FUTUREMATH ERASMUS+ PROJECT

Co-funded by the
Erasmus+ Programme
of the European Union

UNIVERZITET
METROPOLITAN
BEOGRAD

# IT475 - BLOKČEJN TEHNOLOGIJA U ZAŠTITI PODATAKA

## Elliptic curve cryptohraphy

Lekcija 16

PRIRUČNIK ZA STUDENTE

# IT475 - BLOKČEJN TEHNOLOGIJA U ZAŠTITI PODATAKA

Lekcija 16

## *ELLIPTIC CURVE CRYPTOHRAPHY*

# Uvod

## INTRODUCTION

*This lesson gives an introduction to elliptic curve cryptography and its application to blockchain technologies*

In this lesson, we will talk about elliptic curve cryptography.
Firstly, we will present the mathematical concepts of elliptic curves and present the two forms of elliptic curves: Weierstrass and Koblitz.
Afterwards, we will discuss the discrete logarithm problem with regards to cryptography, focusing on where elliptic curves can be used to solve this problem.

Next, the application of elliptic curves in asymmetric cryptography is presented, with the main operation being point addition.

Finally, we will present an implementation of the Diffie-Helman algorithm on elliptic curves, and compare the complexity with an equivalent modular arithmetic approach .

# Poglavlje 1

# The math of elliptic curves

## INTRODUCTION: ASYMMETRIC CRYPTOGRAPHY

### *The eras of cryptography*

The history of cryptography can be divided into two eras – the classical era and the modern era. The paradigm shift occurred with the introduction of asymmetric cryptography. These new algorithms were revolutionary are their security relied on number theory – these were the first algorithms which allowed secure transmission between two parties without a common (shared) key.

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.
The application of the RSA algorithm relies in the following statement:
*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*

$$x^b \equiv b \pmod N$$

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: INTRODUCTION

### *Eliptična kriva Vaještrasovog tipa zadoboljava jednačinu y^2 = x^3 + ax + b*

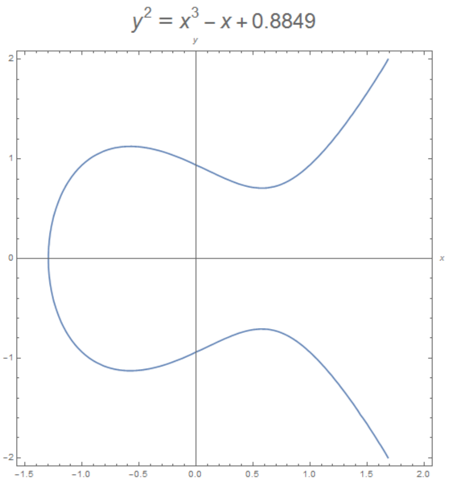
An elliptic curve E has a Weierstrass form has the following form:

$$E:\ y^2 = x^3 + ax + b$$

Where constants a and b fulfil the condition:

$$4a^3 + 27b^2 \neq 0$$

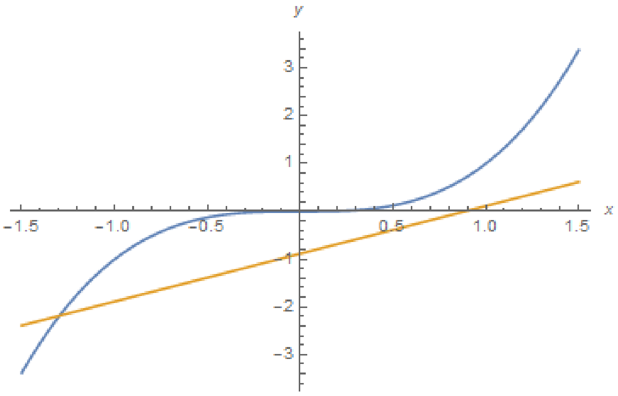
This condition is the cube polynomial non-zero discriminate condition, which guarantees three different roots, which can in general be complex numbers.
To illustrate why this condition is important, we will see a couple of these curves (Image 1)



Slika 1.1 elliptic curve 1. [Source: Author]

The properties can be seen if we include two functions, **f1(x) = x3** and **f2(x) = −ax − b**, where:

$$a = -1$$
$$b = \sqrt{-\frac{4a^2}{27}} + 0.5$$



Slika 1.2 Functions f1 and f2. [Source: Author]

We can see that the elliptic curve intersects the x-axis at the point where f1 and f2 intersects as well.

