

**UNIwersYTET WARMIŃSKO-MAZURSKI**  
**W OLSZTYNIE**  
**WYDZIAŁ MATEMATYKI I INFORMATYKI**

**Kierunek: Informatyka**  
**Projektowanie systemów informatycznych i sieci**  
**komputerowych**

**Karol Budzyński**

**Zastosowanie krzywych eliptycznych w**  
**kryptografii**

Praca magisterska wykonana  
w Katedrze Elektrotechniki,  
Energetyki, Elektroniki i Automatyki  
pod kierunkiem  
*Dr inż. Zenona Syroka*

Olsztyn 2022

**UNIVERSITY OF WARMIA AND MAZURY  
IN OLSZTYN  
FACULTY OF MATHEMATICS AND COMPUTER  
SCIENCE**

**Field of study: Computer Science**

**Designing information systems and computer networks**

**Karol Budzyński**

**Application of elliptic curves in cryptography**

Master's Thesis written in  
Department of Electrotechnics, Power  
Engineering, Electronics and  
Automation  
under supervision of  
*Dr inż. Zenon Syroka*

Olsztyn 2022



# SPIS TREŚCI

<b>Streszczenie .....</b>	<b>6</b>
<b>Streszczenie w języku angielskim .....</b>	<b>7</b>
<b>1. Wstęp .....</b>	<b>6</b>
1.1. Cel pracy .....	6
1.2. Treść pracy .....	6
<b>2. Krzywe eliptyczne .....</b>	<b>10</b>
2.1. Krzyw eliptyczne nad ciałem rzeczywistym $\mathbb{R}$ .....	10
2.2. Prawo grupowe dla krzywych eliptycznych .....	11
2.2.1. Dodawanie geometryczne i algebraiczne .....	12
2.2.2 Mnożenie przez skalar .....	14
2.3. Krzywe eliptyczne nad ciałem skończonym $\mathbb{F}_p$ .....	15
2.3.1 Dodawanie geometryczne i algebraiczne .....	17
2.3.2 Mnożenie przez skalar i cykliczne podgrupy .....	18
<b>3. Kryptografia oparta na krzywych eliptycznych .....</b>	<b>23</b>
3.1. Algorytm ElGamala z implementacją krzywych eliptycznych .....	23
3.2. Algorytm Diffiego-Hellmana w przestrzeni krzywych eliptycznych ECDH .....	25
3.3. Algorytm podpisu cyfrowego z wykorzystaniem krzywych eliptycznych ECDSA .....	26
3.4 Kryptografia krzywych eliptycznych a RSA .....	29
<b>4. Część programistyczna .....</b>	<b>31</b>
4.1. Technologie wykorzystane przy tworzeniu aplikacji .....	31
4.2. Funkcjonalność aplikacji .....	31
<b>5. Podsumowanie i wnioski .....</b>	<b>46</b>
<b>Literatura .....</b>	<b>47</b>
<b>Spis rysunków .....</b>	<b>48</b>

# CONTENTS

<b>Summary .....</b>	<b>6</b>
<b>Summary in English .....</b>	<b>7</b>
<b>1. Introduction .....</b>	<b>6</b>
1.1. Objective of the work .....	6
1.2. The content of the work .....	6
<b>2. Elliptic curves .....</b>	<b>10</b>
2.1. Elliptic curves over the real body $\mathbb{R}$ .....	10
2.2. Group law for elliptic curves .....	11
2.2.1. Geometric and algebraic addition .....	12
2.2.2 Scalar multiplication .....	14
2.3. Elliptic curves over the finite field $\mathbb{F}_p$ .....	15
2.3.1 Geometric and algebraic addition.....	17
2.3.2 Scalar multiplication and cyclic subgroups .....	18
<b>3. Cryptography based on elliptic curves .....</b>	<b>23</b>
3.1. ElGamal algorithm with implementation od elliptic curves .....	23
3.2. Diffie-Hellman algorithm in space of elliptic curves ECDH .....	25
3.3. Digital signature algorithm with the use of elliptic curves ECDSA .....	26
3.4 Elliptic curves cryptography and RSA comparison .....	29
<b>4. Programming part .....</b>	<b>31</b>
4.1. Technologies used to create the application .....	31
4.2. Functionality of the application .....	31
<b>5. Summary and conclusion .....</b>	<b>46</b>
<b>Literature .....</b>	<b>47</b>
<b>List of drawings .....</b>	<b>48</b>

## **Streszczenie**

W niniejszej pracy zaprezentowane zostaną sposoby wykorzystania krzywych eliptycznych w kryptografii. Począwszy od przedstawienia, czym są krzywe eliptyczne, jakie są rodzaje krzywych eliptycznych, poprzez działania matematyczne, jakie mogą być na nich przeprowadzane. W dalszej kolejności praca zajmuje się zaprezentowaniem możliwości wykorzystania krzywych eliptycznych w różnych algorytmach szyfrowania, takich jak ElGamal czy Diffie-Hellmann. Na koniec zaprezentowany zostaje program komputerowy mojego autorstwa, który pozwala w praktyce sprawdzić wiedzę przedstawioną w niniejszej pracy.

## Summary

In this paper, methods of using elliptic curves in cryptography will be presented. Starting from the presentation of what are elliptic curves, what are the types of elliptic curves, through the mathematical operations that can be carried out on them. Next, the work presents the possibility of using elliptic curves in various encryption algorithms, such as ElGamal or Diffie-Hellmann. At the end, a computer program of my authorship is presented, which allows to test the knowledge presented in this paper in practice.

# 1. Wstęp

W dzisiejszym świecie bezpieczeństwo danych jest jedną z najistotniejszych kwestii w informatyce. Aby zapewnić bezpieczeństwo w sieci oraz w przepływie informacji, stosuje się coraz to silniejsze metody kryptograficzne. Wraz z rozwojem technologii i coraz szybszym zwiększeniem mocy obliczeniowych komputerów dotychczasowe metody zabezpieczeń zostają coraz szybciej łamane. Aby temu zapobiec wprowadzono kryptografię opartą o krzywe eliptyczne (ECC, ang. Elliptic Curve Cryprography). Dziś kryptosystemy krzywych eliptycznych możemy znaleźć w takich rozwiązaniach jak TLS (ang. Transport Layer Security), PGP (ang. Preety Good Privacy) i SSH (ang. Secure Shell), czyli technologiach, na których opiera się współczesny świat web i IT. Kryptografia oparta o krzywe eliptyczne wykorzystywana jest także w cyfrowej monecie Bitcoin ora innych kryptowalutach.

Zanim kryptografia oparta o krzywe eliptyczne stała się popularna, większość algorytmów klucza publicznego oparte były na RSA (ang. Rivest-Shamir-Adleman), DSA (ang. Digital Signature Algorithm), DH (ang. Diffie-Hellman), alternatywnych kryptosystemach opartych na arytmetyce modularnej. Te kryptosystemy nadal są bardzo ważne i często są używane razem z kryptografią krzywych eliptycznych. Metodologia stojąca za RSA jest powszechnie znana i rozumiana, lecz podstawy kryptografii opartej o krzywe eliptyczne wciąż są dla większości tajemnicą.

## 1.2 Cel pracy

W swojej pracy zamierzam przedstawić krzywe eliptyczne oraz matematyczne działania z nimi związane. Moim celem jest zapewnienie prostego przedstawienia, czym jest kryptografia krzywych eliptycznych i dlaczego jest uznawana za bezpieczną. Przedstawione zostanie wykorzystanie krzywych eliptycznych w kryptografii na podstawie algorytmu ElGamal, oraz dwa algorytmy ECC, czyli ECDH (ang. Elliptic Curve Diffie-Hellman) i ECDSA (ang. Elliptic Curve Digital Signature Algorithm).

## 1.3 Treść pracy

Praca podzielona jest na pięć rozdziałów. Rozdział drugi zawiera teoretyczne przedstawienie krzywych eliptycznych i działań matematycznych, jakie można na nich wykonywać. Zostały przedstawione krzyw eliptyczne na ciałem rzeczywistym  $R$  i nad ciałem skończonym  $F(p)$ . W rozdziale trzecim przedstawiono zastosowanie



krzywych eliptycznych w kryptografii. Opisany w nim został protokół wymiany kluczy Diffiego-Hellmana, algorytm szyfrowania ElGamal z wykorzystaniem krzywych eliptycznych oraz algorytmy oparte o krzywe eliptyczne ECDH i ECDSA. Rozdział czwarty zawiera prezentację aplikacji komputerowej mojego autorstwa, której zadaniem jest zaprezentowanie użytkownikowi krzywych eliptycznych nad ciałem rzeczywistym  $\mathbb{R}$ , nad ciałem skończonym  $\mathbb{F}(p)$  oraz działań matematycznych na tych krzywych (dodawanie punktów, mnożenie punktów). Dodatkowo w programie zaimplementowano algorytm szyfrowania ElGamal z wykorzystaniem krzywych eliptycznych do szyfrowania punktów oraz cyfrowy podpis ECDSA, dzięki któremu użytkownik może podpisać dowolną wiadomość oraz zweryfikować poprawność otrzymanego podpisu.

## 2. Krzywe eliptyczne

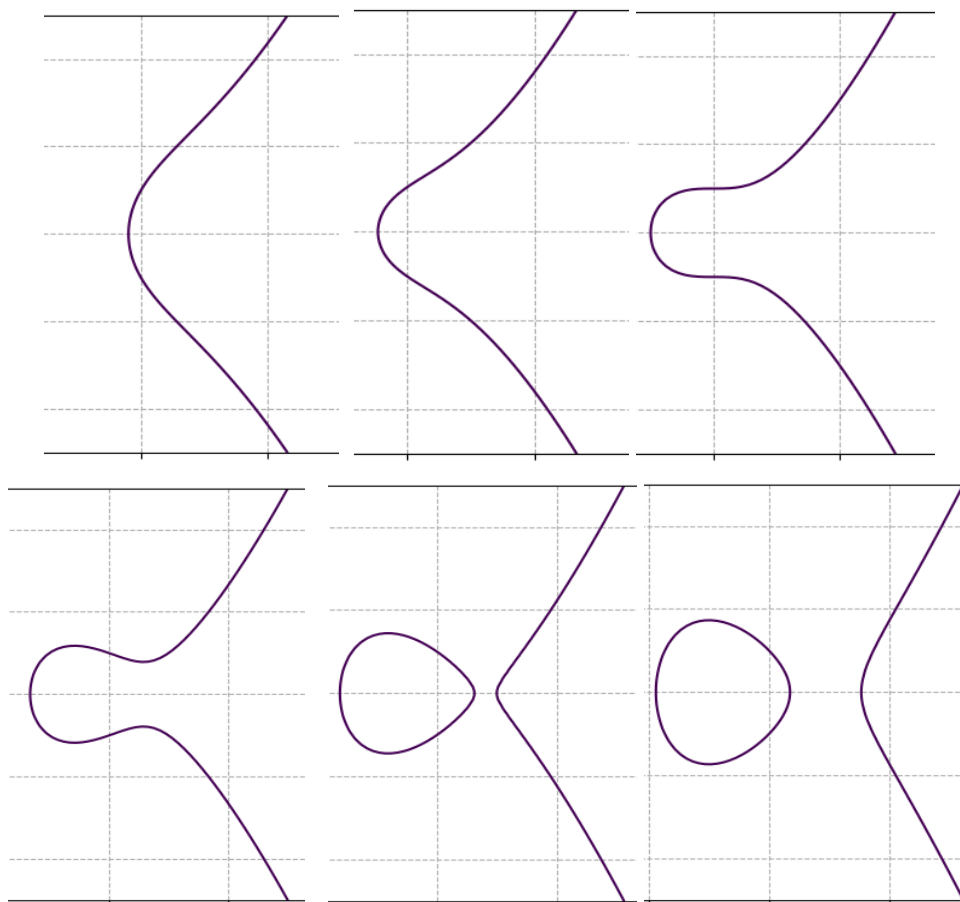
### 2.1 Krzywe eliptyczne nad ciałem rzeczywistym $\mathbb{R}$

Musimy zacząć od tego, czym jest krzywa eliptyczna. Nieformalnie krzywa eliptyczna jest rodzajem krzywej sześcienniej, której rozwiązania są ograniczone do obszaru przestrzeni, który jest topologicznie równoważny torusowi. Funkcja eliptyczna Weierstrassa opisuje, jak przejść z niego do postaci algebraicznej krzywej eliptycznej. Dla celów tej pracy krzywa eliptyczna będzie po prostu zbiorem punktów opisanych równaniem:

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

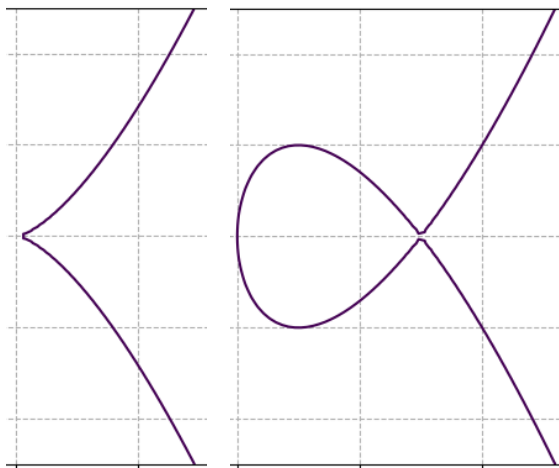
gdzie:  $4a^3 + 27b^2 \neq 0$  (warunek konieczny aby wykluczyć krzywe osobliwe)

Powyższe równanie nazywane jest normalną postacią Weierstrassa dla krzywych eliptycznych. Na rysunku 2.1.1 przedstawiono różne kształty prawidłowo wyznaczonych krzywych eliptycznych. W tych przykładach przyjęto parametr  $b = 1$ , natomiast parametr  $a$  waha się od 5 do -3.



Rys. 2.1.1. Różne kształty prawidłowych krzywych eliptycznych nad ciałem rzeczywistym  $\mathbb{R}$

Natomiast na rysunku 2.1.2 przedstawione zostały nieprawidłowe krzywe eliptyczne. Są to tak zwane osobliwości. Po lewej ukazany jest łuk z wierzchołkiem ( $y^2 = x^3$ ), obok znajduje się łuk z samoprzecięciem ( $y^2 = x^3 - 3x + 2$ ).



Rys. 2.1.2. Nieprawidłowe krzywe eliptyczne nad ciałem rzeczywistym  $\mathbb{R}$

W zależności od wartości parametrów  $a$  i  $b$ , krzywe eliptyczne przyjmują różne kształty na płaszczyźnie. Można zaobserwować, iż są one symetryczne względem osi  $x$ . Do obliczeń potrzebujemy także punktu w nieskończoności, który nazywany jest także punktem idealnym i jest częścią krzywej eliptycznej (będzie oznaczany jako  $0$ ).

