Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет информационных технологий

Кафедра информационных систем и технологий

Специальность 1-40 01 01 Программное обеспечение информационных технологий

**Отчёт по лабораторной работе №10**

**«Исследование ассиметричных шифров RSA и Эль-Гамаля»**

**Вариант 3**

Исполнитель:

Студент 3 курса группы 4

Гурина К. С.

Руководитель:

Ассистент Сазонова Д. В.

1. **Цель и задачи работы**

**Цель:** изучение и приобретение практических навыков разработки и использования приложений для реализации асимметричных шифров RSA и Эль-Гамаля.

**Задачи:**

1. Закрепить теоретические знания по алгебраическому описанию, алгоритмам реализации операций зашифрования/расшифрования и оценке криптостойкости ассиметричных шифров RSA и Эль-Гамаля.

2. Разработать приложение для реализации асимметричного зашифрования/ расшифрования на основе алгоритмов RSA и Эль-Гамаля.

3. Выполнить анализ криптостойкости ассиметричных шифров RSA и Эль-Гамаля.

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

**2. Теоретические сведения**

Асимметричная криптография основана на сложности решения некоторых математических задач. По существу, таких задач две:

• разложение больших чисел на простые сомножители (задача факторизации);

• вычисление дискретного логарифма в конечном поле, а также вычислительные операции над точками эллиптической кривой.

Задача дискретного логарифмирования формулируется так: для данных целых чисел а и b, 1 < а, b < n, найти логарифм – такое целое число х, что

ax ≡ b (mod n),

если такое число существует.

Китайская теорема об остатках. В общем случае если разложение числа N на простые множители представляет собой p1p2…pt (некоторые простые числа могут встречаться несколько раз), то система уравнений

(x mod pi) ≡ ai,

где i = 1, 2, …, t, имеет единственное решение: x, меньшее N.

Алгоритм RSA. Для генерации двух ключей: тайного и открытого (а по сути –двух взаимосвязанных частей одного ключа, т. е. ключа, принадлежащего одному физическому лицу (или группе лиц), либо одному юридическому лицу), используются два больших случайных простых числа p и q. Для максимальной большей криптостойкости нужно выбирать p и q равной длины. Рассчитывается произведение: n = pq. Это есть один из трех компонент ключа, состоящего из чисел n, e, d.

Затем случайным образом выбирается второй компонент ключа: открытый ключ или ключ зашифрования, e, такой что e и (p – 1)(q – 1) являются взаимно простыми числами; (p – 1)(q – 1) = φ(n) – функция Эйлера.

Наконец, расширенный алгоритм Евклида используется для вычисления третьего компонента ключа: ключа расшифрования d такого, что выполняется условие:

ed ≡ 1 (mod φ(n)).

Зашифрование. Если шифруется сообщение М, состоящее из r блоков: m1, m2, …, mi, …, mr, то шифртекст С будет состоять из такого же числа (r) блоков, представляемых числами:

ci ≡ (mi)e mod n.

Расшифрование. Для расшифрования каждого зашифрованного блока производится вычисление вида:

mi ≡ (ci)dmod n.

Алгоритм Эль-Гамаля. Безопасность алгоритма Эль-Гамаля, как и безопасность алгоритма Диффи – Хеллмана, основана на трудности вычисления дискретных логарифмов. Алгоритм Эль-Гамаля фактически использует схему Диффи – Хеллмана, чтобы сформировать общий секретный ключ для абонентов, передающих друг другу сообщение, и затем сообщение шифруется путем умножения его на этот ключ.

**3. Ход работы**

**Практическое задание:**

1. С помощью простого консольного приложения составить табличную или графическую форму зависимости времени вычисления параметра у, функционально заданного выражением вида: у ≡ ax mod n, от параметров: а (десятичные числа от 5 до 35; можно взять 1 или 2 числа), х (числа, желательно простые, из диапазона от 103 до 10100; для примера взять 5–10 чисел, равномерно распределенных в указанном диапазоне), n (для примера взять числа, в двоичном виде состоящие из 1024 и 2048 битов).

2. Разработать авторское оконное приложение в соответствии с целью лабораторной работы. При этом можно воспользоваться доступными библиотеками либо программными кодами. В основе вычислений – кодировочные таблицы Base64 и ASCII.

Приложение должно реализовывать следующие операции:

• зашифрование и расшифрование текстовых документов на основе алгоритмов RSA и Эль-Гамаля;

• определение времени выполнения операций.

3. Используя примерно одинаковый порядок ключевой информации, оценить производительность обоих алгоритмов и относительное изменение объемов криптотекстов (по отношению к объемам открытых текстов).

