Homework Problems : Public Key Encryption

1. Recall that the Diffie Hellman Exchange (DHE) protocol described in the lecture establishes a common secret (g^{ab}) between two parties A and B, with a single round communication, where A shares g^a with B and B shares g^b with A. This is called a single-round two-party key exchange.

   Based on the DHE protocol, devise a protocol to share a common key between three parties A, B and C. What is the complexity of this protocol in terms of number of rounds of communication between the parties? Is it possible to devise a single-round three-party key exchange?

2. Study the algorithm for RSA key generation, as follows, and provide an estimate for the runtime of each step in this algorithm (in terms of n), and the justification for the estimates you provide. Please note that the following algorithm is for textbook-RSA, and not the real-world RSA.

   Input : 1^n — where n is the security parameter for key generation
   Output : (pk, sk) — the public key and secret key pair for RSA

   Step 1 : Choose two random primes p, q of bit-size between $[n/2] - 1$ to $[n/2] + 1$
   Step 2 : Compute $N = pq$ and $\text{phi}(N) = (p-1)(q-1)$, where phi is the Euler totient function
   Step 3 : Choose a random integer e such that $1 < e < \text{phi}(N)$ and $\gcd(e, \text{phi}(N)) = 1$
   Step 4 : Compute d, the multiplicative inverse of e modulo $\text{phi}(N)$ (i.e., $ed = 1 \bmod \text{phi}(N)$)
   Step 5 : Set $pk = (N, e)$, $sk = (N, d)$, and output the pair of keys (pk, sk)

3. Construct a Public Key Encryption (PKE) scheme, given a secure Trapdoor Permutation (TDP) and a secure Symmetric Key Encryption (SKE) scheme. For instance, you may consider that you are provided with the RSA trapdoor permutation, as described in the lecture, and any set of nice symmetric key encryption and decryption algorithms $E_k(.)$ and $D_k(.)$, as taught earlier.