Министерство цифрового развития, связи и массовых коммуникаций Российской Федерации

ФГБОУ ВО «Санкт-Петербургский государственный университет телекоммуникации им. проф. М.А. Бонч-Бруевича»

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Факультет: ИКСС

Отчет по лабораторной работе №4

ТЕСТИРОВАНИЕ ПРОСТЫХ ЧИСЕЛ И НАХОЖДЕНИЕ КВАДРАТИЧНЫХ ВЫЧЕТОВ

Выполнил: Громов А.А.

Группа: ИКТЗ-83

Проверил: Яковлев В.А.

Санкт-Петербург

2021 г.

**Цель работы:**

Закрепить знания, полученные на лекциях, по теме “ Криптосистема РША”.

Используемое программное обеспечение:

Для работы используется программа “RSA”.

Задание

1. Выполнить упражнения по возведению в степень больших целых чисел по модулю.

2. Произвести генерирование ключей для криптосистемы РША, а также шифрование и дешифрование с ее помощью коротких текстов.

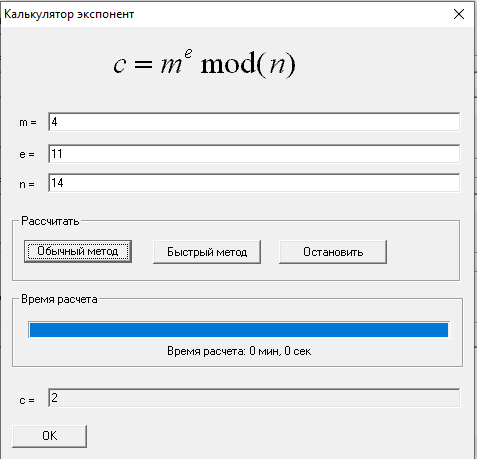
3. Произвести шифрование файлов большого размера безопасной криптосистемой РША и оценить время шифрования и дешифрования при ее простейшей программной реализации.

4. Произвести подписание сообщений и проверку подписи.

5. Осуществить обмен зашифрованными и подписанными сообщениями в локальной сети.

Порядок выполнения работы

1. Запустить программу “RSA”.
2. Выбрав в главном меню опцию “Калькулятор экспонент”, рассчитать степени малых чисел по малому модулю и проверить их правильность расчетами на бумаге.



411

11 = 8 + 2 + 1

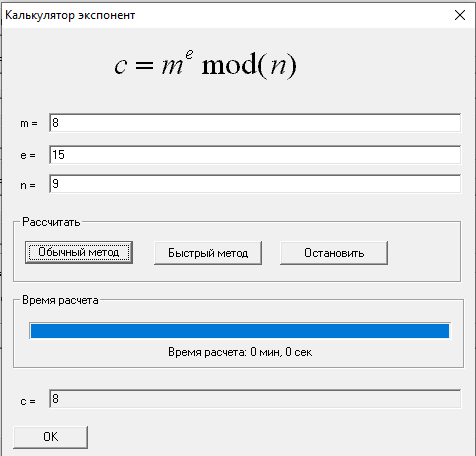
4 = 4 mod14 = 4

42 = 16 mod14 = 2

44 = 4 mod14 = 4

48 = 16 mod14 = 2

411 = (2\*2\*4) mod14 = 2



815

15 = 8 + 4 + 2 + 1

81 = 8 mod9 = 8

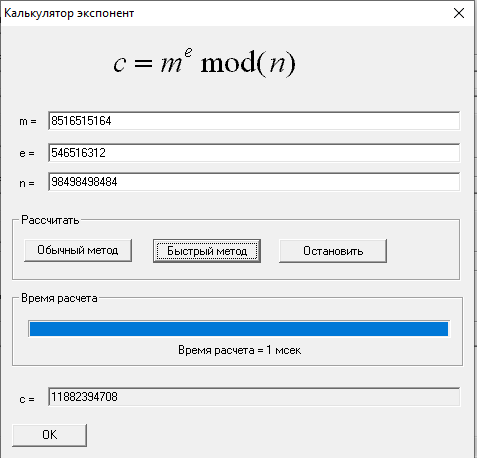
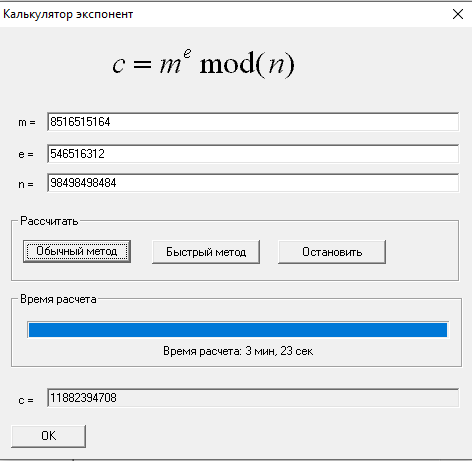
82 = 64 mod9 = 1

84 = 1 mod9 = 1

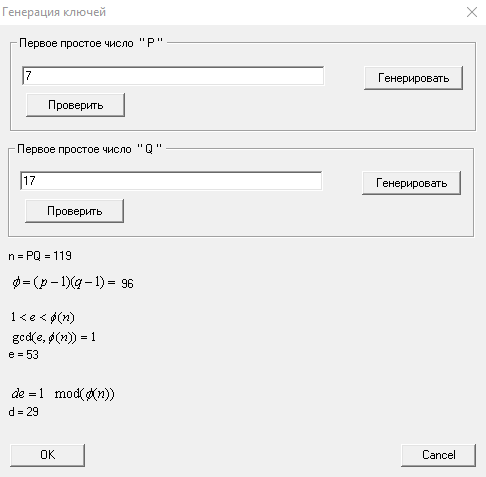
88 = 1 mod9 = 1

815 = 8 mod9 = 8

3. Рассчитать степени произвольных больших чисел по модулю прямым методом и быстрым методом. Убедиться в эффективности быстрого метода.

**

4. Выбрав в главном меню опцию “Генерировать ключи”, задать малые простые числа *p,q* и сгенерировать ключи для РША. Проверить правильность вычислений расчетами на бумаге.

