13.2 §13.5 Replicated KDCs explains that the KDC database isn’t encrypted as a unit. Rather each user’s master key is independently encrypted with the KDC master key. Suppose replication were done with a simple download (i.e., no cryptographic integrity check is performed). How could a bad guy who is a principal registered with a KDC impersonate Alice, another principal registered with that KDC? Assume he can see and modify the KDC database in transit, but that he does not know the KDC master key.

All he has to do is replace Alice’s encrypted key with his encrypted key.

13.3 Why is the authenticator field not of security benefit when asking the KDC for a ticket for Bob, but useful when logging into Bob?

The authenticator in the TGS.REQ is used to prove knowledge of the session key. But without knowledge of the session key, the ticket requester can’t decrypt the CREDENTIALS field in the TGS.REP to get the ticket. The authenticator in the AP.REQ is used to prove knowledge of the shared key. This is important because further messages in the session may be unencrypted, so there is no other opportunity for authentication.

15.1 Referring to Figure 15-1, which CAs must B/Y/Z/A/C trust in order to find a path to A/C/Y? B/Y/Z/A and A/C.

15.5 Why must a CRL be reissued periodically, even when no new certificates have been revoked?

Otherwise someone could post an old CRL, from before they were on the revocation list.

15.6 If there is a revocation mechanism, why do certificates need an expiration date?

To limit the size of the CRL. Also (and in practice the real reason), so that entities that charge for certificates can get a continuing revenue stream.

15.8 Why is it important in a good-list revocation scheme to keep hashes of the valid certificates, rather than just their serial numbers?

Because someone might have issued a bad cert with the same serial number as a good cert, and people might not know of the existence of the bad cert.
16.1 Talk about the properties of each of the following protocols, such as perfect forward secrecy, escrow foliage against passive attacks, escrow foliage against active attacks, identity hiding, perfect forward secrecy for identity hiding. Assume private encryption keys are escrowed and private signature keys are not escrowed.

- Protocol 16-2.

**PFS, escrow foliage against passive attacks, escrow foliage against active attacks (unless signature key is escrowed), no identity hiding.**

- A modified form of Protocol 16-2 in which the first two messages are encrypted with the other end’s public key rather than signed by the transmitter’s private signature key. So in message 1 Alice sends \{“Alice”, \(g^a \mod p\)\} encrypted with Bob’s public key, and Bob in message 2 sends \{“Bob”, \(g^b \mod p\)\} encrypted with Alice’s public key.

**PFS, escrow foliage against passive attacks, no escrow foliage for active attacks, identity hiding, no PFS for identity hiding.**

- Protocol 16-4.

**PFS, escrow foliage against passive and active attacks (assuming signature key not escrowed), identity hiding, PFS for identity hiding, active attacker can discover Alice’s identity.**

- Protocol 16-9, where Alice and Bob share a secret key \(S\).

**PFS, escrow foliage against passive attacks, no escrow foliage against active attacks, no identity hiding.**

- Each side sends a nonce encrypted with other’s public encryption key, resulting key is \(\oplus\) of two nonces

**no PFS, no escrow foliage, no identity hiding**

- Assume Alice and Bob share a secret \(S\). Design a protocol in which they can do mutual authentication and establish a shared secret with PFS. Can it be done without Diffie-Hellman or any other form of public key cryptography?

*The protocol above ("protocol for Homework Problem 1") will work. And it cannot be done without Diffie-Hellman or some other means of PFS. A method of using RSA keys is to have one side create an ephemeral key pair, and send the public key to the other side, integrity protected with the shared secret key, and have the session secret (with which they’d mutually authenticate) be a function of a random number sent encrypted with the ephemeral public key and the shared secret key.*

- Protocol 16-2, but with each side deterministically generating the Diffie-Hellman private numbers as described in §16.4 *PFS-Foilage* from a seed given to the client machine and escrowed at the server machine.

**no PFS, no escrow foliage, no identity hiding.**
16.12 Suppose a stateless cookie mechanism is used by Bob, and suppose he changes his secret periodically. What can he do to assure a connection attempt will succeed even if he changed his secret between the time the initiator asked for the cookie and returned it (assuming the initiator doesn't wait too long before returning the cookie)?

He can save the previous secret for a reasonable overlap period, and for the overlap period after he's changed his secret, accept a cookie based on either secret. He can either try both secrets or have a bit in the cookie that specifies which secret should be used.

17.6 Referring to Figure 17-2, suppose A and B already have an IPsec SA between them and are using ESP. What would be the advantages/disadvantages of having F1, in the case where there's already an ESP header, merely forwarding the packet to F2 without doing a second encryption?

It might save some computation time, avoiding the second encryption and decryption. Both F1 and F2 would have to be configured to allow this, and they would lose the ability to assure that A and B are communicating with systems on the private network (e.g., A could be on the public network communicating with B using ESP). The security of the communication would depend solely on the strength of the authentication mechanisms and cryptographic algorithms selected by A and B (which might not be as well managed as the firewalls).

17.7 Referring to Figure 17-2, assume that A and B are using IPsec in transport mode, and F1 and F2 have established an encrypted tunnel using IPsec. Assume A sends a TCP packet to B. Show the relevant fields of the IP header(s) as given to A's IPsec layer, as transmitted by A, as transmitted by F1, and as received by B.

As given to IPsec layer: SOURCE=A, DESTINATION=B

As transmitted by A: SOURCE=A, DESTINATION=B, PROTOCOL=ESP (or AH), ESP header NEXT HEADER=TCP

As transmitted by F1: starting from the outer header, SOURCE=F1, DESTINATION=F2, PROTOCOL=ESP, ESP header NEXT HEADER=IP; inner IP header exactly as transmitted by A

As received by B: exactly as sent by A.