Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный технологический университет»

Факультет информационных технологий

Кафедра информационных систем и технологий

**Отчет к лабораторной работе №8**

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

Выполнил:

студентка 3 курса 2 группы

Черноок Ю. С.

Проверил:

ассистент

Копыток Д. В.

Минск 2021

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

**Теоретическая часть**

Как отмечалось выше, 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.

Эти задачи объединяет то, что они используют операцию получения остатка от целочисленного деления.

В силу этого практически все системы асимметричного зашифрования/расшифрования основаны либо на проблеме факторизации (среди них – RSA), либо на проблеме дискретного логарифмирования (среди них – Эль-Гамаля).

**Алгоритм RSA**

Рассматриваемый алгоритм появился (1977 г.) после алгоритма ранца Меркла. Он стал первым полноценный алгоритмом с открытым ключом, который впоследствии стал одним из основных для шифрования и для электронных цифровых подписей.

Из всех предложенных алгоритмов с открытыми ключами RSA проще всего понять и реализовать. Он назван в честь трех его создателей: Рона Ривеста (Ron Rivest), Ади Шамира (Adi Shamir) и Леонарда Эдлемана (Leonard Adleman).

Как было отмечено, безопасность RSA основана на трудности разложения на множители больших чисел. Открытый и закрытый ключи являются функциями двух больших простых чисел. Предполагается, что восстановление открытого текста по шифртексту и открытому ключу эквивалентно разложению на множители двух больших чисел.

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).

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

ed ≡ 1 (mod φ(n)). (1)

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 зашифрования/расшифрования используется ключ получателя: отправитель шифрует сообщение открытым ключом, а получатель расшифровывает шифртекст своим тайным ключом.

*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)еmod n. (2)

*Расшифрование*. Для расшифрования каждого зашифрованного блока производится вычисление вида:

mi ≡ (ci)dmod n. (3)

**Алгоритм Эль-Гамаля**

Предложен Т. Эль-Гамалем (T. El-Gamal) в 1985 г. Он может быть использован для решения трех основных криптографических задач: для зашифрования/расшифрования данных, для формирования цифровой подписи и для согласования общего ключа. Кроме того, возможны модификации алгоритма для схем проверки пароля, доказательства идентичности сообщения и другие варианты.

Как подчеркивалось выше, 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.

*Генерация ключевой информации.* Выбирается простое число р. Выбирается число (g, g < p), являющееся первообразным корнем числа р – очень важный элемент с точки зрения безопасности алгоритма.

Далее выбирается число х (х < p) и вычисляется последний компонент ключевой информации:

y ≡ gхmod р. (4)

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.

Как видим, на самом деле тайным является лишь одно число (как и в RSA): х.

*Зашифрование*. Блок шифртекста (ci) состоит из двух чисел – аi и bi:

ai ≡ gkmod p; (5)

bi ≡ (ykmi) mod p. (6)

Здесь стал очевидным упомянутый недостаток алгоритма шифрования Эль-Гамаля: удвоение (реально – примерно в 1,5 раза) длины зашифрованного текста по сравнению с начальным текстом.

Случайное число k должно сразу после вычисления уничтожаться. *Расшифрование ci*. Выполняется по следующей формуле:

mi ≡ (bi(ai)р – x – 1) mod p. (7)

**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:

* зашифрование и расшифрование текстовых документов на основе алгоритмов RSA и Эль-Гамаля;
* определение времени выполнения операций.

Программный код основных функций для реализации шифрования и расшифрования алгоритмом RSA представлен на рисунках 1 и 2.

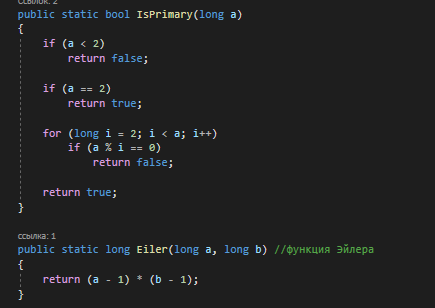
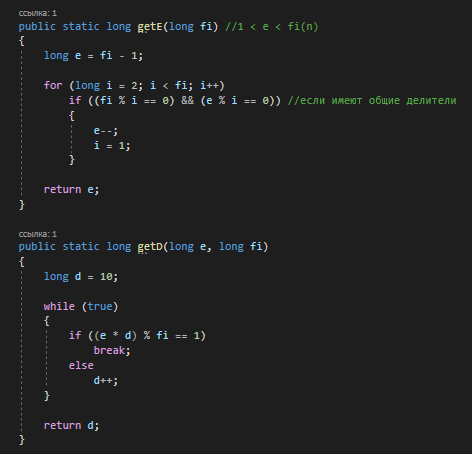


Рисунок 1 – Функции IsPrimary и Eiler



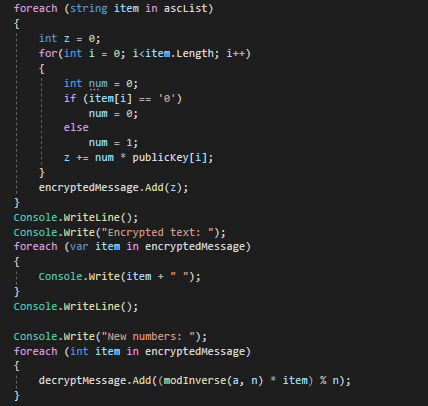


Рисунок 2 – Программная реализация алгоритма укладки ранца

Программный код реализации алгоритма Эль-Гамаля представлен на рисунке 3.

