Midterm Examination
CS 236
On-Line Program
Computer Security
Spring 08

Answer all questions. All questions are equally weighted. The test is open book/open notes.

1. Consider the example of the Marx Brothers using digital signatures for a contract as discussed in lecture 5. Here, Groucho passed a contract with 1000 clauses to Chico, also providing a digital signature based on public key cryptography that covered the entire 1000 clauses. But Groucho knows that Chico frequently dislikes some clauses in a contract and will reject some of them. So Groucho plans instead to individually sign each clause and send the contract to Chico with 1000 signatures, one for each clause. Chico can then choose the clauses he likes and sign them himself, using the same digital signature procedure based on a secure hash and Chico's private key. (Assume Groucho and Chico already know each other's public keys. They are brothers, after all.) Chico will send the altered contract along with the set of his signatures for the chosen clauses back to Groucho. Will this procedure allow Groucho and Chico to agree on a contract that contains only clauses they both agreed upon? Will it allow them both to know that they have agreed on the same set of clauses? None of the Marx Brothers trust each other. What if Groucho tries to cheat? What if Chico tries to cheat? What if Harpo slips in between them and tries to cheat them both by tampering with either Groucho or Chico's messages?

2. Consider a networked system where access control is provided by capabilities. To prevent forging of capabilities, they are cryptographically created by trusted authorities, and their use requires checking of the cryptography. Further, to ensure that users can't trade capabilities at will, the cryptography ties the identity of the user to the capability. The designers of the system want one possible capability-based access right to be the ability to give others access to the protected resource. In other words, some capabilities allow the holder to transfer their read/write/execute/append/etc. rights to another user. Other capabilities do not allow such transfer of rights.

   User A holds a capability for resource X allowing read and allowing user A to transfer read access to other users. User A now wants user B to have read access to resource X, but not be able to transfer read access to anyone else. What steps would the system take to perform this operation? What kind of information would the system require to be able to create the necessary capability? What information would have to be available to the reference monitor of resource X to determine whether read requests from users A and B should be satisfied?

3. One thing TPM hardware could be used for (with suitable software support) is to allow a remote machine to run code on our machine with useful security

guarantees. For example, a secure environment might want to insist that our machine run a particular virus scanning program while we are connected to its network, and might demand that the results of the scans be reported to its central facility. The remote user would want assurances that the piece of code he submitted to be run on our machine really was running, and really was the right piece of code. We want assurances that the remote user's code is limited in its functionality and in what it can change and reveal about our machine.

What is the "suitable software support" that the operating system must have to make this possible, beyond the features inherent in the TMP hardware?