<table>
<tr><th colspan="3">TOPIC PLAN</th></tr>
<tr><td>Partner organization</td><td colspan="2">Belgrade Metropolitan University</td></tr>
<tr><td>Topic</td><td colspan="2">Blockchain Technology in Data Protection</td></tr>
<tr><td>Lesson title</td><td colspan="2">Elliptic Curve Cryptography</td></tr>
<tr><td>Learning objectives</td><td>Students can understand the definition of elliptic curves
Students can understand the propertied of point addition in elliptic curves
Students can develop programs for ECC key exchange
Students can apply elliptic curve cryptography in Diffie-Hellman key exchange
Students can apply ECC in blockchain technology</td><td rowspan="3">Methodology
☑ Modeling
☐Collaborative learning
☐ Project based learning
☐ Problem based learning

Strategies/Activities
☐Graphic Organizer
☑ Think/Pair/Share
☐Discussion questions

Assessment for learning
☑ Observations
☑ Conversations
☐Work sample
☐Conference
☐Check list
☐Diagnostics</td></tr>
<tr><td>Aim of the lecture / Description of the practical problem</td><td>The aim of the lecture is to introduce students to the basic principles of elliptic curve cryptography (ECC). Students firstly review the discrete logarithm problem in asymmetric cryptography, with an overview of the Diffie-Hellman key-exchange algorithm, which focuses on modular arithmetic. Afterwards, elliptic curves with their properties are described. The key exchange algorithm is then applied using multiplicative addition of points. Finally, students are given examples of the application of elliptic curve cryptography in Blockchain technology.

Students are tasked with writing a computer program which displays elliptic curves with different parameters, and writing a program for key exchange using elliptic curve cryptography.</td></tr>
<tr><td>Previous knowledge assumed:</td><td>Symmetric cryptography
Asymmetric cryptography
Basics of Linear algebra
Basics of Calculus
Programming in Python/Java</td></tr>
</table>

Introduction / Theoretical basics	**Asymmetric cryptography** The basic concept is that a public key may exist to encrypt the data, while a private key is used to decrypt the data. This concept can be achieved with a set of algorithms that can be easy to implement in one direction, while it can be extremely difficult to implement in the inverse direction. The first, and still most used algorithm is the RSA algorithm. The security of this algorithm relied on simple computation in one side (the multiplication of large prime numbers) while the inverse (factorization) is extremely complex. After the RSA algorithm, scientists have examined other mathematical-based cryptographic algorithms, besides factorization, which can be used in asymmetric cryptography. *Let a and b be real numbers, and let N be natural number. The security of RSA relies on the fact that it is difficult to find x such as* $$x^b \equiv b(\bmod N)$$ Such a problem can theoretically be solved with Shore's algorithm, which predicts that the factorization can be done in polynomial time if quantum computers are used. **Weierstrass form of elliptic curves** An elliptic curve E has a Weierstrass form has the following form: $$E: y^2 = x^3 + ax + b\ ,$$ where constants ***a*** and ***b*** fulfil the condition: $$4a^3 + 27b^2 \neq 0$$ This condition is the cube polynomial non-zero discriminate condition, which guarantees three different roots, which can in general be complex numbers. A Weierstrass clliptic curve is shown in Figure 1.	**Assessment as learning** ☐ Self-assessment ☐Peer-assessment ☐Presentation ☐Graphic Organizer ☑ Homework **Assessment of learning** ☑ Test ☑ Quiz ☐Presentation ☐Project ☐Published work

$y^2 = x^3 - x + 0.8849$

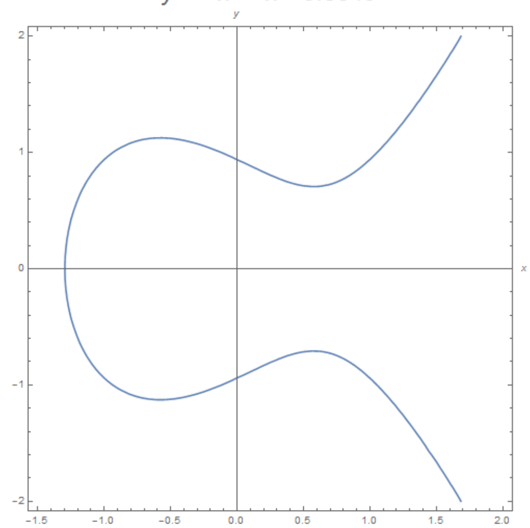

Figure 1

Koblitz form of elliptic curves

The Koblitz form of elliptic curve is a sub-type of the co-called generalized Weierstrass form. It is defined by the following equation:

$$E_a : y^2 + xy = x^3 + ax + 1$$

A Koblitz elliptic curve is shown in Figure 2

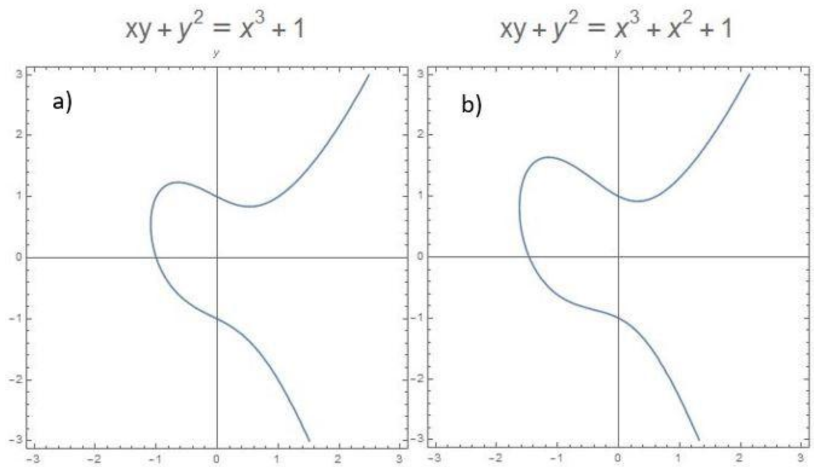

Figure 2

The discrete logarithm problem: formal definition

Let **a** be a primitive root of a finite group (a field with an associated unit element) of order **p** (**p** times applying a binary operation on a will yield a unit element), where **p** is a prime number and let **b** be a non-zero element of that group. The discrete logarithm problem is to find an exponent **c** such that:

$$a^c \equiv b \pmod p$$

The number c is called the discrete logarithm of b. Based on one of the properties of primitive roots, if a is a primitive root, c must be a natural number belonging to the segment from 0 to *p* − *2* (the given numbers are class representatives).

Action

Discussion with students to implement point addition in elliptic curves.

Point addition in elliptic curves

Before proceeding to a more detailed analysis, several definitions are introduced:

- Let G be set. A binary operation in that set is every function f : G2 → G, from the direct square G2 = G x G of the set G into the set G itself.

- Let G be a nonempty set and let B be a binary operation in G. An ordered pair (G, B) is called a groupoid.

- A groupoid (G, B) is called a semigroup if the operation B is associative.

- An element e of a groupoid (semigroup) is called a unit or neutral element if x B e = e B x = x for every x belonging to the groupoid (semigroup).

- Let us have a groupoid with unit element. An element y is an inverse element of x, if x B y = y B x = e. An element is invertible if it has an inverse element. As a consequence, it is easy to prove that in a semigroup with unit element, each element has exactly one inverse element, or none at all.

If the points Ti = (xi, yi) and Tj = (xj , yj) are known, where to start with the case where xi /= xj and yi /= yj , then the equation of the line through those points is:

$$y = m_{ij}\left(x - x_i\right) + y_i = m_{ji}\left(x - x_j\right) + y_j$$

where

$$m_{ij} = \frac{y_i - y_j}{x_i - x_j} = \frac{y_j - y_i}{x_j - x_i} = m_{ji}$$

and the coefficient of the direction of the straight line If we replace this equation with y and the expression for the elliptic curve, we get:

$$\begin{aligned} y^2 &= m_{ij}^2 x^2 + 2m_{ij}\left(y_i - m_{ij}x_i\right)x + \left(y_i - m_{ij}x_i\right)^2 = x^3 + ax + b \\ &\Rightarrow x^3 - m_{ij}^2 x^2 + \left(a - 2m_{ij}\left(y_i - m_{ij}x_i\right)\right)x - \left(\left(y_i - m_{ij}x_i\right)^2 - b\right) = 0 \end{aligned}$$

After arranging the expression, you will get a point that lies on the straight line and intersects the elliptic curve, i.e. will get:

$$y'_k = m_{ij}\left(x_k - x_i\right) + y_i = m_{ji}\left(x_k - x_j\right) + y_j$$

The point (xk , yk) belongs to the elliptic curve, so due to the symmetry y <-> −y it follows that the point Tk also belongs to the curve. Point addition is shown in Figure 3.