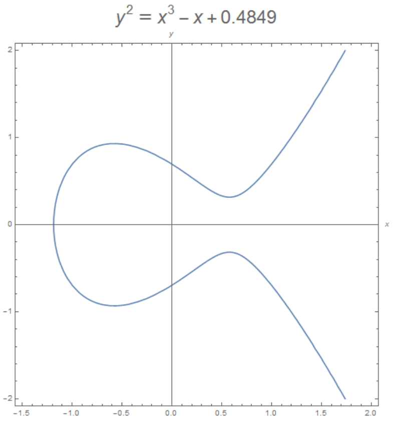
# WEIERSTRASS FORM OF ELLIPTIC CURVES: PARAMETERS

## *We can change parameters a and b to get different curves*

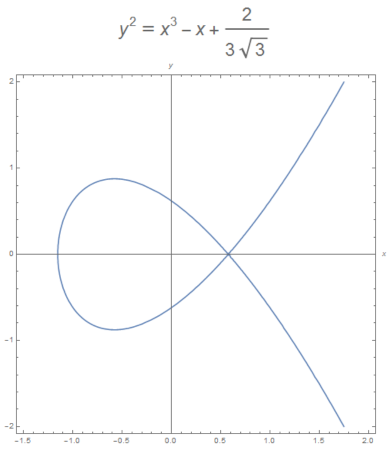
If we try again for a=-1 and different values of b, we can see how the curves change their overall shapes.

We will illustrate for b =0.1, 0, −0.1, and −0.5 respectively.

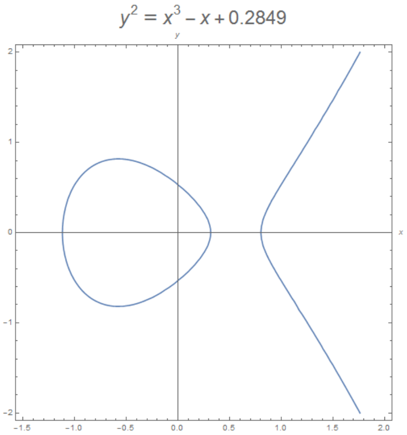
We can see that by changing the parameter b we can get (instead of one real and two complex roots) three real roots, which is important for point addition.



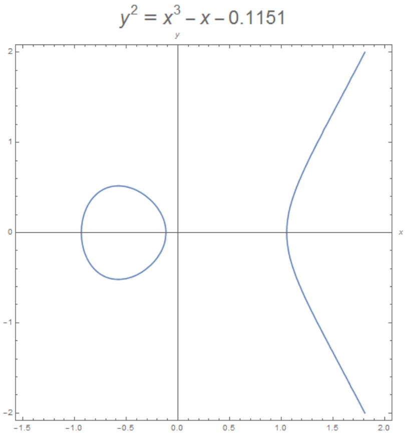
Slika 1.3 First change of elliptic curve. [Source: Author]



Slika 1.4 Second change of elliptic curve. [Source: Author]



Slika 1.5 third change of elliptic curve. [Source: Author]



Slika 1.6 Fourth change of elliptic curve. [Source: Author]

# Poglavlje 2

# Koblitz form of elliptic curves

## KOBLITZ FORM OF ELLIPTIC CURVES: INTRODUCTION

*Koblicova eliptička kriva je podtip tzv. generalizovane Vajerštrasove krive.*

The Weierstrass form of elliptic curves is symmetric. To include a non-zero discriminant, a different form is used.

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

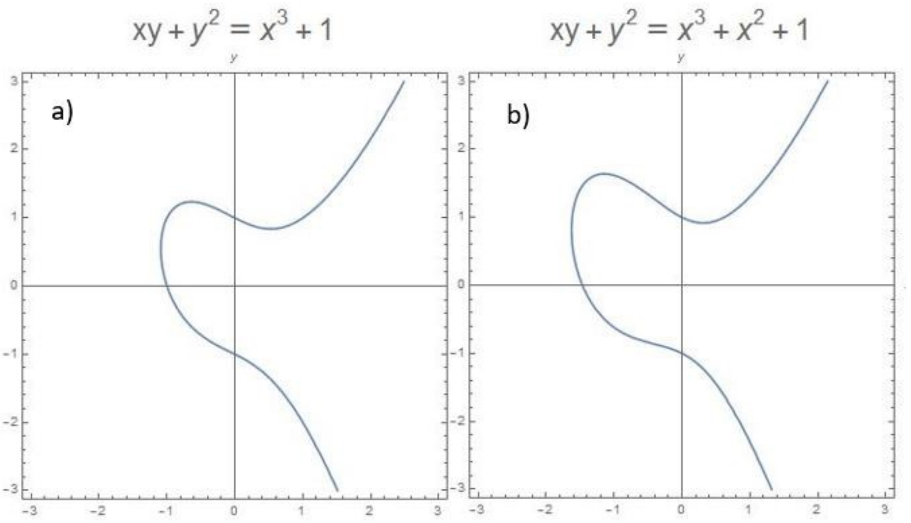
$$E_a : y^2 + xy = x^3 + ax + 1$$

Where a is within the set {0,1} and its discriminant is equal to 1.

