Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет информационных технологий

Кафедра информационных систем и технологий

Специальность 1-40 01 01 Программное обеспечение информационных технологий

**Отчёт по лабораторной работе №9**

**«Исследование ассиметричных шифров»**

**Вариант 3**

Исполнитель:

Студент 3 курса группы 4

Гурина К. С.

Руководитель:

Ассистент Сазонова Д. В.

1. **Цель и задачи работы**

**Цель:** изучение и приобретение практических навыков разработки и использования приложений для реализации ассиметричных шифров.

**Задачи:**

1. Закрепить теоретические знания по алгебраическому описанию, алгоритмам реализации операций зашифрования/расшифрования и оценке криптостойкости ассиметричных шифров.

2. Разработать приложение для реализации указанных преподавателем методов генерации ключевой информации и ее использования для ассиметричного зашифрования/расшифрования.

3. Выполнить анализ криптостойкости ассиметричных шифров.

4. Оценить скорость зашифрования/расшифрования реализованных шифров.

5. Результаты выполнения лабораторной работы оформить в виде описания разработанного приложения, методики выполнения экспериментов с использованием приложения и результатов эксперимента

**2. Теоретические сведения**

В основу асимметричной криптографии положена идея использовать ключи парами: один – для зашифрования (открытый, или публичный, ключ), другой – для расшифрования (тайный ключ). Отметим, что указанная пара ключей принадлежит получателю зашифрованного сообщения. Все алгоритмы шифрования с открытым ключом основаны на использовании односторонних функций, к числу которых, как известно, относится вычисление дискретного логарифма.

Односторонней функцией (one-way function) называется математическая функция, которую относительно легко вычислить, но трудно найти по значению функции соответствующее значение аргумента, т. е. зная х, легко вычислить f(x), но по известному f(x) трудно найти подходящее значение x.

Алгоритмы шифрования с открытым ключом можно использовать для решения следующих задач:

* зашифрования/расшифрования передаваемых и хранимых данных в целях их защиты от несанкционированного доступа;
* формирования цифровой подписи под электронными документами;
* распределения секретных ключей, используемых далее при шифровании документов симметричными методами.

Ранцевый (рюкзачный) вектор S = (s1, ..., sz) – это упорядоченный набор из z, z ≥ 3, различных натуральных чисел si. Входом задачи о ранце (рюкзаке) называем пару (S, S), где S – рюкзачный вектор, а S – натуральное число. Решением для входа (S, S) будет такое подмножество из S, сумма элементов которого равняется S.

Суть метода для шифрования состоит в том, что существуют две различные задачи укладки ранца: одна из них решается легко и характеризуется линейным ростом трудоемкости, а другая решается трудно. Легкий для укладки ранец можно трансформировать в трудный. Трудный для укладки ранец применяется в качестве открытого ключа, который легко использовать для зашифрования, но невозможно – для расшифрования. В качестве закрытого ключа применяется легкий для укладки ранец, который предоставляет простой способ расшифрования сообщения.

В качестве закрытого ключа d (легкого для укладки ранца) используется сверхвозрастающая последовательность, состоящая из z элементов: d1, d2, …, dz: d = {di}, i = 1, …, z.

Сверхвозрастающей называется последовательность, в которой каждый последующий член больше суммы всех предыдущих.

Ранцевые криптосистемы не являются криптостойкими. А. Шамир и Р. Циппел обнаружили, что зная числа а, a-1 и n, можно восстановить сверхвозрастающую последовательность по нормальной последовательности.

**3. Ход работы**

**Практическое задание:**

1. Разработать авторское оконное приложение в соответствии с целью лабораторной работы. При этом можно воспользоваться доступными библиотеками либо программными кодами.

В основе вычислений – кодировочные таблицы Base64 и ASCII. Приложение должно реализовывать следующие операции:

• генерация сверхвозрастающей последовательности (тайного ключа); старший член последовательности – 100-битное число; в простейшем случае принимается z = 6 (для кодировки Base64) и z = 8 (для кодировки ASCII);

• вычисление нормальной последовательности (открытого ключа);

• зашифрование сообщения, состоящего из собственных фамилии, имени и отчества;

• расшифрование сообщения;

• оценка времени выполнения операций зашифрования и расшифрования.

2. Проанализировать время выполнения операций зашифрования/расшифрования при увеличении числа членов ключевой последовательности; при использовании разных таблиц кодировки.

