National University of Singapore
School of Computing

CS2105 **Tutorial 9** Semester 2 AY18/19

---

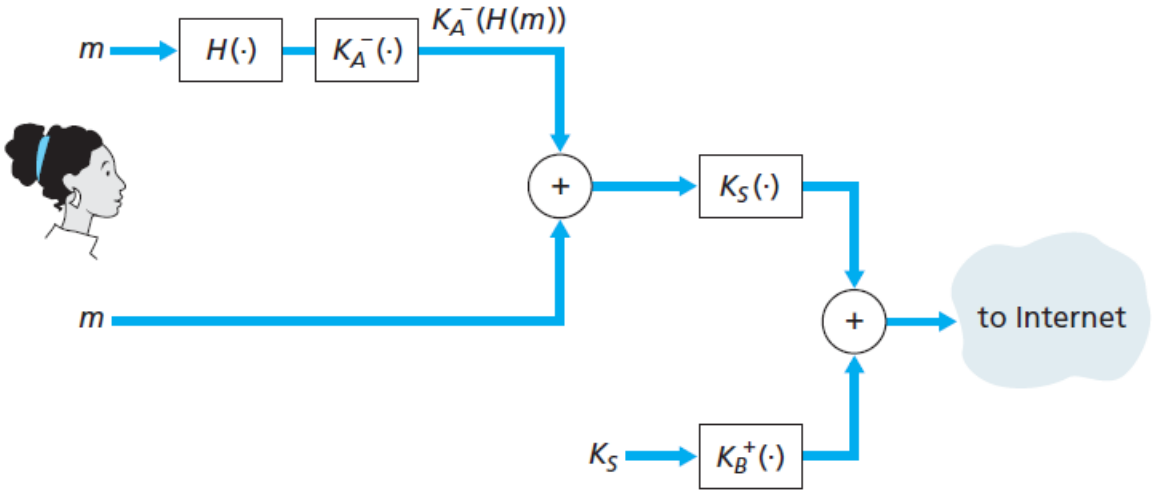
1. **[KR, Chapter 8, P1]** Using the mono-alphabetic cipher shown in lecture 10 notes,
   a) Encode the message "`This is a secret message`"
   b) Decode the message "`fsgg ash`"

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?
   b) Public key encryption is used in each communication?

3. **[KR, Chapter 8, R15]** Suppose Alice has a message that she is ready to send to anyone who asks. Thousands of people want to obtain Alice's message, but each wants to be sure of the integrity of the message. In this context, do you think a MAC-based or a digital-signature-based integrity scheme is more suitable? Why?

4. **[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.
   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.

5. 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):



1. generates a random session key K_S
2. encrypts the session key K_S with Bob's public key K_B^+, obtaining $K_B^+(K_S)$
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 $K_S(\ m \oplus K_A^-(H(m))\ )$
6. finally, sends $K_S(\ m \oplus K_A^-(H(m))\ ) \oplus\ K_B^+(K_S)$ to Bob

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