MATH 406 Section 10.2: ElGamal Cryptosystem

1. **Introduction:**

   The ElGamal cryptosystem is based on the fact that it is easy to raise things to powers in modular arithmetic but difficult to take (discrete) logarithms.

2. **Key Creation:**

   Bob selects a prime p, a primitive root r of p, and an integer a with $0 \leq a \leq p-1$. The value is kept private.

   Bob then calculates the least nonnegative residue $b \equiv r^a \bmod p$. The triple (p, r, b) is then made public.

   Observe that $a \equiv \text{ind}_r b \bmod p-1$ and this is difficult to calculate.

3. **Encryption:**

   Suppose Alice wants to send a plaintext block $0 \leq P < p$ to Bob. She first chooses a random number $1 \leq k \leq p-2$. Then we encrypt via the encryption function by:

   $$\mathcal{E}(P) \equiv (r^k, Pb^k) \bmod p$$

   This produces a pair (γ, δ) which constitutes the cipertext. That is, $\gamma \equiv r^k \bmod p$ and $\delta = Pb^k \bmod p$.

   Typically a different k is used for each plaintext block. The primary reason for this is that then identical plaintext blocks will have different corresponding ciphertext blocks.

4. **Decryption:**

   To decrypt the ciphertext block (γ, δ) we claim that;

   $$P \equiv \gamma^{p-1-a}\delta \bmod p$$

   This works because of the following, where the negative exponent means the multiplicative inverse.

   $$\begin{aligned}
   \gamma^{p-1-a}\delta &= (r^k)^{p-1-a}Pb^k \bmod p \\
   &= (r^{p-1})^k (r^a)^{-k} b^k P \bmod p \\
   &= (1)^k b^{-k} b^k P \bmod p \\
   &= P \bmod p
   \end{aligned}$$

   Thus we have the decription function:

   $$\mathcal{E}^{-1}(\gamma, \delta) \equiv \gamma^{p-1-a}\delta \bmod p$$

5. **Example:**

   Bob selects $p = 2539$, $r = 2$ (this is a primitive root), and $a = 42$. He keeps $a = 42$ private. He calculates $b \equiv 2^{42} \equiv 1305 \bmod 2539$ and makes $(p, r, a) = (2539, 2, 1305)$ public.

   Alice wants to send him `OHNO` so she splits it as `OH`=1407 and `NO`=1314.

   - To encrypt 1407 she chooses $k = 100$ and calculates:
   $$\mathcal{E}(1407) \equiv (2^{100}, 1407 \cdot 1305^{100}) \equiv (613, 635) \bmod 2539$$
   - To encrypt 1314 she chooses $k = 200$ and calculates:
   $$\mathcal{E}(1314) \equiv (2^{200}, 1314 \cdot 1305^{200}) \equiv (2536, 1494) \bmod 2539$$

   She sends these two pairs to Bob.

   Bob receives them and decrypts via:

   - To decrypt $(613, 635)$ he does:
   $$\mathcal{E}^{-1}(613, 635) \equiv 613^{2539-1-42} 635 \equiv 1407 \bmod 2539$$
   - To decrypt $(2536, 1404)$ he does:
   $$\mathcal{E}^{-1}(2536, 1404) \equiv 2536^{2539-1-42} 1404 \equiv 1314 \bmod 2539$$

6. **Notes:**

   (a) To decrypt a message Eve needs to know a which as we have said involves taking the discrete logarithm of b using the primitive root r.

   (b) Although r is known and arguably one could try lots of a until $r^a \equiv b \bmod p$, in practice if p is large this is impractical.

   (c) Because each plaintext block gets its own choice of k, identical plaintext blocks will have different ciphertext blocks.

   (d) The ciphertext is twice as long as the plaintext. The advantage is that noted above.

   (e) As with most cryptosystems (we didn't mention this with RSA) in practice this would be used to share symmetric key which is then used to encrypt and decrypt the message.

   (f) While it's possible to sign messages using ElGamal, the method is not quite as simple as with RSA. With RSA Alice would simply user her own decryption key but here decryption works on pairs (γ, δ) which emerge from (P, k) and not on single blocks of text.

   (g) Verifying primitive roots can take a while but since Bob's p and r are public it doesn't matter how he obtains them or who knows. Heck, Alice could have given them to him. It's the a that's critical.

   (h) Alice should definitely choose a random k each time. If she encrypts both P_1 and P_2 with the same k and if Eve figures out P_1 then she can figure out P_2. This is because $\gamma_1 \equiv P_1 b^k \bmod p$ and $\gamma_2 \equiv P_2 b^k \bmod p$ and since Eve knows γ_1 and γ_2 she can calculate:
   $$P_2 \equiv \gamma_2 (b^k)^{-1} \equiv \gamma_2 (\gamma_1 P_1^{-1})^{-1} \equiv \gamma_2 \gamma_1^{-1} P_1 \bmod p$$