KFUPM
College of Computer Science and Engineering
Computer Engineering Department
COE 426/526: Data Privacy

Fall 2020 (201)

Assignment 3: Due date Saturday 28/11/2019

Tasks

Q1: Homomorphic Encryption (10 points) Elgamal encryption is an asymmetric key encryption algorithm for public-key cryptography which is based on the Diffie–Hellman key exchange. It consists of three components: key generation, encryption and decryption.

- Key generation: Alice generates a key pair as follows:
  - Generate an efficient description of a cyclic group G of order q, with generator g.
  - Let e represent the unit element of G.
  - Choose an integer x randomly from $\{1, \ldots, q-1\}$.
  - Compute $h := g^x$.
  - The public key consists of the values (G, q, g, h). Alice publishes this public key and retains x as her private key, which must be kept secret.
- Encryption: Bob encrypts a message M to Alice using her public key (G, q, g, h) as follows:
  - Map the message M to an element m of G using a reversible mapping function.
  - Choose an integer y randomly from $\{1, \ldots, q-1\}$
  - Compute $s := h^y$. This is called the shared secret.
  - Compute $c_1 := g^y$.
  - Compute $c_2 := m \cdot s$.
  - Bob sends the ciphertext (c_1, c_2) to Alice
- Decryption: Alice decrypts a ciphertext (c_1, c_2) with her private key x as follows:

- Compute $s := c_1^x$. Since $c_1 = g^y$, $c_1^x = g^{xy} = h^y$ and thus it is the same shared secret that was used by Bob in encryption.
- Compute s^{-1}, the inverse of s in the group G. This can be computed in one of several ways. If G is a subgroup of a multiplicative group of integers modulo n, the modular multiplicative inverse can be computed using the Extended Euclidean Algorithm. An alternative is to compute s^{-1} as c_1^{q-x}. This is the inverse of s because of Lagrange's theorem, since $s \cdot c_1^{q-x} = g^{xy} \cdot g^{(q-x)y} = (g^q)^y = e^y = e$.
- $m := c_2 \cdot s^{-1}$. This calculation produces the original message m, because $c_2 = m \cdot s$; hence $c_2 \cdot s^{-1} = (m \cdot s) \cdot s^{-1} = m \cdot e = m$.
- Map m back to the plaintext message M.

Answer the following questions.

(a) (5 points) Show that the above Elgamal encryption scheme is homomorphic with respect to multiplication.

(b) (5 points) Show that the above Elgamal encryption scheme is not homomorphic with respect to addition.

Q2: Homomorphic-Based Yao Millionaire Problem (15 points)

(a) (5 points) Explain why does the Homomorphic based protcol for Yao's millionaire problem (in Lecture 11 slides 22-23) fail when using using unpadded RSA?

(b) (10 points) Design a protocol that uses unpadded RSA. Verify that your protocol works by implementing your proposed protocol using the notebook file ("Yao_RSA.ipnyb").

Q3: Oblivious Transfer (OT) (10 pts)

(a) (10 points) Design a simple protocol for 1-out-of-n OT starting from 1-out-of-2 OT. Assume that both Alice and Bob are honest-but-curious. i.e., they follow the protocol but from time to time they collect extra information looking for exposing private data about each other. In your protocol, Alice and Bob can access the 1-out-of-2 functionality n times. Explain your protocol n details (Hint: Think of how to extend 1-out-of-2 to 1-out-of-3 and then generalize it to 1-out-of-n)

(b) (Bonus 10 points) Implement the 1-out-of-n OT protocol in "OT_1_n.ipynb" using Socket Programming.