Учреждение образования «БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ  
ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет информационных технологий

[Кафедра информационных](https://www.belstu.by/fakultety/fit/vm) систем и технологий

Специальность 1-40 01 01 Программное обеспечение информационных технологий

**Отчёт по лабораторной работе №9**

по дисциплине Информационная безопасность

Тема: Исследование ассиметричных шифров

        Исполнитель:

Студент 3 курса группы 6

Руководитель:

Ассистент Нистюк О. А.

Минск, 2024

**Лабораторная работа №9**

**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

# 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.

# 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.

Функция generatePrivateKey принимает на вход начальное число последовательности и длину генерируемой последовательности. Начальное число последовательности – это случайное 100-битное число. Код функции представлен на рисунке 2.1.

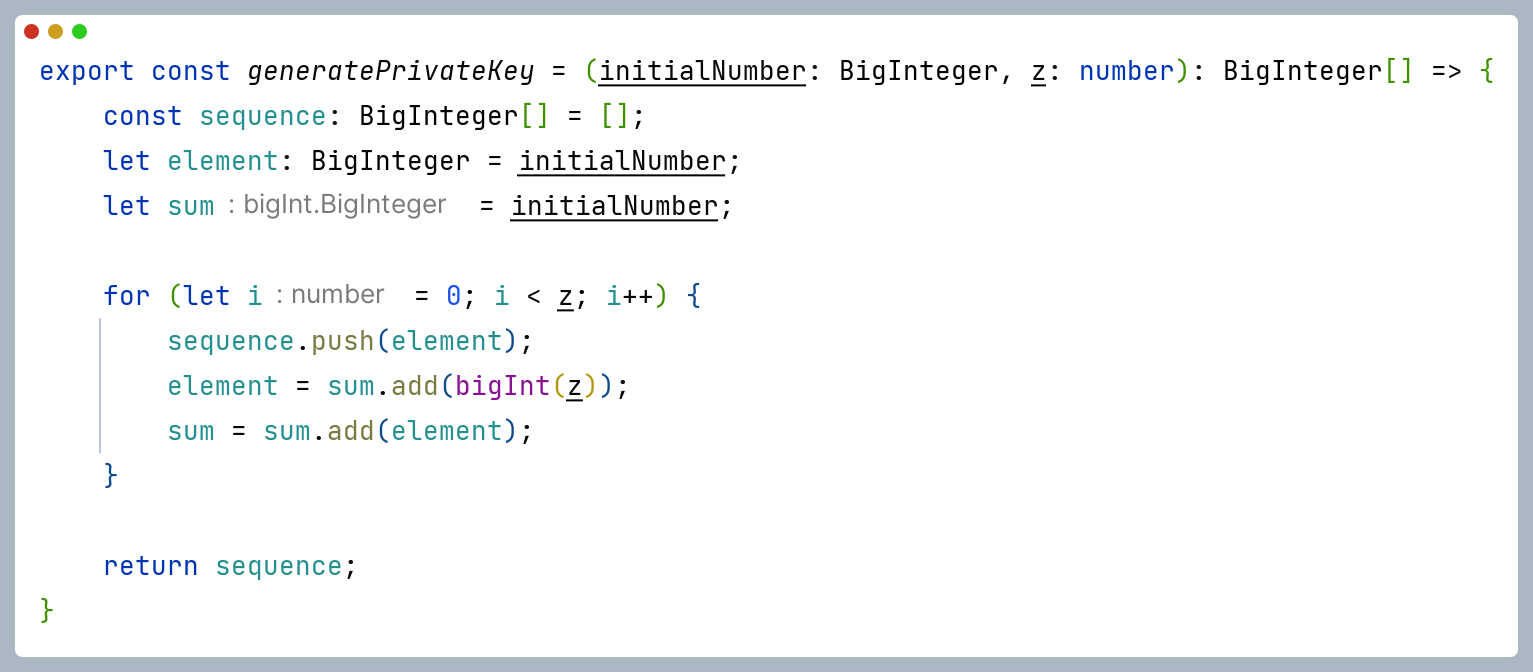


Рисунок 2.1 – Код функции генерации приватного ключа

Результат работы приложения с начальным числом 373977097140245897443365283506 и длиной z = 8 (для кодов ASCII) приведен на рисунке 2.2.

