**Шифрование с использованием эллиптических кривых**

**I) Эллиптическая криптография** — раздел криптографии, который изучает асимметричные криптосистемы, основанные на эллиптических кривых над конечными полями. Основное преимущество эллиптической криптографии заключается в том, что на сегодняшний день неизвестны субэкспоненциальные алгоритмы дискретного логарифмирования.

Использование эллиптических кривых для создания криптосистем было независимо друг от друга предложено Нилом Коблицем и Виктором Миллером в 1985 году.

**II) Эллиптические кривые над конечными полями**

Эллиптическая кривая — это математическое уравнение вида:

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

где a и b — коэффициенты, определяющие форму кривой. Уравнение должно удовлетворять условиям:

**4a^3 + 27b^2 != 0**

чтобы кривая не имела особенностей (разрывов или самопересечений).

Кривая рассматривается над конечным полем (например, Fp\mathbb{F}\_pFp​, где ppp — простое число). Это означает, что координаты точек xxx и yyy берутся из конечного множества чисел.

На этой кривой определяются операции сложения точек, которые играют ключевую роль в криптографии.

### III)Как работает ECC?

ECC основывается на математических задачах, которые трудно решить:

1. **Сложение точек на кривой:** Если даны две точки P и Q на кривой, их сумма R=P+Q тоже будет точкой на кривой.
2. **Умножение точки на число:** kP (где k — целое число) означает сложение точки P с собой k раз.

Обратная задача — найти k, зная P и kP, — называется **дискретным логарифмом на эллиптической кривой**. Эта задача является основой безопасности ECC, так как её трудно решить даже для мощных компьютеров.

**IV) Алгоритм шифрования**

#### **Шаг 1. Подготовка:**

1. **Сообщение**:  
   Имеется сообщение, например, "hello, world!".
2. **Преобразование в числа**:  
   Каждую букву переводим в ASCII-код (или другим методом).  
   Результат: массив чисел, например, M=[104,101,108,… ]
3. **Сопоставление с точками на кривой**:  
   Для каждого числа из M находим точку Pm на эллиптической кривой.  
   Результат: массив точек Pm=[Pm1,Pm2,… ]

#### **Шаг 2. Генерация ключей:**

1. **Получатель**:
   * Генерирует закрытый ключ d (случайное число).
   * Вычисляет открытый ключ Q=d⋅G — генераторная точка кривой.
2. **Открытый ключ Q** передаётся отправителю.

#### **Шаг 3. Шифрование (отправитель):**

Для каждой точки Pm​ из массива:

1. Выбираем случайное k (одноразовый ключ).
2. Вычисляем:
   * C1=k⋅G (первый компонент шифротекста).
   * C2=Pm+k⋅Q (второй компонент шифротекста).
3. Результат для одной точки: шифротекст (C1,C2)

Для всего сообщения формируется массив шифротекстов:

C=[(C11,C12),(C21,C22),… ]

#### **Шаг 4. Передача:**

* Отправитель передаёт получателю массив шифротекстов C.

#### **Шаг 5. Расшифрование (получатель):**

Для каждой пары (C1,C2)

1. Вычисляем k⋅Q как d⋅C
2. Восстанавливаем исходную точку Pm: Pm=C2−d⋅C1.

#### **Шаг 6. Обратное преобразование:**

1. Преобразуем каждую точку Pm​ обратно в число M.
2. Переводим числа обратно в символы (например, через ASCII).  
   Результат: исходное сообщение "hello, world!".

### ****Как делается сопоставление с точками на кривой?****

Эллиптическая кривая E состоит из множества точек (x,y), которые удовлетворяют уравнению кривой:

y^2≡x^3+ax+b(mod p).

Для шифрования каждое число M из сообщения нужно преобразовать в точку Pm=(x,y) на этой кривой.

#### Пошаговый процесс:

1. **Определяем x=M**
   * Прямо используем число M как координату x на кривой.
2. **Находим y:**
   * Подставляем x в уравнение кривой и ищем y: y^2≡x^3+ax+b(mod p)
   * Это решается подбором y (обычно перебор по модулю p).
3. **Проверка точки:**
   * Если для данного x нет y, пробуем x+1,x+2…, пока не найдем точку на кривой.

#### Пример:

Для кривой y^2≡x^3+2x+3(mod7) и M=104

* Возьмём x=104 mod  7=6 (привели x в пределы поля).
* Подставим в уравнение:  
  y2≡63+2⋅6+3(mod7)≡216+12+3≡231mod  7≡0
* Решаем y2≡0 mod  7  
  y=0

Таким образом, Pm = (6,0)

### ****Пример процесса шифрования на эллиптических кривых (Эль-Гамаль):****

### 

### ****I) Подготовка****

### 

### 

### ****II) Генерация ключей****

### 

### ****III) Шифрование****

### 

### 

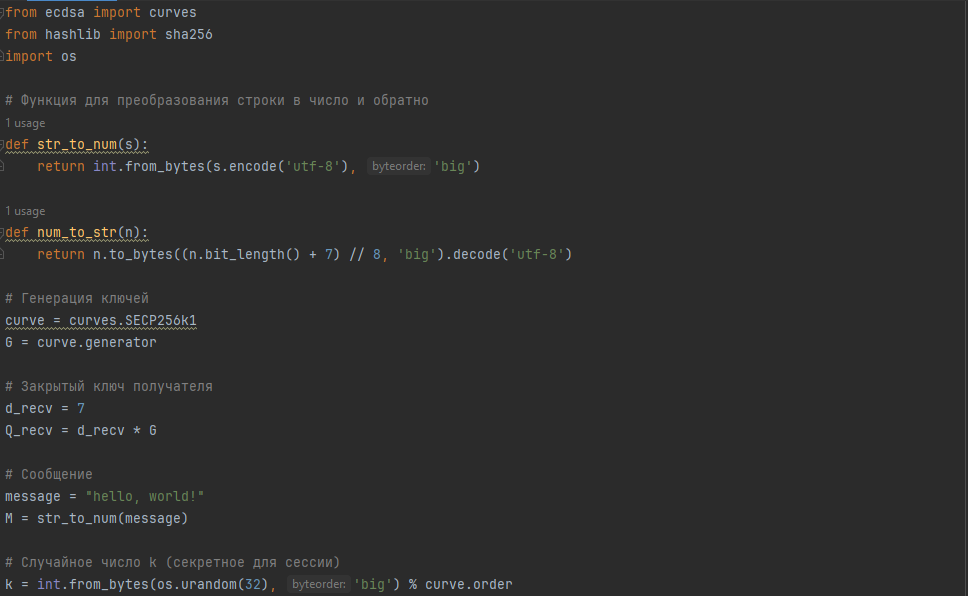
### ****IV) Расшифрование****

### 

### ****V) Обратное преобразование****

### 

Реализация кода на python





Результат:

