МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра информационной безопасности

ОТЧЕТ

по лабораторной работе №7
по дисциплине «Криптография и защита информации»
Тема: Изучение асимметричных протоколов и шифров

Студент гр. 9381 _____ Колованов Р.А.

Преподаватель _____ Племянников А.К.

Санкт-Петербург
2022

Цель работы.

Исследовать протокол Диффи-Хеллмана, шифр RSA и получить практические навыки работы с ними, в том числе с использованием приложения Cryptool 1 и 2.

Основные теоретические положения.

Протокол Диффи-Хеллмана.

Протокол Диффи-Хеллмана является первым из опубликованных криптопреобразований на основе открытых ключей. Поэтому этот протокол ещё называют обменом ключами по схеме Диффи-Хеллмана.

Цель протокола – обеспечить двум пользователям возможность получения симметричного секретного ключа путем обмена данными по незащищенному каналу связи.

Протокол Диффи-Хеллмана состоит из следующих операций (рисунок 7.1):

1. Устанавливаются открытые параметры p, g:
    a. p – большое простое число порядка 300 десятичных цифр (1024 бита),
    b. g – первообразный корень по модулю p.
2. Каждая из сторон генерирует закрытый ключ - большое число x и y соответственно;
3. На каждой стороне вычисляется открытый ключ:
    a. $R_1 = g^x \bmod p$,
    b. $R_2 = g^y \bmod p$.
4. Стороны обмениваются открытыми ключами и вычисляют симметричный общий ключ K:

$$K = R_2{}^x \bmod p = R_1{}^y \bmod p$$

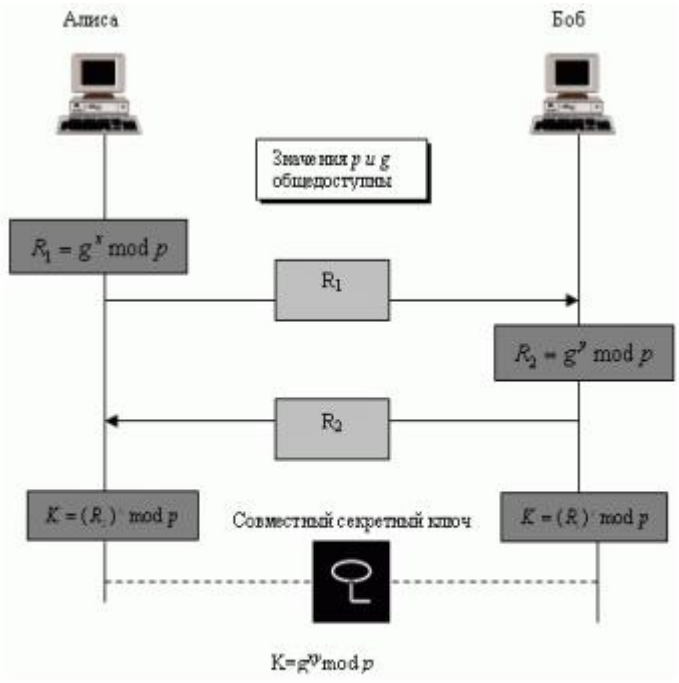


Рисунок 7.1

Шифр RSA.

Алгоритм RSA представляет собой асимметричный блочный шифр, в котором и открытый, и шифрованный текст представляются целыми числами из диапазона от 0 до *n-1* для некоторого *n*.

Алгоритм шифрования RSA состоит из следующих операций (рисунок 7.2):

1. Вычисление ключей:
    a. Генерация двух больших простых чисел p и q (p и q держаться в секрете);
    b. Вычисление $n = p * q$;
    c. Выбор произвольного e ($e < n$), взаимно простого с $\varphi(n)$ – функцией Эйлера;
    d. Вычисление d: $e * d = 1 \ mod \ \varphi(n)$;
    e. Числа (e, n) – открытый ключ, d – закрытый ключ, p и q уничтожаются.
2. Шифрование:

a. Открытый текст разбивается на блоки m_i: $m_i < n$;
b. Каждый блок открытого текста преобразуем в шифротекст по формуле:

$$c_i = m_i^e \bmod n$$

3. Расшифровка:
   a. Шифротекст представляется блоками c_i: $c_i < n$;
   b. Каждый блок шифротекста преобразуется в открытый текст по формуле:

$$m_i = c_i^d \bmod n$$

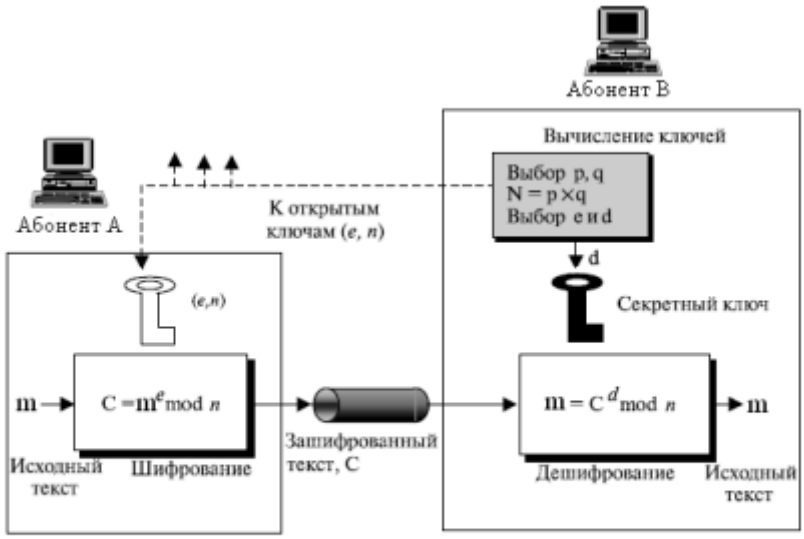


Рисунок 7.2 Протокол шифрования RSA

Имитация атаки на гибридную криптосистему.

