МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образования «БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ

ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Отчет по лабораторной работе №7

Исследование ассиметричных шифров

Выполнил студент: Плюто Э. В.

ФИТ 3 курса, 5 группа

Проверил: Савельева М. Г.

Минск 2024

**Практическое задание:**

**1)** Разработать авторское приложение, которое должно реализовывать генерацию сверхвозрастающей последовательности (тайного ключа); старшего члена последовательности – 100-битное число; вычисление нормальной последовательности (открытого ключа); зашифрование сообщения, состоящего из собственных фамилии, имени и отчества; расшифрование сообщения; оценку времени выполнения операций зашифрования и расшифрования.

Реализация создания тайного ключа представлена на листинге 1.

|  |
| --- |
| public static int[] GenerateSecretKey(int size)  {  Console.WriteLine("Тайный ключ: ");  int[] key = new int[size];  int from = 0, to = 1000;  Random random = new Random();  for (int i = 0; i < size; i++)  {  key[i] = (random.Next(from, to));  from = key[i];  to += key[i] \* 2;  Console.Write(Math.Abs(key[i]) + " ");  }  return key;  } |

Листинг 1 – Функция генерации тайного ключа

Реализация создания открытого ключа представлена на листинге 2.

|  |
| --- |
| public static int[] GenerateOpenKey(int[] d, int a, int n, int size)  {  Console.WriteLine("\nОткрытый ключ: ");  int[] key = new int[size];  for (int i = 0; i < size; i++)  {  key[i] = (d[i] \* a) % n;  Console.Write(Math.Abs(key[i]) + " ");  }  return key;  } |

Листинг 2 – Функция генерации открытого ключа

Результат выполнения вышеописанных функций представлен на рисунке 1.

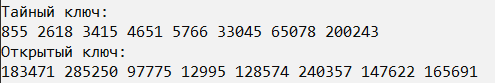


Рисунок 1 – Сгенерированные тайный и открытый ключи

Реализация шифрования сообщения представлена ниже.

|  |
| --- |
| public static string Encrypt(int[] key, string message)  {  Console.WriteLine("\nЗашифрованный текст: ");  int[] result = new int[message.Length];  int j = 0;  StringBuilder encryptedMessage = new StringBuilder();  foreach (char ch in message)  {  int total = 0;  string binary = "0" + Convert.ToString(ch, 2);  if (binary.Length == 7)  binary = "0" + binary;  encryptedMessage.Append(binary + " ");  for (int i = 0; i < binary.Length; i++)  {  if (binary[i] == '1')  {  total += key[i];  }  }  result[j] = total;  Console.WriteLine(binary + " - " + result[j] + " ");  j++;  }  return encryptedMessage.ToString();  } |

Листинг 3 – Функция шифрования сообщения

Время выполнения шифрования и расшифрования представлены ниже.

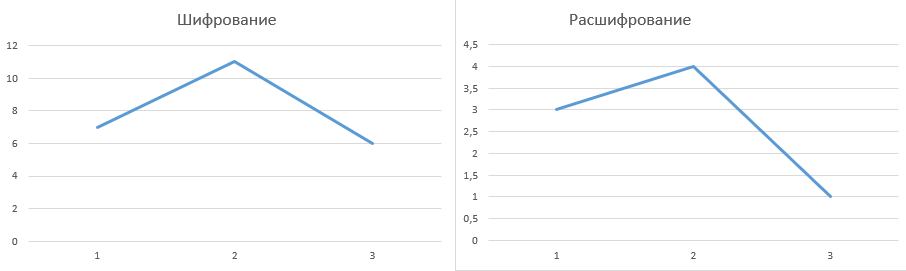


Рисунок 2 – Время выполнения шифрования и расшифрования

Результат выполнения шифрования и расшифрования представлен на рисунке 3.