Uwzględniając punkt w nieskończoności, możemy doprecyzować definicję krzywej eliptycznej w taki oto sposób:

$$\{(x, y) \in \mathbb{R}^2 \mid y^2 = x^3 + ax + b, 4a^3 + 27b \neq 0\} \cup \{0\}$$

## 2.2 Prawo grupowe dla krzywych eliptycznych

Zacznę od wyjaśnienia pojęcia grupy. W matematyce jest to zbiór, ze zdefiniowaną operacją binarną, którą nazywamy dodawaniem i oznaczmy symbolem  $+$ . Aby zbiór  $G$  mógł być nazywany grupą, dodawanie musi być tak zdefiniowane, aby spełniało te cztery właściwości:

- zamknięcie: jeżeli  $a$  i  $b$  należą do zbioru  $G$ , to  $a + b$  też należy do zbioru  $G$ ;
- asocjacja:  $a + (b + c) = (a + b) + c$ ;
- istnieje element zbioru  $0$  taki, że  $0 + a = a + 0 = a$ ;
- wszystkie elementy zbioru mają odwrotność, czyli dla każdego elementu  $a$  istnieje taki element  $b$ , że spełniony jest warunek  $a + b = 0$ .

Jeżeli spełniony zostanie jeszcze jeden warunek (przemienność:  $b + a = a + b$ ), to grupa taka jest grupą abelową (grupą przemenną). Zbiór liczb całkowitych to grupa (a nawet grupa abelowa). Natomiast zbiór liczb naturalnych nie jest grupą, ponieważ nie zostaje spełniona czwarta właściwość grupy (brak odwrotności).

Definicja grupy nad krzywą eliptyczną jest następująca:

- elementy grupy są punktami krzywej eliptycznej;
- punkt w nieskończoności  $0$ , jest elementem tożsamości;
- odwrotność punktu  $P$ , jest punktem symetrycznym względem osi  $x$ ;
- dodawanie jest zdefiniowane przez następującą regułę: trzy zbieżne, niezerowe punkty  $P, Q, R$  spełniają warunek  $P + Q + R = 0$ .

Na tej podstawie możemy wywnioskować, że jeżeli  $P, Q$  i  $r$  są wyrównane to spełniony zostaje warunek:

$$P + (Q + R) = Q + (P + R) = R + (P + Q) = 0$$

Operator dodawania jest asocjacyjny i przemienny, a więc mamy do czynienia z grupą ablową.

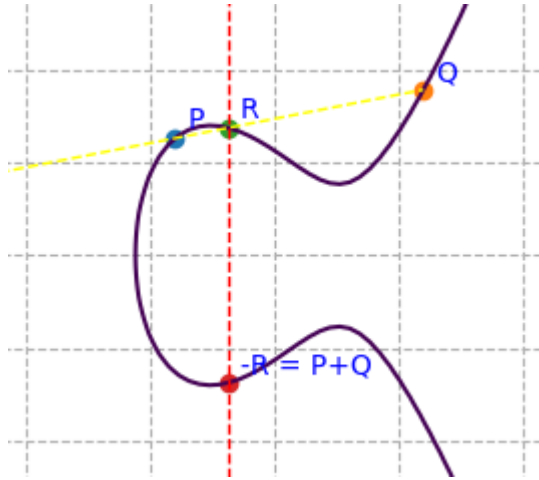
## 2.2.1 Dodawanie geometryczne i algebraiczne

### Dodawanie geometryczne

Dzięki temu, że grupa krzywej eliptycznej, jest grupą ablową, możemy zapisać:

$$P + Q + R = 0, \text{ jako } P + Q = -R$$

Równanie w tej postaci, umożliwia wyprowadzenie metody geometrycznej na obliczenie sumy między dwoma punktami  $P$  i  $Q$ . Jeżeli wyznaczymy linię przechodzącą przez te punkty, linia ta przetnie kolejny punkt na krzywej  $R$ . Następnie, biorąc odwrotność tego punktu,  $-R$  otrzymujemy wynik  $P + Q$ . Przedstawione to zostało na Rys. 2.2.1.1.



Rys. 2.2.1.1. Dodawanie geometryczne punktów na krzywej eliptycznej nad ciałem rzeczywistym  $\mathbb{R}$

Dodatkowo musimy rozważyć parę przypadków szczególnych:

- $P = 0$  lub  $Q = 0$ : ponieważ zdefiniowaliśmy 0 jako punkt w nieskończoności, więc  $P + 0 = P$ , oraz  $Q + 0 = Q$ , dla każdego  $P$  i  $Q$
- $P = -Q$ : punkt  $P$  jest odwrotnością punktu  $Q$ , a więc  $P + Q = P + (-P) = 0$ ;

### Dodawanie algebraiczne

Aby komputer mógł przeprowadzić dodawanie punktów na krzywej eliptycznej, należy zamienić metodę geometryczną na algebraiczną. Poniżej przedstawię wyniki tego przekształcenia oraz przykłady potwierdzające ich prawidłowość.

Najpierw należy odrzucić skrajne przypadki. Wiemy, iż  $P + (-P) = 0$  oraz  $P + 0 = 0 + P = 0$ . A więc w równaniach pominę te dwa przypadki i zajmę się tylko dwoma niezerowymi i niesymetrycznymi punktami  $P = (px, py)$ ,  $Q = (qx, qy)$ .

Jeżeli  $P$  i  $Q$  są różne ( $px \neq qx$ ), linia przechodząca przez te punkty ma nachylenie:

$$m = \frac{py - qy}{px - qx}$$

Przecięcie tej linii z krzywą eliptyczną wyznacza trzeci punkt  $R = (rx, ry)$ :

$$\begin{aligned} rx &= m^2 - px - qx \\ ry &= py + m(rx - px) \end{aligned}$$

lub równoważnie:

$$ry = qy + m(rx - qx)$$

stąd wynika:

$$(px, py) + (qx, qy) = (rx, -ry)$$

Nieco inaczej to prezentuje się w przypadku, gdy  $P = Q$ . Równania dla  $rx$  i  $ry$  są takie same, ale zważywszy na to, iż  $px = qx$ , musi zostać użyte inne równanie dla nachylenia:

$$m = \frac{3px^2 + a}{2py}$$

Dzięki postaci normalnej Weierstrassera, wyprowadzone równania mają kompaktową formę, bez niej te równania były by naprawdę długie i skomplikowane.

### 2.2.2 Mnożenie przez skalar

Poza dodawaniem punktów na krzywej eliptycznej, można także zdefiniować operację mnożenia przez skalar:

$$nP = \underbrace{P + P + \dots + P}_{n \text{ razy}}$$

gdzie:  $n$  jest liczbą naturalną.

Jak widać tak naprawdę mnożenie przez skalar jest sprowadzone do wykonania operacji dodawania  $n$  razy. Jeżeli  $n$  posiada  $k$  cyfr binarnych, wtedy podany algorytm miałby złożoność  $O(2^k)$ , co powodowałoby, iż był by on czasochłonny. Lecz istnieją szybsze algorytmy. Jednym z nich jest metoda podwój i dodaj. Najprościej będzie go wyjaśnić na przykładzie. Niech  $n = 151$ . Wartość w systemie binarnym to  $10010111_2$ . Ta wartość może zostać zamieniona na sumę potęg liczby 2:

$$151 = 1 \cdot 2^7 + 0 \cdot 2^6 + 0 \cdot 2^5 + 1 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0$$

$$151 = 2^7 + 2^4 + 2^2 + 2^1 + 2^0$$

Następnie możemy zapisać:

$$151 * P = 2^7 * P + 2^4 * P + 2^2 * P + 2^1 * P + 2^0 * P$$

Jakie są kroki algorytmu podwój i dodaj:

- weź punkt  $P$ ;
- podwój go, aby uzyskać  $2P$ ;
- dodaj  $2P$  do  $P$  (aby uzyskać wynik  $2^1P + 2^0P$ );

- podwój  $2P$ , aby uzyskać  $2^2P$ ;
- dodaj to do wyniku (aby uzyskać wynik  $2^2P + 2^1P + 2^0P$ );
- podwój  $2^2P$ , aby uzyskać  $2^3P$ ;
- nie wykonuj dodawania z  $2^3P$ ;
- podwój  $2^3P$ , aby uzyskać  $2^4P$ ;
- dodaj to do wyniku (aby uzyskać wynik  $2^4P + 2^2P + 2^1P + 2^0P$ );
- ....

Ostatecznie możemy obliczyć  $151 * P$  wykonując tylko siedem podwojeń i cztery dodawania. Jeżeli zarówno podwajanie jak i dodawanie ma złożoność na poziomie  $O(1)$ , wtedy algorytm podwój i dodaj ma złożoność  $O(\log n)$  (lub  $O(k)$ , jeżeli rozpatrujemy długość bitową), co jest bardzo dobrym wynikiem, znacznie lepszym niż początkowy algorytm o złożoności  $O(n)$ .

## 2.3 Krzywe eliptyczne nad ciałem skończonym $F_p$

Ciało skończone to przede wszystkim zbiór o skończonej liczbie elementów. Jego przykładem jest zbiór liczb całkowitych modulo  $p$ , gdzie  $p$  jest liczbą pierwszą. Oznaczane jest, jako  $F_p$ . W ciele można wykonywać dwie operacje binarne: dodawanie (+) i mnożenie (\*). Obie te operacje są zamknięte, asocjacyjne i przemienne. Dla obu istnieje unikatowy element tożsamości, a dla każdego elementu znajduje się unikatowy element odwrotny. Mnożenie jest rozdzielcze względem dodawania:

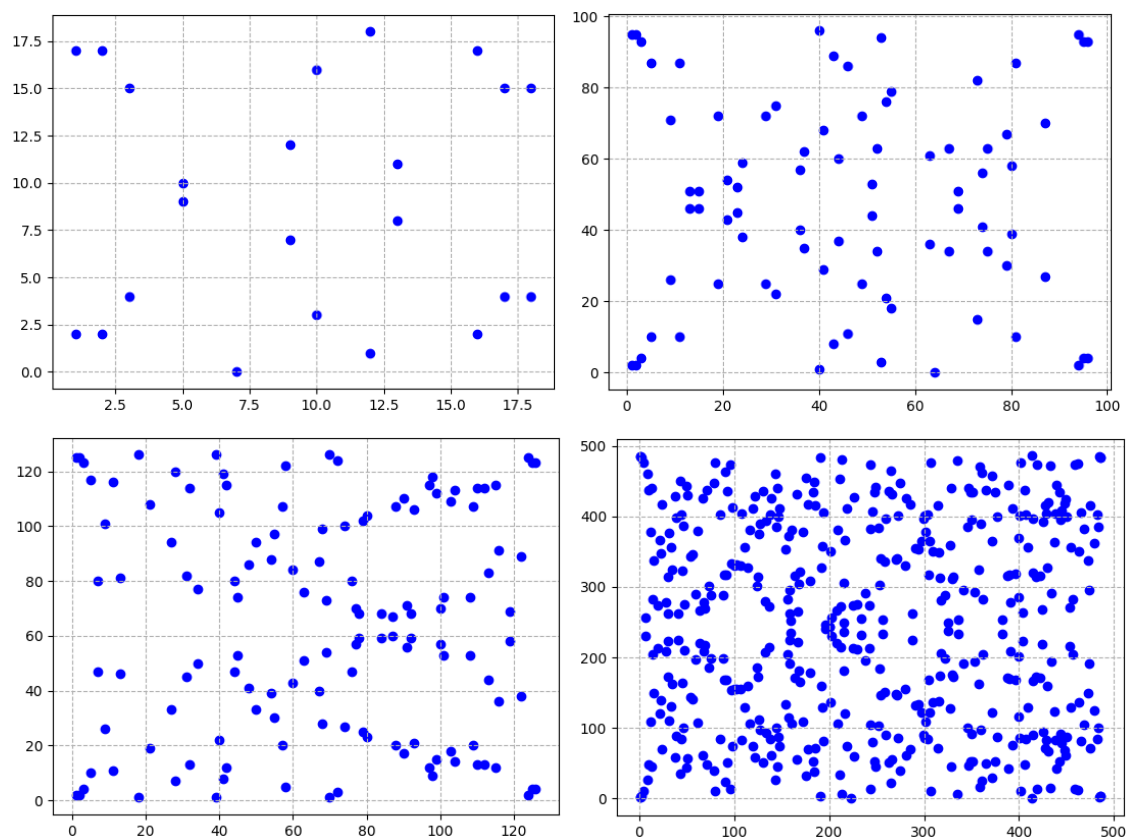
$$x * (y + z) = x * y + x * z$$

Zbiór liczb całkowitych modulo  $p$ , składa się ze wszystkich liczb całkowitych od 0 do  $p-1$ .

Gdy przedstawione zostało, czym jest ciało, należy przedstawić czym są krzywe eliptyczne nad ciałem skończonym  $F_p$ . Zestaw punktów przedstawiony w części 2.1, teraz staje się:

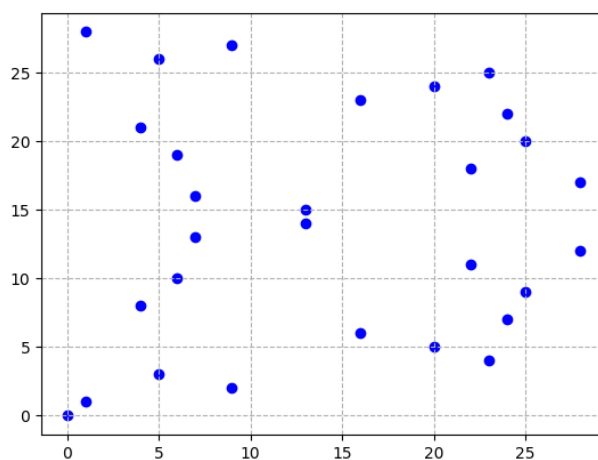
$$\{(x, y) \in (F_p)^2 \mid y^2 = x^3 + ax + b \bmod(p), 4a^3 + 27b \bmod(p) \neq 0\} \cup \{0\}$$

gdzie: 0 jest nadal punktem w nieskończoności, a oraz  $b$  są liczbami całkowitymi w  $F_p$ . Poniżej przedstawione zostały wykresy prawidłowych krzywych eliptycznych nad ciałem skończonym  $F_p$  (Rys. 2.3.1.). Użyta została krzywa  $y^2 = x^3 - 7x + 10 \pmod{p}$  z  $p$  wynoszącym 19, 97, 127, 487. Można zaobserwować, iż dla każdego  $x$ , istnieją co najwyżej dwa punkty.



Rys. 2.3.1. Prawidłowe krzywe eliptyczne nad ciałem skończonym  $F_p$

Natomiast poniżej zaprezentowano przykład nieprawidłowej krzywej eliptycznej nad ciałem skończonym  $F_p$  (Rys. 2.3.2.). Jest to krzywa  $y^2 = x^3 \pmod{29}$ , która jest osobliwa i posiada punkt  $(0,0)$ .



