Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный университет   
информатики и радиоэлектроники»

Факультет компьютерных систем и сетей

Кафедра информатики

Дисциплина: Методы защиты информации

ОТЧЁТ

к лабораторной работе №7

на тему

**КРИПТОГРАФИЯ С ИСПОЛЬЗОВАНИЕМ ЭЛЛИПТИЧЕСКИХ КРИВЫХ**

Выполнил: студент гр.253505 Ющук И.А.

Проверил: ассистент кафедры информатики Герчик А.В.

Минск 2025

**СОДЕРЖАНИЕ**

[Введение 3](#_Toc207721845)

[1 Ход работы 4](#_Toc207721846)

[Заключение 5](#_Toc207721847)

[Приложение А (обязательное) Листинг программного кода 6](#_Toc207721848)

# ВВЕДЕНИЕ

Современный этап развития информационных технологий характеризуется повсеместным использованием криптографических методов для защиты данных. Одним из наиболее перспективных направлений в этой области является криптография на эллиптических кривых (Elliptic Curve Cryptography, ECC), которая предлагает высокий уровень безопасности при относительно небольших вычислительных затратах. Это особенно важно в условиях растущих требований к производительности и энергоэффективности вычислительных систем.

Математический аппарат эллиптических кривых основан на использовании алгебраических структур специального типа. Криптосистемы на эллиптических кривых опираются на сложность решения задачи дискретного логарифмирования в группе точек кривой. Важным преимуществом ECC является возможность использования ключей значительно меньшей длины по сравнению с традиционными криптосистемами при обеспечении сопоставимого уровня безопасности. Это делает данную технологию особенно attractive для применений в мобильных устройствах и системах с ограниченными ресурсами.

В Российской Федерации стандартом в области криптографии на эллиптических кривых является ГОСТ Р 34.10-2012. Этот стандарт определяет алгоритмы генерации ключей, создания и проверки электронных подписей с использованием математического аппарата эллиптических кривых. Стандарт предусматривает использование определенных параметров эллиптических кривых, рекомендованных для обеспечения необходимого уровня криптостойкости.

Изучение криптографии на эллиптических кривых представляет значительный теоретический и практический интерес. Понимание математических основ ECC позволяет специалистам в области информационной безопасности более эффективно проектировать и реализовывать защищенные системы. Практическое освоение алгоритмов работы с эллиптическими кривыми способствует deeper пониманию современных криптографических методов и принципов их применения.

Реализация базовых операций с точками эллиптических кривых, таких как сложение точек и умножение точки на скаляр, составляет основу для понимания работы криптографических алгоритмов этого класса. Эти операции являются фундаментальными building blocks для более сложных криптографических протоколов и механизмов защиты информации.

Исследование особенностей отечественного стандарта ГОСТ Р 34.10-2012 позволяет получить представление о требованиях, предъявляемых к криптографическим алгоритмам в Российской Федерации. Анализ параметров кривых, используемых в стандарте, и алгоритмов работы с ними способствует формированию комплексного понимания современных подходов к обеспечению информационной безопасности.

# **1 ХОД РАБОТЫ**

В рамках выполнения лабораторной работы была реализована система шифрования на эллиптических кривых на языке Python.

На первом этапе проведено изучение математических основ и принципов работы с эллиптическими кривыми. Особое внимание уделено изучению теории групп точек эллиптической кривой, операций сложения и удвоения точек, а также алгоритма умножения точки на скаляр. Исследованы механизмы генерации ключевых пар и принципы использования открытого и закрытого ключей для шифрования и дешифрования данных.

После освоения теоретического материала приступили к практической реализации. Были написаны функции для задания параметров эллиптической кривой, генерации базовой точки и создания ключевой пары. Реализованы алгоритмы шифрования данных с использованием открытого ключа получателя и дешифрования с применением закрытого ключа.

Для проверки корректности работы алгоритма проведено тестирование на различных тестовых векторах. Реализована функция визуализации параметров кривой и выполнения промежуточных вычислений в процессе шифрования. Проверена возможность успешного восстановления исходного сообщения после процедур шифрования и дешифрования.

Исходные данные представлены на рисунке 1.1.



Рисунок 1.1 – Входные данные

Программа была в результате появился текст. Процесс работы программы представлен на рисунке 1.2.

