Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Криптографические методы защиты информации

Студент: Сенченя В.И.

ФИТ 3 курс 5 группа

Преподаватель: Савельева М. Г.

Минск 2023

**Лабораторная работа №9**

**Тема «****Исследование асимметричных шифров»**

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

Задачи:

1. Закрепить теоретические знания по алгебраическому описанию, алгоритмам реализации операций зашифрования/расшифрования и оценке криптостойкости асимметричных шифров.
2. Разработать приложение для реализации указанных преподавателем методов генерации ключевой информации и ее использования для потокового зашифрования/расшифрования.
3. Выполнить анализ криптостойкости потоковых 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.

**Ассиметричный шифр**

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.

**Алгоритм шифрования укладкой ранца**

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: *d*1, *d*2, …, *dz*: *d* = {*di*}, *i* = 1, …, *z*.

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

Для получения открытого ключа *e* (*e* = {*ei*}, *i* = 1, …, *z*) все значения закрытого ключа умножаются на некоторое число a по модулю n:

*ei ≡ dia mod n*

Код реализации генерации сверхвозрастающей последовательности предоставлен на рисунке 1.1, а результат выполнения на рисунке 1.2

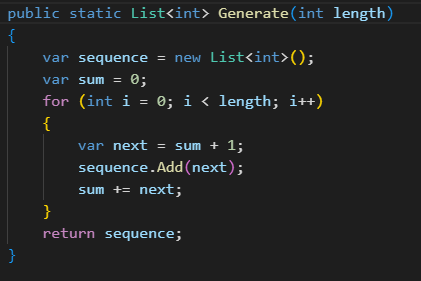


Рисунок 1.1 – Код реализации генерации сверхвозрастающей последовательности



Рисунок 1.2 – Результат работы кода генерации сверхвозрастающей последовательности

Открытый ключ e представляет собой нормальную (не сверхвозрастающую) последовательность. Он формируется на основе закрытого ключа и не позволяет легко решить задачу об укладке ранца. Для получения открытого ключа *e* (*e* = {*ei*}, i = 1, …, z) все значения закрытого ключа умножаются на некоторое число *a* по модулю *n:*

*ei* ≡ *dia mod n*.

Значение модуля n должно быть больше суммы всех чисел последовательности; кроме того, НОД (*а*, *n*) = 1.

Код реализации алгоритма по получению нормальной последовательности предоставлен на рисунке 1.3, а пример работы алгоритма на рисунке 1.4

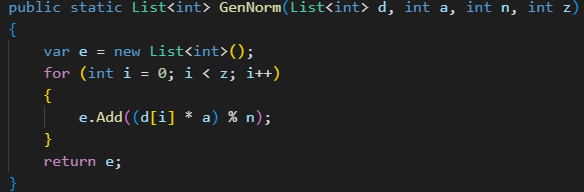


Рисунок 1.3 – Код реализации генератора нормальной последовательности

****

Рисунок 1.4 – Результат работы генератора нормальной последовательности

Шифрование происходит по алгоритму укладке рюкзака, где мы разбиваем сообщение на символы, каждый символ на байты и позицию байтов, которые хранят в себе 1, заменяем на соответствующую позицию в открытом ключе. Пример кода реализации предоставлен на рисунке 1.5, а пример работы программы на рисунке 1.6.

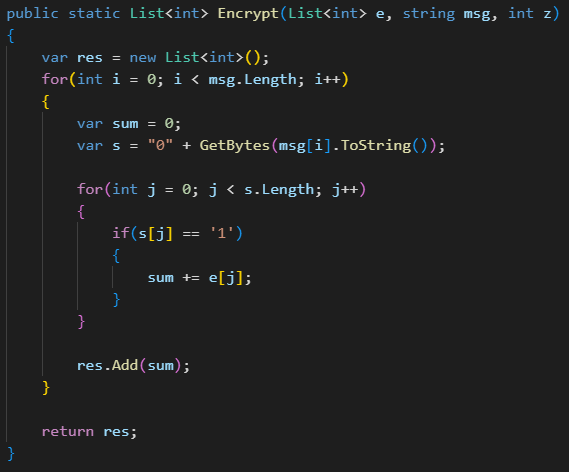


Рисунок 1.5 – Код реализации алгоритма укладки рюкзака



Рисунок 1.6 – Пример работы алгоритма укладки рюкзака

Для расшифрования необходимо посчитать обратное числа *a* по модулю *n.* После определения обратного числа каждое значение шифрограммы (*ci*) преобразуется в соответствии со следующим соотношением:

*Si* ≡ *ciа*-1 *mod n*.

Далее происходит распаковка рюкзака примерно по тому же принципу, что и упаковка, только используется тайный ключ получателя и вычитание вместо суммирования.

Пример программы реализация алгоритма распаковки предоставлен на рисунке 1.7, а пример работы программы на рисунке 1.8.

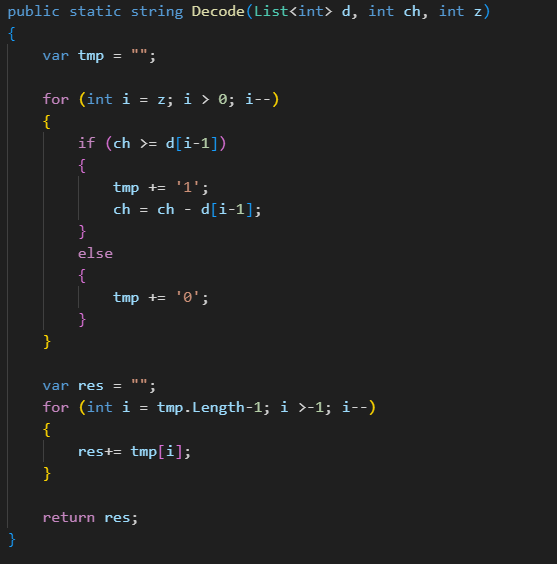


Рисунок 1.7 – Пример программы реализации алгоритма распаковки

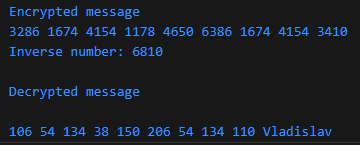


Рисунок 1.8 – Результат работы алгоритма распаковки

Скорости шифрование и дешифрования при различных **z** предоставлены на рисунке 1.9 и 1.10

Рисунок 1.9 – Время выполнения программы при *z* = 8

Рисунок 1.10 – Время выполнения программы при *z* = 6

**Вывод**

В ходе выполнения лабораторной работы я приобрёл и закрепил навыки практические навыки разработки приложения для реализации ассиметричных шифров.