2011

Question 1

a

- $p = 3$
- $q = 11$
- $n = p \times q = 33$
- $\phi(n) = (p-1)(q-1) = 20$
- Choose e such that e and n are relatively prime and $e < n$
  - $e = 3$

$$d \equiv e^{-1} \mod \phi(n)$$

- $d \equiv 3^{-1} \mod 20$
- $\gcd(20, 3) = 1$
  - $20 = 3 \times 6 + 2$
  - $3 = 2 \times 1 + 1$
- Backtrack
  - $1 = 3 - 2 \times 1$
  - $1 = 3 - (20 - 3 \times 6) \times 1$
  - $1 = 7 \times 3 - 20$
- $3^{-1} = 7$

$d \equiv 7 mod 20 \equiv 7$

- Ciphertext = Plaintext$^e \mod \phi(n)$
- Plaintext = Ciphertext$^d \mod \phi(n)$

Letter	Value	Encrypted	Decrypted	Letter
h	8	12	8	h
e	5	5	5	e
l	12	8	12	l
l	12	8	12	l
o	15	15	15	o

b

Alice establishes a communication channel with X and asks for their certificate to verify his public key. If X provides a certificate signed by another CA Y, and Alice doesn't know Y, she repeats the process. This continues until she knows a CA's public key. Alice recursively verifies each certificate, also checking the CRL. After verifying Bob's public key, she picks a nonce and sends it to him with his public key. If he can send it back in plaintext then she's convinced it's Bob.

c

1. Alice creates a message
2. She calculates the HMAC using the shared symmetric key and the message
3. She sends the message to Bob appended with the HMAC
4. Bob receives the HMAC and message
5. He calculates another HMAC of the message using the symmetric key
6. He compares both HMACs to see if they match

A HMAC is not used for encryption, it is simply sent alongside the message to verify data integrity using a symmetric key. RSA uses key pairs, where the hashed message is signed using the private key an verified with the public key.

A HMAC is much faster to compute, and more secure (even if the hash function is broken) because a symmetric key is an arbitrary combination of bits whereas a key pair have to follow a set of rules.

d

1. Choose a large prime p and integer α such that $\alpha \in \{2, 3, \ldots, p-2\}$
   - α must be a generator of group $\mathbb{Z}_p^*$
2. Alice chooses value x and computers α^x and sends this to Bob
3. Bob chooses value y and computes α^y and sends this to Alice
4. Alice computes $(\alpha^y)^x \mod p$
5. Bob computes $(\alpha^x)^x \mod p$
6. Both Alice and Bob now share a key

MITM Attack:

1. Alice computes $(\alpha^z)^x \mod p$ for messages to who she thinks is Bob
2. Trudy computes $(\alpha^x)^z \mod p$ for messages to Alice
3. Bob computes $(\alpha^z)^y \mod p$ for messages to who he thinks is Alice
4. Trudy computes $(\alpha^y)^z \mod p$ for messages to Bob

Trudy has now established seperate connections to Alice and Bob, who think they've established a connection with each other.