**

*НОД(11,96) = 1*

*96 = 11\*8 + 8*

*11 = 8 + 3*

*8 = 3\*2 +2*

*3 = 2 + 1*

*2 = 1\*2 +0*

*d = 11-1mod96 = 35*

*1 = (z1\*11 + z2\*96) mod96 = (z1\*11)mod96*

*1 = 3 – 2*

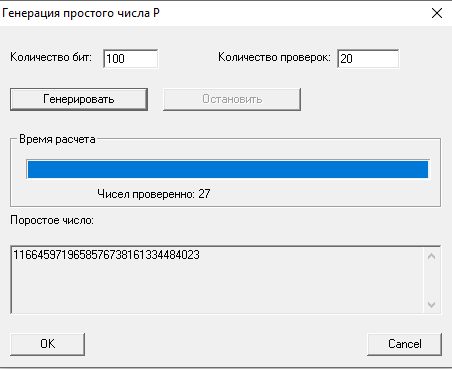
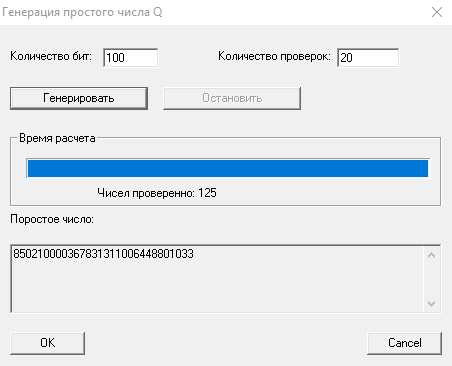
*2 = 8 – 3\*2*

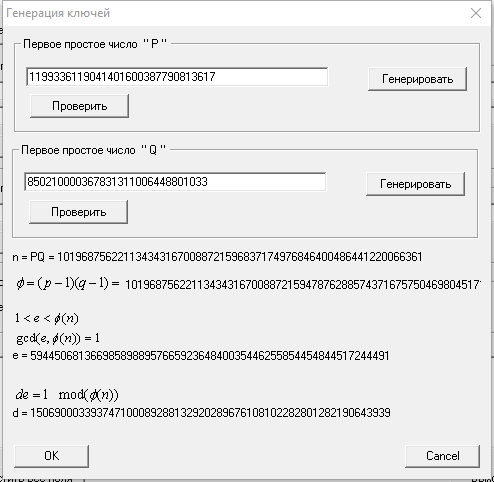
*3 = 11 – 8*

*8 = 96 – 11\*8*

*1 = 11 – 8 – 8 + 3\*2 = 11 – 96 + 11\*8 – 96 + 11\*8 + 11\*2 – 2\*96 + 11\*16 =35\*11 – 4\*96*

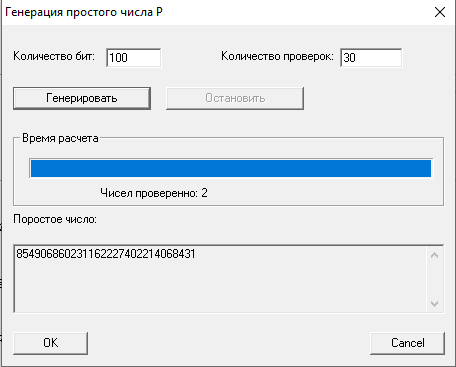
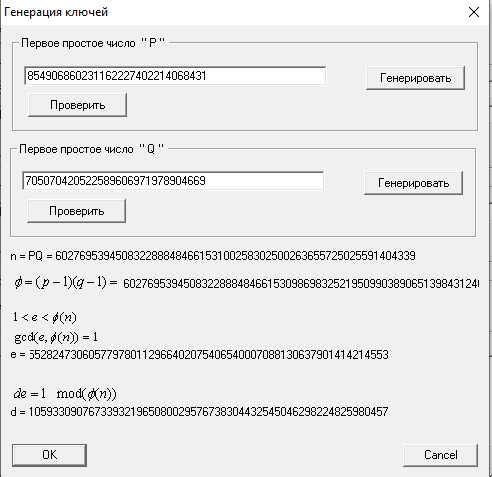
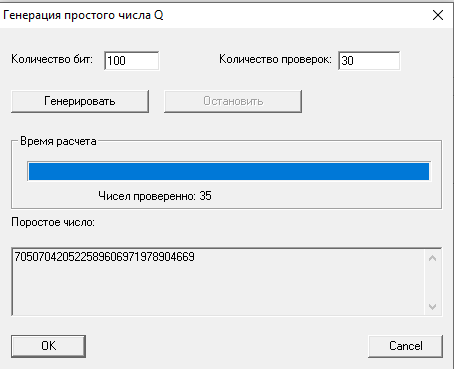
*z1 = d = 35*

5. Произвести случайное генерирование простых чисел *p,q* разрядности по 100 бит каждое, задаваясь количеством проверок *n=20* и используя опцию “Генерировать” в подменю “Генерация ключей”*.* Выполнить затем генерацию ключей для системы РША.



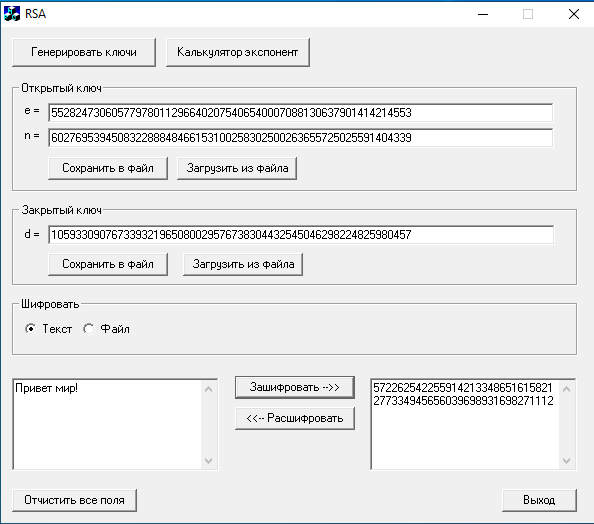
6. Повторить п.5 для *n=30.* Рассчитать вероятность ошибочного выполнения тестирования простых чисел для этого и для предыдущего пунктов.

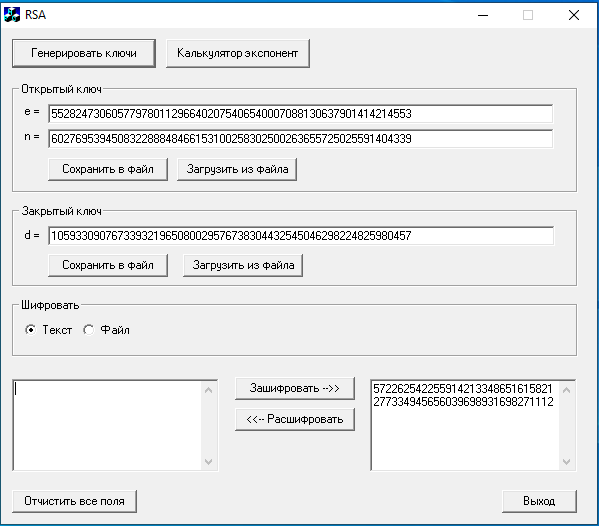
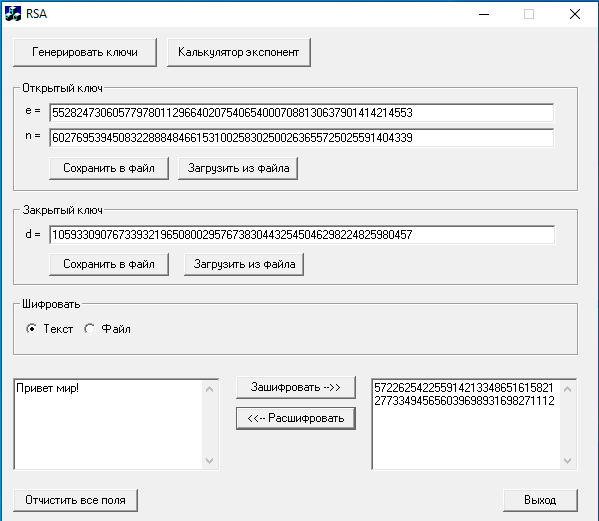
*В пункте 5 вероятность ошибки:*

