Solutions to Problem Set 2

Problem 2-1. DES

It is enough to prove the statement for one iteration of DES. Look at figure 3.15 in the textbook. The first thing to notice is that if we complement *both* message and key the inputs to the S–boxes will be the same as in the uncomplemented case. This is because the bit-wise $\oplus$ before the S–boxes nullifies the effect of the complementation ($a \oplus b = \bar{a} \oplus \bar{b}$)

So if the inputs to the S-boxes are the same the outputs are the same as well. But now the outputs are XORed with the left part of the message which is complemented as well so it will flip all the bits ($a \oplus b = \overline{\bar{a} \oplus b}$)

Problem 2-2. Message Authentication Codes

(a) Consider the following attack. Eve asks for two distinct one-block messages m_1 and m_2 and she receives

$$t_i = \mathrm{MAC}_k(m_i) = f_k(m_i)$$

Now consider the following two–block message $m = m_1 \circ x$ where

$$x = t_1 \oplus m_2$$

Now

$$\mathrm{MAC}_k(m) = f_k(f_k(m_1) \oplus x) = f_k(t_1 \oplus t_1 \oplus m_2) = f_k(m_2) = \mathrm{MAC}_k(m_2)$$

So Eve has obtained the MAC of m a message she has not asked for.

(b) Let NEWMAC denote the MAC obtained by appending the length of the message at the end and then calling the CBC-MAC from above

$$\mathrm{NEWMAC}_k(m = m_1 \circ \ldots \circ m_b) = \mathrm{CBCMAC}_k(m \circ b)$$

where by b we denote the ℓ-bit block encoding the number b.
Let m_1, m_2, m_3 be three distinct one-block messages. Eve asks for the MACs of m_1 and m_2 and she gets back

$$t_i = \mathrm{NEWMAC}_k(m_i) = f_k(f_k(m_i) \oplus 1)$$

She then asks for the MAC of the message $m = m_1 \circ 1 \circ m_3$ and she gets back

$$t = \mathrm{NEWMAC}_k(m) = f_k(f_k(f_k(f_k(m_1) \oplus 1) \oplus m_3) \oplus 3) = f_k(f_k(t_1 \oplus m_3) \oplus 3)$$

Now consider the following message $x = m_2 \circ 1 \circ y$ where

$$y = t_1 \oplus t_2 \oplus m_3$$

We have that

$$\text{NEWMAC}_k(x) = f_k(f_k(f_k(f_k(m_2) \oplus 1) \oplus y) \oplus 3) = f_k(f_k(t_2 \oplus t_1 \oplus t_2 \oplus m_3) \oplus 3) = t$$

and again Eve has the MAC of a message she did not see before.

Problem 2-3. Protocol Failure involving RSA

The security of RSA is based on the assumption that computing cube roots modulo a composite number of unknown factorization is hard. However this is not true for *all* possible ciphertexts. Let n be an RSA modulus. Suppose I encrypt a message m small enough that $m^3 < n$. Now the ciphertext $c = m^3 \bmod n$ is exactly equal to m^3 and so the message m can be easily recovered from c by computing $\sqrt[3]{c}$ over the integers.

Now the above observation does not yield directly the solution to our problem since we did not put any restriction on the message m sent by David. However it can be generalized. Let's distinguish two cases.

Case 1: n_A, n_B, n_C are *not* relatively prime to each other. But in this case Eve can just compute the GCD of them and factor them completely.

Case 2: n_A, n_B, n_C are relatively prime to each other. In this case we can apply the Chinese Remainder Theorem (see Chapter 33 of Cormen, Leiserson and Rivest *Introduction to Algorithms.*) The CRT says that given $x_i \bmod n_i$ where n_i are numbers relatively prime to each other, then there is an unique $x \bmod n = \Pi_i n_i$ such that $x = x_i \bmod n_i$. Also such x is efficiently computable. Applying the CRT to our case we can compute $c = m^3 \bmod n = n_A n_B n_C$. Since $m < n_A, n_B, n_C$ we have that $m^3 < n$ and so $c = m^3$ over the integers. So m can be computed as $\sqrt[3]{c}$

Problem 2-4. Perfect Forward Secrecy

For each session Alice and Bob are going to exchange session public keys, that is public keys whose secret key will be destroyed at the end of the session.

Let (PK_A, SK_A) and (PK_B, SK_B) the long lived public–secret key pairs of Alice and Bob. Here is the protocol

1. Alice generate a public-secret key pair for the session (PK_S, SK_S). She sends to Bob the session public key PK_S and a siganture of it using her long–lived secret key $SK_A(PK_S)$
2. Bob computes PK_S and generates a secret key K and sends it to Alice using the newly generated session public key $PK_S(K)$
3. Alice decrypts K. Now the session can begin by Alice and Bob exchanging messages encrypted with K (using say DES). At the end of the session Alice destroys SK_S.

When Eve breaks later into Alice's and Bob's computer she will not find SK_S and so she will not be able to find the session key K. Notice also that the communication is authenticated, i.e. Eve will not be able to impersonate Alice.

Some people were confused by wording of the question ("using only public–key cryptography") and thought that they were not allowed to use DES. Of course the above solution works in that case too. Only that in step 2 Bob sends to Alice his session public key and now they communicate encrypting their messages with the session public keys and not with the long–lived ones or with a DES key. The only difference is in the efficiency of the communication.