National University of Singapore
School of Computing

CS2105 **Tutorial 9** Answer paper

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

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

    uasi si m icbocu hciimzc

    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?

    There are $N * (N-1)/2$ pairs of people and each pair needs to share a symmetric key. The total number of keys is $N * (N-1)/2$.

    b) Public key encryption is used in each communication?

    With public key encryption, each person has a public key which is known to all, and a private key which is secret and known to the user only. There are thus $2 * N$ keys.

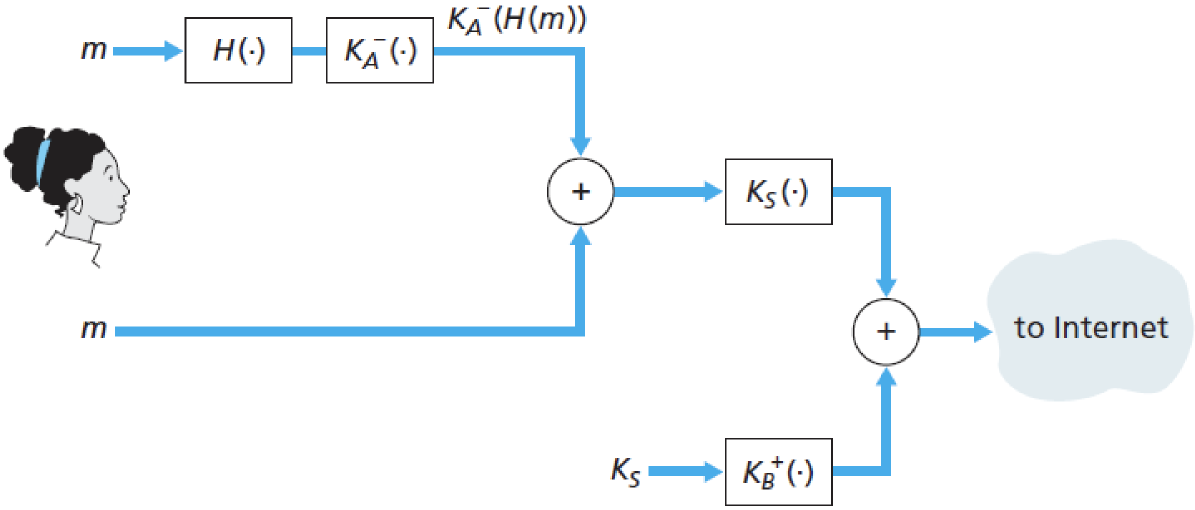
3. [KR, Chapter 8, P13] In the BitTorrent P2P file distribution protocol, the seed breaks a file into blocks, and the peers redistribute the blocks to each other. Without any protection, an attacker can easily wreak havoc in a torrent by masquerading as a benevolent peer and sending bogus blocks to a 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.

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.

A file is broken into a number of blocks of identical size. For each block, a hash is calculated (e.g., using MD5 or SHA-1). The hashes for all of the blocks are saved in the `.torrent` file.

When a block is downloaded, a peer calculates the hash of this block and compares it to the recorded hash in the `.torrent` file. If the two hashes are equal, this block is error-free. Otherwise, the block is bogus and should be discarded.

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):



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 $K_S(\ m \oplus K_A^-(H(m))\ )$
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 crafted by Alice and has not been modified during transmission.

Bob has to:

1. **compute $K_B^-\left(K_B^+(K_S)\right) = K_S$ to recover the session key using Bob's private key**
2. **with K_S, Bob decrypts the message $(K_S(K_S(m \oplus K_A^-(H(m)))))$, gets m and $K_A^-(H(m))$**
3. **use Alice's public key K_A^+ to recover $H(m)$: $K_A^+\left(K_A^-\left(H(m)\right)\right) = H(m)$**
4. **with m, Bob computes $H(m)$ and verifies that it is equal to $H(m)$ from step 3**

NOTE: symmetric key crypto (K_S) is used to encrypt m, instead of public key crypto because public key crypto is much slower.

Since only Bob has his private key, the intruder cannot decrypt $K_B^+(K_S)$ and therefore cannot get the session key K_S. Without K_S, the intruder cannot decrypt m.

The intruder cannot impersonate Alice, since Alice digitally signed the message with her private key $K_A^-(H(m))$. Otherwise, Bob cannot get the right $H(m)$ when trying to decrypt with Alice's public key.

The intruder cannot tamper with m, since modifying m would cause its hash to be different and Bob would detect this.