The non-zero discriminant of the Weierstrass form of elliptic curve is imposed by the equation:

$$4a^3 + 27b^2 \neq 0$$

We can see in the Image that the symmetry is gone by adding a new parameter:



Slika 2.1 Koblitz elliptic curves with parameters a = 1 (left panel)) i and with a = 0 (right panel)). [Source: Autho]

# ELLIPTIC CURVES AND SO-CALLED BACKDOOR FUNCTIONS

## *Videos regarding elliptic curves and so-called backdoor functions*

Elliptic curves on Computerphile:

**Ova lekcija sadrži video materijal. Ukoliko želite da pogledate ovaj video morate da otvorite LAMS lekciju.**

Elliptic curves and backdoor functions on Computerphile:

**Ova lekcija sadrži video materijal. Ukoliko želite da pogledate ovaj video morate da otvorite LAMS lekciju.**

# Poglavlje 3

# The discrete logarithm problem

## THE DISCRETE LOGARITHM PROBLEM: INTRODUCTION

*It was the factorization of large numbers that served as the basis for the RSA construction of a practical and secure asymmetric (public key)cryptographic system.*

In cryptography, the main idea related to the construction of backdoor functions, is the existence of an mathematical operation or function in one direction is relatively easy to perform, while the inverse operation is extremely complicated. For instance, factoring large numbers is known to be a complex challenge, even with the help of the most powerful supercomputers.

It was the factorization of large numbers that served as the basis for the RSA construction of a practical and secure asymmetric (public key)cryptographic system.

However, no matter how demanding factorization is, it turns out that representing a point on an elliptic curve as a multiple sum (in the sense in which addition on an elliptic curve is already defined) of some other point on the same curve is significantly more complicated than factorization.

At this point, we will briefly go through the theory related to the discrete logarithm problem, and a more complete treatment of some of the terms that will appear here.

The discrete logarithm problem in its original form occurs through a one-step modular equation. Searching for exponents in the field of real numbers would be reduced to ordinary logarithms.

Searching for exponents in the field of real numbers would be reduced to ordinary logarithms.

As the finite fields represent a structure that is not continuous, finding the allowed values is made difficult by this restriction.

Elliptic curves will allow to completely revise the notion of gradation and reduce it to any binary operation that among the elements of a set will preserve its closure, the existence of unit and the existence of an inverse element for each element of that set.

Grading of an element will be reduced to element addition on elliptic curves, i.e. points on the curve, itself as many times as the value of the exponent would be in the standard case.

Due to the closure of the binary operation, the resulting element, i.e. the point, must still belong to the selected set. It will be shown that prime numbers play a crucial role in the construction of such a set.

The problem of determining the point whose multiplication yields another point is known as the discrete logarithm problem on an elliptic curve and is the basis of this type of cryptography.

# THE DISCRETE LOGARITHM PROBLEM: FORMAL DEFINITION

## *Point addition for ellipctic curves can be used for solving the discrete logarithm problem.*

Elliptic curves, which could represent an alternative to the discrete logarithm problem over some finite field, were introduced in the mid-80s in the works of Koblitz and Miller.

Although the authors laid the theoretical foundations, their focus was academic rather than practical and commercial application. By not patenting their discovery, the authors allowed other researchers in the field to freely use and improve the idea, as well as work on implementation.

That's why today there are a large number of solutions ranging from freely available codes to those that require a license. In the mid-1970s, Whitfield Diffie and Martin Hellman introduced the concept of a public key into cryptography using a discrete logarithm over a finite field of order **p**, where **p** is a prime number.

As will be shown, to understand the theory based on any finite field of the order of the prime number **p**, it is sufficient to analyze the field **Zp**, which is made up of representatives of the equivalence classes of those numbers which, when divided by the prime number **p**, give the remainders

**0, 1, 2, 3, ..., p-1.**

Within such a set there can be subsets, each of which represents an entity that satisfies the condition of closure and the existence of an inverse element for each element, while the unit element would be the only element common to all subsets.

Among the representatives of the classes within each of these subsets, one representative can be chosen one by one, from which all the other representatives that make up the sub-field will be generated by the successive application of the binary operation. We call such an element a primitive root.

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 DISCRETE LOGARITHM PROBLEM AND ELLIPTIC CURVES

## *Raising to some degree an element of a finite field implies that a binary operation on the elements is defined*

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).

Raising to some degree an element of a finite field implies that a binary operation on the elements is defined. We can call it multiplication, addition, it is up to us

The generality of defining an arbitrary binary operation allows the binary operation to be applied c times over an element of the field **a**.

In this sense, the rule of addition, as it will be introduced for elliptic curves, will be able to be applied in the case of discrete logarithms as well.

# VIDEO: THE DISCRETE LOGARITHM PROBLEM

## *A video explanation on the discrete logarithm problem*

**Ova lekcija sadrži video materijal. Ukoliko želite da pogledate ovaj video morate da otvorite LAMS lekciju.**

# Poglavlje 4

# Applying ECC in PKI

## POINT ADDITION: INTRODUCTION

*The binary operation of "addition" should provide closure, a unit element and an inverse element for each element of the group.*

When the condition $\mathbf{4a^3+27b^2 /= 0}$ ensures that the elliptic curve of the Weierstrass type at each point has a well-defined first derivative, we can proceed to the construction of the addition operation.

