CS181A Notes #5 Notes on Elliptic Curve for ElGamal

Here we describe the ElGamal *probabilistic* public-key cryptosystem on an additive group generated by an elliptic curve of the form $y^2 \equiv x^3 + Ax + B \pmod{q}$, where q is a prime and $\Delta = 4A^3 + 27B^2 \not\equiv 0 \pmod{q}$. The last condition ensures that we obtain three distinct roots from the cubic form and we explain this in what follows. The idea of using elliptic curves as an alternate group for the ElGamal system was due to Koblitz and independently Miller. This switch is motivated by an attempt to find a group where the Discrete Logarithm problem is harder than on $\mathbb{Z}_q^\star$.

Cubic curves Consider the problem of solving the cubic polynomial:

$$x^3 + ax^2 + bx + c = 0.$$

Suppose we set $x = z - \alpha$. Then,

$$z^3 + (a - 3\alpha)z^2 + (3\alpha^2 - 2a\alpha + b)z + (-\alpha^3 + a\alpha^2 - b\alpha + c) = 0.$$

If we set $\alpha = a/3$, we get a simpler cubic form (the so-called Weirstrass form):

$$x^3 + ax + b = 0.$$

Now, assume $x = y + z$ for two additional variables. Then, we get

$$y^3 + (3yz + a)(y + z) + z^3 + b = 0.$$

If we set $yz = -a/3$, then we obtain two equations in two unknowns:

$$y^3 + z^3 = -b \tag{1}$$
$$y^3 z^3 = -a^3. \tag{2}$$

Letting, $Y = y^3$ and $Z = z^3$, we have a quadratic equation

$$Y^2 + bY - a^3/27 = 0.$$

The discriminant of this quadratic (after normalization) is

$$\Delta = 4a^3 + 27b^2.$$

We require $\Delta \neq 0$ to avoid getting repeated roots.

Group Law for Elliptic Curves Consider the curve $y^2 \equiv x^3 + Ax + Bx \pmod{q}$, where q is a prime and $\Delta = 4A^3 + 27B^2 \not\equiv 0 \pmod{q}$. We are interested in the set of points

$$E(C) = \{(x, y) \in \mathbb{Z}_q \times \mathbb{Z}_q : y^2 \equiv x^3 + Ax + B \pmod{q}\}$$

and viewing them as an additive *group*. For this, we need to define an *addition* operation between pairs of points $P, Q \in E(C)$ so that $P + Q \in E(C)$ and that the operation is associative, there is an identity element $\mathcal{O}$, and each point has an inverse.

(a) $P + (Q + R) = (P + Q) + R$, for all points P, Q and R.

(b) $P + \mathcal{O} = \mathcal{O} + P = P$, for all points P.

(c) For each point P, there is a point Q, so that $P + Q = Q + P = \mathcal{O}$.

The *identity* element $\mathcal{O}$ is called **Point at Infinity** and represents all vertical lines.

The **addition** operation can be motivated using analytic geometry arguments. Suppose $P = (x_1, y_1)$ and $Q = (x_2, y_2)$, and we wish to define $P + Q$.

(I) Case $x_1 \not\equiv x_2 \pmod{q}$.
We locate the line $y = mx + c$ that passes through P and Q whose slope m and intercept c are

$$\begin{aligned} m &\equiv (y_2 - y_1)(x_2 - x_1)^{-1} \pmod{q} \\ c &\equiv y_1 - mx_1 \pmod{q}. \end{aligned}$$

This line must intersect a third point $R = (x_3, y_3)$ on the curve unless it is tanget at P or Q. Thus, using the line expression for y in the cubic form, we get

$$(mx + c)^2 = x^3 + Ax + B$$

which yields $x^3 - m^2x^2 + (a - 2mc)x + (b - c^2) = 0$. Since the cubic splits into three roots (by the assumption on the discriminant):

$$(x-x_1)(x-x_2)(x-x_3) = x^3-(x_1+x_2+x_3)x^2+(x_1x_2+x_1x_3+x_2x_3)x-x_1x_2x_3.$$

