Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный технологический университет»

Факультет информационных технологий

Кафедра информационных систем и технологий

**Отчет к лабораторной работе №7**

**«Исследование ассиметричных шифров»**

Выполнил:

студентка 3 курса 2 группы

Черноок Ю. С.

Проверил:

ассистент

Копыток Д. В.

Минск 2021

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

**Теоретическая часть**

Две известные нам проблемы, связанные с практическим использованием симметричных криптосистем, стали важными побудительными мотивами для разработки принципиально нового класса методов шифрования: криптографии с открытым ключом, или асимметричной криптографии. Концепция нового подхода предложена УитфилдомДиффи (WhitfieldDiffie) и Мартином Хеллманом (MartinHellman) и, независимо, Ральфом Мерклом (RalphMerkle).

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-wayfunction) 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.

Практически первой реализацией идеи Диффи – Хеллмана стал алгоритм согласования по открытому каналу тайного ключа между абонентами А и В. 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.

В данной работе мы будем работать над аспектами решения первой из указанных задач. По мнению Диффи и Хеллмана, алгоритм шифрования с открытым ключом должен:

* вычислительно легко создавать пару (открытый ключ e – закрытый ключ d);
* вычислительно легко зашифровывать сообщение Mi открытым ключом;
* вычислительно легко расшифровывать сообщение Ci, используя закрытый ключ;
* обеспечивать непреодолимую вычислительную сложность определения соответствующего закрытого ключа при известном открытом ключе;
* обеспечивать непреодолимую вычислительную сложность восстановления исходного (открытого сообщения Mi) зная только открытый ключ и зашифрованное сообщение Ci.

**Общая характеристика алгоритма**.

Алгоритм разработан Р. Мерклом и М. Хеллманом. Это первый алгоритм шифрования с открытым ключом широкого назначения. Определение 2. 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. В наиболее известном варианте задачи о ранце требуется выяснить, обладает или нет данный вход (S, S) решением. В варианте, используемом в криптографии, нужно для данного входа (S, S) построить решение, зная, что такое решение существует. Оба эти варианта являются NP-полными. Имеются также варианты этой задачи, которые не лежат даже в классе NP. Как видим, проблема укладки ранца формулируется просто. Дано множество предметов общим числом z различного веса. Спрашивается, можно ли положить некоторые из этих предметов в ранец так, чтобы его вес стал равен определенному значению S? Более формально задача формулируется так: дан набор значений k1, k2, …, kz и суммарное значение S. Требуется вычислить значения bz такие, что S = b1k1 + b2k2 + ... + bzkz. (7.1) Здесь bi может быть либо нулем, либо единицей. Значение bi = 1 означает, что предмет mi кладут в рюкзак, а bi = 0 – не кладут.

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.

**Алгоритм укладки ранца на основе сверхвозрастающей последовательности**

Необходимо по очереди анализировать некоторый «текущий вес» S предметов, составляющих сверхвозрастающую последовательность; в результате анализа нужно упаковать (доупаковать) ранец.

1. В качестве текущего выбирается число S, которое сравнивается с «весом» самого тяжелого предмета (dz); если текущий вес меньше веса данного предмета, то его в ранец не кладут (0), в противном случае его укладывают (1) в ранец и переходят к анализу очередного (в общем случае – i-го предмета).

2. Если на предыдущем (i-м шаге) предмет пополнил ранец, то текущий вес уменьшают на вес положенного предмета (S = S – di); переходят к следующему по весу предмету в последовательности: di – 1.

Шаги повторяются до тех пор, пока процесс не закончится. Если текущий вес уменьшится до нуля (S = 0), то решение найдено. В противном случае – нет.

**Практическая часть**

Разработать авторское оконное приложение в соответствии с целью лабораторной работы. 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.

Программная реализация выполнена на языке C# с использованием консольного приложения. Код представлен на рисунках 1-4.

