[CS304] Introduction to Cryptography and Network Security

Course Instructor: Dr. Dibyendu Roy
Scribed by: Sanidhya Kumar (202151138)

Winter 2023-2024
Lecture (Week 10)

1 Modular Equations

We'll explore solving systems of linear equations to find x in the form:

$$a \cdot x \equiv b \mod m \text{ (Eq.1)}$$

To begin, let's express Eq.1 as:

$$a \cdot x - m \cdot y = b \text{ (Eq.2)}$$

where y is an integer. Utilizing Bezout's Identity:

$$a \cdot x_0 + m \cdot y_0 = \gcd(a, m) \text{ (Eq.3)}$$

where x_0 and y_0 can be determined using the Extended Euclidean Algorithm.

Eq.2 is solvable if and only if $\gcd(a, m)$ divides b. Assuming $\gcd(a, m)$ divides b, we have:

$$t \cdot \gcd(a, m) = b$$

By multiplying Eq.3 by t, we get:

$$a \cdot (t \cdot x_0) + m \cdot (t \cdot y_0) = t \cdot \gcd(a, m) \implies a \cdot X_0 + m \cdot Y_0 = b$$

Hence, given an equation to solve, we first verify if $\gcd(a, m)$ divides b. If so, a solution exists. Then, we find x_0 and y_0 using the Extended Euclidean Algorithm and multiply them by $t = \frac{b}{\gcd(a,m)}$ to obtain X_0 and Y_0.

Once X_0 and Y_0 are identified as solutions of Eq.2, we can substitute x and y as follows:

$$x = X_0 + \frac{m}{\gcd(a,m)} \cdot n$$
$$y = Y_0 + \frac{a}{\gcd(a,m)} \cdot n$$

where n is an integer. For any n, the derived x and y satisfy Eq.2, establishing them as the general solution.

Now, let's consider a system of two modular equations:

$$x \equiv a_1 \mod m_1 \text{ (Eq.1)}$$
$$x \equiv a_2 \mod m_2 \text{ (Eq.2)}$$

where m_1 and m_2 are coprime. We aim to find x satisfying both equations. If x is a solution of Eq.1, then:

$$x = a_1 + m_1 \cdot y \text{ (Eq.3)}$$

If x is also a solution to Eq.2, then:

$$x \equiv a_2 \mod m_2$$

Substituting the value of x from Eq.3 into Eq.2, we obtain:

$$m_1 \cdot y \equiv (a_2 - a_1) \mod m_2 \text{ (Eq.4)}$$

Since $\gcd(m_1, m_2) = 1$, we find the solution to Eq.4 as:

$$y = y_0 + m_2 \cdot n$$

From Eq.3, we deduce:

$$x = (a_1 + m_1 \cdot y_0) + n \cdot m_1 \cdot m_2$$

If y_0 is known, let $x_0 = a_1 + m_1 \cdot y_0$, then:

$$x = x_0 + m_1 \cdot m_2 \cdot n$$
$$x \equiv x_0 \mod m_1 \cdot m_2$$

x_0 is congruent modulo to any x under modulo $m_1 \cdot m_2$. Since x is the general solution of the two equations, every solution of the given system of equations will always be congruent to x_0 under modulo $m_1 \cdot m_2$.

Chinese Remainder Theorem (CRT)

The Chinese Remainder Theorem (CRT) is a fundamental concept in number theory used to solve systems of simultaneous congruences.

Consider a system of congruences:

$$\begin{cases} x \equiv a_1 \pmod{m_1} \\ x \equiv a_2 \pmod{m_2} \\ \vdots \\ x \equiv a_n \pmod{m_n} \end{cases}$$

where $m_1, m_2, \ldots, m_n$ are pairwise coprime positive integers.

The CRT states that this system of congruences has a unique solution modulo $M = m_1 \cdot m_2 \cdots m_n$. Furthermore, the solution x can be expressed as:

$$x \equiv a_1 \cdot M_1 \cdot y_1 + a_2 \cdot M_2 \cdot y_2 + \cdots + a_n \cdot M_n \cdot y_n \pmod{M}$$

where $M_i = M/m_i$ and y_i is the modular multiplicative inverse of M_i modulo m_i, i.e., $M_i \cdot y_i \equiv 1 \pmod{m_i}$.