**

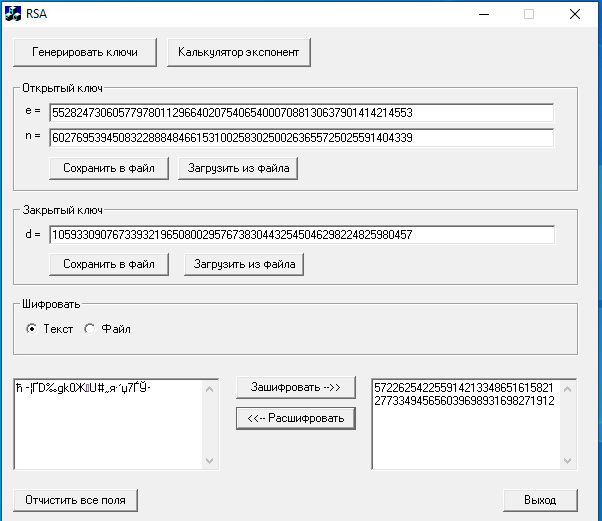
*Вероятность ошибки для пункта 6:*

7. Перейдя в основное меню, произвести шифрование короткого смыслового текста на русском или английском языке, введенного в левое поле с использованием опции “Зашифровать”. Наблюдать появление криптограммы в правом поле.

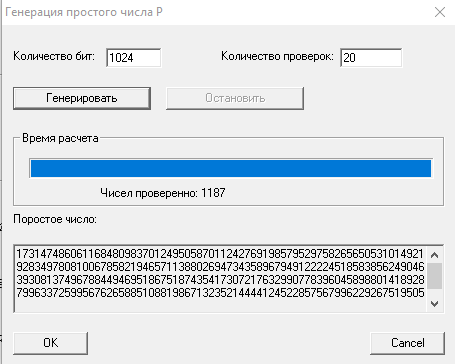
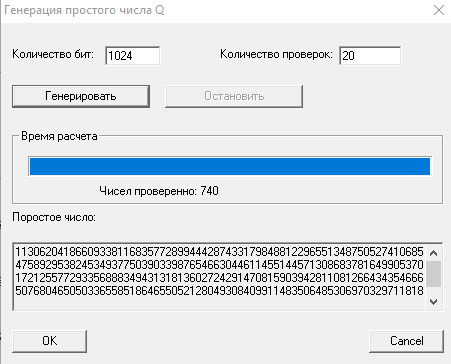


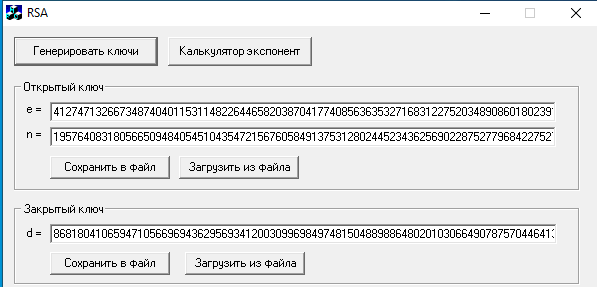
8. Удалить открытый текст в левом поле и произвести дешифрование криптограммы, содержащейся в правом поле, используя опцию “Дешифровать”. Проверить правильность дешифрования.

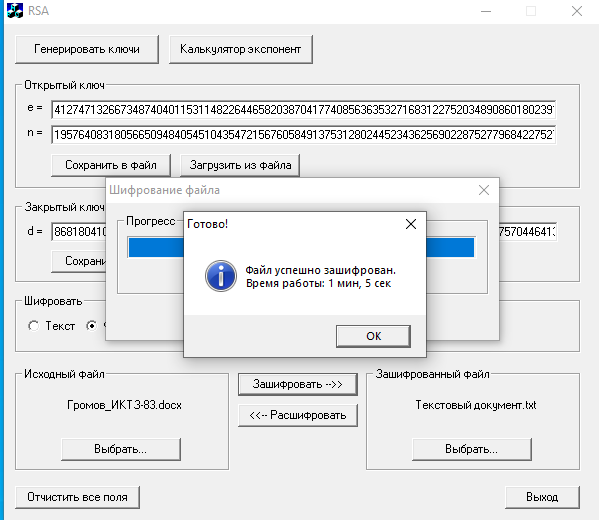
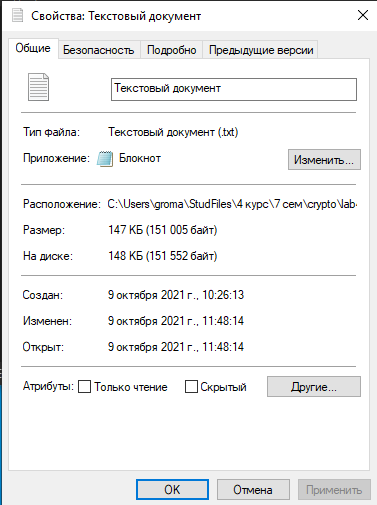
9. Повторить п.8 после изменения одной из цифр криптограммы.