Модель гибридной криптосистемы, асимметричная составляющая которой использует асимметричный шифр (например RSA) представлена на рисунке 7.3.

Шифрование в рамках модели осуществляется следующим образом:

1. Сообщение шифруется симметричным секретным ключом;
2. Секретный ключ шифруется открытым ключом получателя;
3. Зашифрованное сообщение и ключ объединяются в цифровой конверт, который отправляется получателю;

4. Получатель сначала расшифровывает секретный ключ своим закрытым ключом, а затем расшифровывает этим секретным ключом шифровку сообщения.

Атака на модель гибридной криптосистемы основана на том, что злоумышленник сначала перехватывает цифровой конверт, содержащий зашифрованное сообщение и зашифрованный секретный ключ, затем, модифицирует шифровку ключа из конверта и побитово восстанавливает зашифрованный секретный ключ, анализируя положительные и отрицательные ответы сервера.

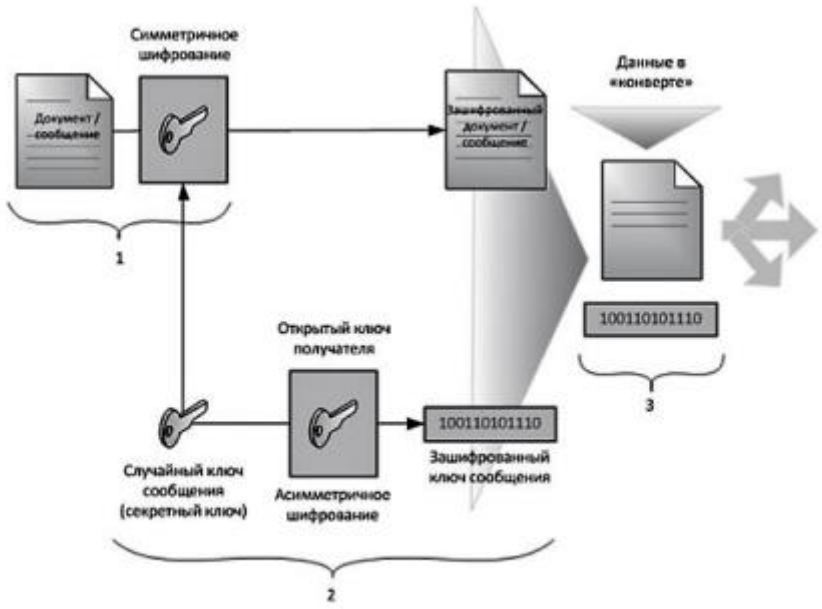


Рисунок 7.3

Ход работы.

Протокол Диффи-Хеллмана.

Задание.

1. Запустите утилиту Indiv.Procedures -> Protocols -> Diffie-Hellman demonstration и установите все опции информирования в ON;
2. Выполните последовательно все шаги протокола;
3. Сохраните лог-файл протокола для отчета (пиктограмма с изображением ключа);
4. Используйте полученный общий ключ для зашифровки и расшифровки произвольного сообщения. Шифр выберите самостоятельно.

Основные параметры протокола.

Основные параметры протокола Диффи-Хеллмана:

- (p, g, R_1) и (p, g, R_2) – открытые ключи сторон;
- x, y – закрытые ключи сторон;
- $R_2^{\,x} \bmod p$ и $R_1^{\,y} \bmod p$ – односторонние функции с секретом (TOWF).

Математическая модель протокола Диффи-Хеллмана:

- p – большое простое число порядка 300 десятичных цифр (1024 бита);
- g – порождающий элемент циклической группы (генератор) порядка p, для которого справедливо: $g \bmod p$, $g^2 \bmod p$, $g^3 \bmod p$, … , $g^{p-1} \bmod p$ являются различными целыми из [1, p - 1];
- x, y – большие случайные числа такие, что $0 < x < p - 1$, $0 < y < p - 1$;
- Поскольку:

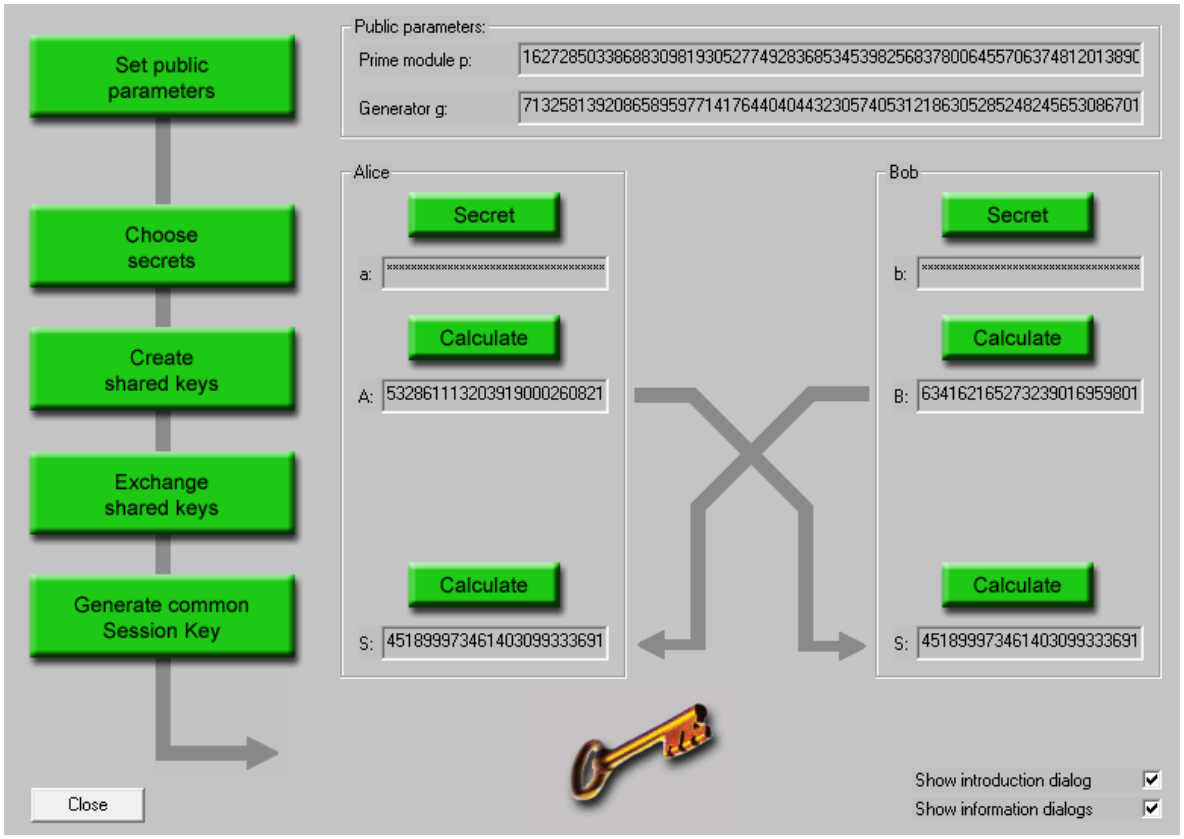
$$R_2^{\,x} \bmod p = \left(g^{y} \bmod p\right)^{x} \bmod p = g^{xy} \bmod p$$

$$R_1^{\,y} \bmod p = \left(g^x \bmod p\right)^y \bmod p = g^{xy} \bmod p$$

- Стороны фактически создают материал для симметричного ключа сеанса без Центра распределения ключей (KDC).

Скриншот схемы протокола, реализованной в CrypTool.

При помощи утилиты Indiv.Procedures -> Protocols -> Diffie-Hellman demonstration была рассмотрены схема работы протокола Диффи-Хеллмана представлена на следующем рисунке:



Лог-файл работы протокола Диффи-Хеллмана в *CrypTool 1* представлен в Приложении А.

Таблица соответствия схемы протокола (CrypTool) и параметров протокола.

Сопоставим параметры из приведенной выше схемы протокола, реализованного в *CrypTool*, с параметрами протокола Диффи-Хеллмана:

Параметр протокола	Параметр из схемы протокола в CrypTool	Описание
p	p	Простое число порядка 300 десятичных цифр.
g	g	Первообразный корень по модулю p.
x	a	Закрытый ключ Алисы.
y	b	Закрытый ключ Боба.
R_1	A	Открытый ключ Алисы.
R_2	B	Открытый ключ Боба.
K	S	Общий симметричный ключ.

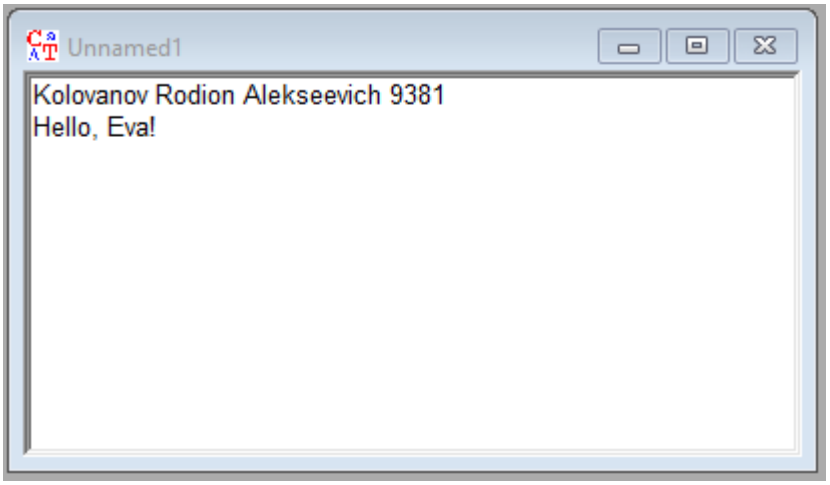
Шифровка и расшифровка произвольного сообщения.

Используя полученный общий ключ из демонстрации протокола Диффи-Хеллмана, было зашифровано и расшифровано произвольное сообщение. В качестве шифра был выбран шифр AES.

