Elliptic Curve Cryptography

Tomislav Medan, Justin Singh-Mohudpur, and Youzhi Wang
Advisor: Garo Sarajian
Directed Reading Program

Introduction

The ancient practice of cryptography concerns how two parties can securely send and receive private information in a way that adversarial third parties cannot interpret or interfere with the messaging. Cryptography has undergone a variety of changes in modern times due to the advent of computers and the introduction of public-key cryptosystems and elliptic curve-based tools. In 1985, Neal Koblitz and Victor Miller[1] independently suggested using elliptic curves to provide the structure for cryptosystems and starting in the mid-2000s, these ideas became widely implemented, in large part due to it needing smaller key sizes to attain similar levels of security when compared to more traditional cryptosystems.

Objectives

We introduce elliptic curves and describe the discrete logarithm problem in both the elliptic curve and traditional Z/pZ settings. We then describe Shanks' algorithm[1] [2] for solving the discrete logarithm problem, before finally turning to an example of encryption and decryption of a message in both settings.

Definitions

Cryptosystem: A collection of algorithms that outline the encryption and decryption processes.

Elliptic curve (denoted E): The graph of a nonsingular cubic equation in two variables defined over a field K.

Point on the elliptic curve E: Either the identity element, denoted ∞, or an ordered pair $(x, y) \in K^2$ that satisfies the equation given by the elliptic curve.

Elliptic Curve Arithmetic

Point Addition and Doubling

Let $P, Q \in K^2$ be distinct points, such that $P = (x_p, y_p), \quad Q = (x_q, y_q)$. Then, to find a point $R = (x_r, y_r)$ on our elliptic curve such that

$$P + Q = R,$$

we first calculate the slope λ of the secant line intersecting at P and Q, $\lambda = \frac{y_q - y_p}{x_q - x_p}$. Then, in point-slope form, notice that we can write $y - y_p = \lambda(x - x_p)$. Equivalently, this gives that $y = \lambda x - \lambda x_p + y_p$. If we define $\beta = y_p - \lambda x_p$, we have an equation of the form

$$y = \lambda x + \beta.$$

Notice that squaring this equation makes it equal to our elliptic curve, such that $(\lambda x + \beta)^2 = x^3 + ax + b$. Computing this equation out gives that

$$0 = x^3 - \lambda^2 x^2 - 2\lambda x\beta - \beta^2 + ax + b,$$

which has three roots, particularly, x_p, x_q, x_r. As a result, the coefficient of x^2 is the opposite sum of the roots, giving that $x_p + x_q + x_r = \lambda^2$, or equivalently, $x_r = \lambda^2 - x_p - x_q$. So, we have x_r. To find y_r, we substitute x_r into our point-slope equation for the secant line and reflect it, such that $y_r = \lambda(x_p - x_r) - y_p$. Note that, if we want to add a point to itself, which we call "Point Doubling," then instead of the secant line, we find the line tangent to our point P, where we calculate the slope of the tangent line as $\lambda = \frac{3x_p^2 + a}{2y_p}$, and then compute the remaining steps of point addition.

Point Negation and Identity

Point negation can be thought of as the additive inverse property of elliptic curve addition, such that for a point P, we have that $P + (-P) = \infty$. We can find the additive inverse of a point by negating the y-coordinate of P, such that $-(x_p, y_p) = (x_p, -y_p)$. Notice that this provides the idea that ∞ is the identity element of an elliptic curve, such that $P + \infty = P$.

Discrete Logarithm

Discrete Logarithm Problem

Let g be a primitive root for $(\mathbb{Z}/p\mathbb{Z})^\times$, and let h be a nonzero element of $(\mathbb{Z}/p\mathbb{Z})^\times$. The congruency and problem of finding an exponent x such that,

$$g^x \equiv h \pmod{p}$$

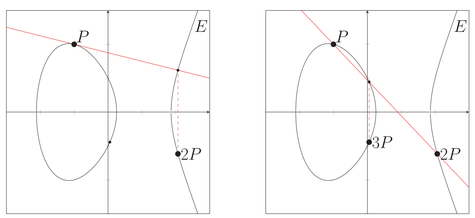
is defined as the *Discrete Logarithm Problem*, where x is the *discrete logarithm of h to the base g*, denoted by $\log_g(h)$, or when referred to as the *index*, $\mathrm{ind}_g(h)$. Take for example the discrete logarithm problem when $p = 11$, $g = 2$, and $h = 4$, such that

$$2^x \equiv 4 \pmod{11}.$$

Clearly, $x = 2$ is a solution to this problem – however, it is not the only solution. Since $p = 11$ is prime, the group $(\mathbb{Z}/11\mathbb{Z})^\times$ is cyclic, and by Fermant's little theorem, we get an equivalence class of solutions. This equivalence class, since $p = 11$ is prime, is defined by the expression $n + (p - 1)k$, where $n = 2$. Then, our equivalence class is $[2] = \{2, 12, 22, 32, \dots\}$,with each element of $[2]$ being a solution to this discrete logarithm problem. We can extend this idea to elliptic curves over finite fields, such as $(\mathbb{Z}/p\mathbb{Z})^\times$, by using a point P that generates the elliptic curve, and find the discrete logarithm k such that

$$kP = Q.$$

Notice that, in general, this problem involves doubling P, then continually performing addition with the points found thereafter. If we let $3P = Q$, then the below diagrams show the steps taken to find Q:



For harder problems, we can consider a much larger k, which would have its own equivalence class of solutions, and thus this problem potentially becomes more intractable.

References

[1] J. Hoffstein, J. Pipher, and J.H. Silverman. *An Introduction to Mathematical Cryptography*. Undergraduate Texts in Mathematics. Springer New York, 2014.

