Answers to Sample Final Questions

1. a. A public key cryptosystem is a cryptosystem with two keys, one of which is known to everyone (the public key) and the other of which is known only to one person (the private key).
   b. A challenge-response refers to a mechanism in which the client is challenged to demonstrate knowledge of a secret to a server. The client's response must demonstrate to the server that the client knows the secret.
   c. A computer worm is a program that replicates and copies itself from system to system.
   d. End-to-end encryption occurs when a message is enciphered, and sent to its destination, and then deciphered at its destination. It is enciphered as it transits the network, leading to the term "end-to-end encryption."
   e. A web cookie is a construct that encapsulates the state of a session between a browser client and a web server.
2. ACLs correspond to the columns of an access control matrix, and C-Lists correspond to the rows of an access control matrix.
3. Suppose we wanted to revoke subject *s*'s access rights *r* to a file *f*. If the system used access control lists, one would revoke the access by going to *f*'s ACL and deleting *s*'s rights *r*. If the system used capability lists, one would revoke the access by going to *s*'s capability list and remove the capability that gives *s* the *r* rights over *f*.
   With ACLs, it is trivial to remove all rights to a given object from all subjects. With C-lists it is much more difficult. For example, suppose we want to remove all users' rights to read a file *f*. We need traverse only one ACL, but will need to traverse every process' C-List to see if that process has read rights over *f*. Conversely, with C-Lists, it is easy to remove all rights to all objects from a given subject. With ACLs, it is much harder. For example, we want to remove a subject *s*'s rights to all objects. We need traverse only one C-List, but would need to traverse every file's ACL.
4. a. The process could access exactly those objects in its security/integrity class. Reading an object in another class would violate either the simple security property (if the object's security level dominates the process' security level) or the integrity *-property (if the process' security level dominates the object's security level). Similarly, writing an object in another class would violate either the simple integrity property (if the object's integrity level dominates the process' security level) or the *-property (if the process' security level dominates the object's security level).
   b. In practise, this limits the sharing that processes can do. As most computing today requires communication and shared resources, the above scheme would inhibit effective use of computers.
5. The primary consideration was the standard for signing certificates. In the PEM hierarchy, each Certification Authority (the entity that issues certificates to individuals) is legally bound to comply with the policies for issuing certificates that its Policy Certification Authority states. Therefore, when one gets a certificate, one can determine the methods used to authenticate the subject (the entity to whom the certificate was issued) by examining the document describing the standards that the PCA requires the CA to use. The Web of Trust, on the other hand, leaves the level of trust that a certificate signer (the PGP equivalent of a certificate issuer) has in the subject to the discretion of the issuer, and the standards that the signer uses need not be written down, or even be consistent among certificates signed by that certificate signer. Hence one cannot determine from the certificate how the signer validated the subject. This makes trust in identity more problematic than for certificates issued under the PEM hierarchical model.