**

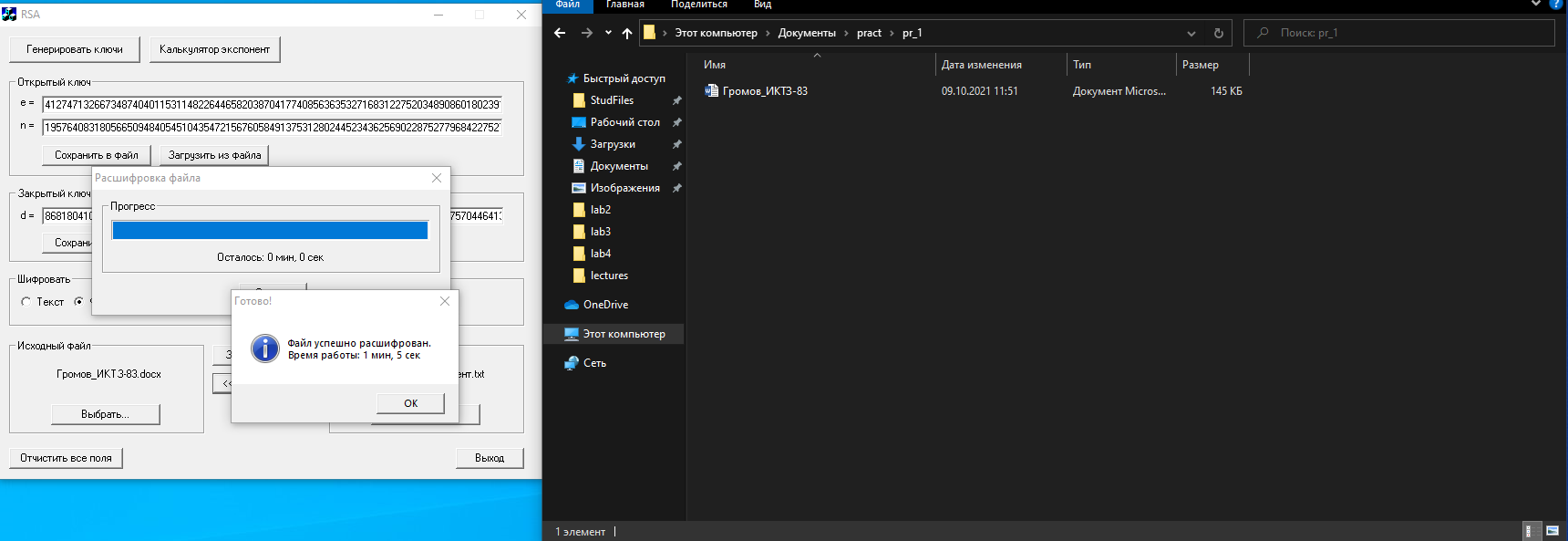
10. Повторить п. 6, выбирая разрядность каждого из простых чисел 1024 бита.

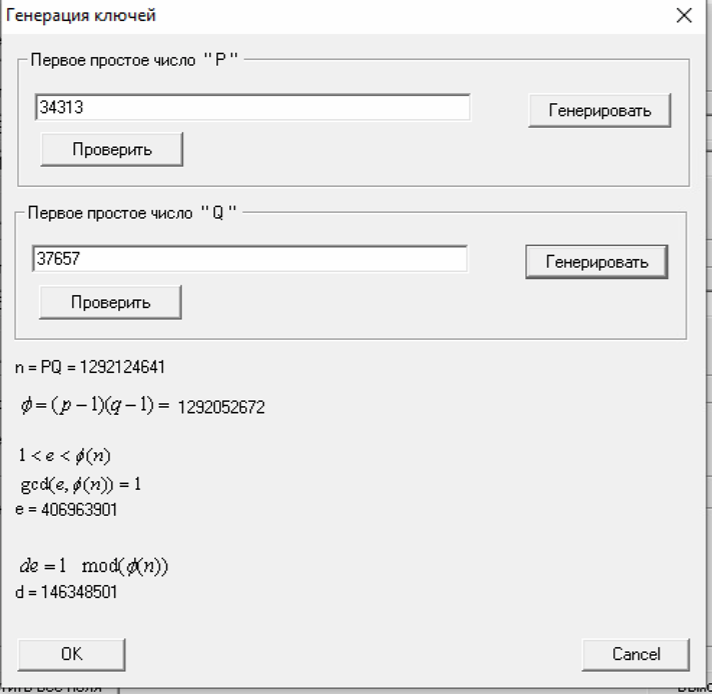


11. Вернувшись в главное меню произвести шифрование произвольного файла большого объема (не менее 100кбайт), фиксируя при этом время шифрования.

12. Произвести дешифрование криптограммы, полученной в п.11 и помещенной в специальный файл, фиксируя при этом время дешифрования.



13. По данным п.11,12 рассчитать скорость шифрования и дешифрования при данной простейшей программной реализации безопасной криптосистемы РША.

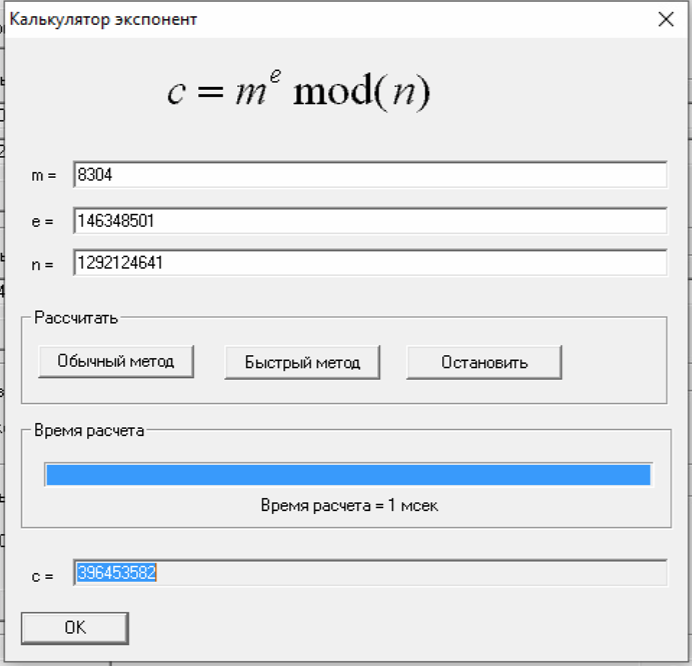
14. Повторить п. 5, выбирая разрядность каждого из простых чисел 16 бит.

e = 406963901

n = 1292124641

d = 146348501

15. Используя подпрограмму «Калькулятор экспонент» Сгенерировать цифровую подпись открытого текста *m* в виде ГГNN (где ГГ- номер группы, NN - ваш номер по журналу)без использования хэш-функции (аналогично шифрованию, но с закрытым ключом) s*=mdmodn*. Подписью будет пара чисел *(m,s)*.