Uniqueness

Let's assume x' is another solution of the above system. Then we have $x' \equiv x \pmod{m_1, m_2, \ldots, m_r}$.

$$\begin{aligned} x' &\equiv x \pmod{m_1} \\ x' &\equiv x \pmod{m_2} \\ x' &\equiv x \pmod{m_3} \\ &\vdots \\ x' &\equiv x \pmod{m_r} \\ x' &\equiv x \pmod{m_1, m_2, \ldots, m_r} \end{aligned}$$

This implies that x' and x are congruent modulo each individual modulus m_i, and thus they are congruent modulo the product of all moduli.

2 Exploring Elliptic Curve Cryptography (ECC)

- **Introduction**: While RSA offers a straightforward approach to cryptography with the Square and Multiply Algorithm, Elliptic Curve Cryptography (ECC) introduces a novel concept.
- **Computations on Curves**: ECC operates on elliptic curves rather than integers, leading to the development of modern cryptographic techniques such as the Diffie-Hellman Key Exchange Algorithm and the Signature Algorithm.
- **Key Exchange**: ECC employs Elliptic Curve Diffie-Hellman (ECDH) for secure key exchange, offering enhanced security with smaller prime numbers compared to RSA.
- **Digital Signatures**: Signatures in ECC are generated using Elliptic Curve Digital Signature Algorithm (ECDSA), providing robust security while minimizing computational complexity.
- **Security Benefits**: ECC's utilization of elliptic curves enables better security using smaller prime numbers, making it a preferred choice over RSA in many applications.
- **Transition to Discrete Structures**: ECC's foundation lies in discrete systems, highlighting the importance of understanding real numbers as a precursor to exploring its discrete aspects.

Let's define two real numbers a and b such that:

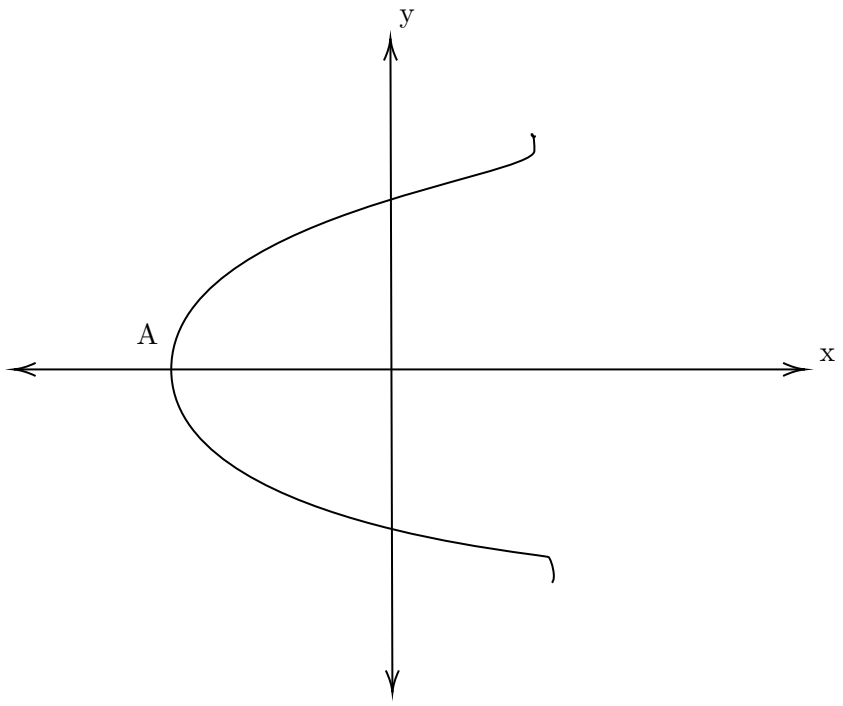
$$a, b \in \mathbb{R} \text{ and } 4a^3 + 27b^2 \neq 0$$

Consider the curve:

$$y^2 = x^3 + ax + b$$

where $(x, y) \in \mathbb{R}_2$. This curve is known as an Elliptic Curve.