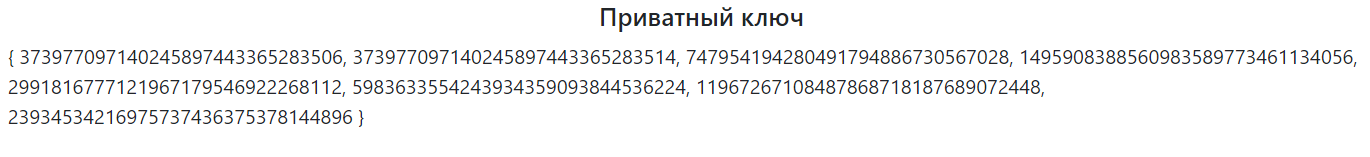


Рисунок 2.2 – Результат генерации последовательности с z = 8

Результат работы приложения с начальным числом 70959993794722279014361958728 и длиной z = 6 (для кодов base64) приведен на рисунке 2.3.

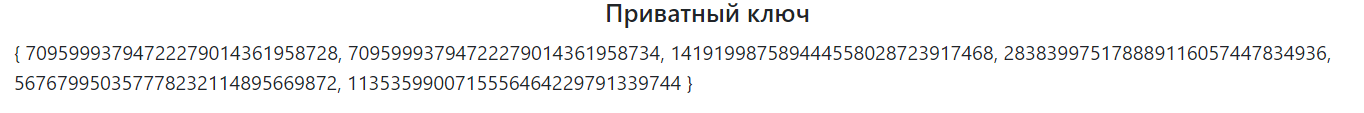


Рисунок 2.3 – Результат генерации последовательности c z = 6

Далее, на основе сгенерированного приватного ключа подбираются параметры n и a для публичного ключа. Значение модуля n должно быть больше суммы всех чисел последовательности; кроме того, НОД (а, n) = 1. Функция generatePublicKey принимает приватный ключ и параметры a и n и генерирует публичный ключ. Код функции представлен на рисунке 2.4.