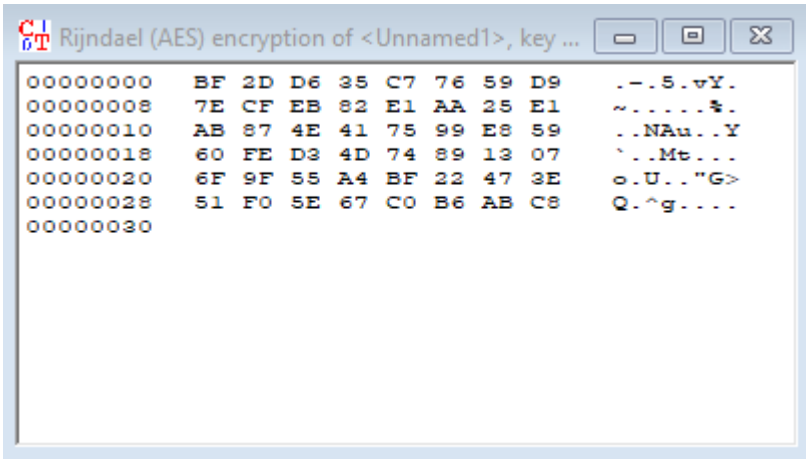
Общий ключ K:

45189997346140309933369168087575158857529838206572205305769286638838484283893 (63 E8 A1 D1 0E FB 17 B7 54 B4 01 EC D2 8B 88 08 60 33 7F 34 69 35 0D 2C 0B 20 A5 15 1F B3 B1 F5).

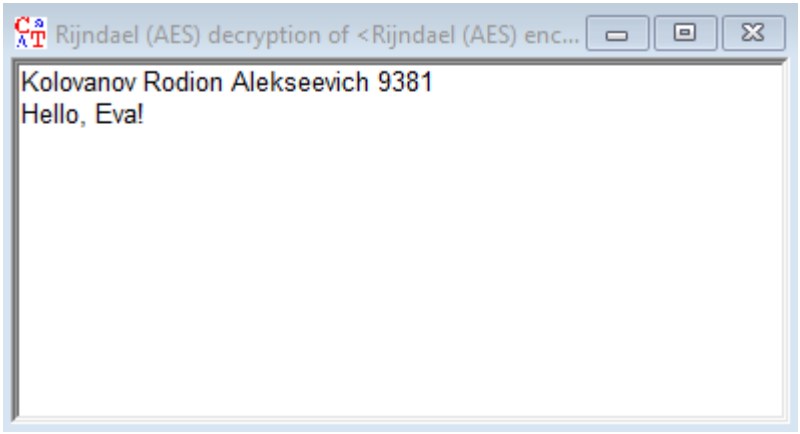
Исходное сообщение:



Зашифрованное сообщение:



Расшифрованное сообщение:



Шифр RSA.

Задание.

1. Запустите утилиту Indiv.Procedures -> RSACryptisystem -> RSA Demonstration;
2. Задайте в качестве обрабатываемого сообщения свою Ф.И.О;
3. Сгенерируйте открытый и закрытый ключи;
4. Зашифруйте сообщение. Сохраните скриншот результата;
5. Расшифруйте сообщение. Сохраните скриншот результата;
6. Убедитесь, что расшифрование произошло корректно.

Обобщенная схема шифра.

Шифр RSA представляет собой блочный алгоритм шифрования, где зашифрованные и незашифрованные данные должны быть представлены в виде целых чисел между 0 и n - *1*.

Шифр RSA базируется на следующих двух фактах из теории чисел:

- Задача проверки числа на простоту является сравнительно легкой;
- Задача разложения чисел вида $n = p * q$ (p и q – простые числа) на множители является очень трудной, если мы знаем только n, а p и q – большие числа (задача факторизации).

Генерация ключей:

- Выбираются два больших простых числа p и q;
- Вычисляется $n = p * q$;
- Выбирается произвольное число e ($e < n$), взаимно простое с $(p - 1) \times (q - 1)$;
- Вычисляется d, такое, что $e \times d \equiv 1 \ mod \ (p - 1) \times (q - 1)$ решением в целых числах уравнения относительно d и y:

$$e \times d + (p-1) \times (q-1) \times y = НОД(e, (p-1) * (q-1)) = 1$$

- Пара чисел (e, n) объявляются открытым ключом;
- Закрытым ключом выбирается d (p и q нужно уничтожить).

Зашифрование:

- Открытый текст разбивается на блоки m_i размером $k \leq [log_2 n]$ бит. Блоки интерпретируются, как числа из диапазона (0, 2^k - 1);
- Ключ шифрации – пара чисел (e, n) (открытый ключ);
- Каждый блок открытого текста преобразуется в шифротекст по формуле:

$$c_i = (m_i^e) \bmod n$$

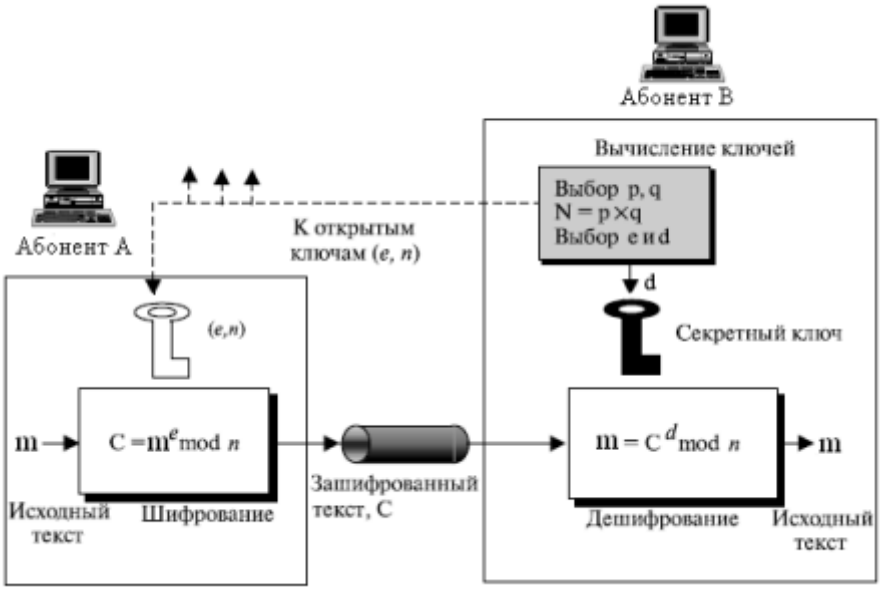
Расшифрование:

- Ключ для расшифровки сообщения – d (закрытый ключ);
- Блок шифротекста преобразуется в открытый текст по формуле:

$$m_i = (c_i^{\,d}) \bmod n$$

- Доказательство основано на теореме Эйлера. Если n представимо в виде произведения простых чисел p и q, то для x (взаимно простого с n) справедливо:

$$\left(x^{(p-1)\times(q-1)}\right) \bmod n = 1$$



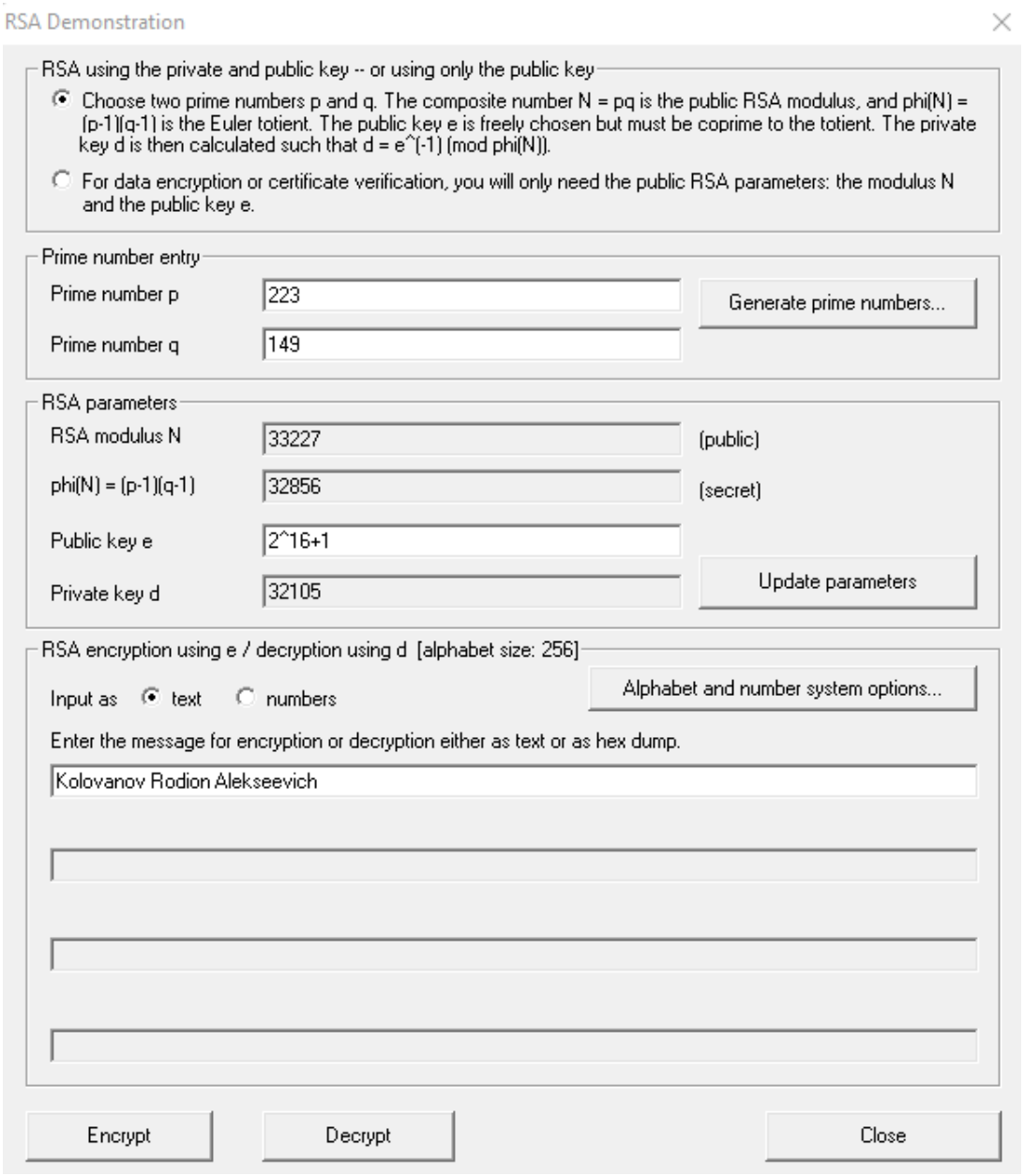
Шифровка и расшифровка сообщения.