When plotted, the curve exhibits two structures, one of which is depicted below:

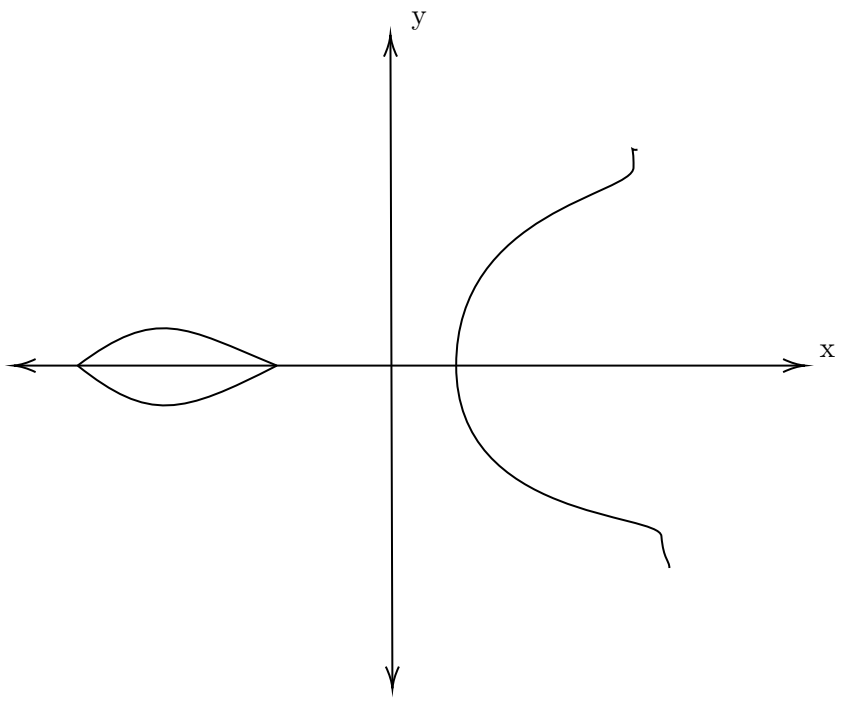


At point A, $y = 0$, implying $x^3 + ax + b = 0$ (Eq.1). This equation has three roots, which can either be:

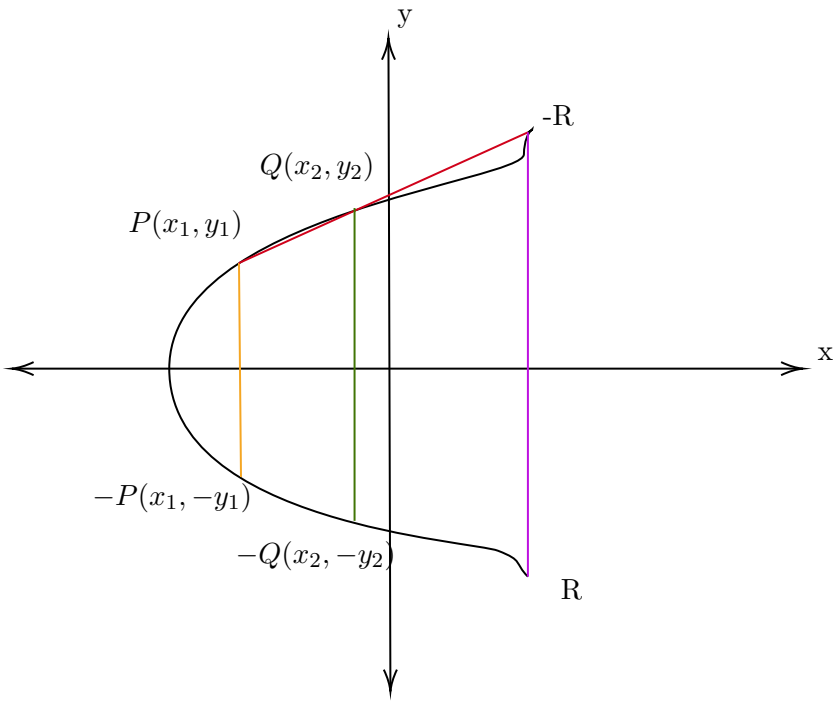
- Three real roots
- One real root and two complex roots

Eq.1 has three distinct roots if and only if $4a^3 + 27b^2 \neq 0$ (which can be real or complex). For the depicted curve, substituting $y = 0$ yields one real root and two complex roots.