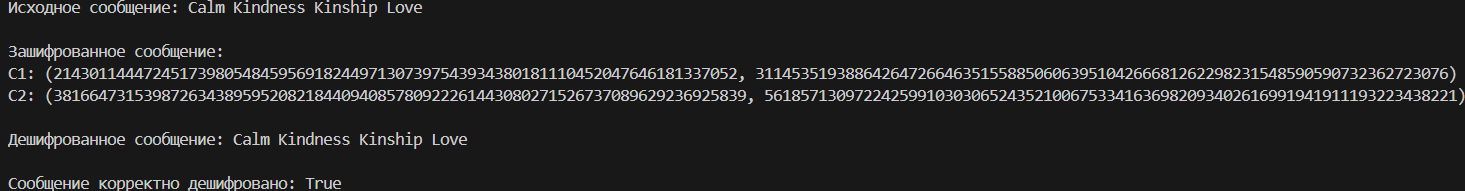


Рисунок 1.2 – Результат работы программы

# ЗАКЛЮЧЕНИЕ

В результате выполнения лабораторной работы была успешно реализована система шифрования на основе методов эллиптической криптографии. Практическая реализация охватила все ключевые аспекты работы с эллиптическими кривыми: генерацию параметров кривой, создание ключевой пары, алгоритмы шифрования данных с использованием открытого ключа и процедуру дешифрования с применением закрытого ключа. Основным достижением работы стало создание работоспособного программного средства, наглядно демонстрирующего принципы построения криптографических систем на эллиптических кривых.

Реализация подтвердила теоретические положения эллиптической криптографии, в частности, важность корректного выполнения математических операций над точками кривой и строгого соблюдения протоколов шифрования. Особое значение имела точная реализация алгоритма умножения точки на скаляр и операций сложения точек эллиптической кривой. Разработанная система продемонстрировала способность надежно шифровать данные и обеспечивать их конфиденциальность при передаче.

Практическая ценность работы заключается в создании инструмента для изучения принципов работы современных криптографических систем. Реализованная система может использоваться для демонстрации механизмов обеспечения конфиденциальности данных, изучения математических основ эллиптической криптографии, а также для понимания особенностей построения защищенных каналов связи. В процессе реализации были приобретены навыки работы с большими числами, освоены методы криптографических преобразований и отработаны приемы тестирования систем шифрования.

Перспективы развития работы включают оптимизацию алгоритмов для работы с большими объемами данных, реализацию поддержки различных форматов ключей и сертификатов, а также исследование стойкости реализованного алгоритма к различным типам криптоатак. Дальнейшее развитие может включать интеграцию с системами защищенной коммуникации и реализацию дополнительных криптографических протоколов.

Таким образом, реализованная система шифрования на эллиптических кривых представляет собой законченное решение, которое демонстрирует принципы построения современных криптографических средств защиты информации и служит основой для дальнейшего изучения методов обеспечения конфиденциальности данных.

# ПРИЛОЖЕНИЕ А (обязательное) Листинг программного кода

import random

# y^2 = (x^3 + ax + b) mod p

p = 0xffffffff00000001000000000000000000000000ffffffffffffffffffffffff

a = -3

b = 0x5ac635d8aa3a93e7b3ebbd55769886bc651d06b0cc53b0f63bce3c3e27d2604b

G = (0x6b17d1f2e12c4247f8bce6e563a440f277037d812deb33a0f4a13945d898c296,

     0x4fe342e2fe1a7f9b8ee7eb4a7c0f9e162bce33576b315ececbb6406837bf51f5)

n = 0xffffffff00000000ffffffffffffffffbce6faada7179e84f3b9cac2fc632551

# Формулы получаются из подстановки уравнения касательной/секущей и далее по Виета находится новая точка

def point\_add(P, Q):

    if P is None:

        return Q

    if Q is None:

        return P

    x1, y1 = P

    x2, y2 = Q

    if x1 == x2 and y1 != y2:

        return None

    if x1 == x2:

        m = (3 \* x1 \* x1 + a) \* pow(2 \* y1, p-2, p) % p

    else:

        m = (y2 - y1) \* pow(x2 - x1, p-2, p) % p

    x3 = (m \* m - x1 - x2) % p

    y3 = (m \* (x1 - x3) - y1) % p

    return (x3, y3)

# В to\_add всегда кратно 2 Point. Далее в соответствии с битами добавляем в результат и получаем kPoint

def point\_mult(k, point):

    result = None

    to\_add = point

    while k:

        if k & 1:

            result = point\_add(result, to\_add)

        to\_add = point\_add(to\_add, to\_add)

        k >>= 1

    return result

# C1 = k×G

# C2 = M + k×Q, Q - открытый ключ

def encrypt\_message(message, public\_key):