Rys. 2.3.2. Nieprawidłowa krzywa eliptyczna nad ciałem skończonym  $F_p$



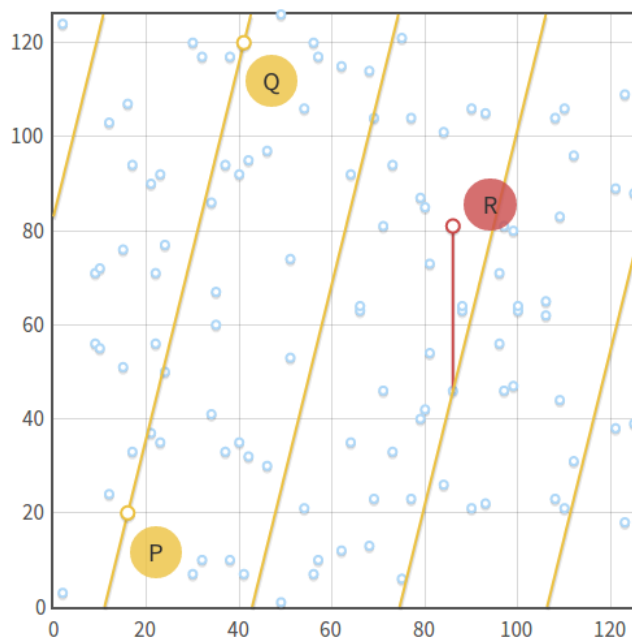
To, co wcześniej było krzywą ciągłą, teraz jest zbiorem rozłącznych punktów w polu  $x, y$ . Krzywe eliptyczne nad dowolnym ciałem  $F_p$  nadal tworzą grupę abelową.

### 2.3.1 Dodawanie geometryczne i algebraiczne

#### Dodawanie geometryczne

Aby definicja przedstawiona w części 2.2.1, działała nad ciałem  $F_p$ , trzeba ją zmodyfikować. W przypadku liczb rzeczywistych stwierdzono, iż suma trzech wyrównanych punktów wynosi zero. Lecz co to oznacza nad ciałem  $F_p$ ? Nieformalnie możemy powiedzieć, iż linia w  $F_p$  to zbiór punktów  $(x, y)$  spełniających równanie  $ax + by + c = 0 \pmod{p}$ .

Na Rys. 2.3.1.1 przedstawiono dodawanie geometryczne na krzywej  $y^2 = x^3 - x + 3 \pmod{123}$ , gdzie  $P = (16, 20)$ ,  $Q = (41, 120)$ . Proszę zauważyć, jak linia łącząca punkty „powtarza się” w płaszczyźnie.



Rys. 2.3.1.1. Dodawanie geometryczne punktów na krzywej eliptycznej nad dowolnym ciałem  $F_p$

Z uwagi na to, iż punkty krzywej należą do grupy, dodawanie zachowuje właściwości, które już przedstawiono.

### Dodawanie algebraiczne

Równania do obliczania sumy punktów są dokładnie takie same jak w rozdziale 2.2.1, z wyjątkiem tego, że należy dodać „mod p” na końcu wyrażenia. Dlatego dane  $P = (px, py)$ ,  $Q = (qx, qy)$ , oraz  $R = (rx, ry)$  obliczamy  $P + Q = -R$  w następujący sposób:

$$\begin{aligned}rx &= (m^2 - px - qx) \bmod p \\ ry &= [py + m(rx - px)] \bmod p \\ ry &= [qy + m(rx - qx)] \bmod p\end{aligned}$$

Jeżeli:  $P \neq Q$ , nachylenie  $m$  przyjmuje postać:

$$m = (py - qy)(px - qx)^{-1} \bmod p$$

W przeciwnym razie, gdy  $P = Q$ , otrzymujemy:

$$m = (3px^2 + a)(2py)^{-1} \bmod p$$

Równania nie zmieniły się, w rzeczywistości działają one w każdej dziedzinie, skończonej lub nieskończonej (z wyjątkiem  $F_2$  i  $F_3$ , które są wyjątkowymi przypadkami).

### 2.3.2 Mnożenie przez skalar i cykliczne podgrupy

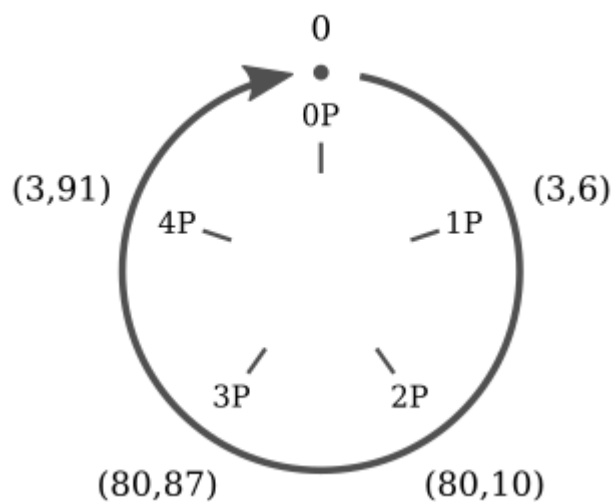
Krzywa eliptyczna nad ciałem skończonym  $F_p$ , posiada skończoną liczbę punktów. Liczba punktów w grupie nazywana jest porządkiem grupy. Do obliczania porządku grupy wykorzystuje się algorytm Schoofa.

Tak jak w liczbach rzeczywistych, mnożenie może być określone, jako:

$$nP = \underbrace{P + P + \dots + P}_{n \text{ razy}}$$

I ponownie możemy wykorzystać algorytm podwój i dodaj.

Mnożenie punktów krzywej eliptycznej nad ciałem skończonym  $F_p$ , posiada ciekawą własność. Dla przykładu weźmy krzywą  $y^2 = x^3 + 2x + 3 \pmod{97}$  oraz punkt  $P = (3, 6)$ . Korzystając z przygotowanej aplikacji obliczamy wielokrotność tego punktu. Wielokrotnościami punktu  $P$ , jest tylko pięć odrębnych punktów  $(0, P, 2P, 3P, 4P)$ , które cyklicznie się powtarzają (Rys. 2.3.2.1).



Rys. 2.3.2.1. Cykliczność wielokrotności punktu krzywej eliptycznej nad ciałem skończonym  $F_p$

- $0P = 0$
- $1P = (3, 6)$
- $2P = (80, 10)$
- $3P = (80, 87)$
- $4P = (3, 91)$
- $5P = 0$
- $6P = (3, 6)$
- $7P = (80, 10)$
- $8P = (80, 87)$
- $9P = (3, 91)$
- ....

Możemy zauważyć dwie sprawy. Po pierwsze, jest tylko pięć wielokrotności  $P$ , pozostałe punkty krzywej eliptycznej nigdy się nie pojawiają. Po drugie punkty te powtarzają się cyklicznie. A więc możemy zapisać:

- $5kP = 0$
- $(5k + 1)P = P$
- $(5k + 2)P = 2P$
- $(5k + 3)P = 3P$
- $(5k + 4)P = 4P$

Z tego wynika:  $kP = (k \bmod 5)P$ .

Dotyczy to każdego punktu. Jeżeli weźmiemy dowolny punkt P:

$$nP + mP = \underbrace{P + P + \dots + P}_{n \text{ razy}} + \underbrace{P + P + \dots + P}_{m \text{ razy}} = (n+m)P$$

Z tego wynika, że jeśli odamy dwie wielokrotności punktu P, to otrzymamy wielokrotność punktu P. Oznacza to, że zbiór wielokrotności punktu P jest cykliczną podgrupą grupy, utworzonej przez krzywą eliptyczną. Punkt P jest generatorem lub punktem bazowym podgrupy cyklicznej. Podgrupy cykliczne są podstawą kryptografii krzywych eliptycznych.

### Kolejność podgrupy

Ważną kwestią jest ustalenie porządku podgrupy generowanej przez punkt P. Niestety nie możemy do tego celu wykorzystać algorytmu Schoofa, ponieważ działa on tylko na całych krzywych eliptycznych, a nie na podgrupach. Aby rozwiązać ten problem, należy założyć:

- Porządek zdefiniowaliśmy, jako liczbę punktów w grupie. W ramach cyklicznej podgrupy, możemy go zdefiniować, jako: porządek P jest najmniejszą dodatnią liczbą całkowitą n, taką, że  $nP = 0$ . W podanym przykładzie, podgrupa zawierała pięć elementów i  $5P = 0$ .
- Porządek P jest związany z porządkiem krzywej eliptycznej poprzez twierdzenia Lagrange'a, mówiącym, że porządek podgrupy jest dzielnikiem porządku grupy macierzystej. Czyli, jeżeli krzywa eliptyczna zawiera N punktów i jedna z jego podgrup zawiera n punktów, to n jest dzielnikiem N.

Dzięki tym założeniom, uzyskujemy sposób na ustalenie kolejności podgrupy z punktem bazowym P:

- 1) Obliczamy porządek krzywej eliptycznej N przy użyciu algorytmu Schoofa.
- 2) Wyszukujemy wszystkie dzielniki N.
- 3) Dla każdego dzielnika n z N obliczmy  $nP$ .
- 4) Porządek podgrupy to najmniejsze n takie, że  $nP = 0$ .

Dla przykładu weźmy krzywą  $y^2 = x^3 - x + 3 \pmod{37}$ . Krzywa ta ma porządek  $N = 42$ , jej podgrupy mogą mieć porządek  $n = 1, 2, 3, 6, 7, 14, 21$  lub 42. Jeżeli sprawdzimy punkt  $P = (2, 3)$ , to zauważymy, że  $P \neq 0$ ,  $2P \neq 0$ , ...,  $7P = 0$ , stąd porządek P wynosi  $n = 7$ .

Bardzo istotne jest to, aby wziąć najmniejszy dzielnik, a nie losowy, gdyż może to doprowadzić do błędnego wyliczenia porządku podgrupy. Kiedy ilość dzielników  $n = 1$ , podgrupa zawiera tylko punkt w nieskończoności, natomiast, gdy  $n = N$ , podgrupa zawiera wszystkie punkty krzywej eliptycznej.

### Znajdowanie punktu bazowego

Algorytmy wykorzystujące kryptografię krzywych eliptycznych potrzebują podgrup o wysokim rzędzie. Aby dokonać odpowiedniego wyboru należałoby wybrać krzywą eliptyczną, obliczyć jej porządek ( $N$ ), dobrać najmniejszy dzielnik, jako kolejność podgrupy ( $n$ ) i ostatecznie wybrać punkt bazowy. W rzeczywistości nie wybieramy punktu bazowego i obliczamy jego porządku, tylko w pierwszej kolejności wybieramy porządek podgrupy, który jest wystarczająco dobry a następnie wyszukujemy odpowiedni punkt bazowy.

Aby tego dokonać, należy wprowadzić jeszcze jeden termin. Z twierdzenia Lagrange'a wynika, iż liczba  $h = N / n$  jest zawsze liczbą całkowitą, (ponieważ  $n$  jest dzielnikiem  $N$ ). Liczba  $h$  jest nazywana kofaktorem podgrupy.

Dla każdego punktu krzywej eliptycznej mamy  $NP = 0$ . Spowodowane jest to tym, ponieważ  $N$  jest wielokrotnością  $n$ . Korzystając z definicji kofaktora, można zapisać:

$$n(hP) = 0$$

Załóżmy teraz, że  $n$  jest liczbą pierwszą. Równanie to, mówi nam, że punkt  $G = hP$ , generuje podgrupę porządku  $n$  (z wyjątkiem kiedy  $G = hP = 0$ , w takim przypadku podgrupa ma porządek 1).

Podsumowując, na podstawie powyższych informacji można zarysować następujący algorytm:

- 1) Oblicz porządek  $N$  krzywej eliptycznej.
- 2) Wybierz porządek podgrupy  $n$ . Aby algorytm zadziałał, liczba ta musi być liczbą pierwszą oraz dzielnikiem  $N$ .
- 3) Oblicz kofaktor  $h = N / n$ .
- 4) Wybierz losowy punkt  $P$  na krzywej.
- 5) Oblicz  $G = hP$ .
- 6) Jeżeli  $G$  wynosi 0, wróć do kroku 4. W przeciwnym razie znaleźliśmy generator podgrupy z porządkiem  $n$  i kofaktorem  $h$ .

Algorytm ten działa tylko wtedy gdy  $n$  jest liczbą pierwszą.

## Logarytm dyskretny

Załóżmy, że znamy wartość punktów  $P$  i  $Q$  na krzywej eliptycznej nad ciałem skończonym  $F_p$ . Czy jesteśmy w stanie odnaleźć taki  $k$  żeby było spełnione równanie  $Q = kP$ ? Otóż jest to zagadnienie znane, jako problem logarytmu dyskretnego dla krzywych eliptycznych. Uznawane jest za „trudne”, ponieważ nie ma znanego algorytmu wielomianowego, który mógłby działać na klasycznym komputerze.

Problem ten jest analogiczny do problemu logarytmów dyskretnych stosowanego w innych kryptosystemach, takich jak algorytm podpisu cyfrowego (DSA), wymiana kluczy Diffie – Hellman (DH) i algorytm ElGamala. Różnica polega na tym, że w przypadku tych algorytmów używamy potęgowania modulo zamiast mnożenia przez skalar. Ich problem z logarytmem dyskretnym można sformułować następująco: jeżeli znamy  $a$  oraz  $b$ , czym jest  $k$ , takie, że  $b = a^k \bmod p$ .

Oba te problemy są dyskretnie, ponieważ dotyczą zbiorów skończonych (a dokładniej cyklicznych podgrup).

To, co sprawia, że kryptografia krzywych eliptycznych jest interesująca, to fakt, że na dzień dzisiejszy problem logarytmu dyskretnego dla krzywych eliptycznych wydaje się być trudniejszy w porównaniu z innymi podobnymi problemami stosowanymi w kryptografii. Oznacza to, że potrzebujemy mniej bitów na liczbę całkowitą  $k$  w celu osiągnięcia takiego samego poziomu bezpieczeństwa, jak w przypadku innych kryptosystemów.

### 3. Kryptografia oparta na krzywych eliptycznych

Kryptografia krzywych eliptycznych (ECC) jest nowoczesną rodziną kryptosystemów z kluczem publicznym, opierająca się na strukturach algebraicznych krzywych eliptycznych nad ciałem skończonym oraz na trudności problemu logarytmu dyskretnego krzywej eliptycznej (ECDLP). Implementuje ona wszystkie główne możliwości asymetrycznych kryptosystemów: szyfrowanie, podpisy, wymiana kluczy. Kryptografia krzywych eliptycznych jest uznawana za naturalnego współczesnego następcę kryptosystemu RSA, ponieważ używa mniejszych kluczy i podpisów niż RSA dla tego samego poziomu bezpieczeństwa i zapewnia bardzo szybkie generowanie kluczy, uzgadnianie kluczy oraz podpisywanie.

