[CS304] Introduction to Cryptography and Network Security

Course Instructor: Dr. Dibyendu Roy — Winter 2023-2024
Scribed by: Raghav Agiwal (202151124) — Lecture (Week 10)

1 Chinese Remainder Theorem

The Chinese Remainder Theorem (CRT) is a mathematical principle used to solve systems of congruences. Given a system of equations in the form:

$$\begin{aligned} x &\equiv a_1 \pmod{m_1} \\ x &\equiv a_2 \pmod{m_2} \\ &\vdots \\ x &\equiv a_r \pmod{m_r} \end{aligned}$$

where $m_1, m_2, \ldots, m_r$ are pairwise coprime integers, CRT guarantees the existence of a unique solution modulo $M = m_1 \cdot m_2 \cdot \ldots \cdot m_r$.

To solve such a system, we introduce the concept of δ_j for each $1 \leq j \leq r$:

$$\delta_j = \begin{cases} 1, & \text{mod } m_j \\ 0, & \text{mod } m_i, \text{ if } i \neq j \end{cases}$$

Expressing the solution x as a summation involving a_j and δ_j, we get:

$$x = \sum_{j=1}^{r} a_j \cdot \delta_j$$

By expanding x, we find:

$$x = \delta_1 \cdot a_1 + \delta_2 \cdot a_2 + \ldots + \delta_r \cdot a_r \quad \text{(Eq. 1)}$$

For each equation $x \equiv a_j \pmod{m_j}$, taking modulus m_j of x leads to:

$$x \equiv (\delta_1 \cdot a_1 + \delta_2 \cdot a_2 + \ldots + \delta_r \cdot a_r) \pmod{m_j}$$

Since δ_i is 0 for $i \neq j$ and 1 for j, we obtain $x \equiv a_j \pmod{m_j}$, proving that x is a solution.

To compute δ_j for each j, we determine $M = m_1 \cdot m_2 \cdot \ldots \cdot m_r$ and find the multiplicative inverse b_j of $M \pmod{m_j}$. Then, δ_j is given by:

$$\delta_j = \frac{M}{m_j} \cdot b_j$$

CRT ensures uniqueness by demonstrating that any other solution x' is congruent to $x \pmod{M}$, thus proving the uniqueness of the solution.

This theorem is a fundamental concept in number theory and has applications in various areas of mathematics and computer science.

1.1 Explanation of the Chinese Remainder Theorem

The Chinese Remainder Theorem (CRT) offers a method to solve a system of congruences, providing a unique solution under certain conditions.

Suppose we have a system of congruences:

$$\begin{aligned} x &\equiv a_1 \pmod{m_1} \\ x &\equiv a_2 \pmod{m_2} \\ &\vdots \\ x &\equiv a_r \pmod{m_r} \end{aligned}$$

where $m_1, m_2, \ldots, m_r$ are pairwise coprime integers. CRT ensures the existence of a unique solution modulo $M = m_1 \cdot m_2 \cdot \ldots \cdot m_r$.

To find the solution, we define δ_j for each $1 \leq j \leq r$ as follows:

$$\delta_j = \begin{cases} 1, & \text{if } i \equiv j \pmod{m_j} \\ 0, & \text{if } i \not\equiv j \pmod{m_j} \end{cases}$$

The solution x can be expressed as:

$$x = \sum_{j=1}^{r} a_j \cdot \delta_j$$

We then calculate δ_j for each j as:

$$\delta_j = \frac{M}{m_j} \cdot b_j$$

where b_j is the multiplicative inverse of $M \pmod{m_j}$.

To ensure uniqueness, we verify that dividing δ_j by m_j yields a remainder of 1, and dividing by any m_i, where $i \neq j$, results in a remainder of 0.

Finally, we compute x as:

$$x = \sum_{j=1}^{r} a_j \cdot \delta_j$$

Following these steps guarantees a unique solution modulo M.

1.2 Explanation of Uniqueness in the Chinese Remainder Theorem

The uniqueness of the solution in the Chinese Remainder Theorem (CRT) ensures that there is only one solution modulo the product of the moduli when the given system of congruences meets certain criteria.

Assume we have a solution x' to the given system of congruences, and x_0 is another solution. From the CRT, we know that if x_0 is a solution, then every solution will be congruent to x_0. Therefore, x' must be congruent to x_0 modulo the product of the moduli: $x' \equiv x_0 \pmod{(m_1 \cdot m_2 \cdot \ldots \cdot m_r)}$.

Since x and x' are solutions to the system of equations, they satisfy each individual congruence:

$$\begin{aligned} x &\equiv a_i \pmod{m_i} \\ x' &\equiv a_i \pmod{m_i} \end{aligned}$$

Subtracting these equations, we find:

$$x' - x \equiv 0 \pmod{m_i} \Rightarrow x' \equiv x \pmod{m_i}, \quad 1 \leq i \leq r$$

Since $x' - x$ is divisible by each m_i and the moduli are pairwise coprime, we can conclude that x' is congruent to x modulo the product of the moduli: $x' \equiv x \pmod{(m_1 \cdot m_2 \cdot \ldots \cdot m_r)}$.

Therefore, the solution is unique under modulo $(m_1 \cdot m_2 \cdot \ldots \cdot m_r)$.

2 Elliptic Curve Cryptography

Elliptic Curve Cryptography (ECC) is a branch of cryptography that utilizes the properties of elliptic curves to provide secure communication and digital signatures. Unlike RSA, which relies on integer factorization, ECC operates on a curve defined over a finite field, offering similar security with smaller key sizes.

In ECC, computations are performed on points that lie on an elliptic curve defined by the equation:

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

where a and b are real numbers and $4a^3 + 27b^2 \neq 0$. This equation describes the elliptic curve, and points (x, y) on this curve satisfy the equation.

ECC provides various cryptographic primitives, including key exchange and digital signatures:

1. **Elliptic Curve Diffie-Hellman (ECDH)**: This key exchange algorithm allows two parties to establish a shared secret over an insecure channel. By exchanging public keys derived from points on the elliptic curve, they can compute a shared secret without exposing it to eavesdroppers.

2. **Elliptic Curve Digital Signature Algorithm (ECDSA)**: ECDSA enables the creation and verification of digital signatures using elliptic curve mathematics. It involves generating a signature from a message using the signer's private key and verifying the signature's authenticity using the corresponding public key.

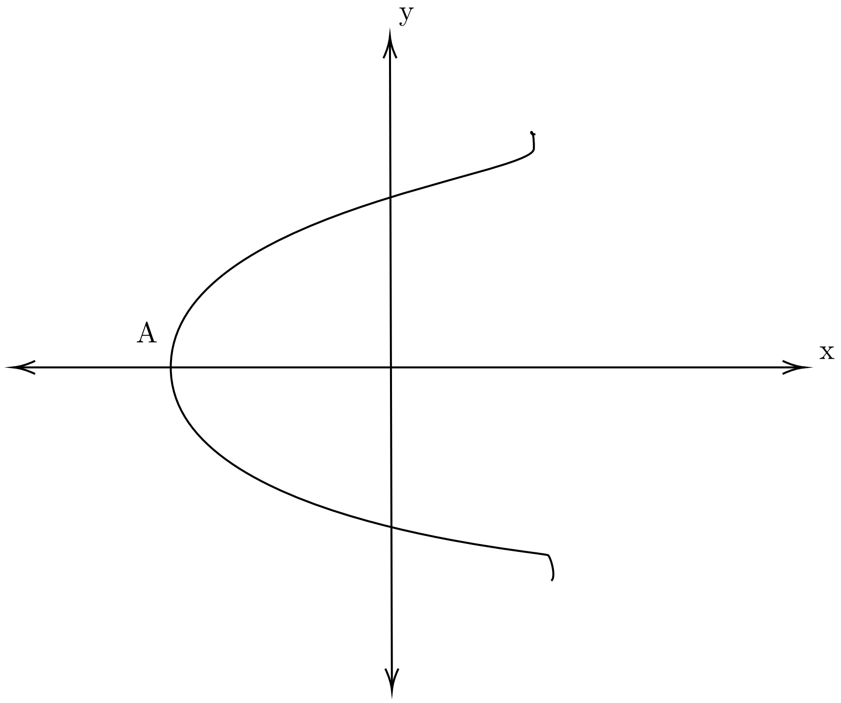
ECC offers several advantages over traditional cryptographic algorithms like RSA:

- **Smaller Key Sizes**: ECC provides comparable security to RSA but with smaller key sizes, making it more efficient in terms of storage and computation.

- **Better Security**: ECC's security relies on the difficulty of the elliptic curve discrete logarithm problem, which is believed to be more resistant to attacks compared to the integer factorization problem used in RSA.

- **Faster Operations**: ECC operations, such as point multiplication, are computationally more efficient, leading to faster cryptographic operations.

In summary, Elliptic Curve Cryptography leverages the mathematical properties of elliptic curves to offer secure and efficient cryptographic primitives for key exchange and digital signatures, making it a popular choice for securing modern communication systems.



2.1 Analysis of the Curve $y^2 = x^3 + ax + b$

Consider the curve defined by the equation $y^2 = x^3 + ax + b$. When $y = 0$, we have:

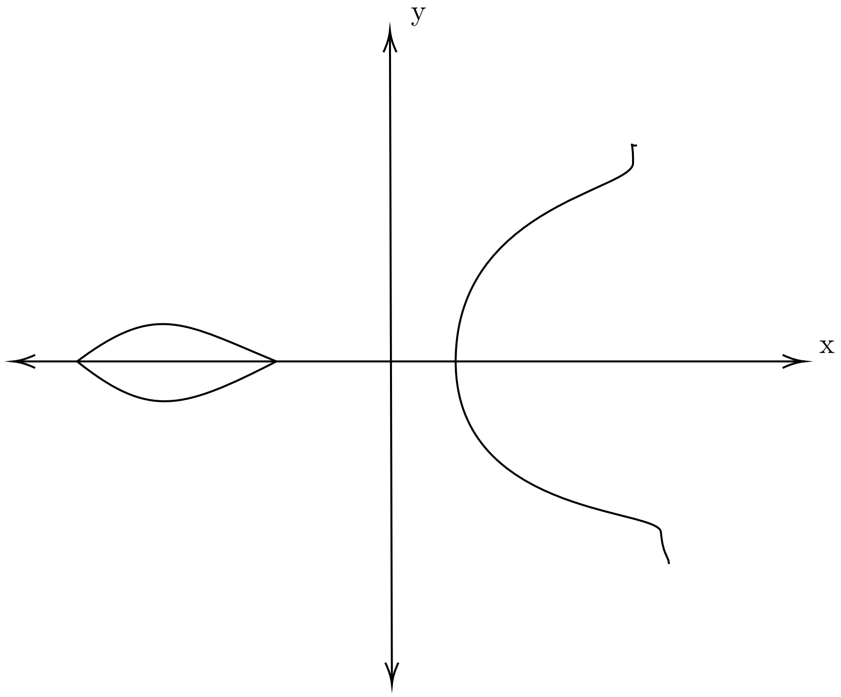
$$x^3 + ax + b = 0 \quad \text{(Eq. 1)}$$

This equation will have three roots, which can take one of the following forms:

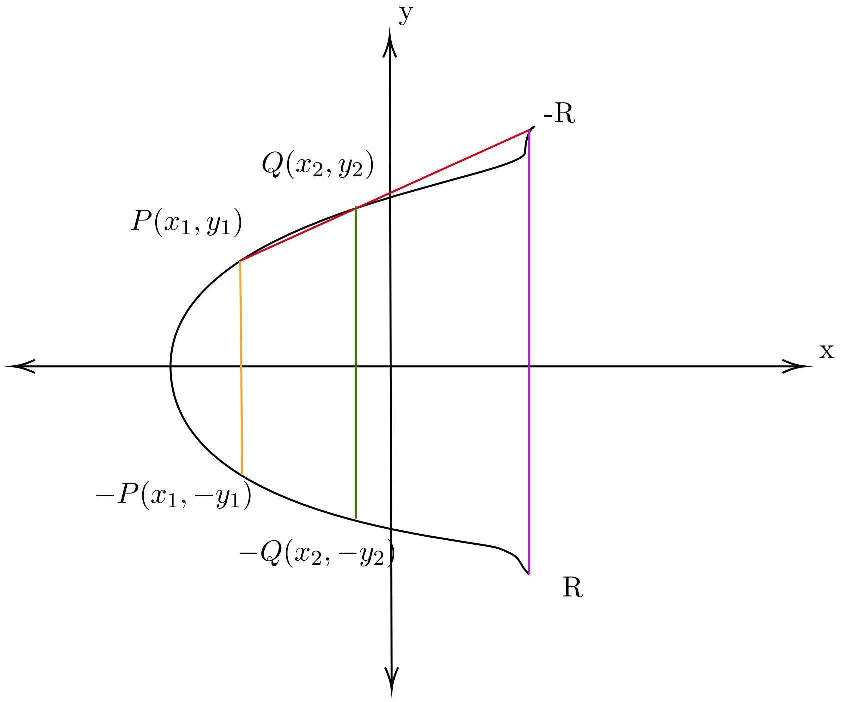
- Three real roots
- One real root and two complex roots

Eq. 1 will have three distinct roots if and only if $4a^3 + 27b^2 \neq 0$, which can be real or complex. By analyzing the curve and setting $y = 0$, we observe that it will have only one real root and two complex roots.

If we consider the case of three real roots for Eq. 1, the curve will have the following appearance:



Let us define some properties on the curve we defined before.



2.2 Properties of Points on the Curve

Consider an elliptic curve defined by an equation, and let P and Q be two points on this curve.

1. **Point Addition**: If we join points P and Q on the curve using a straight line, it will intersect the curve at another point, denoted as R. The point $-R$ is the mirror image of R with respect to the x-axis. Alternatively, we can say that the perpendicular from point R to the x-axis

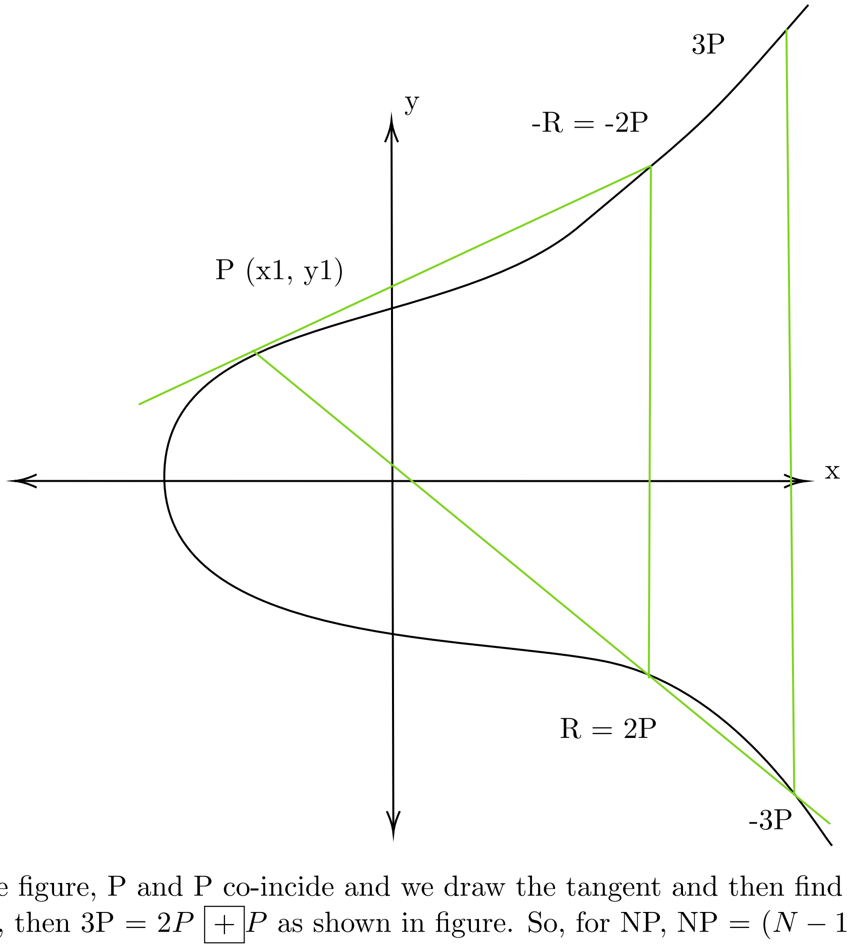
intersects the curve at point $-R$. The addition operation $P + Q = R$ is defined as follows: take the two points, join them using a straight line, and the image of the intersection point with the x-axis is the result.

2. **Point at Infinity (Θ)**: Θ is known as the point at infinity. If we join a point P with its mirror image $-P$, the resulting straight line will be parallel to the y-axis. Since the elliptic curve is infinite, we assume that the line intersects the curve at one point, which is designated as the point at infinity.

3. Properties of Point Addition:
   - $P + (-P) = \Theta$
   - $P + \Theta = P$
   - Associativity: $(P + Q) + R = P + (Q + R)$
   - Commutativity: $P + Q = Q + P$

The associativity and commutativity of the addition operation can be demonstrated graphically. The point Θ serves as the identity element, and $-P$ acts as the inverse of P. Thus, the curve with the addition operation forms a commutative group.

Suppose, we have to find $P \boxed{+} P$, then what we do is that we draw the tangent to the curve at P, and wherever the tangent cuts the curve again, its image is the result, it my result. $P \boxed{+} P = R \implies 2P = R$. Let us see in the graph:



In the above figure, P and P co-incide and we draw the tangent and then find its image. If we have to find 3P, then 3P = $2P \boxed{+} P$ as shown in figure. So, for NP, NP = $(N-1)P \boxed{+} P$.

3 Mathematical Aspects

3.1 Elliptic Curve

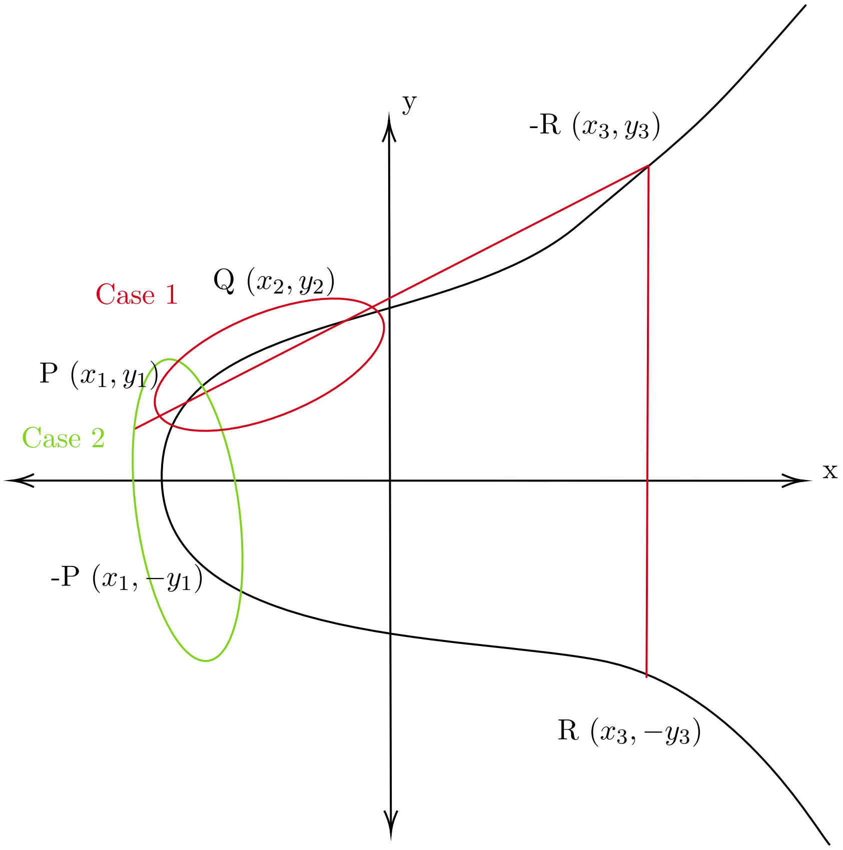
An elliptic curve is defined by the equation:

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

where a and b are constants, and $4a^3 + 27b^2 \neq 0$.

