МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образования «БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ

ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Отчет по лабораторной работе №9

Исследование асимметричных шифров

Выполнил студент: Авсюкевич П.В.

ФИТ 3 курса, 5 группа

Проверил: Савельева М. Г.

Минск 2024

Задание 1. Генерация сверх возрастающей последовательности (тайного ключа). Для данного задания была разработана функция для генерации тайного ключа. Она представлена в листинге 1.

|  |
| --- |
| public static List<BigInteger> GeneratePrivateKey(BigInteger initialNumber, int quantityOfNumbers)  {  List<BigInteger> sequence = new List<BigInteger>();  BigInteger term = initialNumber;  for (int i = 0; i < quantityOfNumbers; i++)  {  sequence.Add(term);  term += initialNumber;  initialNumber <<= 1;  }  return sequence;  } |

Листинг 1 – Генерация тайного ключа

Задание 2. В данном задании требовалось генерировать открытй ключ, генерация представлена на листинге 2. Генерация открытого ключа происходит на основе закрытого ключа.

|  |
| --- |
| public static List<BigInteger> GeneratePublicKey(List<BigInteger> privateKey, BigInteger a, BigInteger n)  {  var sum = Sum(privateKey);  if (n <= sum)  throw new ArgumentException("n should be more than sum of all numbers in private key.");  if (!AreRelativelyPrime(a, n))  throw new ArgumentException("a and n should be coprime.");  var publicKey = new List<BigInteger>();  foreach (BigInteger d in privateKey)  {  BigInteger e = d \* a % n;  publicKey.Add(e);  }  return publicKey;  } |

Листинг 2 – Генерация открытого ключа

Задание 3. Зашифрование сообщения, состоящего из собственных фамилии, имени и отчества. Для реализации данного задания исходный текст переводится в двоичный формат, в тех битах, которые раны 1 будет добавляться значение отрытого ключа. Реализация представлена на листинге 3.

|  |
| --- |
| public static List<BigInteger> Encrypt(List<BigInteger> publicKey, byte[] plaintext)  {  var stopWatch = new Stopwatch();  stopWatch.Start();  var encryptedList = new List<BigInteger>();  foreach (byte b in plaintext)  {  string binaryString = Convert.ToString(b, 2).PadLeft(8, '0');  var positions = new List<int>();  for (int i = 0; i < binaryString.Length; i++)  if (binaryString[i] == '1')  positions.Add(i);  var sum = BigInteger.Zero;  foreach (int position in positions)  if (position < publicKey.Count)  sum += publicKey[position];  encryptedList.Add(sum);  }  stopWatch.Stop();  return encryptedList;  } |

Листинг 3 – Зашифрование ФИО

Задание 4. Расшифрование сообщения. Реализация представлена на листинге 4.

|  |
| --- |
| public static byte[] Decrypt(List<BigInteger> privateKey, List<BigInteger> encryptedText, BigInteger a, BigInteger n)  {  var stopWatch = new Stopwatch();  stopWatch.Start();  var decryptedBytes = new List<byte>();  BigInteger inverse = GetInverseNumber(a, n);  foreach (BigInteger cipher in encryptedText)  {  BigInteger decryptedValue = (cipher \* inverse) % n;  var binaryString = ReverseString(GetBinaryRepresentation(decryptedValue, privateKey));  byte decryptedByte = Convert.ToByte(binaryString, 2);  decryptedBytes.Add(decryptedByte);  }  stopWatch.Stop();  return decryptedBytes.ToArray();  } |

Листинг 4 – Расшифровка закодированного текста

Задание 4. Оценка времени выполнения операций зашифрования и расшифрования.

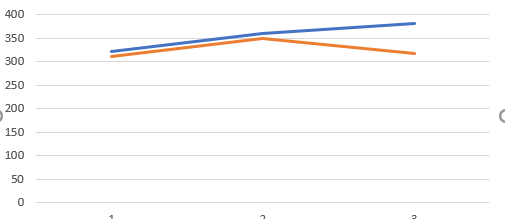


Рисунок 1 – Оценка времени выполнения работы программы

Оценка время выполнения с другой кодировкой, base64.