To define a group property, it is necessary to introduce a binary "addition" operation that will provide closure, a unit element, and an inverse element for each element of the group.

The elements of the group will be points on the curve. Each point is given with two coordinates, so for two different points on the curve it is pointless to introduce an order relation, as it is done in the case when the object is assigned exactly one numerical value.

This means that the concept of addition must be understood conditionally, not as an operation that can raise or lower the value of a given integer, but simply as an operation of fusion of two objects, which results in a third object that, like the previous two from which it was created, must belong to some finite set, i.e. group.

This implies that in a given set there must be an object whose fusion with any object of that group, as a result, leaves that object unchanged.

In this way, we require the existence of a unit element. Likewise, for each object of the group there must be its inverse, in the sense that the fusion of any object and its inverse element gives exactly the unit element.

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.

- A semigroup with a unit element in which every element is invertible is called a group. A group is said to be commutative or Abelian if the binary operation of the group is commutative.

# POINT ADDITION: DEFINITION

## *A polynomial in the field of complex numbers can certainly be factored.*

The group and the binary operation defined in the group will be of utmost importance here. It is important to understand that the group is closed to the binary operation, that there is a unit element and that for each element x there is an inverse y such that the binary operation of those two elements gives a unit element.

*How can group structure be introduced for points on an elliptic curve?*

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:

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

A cubic equation is obtained, for which we know for sure that its two zeros are xi and xj . Let's call the third zero xk. A polynomial in the field of complex numbers can certainly be factored.

# POINT ADDITION: THE POINT BELONGS ON THE ELLIPTIC CURVE

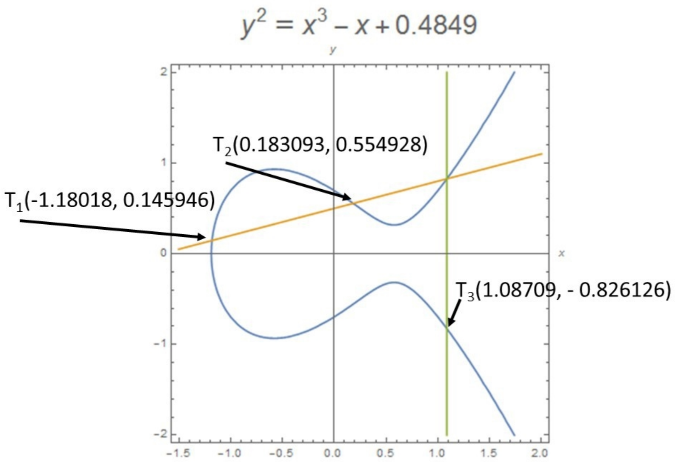
## *The new point belongs on the elliptic curve*

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.

Precisely, the point Tk, as an element of the group, is taken to be obtained as a result of the binary operation Ti B Tj =Tk



Slika 4.1 Point addition. [Source: Author]

# THE COMMUTATIVITY PROPERTY OF BINARY ADDITION OPERATIONS

*The whole concept of key exchange rests on the idea that two parties will eventually arrive at a shared secret key.*

The commutativity property of the binary operation, as introduced, is very important. The whole concept of key exchange rests on the idea that two parties will eventually arrive at a shared secret key.

Here, however, the order in which one side and the other apply their operations was not questioned, simply because binary operations, such as multiplication or addition of numbers over an arbitrary field, are always commutative.

However, commutativity is not guaranteed over every structure and depends on how the binary operation is defined. For example, vector multiplication of two vectors is not a commutative operation, nor is matrix multiplication of two matrices. Since the goal of improving the Diffie-Hellman solution involved the introduction of a new structure, namely the group property of pairs of points on elliptic curves, the whole concept depended on whether the operations could remain invariant on reordering.

Only in that case would the original solution be successfully copied to a more complex structure. Although a binary operation such as that introduced over the points of elliptic curves can be made to be rather random and ad hoc performed, it was actually designed to preserve commutativity, thus opening the door to the application of elliptic curves in cryptography.

In number theory, the concept of a subgroup can be introduced, which is an algebraic structure that is a subset of the elements of the group, and which possesses all the properties of the group (unit element, inverse element, closure).

**Ova lekcija sadrži video materijal. Ukoliko želite da pogledate ovaj video morate da otvorite LAMS lekciju.**

# VIDEO: ELLIPTIC CURVE POINT ADDITION

*A video on elliptic curve point addition.*

**Ova lekcija sadrži video materijal. Ukoliko želite da pogledate ovaj video morate da otvorite LAMS lekciju.**

# Poglavlje 5

# Diffie-Helman algorithm with elliptic curves

## ELLIPTIC CURVE POINT INTEGER PRODUCT

*The discrete logarithm introduces scaling.*

RSA is one of the most well known asymmetric cryptography algorithms, where the public key is taken as a huge number that is the product of two large prime numbers.