При помощи утилиты Indiv.Procedures -> RSACryptisystem-> RSADemonstration было осуществлено шифрование и расшифрование исходного сообщения «Kolovanov Rodion Alekseevich». Для начала были сгенерированы открытый и закрытый ключи.

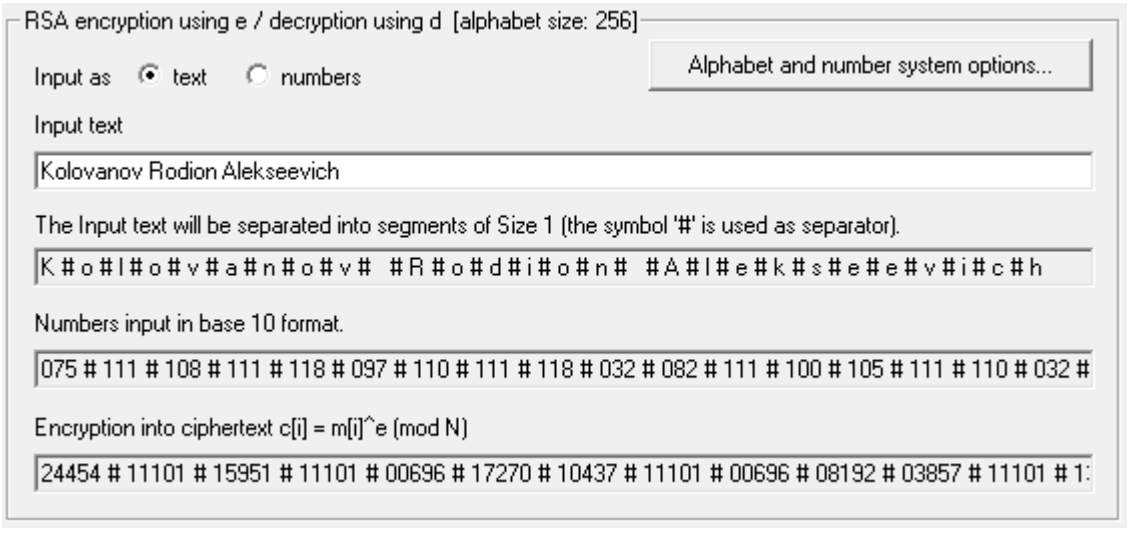
Открытый ключ: $e = 2^{16} + 1$ и $n = 33227$.

Закрытый ключ: $d = 32105$.

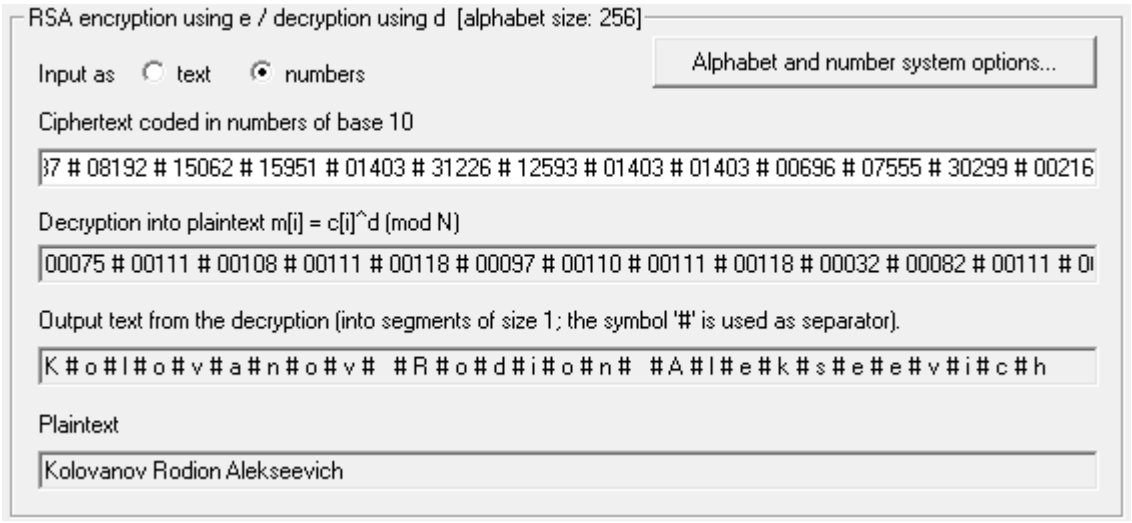
Скриншот результата генерации ключей:



Скриншот результата шифрации:



Скриншот результата расшифровки:



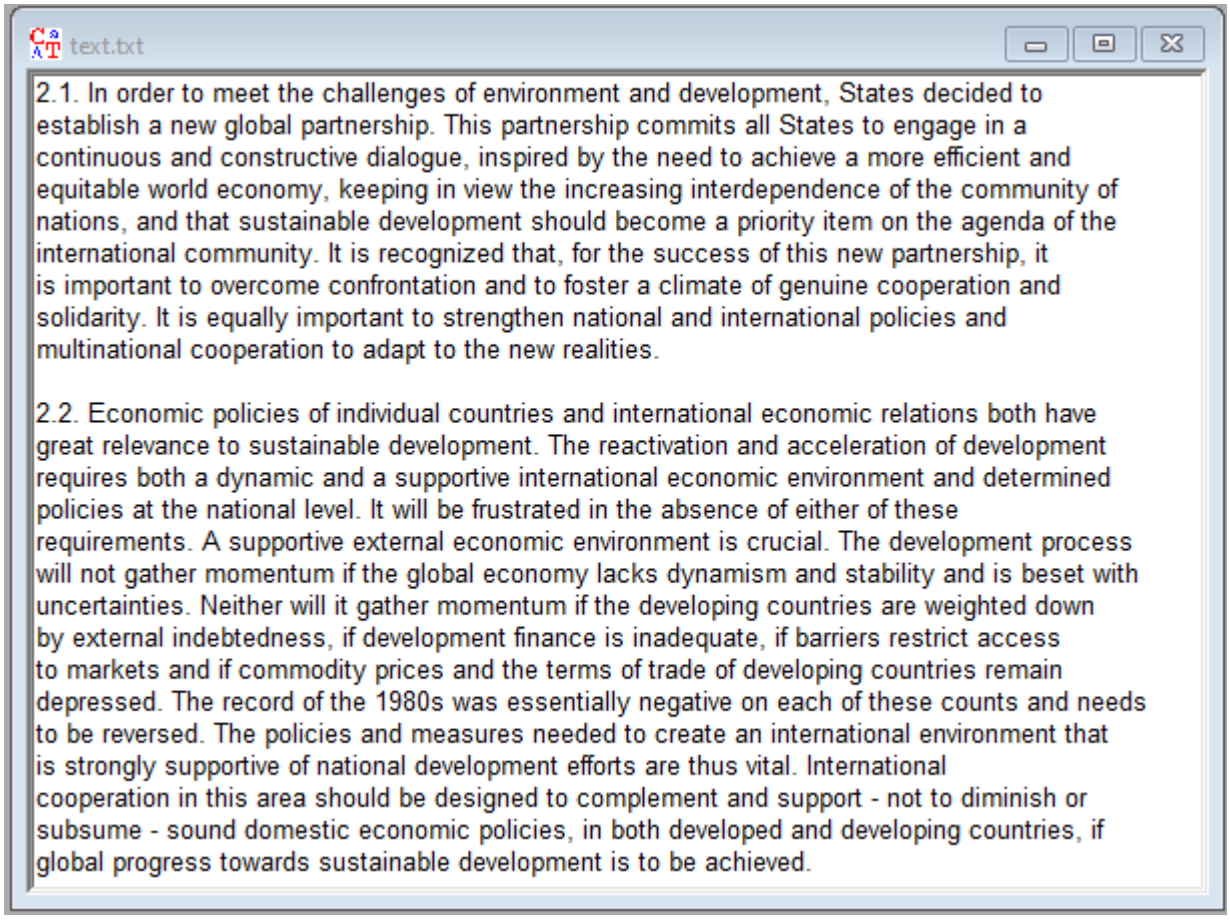
Исследование шифра RSA.

Задание.

1. Выбрать текст на английском языке (не менее 1000 знаков) и сохранить в файле формата *.txt;
2. Сгенерировать пары асимметричных RSA-ключей утилитой Digital Signatures -> PKI -> Generate/Import Keys с различными длинами (4 варианта);
3. Зашифровать текст (примерно 1000 символов) различными открытыми ключами. Зафиксировать время зашифровки;
4. Расшифровать текст различными закрытыми ключами. Зафиксировать время зашифровки;
5. Проверить корректность расшифровки. Зафиксировать скриншоты результата.

Исследование времени работы шифра RSA при различных длинах ключей.

В качестве исходного текста был взят следующий текст (размер текста составляет 2148 символов):



Далее при помощи утилиты Digital Signatures -> PKI -> Generate/Import Keys были сгенерированы пары ассиметричных RSA-ключей длиной 512, 768, 1024, 2048 бит:

Длина ключа	Экспонента	Модуль
512	65537	13402140501835986529348804086681664088829601975528719947844144202637396257687344344639478864046530502837705959293098424853685064842214466402682833095015179
768	65537	153237482207998694263023857283727176158832593789191163676725361623347742443039465635063370764201056085774634150064609331461773699800403717494658774030227656888378532998818247381218836507491221019558 31

