Problem 1

Suppose two people (Bob and Eve) are assigned the same RSA modulus $N$. Someone (say, their boss Alice) selects $p$ and $q$ and computes $N$ while keeping $p$ and $q$ secret. Then, Alice computes two key-pairs: $(e_a, d_a)$ and $(e_e, d_e)$ and gives the first one to Bob and the second one – to Eve. Recall that $e_e \cdot d_e = 1 \mod \phi(n)$ and $e_b \cdot d_b = 1 \mod \phi(n)$.

Now, suppose Alice sends a secret message $M$ to Bob by encrypting it: $C = M^{e_b} \mod n$. Eve sees this encrypted message. Show how Eve can compute $M$ from $C$. In fact, Eve can compute $d_b$ as well!!!

Hints:

• Start by showing that, knowing $e_a$ and $d_a$, Eve can compute a multiple of $\phi(n)$.
• Proceed by showing that, knowing a multiple of $\phi(n)$, Eve can recover $d_b$ from $e_b$.
• At this point decrypting $C$ is trivial...

Problem 2

Consider the following 2 ways to construct a MAC (Message Authentication Code):

$MAC_x(data) = h(K || data)$
$MAC_y(data) = h(data || K)$

Here “——” denotes concatenation. $h()$ is a collision-resistant strong hash function that operates on a sequence of n-bit blocks and produces a n-bit output. Assume $K$ is an n-bit secret and data is $p \cdot n$ bits.

Which one is more secure: $MAC_x$ or $MAC_y$? Assume Alice and Bob share $K$. Eve is listening, as always and sees packets of the type:

packet, MAC(packet)

where MAC is either $MAC_x$ or $MAC_y$. Comment on why $MAC_z(data) = h(K, data, K)$ is better than $MAC_x$ and $MAC_y$.

Problem 3

Consider the following secret sharing scheme:
We take an n-bit secret $K$ and split it into $t$ sub-secrets: $S_1, ..., S_t$ where each $S_i$ is $n/t$ bits long. Each party, $P_i$ receives a share, $S_i$.

Then, to reconstruct $K$, the parties simply concatenate their shares and obtain $K$.

Is this a good t-out-of-t scheme? Evaluate it... Is it better then the one presented in class? Explain your answer well.

Problem 4

Suppose we modify the Diffie-Hellman key exchange method as follows:

1) Alice generates random $a$
   Then, Alice sends to Bob: $g^a \mod p$

2) Bob generates random $b$, computes $g^b \mod p$
   Then, Bob sends to Alice: $g^{ab} \mod p$

   Alice computes $(g^{ab})^{a^{-1}} \mod p = g^b \mod p$ The secret key that Alice and Bob share is $K = g^b \mod p$

Formally show (prove) that this method is as secure as the original Diffie-Hellman method discussed in class and in the book.