As already mentioned, the discrete logarithm introduces scaling, which in the language of an arbitrary field can be understood as a successively applied binary operation on some element. Addition defined on elliptic curves is one such type of operation, possessing all the necessary properties already discussed.

Therefore, the problem of discrete logarithm on an elliptic curve with coordinates in a finite field can be formulated, which requires that for two given points Ti = (xi, yi) and Tj = (xj , yj ) a natural number **n** is found, such that:

$$nT_i = T_j$$

Let's firstly define an elliptic curve over a finite field: Earlier we discussed Weierstrass curves, singular points, and the reasons why binary fields are not a good choice for Weierstrass curves.

The analysis that introduced the concept of a finite field, as well as modular arithmetic, will now help us to see better some of the conclusions that were presented earlier.

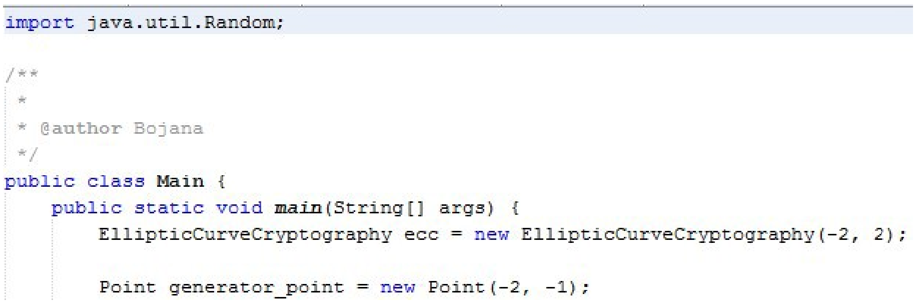
## EXAMPLE: DIFFIE-HELLMAN KEY EXCHANGE IN JAVA: PUBLIC POINT

*In the language of an arbitrary field can be understood as a successively applied binary operation on some element.*

**Example: Application of the Diffie-Hellman key exchange algorithm to elliptic curve cryptography in the Java programming language**

The Diffie Hellman key exchange algorithm based on elliptic curves is implemented.

First, a public generic point was created, the so-called base point in the main method. It is public.

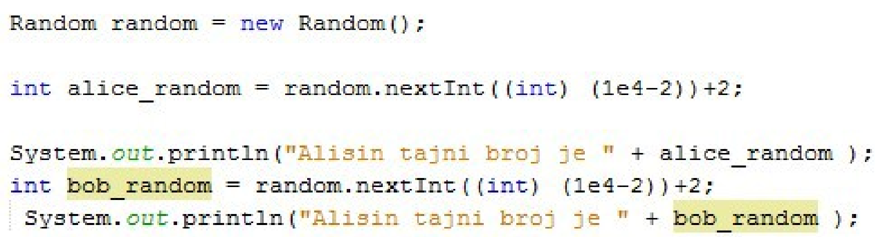


```
import java.util.Random;

/**
 *
 * @author Bojana
 */
public class Main {
    public static void main(String[] args) {
        EllipticCurveCryptography ecc = new EllipticCurveCryptography(-2, 2);

        Point generator_point = new Point(-2, -1);
```

Slika 5.1 Generating the public point. [Source: Author]

In addition to the base point, the equation of the curve and the public keys of Alice and Bob, denoted by ka and kb, are also public.

Random numbers are generated for both Alice and Bob that will take the value from 0 to 10000.



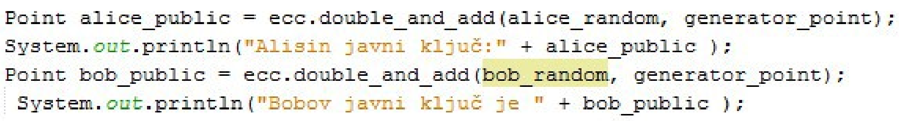
```
Random random = new Random();

int alice_random = random.nextInt((int) (1e4-2))+2;

System.out.println("Alisin tajni broj je " + alice_random );
int bob_random = random.nextInt((int) (1e4-2))+2;
 System.out.println("Alisin tajni broj je " + bob_random );
```

Slika 5.2 Random number generation. [Source: Author]

# EXAMPLE: DIFFIE-HELLMAN KEY EXCHANGE IN JAVA: ALGORITHM

*Alice's shared key is calculated again using the duplicate and add method, and takes as arguments Bob's public key, which will be multiplied by the generated point.*

The duplicate and add method was used for the public key. The input arguments of that method are the random numbers of Alice and Bob that will be multiplied by the generated point:

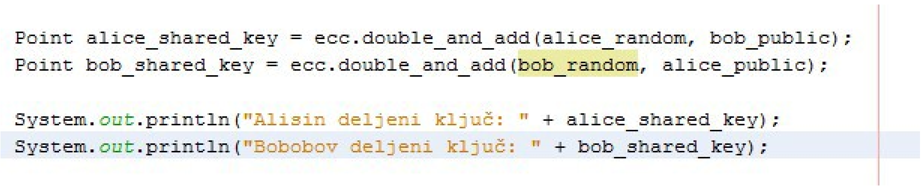


```
Point alice_public = ecc.double_and_add(alice_random, generator_point);
System.out.println("Alisin javni ključ:" + alice_public );
Point bob_public = ecc.double_and_add(bob_random, generator_point);
 System.out.println("Bobov javni ključ je " + bob_public );
```