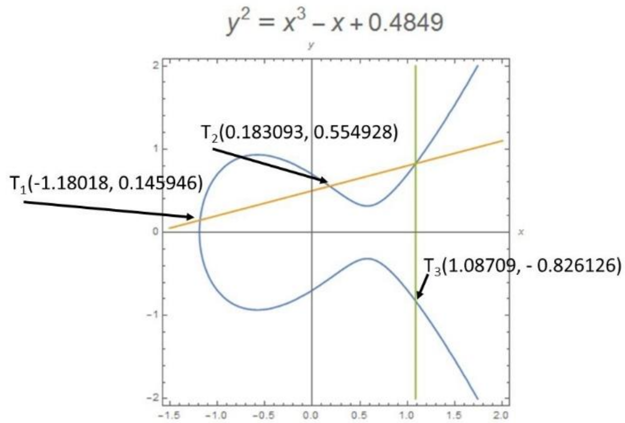

Figure 3

Exercise #1

Drawing elliptic curves in a 2D coordinate system in the Python programming language.

Write a function to display an elliptic curve with arbitrary parameters a and b in the Python programming language using the matplotlib and numpy libraries.

```
import numpy as np
import matplotlib.pyplot as plt
def main():
    a = -1
    b = 1

    y, x = np.ogrid[-5:5:100j, -5:5:100j]
    plt.contour(x.ravel(), y.ravel(), pow(y, 2) -
pow(x, 3) - x * a - b, [0])
    plt.grid()
    plt.show()

if __name__ == '__main__':
    main()
```

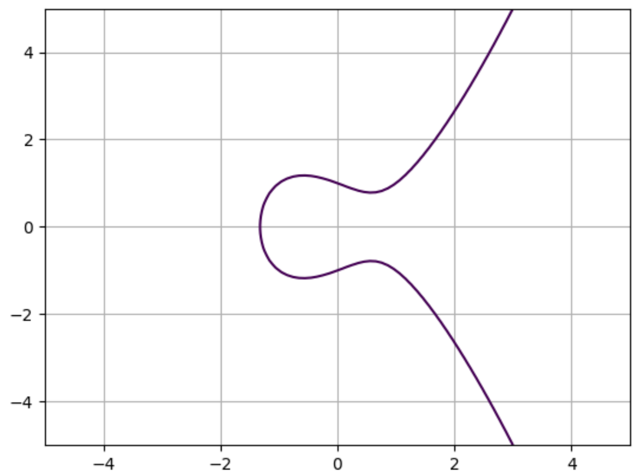

Figure 4 – Elliptic curve drawn in Python

Exercise #2

Application of the addition method in the Java programming language

In order to implement the addition method, which was introduced, we first implemented a class for a point called Point and it will represent a point on an elliptic curve.

The class fields are the x and y coordinates, which take double as the data type. A class constructor with x and y coordinates is created. In the Point class, the toString method has been updated and added so that we have a list of coordinates for a given point.

```java
public class Point {
    public double x, y;

    public Point(double x, double y) {
        this.x = x;
        this.y = y;
    }

    @Override
    public String toString() {
        return x + " - " + y;
    }
```

A class for elliptic curve cryptography called **EllipticCurveCryptography** is required.

In it, we have two main parameters called a and b, which correspond to the equation of an elliptic curve of the Weistrass type.

