МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образование «Белорусский государственный технологический университет»

Кафедра информационных систем и технологий

**Лабораторная работа №10**

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

Студент: Дубровский Михаил Сергеевич

Преподаватель:

Копыток Д.В

2021 г.

1. **Теоретическая часть**

Алгоритм RSA появился (1977 г.) после алгоритма рюкзака 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.

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 = p\*q**. Этой 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; так же нужно помнить, 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))**

Другими словами:

**d-1 = e (mod φ(n))**

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.

Использование ключа. 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)^e mod n**

Расшифрование. Для расшифрования каждого зашифрованного блока производится вычисление вида:

**mi = (ci)^d mod n**

Два числа p и q, произведение которых равно n, должны иметь приблизительно одинаковую длину, поскольку в этом случае найти сомножители (факторы) сложнее, чем в случае, когда длина чисел значительно различается. Например, если предполагается использовать 768-битный модуль, то каждое число должно иметь длину приблизительно 384 бита.

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. И в случае шифрования, и в случае формирования цифровой подписи каждому пользователю необходимо сгенерировать пару ключей.

Рассматриваемый алгоритм отличается от алгоритма RSA несколькими

параметрами и особенностями:

1) генерацией ключевой информации и числом компонент, составляющих ключ;

2) каждому блоку (символу) открытого сообщения в шифртексте на основе алгоритма Эль-Гамаля соответствуют 2 блока (в RSA – один-один);

3) в алгоритме Эль-Гамаля при зашифровании используется число (обозначим его k), которое практически никак не связано с ключевой информацией получателя и которое принимает (по определению) различные значения при зашифровании различных блоков сообщения.

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 (см. ниже).

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

ключевой информации:

**y =gх mod р**

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.

Определение 2. Первообразный корень (primary (residual ) root ) по модулю р является таким числом, что его степени (gi, 1 ≤i≤p-1 ) дают все возможные по модулю р вычеты (остатки), которые взаимно просты с p.

Понятно, что для больших значений р количество всех неповторяющихся остатков (р – 1) будет также большим. А поскольку в равнении (8.6) мы используем модуль р большого простого числа и находим первообразным корень от р, который имеет важное свойство: при использовании разных степеней (аi = ах) решение будет равномерно распределяться от 0 до р – 1, то нахождение криптоаналитиком нужного х чрезвычайно затруднено. В этом заключается односторонность функции. И на этом основывается криптостойкость шифра Эль-Гамаля.

Для схемы вероятностного шифрования само сообщение и ключ не определяют шифртекст однозначно. Зашифрование сообщения. Как ранее, предположим, что сообщение М ={mi}, где – mi – i-й блок сообщения.

Зашифрование отправителем (каждого отдельного блоками исходного

сообщения) предусматривает использование, как это особо подчеркивалось выше, некоторого случайного числа k (1 < k < p – 1). В силу использования случайной величины k шифр Эль-Гамаля называют также шифром многозначной замены, а также схемой вероятностного шифрования.

При примерно одинаковой размерности ключей рассмотренные алгоритмы обеспечивают примерно одинаковый уровень криптостойкости.

1. **Практическая часть**

Основной код программы представлен на рисунке 1:

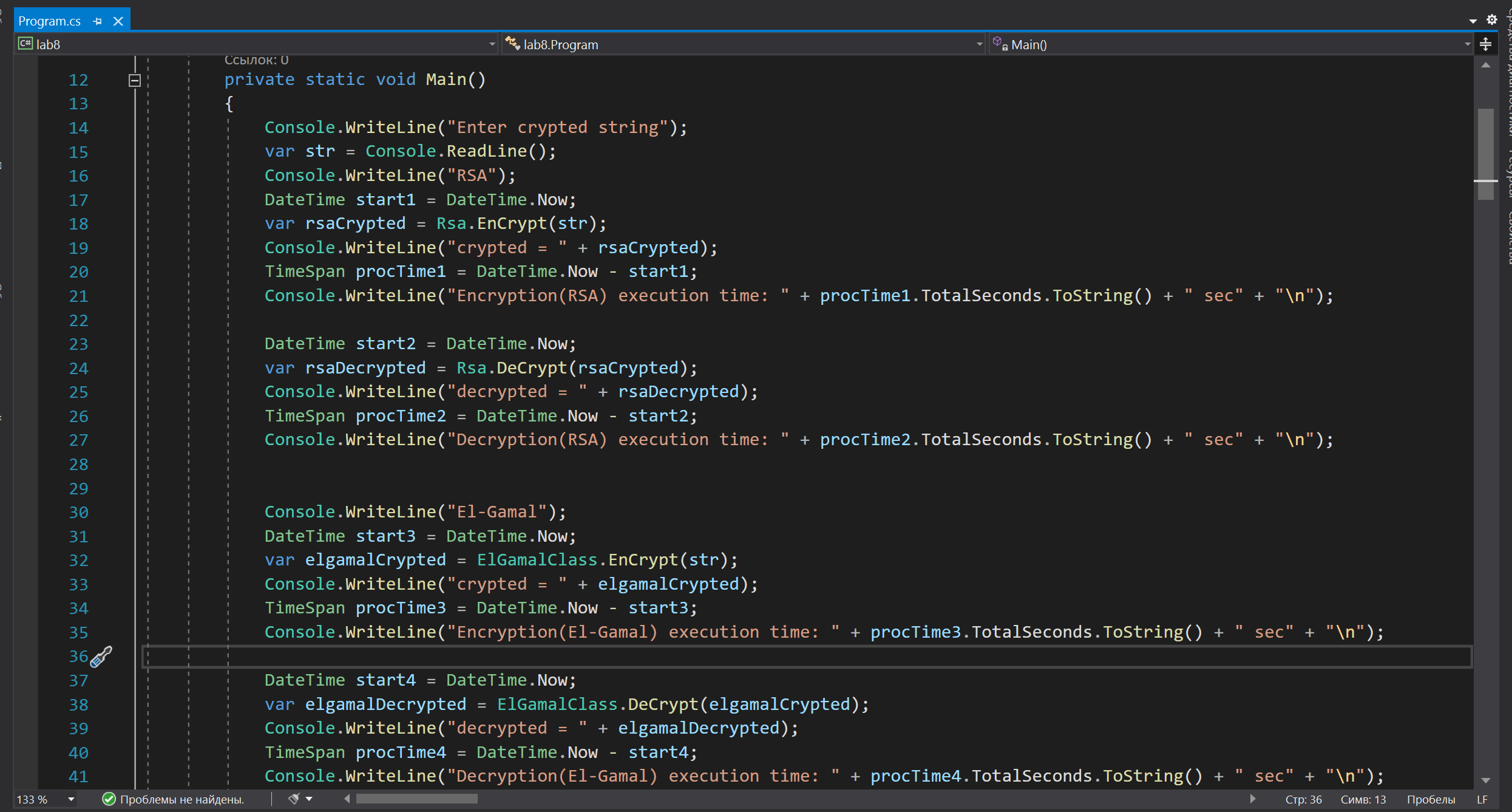


Рисунок 1 – Основной код программы

Для работы RSA была использована сторонняя библиотека и вынесена в класс RSA. Реализация этого класса представлена на рисунке 2.

