Computer Security (COM-301)

Applied cryptography I

OTP is the best?

Agree or disagree and justify :

"A One Time Pad is the best choice to transmit a secret document of 10Mb because we know it provides perfect secrecy"

2

Reasons why OTP is **not** the best choice:

- You need a key *as long as the message*. When the message is 10Mb, this is not so easy to do random without modem crypto
- You need the means to transmit such a long key to the receiver. This must be done on a channel different than the one where the message is sent (otherwise the adversary can access the key!)

One could say that for the application at hand secrecy is so important that OTP must be used, but then you have to solve the two problems above (i.e., explicitly say how the key would be created and transmitted)

Secure streaming

Is the following pseudocode for a function to encrypt movies a secure use of a stream cipher? Justify your answer.

```
var iv = "bestidever"

def EncryptMovie(movie):
      key = GenerateSecureRandomKey() // generates a truly random key
      secureKeyTransfer(key)  // securely sends the key to the
                              // recipient
      EncryptedMovie = Salsa20(key,iv,movie)    // uses Salsa20 stream
                                                // cipher
      return(EncryptedMovie)
```

Yes, it is secure because, even if the IV is constant, a new random key is generated for encryption in every loop -> combination (key, iv) is fresh -> there will be no two messages encrypted with the same pair.

Key exchange

Alice and Bob want to derive a shared secret using the Diffie-Hellman protocol. Alice will use this shared secret as a key to symmetrically encrypt a message ***m*** to Bob so that no one but Bob knows what she says to him. All in all, their protocol goes as follows.

1) The public parameters are the modulus ***p*** and the generator ***g***.
2) Alice chooses a large secret number ***x*** and sends $g^x \pmod p$ to Bob.
3) Bob chooses a large secret number ***y*** and sends $g^y \pmod p$ to Alice.
4) After deriving the shared secret $sk = g^{xy} \pmod p$, Alice uses it as a key to encrypt the message ***m*** to a ciphertext $c = Enc(sk, m)$ and sends ***c*** to Bob.

What Alice and Bob are not aware of is that Mallory eavesdrops on all their communication and can tamper with the messages they send to each other (Mallory can substitute any of their messages with an arbitrary string).

Describe an attack in which Mallory achieves both of the following goals:

1) learns Alice's message ***m***, and then
2) makes sure that instead of Alice's message, Bob reads another message ***m'*** of Mallory's choosing.

Both 1) and 2) have to be done in such a way that Mallory is not detected (that is, from the point of view of Alice and Bob, the protocol goes normally).

1. Mallory picks secret a, sends g^a to Alice and Bob in place of their correspondent's original message (g^x or g^y). Mallory will have two different shared secrets with Alice and Bob: g^{ax} and g^{ay}.
2. Mallory intercepts c from Alice and decrypts with g^{ax} to learn m.
3. Using the shared secret g^{ay} with Bob, Mallory encrypts her message m' to c' = Enc(g^{ay}, m') and sends c' to Bob instead of Alice's original c.