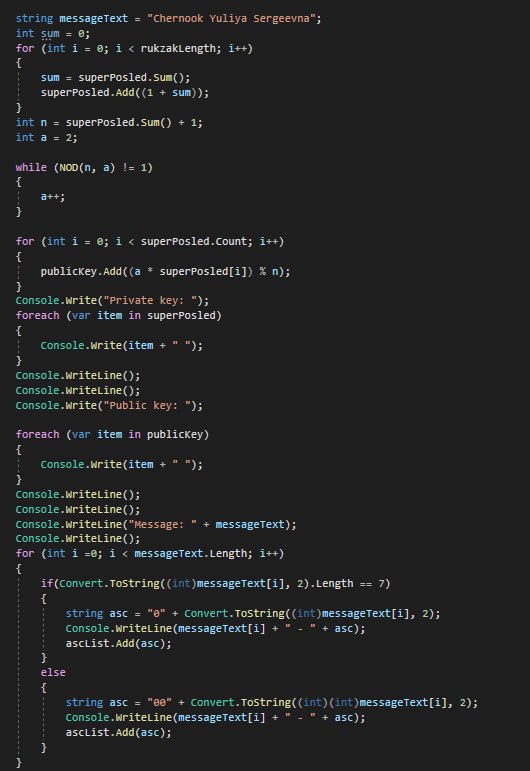


Рисунок 1 – Программная реализация алгоритма укладки ранца

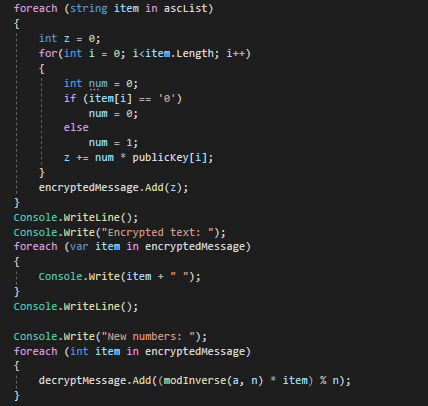


Рисунок 2 – Программная реализация алгоритма укладки ранца

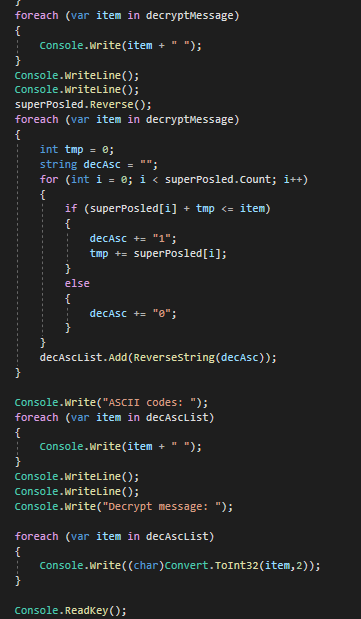


Рисунок 3 – Программная реализация алгоритма укладки ранца

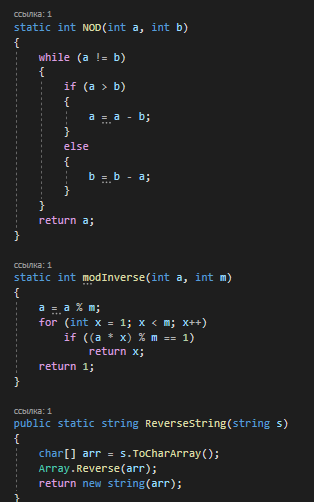


Рисунок 4 – Программная реализация алгоритма укладки ранца

Результат выполнения программы представлен на рисунке 5.

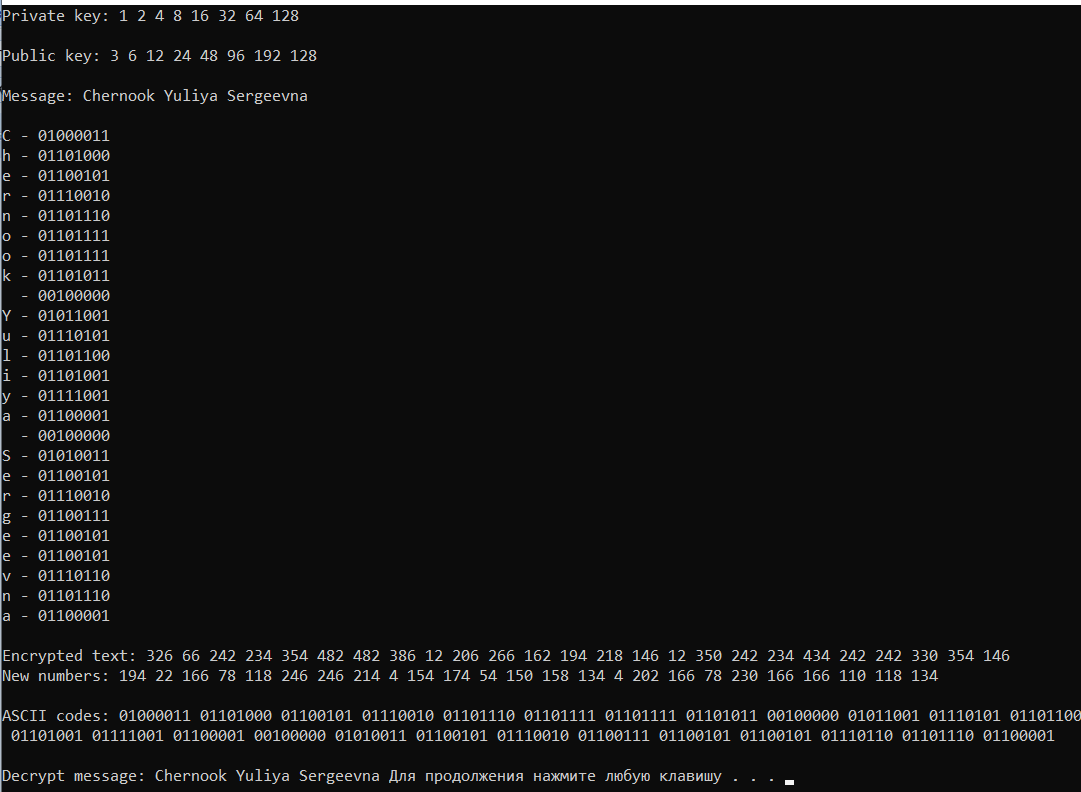


Рисунок 5 – Результат выполнения программы

Вывод: я изучила и приобрела практические навыки разработки и использования приложений для реализации ассиметричных шифров.