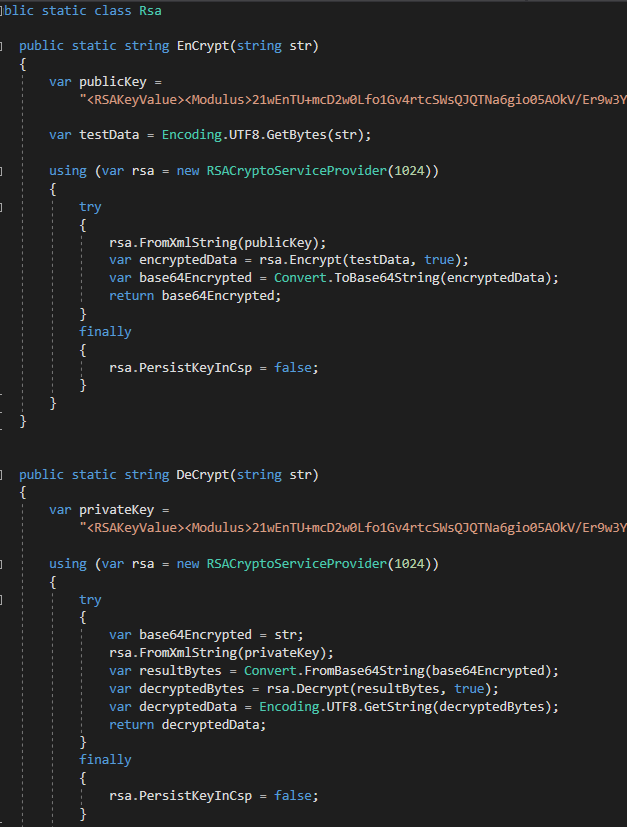
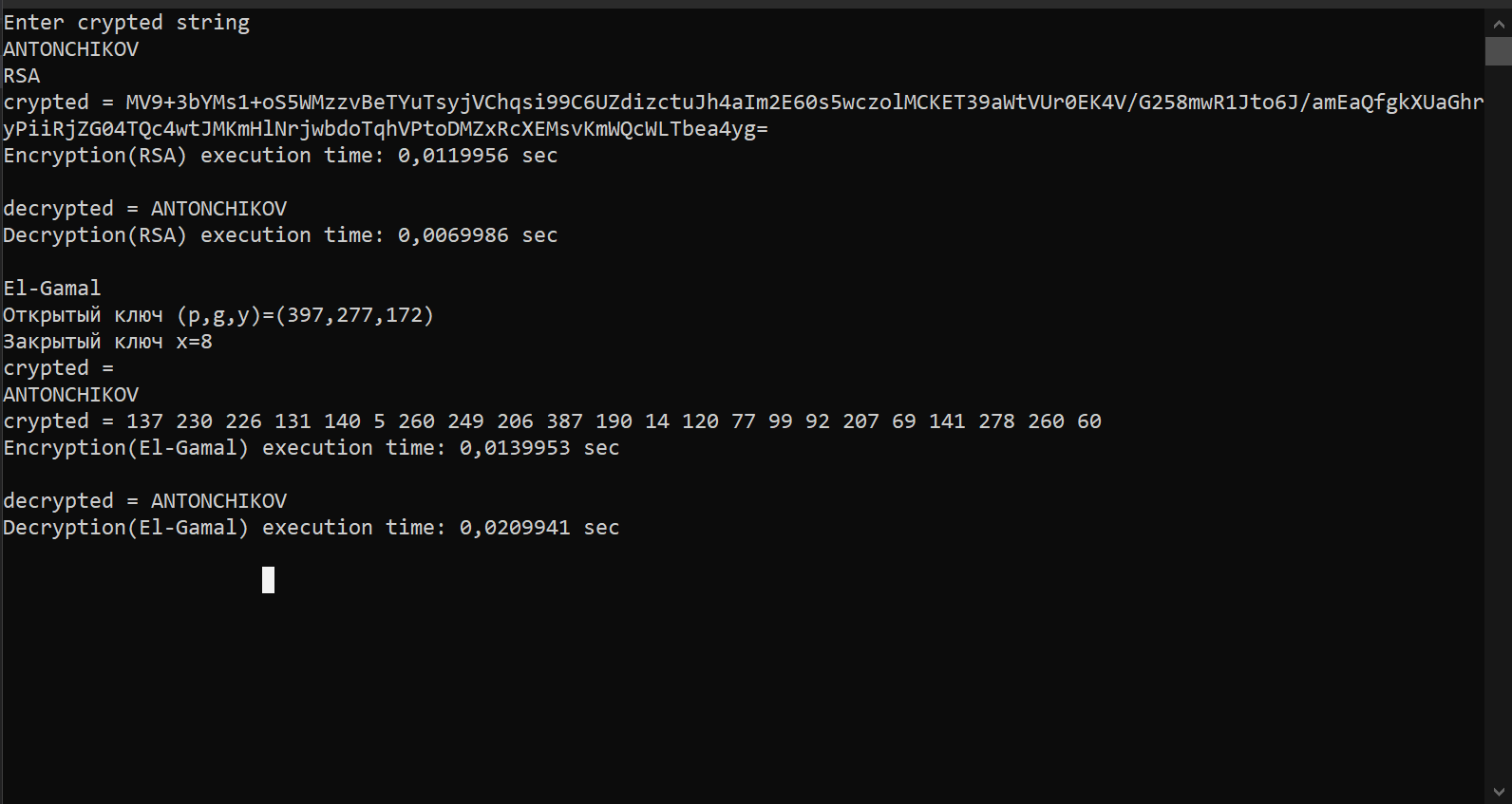


Рисунок 2 – реализация класса RSA

****Рисунок 3 – использование программы для зашифрования фамилии

**Вывод**

В ходе лабораторной работы были получены знания принципе работы ассиметричных шифров. Изучены алгоритмы RSA и Эль-Гамаля. Так же был реализован программный код, который осуществляет процесс шифрования и дешифрования информации, полученной путём ввода из консоли при помощи алгоритма RSA и Эль-Гамаля.