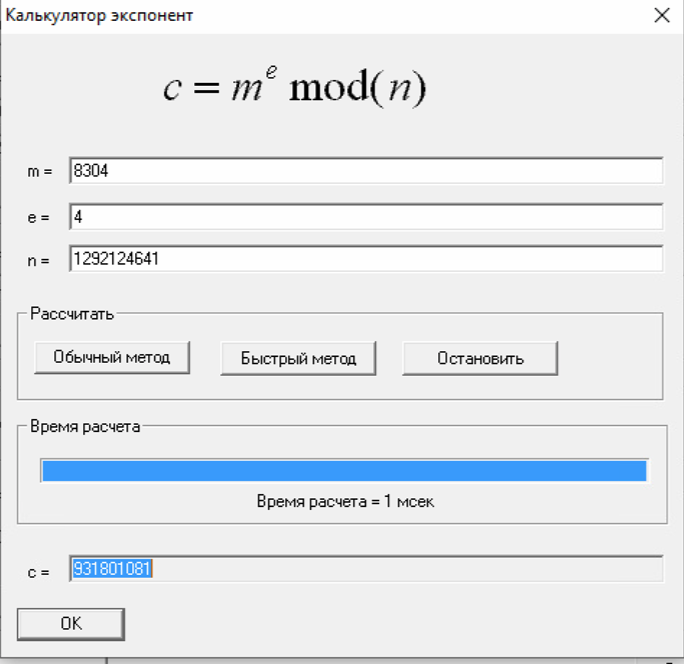


m = 8304

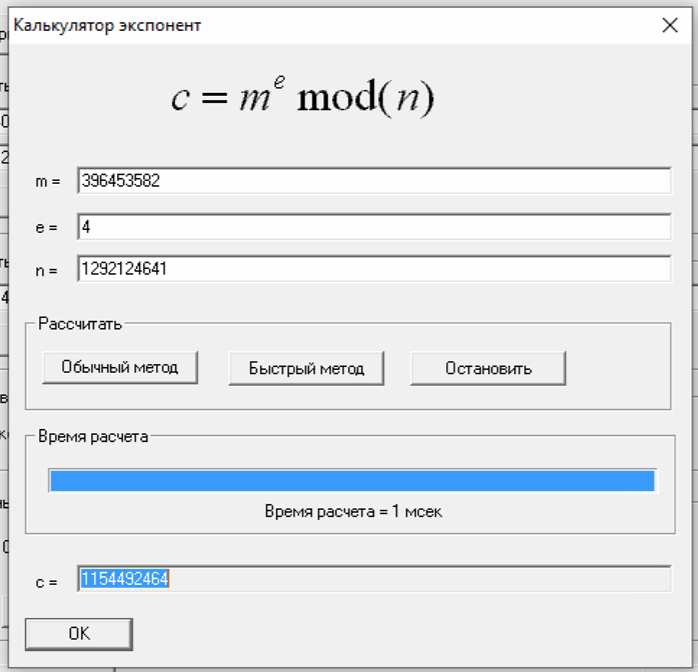
s = 396453582

16. Создать подделку цифровой подписи путем возведения *m* и *s* в степень с произвольным показателем *l*: *m’=mlmodn, s’=slmodn*. (Использовать туже программу вычисления экспонент, *l=4*).

m' = 931801081



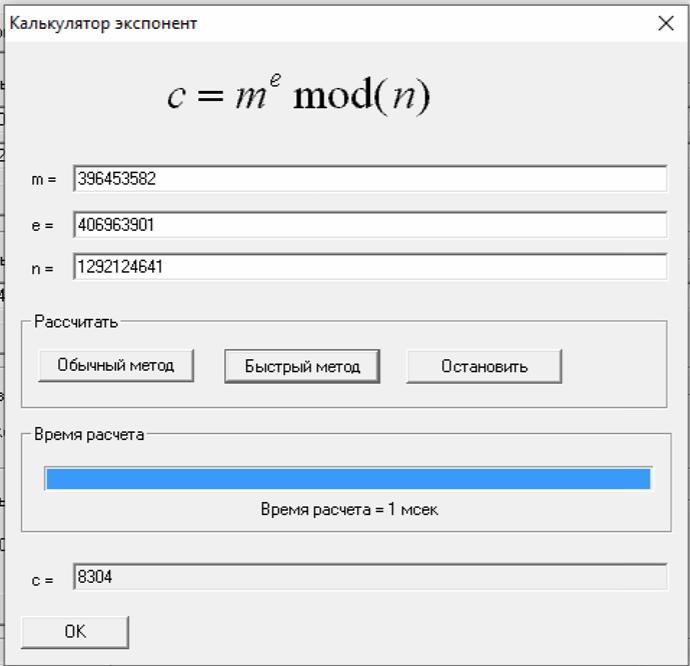
s' = 1154492464



17. Провести верификацию цифровой подписи в обоих случаях:

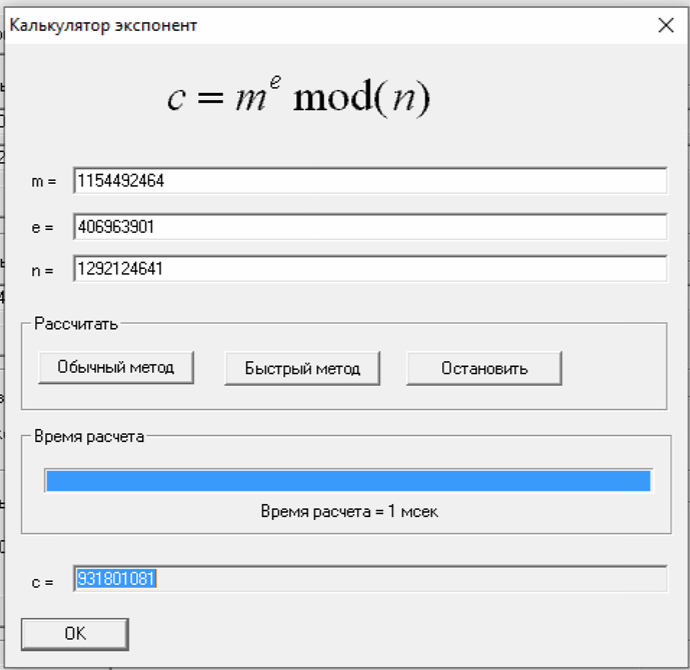
найти *m1=se(mod n)* ипроверить равенство *m1=m,*

*m1 = 8304 = m*

**

найти *m2=s’e(mod n)* ипроверить равенство *m2=m’*.

m2 = 931801081

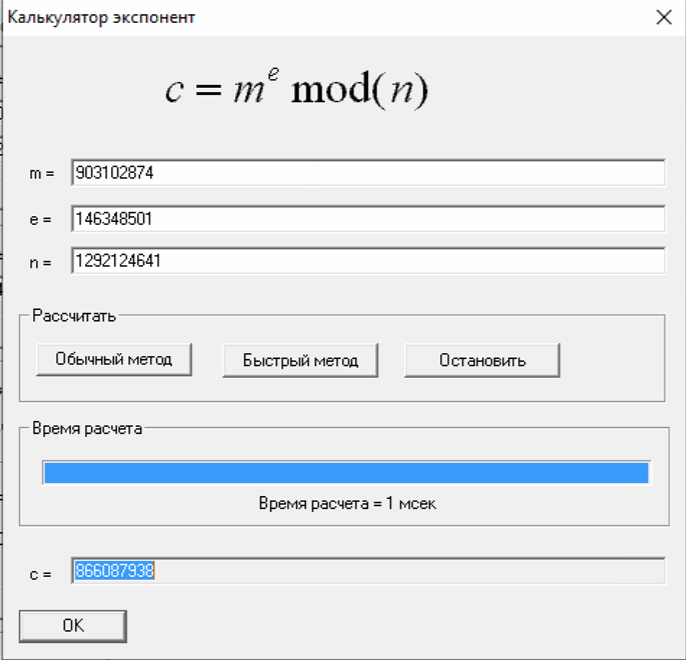


Cделать выводы о безопасности цифровой подписи открытого текста и его целостности без применения хэширования.