Slika 5.3 Random numbers multiplid with the public point. [Source: Author]

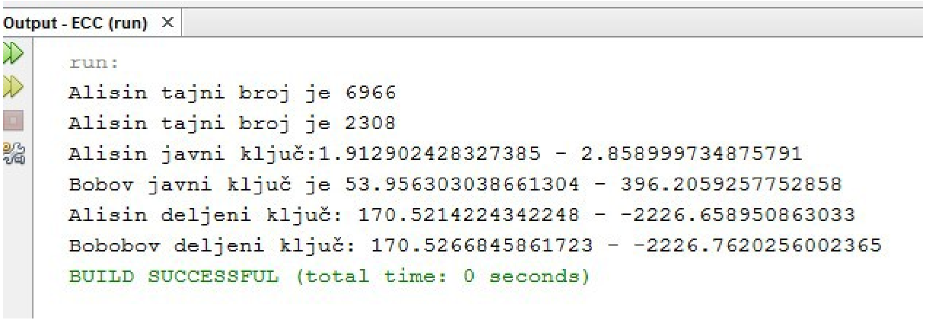
A shared key is then generated. Alice's shared key is calculated again using the duplicate and add method, and takes as arguments Bob's public key, which will be multiplied by the generated point.

The same logic is applied to the implementation of Bob's public shared key. These keys are the same in value.



```
Point alice_shared_key = ecc.double_and_add(alice_random, bob_public);
Point bob_shared_key = ecc.double_and_add(bob_random, alice_public);

System.out.println("Alisin deljeni ključ: " + alice_shared_key);
System.out.println("Bobobov deljeni ključ: " + bob_shared_key);
```

Slika 5.4 Shared key generation. [Source: Author]



```
run:
Alisin tajni broj je 6966
Alisin tajni broj je 2308
Alisin javni ključ:1.912902428327385 - 2.858999734875791
Bobov javni ključ je 53.956303038661304 - 396.2059257752858
Alisin deljeni ključ: 170.5214224342248 - -2226.658950863033
Bobobov deljeni ključ: 170.5266845861723 - -2226.7620256002365
BUILD SUCCESSFUL (total time: 0 seconds)
```

Slika 5.5 Results. [Source: Author]

# VIDEO: DIFFIE-HELLMAN WITH ELLIPTIC CURVES

*A video on the Diffie-Hellman algorithm with Elliptic curves.*

**Ova lekcija sadrži video materijal. Ukoliko želite da pogledate ovaj video morate da otvorite LAMS lekciju.**

# Poglavlje 6

# Practice work

## EXERCISE #1

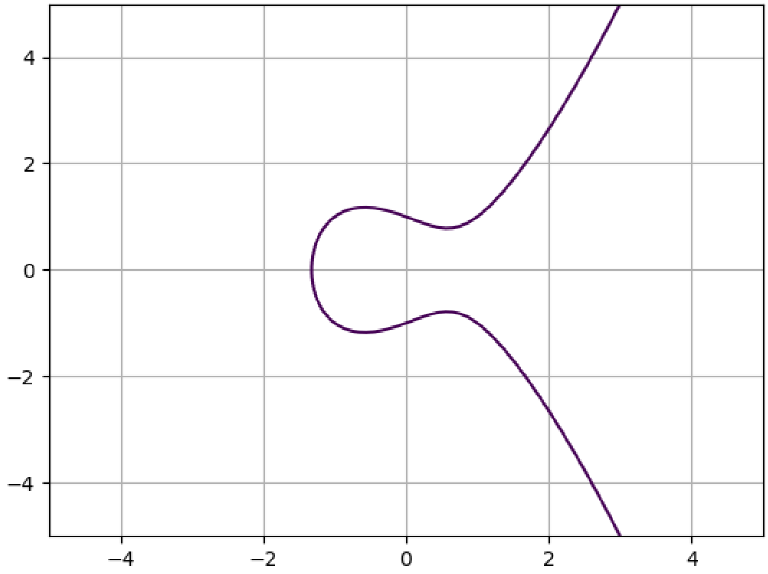
*Exercise #1 takes up to 30 minutes to complete*