Na początek przedstawię, czym jest klucz prywatny i publiczny w kryptografii krzywych eliptycznych:

- 1) Klucz prywatny jest losową liczbą całkowitą  $d$  wybraną z  $\{1, \dots, n - 1\}$ , gdzie:  $n$  to porządek podgrupy
- 2) Klucz publiczny  $H = dG$ , gdzie:  $G$  jest punktem bazowym podgrupy.

Jeżeli znamy  $d$  oraz  $G$  (wraz z innymi parametrami krzywej eliptycznej), znajdowanie  $H$  jest proste. Natomiast, gdy znamy  $H$  oraz  $G$ , odnajdowanie klucza prywatnego  $d$  jest „trudne”, ponieważ wymaga od nas rozwiązanie problemu logarytmu dyskretnego.

W dalszej części pracy opisane zostaną oparte na nich algorytmy klucza publicznego: szyfrowanie ElGamala, ECDH (krzywa eliptyczna Diffie-Hellmana), służące do szyfrowania oraz ECDSA (algorytm podpisu cyfrowego krzywej eliptycznej), używany do podpisywania cyfrowego.

#### 3.1. Algorytm ElGamala z implementacją krzywych eliptycznych

W 1985r. Taher Elgamal opublikował artykuł, w zaprezentował system szyfrowania logarytmem dyskretnym ElGamal, a także schemat podpisu ElGamal, (który stał się rdzeniem metody podpisu DSA). W 2009 roku Elgamal otrzymał nagrodę RSA Conference 2009 Lifetime Achievement Award i został nazwany „ojcem SSL”.

U podstaw metod klucza publicznego ElGamal leży problem logarytmu dyskretnego, w którym mamy:

$$Y = g^x \pmod{p}$$

Gdzie trudno określić  $x$ , nawet, jeśli mamy  $Y$ ,  $g$  i  $p$  (o ile  $p$  jest wystarczająco dużą liczbą pierwszą). Używany jest w metodzie Diffie-Hellmana do wymiany kluczy. Wykorzystywany jest również do podpisywania wiadomości, gdzie tworzy parę kluczy (klucz publiczny i klucz prywatny). Klucz prywatny służy do zaszyfrowania czegoś (na przykład hashu wiadomości), a następnie do potwierdzenia podpisu służy klucz publiczny.

Algorytm ElGamal z implementacją krzywych eliptycznych jest modyfikacją klasycznego algorytmu ElGamal. Metoda ta polega na dodawaniu i odejmowaniu punktów na krzywej eliptycznej. Aby zaszyfrować za jego pomocą należy wybrać bezpieczną krzywą eliptyczną (np. secp256k1).

Poniżej zaprezentowany jest schemat działania ElGamal ECC:

- 1) Na początek Kasia tworzy klucz prywatny  $d_K$ , który jest losową wartością skalarną.
- 2) Następnie obliczamy klucz publiczny Kasi  $H_K = d_K P$ , gdzie:  $P$  jest punktem bazowym krzywej eliptycznej.
- 3) Jeżeli Tomek chce wysłać zaszyfrowaną wiadomość  $M$ , tworzy losową wartość  $k$  i używa klucza publicznego Kasi  $H_K$ , aby stworzyć zaszyfrowaną wiadomość:  $K = kP$  i  $C = kH_K + M$ , gdzie:  $M$  jest połączone z punktem na krzywej eliptycznej.
- 4) Kasia otrzymuje szyfrogram  $(K, C)$  i oblicza wspólny klucz tajny  $S$  używając swojego klucza prywatnego  $d_K$ :  $S = d_K K$
- 5) A następnie odszyfrowuje wiadomość:  $M = C - S$

Wiedząc, że  $C$  oraz  $S$  są punktami na krzywej eliptycznej, więc zostanie to zrobione za pomocą operacji odejmowania punktów. Ogólnie rzecz biorąc, przywróci to oryginalną wiadomość, jako:

$$C - S = kH_K + M - d_K K = kd_K P + M - kd_K P = M$$

W przygotowanej aplikacji komputerowej, zaimplementowałem szyfrowanie ElGamal z wykorzystaniem krzywych eliptycznych. Można wypróbować tą metodę na różnych krzywych eliptycznych. A plikacji tej skupiono się na przedstawieniu metody ElGamal, więc pominięto etap kodowania wiadomości i skupiono się na szyfrowaniu punktów krzywej eliptycznej.



### 3.2 Algorytm Diffiego–Hellmana w przestrzeni krzywych eliptycznych ECDH

ECDH (ang. Elliptic Curve Diffie-Hellman) jest wariantem algorytmu Diffiego-Hellmana dla krzywych eliptycznych. W rzeczywistości jest to protokół uzgodnienia kluczy, a nie algorytm szyfrowania. Zasadniczo oznacza to, iż ECDH definiuje (do pewnego stopnia), sposób generowania i wymiany kluczy między stronami. To jak zostaną zaszyfrowane dane za pomocą takich kluczy, zależy od nas samych.

Problem, który rozwiązuje ECDH jest następujący: dwie strony (np. Kasia i Tomek) chcą bezpiecznie wymieniać informacje, tak, aby osoba trzecia (Człowiek po środku) mogła je przechwycić, ale nie mogła ich odszyfrować. To jedna z zasad TLS (ang. Transport Layer Security).

Poniżej zaprezentowany jest sposób działania ECDH:

- 1) Na początek Kasia i Tomek generują własne klucze prywatne i publiczne: mamy klucz prywatny  $d_K$  i klucz publiczny  $H_K = d_K G$  dla Kasi oraz klucz prywatny  $d_T$  i  $H_T = d_T G$  dla Tomka. Proszę zwrócić uwagę, że zarówno Kasia jak i Tomek używają tych samych parametrów, tego samego punktu bazowego  $G$  na tej samej krzywej eliptycznej nad takim samym ciałem skończonym.
- 2) Kasia i Tomek wymieniają swoje klucze publiczne  $H_K$  i  $H_T$  przez niezabezpieczony kanał. Człowiek w środku może przechwycić  $H_K$  i  $H_T$ , lecz nie pozna kluczy prywatnych  $d_K$  i  $d_T$ , bez rozwiązania problemu logarytmu dyskretnego.
- 3) Kasia oblicza  $S = d_K H_T$  (używając własnego klucza prywatnego i klucza publicznego Tomka), a Tomek oblicza  $S = d_T H_K$  (używając własnego klucza prywatnego i klucza publicznego Kasi). Proszę zwrócić uwagę, iż  $S$  jest takie same dla Kasi i Tomka, ponieważ:

$$S = d_K H_T = d_K (d_T G) = d_T (d_K G) = d_T H_K$$

Człowiek w środku zna tylko  $H_K$  i  $H_T$  (wraz z innymi parametrami krzywej eliptycznej) i nie jest w stanie znaleźć wspólnego sekretu  $S$ . Jest to znane jako problem Diffiego-Hellmana, który możemy przedstawić w następujący sposób:

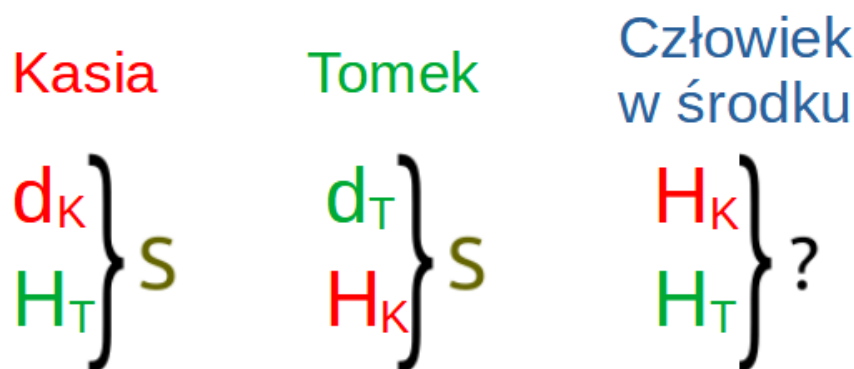
- Biorąc pod uwagę trzy punkty  $P$ ,  $aP$  oraz  $bP$ , jaki jest wynik  $abP$ ?

Lub równoważnie:

- Biorąc pod uwagę trzy liczby całkowite  $k$ ,  $k^x$  oraz  $k^y$ , jaki jest wynik  $k^{xy}$ ?

Ta druga forma jest używana w oryginalnym algorytmie Diffiego-Hellmana, opartym na arytmetyce modularnej.

Poniżej przedstawiono schemat wymiany kluczy Diffiego-Hellmanna: Kasia i Tomek mogą „łatwo” obliczyć wspólny sekret, a człowiek w środku musi rozwiązać „trudny” problem (Rys. 3.2.1).



Rys. 3.2.1. Schemat wymiany kluczy Diffiego-Hellmanna

Zakłada się, iż problem Diffiego-Hellmana dla krzywych eliptycznych jest tak „trudny”, jak problem logarytmu dyskretnego. Rozwiązanie problemu logarytmicznego jest sposobem rozwiązania problemu Diffie-Hellmana.

Gdy Kasia i Tomek zdobyli wspólny sekret, mogą wymieniać dane za pomocą szyfrowania symetrycznego. Na przykład mogą użyć współrzędną  $x$  punktu  $S$ , jako klucz do szyfrowania wiadomości używając bezpiecznych szyfrów takich jak AES lub 3DES. Mniej więcej w taki sposób działa TLS. Różnica polega na tym, że TLS wiąże współrzędną  $x$  z innymi liczbami względem połączenia, a następnie oblicza hash z ciągu bitów.

### 3.3 Algorytm podpisu cyfrowego z wykorzystaniem krzywych eliptycznych ECDSA

ECDSA (ang. Elliptic Curve Digital Signature Algorithm) to wariant algorytmu podpisu cyfrowego z wykorzystaniem krzywych eliptycznych. Pracuje on na hashu wiadomości, a nie na samej wiadomości. Wybór funkcji hashującej zależy od nas, ale należy wybrać funkcję hashującą bezpieczną kryptograficznie (MDA, SHA-1, SHA-2). Hash wiadomości powinien zostać skrócony, aby jego długość w bitach była taka sama, jak długość w bitach  $n$  (kolejność podgrupy). Skrócony hash jest liczbą całkowitą i oznaczmy go literą  $z$ .

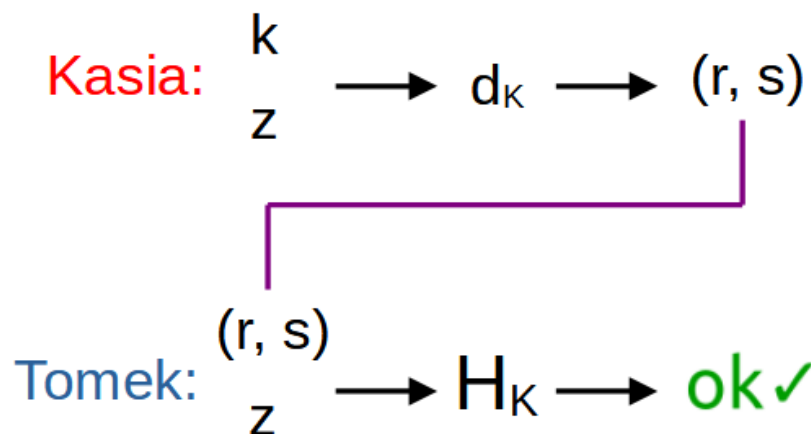
Teraz przedstawię schemat ECDSA na przykładzie. Załóżmy, że Kasia chce podpisać wiadomość swoim prywatnym kluczem ( $d_K$ ), a Tomek chce sprawdzić czy podpis jest prawidłowy za pomocą klucza publicznego Kasi ( $H_K$ ). Nikt poza Kasią, nie powinien być w stanie wykonać prawidłowego podpisu. Każdy powinien mieć możliwość sprawdzenia prawidłowości podpisu.

Algorytm wykonywany przez Kasię do podpisania wiadomości działa w następujący sposób:

- 1) Wybierz losową liczbę całkowitą  $k$  ze zbioru  $\{1, \dots, n - 1\}$ , gdzie:  $n$  jest porządkiem podgrupy.
- 2) Oblicz punkt  $P = kG$ , gdzie:  $G$  jest punktem bazowym.
- 3) Oblicz liczbę  $r = px \bmod n$ , gdzie:  $px$  jest współrzędną  $x$  punktu  $P$ .
- 4) Jeżeli  $r = 0$ , należy wybrać inną wartość  $k$  i spróbować ponownie.
- 5) Oblicz  $s = k^{-1}(z + rd_K) \bmod n$ , gdzie:  $d_K$  jest kluczem prywatnym Kasi,  $k^{-1}$  jest wielokrotnością odwrotności  $k$  modulo  $n$ .
- 6) Jeżeli  $s = 0$ , należy wybrać inną wartość  $k$  i spróbować ponownie.

Para  $(r, s)$  jest podpisem.

Na rysunku 3.2.1 przedstawiono prosty schemat składania i weryfikacji podpisu cyfrowego.



Rys. 3.2.1. Schemat składania i weryfikacji podpisu cyfrowego

Mówiąc wprost, algorytm ten najpierw generuje sekret ( $k$ ). Następnie ukrywa go w  $r$ , dzięki mnożeniu punktów, (co jest „łatwe” w jedną stronę i „trudne” w drugą stronę). W kolejnym kroku  $r$  zostaje powiązane z haszem wiadomości przez równanie:

$$s = k^{-1}(z + rd_K) \bmod n$$

Aby obliczyć  $s$ , w pierw należy wyznaczyć odwrotność  $k$  modulo  $n$ . Da się tego dokonać tylko wtedy, gdy  $n$  jest liczbą pierwszą. Jeżeli porządek podgrupy nie jest liczbą pierwszą, nie może zostać użyta w ECDSA. Nie jest przypadkiem, że wszystkie krzywe standaryzowane mają porządek który jest liczbą pierwszą.

Aby zweryfikować podpis potrzebny jest klucz publiczny  $H_K$ , obciety hash  $z$  oraz podpis  $(r, s)$ . Następnie wykonywane są kolejne kroki:

- 1) Oblicz liczbę całkowitą  $u_1 = s^{-1}z \bmod n$ .
- 2) Oblicz liczbę całkowitą  $u_2 = s^{-1}r \bmod n$
- 3) Oblicz punkt  $P = u_1G + u_2H_K$ .

Podpis jest ważny tylko wtedy, gdy  $r = px \bmod n$

### Poprawność algorytmu

Logika stojąca za tym algorytmem może na pierwszy rzut oka nie wydawać się oczywista, jednak, jeżeli połączymy wszystkie równania, które do tej pory przedstawiono, wszystko będzie jaśniejsze.

