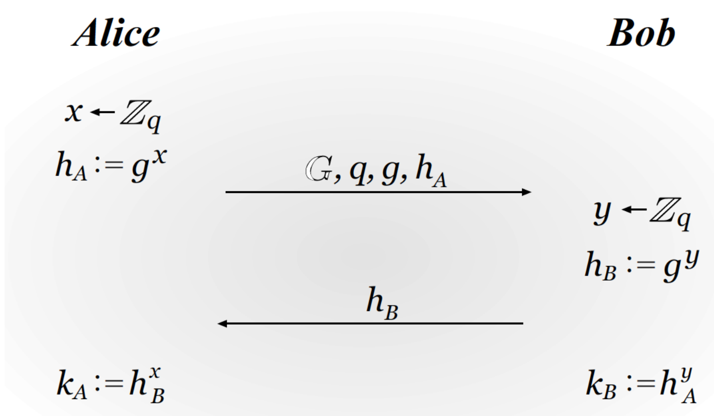
Homework 6: Key Exchange and Encryption

Submission policy. Submit your answers on Blackboard by 11:59pm Friday, **Nov. 19**, 2021. Your submission should include a .PDF file with all the answers to the theoretical problems. You should try to typeset your homeworks. Latex is especially recommended. No late submissions will be accepted. Your writeup MUST include the following information:

1. Your name and whether you take the class at **487** or **587** level.
2. List of references used (online material, course nodes, textbooks, wikipedia, etc.)

The homework will be graded by the class TA Anuj Pokhrel (apokhre@gmu.edu).

Exercise 1. Key-Exchange and Public Key Encryption [30 points]
Let Π_1 be the Diffie-Hellman Key-exchange protocol where $k = k_A = k_B$.



Using Π_1, we will create a public-key encryption scheme Π_2 which works as follows:

- KeyGen outputs $pk = h_A$ and $sk = x$ selected as in the DH key-exchange protocol.
- $Enc(pk, m)$ outputs $c = (h_B, m \oplus k)$ i.e, in order to encrypt a message first create a value of the format of $h_B = g^y$ (as in DH key exchange), compute a value $k = pk^y$ and construct the ciphertext.

- $Dec(sk, c)$ Let c_1 be the first part of the ciphertext and c_2 be the second part of the ciphertext. Using $c_1 = h_B$, compute k and output $m = k \oplus c_2$.

1. For the Diffie-Hellman Key-exchange protocol explain why an eavesdropping adversary cannot simply compute $k_A = k_B = h_a h_b$.

2. Show why the proposed public key encryption scheme Π_2 is correct (i.e. why decryption is successful).

3. Prove (via a reduction) that the encryption scheme Π_2 is CPA secure assuming Π_1 is a secure key exchange protocol.

Exercise 2. Key Exchange [20 points] In class we discussed Diffie-Hellman Key Exchange. Here we consider an alternative protocol for Alice and Bob to agree on a key k:

- Bob starts by picking two values a, b from $\{0,1\}^n$ uniformly at random.
- Bob computes $w_1 = a \oplus b$ and sends w_1 to Alice.
- Alice picks a value t from $\{0,1\}^n$ uniformly at random.
- Alice sends $w_2 = w_1 \oplus t$ to Bob.
- Bob sends $w_3 = w_2 \oplus b$ to Alice.
- Bob sets $k = a$ and Alice sets $k = w_3 \oplus t$.

1. Explain why the protocol is correct, i.e. show that Alice and Bob compute the same exact key.

2. Is this protocol a secure key-exchange protocol as we defined it in class? If you say yes prove it, if you say no show a concrete attack.

Exercise 3. El Gamal Encryption [30 points] In class we discussed El Gamal encryption for messages $m \in G$. We now present an alternative encryption scheme for encrypting the output of a single coin flip = {head, tail}. The algorithm works as follows:

- *KeyGen*: exactly as in ElGamal

- $Enc_{pk}(m)$: (a) if m =head, then pick $y \in \mathbb{Z}_q$ uniformly at random and output $c = (g^y, h^y)$, (b) if m =tail, then pick $y, z \in \mathbb{Z}_q$ uniformly (and independently) at random and output $c = (g^y, g^z)$.

1. Show how decryption would work (assuming knowledge of secret key of course).

2. Prove that the scheme is CPA secure if the decisional Diffie-Hellman Problem is hard in G.

Exercise 4. RSA [20 points] Consider the following fix for plain RSA in order to address the issues we discussed in class. This alternative RSA encryption scheme *only* works with messages that have length exactly half of the bit-length of the RSA modulus N, i.e. $|m| = \| N \| /2$.

- KeyGen works exactly as in RSA.
- To encrypt a message m, compute m' first compute $m' = 00000000||r||00000000||m$ where r is a uniform string of length $(\| N \| /2) - 16$ and 2 strings of all zeros of length 8 bits each are used in the concatenation. Compute the ciphertext to be $c = [m'^e \bmod N]$.
- To decrypt, a ciphertext c, the receiver computes $m' = [c^d \bmod N]$ and returns an error if m' does not consist of 00000000 followed by $(\| N \| /2) - 16$ arbitrary bits followed by 00000000.

  1. This scheme is NOT CCA secure. Show an attack.
  2. Why is it easier to construct a chosen-ciphertext attack on this scheme than on PKCS #1 v1.5?

Exercise 5. Only if in 587 [20 points]

1. Suppose you are given an El Gamal encryption of an unknown plaintext $M \in G$. Show how to construct a different ciphertext that also decrypts to the same M.
2. Suppose you are given two El Gamal encryptions, of unknown plaintexts $M_1, M_2 \in G$. Show how to construct a ciphertext that decrypts to their product $M_1 \cdot M_2$.