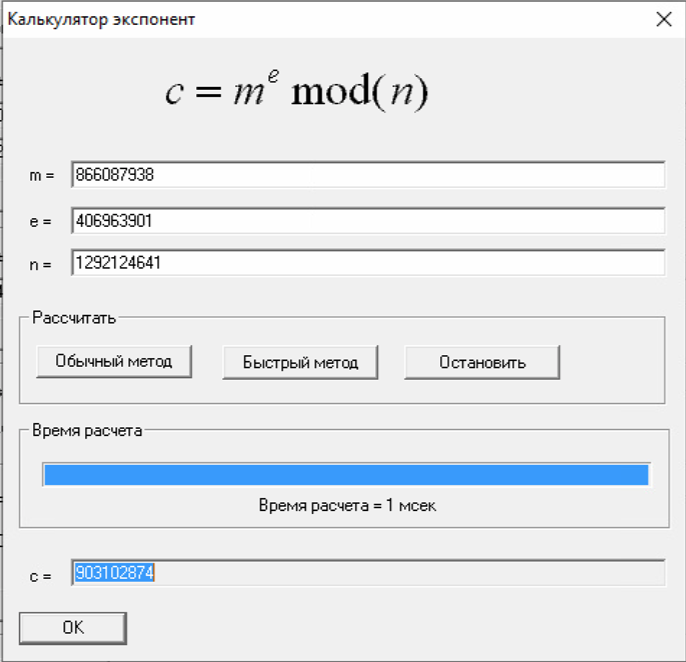
18. Получить хэш-образ открытого текста *h(m)*, используя следующий упрощенный алгоритм: сложить номера всех символов сообщения. Результат умножить на d с приведением по *modn*.

h(m) = (8+3+0+4)\*146348501 mod 1292124641=903102874

19. Получить подпись для хэш-образа *s=hd(m)modn*, используя алгоритм модульного возведения в степень. Подписью будет пара *(m,s)*. Верифицировать подпись проверкой сравнения *h(m)=semodn.*

**

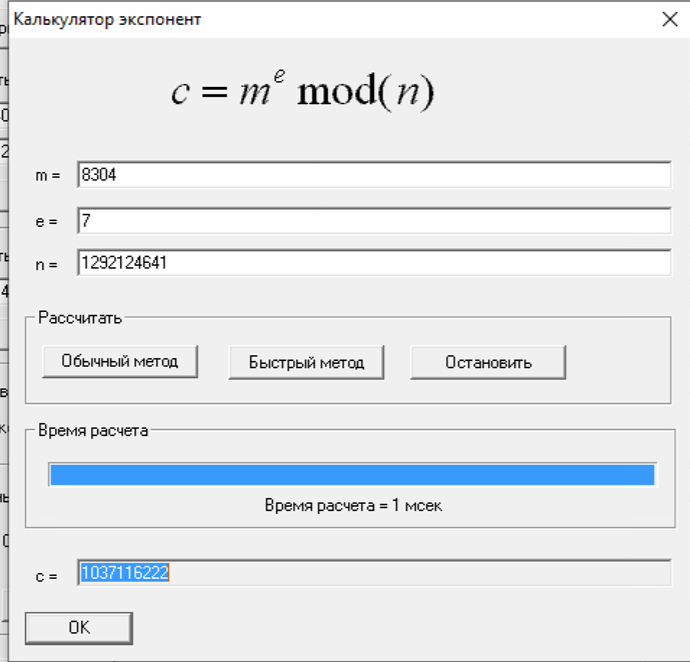
*s =* *866087938*

**

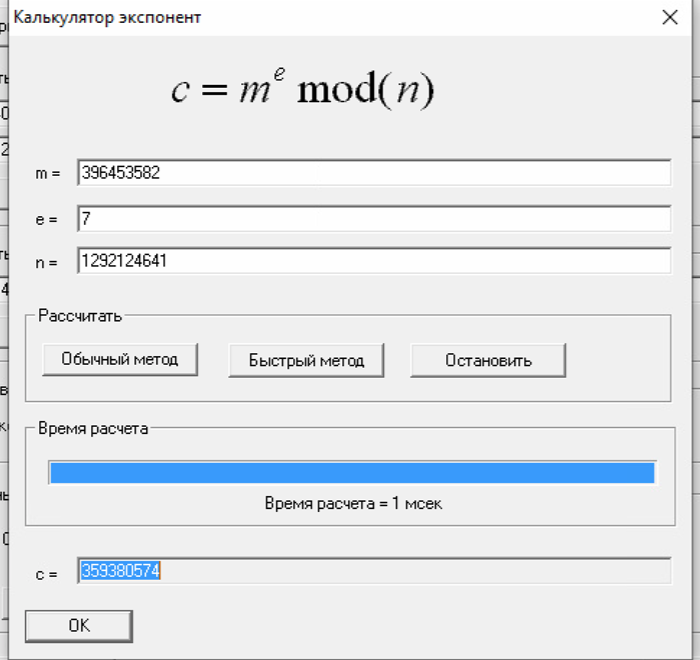
*h(m) = 903102874 - верно*

20. Создать подделку цифровой подписи путем возведения *m* и *s* в степень с произвольным показателем *l*: *m’=mlmodn, s’=slmodn*. (*l* выбирать в диапазоне 6-10).