**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()
```



Slika 6.1 Results. [Source: Author]

# EXERCISE #2: INTRODUCTION

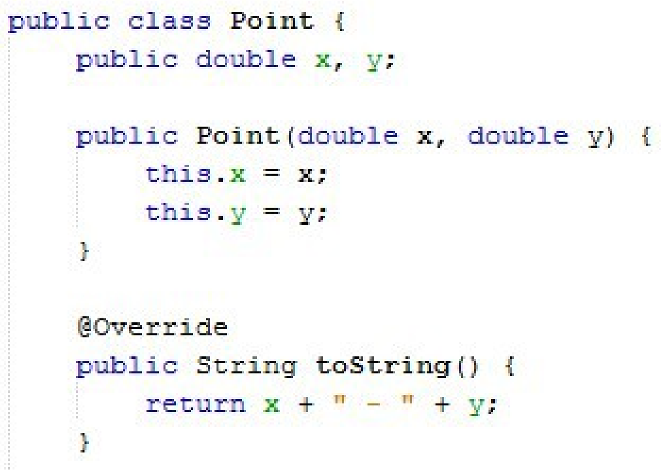
*Exercise #1 takes up to 60 minutes to complete*

**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.



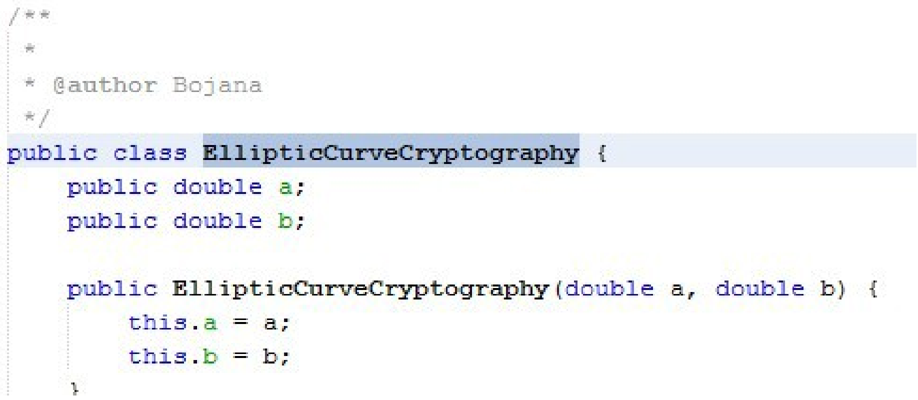
```
public class Point {
    public double x, y;

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

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

Slika 6.2 Coordinates. [Source: Author]

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**.



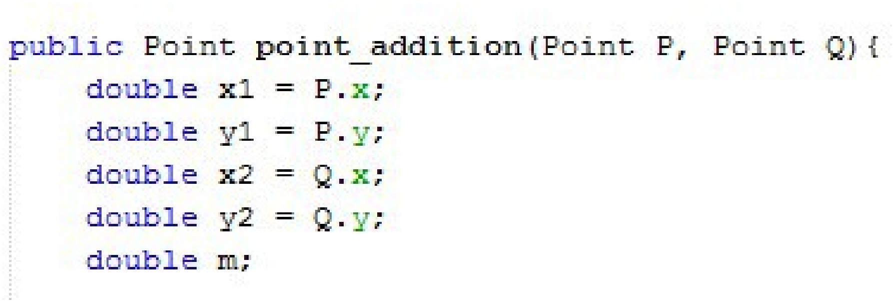
```
/**
 *
 * @author Bojana
 */
public class EllipticCurveCryptography {
    public double a;
    public double b;

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

Slika 6.3 Bitcoin curve parameters. [Source: Author]

# EXERCISE #2 - POINT ADDITION

## *We need a point addition function*

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:



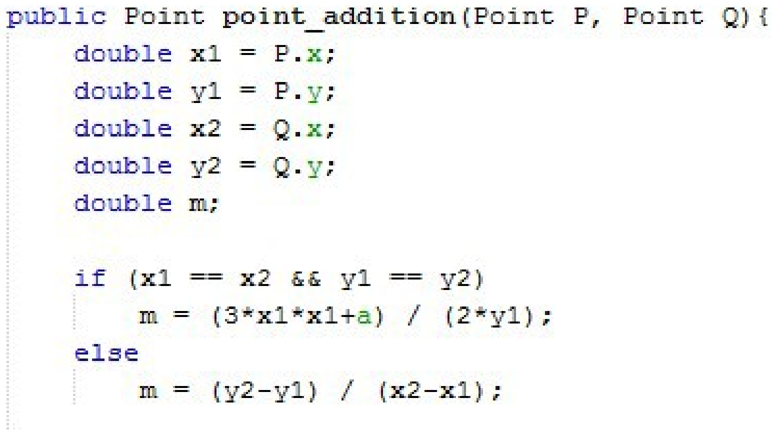
```
public Point point_addition(Point P, Point Q){
    double x1 = P.x;
    double y1 = P.y;
    double x2 = Q.x;
    double y2 = Q.y;
    double m;
```

Slika 6.4 Point addition function [Source: Author]

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);
```