3. Результаты оформить в виде отчета по установленным правилам

**Ход работы**

В начале, нужно было разработать приложение, реализующее генерацию приватного и публичного ключа, а также осуществляющее зашифрование и расшифрование сообщения, состоящего из собственных фамилии, имени и отчества.

Генерация ключей включает в себя следующие этапы:

1. Генерация супервозрастающей последовательности. Создается последовательность чисел, где каждый следующий элемент больше суммы всех предыдущих. Код функции представлен на рисунке 3.1.

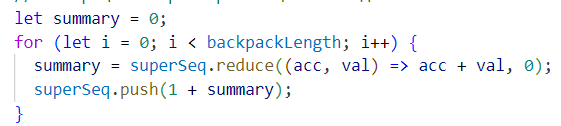


Рисунок 3.1 – Генерация сверхвозрастающей последовательности

2. Выбор значений n и a: n выбирается как число больше суммы всех элементов супервозрастающей последовательности, a выбирается таким, чтобы быть взаимно простым с n (НОД(a, n) = 1).

3. Генерация публичного ключа. Каждый элемент супервозрастающей последовательности умножается на **a** и берется по модулю **n**, чтобы получить элементы публичного ключа. Код функции представлен на рисунке 3.2.

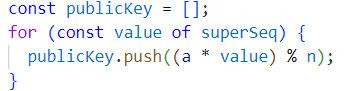


Рисунок 3.2 – Генерация публичного ключа

Для шифрование сообщения оно преобразуется в двоичный формат. Каждый бит двоичного представления умножается на соответствующий элемент публичного ключа и результаты суммируются для получения зашифрованного значения. Код функции представлен на рисунке 3.3.



Рисунок 3.3 – Шифрование сообщения

Код для расшифровки сообщение представлен на рисунке 3.4

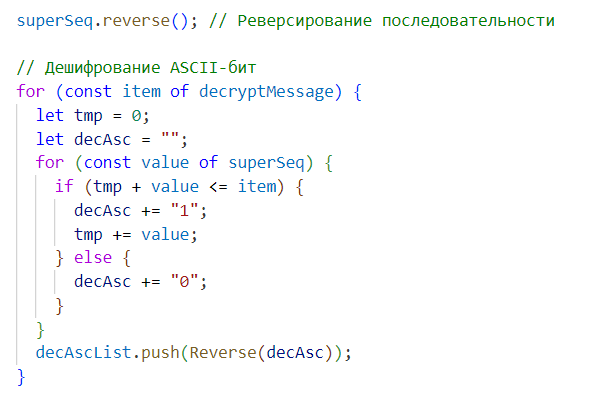


Рисунок 3.4 – Дешифрование сообщения

Результат работы приложения представлен на рисунке 3.5.

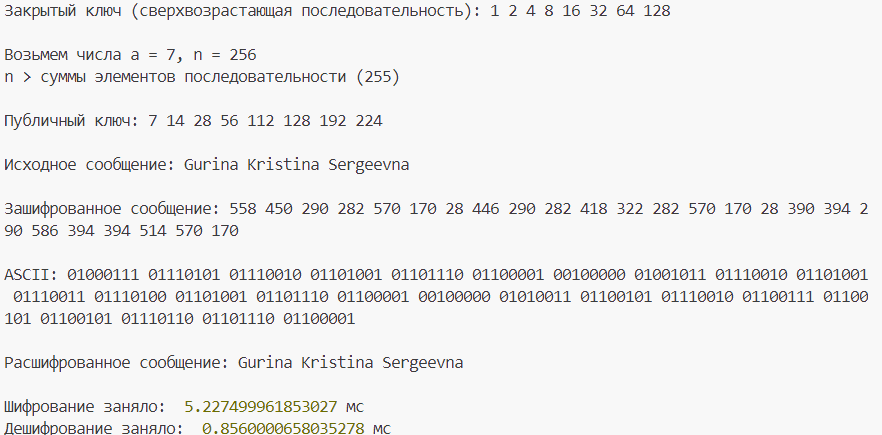


Рисунок 3.5 – Результат работы приложения

**Вывод**

В ходе выполнения лабораторной работы был изучен принцип реализации асимметричных шифров. Также было разработано приложение, выполняющее шифрование и расшифрование с помощью шифра, основанного на алгоритме об укладке ранца. Была оценена скорость шифрования и дешифрования сообщений, представленных в кодах ASCII.