**E5. Group work: knowledge of cryptography**

**NAME\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**NAME\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**NAME\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**NAME\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**NAME\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**NAME\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

1. How can I distribute a secret key to two participants in a communication? Give an example using any of the methods seen in class.
2. With ECB mode, if there is an error in a transmitted ciphertext block, it only affects the corresponding plaintext block. However, in CBC mode, this error spreads. For example an error when transmitting Y1 obviously affects X1 and X2.
3. Suppose Y1 is a byte, like X1 and X2, are the bytes after X2 affected? Justify.
4. Suppose there is a bad bit in X1. Through how many blocks of ciphertext does this error propagate? What effect does it have on the receiver?
5. Suppose someone suggests the following way to confirm that two of you are in possession of the same secret key: “Create a random sequence of bits the same size as the key, XOR it with the key, and send the result over the channel; your receiver XORs the incoming block with the key (which should be the same as yours) and sends the result back; observe what it returns and if it matches your random sequence then you have verified that your receiver has the same secret key as you, although neither has transmitted the key. " Is there a defect in this scheme?
6. In a public key system using RSA, you intercept the ciphertext Y = 10, sending to a user whose public key is e = 5, n = 35. What is the plain text X?
7. “We are under great pressure, Holmes.” Detective Lestrade looked nervous. “We have learned that copies of sensitive government documents are stored in computers of one foreign embassy here in London. Normally these documents exist in electronic form only on a selected few government computers that satisfy the most stringent security requirements. However, sometimes they must be sent through the network connecting all government computers. But all messages in this network are encrypted using a top-secret encryption algorithm certified by our best crypto experts. Even the NSA and the KGB are unable to break it. And now these documents have appeared in hands of diplomats of a small, otherwise insignificant, country. And we have no idea how it could happen.”

“But you do have some suspicion who did it, do you?” asked Holmes. “Yes, we did some routine investigation. There is a man who has legal access to one of the government computers and has frequent contacts with diplomats from the embassy. But the computer he has access to is not one of the trusted ones where these documents are normally stored. He is the suspect, but we have no idea how he could obtain copies of the documents. Even if he could obtain a copy of an encrypted document, he couldn’t decrypt it.”

“Hmm, please describe the communication protocol used on the network.” Holmes opened his eyes, thus proving that he had followed Lestrade’s talk with an attention that contrasted with his sleepy look. “Well, the protocol is as follows. Each node *n* of the network has been assigned a unique secret key K*n*. This key is used to secure communication between the node and a trusted server. That is, all the keys are stored also on the server. User A, wishing to send a secret message M to user B, initiates the following protocol:

1. A generates a random number R and sends to the server his name A, destination B, and EKA(R).
2. Server responds by sending EKB(R) to A.
3. A sends ER(M) together with EKB(R) to B.
4. B knows KB, thus decrypts EKB(R), to get R and will subsequently use R to decrypt ER(M) to get M

I have already stated that the encryption algorithm used is in fact unbreakable, and you see that a random key is generated every time a message has to be sent. You know that the use of a random one-time key belongs to proper procedures that contribute to the strength of cryptosystems. I admit the man could intercept messages sent between the top secret trusted nodes, but I see no way he could decrypt them.”

“Well, I think you have your man, Lestrade. The protocol isn’t secure because the server doesn’t authenticate users who send him a request. Apparently designers of the protocol have believed that sending EKx(R) implicitly authenticates user X as the sender, as only X (and the server) knows Kx. But you know that EKx(R) can be intercepted and later replayed. Once you understand where the hole is, you will be able to obtain enough evidence by monitoring the man’s use of the computer he has access to. Most likely he works as follows: After intercepting EKA(R) and ER(M) (see steps 1 and 3 of the protocol), the man, let’s denote him as Z, will continue by pretending to be A and...”

*Finish the sentence for Holmes.*