If Eq.1 has three real roots, the resulting curve appears as shown below:



Let's introduce some properties of the previously defined curve:



Consider two points P and Q on the curve. When joined with a straight line, they intersect the curve again at a point, denoted as -R. The point -X is the mirror image of X with the x-axis as the mirror. Alternatively, we can say that the perpendicular from point X to the x-axis intersects the curve again at point -X.

1. $P \boxed{+} Q = R$. The $\boxed{+}$ operation is a binary operator defined as follows: take two points, join them with a straight line. The line intersects the curve again, and the image of this point on the x-axis is the output point.
2. Θ, known as the point at infinity, is introduced. Joining P and -P results in a straight line parallel to the y-axis, intersecting the curve at one point, assumed to be the point of infinity.
3. $P \boxed{+} -P = \Theta$
4. $P \boxed{+} \Theta = P$
5. $(P \boxed{+} Q) \boxed{+} R = P \boxed{+} (Q \boxed{+} R)$
6. $P \boxed{+} Q = Q \boxed{+} P$

The associativity and commutativity of the $\boxed{+}$ operator can be proved graphically. Treating Θ as an identity element and $-P$ as the inverse of P, the curve with the $\boxed{+}$ operator forms a commutative group.

Suppose we need to find $P \boxed{+} P$. We draw the tangent to the curve at point P, and wherever this tangent intersects the curve again, its image is the result. So, $P \boxed{+} P = R$ implies $2P = R$.

Elliptic Curve Mathematics:

$$y^2 = x^3 + ax + b$$
$$4a^3 + 27b^2 \neq 0$$

Let us consider two points $P(x_1, y_1)$ and $Q(x_2, y_2)$. We have three cases:

1. $x_1 \neq x_2$, $y_1 \neq y_2$
2. $x_1 = x_2$, $y_1 = -y_2$
3. $x_1 = x_2$, $y_1 = y_2$

Case-1:

$$y = mx + c \text{ ...Eqn(a)}$$
$$m = \frac{y_2 - y_1}{x_2 - x_1}$$
$$c = y_1 - mx_1,\ c = y_2 - mx_2$$

All the points on this line will satisfy this equation of the straight line. Equation of the straight line (Eqn(a)) will cut the curve at a point, so we substitute the value of y in the curve equation:

$$y_2 = x_3 + ax + b$$
$$m^2x^2 + 2mxc + c^2 = x_3 + ax + b$$
$$x^3 - m^2x^2 + (a - 2mc)x + (b - c^2) = 0$$

We already know that (x_1, y_1) and (x_2, y_2) will satisfy this equation. If x_3 is another solution of the above system, then:

$$x_1 + x_2 + x_3 = m^2$$
$$\implies x_3 = m^2 - x_1 - x_2$$
$$\text{We already know that } m = \frac{y_2 - y_1}{x_2 - x_1} = \frac{y_3 - y_1}{x_3 - x_1}$$
$$\implies y_3 = y_1 + m(x_3 - x_1)$$

So, we see that we obtained the coordinate of $R(x_3, y_3)$:

$$P \boxed{+} Q = R$$

Case-2: P = (x_1, y_1), Q = (x_2, y_2) where $x_1 = x_2$, $y_1 = -y_2$. In this case:

$$P \boxed{+} Q = \theta$$

Case-3: P = (x_1, y_1), Q = (x_2, y_2) where $x_1 = x_2$, $y_1 = y_2$:

$$y = mx + c$$
$$y_2 = x_3 + ax + b$$
$$\implies 2y\frac{dy}{dx} = 3x^2 + a$$
$$\implies \frac{dy}{dx} = \frac{3x^2 + a}{2y}$$
$$\left(\frac{dy}{dx}\right)_{(x_1, y_1)} = \frac{3x_1^2 + a}{2y_1} = m$$
$$c = y_1 - mx_1$$

Let us substitute in the curve:

$$y_2 = x_3 + ax + b$$
$$\implies (mx + c)^2 = x_3 + ax + b$$
$$x_1 + x_2 + x_3 = m^2$$
$$\implies x_3 = m^2 - x_1 - x_2$$
$$m = \frac{y_3 - y_1}{x_3 - x_1}$$
$$\implies y_3 = y_1 + m(x_3 - x_1)$$
$$\text{R} \rightarrow (x_3, -y_3)$$

Now, we will be considering the same curve in $\mathbb{Z}_{\mathbb{P}} \times \mathbb{Z}_{\mathbb{P}}$, where P is a prime number:

$$y^2 = x^3 + ax + b, \text{ where } (x, y) \in \mathbb{Z}_{\mathbb{P}} \times \mathbb{Z}_{\mathbb{P}} \text{ and } a, b \in \mathbb{Z}_{\mathbb{P}}$$
$$4a^3 + 27b^2 \neq 0 \mod P$$

Since we are now working on discrete values, we will not obtain this curve. We will obtain points.

Case-1:

$$x^3 = m^2 - x_1 - x_2$$
$$m = \frac{y_2 - y_1}{x_2 - x_1}$$

Now, here we don't divide, we take the inverse under mod P. Since x_2, x_1 are different values, $x_2 - x_1$ will be non-zero, and we will be able to find its inverse under mod P since P is prime, so its gcd with $(x_2 - x_1)$ will be 1. $m = (y_2 - y_1) \times (x_2 - x_1)^{-1} \mod P$

$$\implies y_3 = y_1 + m(x_3 - x_1) \in \mathbb{Z}_{\mathbb{P}}$$

2.0.1 Elliptic Curve Diffie-Hellman (ECDH)

Let us consider a scenario where Alice and Bob want to exchange messages. They have a curve E and a point P, and (E, P) is public.

(E, P)

Alice		Bob
secret: a		secret: b
public: aP		public: bP
	aP →	
	← bP	
$a(bP)$		$b(aP)$
abP		baP

$$abP = baP$$

Shared secret key $= abP$

In the above scenario, E and P were public while a and b were secret. Since a, b, P are all discrete, we can find aP (a times P), bP (b times P), and so on. Since they are discrete, abP and baP are the same. Since both Alice and Bob finally reached the same point on the curve, they have successfully exchanged messages.

Note: Security of ECDH depends on the fact that finding xP from P is computationally difficult. This hard problem is known as the **Discrete Log Problem on EC**.

RSA Signature

RSA Signature Encryption/Decryption:

$$\text{Encryption: } c = x^e \mod n, \quad \text{Decryption: } x = c^d \mod n$$

$$\text{Signature: } s = x^d \mod n, \quad \text{Verification: } x = s^e \mod n$$

Elliptic Curve Digital Signature Algorithm (ECDSA)

In ECDSA, a public key P corresponds to a secret key a, where the public key is represented as aP.

Properties of ECDSA

- ECDSA relies on a large prime number n, ensuring that $nG = 0$ on the elliptic curve, where G is the base point, and $(n-1)G \oplus G = nG$.

Signature Generation Process

1. Compute the hash e of the message m.
2. Select the leftmost L_n bits of e, where L_n is the bit length of n.
3. Choose a random integer k from the range $[1, n-1]$.
4. Generate a key pair (x_1, y_1).
5. Calculate $r = x_1 \mod n$. If $r = 0$, repeat step 1.
6. Calculate $s = k^{-1}(z + r \cdot d_A) \mod n$, where d_A is the secret key. If $s = 0$, repeat step 1.
7. The signature on message m is (r, s).

Verification Process by Bob

1. Ensure that the public key Q_A is not equal to 0.
2. Verify if Q_A lies on the elliptic curve.
3. Verify if $n \cdot Q_A = n \cdot d_A \cdot a$, where $Q_A = d_A \cdot G$.

Additional Verification Steps

1. Check if r and s are within the range $[1, n-1]$.
2. Compute e by hashing the message m.
3. Select the leftmost bits of e.
4. Compute $u_1 = z \cdot s^{-1} \mod n$ and $u_2 = r \cdot s^{-1} \mod n$.
5. Calculate the point $(x_2, y_2) = u_1 \cdot a + u_2 \cdot Q_A$. If $(x_2, y_2) = 0$, the signature is invalid.
6. Verify if $r \equiv x_2 \mod n$. If this condition holds, the signature is valid; otherwise, it is invalid.
7. Recompute e using $u_1 \cdot a + u_2 \cdot Q_A$. If the result matches the original hash e, the verification is successful.