The Bitcoin network takes the values a = 0, b = 7 for parameters, so their equation is: y^2 = x^3 + 7.

```java
/**
 *
 * @author Bojana
 */
public class EllipticCurveCryptography {
    public double a;
    public double b;

    public EllipticCurveCryptography(double a, double b) {
        this.a = a;
        this.b = b;
    }
```

A point addition function called point addition is required. Its input arguments are the points P and Q, so we need to have their coordinates, which are also included in the given points method:

```
public Point point_addition(Point P, Point Q){
    double x1 = P.x;
    double y1 = P.y;
    double x2 = Q.x;
    double y2 = Q.y;
    double m;

    if (x1 == x2 && y1 == y2)
        m = (3*x1*x1+a) / (2*y1);
    else
        m = (y2-y1) / (x2-x1);

    double x3 = m*m - x1 - x2;
    double y3 = m*(x1-x3) - y1;

    return new Point(x3, y3);
}
```

There are two cases. Addition of points when point P is not the same as point Q, that is, they do not have the same coordinates, and the method of duplicating points when the coordinates of point P are equal in value to the coordinates of point Q. In the implementation, a check was made whether x1 = x2 and whether y1 = y2.

If the points are different, the addition operation is applied.

After checking, the x3 and y3 coordinates are updated, after which the function returns a new point that was generated with the x3 and y3 coordinates.

```
public Point point_addition(Point P, Point Q){
    double x1 = P.x;
    double y1 = P.y;
    double x2 = Q.x;
    double y2 = Q.y;
    double m;

    if (x1 == x2 && y1 == y2)
        m = (3*x1*x1+a) / (2*y1);
    else
        m = (y2-y1) / (x2-x1);
```

An instance of the elliptic curve cryptographic class is created in the main application class. Its parameters are set to zero for a and seven for parameter b. So it uses the same elliptic curve that is used in the Bitcoin network. After that, a new point was created, whose coordinates are one and one. Calling the print function will print the result of the point addition method, in which two identical points P and P are taken for the purposes of the exercise.

As the same point is used, the duplication method will be applied.

```
public Point point_addition(Point P, Point Q){
    double x1 = P.x;
    double y1 = P.y;
    double x2 = Q.x;
    double y2 = Q.y;
    double m;
```

```
/**
 *
 * @author Bojana
 */
public class Main {
    public static void main(String[] args) {
        EllipticCurveCryptography ecc = new EllipticCurveCryptography(0, 7);
        Point point = new Point(1, 1);
        System.out.println(ecc.point_addition(point, point));
```

```
t - ECC (run) ×
run:
0.25 - 0.125
BUILD SUCCESSFUL (total time: 0 seconds)
```

Homework

Write a key exchange program using elliptic curves in a programming language of your choice.

<table>
<tr><td>Materials / equipment / digital tools / software</td><td>The materials for learning are given as a part of references of the end from this topic plan;
Equipment: classroom, board, chalk;
Digital tools: computer with programming languages Python and Java, projector for slides;</td><td></td></tr>
<tr><td>Consolidation</td><td colspan="2">• The teacher's discussion with the students through appropriate questions;
• Independent solving of simple tasks by the students under the supervision of the teacher;
• Given of examples by the teacher for introducing a new concept in a cooperation and a discussion with the students;
• Assignment of homework by the teacher with a time limit until the next class.</td></tr>
<tr><td colspan="3">Reflections and next steps</td></tr>
<tr><td colspan="2">Activities that worked</td><td>Parts to be revisited</td></tr>
<tr><td colspan="2">After the class, the teacher according to his personal perceptions regarding the success of the class fills in this part.</td><td>Through the success of the homework done by the students, questions and discussion at the beginning of the next class, the teacher comes to the conclusion which parts of this class should be revised.</td></tr>
<tr><td colspan="3">References</td></tr>
<tr><td colspan="3">[1] Nemanja Zdravkovic, IT475 - Blockchain Technology in Data Protection, Authorized Lectures on Metropolitan University Belgrade eLearning platform – LAMS, 2021.
[2] D. Hankerson, A. J. Menezes, S. Vanstone, Guide to Elliptic Curve Cryptography, Springer, 2004.
[3] I. Bashir, Mastering Blockchain, Packt Publishing, 2017.
[4] Wolfram Demonstration Project, Addition of Points on an Elliptic Curve over the Reals, https://demonstrations.wolfram.com/AdditionOfPointsOnAnEllipticCurveOverTheReals/</td></tr>
</table>