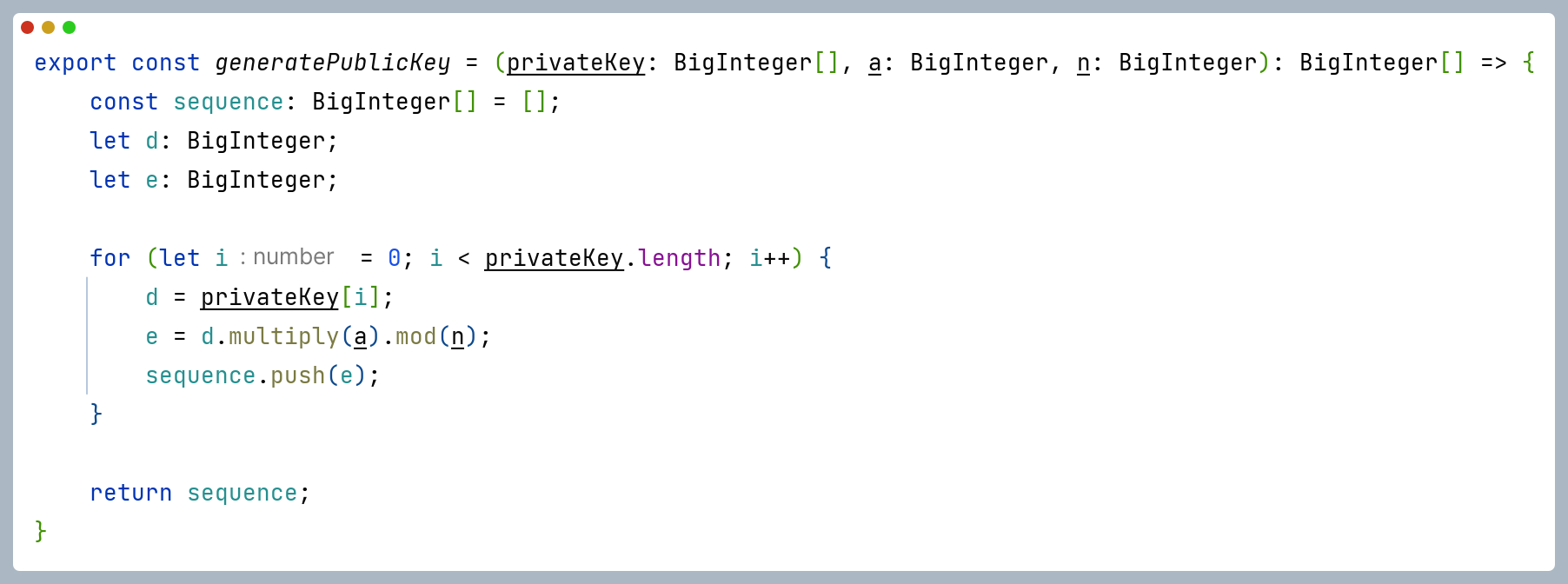


Рисунок 2.4 – Код функции, создающей публичный ключ

Результат генерации публичного ключа для сообщения в кодах ASCII представлен на рисунке 2.5.

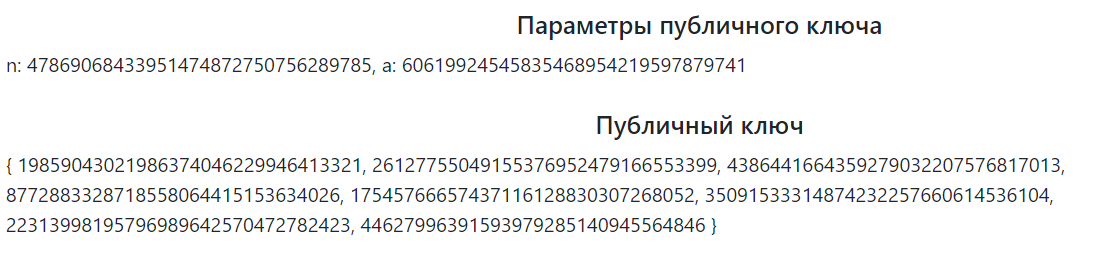


Рисунок 2.5 – Результат генерации публичного ключа для сообщения в кодах ASCII

Результат генерации публичного ключа для сообщения в кодах base64 представлен на рисунке 2.6.

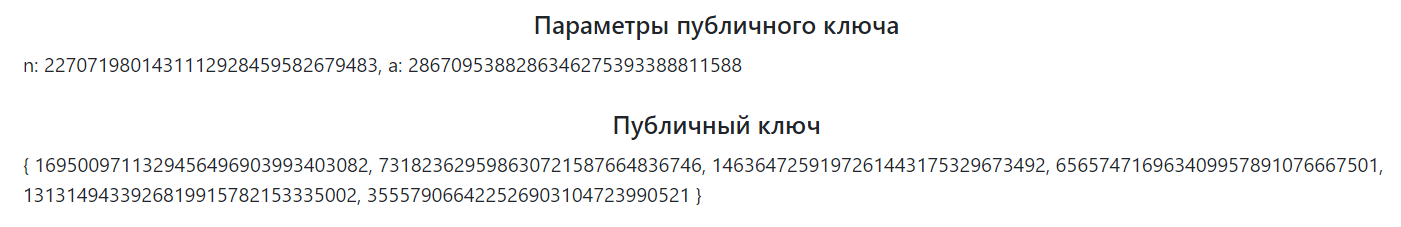


Рисунок 2.6 – Результат генерации публичного ключа для сообщения в кодах base64

Функция шифрования encrypt принимает на вход публичный ключ, открытый текст и кодировку сообщения и выполняет зашифрования сообщения на основе алгоритма укладки ранца. Код функции представлен на рисунке 2.7.