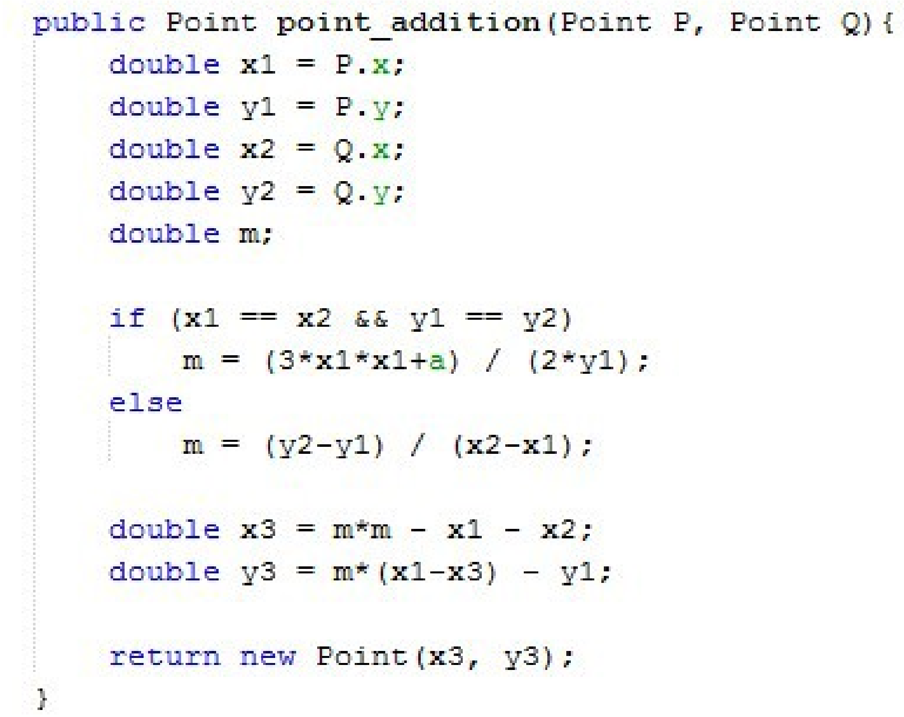
Slika 6.5 Cuefficient equation. [Source: Author]

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.

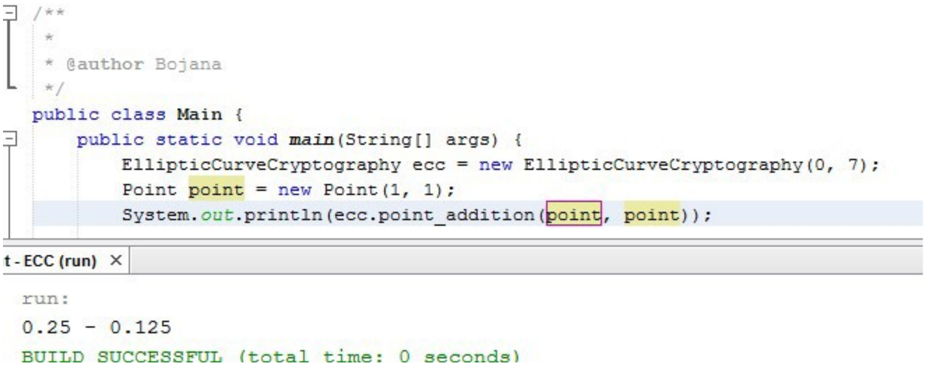
# EXERCISE #2 - RESULTS

## *The final results of exercise #2 are shown*



```
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);
}
```

Slika 6.6 Point coordinate computation. [Source: Author]



```
/**
 *
 * @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));
```

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

Slika 6.7 Printing the results. [Source: Author]

# Poglavlje 7

# Homework

## ECC HOMEWORK

*Homework can be completed in about two hours.*

**Homework**

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

# Poglavlje 8

# Zaključak

## CONCLUSION

### *Elliptic curve cryptography was discussed in this lesson.*

Elliptic curve cryptography was discussed in this lesson.

First, students were introduced to the mathematical concept of elliptic curves and two types of these curves - Weierstrass and Koblitz curves.

Next, the students were re-introduced to the discrete logarithm problem in cryptography, looking at how elliptic curves can be used to solve this problem.

The application of elliptic curves in asymmetric cryptography algorithms is presented, along with the function of binary addition of points on the curve.

The last learning object is an implementation of the Diffie-Hellman algorithm on an elliptic curve, and a comparison with an equivalent algorithm using modular arithmetic of large prime numbers.

## REFERENCES

### *References for the lesson*

**References:**

- R. Gupta, Hands-On Cybersecurity with Blockchain: Implement DDoS protection, PKI-based identity, 2FA, and DNS security using Blockchain, Packt Publishing, 2018.
- S. Shetty, C. Kamhoua, L. Njilla (Editors), Blockchain for Distributed Systems Security, Wiley, 2019.
- I. Bashir, Mastering Blockchain: Distributed ledger technology, decentralization, and smart contracts explained, 2nd edition, Packt Publishing, 2018.

**Papers:**

Satoshi Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System", [Online], Available: https://bitcoin.org/bitcoin.pdf

**Web locations:**

Blockchain: A Beginner's Guide. https://s3.eu-west-2.amazonaws.com/blockchainhub.media/Blockchain+Technology+Handbook.pdf