1. [KR, Chapter 8, P1] Using the substitution cipher (monoalphabetic cipher) shown in Lecture 10, on page 15, notes:

   uasi si micbocu h ciimzc

   a) encode the message "this is a secret message"

   b) decode the message "tcow ihmou"

   very smart

2. [KR, Chapter 8, R6] Suppose N people each want to communicate with $N-1$ other people. All communication between any two people, i and j, is visible to all other people but no other person should be able to decode their communication. In total, how many keys are required in this group if:
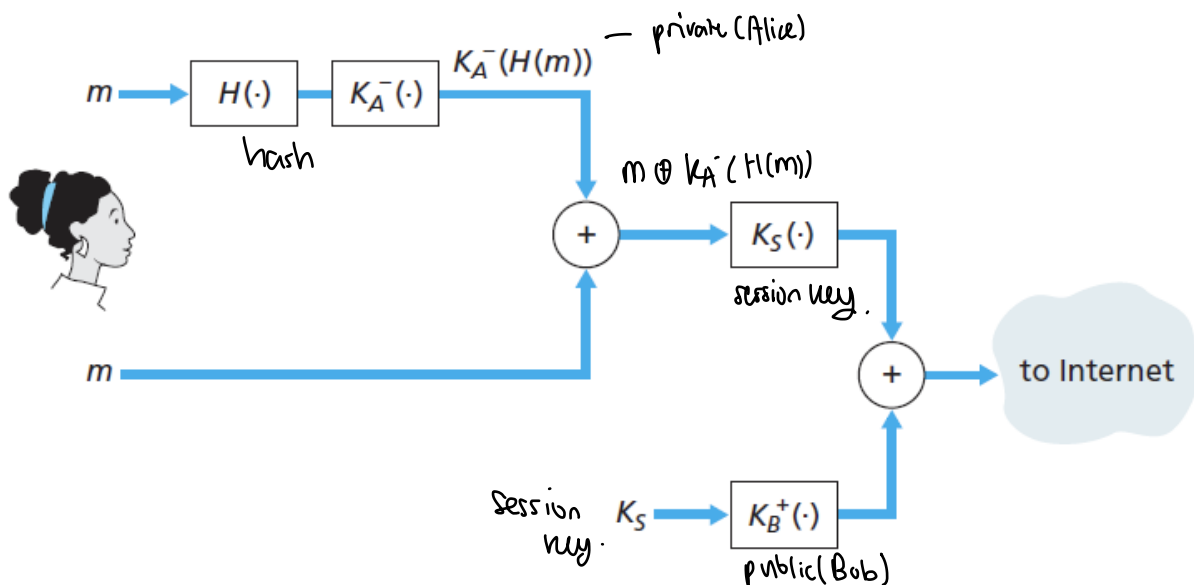
   a) Symmetric key encryption is used in each communication? $N(N-1)/2$ (complete graph).

   b) Public key encryption is used in each communication? $2N$

   — block is not edited.

   hash = integrity

3. [KR, Chapter 8, P13] In the BitTorrent P2P file distribution protocol, the seed breaks a file into blocks, and the peers redistribute the blocks to each other. Without any protection, an attacker can easily wreak havoc in a torrent by masquerading as a benevolent peer and sending bogus blocks to a small subset of peers in the torrent. These unsuspecting peers then redistribute the bogus blocks to other peers, which in turn redistribute the bogus blocks to even more peers. Thus, it is critical for BitTorrent to have a mechanism that allows a peer to verify the integrity of a block, so that it doesn't redistribute bogus blocks.

   , Contains Authentication keys.

   hashes for all block saved in .torrent.

   Assume that when a peer joins a torrent, it initially gets a .torrent file from a *fully trusted* source. Describe a simple scheme that allows peers to verify the integrity of blocks.

   Send $(m, H(m+s))$, on receiving side can check if $H(m+s)$ is the same using s to verify the message's integrity.

   Message Authentication code.

   ✱ 4. Suppose Alice wants to send a secure email m to Bob, and wants to ensure its confidentiality and integrity. Alice performs the following steps (Figure 8.21 on textbook which is reproduced below):

Annotations on the figure: — private (Alice); $K_A^-(H(m))$; hash; $m \oplus K_A^-(H(m))$; session key; session key; public (Bob); to Internet

1. generates a random session key K_S

2. encrypts the session key with Bob's public key K_B^+, obtaining $K_B^+(KS)$

3. hashes the message m with a cryptographic hash function H, obtaining message digest $H(m)$

4. encrypts the hash with Alice's private key K_A^-, obtaining digital signature $K_A^-(H(m))$

5. encrypts the message m, concatenated ($\oplus$) with $K_A^-(H(m))$, using the session key K_S to obtain $K_S(\, m \oplus K_A^-(H(m))\,)$ — authenticity   confidentiality.

6. finally, sends $K_S(\, m \oplus K_A^-(H(m))\,) \oplus K_B^+(K_S)$ to Bob   — integrity.

Show what Bob has to do to verify that m is indeed ==crafted by Alice== and has ==not been== ==modified during transmission==.

$$K_S(\, m \oplus K_A^-(H(m))\,) \oplus K_B^+(K_S).$$

parse

$$K_S(m \oplus K_A^-(H(m)))$$ $$K_B^-(K_B^+(K_S))$$

decrypt using session key. $$K_S$$

$$m \oplus K_A^-(H(m))$$

parse

$$m$$ $$K_A^+(K_A^-(H(m)))$$

$$H(m)$$ ———————— $$H(m)$$

Check?