: Roughly 5: the file data block, the file's i-node, the root directory contents, the root directory's i-node, and the root-block.

11. [7 points]: Ben would like multiple people on different computers to be able to modify the same CFS–R/W file system. He decides this is easy; it is enough if all the participants have a copy of the root-block's private key, so any of them can change it.

After the system is deployed, two users, fdabek and rtm, try to create new files, x and y, at approximately the same time. Later Ben attempts to access the files. Other than these actions there are no events that affect the file system. For each of the outcomes listed below, say whether it could or could not occur, and why.

A. Ben sees both x and y. **Yes**, if fdabek finishes updating before rtm starts.

B. Ben sees neither x nor y. **No**, since (given that nothing else happened) at least one of fdabek or rtm must update the root-block.

C. Ben sees one file, but not the other. **Yes**, if both fdabek and rtm read the old root-block at the same time, so their updates don't reflect each other's change.

D. Ben sees just file x, but with the contents rtm wrote to file y (or the other way around). **No**, except if their file contents are different but hash to the same value (very unlikely).

VI Harp

Consider the operation of a system just like Harp, but with a total of five servers: the designated primary (p), two backups (b_1 and b_2), and two witnesses (w_1 and w_2). Assume that clients do not ever re-send requests (this is to keep the situation simple).

Suppose the following sequence of events occurs:

A. Client c_1 issues a request to create a new file named x. p receives the request and sends a copy of the request to the two backups.

B. Both backups receive the request to create x. Both send acknowledgments to p, but the network drops the acknowledgment from b_2 so p only sees the ACK from b_1.

C. Client c_2 issues a request to create a new file named y. p receives the request, and tries to send a copy to the two backups.

D. Just before p sends the request, something goes wrong with the network. As a result, p and b_1 can only talk to each other, and b_2, w_1, and w_2 can only talk to each other. Thus only b_1 receives the request to create file y.

E. b_2, w_1, and w_2 form a new view. b_2, acting as primary, receives a request from client c_3 to create a new file named z. Eventually b_2 sends a reply to c_3 indicating that the creation completed successfully.

F. b_1 and b_2 both crash suddenly due to hardware failures that permanently destroy their disks and memory.

G. The network problem that isolated p is fixed, so p rejoins the system. p has been alive the whole time, with disk and memory intact.

12. [7 points]: It's only legal to form a new view if the participants in the new view are guaranteed to know about all committed operations in previous views, including operations that might have committed while the system was partitioned. In step **E**, what reasoning do b_2, w_1, and w_2 use to convince themselves that a new view is legal?

Answer: The servers must ensure three things. First, that there is at most one view at a time, so they don't miss commits performed in a parallel view. Second, that they know for sure what the previous view was (so they don't miss an entire view). Third, that they they have a server with all the commits from the previous view.

Assembling a majority (3 servers) is enough to satisfy the first property.

During the previous view change, each server in the view (including witnesses) wrote the view number to stable storage. Thus the second property can be ensured if a majority of servers with intact preview view numbers is assembled. One of them must have been among the majority in the previous view. The highest view number among the majority must be the previous view's number.

Name:

An operation might have committed in the previous view with as few as a majority (3 servers) knowing about it, in the sense of having written the operation to the log in memory. The third property can be ensured if enough servers survive *with intact in-memory logs* from the previous view to ensure that no committed operation will be missed. Three nodes *known to have been in the previous view* are always enough; for example, if two are witnesses and one is a backup, that implies that that backup saw every commit, since the witnesses do not log or participate in commits.

13. [8 points]: After step **G**, do p, w_1, and w_2 have enough information to convince themselves that they can legally form a new view? Why or why not?

Answer: We don't have a majority of servers from the previous view; we only have two, not three. So it might seem that we cannot form a new view. In fact, the servers can convince themselves that in this case a majority is not needed. p, since its memory is intact, knows that it did not participate in the previous view, so it knows that there were at most four servers in the previous view. That means that every committed operation must have been seen by either w_1 or w_2. This is only useful if both w_1 and w_2 were promoted; but they were, and they know they were. Thus between them they know about every committed operation in the previous view.

Since p has intact memory, it knows it is up to date as of the view before the most recent view, so it knows it can become fully up to date using the merged logs from w_1 and w_2.

14. [8 points]: What is the final fate of each of the three file creation requests (x, y, and z)? For each one, state whether the file must exist, must definitely *not* exist, or could legally either exist or not exist. State your reasons for each in a sentence or two. Assume that the files did not exist before step **A**, and that no client requests involve those files except as mentioned above.

Answer: x must exist; b_2 received and acknowledged it, so it has to assume that p might have received a majority of ACKs and committed the operation. y probably doesn't exist; it was not committed by p, and b_2 didn't know about it. z must exist; b_2 replied to the client, so it must have received ACKs from the promoted witnesses, so p (in the view formed in step **G**) learns about z from one of the promoted witnesses.

End of Quiz

Name: