Question 1

(a) Doing a signature with RSA alone on a long message would be too slow (presumably using cipher block chaining). Suppose we could do division quickly. Would it be reasonable to compute an RSA signature on a long messages by first finding what the message equals, mod n, and signing that?

(b) Suppose Alice sends a message to Bob by representing each alphabetic character as an integer between 0 and 25 (A $->$ 0, B $->$ 1, ... , Z $->$ 25) and then encrypting each number separately using RSA with a large e and a large n. Describe an efficient attack against this encryption method.

Alice's public key $<e, n>$ is known by everyone including the attacker. Since the attacker knows Alice's public key, they can create a ciphertext c of a message m by calculating m^e mod n. The attacker can create a table with 26 entires, mapping each of the 26 ciphertexts to the corresponding message (0, 1, ... , 25).

Now the attacker can listen between Alice and Bob, intercept Alice's ciphertext for each character of the message, decode it using the table, map digits back to characters, and generate Alice's message.

This is assuming the messages are not padded before encryption.

(c) Show that $(x^c$ mod $n)^d$ mod $n = x^{cd}$ mod n.

We will start with the left hand side and show that it is equivalent to the right hand side

$$(x^c \ mod \ n)^d \ mod \ n$$

Modular arithmetic produces the remainder of some integer x when divided by n. This is similar to subtracting Kn from x for some integer K. For example 15 mod 4 $= 3$ and $15 - (3 * 4) = 3$. With this we can rewrite the LHS of our original equation

$$(x^c - Kn)^d \ mod \ n$$

Next we will expand the expression

$$(x^{cd} - ....) \ mod \ n$$

Where we will have x^{cd} and each other term in the expanded expression will be some multiple of n as a result of the Kn. Since each of these terms is a multiple of n, they will have no result on the $mod \ n$ operation.

$$x^{cd} \ mod \ n$$

We have arrived at the RHS side of the original equation, show their equality.

Question 2

(a) Encrypting the Diffie-Hellman value with the other side's public key prevents the person-in-the-middle attack. Why is this the case, given the attacker can encrypt whatever it wants with the other side's public key?

During a Diffie-Hellman key exchange, two parties (Alice and Bob) arrive at a shared private key that no one else knows. A person-in-the-middle attack is possible by an intruder (Trudy) sitting between Alice and Bob. Trudy maintains a shared public key with Alice K_{AT} and a shared public key with K_{BT} as outlined in the textbook on page 168.

This attack can be prevented by encrypting the Diffie-Hellman values $T_A = g^{S_A}\ mod\ p$ and $T_B = g^{S_B}\ mod\ p$ with the other side's public key. This disrupts Trudy's plan as she needs T_A to compute K_{AT} and she needs T_B to compute K_{BT}.

$$K_{AT} = T_A^{S_T}\ mod\ p$$
$$K_{BT} = T_B^{S_T}\ mod\ p$$

Where S_T is a number of Trudy's choosing. Trudy can't recover T_A because it's encrypted with Bob's public key (she would need Bob's private key) and likewise T_B is encrypted with Alice's public key.

(b) Suppose the public Diffie-Hellman key of Bob is $T_B = g^{S_B}$ mod p. How does Alice send a secret message m using the Diffie-Hellman scheme to Bob? [Assume that (g, p) are known to Alice and Bob ahead of time.]

Using the known and shared g and p, Alice first chooses a random number S_A and computes her public Diffie-Hellman key $T_A = g^{S_A}$ mod p. Next using Bob's public key T_B, Alice computes the shared key $K_{AB} = T_B^{S_A}$ mod p. This key is only obtainable by Alice and Bob, so Alice uses it to encrypt m using a secret key encryption technique. Alice sends $K_{AB}\{m\}$ and T_A to Bob.

To recover the message, Bob computes the shared key himself $K_{AB} = T_A^{S_B}$ mod p. With this, Bob can decrypt $K_{AB}\{m\}$ using the same cryptographic algorithm Alice used and recover the message m.

(c) Let there be n people in a group. Each person in the group wishes to establish a secret with every other person in the group. Let us assume that each person can send broadcast messages to reach all the other members of the group. Show an efficient Diffie-Hellman exchanges that allows each member of the group to establish a secret with every other member of the group. How many broadcast messages does your scheme use?

The public prime base g and public prime modulus p are known to all n people in the group. Each person generates their own random number S_n to use as a private key. Using this private key, they generate a public key $T_n = g^{S_n}\ mod\ p$. Each person in the group takes turns broadcasting their T_n for every other person in the group to hear. Each person computes $n-1$ shared keys (one for each other person in the group) using the other person's public key and their own private key $secret_n\ = T_n^{S_{self}}\ mod\ p$.

Now each member of the group has $n-1$ secrets, one with each other member of the group. This was accomplished by sending out n broadcast messages.

Question 3

(a) Design your own zero knowledge proof system for interactive authentication using the ideas presented in Section 6.8 of the textbook. You must present arguments to show that your scheme is secure. (You can find a long list of NP-complete problems in the book by Michael Garey and David Johnson.

(b) Transform your scheme into a zero knowledge signature scheme and also show that your signature scheme is secure.

Question 4

(a) Briefly describe the properties of the wireless channel between a pair of wireless nodes that enable these nodes to extract a symmetric/secret bit sequence?

(b) What is the similarity between a secret key extraction presented in this paper and the Diffie-Hellman cryptosystem?

(c) Does this method of secret key extraction from the wireless channel between Alice and Bob provide perfect forward secrecy? Explain briefly.

(d) Why is it not useful to extract secret bits from wireless channels in static environments? How can an adversary make Alice and Bob agree upon a predictable key pattern?