**Ход работы**

 С помощью консольного приложения составить табличную форму зависимости времени вычисления параметра у, функционально заданного выражением вида:

у ≡ ax mod n,

от параметров a, x, n.

Параметр а – это десятичные числа от 5 до 35. Было выбраны числа: 5, 25. Параметр x – это простые числа из диапазона от 103 до 10100.  Было выбрано 10 чисел: 10007, 20483, 40961, 65537, 131071, 262147, 524287, 1048573, 2088571, 4176901. Время вычисления значения y возрастает при увеличении числа x. Результат работы приложения приведен на рисунке 3.1.

|  |
| --- |
|  |

Рисунок 3.1 – Вывод в консоль зависимости вычисления параметра y

Для генерации двух ключей: тайного и открытого (а по сути – двух взаимосвязанных частей одного ключа, т. е. ключа, принадлежащего одному физическому лицу (или группе лиц), либо одному юридическому лицу), используются два больших случайных простых числа p и q. Для максимальной большей криптостойкости нужно выбирать p и q равной длины. Рассчитывается произведение: n = pq. Это есть один из трех компонент ключа, состоящего из чисел n, e, d.

Затем случайным образом выбирается второй компонент ключа (открытый ключ или ключ зашифрования, e, такой что e и (p – 1)(q – 1) являются взаимно простыми числами; вспомним, что (p – 1)(q – 1) = φ(n) – функция Эйлера). Б. Шнайер [5] рекомендует число е выбирать из ряда: 3, 17, 216 + 1. Наконец, расширенный алгоритм Евклида используется для вычисления третьего компонента ключа: ключа расшифрования d такого, что выполняется условие: ed ≡ 1 (mod φ(n)). (8.4) Другими словами: d–1 ≡ e (mod φ(n)). (8.5)

Таким образом, сформирован ключ, состоящий из трех чисел, которые в свою очередь образуют две вышеупомянутые взаимосвязанные части: открытый (публичный) ключ (e, n) и тайный ключ (d, n; на самом деле, как видим, тайным здесь является лишь первое из пары чисел).

Для дешифрации сообщения нам необходимо найти число, обратное A по модулю N. После нахождения A-1 нам поэлементно найти остаток от деления произведения числа шифротекста и A-1 по модулю N. Мы получим какое-то число. Это число является суммой элементов изначальной последовательности, на месте которых была единичка в 6- или 8-битном блоке сообщения. Код функции для расшифрования представлен в листинге 2.3.

Код функции, выполняющей шифрование и дешифрование с помощью алгоритма RSA, представлен на рисунке 3.2.



Рисунок 3.2 – Код функции runRSA

Результат выполнения шифрования и дешифрование, а также время можно увидеть на рисунке 3.3.

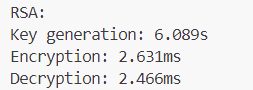


Рисунок 3.3 – Результат работы функции runRSA

Шифрование и расшифрование текстовых документов на основе алгоритма алгоритма Эль-Гамаля. Выбирается простое число р. Выбирается число (g, g < p), являющееся первообразным корнем числа р – очень важный элемент с точки зрения безопасности алгоритма (см. ниже). Далее выбирается число х (х < p) и вычисляется последний компонент ключевой информации: y ≡ gх mod р. (8.8) Владельцу сформированной ключевой информации, состоящей из 4 чисел, может посылаться некоторый шифртекст, созданный с использованием открытого ключа получателя: p, g, y. Расшифрование шифртекста получатель производит своим тайным ключом: p, g, х.

Функция принимает на вход оригинальный текст и параметры публичного ключа и осуществляет зашифрование с помощью алгоритма Эль-Гамаля. Код функции представлен на рисунке 3.4.



Рисунок 3.4 – шифрования и дешифрование сообщения с помощью алгоритма «Эль-Гамаля»

Результат выполнения шифрования и дешифрование, а также время можно увидеть на рисунке 3.6.

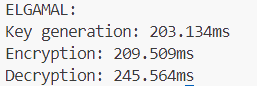


Рисунок 3.6 – Результат работы функции runElGamal

Расшифрование с помощью алгоритма Эль-Гамаля заняло больше времени по сравнению с алгоритмом RSA.

**Вывод**

В ходе выполнения лабораторной работы был изучен принцип реализации асимметричных шифров RSA и Эль-Гамаля. Также было разработано приложение, выполняющее шифрование и расшифрование с помощью шифра, основанного на алгоритме об укладке ранца. Была оценена скорость шифрования и дешифрования сообщений.