Zacznijmy od  $P = u_1G + u_2H_K$ . Z definicji klucza publicznego, wiemy że  $H_K = d_KG$  (gdzie  $d_K$  jest kluczem prywatnym). Można zapisać:

$$\begin{aligned} P &= u_1G + u_2H_K \\ P &= u_1G + u_2d_KG \\ P &= (u_1 + u_2d_K)G \end{aligned}$$

Korzystając z definicji  $u_1$  i  $u_2$ , możemy zapisać:

$$\begin{aligned} P &= (u_1 + u_2d_K)G \\ P &= (s^{-1}z + s^{-1}rd_K)G \\ P &= s^{-1}(z + rd_K)G \end{aligned}$$

Pominięte tutaj zostało  $\bmod n$ , zarówno dla zwięzłości, jak i dlatego, że podgrupa cykliczna generowana przez  $G$  ma porządek  $n$ . W związku z tym  $\bmod n$  jest zbędne.

Wcześniej zdefiniowane zostało  $s = k^{-1}(z + rd_K) \bmod n$ . Mnożąc każdą ze stron przez  $k$  i dzieląc przez  $s$  otrzymujemy:  $k = s^{-1}(z + rd_K) \bmod n$ . Podstawiając ten wynik do naszego równania  $P$ , otrzymujemy:

$$P = s^{-1}(z + rd_K)G = kG$$

Otrzymane równanie, jest takie samo jak w kroku drugim algorytmu generowania podpisu. Generując i weryfikując podpisy liczymy ten sam punkt P, tylko z innym zestawem równań. Dlatego algorytm działa.

### **Znaczenie k**

Podczas generowania podpisów ECDSA ważne jest, aby zachować wartość k w tajemnicy. Jeżeli będziemy używać tego samego k dla wszystkich podpisów, albo nasz losowy generator był w pewien sposób przewidywalny, włamywacz byłby w stanie odkryć nasz klucz prywatny.

Ten rodzaj błędu popełniło Sony kilka lat wcześniej. Zasadniczo konsola do gier PlayStation 3 mogła uruchamiać tylko gry sygnowane przez Sony z ECDSA. Problem w tym, że wszystkie wykonane podpisy zostały wygenerowane za pomocą statycznego k. W takiej sytuacji można było bez problemu odzyskać klucz prywatny Sony ds kupując tylko dwie podpisane gry, wydobywając ich skrót ( $z_1$  oraz  $z_2$ ) i ich podpisy  $((r_1, s_1)$  i  $(r_2, s_2)$  wraz z parametrami krzywej eliptycznej. Można tego było dokonać w następujący sposób:

- Po pierwsze, zauważ, że  $r_1 = r_2$  (dlatego, że  $r = px \bmod n$  oraz  $P = kG$ , jest taki sam dla obu podpisów).
- Uwzględniając  $(s_1 - s_2) \bmod n = k^{-1}(z_1 - z_2) \bmod n$  (wynik ten pochodzi bezpośrednio z równania na s).
- Mnożymy każdą ze stron równania przez k:  $k(s_1 - s_2) \bmod n = (z_1 - z_2) \bmod n$ .
- Dzielimy przez  $(s_1 - s_2)$  i otrzymujemy:  $k = (z_1 - z_2)(s_1 - s_2)^{-1} \bmod n$ .

Ostatnie równanie pozwala nam obliczyć k używając tylko dwóch skrótów i odpowiadających im podpisów. Teraz można wyodrębnić klucz prywatny za pomocą równania dla s:

$$s = k^{-1}(z + rd_s) \bmod n \Rightarrow d_s = r^{-1}(sk - z) \bmod n$$

Podobne techniki można zastosować, jeżeli k nie jest statyczna, ale w jakiś sposób przewidywalna.

## **3.4 Kryptografia krzywych eliptycznych a RSA**

Należy zadać sobie pytanie: po co wykorzystywać krzywe eliptyczne w kryptografii, jeżeli kryptosystem RSA, wykorzystywany od lat ciągle sprawdza się znakomicie? Otóż z odpowiedzią na to pytanie przychodzi NIST (ang. National Institute

of Standards and Technology), który przedstawia tabelę, która porównuje długość kluczy RSA i ECC wymaganych do osiągnięcia takiego poziomu bezpieczeństwa (Rys.3.4.1).

Długość klucza RSA (w bitach)	Długość klucza ECC (w bitach)
1024	160
2048	224
3072	256
7680	384
15360	521

Rys. 3.4.1. Porównanie długości kluczy RSA i ECC

Można zauważyć, iż nie ma liniowej zależności między długościami klucza RSA, a długościami klucza ECC (innymi słowy, jeżeli podwajamy rozmiar klucza RSA, nie musimy podwajać klucza ECC). Tabela mówi nam nie tylko, że ECC zużywa mniej pamięci, ale także, że generowanie i podpisywanie kluczy jest znacznie szybsze.

## 4. Część programistyczna

W tej części pracy przedstawiona została, stworzona przeze mnie aplikacja komputerowa, która ma za zadanie zapoznać użytkownika z krzywymi eliptycznymi (nad ciałem rzeczywistym oraz nad ciałem skończonym), z działaniami, jakie można wykonywać na punktach krzywej eliptycznej (dodawanie punktów, mnożenie punktów). Dodatkowo zademonstrowane zostało szyfrowanie i odszyfrowywanie punktów za pomocą metody ElGamal z użyciem krzywych eliptycznych oraz wykorzystanie podpisu cyfrowego ECDSA (podpisywanie wiadomości oraz weryfikacja poprawności podpisu).

### 4.1 Technologie wykorzystane przy tworzeniu aplikacji

Do stworzenia aplikacji wykorzystano środowisko programistyczne JetBrains PyCharm Edu 2022. Jest ono wykorzystywane przy tworzeniu aplikacji komputerowych w języku Python, który jest obecnie jednym z najpopularniejszych i najszybciej rozwijającym się językiem programowania. Głównymi bibliotekami wykorzystanymi w trakcie tworzenia aplikacji były:

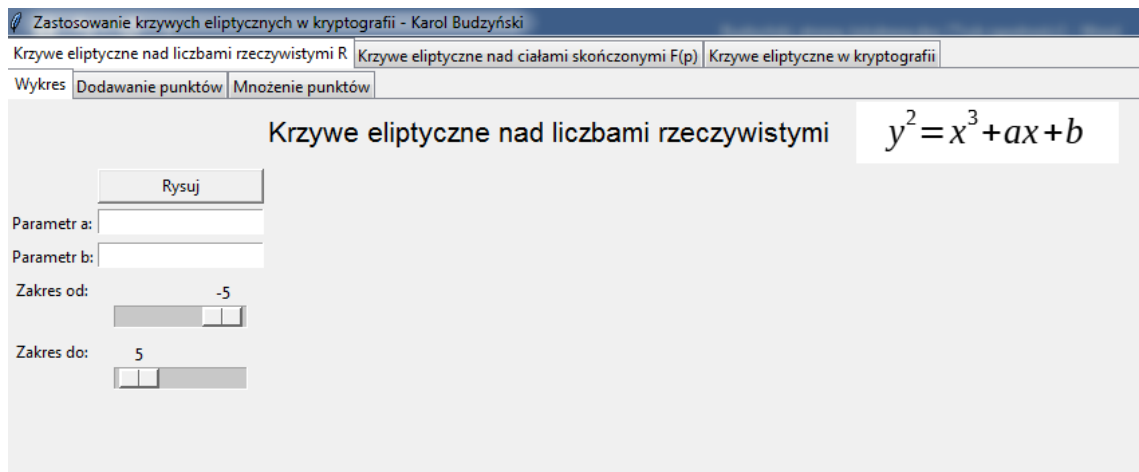
- NumPy - biblioteka umożliwiająca obsługę tabel i macierzy,
- Matplotlib - biblioteka pozwalająca na tworzenie wykresów
- Tkinter – biblioteka odpowiedzialna za tworzenie interfejsu graficznego.

### 4.2 Funkcjonalność aplikacji

Aplikacja podzielona została na trzy główne części:

- Krzywe eliptyczne nad liczbami rzeczywistymi  $\mathbb{R}$ ,
- Krzywe eliptyczne nad ciałami skończonymi  $\mathbb{F}(p)$ ,
- Krzywe eliptyczne w kryptografii.

Po uruchomieniu aplikacji, na ekranie ukazuje się okno główne z menu znajdującym się na samej górze (Rys. 4.1). Program domyślnie uruchamia się na pierwszej pozycji menu, czyli na *Krzywe eliptyczne nad liczbami rzeczywistymi  $\mathbb{R}$* . W podmenu pierwszej pozycji mamy do wyboru następujące pozycje: *Wykres*, *Dodawanie punktów*, *Mnożenie punktów*.

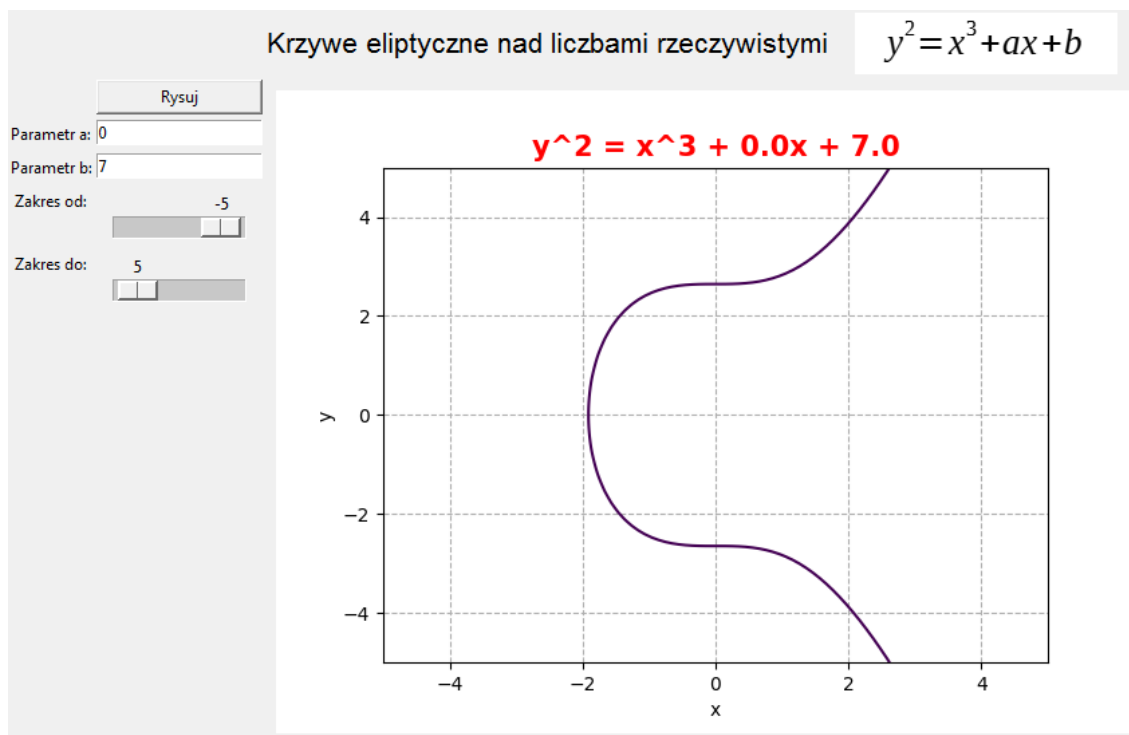


Rys. 4.1. Krzywe eliptyczne nad liczbami rzeczywistymi - Wykres

### Krzywe eliptyczne nad liczbami rzeczywistymi R - Wykres

W pierwszym oknie aplikacji mamy możliwość wyznaczenia krzywej eliptycznej nad liczbami rzeczywistymi oraz przedstawienia jej wykresu. W tym celu należy podać parametry interesującej nas krzywej eliptycznych w polach *Parametr a* oraz *Parametr b*. Po prawidłowym uzupełnieniu danych należy wcisnąć przycisk *Rysuj*. Jeżeli wprowadzone dane są poprawne, aplikacja przedstawi wykres krzywej eliptycznej (Rys. 4.2). Istnieje możliwość regulacji zakresu wykresu za pomocą suwaków *Zakres od* i *Zakres do* w granicach od -100 do 100 (domyślnie -5 i 5). Gdy wprowadzone dane nie będą prawidłowe, pojawi się informacja z prośbą o podanie prawidłowych parametrów.





Rys. 4.2. Wykres krzywej eliptycznej nad liczbami rzeczywistymi  $\mathbb{R}$

### Krzywe eliptyczne nad liczbami rzeczywistymi $\mathbb{R}$ - Dodawanie punktów

Następną pozycją w pierwszym podmenu jest *Dodawanie punktów* (Rys. 4.3). W oknie tym mamy możliwość dodania dwóch punktów znajdujących się na krzywej eliptycznej.

