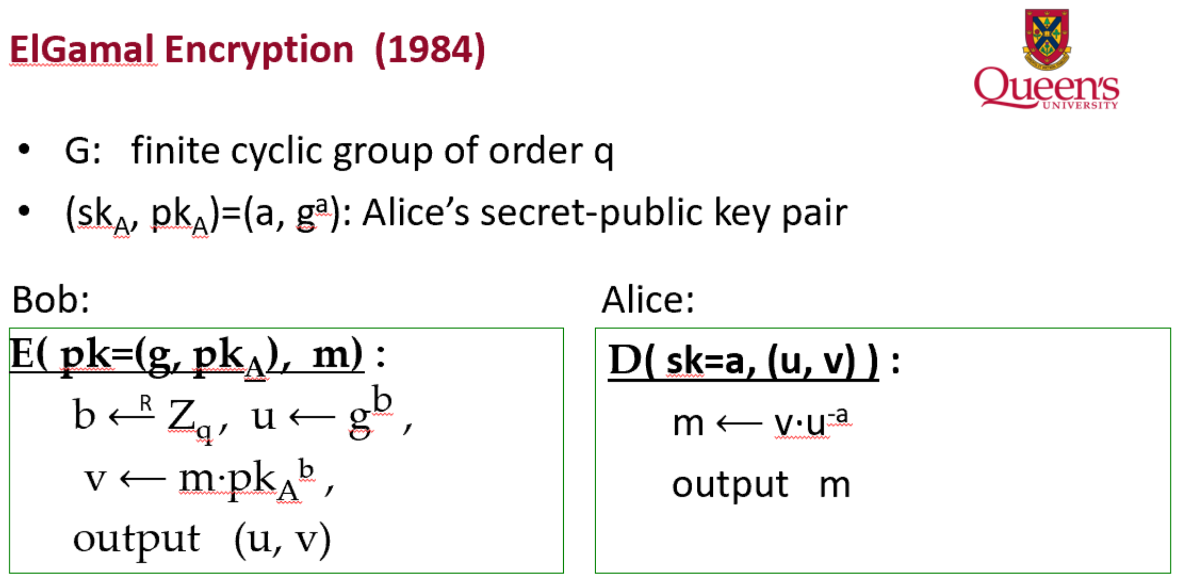
Assignment 2 of ELEC 473

Use C/C++/Java/Python to implement one of the following two schemes.

1. Elliptic Curve ElGamal Encryption



ElGamal Encryption (1984)

- G: finite cyclic group of order q
- $(sk_A, pk_A)=(a, g^a)$: Alice's secret-public key pair

Bob:

$E(\ pk=(g, pk_A),\ m)$:

$b \xleftarrow{R} Z_q$, $u \leftarrow g^b$,

$v \leftarrow m \cdot pk_A^{\,b}$,

output (u, v)

Alice:

$D(\ sk=a, (u, v)\ )$:

$m \leftarrow v \cdot u^{-a}$

output m

Setting: 1. Choose an elliptic curve from four different standardized elliptic curves (secp160/192/224/256).

2. Select the generator, private key, and random integers as you want.

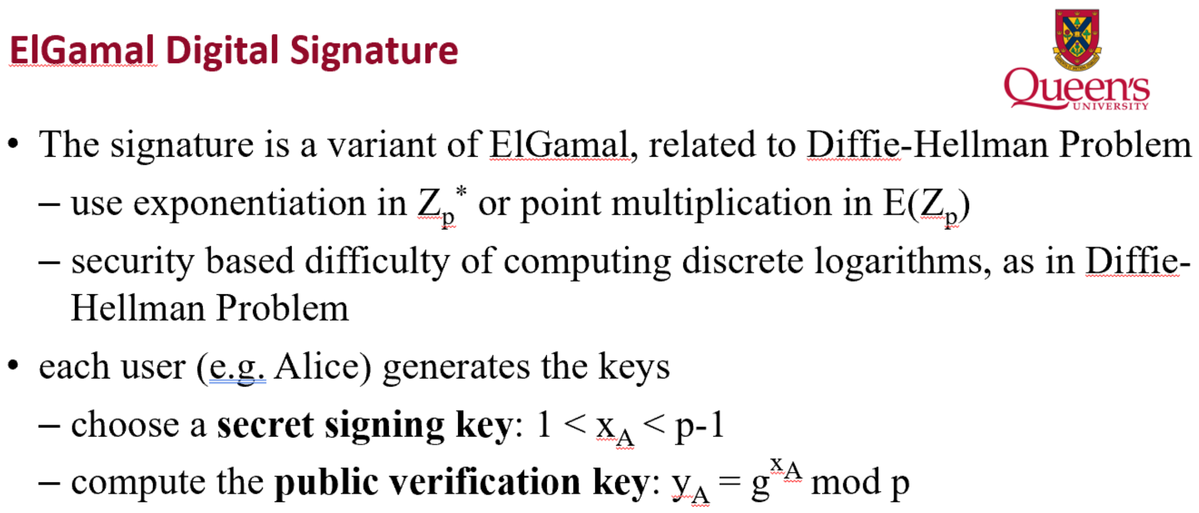
Objective: Implement Setup, KeyGen, Encryption, and Decryption of Elliptic Curve ElGamal Encryption and run 2 test cases:

- Test Case 1: Plaintext: I am an undergraduate student at queen's university.
- Test Cast 2: plaintext: (your full name)

Deliverables:

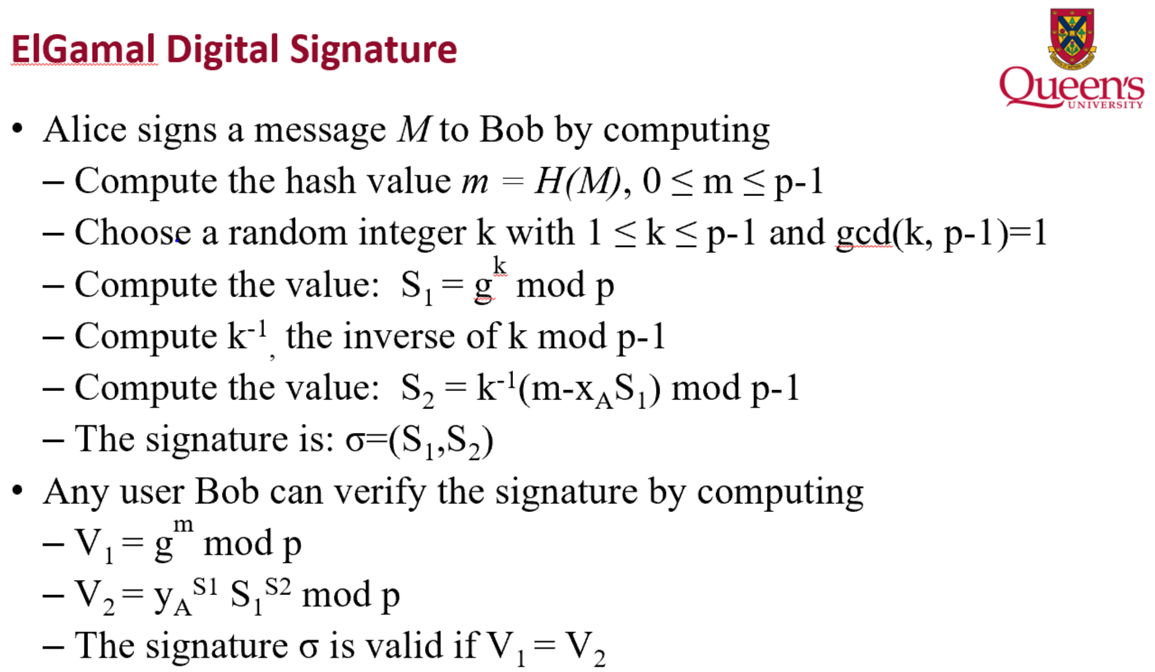
- The source codes
- The generator and the random values you use
- The private key and public key files
- The ciphertext files
- The decrypted plaintext files

2. Elliptic Curve ElGamal Digital Signature



ElGamal Digital Signature

- The signature is a variant of ElGamal, related to Diffie-Hellman Problem
  - use exponentiation in Z_p^* or point multiplication in $E(Z_p)$
  - security based difficulty of computing discrete logarithms, as in Diffie-Hellman Problem
- each user (e.g. Alice) generates the keys
  - choose a **secret signing key**: $1 < x_A < p-1$
  - compute the **public verification key**: $y_A = g^{x_A} \bmod p$



ElGamal Digital Signature

- Alice signs a message *M* to Bob by computing
  - Compute the hash value $m = H(M)$, $0 \leq m \leq p-1$
  - Choose a random integer k with $1 \leq k \leq p-1$ and $\gcd(k, p-1)=1$
  - Compute the value: $S_1 = g^k \bmod p$
  - Compute k^{-1}, the inverse of k mod p-1
  - Compute the value: $S_2 = k^{-1}(m-x_A S_1) \bmod p-1$
  - The signature is: $\sigma=(S_1,S_2)$
- Any user Bob can verify the signature by computing
  - $V_1 = g^m \bmod p$
  - $V_2 = y_A^{S1}\, S_1^{S2} \bmod p$
  - The signature σ is valid if $V_1 = V_2$

Setting: 1. Choose an elliptic curve from four different standardized elliptic curves (secp160/192/224/256).

2. Select the generator, private key, and random integers as you want.

Objective: Implement Setup, KeyGen, Signing, and Verification of Elliptic Curve ElGamal Digital Signature and run 2 test cases:

- Test Case 1: Message: I am an undergraduate student at queen’s university.
- Test Cast 2: Message: (your full name)

Deliverables:

- The source codes
- The generator and the random values you use
- The private key and public key files
- The signature files
- The successful verification screenshot