So, we have

$$\begin{aligned} x_3 &\equiv m^2 - (x_1 + x_2) \pmod{q} \\ y_3 &\equiv mx_3 + c \equiv y_2 + m(x_3 - x_2) \pmod{q}. \end{aligned}$$

(II) Case $x_1 \equiv x_2 \pmod q$ but $y_1 \equiv -y_2 \pmod q$:
So, P and Q are reflections of each other across the x-axis. In this case $P + Q = \mathcal{O}$.

(III) Case $x_1 \equiv x_2 \pmod q$ and $y_1 \equiv y_2 \pmod q$:
Here, we have $P = Q$ and this case is also called *point doubling*. We locate the line tangent to the point P by taking implicit differentiation of the curve to get $dy/dx = (3x^2 + a)/(2y)$. Thus, we have

$$\begin{aligned} m &\equiv (3x_1^2 + a)(2y_1)^{-1} \pmod q \\ c &\equiv y_1 - mx_1 \pmod q. \end{aligned}$$

Finally, our "third point" is given by

$$\begin{aligned} x_3 &\equiv m^2 - 2x_1 \pmod q \\ y_3 &\equiv y_1 + m(x_3 - x_1) \pmod q. \end{aligned}$$

Another useful operation is **multiplication** (akin to exponentiation for the multiplicative group $\mathbb{Z}_q^\star$. Here, we use a similar idea to the fast modular exponentiation algorithm:

$$n * P = \begin{cases} \mathcal{O} & \text{if } n = 0 \\ 2 * (n/2 * P) & \text{if } n > 0 \text{ is even} \\ P + (n-1) * P & \text{if } n > 0 \text{ is odd} \end{cases}$$

Now, we are ready to describe the ElGamal cryptosystem adapted to the group induced by elliptic curves modulo a prime q.

Setup phase Bob prepares his cryptographic keys as follows:

1. Choose a random k-bit prime numbers q.
2. Choose random parameters A and B.
   This defines the curve C of $y^2 \equiv x^3 + ax + b \pmod q$.
3. Choose a random point $G = (G_x, G_y)$ on the curve C.
   This is the "generator" (but there is no guarantee that the group is cyclic).
4. Choose a random multiplier $N \in \{1, \ldots, |C|\}$.
   The exact value of $|C|$ is not easy to compute (so an approximation is required in most cases).

5. Compute $P = N * G$.
   This is the public point.

The *public* keys are (q, A, B, G, P) and the *secret* key is N.

Encryption For Alice to encrypt a message $x \in \mathcal{G}$, she performs these steps:

1. Choose a random multiplier M.
2. Compute $\alpha = M * G$. We call this the *half-mask*.
3. Compute $\omega = M * P$. We call this the *full-mask*.
4. Compute $y = x + \omega$.
5. Send the ciphertext pair (y, α).

So, $\mathrm{Enc}(x) = (x + \omega, \alpha)$. Note that the encryption is probabilistic since M is chosen randomly for each message (which will mask a repeated message). Also, $\omega = N * \alpha$ and Bob can recover the full-mask using his secret multiplier N.

Decryption For Bob to decrypt the ciphertext pair (y, α), he simply computes $\mathrm{Dec}(y, \alpha) = y - N * \alpha$.

Koblitz salting In implementing the above system, the mapping from a plaintext message (say an ASCII character) to a point on the curve requires some "salting" since not all choices of x lead to points on the curve (since $x^3 + Ax + B$ must be a quadratic residue modulo q). Koblitz suggested the following method. Fix a padding length K and let the padded message be

$$m = Kx + j, \quad \text{where } j = 0, \ldots, K - 1$$

With high probability (in practice), one of the choices of j will lead to m being a quadratic residue modulo q. To decrypt, Bob simply removes the low-order bits by computing $\lfloor m/K \rfloor$.