Zastosowanie krzywych eliptycznych w kryptografii - Karol Budzyński

Krzywe eliptyczne nad liczbami rzeczywistymi  $\mathbb{R}$  | Krzywe eliptyczne nad ciałami skończonymi  $F(p)$  | Krzywe eliptyczne w kryptografii

**Wykres** | **Dodawanie punktów** | Mnożenie punktów

Dodawanie punktów na krzywej eliptycznej  $y^2 = x^3 + ax + b$

**Oblicz**

Parametr a:

Parametr b:

Zakres od:

Zakres do:

Punkt P x:

Punkt P y:

Punkt Q x:

Punkt Q y:

Punkt R = P + Q x:

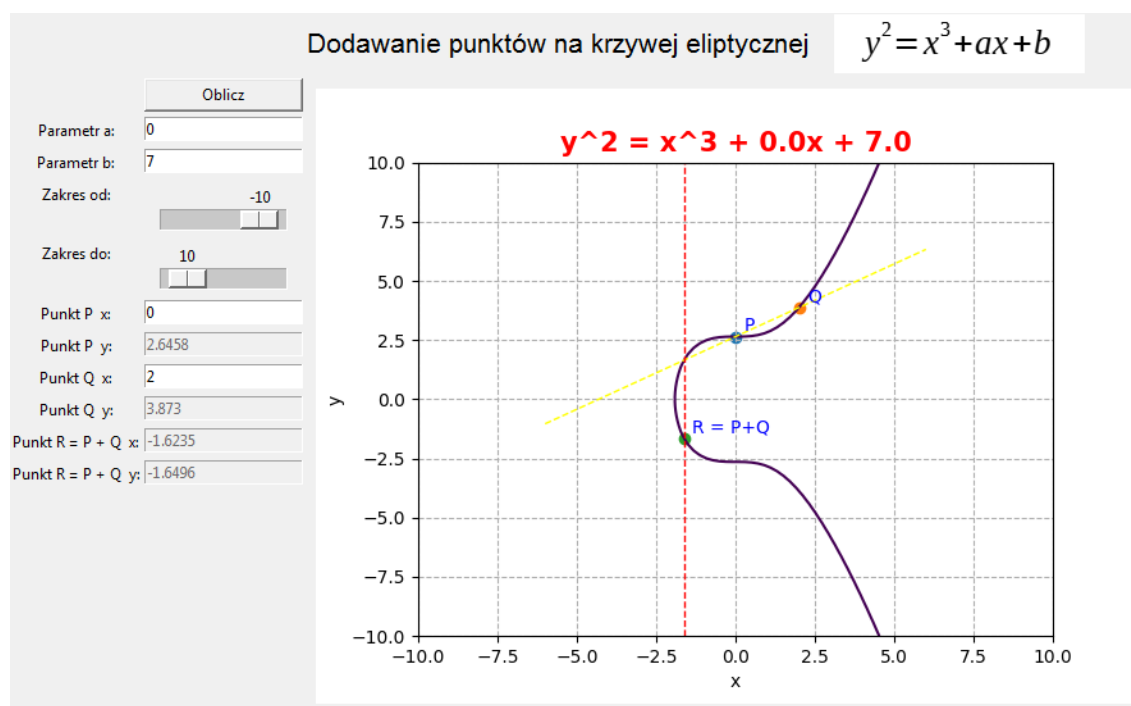
Punkt R = P + Q y:

Rys. 4.3. Krzywe eliptyczne nad liczbami rzeczywistymi – Dodawanie punktów

W tym celu musimy podać parametry interesującej nas krzywej eliptycznej (w polach *Parametr a* oraz *Parametr b*) oraz współrzędną x punktów P i Q, które chcemy dodać (współrzędne y zostaną obliczone automatycznie). Po uzupełnieniu wszystkich pól należy wcisnąć przycisk *Oblicz*. Jeżeli wprowadzone dane są prawidłowe, program narysuje krzywą eliptyczną, zaznaczy na wykresie dodawane punkty P, Q oraz punkt R, który jest wynikiem dodawania. Dodatkowo wynik ten zostanie wyświetlony w polach *Punkt R = P + Q x* i *Punkt R = P + Q y* (Rys. 4.4).

Istnieje możliwość regulacji zakresu wykresu za pomocą suwaków *Zakres od* i *Zakres do* w granicach od -100 do 100 (domyślnie -5 i 5).

Gdy któryś z parametrów zostanie podany nieprawidłowo, bądź punkty P lub Q nie będą znajdować się na krzywej eliptycznej, pojawi się informacja z prośbą o podanie prawidłowych danych.



Rys. 4.4. Wynik dodawania punktów na krzywej eliptycznej nad liczbami rzeczywistymi R

### Krzywe eliptyczne nad liczbami rzeczywistymi R - Mnożenie punktów

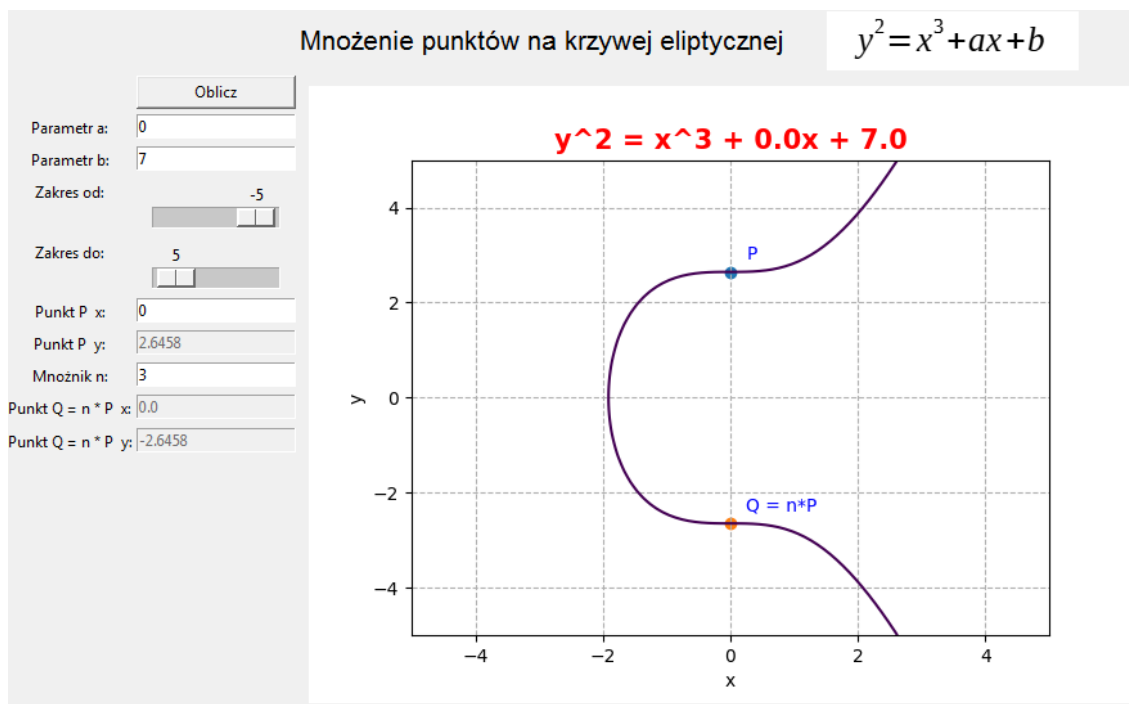
Kolejną opcją w pierwszym podmenu jest *Mnożenie punktów* (Rys. 4.5).

Rys. 4.5. Krzywe eliptyczne nad liczbami rzeczywistymi – Mnożenie punktów

Ta część aplikacji służy do obliczania wielokrotności danego punktu krzywej eliptycznej. Aby aplikacja dokonała tego obliczenia, należy podać parametry interesującej nas krzywej eliptycznej (w polach *Parametr a* oraz *Parametr b*), współrzędną  $x$  punktu  $P$ , którego wielokrotność chcemy uzyskać (współrzędna  $y$  zostanie obliczona automatycznie) oraz mnożnik  $n$ , czyli interesującą nas wielokrotność. Gdy wszystkie pola są uzupełnione, należy wcisnąć przycisk *Oblicz*. Jeżeli przekazane dane są poprawne, aplikacja narysuje krzywą eliptyczną, zaznaczy na wykresie  $P$ , punkt  $R$  będący wielokrotnością  $P * n$ . Wynik ten zostanie wyświetlony w polach *Punkt  $R = n * P$  x* i *Punkt  $R = n * P$  y* (Rys. 4.6).

Istnieje możliwość regulacji zakresu wykresu za pomocą suwaków *Zakres od* i *Zakres do* w granicach od -100 do 100 (domyślnie -5 i 5).

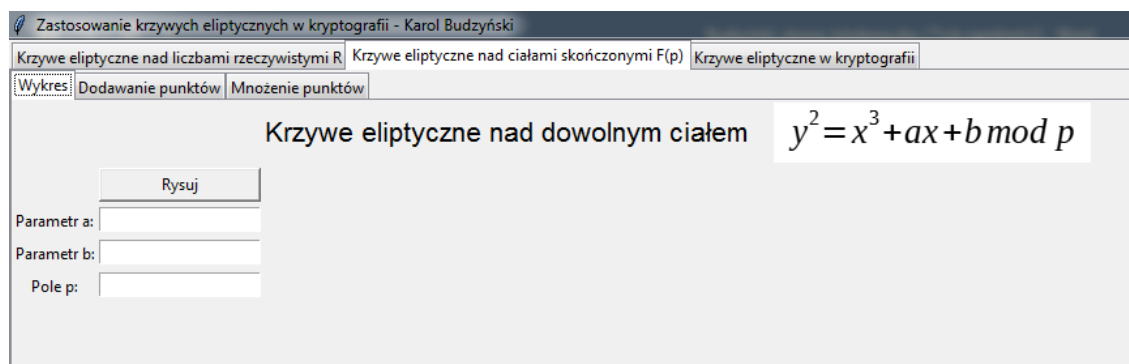
Gdy któryś z parametrów zostanie podany nieprawidłowo, bądź punkt  $P$  nie będzie znajdować się na krzywej eliptycznej, pojawi się informacja z prośbą o podanie prawidłowych danych.



Rys. 4.6. Wynik mnożenia punktów na krzywej eliptycznej nad liczbami rzeczywistymi  
R

### Krzywe eliptyczne nad ciałami skończonymi $F(p)$ - Wykres

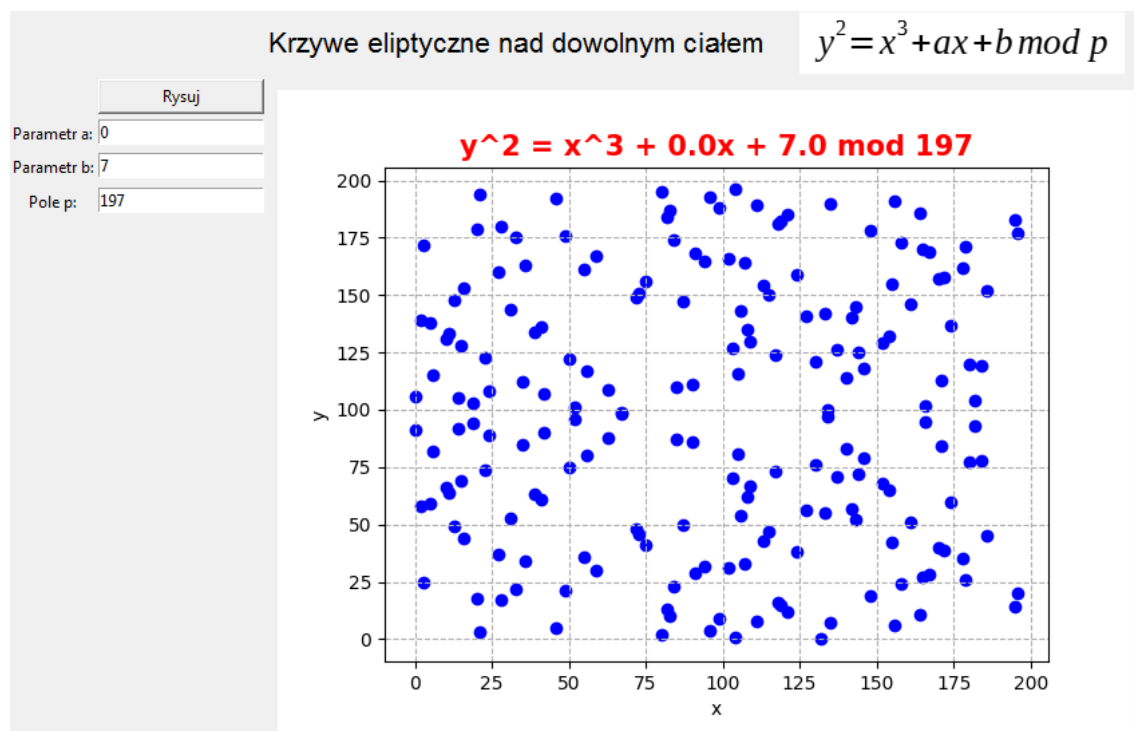
Kolejna pozycja w menu głównym aplikacji to *Krzywe eliptyczne nad ciałami skończonymi  $F(p)$* . Pierwszym pod elementem tej części programu jest *Wykres* (Rys. 4.7).



Rys. 4.7. Krzywe eliptyczne nad ciałami skończonymi  $F(p)$  – Wykres

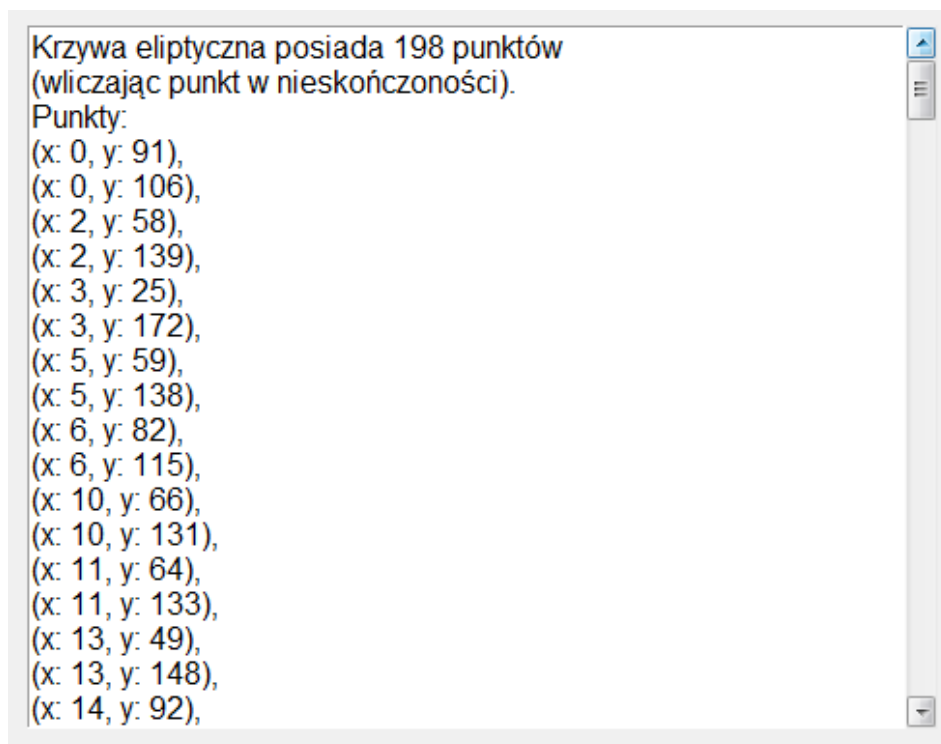
Aby uzyskać wykres krzywej eliptycznej nad ciałem skończonym  $F(p)$ , należy podać parametry krzywej (*Parametr a* i *Parametr b*) oraz pole ciała  $F(p)$  (*Pole p*), które musi być liczbą pierwszą. Po uzupełnieniu wszystkich danych należy wcisnąć przycisk *Rysuj*.

Jeżeli wprowadzone parametry są prawidłowe, program wyznaczy wykres krzywej eliptycznej nad podanym polem (Rys. 4.8). Gdy któryś z parametrów zostanie podany nieprawidłowo, pojawi się informacja z prośbą o podanie prawidłowych parametrów.



Rys. 4.8. Wykres krzywej eliptycznej nad ciałami skończonymi  $F(p)$

Dodatkowo oprócz wykresu krzywej, w polu tekstowym obok wykresu zostaje wypisana ilość punktów wygenerowanych przez krzywą eliptyczną oraz współrzędne każdego z punktów (Rys. 4.9).