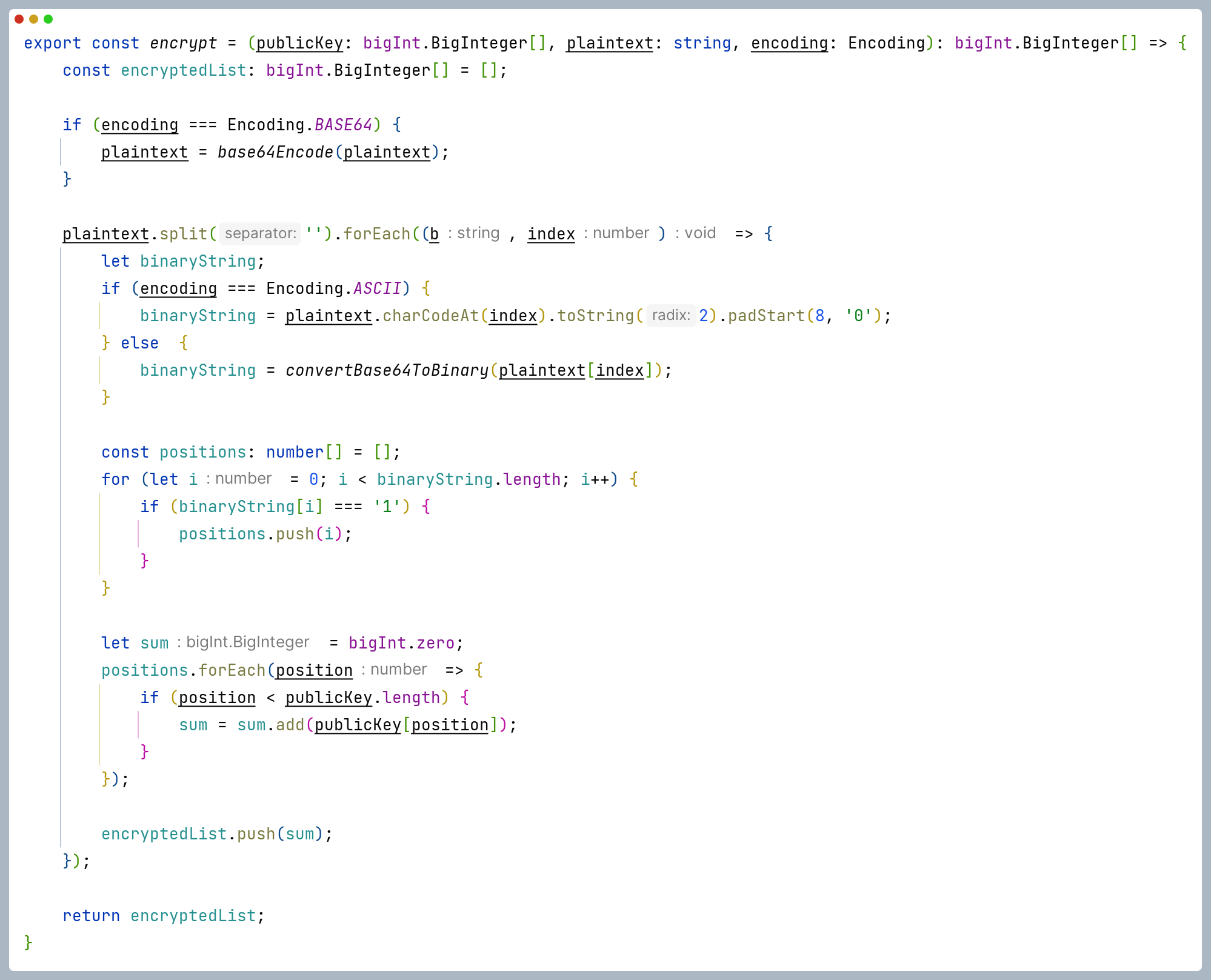


Рисунок 2.7 – Код функции шифрования

Результат шифрования текста «Ivanov Ivan Ivanovich» (в кодах ASCII) представлен на рисунке 2.8.