[2] L.C. Washington. *Elliptic Curves: Number Theory and Cryptography*. Discrete Mathematics and Its Applications. CRC Press, 2003.

Overview of Shanks Algorithm

Shanks Algorithm for $\mathbb{Z}/pZ$

Let G be a group and take $g \in G$ of order $N \geq 2$, where the order in $\mathbb{Z}/pZ$ is defined as the smallest possible integer n such that $g^n = e$ in $\mathbb{Z}/pZ$ and e is the identity element. The Babystep-Giantstep algorithm solves the discrete algorithm problem $g^x = h$ in $\mathbb{Z}/pZ$ for prime number p.

1. Let $n = 1 + [\sqrt{N}]$ which implies $n > \sqrt{N}$
2. Create two lists, namely:
   List 1: $e, g, g^2, g^3, ...g^n$
   List 2: $h, h * g^{-n}, h * g^{-2n}, h * g^{-3n}, ...h * g^{-n^2}$
3. Find a match between the two lists, say $g^i = hg^{-jn}$
4. Then $x = i + jn$ is a solution to $g^x = h$.

Shanks Algorithm for $E(F_q)$

Let our group G be the set of all points on elliptic curve $E(F_x)$ where F_x is the modular space our curve lives in. We are given $P, Q \in G$ and we are trying to solve a similar problem as before, namely $kP = Q$. Let N be the order of G. Shanks Algorithm then goes as prescribed here:

1. Fix an integer $m \geq \sqrt{N}$ and compute the multiplication mp
2. Create two lists, namely:
   List 1: iP for $0 \leq i < m$
   List 2: $Q - jmP$ for $j = 0, 1, ...m - 1$ where jmP is multiplication in $E(F_q)$
3. Find a match between the two lists, or where $iP = Q - jmP$
4. With our match, $Q = kP$ for $k \equiv i + jm$ (mod N)

Using Shanks Algorithm

For the problem $g^x = h$ mod(p), let us complete an example with $g = 232, h = 290, p = 331$. Using Lagrange's theorem, we find that $g = 232$ has order of 165 in F^*_{331}. Then, we compute our n value of the algorithm: $n = \lfloor\sqrt{165}\rfloor + 1 = 13$ and we now have the means to calculate our value u. We then calculate

$$u = g^{-n} = 232^{-13} = 232^{330-13} = 232^{317} = 272$$

in F^*_{331}. We then will create our lists of g^k for $k = 1, 2, ...$ and $h * u^k$ and find the matching values and indexes. In this example, this yields

$$232^{13} = 230 = 290 * 272^3$$

in F^*_{331}. Thus, using $272 = 232^{-13}$, we have $290 = 232^{13} * 272^{-3} = 232^{13} * (232^{13})^3 = 232^{52}$. And our answer of $x = 52$ solves the discrete log problem.

Now we will go into an example using the elliptic curve $y^2 = x^3 - 4x + 1$, group $G = E(F_{331})$, our base point $P = (2, 1)$ and $Q = (3, -4)$ to create the problem $(3, -4) = k(2, 1)$ mod(331). Through Hasse's theorem G has order $N = 324$ and so $m = 18$. We now will store our list of points iP for the values $0 \leq i \leq 18$. These points are:

$(2, 1), (12, 290), ...(330, 329)..., (83, 124)$.

We then compute $Q - jmp$ for $j = 0, 1, ...17$ and have a match: $(330, 329)$ at $15P$ in our first list and at $j = 1$ on our second list. Therefore $(3, -4) = (15 + 1 * 18)P = 34P$ and so $k = 34$

Elliptic Curve Cryptography Example

Here is a simplified version of how ECC can be used to encrypt and decrypt messages.
Alice wants to send "Hello" to Bob using the curve $y^2 = x^3 - 4x + 1$ over the prime 331 with base point (2,1). Alice chooses an encryption key that is coprime with the order of the group. The order is 324, she decides to use 5.

Encryption: Alice first assigns each letter of "Hello" to a point on the Elliptic Curve by using the letter's order in the alphabet as a multiple of the base point P(2,1):

$$\text{plaintext} = [8P, 5P, 12P, 12P, 15P]$$

Calculating the points would give:

$$\text{PlainTXT} = [(309, 258), (233, 208), (8, 207), (8, 207), (330, 329)]$$

To encrypt these points, Alice multiplies each point of PlainTXT by 5:

$$\begin{aligned}\text{CipherTXT} &= [5 * 8P, 5 * 5P, 5 * 12P, 5 * 12P, 5 * 15P] \\ &= [(286, 322), (315, 250), (54, 182), (54, 182), (21, 127)]\end{aligned}$$

With that the encryption is completed, CipherTXT is now considered as encrypted text that can now be sent to Bob.

Decryption: When Bob receives CipherTXT from Alice, he can decrypt the message by multiplying the list with the inverse of the number used to encrypt the message. In this case the inverse of 5 is 65, since,

$$5^{-1} \equiv 65 \pmod{324}$$

However, Bob can only do this because Alice had already told him the group order is 324. Without knowing the group order it would be very difficult to find the inverse of a number, since this involves solving a Discrete Log Problem described above.
To decrypt the message Bob multiplies each point from CipherTXT by 65.
The decrypted message, will be

$$\begin{aligned}\text{DecryptTXT} = [&65 * (5 * 8P), 65 * (5 * 5P), 65 * (5 * 12P), \\ &65 * (5 * 12P), 65 * (5 * 15P)]\end{aligned}$$

which comes out to be the same as PlainTXT,

$$\text{DecryptTXT} = [(309, 258), (233, 208), (8, 207), (8, 207), (330, 329)]$$

Matching the corresponding letter with each point would give the message "Hello".