Rys. 4.9. Pole tekstowe z wypisanymi punktami należącymi do krzywej eliptycznej

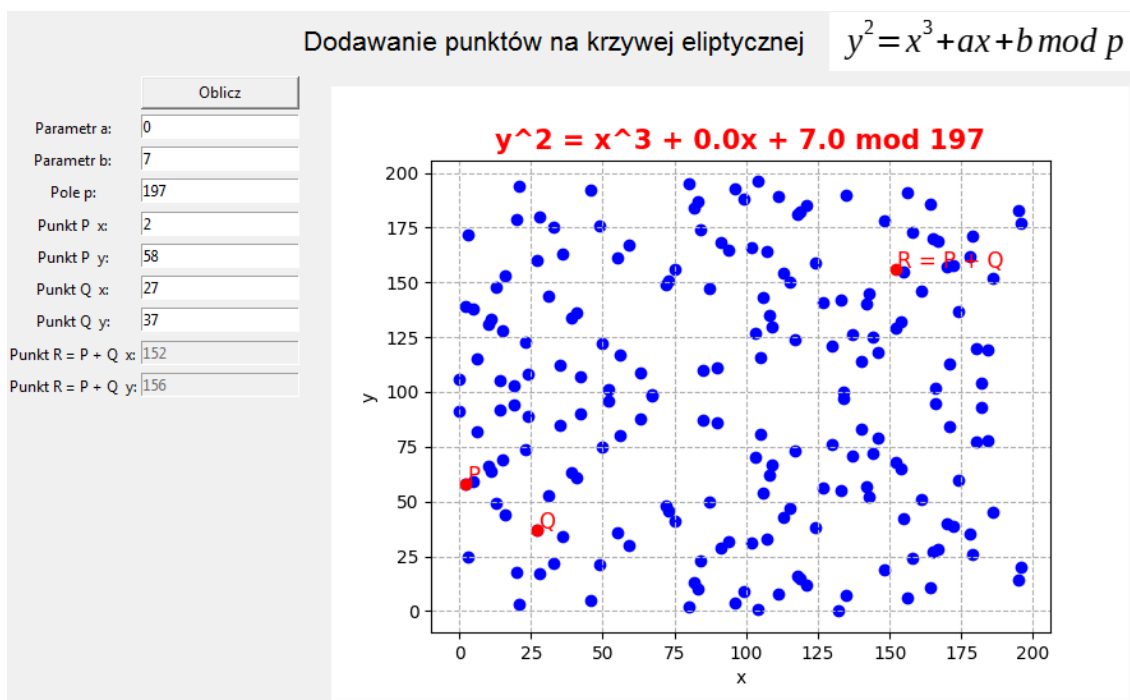
### Krzywe eliptyczne nad ciałami skończonymi $F(p)$ - Dodawanie punktów

W podmenu *Dodawanie punktów*, jak sama nazwa wskazuje, mamy możliwość dodawania punktów znajdujących się na krzywej eliptycznej nad ciałem skończonym  $F(p)$  (Rys. 4.10).

Rys. 4.10. Krzywe eliptyczne nad ciałami skończonymi  $F(p)$  – Dodawanie punktów

Aby uzyskać poprawny wynik, należy podać parametry krzywej eliptycznej (*Parametr a* i *Parametr b*), pole ciała  $F(p)$  (*Pole p*) oraz punkty  $P$  ( $P_x$ ,  $P_y$ ),  $Q$  ( $Q_x$ ,  $Q_y$ ) należące

do krzywej. Parametry i punkty można uzyskać w poprzedniej części programu (Wykres). Gdy wszystkie pola zostały uzupełnione, należy wcisnąć przycisk *Oblicz*. Jeżeli wprowadzone dane są prawidłowe, program wyznaczy wykres krzywej eliptycznej nad podanym polem oraz zaznaczy na nim punkty P, Q oraz R będący sumą P i Q (Rys. 4.11).



Rys. 4.11. Wynik dodawania punktów na krzywej eliptycznej nad ciałami skończonymi  $F(p)$

Oprócz tego, że punkt R zostaje umieszczony na wykresie ( $R = P + Q$ ), jego współrzędne wypisane są w polu tekstowym (*Punkt R = P + Q x* i *Punkt R = P + Q y*). Gdy któryś z parametrów zostanie podany nieprawidłowo lub któryś z punktów P, Q nie należy do krzywej eliptycznej, pojawi się informacja z prośbą o podanie prawidłowych danych.

### Krzywe eliptyczne nad ciałami skończonymi $F(p)$ - Mnożenie punktów

Kolejną pozycją podmenu jest *Mnożenie punktów*. W oknie tym (Rys 4.12), mamy możliwość wyliczenia wielokrotności punktu należącego do krzywej eliptycznej nad ciałem skończonym  $F(p)$ .

Zastosowanie krzywych eliptycznych w kryptografii - Karol Budzyński

Krzywe eliptyczne nad liczbami rzeczywistymi R   Krzywe eliptyczne nad ciałami skończonymi F(p)   Krzywe eliptyczne w kryptografii

Wykres   Dodawanie punktów   Mnożenie punktów

Mnożenie punktów na krzywej eliptycznej  $y^2 = x^3 + ax + b \bmod p$

Oblicz

Parametr a:

Parametr b:

Pole p:

Punkt P x:

Punkt P y:

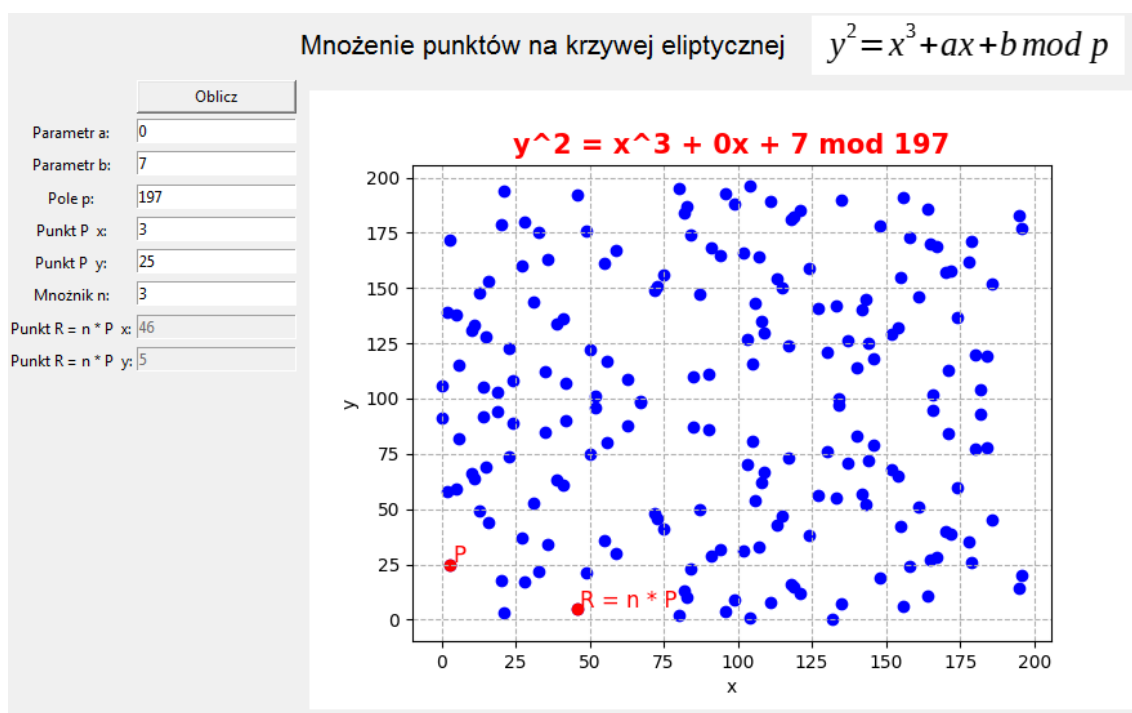
Mnożnik n:

Punkt R = n \* P x:

Punkt R = n \* P y:

Rys. 4.12. Krzywe eliptyczne nad ciałami skończonymi F(p) - Mnożenie punktów

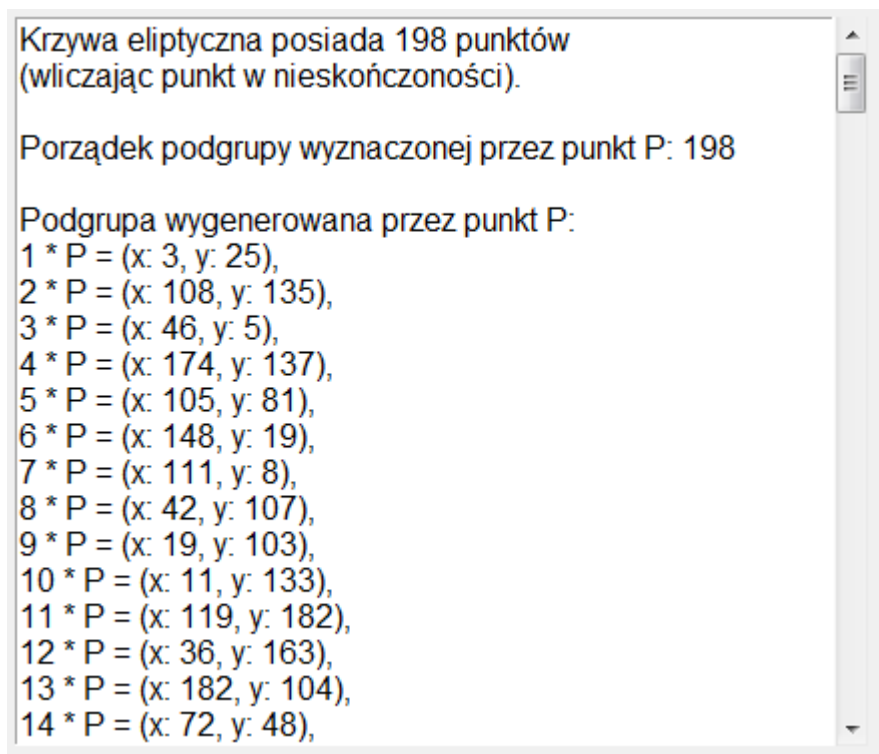
Aby obliczyć wielokrotność punktu, należy podać parametry krzywej eliptycznej (*Parametr a* i *Parametr b*), pole ciała F(p) (*Pole p*), punkt P (*P x*, *P y*) należący do krzywej oraz *Mnożnik n*. Parametry i punkt można uzyskać w poprzedniej części programu (*Wykres*). Gdy wszystkie pola zostały uzupełnione, należy wcisnąć przycisk *Oblicz*. Jeżeli wprowadzone dane są prawidłowe, program wyznaczy wykres krzywej eliptycznej nad podanym polem oraz zaznaczy na nim punkty P oraz R będący wielokrotnością  $n * P$  (Rys. 4.13). Wartość punktu R wypisana jest także w polach tekstowych ( $R = n * P x$  i  $R = n * P y$ ).



Rys. 4.13. Wynik mnożenia punktów na krzywej eliptycznej nad ciałami skończonymi F(p)



Dodatkowo w polu tekstowym obok wykresu, umieszczone są informacje o ilości punktów krzywej eliptycznej, porządku podgrupy wyznaczonej przez podany punkt P oraz wszystkie punkty należące do danej podgrupy (Rys 4.14).

A screenshot of a text field from a software application. The text is as follows:

Krzywa eliptyczna posiada 198 punktów  
(wliczając punkt w nieskończoności).

Porządek podgrupy wyznaczonej przez punkt P: 198

Podgrupa wygenerowana przez punkt P:

- 1 \* P = (x: 3, y: 25),
- 2 \* P = (x: 108, y: 135),
- 3 \* P = (x: 46, y: 5),
- 4 \* P = (x: 174, y: 137),
- 5 \* P = (x: 105, y: 81),
- 6 \* P = (x: 148, y: 19),
- 7 \* P = (x: 111, y: 8),
- 8 \* P = (x: 42, y: 107),
- 9 \* P = (x: 19, y: 103),
- 10 \* P = (x: 11, y: 133),
- 11 \* P = (x: 119, y: 182),
- 12 \* P = (x: 36, y: 163),
- 13 \* P = (x: 182, y: 104),
- 14 \* P = (x: 72, y: 48),

Rys. 4.9. Pole tekstowe z wypisanymi punktami należącymi do podgrupy wygenerowanej przez punkt P

Gdy któryś z parametrów zostanie podany nieprawidłowo lub punkt P nie należy do krzywej eliptycznej, pojawi się informacja z prośbą o podanie prawidłowych danych.

### Krzywe eliptyczne w kryptografii - ElGamal - Punkty

Kolejna pozycja w menu głównym to *Krzywe eliptyczne w kryptografii*. Pierwszą opcją w tej części programu to *ElGamal – Punkty*. W tym miejscu możemy dokonać szyfrowania, bądź deszyfrowania punktu znajdującego się na krzywej eliptycznej za pomocą algorytmu ElGamal z wykorzystaniem krzywych eliptycznych. Okno programu podzielone jest na dwie części. W części górnej mamy możliwość zaszyfrowania punktu, natomiast w części dolnej następuje odszyfrowanie punktu (Rys. 4.13).

Zastosowanie krzywych eliptycznych w kryptografii - Karol Budzyński

Krzywe eliptyczne nad liczbami rzeczywistymi  $\mathbb{R}$  | Krzywe eliptyczne nad ciałami skończonymi  $F(p)$  | Krzywe eliptyczne w kryptografii

ElGamal - Punkty | Podpis cyfrowy ECDSA

### Szyfrowanie ElGamal z wykorzystaniem krzywych eliptycznych

Szyfruj

Parametr a:   
Parametr b:   
Pole p:   
Generator x:   
Generator y:   
Punkt do zaszyfrowania x:   
Punkt do zaszyfrowania y:   
Klucz prywatny:

Deszyfruj

Parametr a:   
Parametr b:   
Pole p:   
Punkt K x:   
Punkt K y:   
Punkt C x:   
Punkt C y:   
Klucz prywatny:

Rys. 4.13. Krzywe eliptyczne w kryptografii - ElGamal - Punkty

Aby przeprowadzić szyfrowanie punktu należy podać parametry krzywej eliptycznej (*Parametr a* i *Parametr b*), pole ciała  $F(p)$  (*Pole p*), punkt Generatora (*Generator x*, *Generator y*), punkt do zaszyfrowania (*Punkt do zaszyfrowania x* i *Punkt do zaszyfrowania y*) oraz *Klucz prywatny*. Punkt Generatora i punkt do zaszyfrowania muszą należeć do krzywej eliptycznej. Dodatkowo punkt Generatora powinien być takim punktem na krzywej, dla którego wytworzona podgrupa jest w miarę największa, tak, aby stosunek ilości punktów na krzywej do ilości punktów znajdujących się w podgrupie był jak najbliższy 1. Punkt taki możemy wybrać w zakładce *Krzywe eliptyczne nad ciałami skończonymi  $F(p)$  - Mnożenie punktów*, gdzie mamy podane podgrupy dla podanego punktu. Jeżeli wszystkie wprowadzone dane są prawidłowe na ekranie ukaże się pole tekstowe, w którym będą umieszczone wszystkie dane na temat szyfrowania takie jak: klucz publiczny, klucz losowy, zaszyfrowana wiadomość (punkty K i C) (Rys. 4.14).