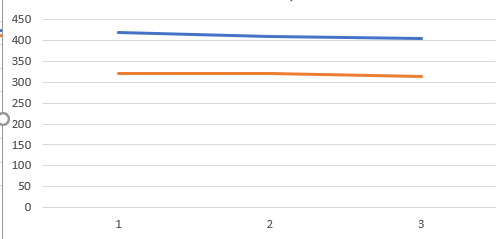


Рисунок 2 – Время выполнения с base64

Время выполнения для количества символов: 100, 1000, 10000 представлено на рисунке 3.

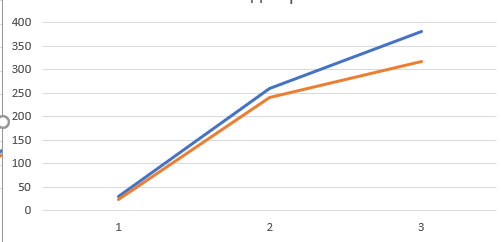


Рисунок 3 – Время выполнения с различным количеством символов

Результат выполнения представлен на рисунке 4.

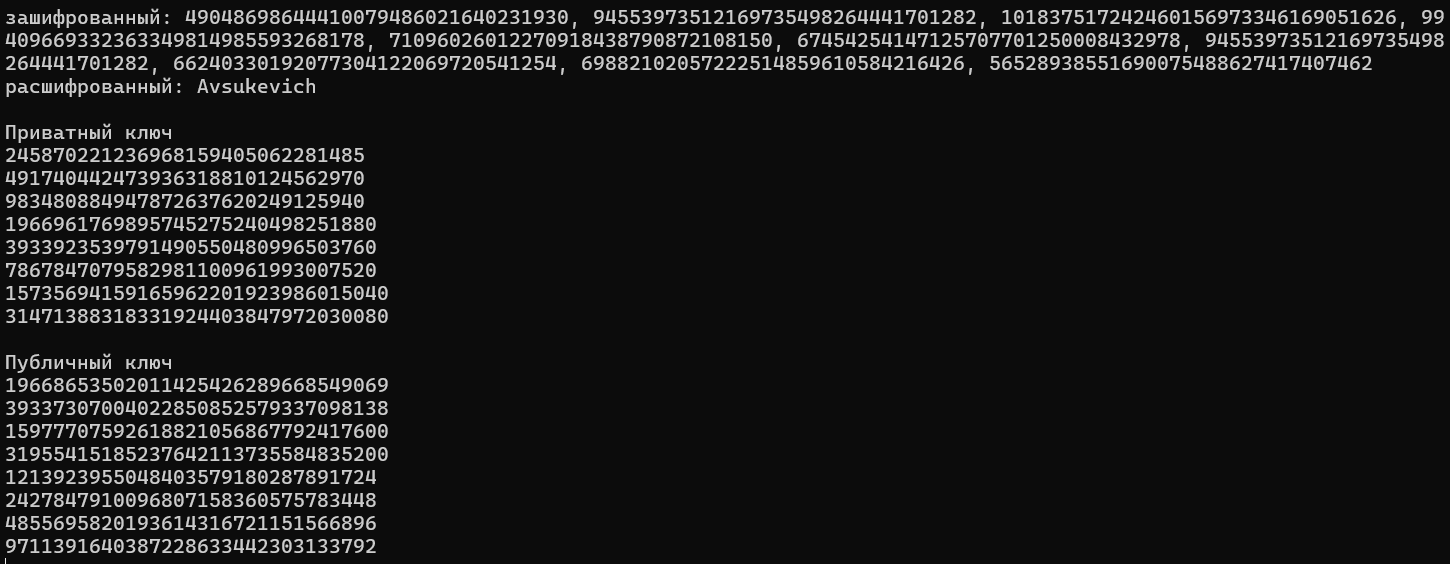


Рисунок 4 – Результат выполнения

В результате выполнения лабораторной работы были освоены и применены практические навыки в создании и использовании приложений для применения асимметричного шифрования. Кроме того, было разработано собственное приложение в соответствии с поставленными целями, включая функции генерации открытого и закрытого ключей, шифрования и расшифрования текста, а также оценки времени выполнения данных операций.