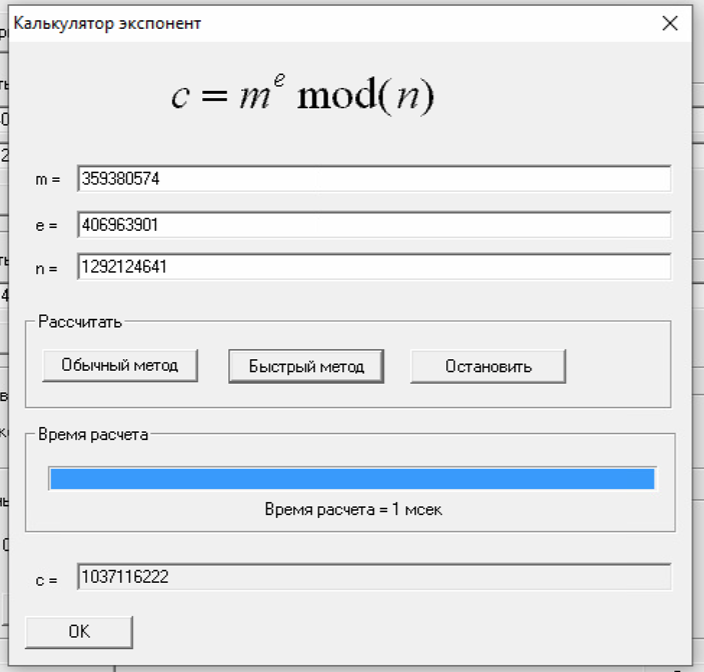
m' = 1037116222



s’ = 359380574

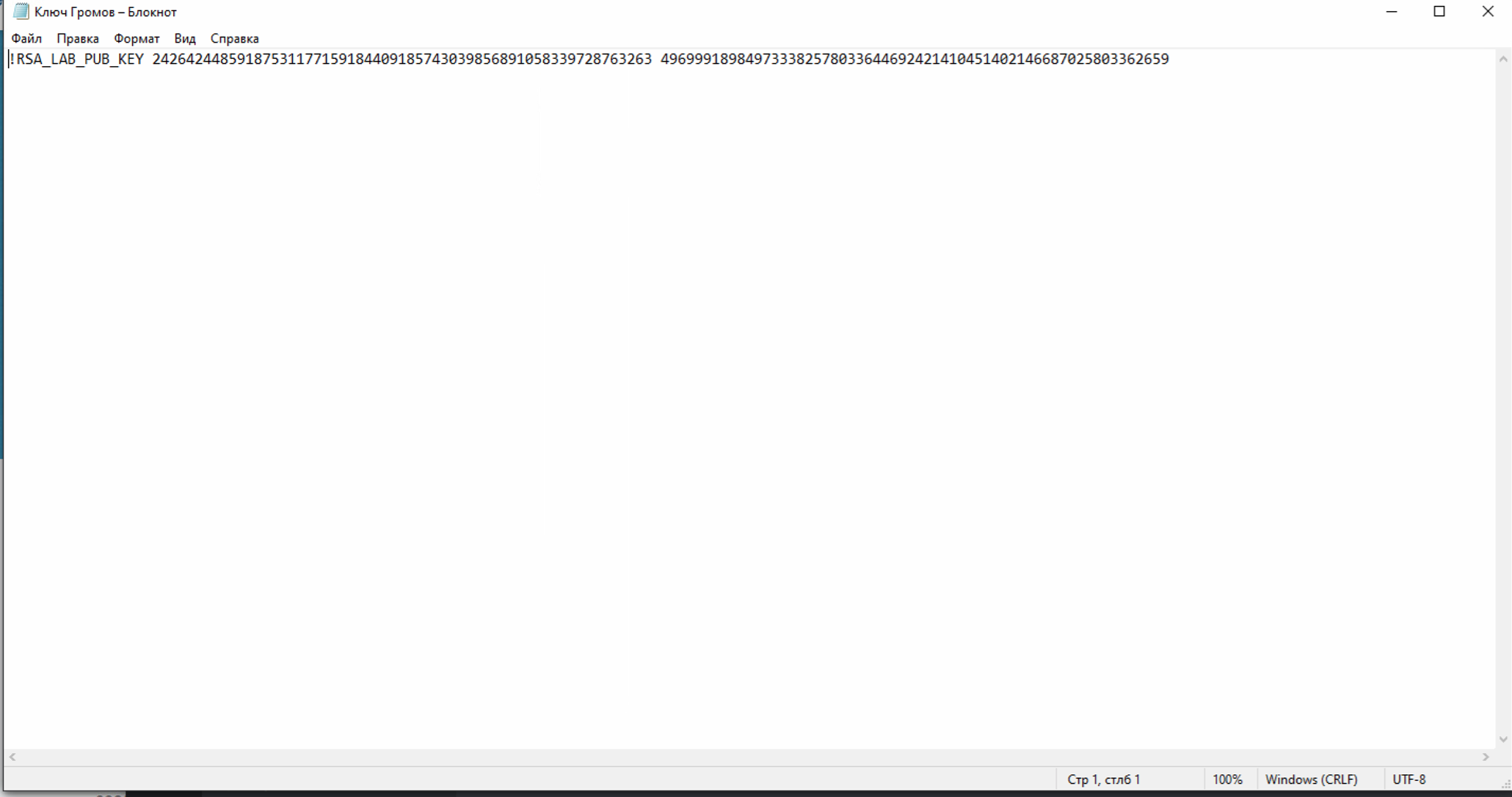


21. Верифицировать подделанную подпись путем проверки сравнения *h(m’)* = *s’emodn.*