Gdy któryś z parametrów zostanie podany nieprawidłowo lub punkty Generatora i do zaszyfrowania nie należą do krzywej eliptycznej, pojawi się informacja z prośbą o podanie prawidłowych danych.

Szyfruj	
Parametr a:	0
Parametr b:	7
Pole p:	197
Generator x:	3
Generator y:	25
Punkt do szyfrowania x:	15
Punkt do szyfrowania y:	128
Klucz prywatny:	160

Parametry krzywej eliptycznej a: 0, b: 7, p: 197  
Generator: (3, 25)  
Szyfrowany punkt: (15, 128)  
Klucz prywatny: 160  
Klucz publiczny: (67, 99)  
Klucz losowy: 169805655747416681388291014369032782069  
  
Zaszyfrowana wiadomość K: (6, 82), C: (115, 47)

Rys. 4.14. Rezultat szyfrowania punktu ElGamal

Aby przeprowadzić deszyfrowanie punktu należy podać parametry krzywej eliptycznej (*Parametr a* i *Parametr b*), pole ciała  $F(p)$  (*Pole p*), punkt K (*Punkt K x*, *Punkt K y*), punkt C (*Punkt C x*, *Punkt C y*) oraz *Klucz prywatny*. Dane te możemy skopiować z pola, które ukazało się po szyfrowaniu punktu. Jeżeli wszystkie wprowadzone dane są prawidłowe na ekranie ukaże się pole tekstowe, w którym będą umieszczone wszystkie dane na temat deszyfrowania oraz odszyfrowany punkt (Rys. 4.15).

Gdy któryś z parametrów zostanie podany nieprawidłowo lub punkty K i C nie należą do krzywej eliptycznej, pojawi się informacja z prośbą o podanie prawidłowych danych.

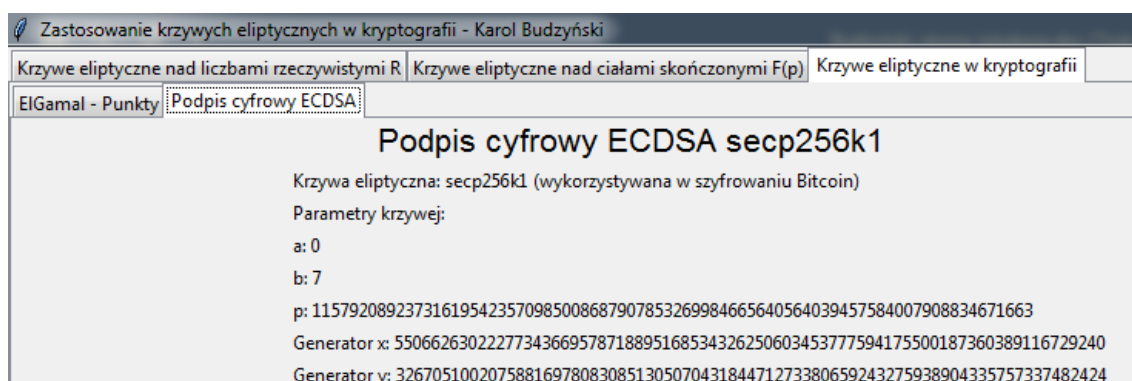
Deszyfruj	
Parametr a:	0
Parametr b:	7
Pole p:	197
Punkt K x:	6
Punkt K y:	82
Punkt C x:	115
Punkt C y:	47
Klucz prywatny:	160

Parametry krzywej eliptycznej a: 0, b: 7, p: 197  
Zaszyfrowana wiadomość K: (6, 82), C: (115, 47)  
Klucz prywatny: 160  
  
Odszyfrowany punkt: (15, 128)

Rys. 4.15. Rezultat deszyfrowania punktu ElGamal

### Krzywe eliptyczne w kryptografii - Podpis cyfrowy ECDSA

Ostatnia część aplikacji to *Podpis cyfrowy ECDSA*. W oknie tym mamy możliwość podpisania wiadomości za pomocą algorytmu podpisu cyfrowego opartego o krzywe eliptyczne oraz sprawdzania czy podpis użyty do podpisania wiadomości jest prawidłowy. Ta część aplikacji jest podzielona na dwie części. W górnej części okna umieszczona została informacja o krzywej eliptycznej wykorzystywanej do składania podpisu. Jest to krzywa Secp256k1, która jest używana w kryptografii klucza publicznego Bitcoin (Rys. 4.16).



Rys. 4.16. Krzywe eliptyczne w kryptografii - Podpis cyfrowy ECDSA - informacja

Natomiast w dolnej części okna znajdują się pola do podpisywania wiadomości oraz weryfikacji podpisu (Rys. 4.17).

Rys. 4.17. Krzywe eliptyczne w kryptografii - Podpis cyfrowy ECDSA – podpisywanie i weryfikacja podpisu

W celu podpisania wiadomości, należy wpisać jej treść w polu *Wiadomość do podpisania* i wcisnąć przycisk *Podpisz*. Następnie poniżej wyskoczy pole tekstowe z danymi takimi jak klucz prywatny, klucz publiczny i podpis (Rys. 4.18). Dane te są potrzebne do weryfikacji podpisu.

Rys. 4.18. Pole tekstowe z wynikiem podpisywania wiadomości

Aby zweryfikować poprawność podpisu należy uzupełnić wszystkie pola:

- Podpisana wiadomość,
- Klucz publiczny x,
- Klucz publiczny y,
- Podpis x,
- Podpis y.

W przypadku uzupełnienia wszystkich pól prawidłowymi wartościami, następuje weryfikacja poprawności podpisywanej wiadomości z podpisem. Jeżeli weryfikacja jest pozytywna zostanie wyświetlony komunikat jak na Rys. 4.19. W przeciwnym przypadku, gdy wiadomość różni się od podpisu na ekranie pojawi się informacja jak na Rys 4.20.

Podpisywana wiadomość
0xc661f8bd12ee3ae8b7717d44ab08c2c679aa60104ce1d866840decfd53eae3d4
0x4a882c3a4e275edd9afc53f2a1999b9e4fac39375f92d9825c7cf17e45f67dac
0x59e56d6782f1a6104d5f33861a2c14e4954d642c47b18c4874f586b811ea7746
0x4bb8dacbd983e1453e5a174ab5dd9328b9f8572795bd86333509193b3cace9e8

Podpis prawidłowy.

Rys. 4.19. Pozytywna weryfikacja podpisu

Inna wiadomość
0xc661f8bd12ee3ae8b7717d44ab08c2c679aa60104ce1d866840decfd53eae3d4
0x4a882c3a4e275edd9afc53f2a1999b9e4fac39375f92d9825c7cf17e45f67dac
0x59e56d6782f1a6104d5f33861a2c14e4954d642c47b18c4874f586b811ea7746
0x4bb8dacbd983e1453e5a174ab5dd9328b9f8572795bd86333509193b3cace9e8

Podpis nieprawidłowy.

Rys. 4.20. Negatywna weryfikacja podpisu

## 5. Podsumowanie i wnioski

Przygotowana przeze mnie praca miała na celu przybliżyć temat krzywych eliptycznych oraz wykorzystania ich w kryptografii. Jest to bardzo szeroki zakres wiedzy, ale myślę iż w sposób prosty, przystępny i zwięzły udało mi się zaprezentować najważniejsze pojęcia związane z krzywymi eliptycznymi oraz ich własności, dzięki którym są one tak chętnie wykorzystywane w kryptografii. Przedstawiłem i opisałem także kryptosystemy oparte o krzywe eliptyczne takie jak: ElGamal, ECDH, ECDSA. W ciągu ostatnich lat, krzywe eliptyczne stały się jednymi z najbardziej efektywnych narzędzi w systemach kryptograficznych. Spowodowane jest to tym, iż pozwalają one na zastosowanie kluczy kryptograficznych o mniejszej długości, zachowując taki sam stopień bezpieczeństwa w porównaniu chociażby do kryptosystemu RSA. Dzięki temu są one dużo szybsze w działaniu i wymagają mniejszej ilości zasobów pamięci. W dzisiejszych czasach jest to bardzo istotne ze względu na ilość operacji wykonywanych w internecie oraz na szybkość komunikacji. Przygotowana przeze mnie aplikacja, udowadnia, że implementacja ECC jest dosyć prosta. Program może służyć do prezentacji zagadnień związanych z krzywymi eliptycznymi i kryptografią krzywych eliptycznych na zajęciach z zakresu kryptografii.

Praca ta może być podstawą do dalszego zagłębiania się w temacie wykorzystania krzywych eliptycznych. W pracy przedstawione zostały krzywe eliptyczne Weierstrassa nad polami pierwszymi, ale istnieją także inne rodzaje pól i krzywych, między innymi: krzywe Koblitz nad polami binarnymi, krzywe binarne, krzywe Edwardsa.

## Literatura

1. M. Węgrzyn, J. Jabłoński, M. Nowakowski, *Transakcje i monety internetowe. Kryptologia a biznes – bezpieczeństwo stosowane*, Wydawnictwo BTC, Legionowow 2014.
2. S. W. Bray, *Algorytmy kryptograficzne w Pythonie Wprowadzenie*, Wydawnictwo Helion, Gliwice 2021.
3. Daniel Volya ECC  
<https://volya.xyz/ecc>, stan z dnia 09.09.2022.
4. S. Nakov, *Practical cryptography for developers*, Sofia, November 2018  
<https://cryptobook.nakov.com/>, stan z dnia 09.09.2022.
5. I. Blake, G. Seroussi, N. Smart, *Krzywe eliptyczne w kryptografii*, Wydawnictwo Naukowo Techniczne, Warszawa 2004.
6. Python Tutorial  
<https://www.pythontutorial.net/tkinter/>, stan z dnia 09.09.2022.
7. I. Blake, G. Seroussi, N. Smart, *Krzywe eliptyczne w kryptografii*, Wydawnictwo Naukowo Techniczne, Warszawa 2004.
8. L. C. Washington, *Elliptic Curves: Number Theory and Cryptography, Second Edition*, Taylor & Francis Ltd, Edycja 2, 2008.
9. M. Rosing, *Implementing Elliptic Curve Cryptography*, Manning Puplications, 1998.
10. S. Ling, H. Wang, C. Xing, *Algebraic Curves in Cryptography*, Chapmann & Hall, 2019.

## Spis rysunków

<b>Rys. 2.1.1.</b>	Różne kształty prawidłowych krzywych eliptycznych nad ciałem rzeczywistym $R$ .....	9
<b>Rys. 2.1.2.</b>	Nieprawidłowe krzywe eliptyczne nad ciałem rzeczywistym $R$ .....	10
<b>Rys. 2.2.1.1.</b>	Dodawanie geometryczne punktów na krzywej eliptycznej nad ciałem rzeczywistym $R$ .....	12
<b>Rys. 2.3.1.</b>	Prawidłowe krzywe eliptyczne nad ciałem skończonym $F_p$ .....	15
<b>Rys. 2.3.2.</b>	Nieprawidłowa krzywa eliptyczna nad ciałem skończonym $F_p$ .....	15
<b>Rys. 2.3.1.1.</b>	Dodawanie geometryczne punktów na krzywej eliptycznej nad dowolnym ciałem $F_p$ .....	16
<b>Rys. 2.3.2.1.</b>	Cykliczność wielokrotności punktu krzywej eliptycznej nad ciałem skończonym $F_p$ .....	18
<b>Rys. 3.2.1.</b>	Schemat wymiany kluczy Diffiego-Hellmanna .....	25
<b>Rys. 3.2.1.</b>	Schemat składania i weryfikacji podpisu cyfrowego .....	26
<b>Rys. 3.4.1.</b>	Porównanie długości kluczy RSA i ECC .....	29
<b>Rys. 4.1.</b>	Krzywe eliptyczne nad liczbami rzeczywistymi – Wykres .....	31
<b>Rys. 4.2.</b>	Wykres krzywej eliptycznej nad liczbami rzeczywistymi $R$ .....	32
<b>Rys. 4.3.</b>	Krzywe eliptyczne nad liczbami rzeczywistymi – Dodawanie punktów .....	32
<b>Rys. 4.4.</b>	Wynik dodawania punktów na krzywej eliptycznej nad liczbami rzeczywistymi $R$ .....	33
<b>Rys. 4.5.</b>	Krzywe eliptyczne nad liczbami rzeczywistymi – Mnożenie punktów .....	34
<b>Rys. 4.6.</b>	Wynik mnożenia punktów na krzywej eliptycznej nad liczbami rzeczywistymi $R$ .....	35
<b>Rys. 4.7.</b>	Krzywe eliptyczne nad ciałami skończonymi $F(p)$ – Wykres .....	35
<b>Rys. 4.8.</b>	Wykres krzywej eliptycznej nad ciałami skończonymi $F(p)$ .....	36
<b>Rys. 4.9.</b>	Pole tekstowe z wypisanymi punktami należącymi do krzywej eliptycznej .....	37
<b>Rys. 4.10.</b>	Krzywe eliptyczne nad ciałami skończonymi $F(p)$ – Dodawanie punktów .....	37



<b>Rys. 4.11.</b>	Wynik dodawania punktów na krzywej eliptycznej nad ciałami skończonymi $F(p)$ .....	38
<b>Rys. 4.12.</b>	Krzywe eliptyczne nad ciałami skończonymi $F(p)$ - Mnożenie punktów .....	39
<b>Rys. 4.13.</b>	Wynik mnożenia punktów na krzywej eliptycznej nad ciałami skończonymi $F(p)$ .....	39
<b>Rys. 4.9.</b>	Pole tekstowe z wypisanymi punktami należącymi podgrupy wygenerowanej przez punkt $P$ .....	40
<b>Rys. 4.13.</b>	Krzywe eliptyczne w kryptografii - ElGamal – Punkty .....	41
<b>Rys. 4.14.</b>	Rezultat szyfrowania punktu ElGamal .....	42
<b>Rys. 4.15.</b>	Rezultat deszyfrowania punktu ElGamal .....	42
<b>Rys. 4.16.</b>	Krzywe eliptyczne w kryptografii - Podpis cyfrowy ECDSA – informacja .....	43
<b>Rys. 4.17.</b>	Krzywe eliptyczne w kryptografii - Podpis cyfrowy ECDSA – podpisywanie i weryfikacja podpisu .....	43
<b>Rys. 4.18.</b>	Pole tekstowe z wynikiem podpisywania wiadomości .....	43
<b>Rys. 4.19.</b>	Pozytywna weryfikacja podpisu .....	44
<b>Rys. 4.20.</b>	Negatywna weryfikacja podpisu .....	44