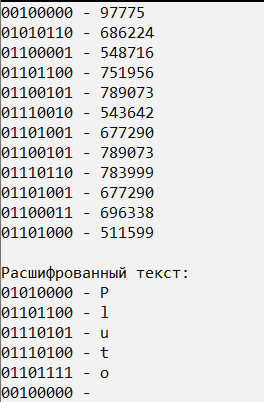


Рисунок 3 – Результат выполнения шифрования и расшифрования

Время выполнения шифрования и расшифрования при увеличении количества членов последовательности представлены ниже.

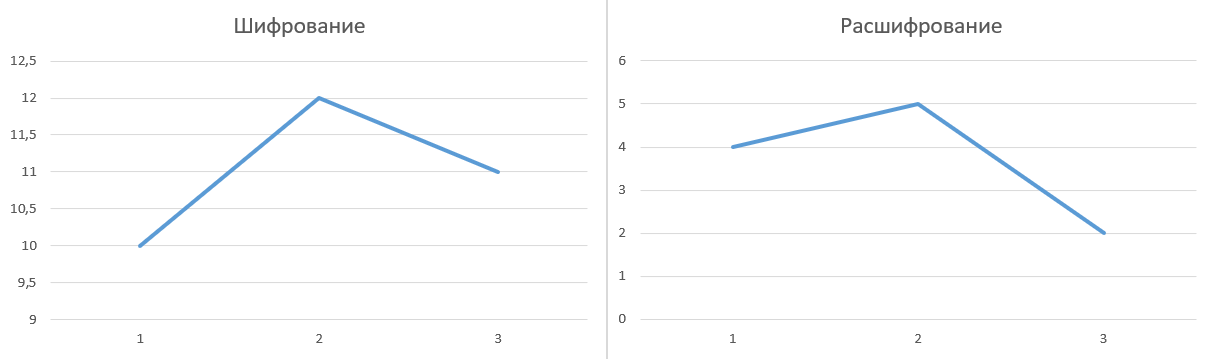


Рисунок 4 – Время выполнения с увеличенным количеством членов последовательности

При увеличении количества членов последовательности также увеличивается и время выполнения шифрования и расшифрования, что можно заметить на представленных выше графиках.

Время выполнения шифрования и расшифрования с использованием кодировки base64 представлено на рисунке 5.

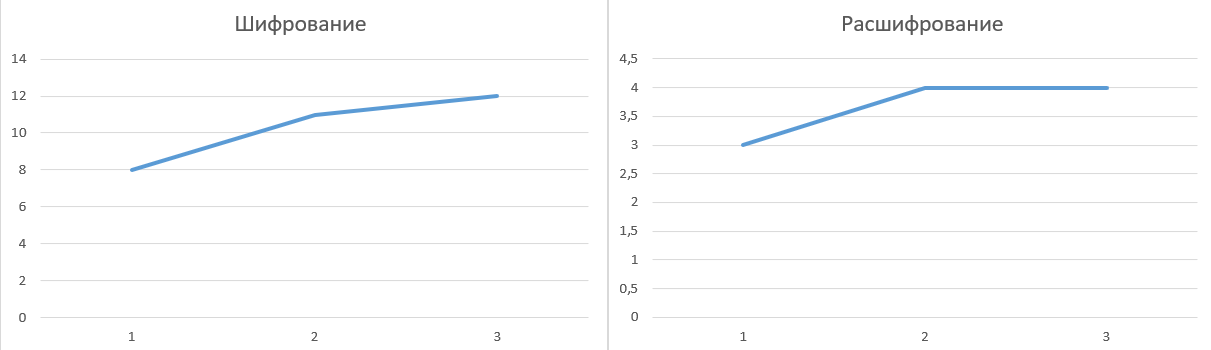


Рисунок 5 – Время выполнения с кодировкой base64

Из графиков можно заметить, что время выполнения схоже с исходным.

**Вывод:**

Были приобретены практические навыки разработки и использования приложений для реализации ассиметричных шифров. Были реализованы функции для генерации открытого и закрытого ключей. Также было замерено время выполнения шифрования и расшифрования, а также построен соответствующий график.