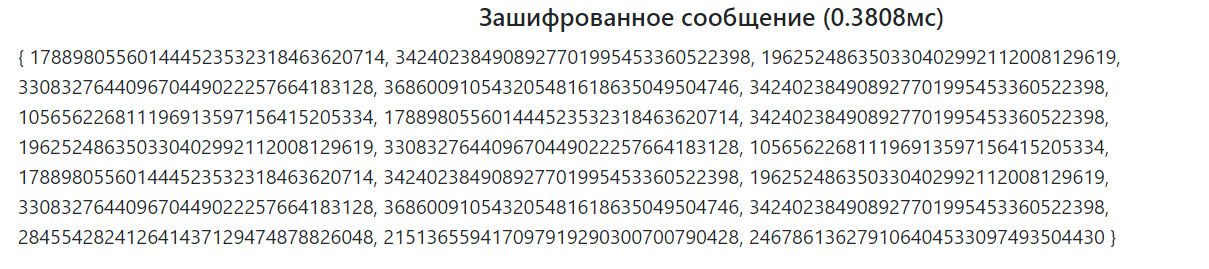


Рисунок 2.8 – Шифрование текста, представленного в кодах ASCII

Результат шифрования текста «Ivanov Ivan Ivanovich» (в кодах base64 SXZhbm92IEl2YW4gSXZhbm92aWNo) представлен на рисунке 2.9.

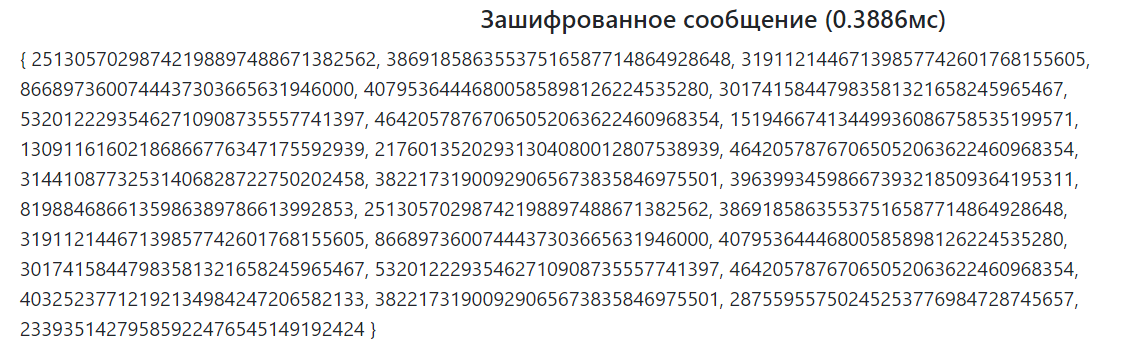


Рисунок 2.9 – Шифрование текста, представленного в кодах base64

Функция дешифрования decrypt принимает на вход приватный ключ, зашифрованный текст и параметры a и n. Код функции представлен на рисунке 2.10.



Рисунок 2.10 – Код функции дешифрования

Результат дешифрования текста в кодах ASCII представлен на рисунке 2.11.

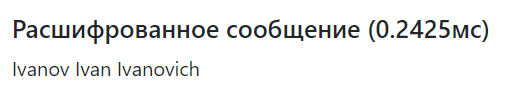


Рисунок 2.11 – Дешифрование текста, представленного в кодах ASCII

Результат дешифрования текста в кодах base64 представлен на рисунке 2.12.

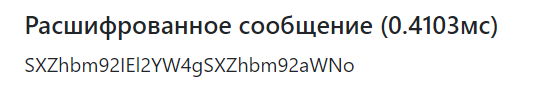


Рисунок 2.12 – Дешифрование текста, представленного в кодах base64

Кроме того, приложение выполняет оценку скорости шифрования и дешифрования. Можно заметить, что шифрование сообщения, представленного в кодах ASCII и base64 выполняется примерно за одно и тоже время. Дешифрование сообщения, представленного в кодах ASCII, выполняется немного дольше, чем дешифрование сообщения в кодах base64. При увеличении числа членов ключевой последовательно время шифрования и дешифрования растет как для текста, представленного в кодах ASCII, так и для текста, представленного в кодах base64.

**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 и base64.