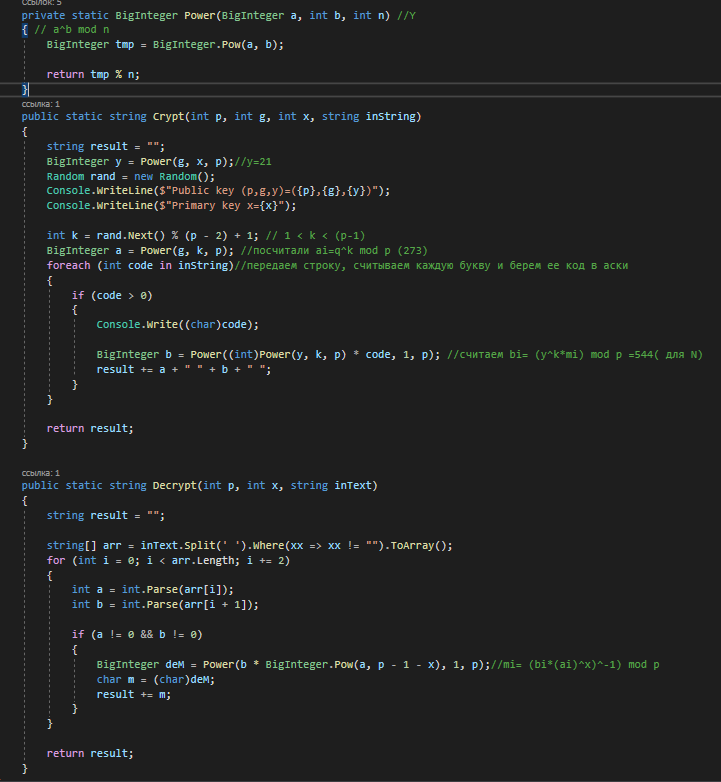


Рисунок 3 – Программная реализация алгоритма Эль-Гамаля

Результат работы алгоритма RSA представлен на рисунке 4, а алгоритма Эль-Гамаля – на рисунке 5.

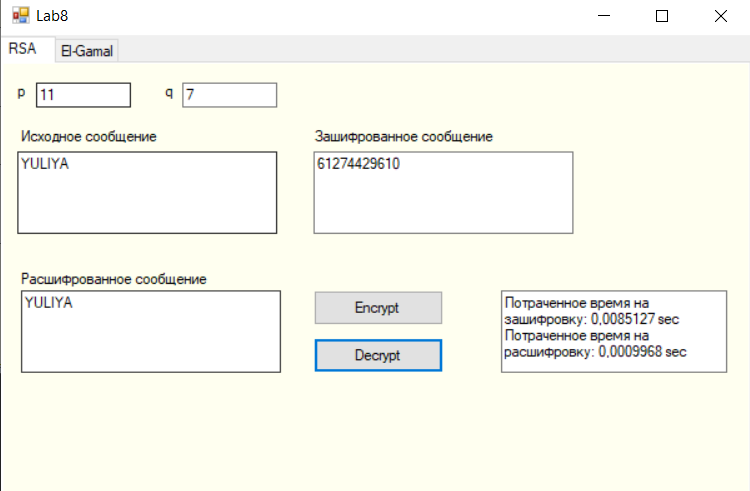


Рисунок 4 – Алгоритм RSA

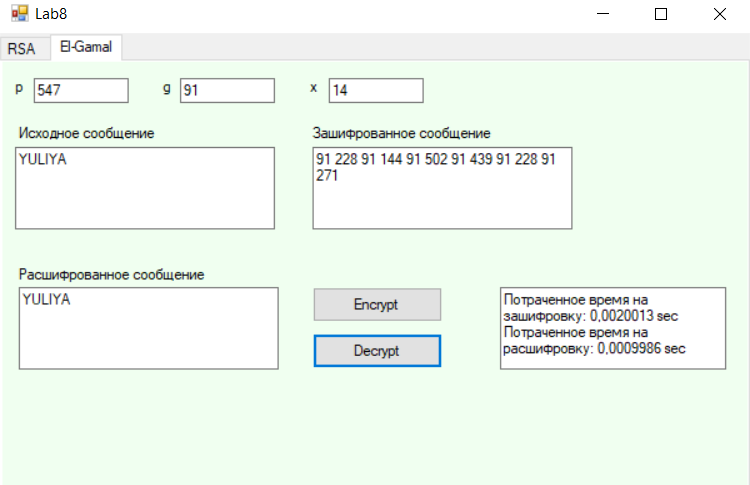


Рисунок 5 – Алгоритм Эль-Гамаля

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