Answers to Practice Questions for Exam 1 (Crypto Basics)

Answer 1-Crypto:

a) The three basic requirements that we discussed in class are:

1. *Cryptosystem security*: Roughly speaking, it must be infeasible for an eavesdropper to compute x from y without knowing K_{AB}. Moreover, it must be infeasible to infer K_{AB} even if the eavesdropper has access to many ciphertexts exchanged by Alice and Bob.

2. *Endpoint security*: Alice and Bob must store K_{AB} securely (so that no one can steal it) and use it in a secure computing environment (so that the eavesdropper cannot simply obtain the plaintext before it is sent or after it is received, without having to "break" the cryptosystem).

3. *Secure key management*: Alice and Bob, typically with the assistance of a *key-distribution center*, must obtain a unique key K_{AB} and maintain its security until it is retired.

.

b) Secure key management cannot be done at Internet scale. In particular, e-commerce websites need to communicate securely with potential customers whom they have never "met" and with whom they have not been able to establish shared keys.

Answer 2-Crypto:

a) Public-key certificates are *signed* (name, public key) pairs. The problem addressed is that an impostor or man-in-the-middle, say Manny, could generate a key pair (PK_M, SK_M) and then impersonate Alice by publishing the directory entry (Alice, PK_M). Subsequently, if someone sends a ciphertext to Alice that was encrypted using the encryption key PK_M, Manny could intercept it and decrypt it, because it is he, not Alice, who knows the corresponding decryption key SK_M. To prevent this, we require that public-key directory entries be signed by *certifying authorities* or CAs. Instead of simply generating her key pair and publishing her public key, Alice takes her public key PK_A to a CA, along with proof that she is indeed Alice. The CA verifies that her identification documents are valid and then signs the (name, key) pair (Alice, PK_A). The entry that is published in the directory is

$$\text{Alice}, PK_A, \sigma_{Alice,CA},$$

where $\sigma_{Alice,CA}$ is the CA's signature on (Alice, PK_A). Someone who wants to use PK_A must have a trustworthy copy of the CA's verification key, but we have reduced the problem of needing trustworthy copies of PK_A, for *every* user A of the public-key system, to the much smaller-scale problem of needing a trustworthy verification key (or keys) for one (or a small number of) CA(s).

b) If Bob wants to send a long traffic stream to Alice, he can obtain her public key PK_A (and a certificate that allows him to verify that it is hers), generate a shared key K_{AB} for a (fast) symmetric-key cryptosystem, and send her $E(K_{AB}, PK_A)$, where E is the agreed-upon, (slow) public-key encryption function. Alice can decrypt this message (using SK_A, which only she knows) to recover K_{AB}. Bob can then encrypt his long traffic stream using the symmetric-key cryptosystem and the key K_{AB} that only he and Alice know. This basic technique is an essential component of the SSL/TLC protocol for encrypting web traffic.

Answer 3-Crypto:

a) If Alice (A) is a user, then her signing key sk_A consists of (d, n), where $n = pq$ is the product of two equal-length, large primes p and q, and d is an integer that is relatively prime to $\phi(n) = (p-1)(q-1)$. Her verification key pk_A consists of (e, n), where $ed \equiv 1 \bmod \phi(n)$. In order to sign a message M, where $M \in [0, n-1]$, Alice computes $\sigma_A = M^d \bmod n$. In order to verify this signature, anyone can look up Alice's (public) verification key and check that $M = \sigma_A^e \bmod n$.

b) At the very least, multiplication of equal-length primes must be one-way; that is, while it is easy to compute $n = pq$, it is hard to compute p and q given n, even if one is told that n is the product of two equal-length primes. Note that we actually need it to be easy to *find* large p and q of the same length and verify that they are prime, not just that it is easy to multiply them once we have them.

More generally, we need it to be hard to find e if one is given (d, n) but is not given the factors p and q.

c) No

Answer 4-Crypto:

a) See Section 2.4 of Schneier's book (`http://proquest.safaribooksonline.com/book/-/9780471117094`).

b) Because public-key signature and verification functions are often slow, one typically computes a one-way hash, say t, of a long document M and then signs t, which is much shorter than M. To verify a signature, one must first check that t *is* indeed the output that the one-way hash function produces on input M and then verify the signature on t.