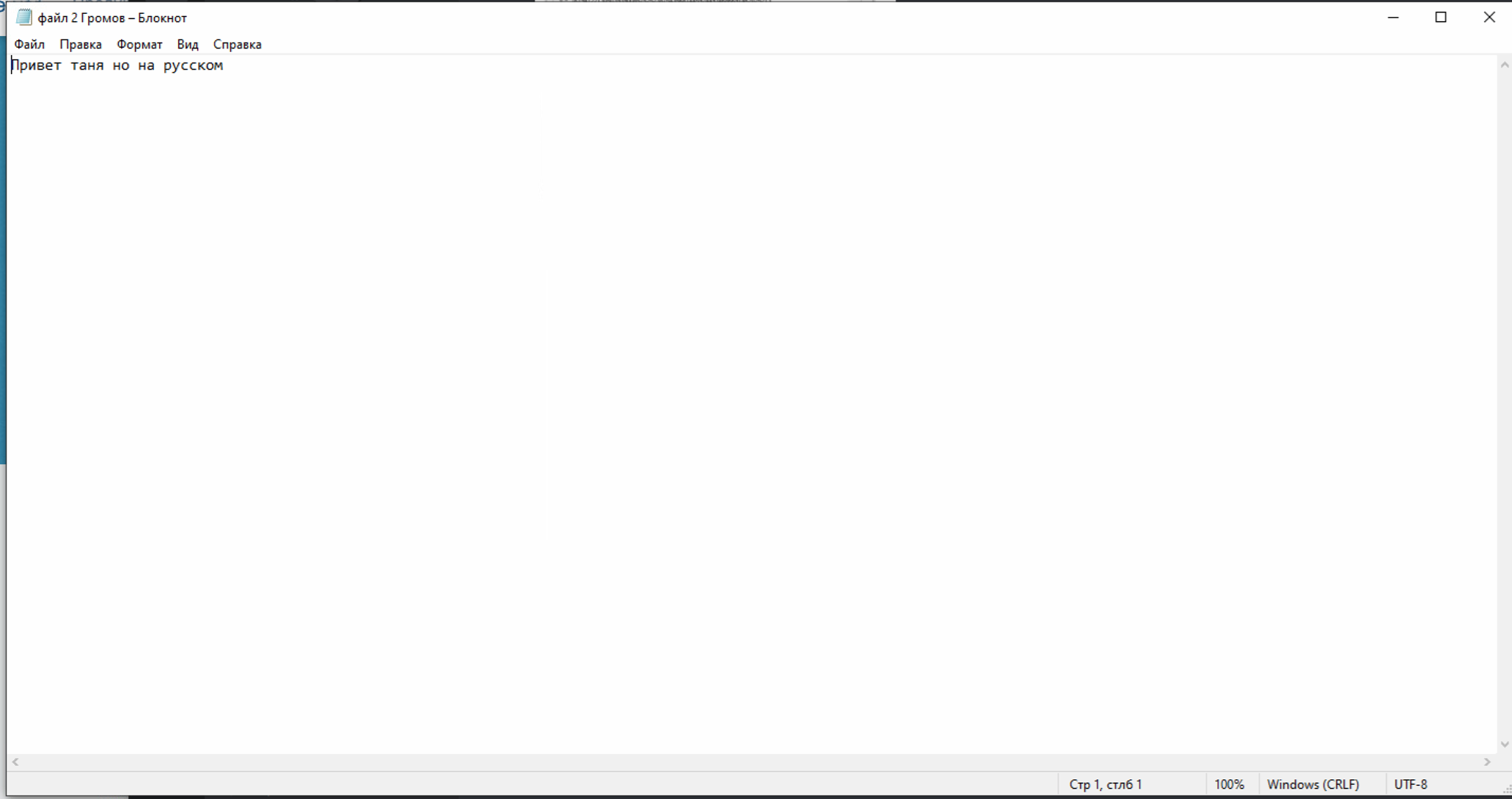
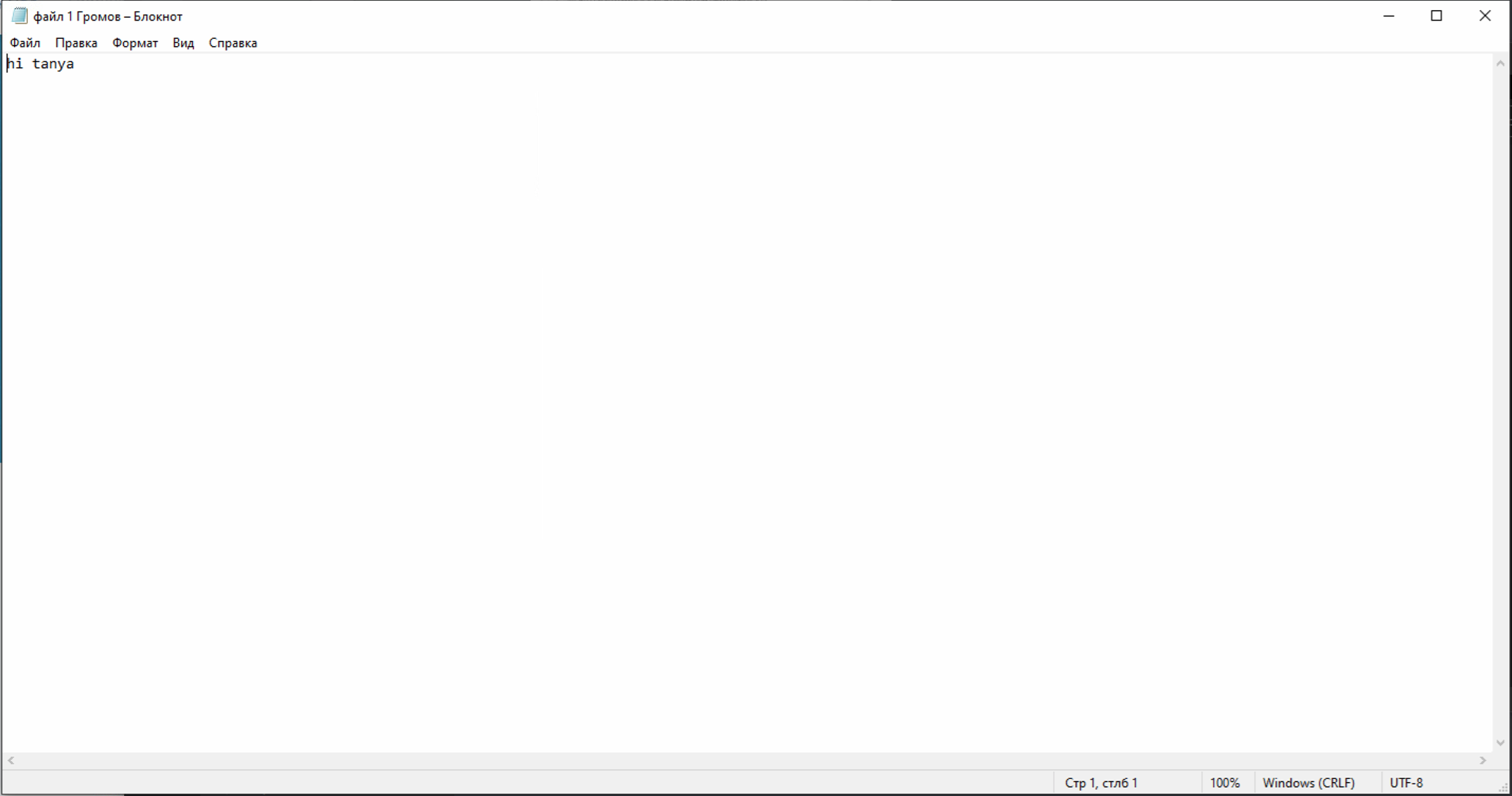
**

*h(m') = 1037116222 != h(m) –* таким образом, использование хэш-функции повышает безопасность цифровой подписи.

22. Провести генерацию ключей в криптосистеме РША (см. п 8). Открытый ключ *(N, e)* сохранить, как файл, использую программу Блокнот. Дать файлу имя: «Ключ Фамилия», где «Фамилия» - фамилия владельца закрытого ключа *d.* Ключ поместить в общедоступную папку «№ группы». Паку создает один из студентов и всем сообщает ее местонахождение.



23. Подготовить два коротких текстовых сообщения в программе Блокнот. Файлам, содержащим сообщения дать имена «Сообщение1 Фамилия» и «Сообщение2 Фамилия».



24. Первое сообщение зашифровать на открытом ключе получателя, предварительно экспортировав его из файла, содержащего этот ключ. Зашифрованному сообщению (файлу) дать имя, например, «Иванов-Петрову». Поместить в файл в папку группы.

25. Второе сообщение подписать своим закрытым ключом. Подписанному сообщению дать имя, например, «подпись Иванова». Сообщение 2 (файл) и подпись к нему поместить в папку группы.

26. Получателю сообщений на основе своего закрытого ключа расшифровать зашифрованное первое сообщение и проверить подпись второго сообщения, используя свой закрытый ключ.

