Министерство образования Республики Беларусь
Учреждение образования «Белорусский государственный университет информатики и радиоэлектроники»

Факультет компьютерных систем и сетей
Кафедра информатики
Дисциплина «Методы защиты информации»

ОТЧЕТ
к лабораторной работе
на тему:
«Криптография с использованием эллиптических кривых»

Выполнил:
Слуцкий Никита Сергеевич,
студент группы 053505
Проверил:
Лещенко Евгений Александрович,
ассистент каф. Информатики

Минск 2023

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

В рамках данной лабораторной работы будут изучены основные принципы работы алгоритма Эль-Гамаля на эллиптических кривых, а также реализованы соответствующие процедуры для шифрования и дешифрования данных. Такой аналог алгоритма Эль-Гамаля на основе эллиптических кривых позволит оценить эффективность и надежность данной криптографической системы. Также оформить отчёт в соответствии со стандартом предприятия БГУИР.

1 КРАТКИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Схема Эль-Гамаля (Elgamal) – криптосистема с открытым ключом, основанная на трудности вычисления дискретных логарифмов в конечном поле. Криптосистема включает в себя алгоритм шифрования и алгоритм цифровой подписи. Схема Эль-Гамаля лежит в основе бывших стандартов электронной цифровой подписи.

Схема генерации ключей следующая:

– Выбирается E (a,b) с эллиптической кривой в GF(p) или GF(2n);

– Выбирается точку на кривой, e1 (x1, y1);

– Выбирается целое число d.

– Вычисляется e2(x2, y2);

– Объявляется E (a,b), e1 (x1, y1) и e2 (x2, y2) как свой открытый ключ доступа и он сохраняет d как секретный ключ.

На рисунке 1 представлена схема шифрования.

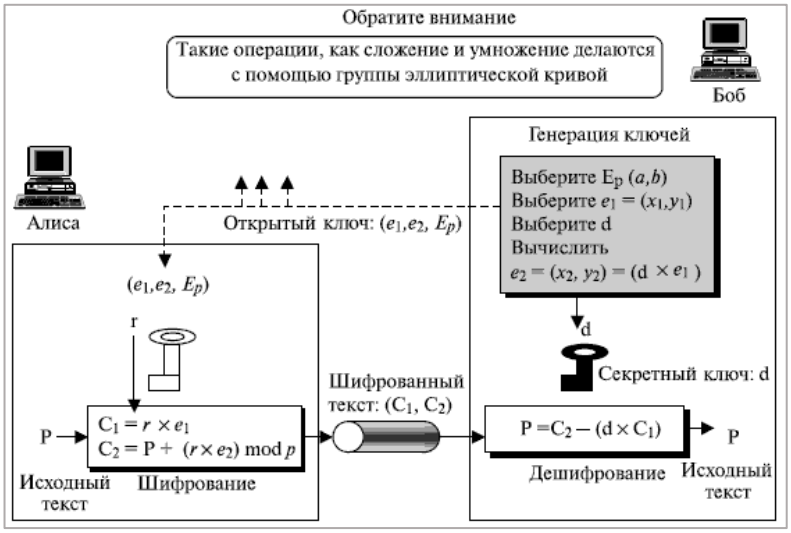

Рисунок 1 – Схема шифрования

Избирательное сравнение алгоритма с эллиптической прямой и без неё представлено ниже:

Ниже приводится краткое сравнение алгоритма Эль-Гамаля с его вариантом,

– секретный ключ в каждом алгоритме – целое число;

– секретные числа, выбираемые в каждом алгоритме, – целые числа;

– возведение в степень в алгоритме Эль-Гамаля заменено умножением точки на константу;

– вычисление обычно легче в эллиптической кривой, потому что умножение проще, чем возведение в степень.

2 ХОД ВЫПОЛНЕНИЯ РАБОТЫ

В рамках лабораторной работы был создан проект на языке программирования Python. Ввиду требований работы с крайней большими числами целесообразным для дальнейшего развития оказался вариант на именно упомянутом языке программирования, потому что он поддерживает работу с длинной арифметикой из коробки.

ЗАКЛЮЧЕНИЕ

В ходе выполнения лабораторной работы была реализована схема шифрования и дешифрования на основе эллиптических кривых, аналогичная алгоритму Эль-Гамаля. Работа с эллиптическими кривыми позволяет повысить уровень безопасности криптографических операций, а также улучшить эффективность передачи и защиту данных. В ходе лабораторной работы было реализовано программное средство получающее, при дешифрации, лишь точку на эллиптической кривой, для получения исходного сообщения необходимо решить задачу ECDLP, что является довольно сложной задачей.

В заключение, лабораторная работа по реализации схемы шифрования на основе эллиптических кривых, подобной алгоритму Эль-Гамаля, позволила понять принципы работы этой криптографической системы и оценить ее эффективность. Эллиптические кривые продолжают оставаться актуальным и перспективным инструментом в области информационной безопасности, и их применение может быть ключевым для обеспечения конфиденциальности данных в современном цифровом мире.

ПРИЛОЖЕНИЕ А

(Листинг кода)

```
from data import CURVE_A, CURVE_B, ENCODING_RANDOM_K, PRIVATE_KEY, SENDER_RANDOM, prime_number
from ElGamal import ElGamal

def main():
    elGamalClass = ElGamal(
        CURVE_A,
        CURVE_B,
        prime_number,
        ENCODING_RANDOM_K,
        SENDER_RANDOM,
        PRIVATE_KEY
    )

    with open('input.txt', 'r', encoding='utf-8') as file:
        plain_message = file.read()

    print('Source' + str(plain_message))

    cipherpoints = elGamalClass.encrypt(plain_message)

    plaintext = elGamalClass.decrypt(cipherpoints)

    print('Retrieved message is : ' + plaintext)

    with open('encrypted.txt', 'w', encoding='utf-8') as file:
        file.write(str([str(item[1]) for item in cipherpoints]))

    with open('decrypted.txt', 'w', encoding='utf-8') as file:
        file.write(plaintext)


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