Let's consider two points $P(x_1, y_1)$ and $Q(x_2, y_2)$. There are three cases:

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



3.2 Elliptic Curve Operations

3.2.1 Case-1: Points P and Q on a Line

Consider the equation of a line:

$$y = mx + c \quad \text{(Eqn (a))}$$

where $m = \frac{y_2 - y_1}{x_2 - x_1}$ and $c = y_1 - mx_1 = y_2 - mx_2$. All points on this line satisfy Eqn (a).

This line intersects the curve at some point. Substituting $y = mx + c$ into the curve equation yields:

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

$$(mx + c)^2 = x^3 + ax + b$$

$$m^2x^2 + 2mxc + c^2 = x^3 + ax + b$$

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

Since (x_1, y_1) and (x_2, y_2) satisfy this equation, another solution (x_3, y_3) can be obtained:

$$x_1 + x_2 + x_3 = m^2$$

$$x_3 = m^2 - x_1 - x_2$$

And:

$$y_3 = y_1 + m(x_3 - x_1)$$

So, $R(x_3, y_3) = P + Q$.

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 $x_1 = x_2, y_1 = y_2$

$$\begin{gathered}
y = mx + c \\
y_2 = x_3 + ax + b \\
\implies 2y\tfrac{dy}{dx} \;=\; 3x^2 + a \\
\implies \tfrac{dy}{dx} \;=\; \tfrac{3x^2+a}{2y} \\
(\tfrac{dy}{dx})_{(x_1,y_1)} \;=\; \tfrac{3{x_1}^2+a}{2y_1} \;= \text{m} \\
\text{c} = y_1 - mx_1
\end{gathered}$$

Let us substitute in curve

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

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.

3.3 Case-1: Points P and Q on a Line

For $P(x_1, y_1)$ and $Q(x_2, y_2)$, where $x_2 \neq x_1$, the slope m between P and Q is given by:

$$m = \frac{y_2 - y_1}{x_2 - x_1}$$

Now, to obtain the x-coordinate of the third point x_3, we take the inverse of $x_2 - x_1$ modulo P. Since x_2 and x_1 are distinct, $x_2 - x_1$ will be non-zero, and its inverse modulo P exists because P is prime:

$$m = (y_2 - y_1) \times (x_2 - x_1)^{-1} \mod P$$

Thus, the y-coordinate y_3 of the third point R is given by:

$$y_3 = y_1 + m(x_3 - x_1) \in ZP$$

3.4 Elliptic Curve in Discrete Domain

Now consider the curve in the discrete domain $ZP \times ZP$, where P is a prime number:

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

with $(x, y) \in ZP \times ZP$ and $a, b \in ZP$. Ensure that $4a^3 + 27b^2 \neq 0 \mod P$. In this discrete context, points are obtained instead of the curve itself.

4 Elliptic Curve Diffie-Hellman (ECDH)

Consider the scenario where Alice and Bob want to exchange messages securely. They both have access to a public curve E and a base point P, denoted as (E, P), which is publicly known.

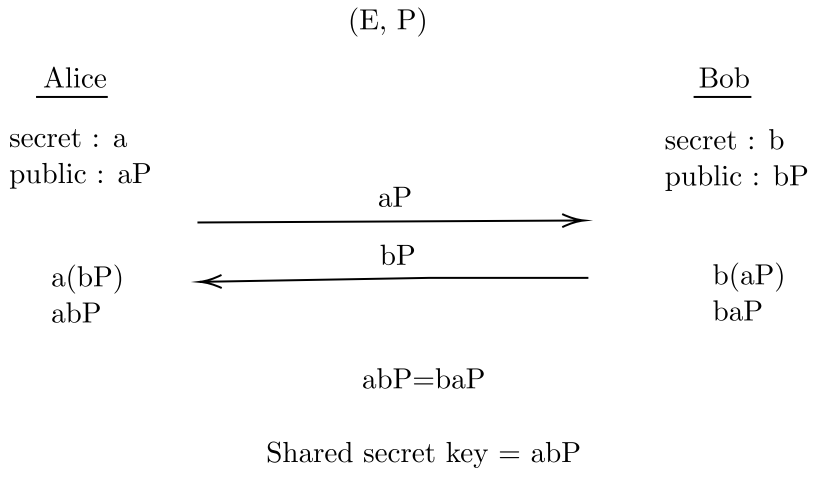
4.1 Key Generation

Alice:

- **Secret Key:** a
- **Public Key:** aP

Bob:

- **Secret Key:** b
- **Public Key:** bP



4.2 Key Exchange

To generate a shared secret key, Alice and Bob perform the following computations:

$$\begin{aligned} \text{Alice:} \quad & a(bP) = abP \\ \text{Bob:} \quad & b(aP) = baP \end{aligned}$$

Since the operation is commutative in elliptic curve arithmetic, $abP = baP$. Thus, both Alice and Bob end up with the same shared secret key $abP = baP$.

4.3 Security Considerations

The security of ECDH relies on the computational difficulty of finding xP from P, known as the Discrete Logarithm Problem on Elliptic Curves.

Note: The elliptic curve calculations can be visualized and executed using tools like Jupyter Notebook, with code available from resources like SageMath.

5 Elliptic Curve Digital Signature Algorithm (ECDSA)

In ECDSA, cryptographic operations are performed using elliptic curves. Let (E, P) denote the public keys, where E represents the elliptic curve and P is a base point on the curve. Each party, say Alice, possesses a secret key d_A and a corresponding public key $Q_A = d_A \cdot P$.

5.1 Signature Generation

Given a message m to be signed, the signer (Alice) computes the signature s using her secret key d_A:

$$s = k^{-1}(z + r \cdot d_A) \mod n$$

where k is a randomly generated number, z is the hash of the message, r is the x-coordinate of the point resulting from $k \cdot P$ on the curve, and n is a large prime number associated with the curve.

5.2 Signature Verification

The verifier (Bob) receives the message m, along with the signature s and the public key Q_A. Bob then computes:

$$r = \text{the } x\text{-coordinate of} \left((s \cdot Q_A - z \cdot G) \cdot w \right)$$

where G is the base point, w is the multiplicative inverse of s modulo n, and z is the hash of the message. If r matches the x-coordinate of the original signature, the signature is considered authentic.

ECDSA ensures message integrity and authenticity in cryptographic communications, offering a robust mechanism for digital signature generation and verification.

(E, G, n) is known to everyone

Alice	Bob
secret : d_A public : $Q_A = d_A G$	
messange : m	

5.3 Signature Generation

1. **Hash Calculation:** Compute the hash of the message m, denoted as e.
2. **Bit Extraction:** Extract L_n leftmost bits of e, where L_n is the bit length of n.
3. **Random Number Generation:** Generate a random number K from the range $[1, n-1]$.
4. **Point Computation:** Compute the point $(x_1, y_1) = K \cdot G$, where G is the base point on the curve.
5. **Signature Components:** - Calculate $r = x_1 \mod n$. If $r = 0$, return to step 3. - Compute $s = K^{-1} \cdot [Z + r \cdot d_A] \mod n$, where d_A is the signer's secret key. - If $s = 0$, return to step 3. 6. **Signature Generation:** Generate the signature (r, s) for the message m.

5.4 Signature Verification (by Bob)

1. **Validity Checks:** - Verify that $Q_A \neq 0$. - Check if Q_A lies on the curve E. - Ensure that $n \cdot Q_A = d_A \cdot (n \cdot G) = 0$. 2. **Verification Steps:** - Verify that r and s are within the range $[1, n-1]$. - Compute $e = \text{Hash}(m)$. - Extract L_n leftmost bits of e to obtain Z^-. - Calculate $u_1 = Z^- \cdot s^{-1} \mod n$ and $u_2 = r \cdot s^{-1} \mod n$. - Perform point addition: $(x_2, y_2) = u_1 \cdot G + u_2 \cdot Q_A$. - If $(x_2, y_2) = 0$, the signature is invalid. - Check if $r \equiv x_2 \mod n$. If true, the signature is valid; otherwise, it's invalid.

5.5 Proof of Validity

The verification involves computing $c = u_1 \cdot G + u_2 \cdot Q_A$, where c is the computed point. By substituting the value of s^{-1}, it can be shown that $c = K \cdot G$, thus proving the validity of the signature.