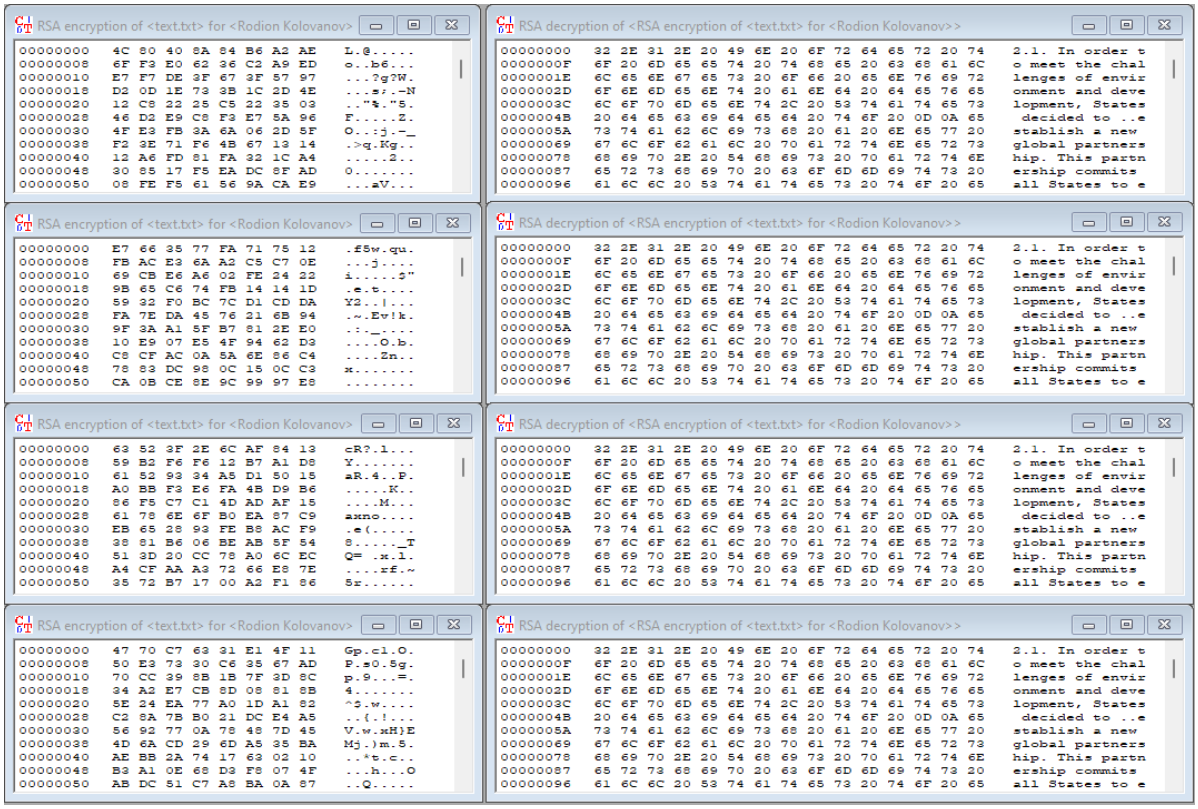
		160933752110573317869239452019 53
1024	65537	179646968079794852934499050935586406151741487959774288402068477160206304703026798653065812592497064604160292699265203433217961979535847858729892200520573719451648230686044931633272855414742663658931116996832034387361611549281954994825922526093323661292651913277281621146882111997255381093561420527683087423349
2048	65537	32012037223752464167472331473230645720010327339305496024210098168024083559099589923842090934652363981257014714275010491166622331631610874626795198160540496236799946181757260330264290809287990645872192585394318770282476394706191614394745207482999687607757615458778789187330791412562530985422999079150063725961747547676306065051773608050661854763287251355112856801539590019326752177058495278906795881093784408767086292257653559941146171307605197119600951793102407803718869771159680006404583301873748036357665017463804356865194972968481768675066252807316124893996461223054746164823518891829585913460136510318884738565527

Last name	First name	Key type	Key identifier	Created	Internal ID no.
Kolovanov	Rodion	RSA-1024		17.11.2022 03:15:20	1668644120
Kolovanov	Rodion	RSA-2048		17.11.2022 03:14:50	1668644090
Kolovanov	Rodion	RSA-512		17.11.2022 03:15:43	1668644143
Kolovanov	Rodion	RSA-768		17.11.2022 03:15:32	1668644132
SideChannelAt...	Bob	RSA-512	PIN=1234	06.07.2006 12:51:34	1152179494

Далее исходный текст был зашифрован и расшифрован ключами различной длины. Время зашифровки и расшифровки было зафиксировано. Результаты приведены в следующей таблице:

Длина ключа	Время зашифровки	Время расшифровки
512	0.000 секунд	0.008 секунд
768	0.000 секунд	0.014 секунд
1024	0.000 секунд	0.020 секунд
2048	0.002 секунд	0.078 секунд

Все расшифрованные тексты совпадают с исходным текстом. Зашифрованные и расшифрованные тексты (зашифрованный – слева, расшифрованный – справа) при использовании ключа длиной 512, 768, 1024, 2048 соответственно (сверху вниз, 512 – сверху, 2048 – снизу) представлены на следующем скриншоте:



Как видно из результатов, шифрование данных алгоритмом RSA занимает значительно больше времени по сравнению с алгоритмами симметричного шифрования.

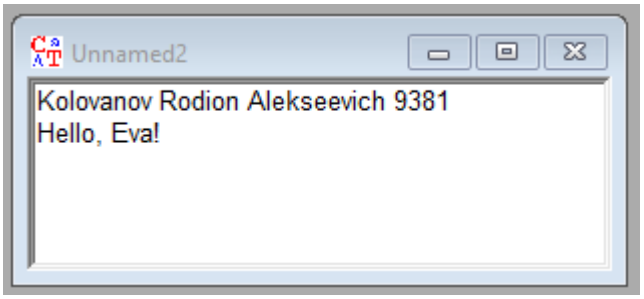
Атака грубой силы на RSA.

Задание.

1. Запустите утилиту Indiv.Procedures -> RSACryptosystem -> RSA Demonstration;
2. Установите переключатель в режим «Choose two prime…»;
3. Выберите параметры p и q так, чтобы $n = p * q > 256$;
4. Задайте открытый ключ e;
5. Зашифруйте произвольное сообщение и передайте его вместе с n и e коллеге. В ответ получите аналогичные данные от коллеги;
6. Запустите утилиту Indiv.Procedures -> RSACryptosystem -> RSADemonstration и установите переключатель в режим «For data encryption…»;
7. Выполните факторизацию модуля n командой Factorize;
8. Используйте полученный результат для расшифровки сообщения, полученного от коллеги. Проверьте корректность.

Атака грубой силы на шифр RSA.

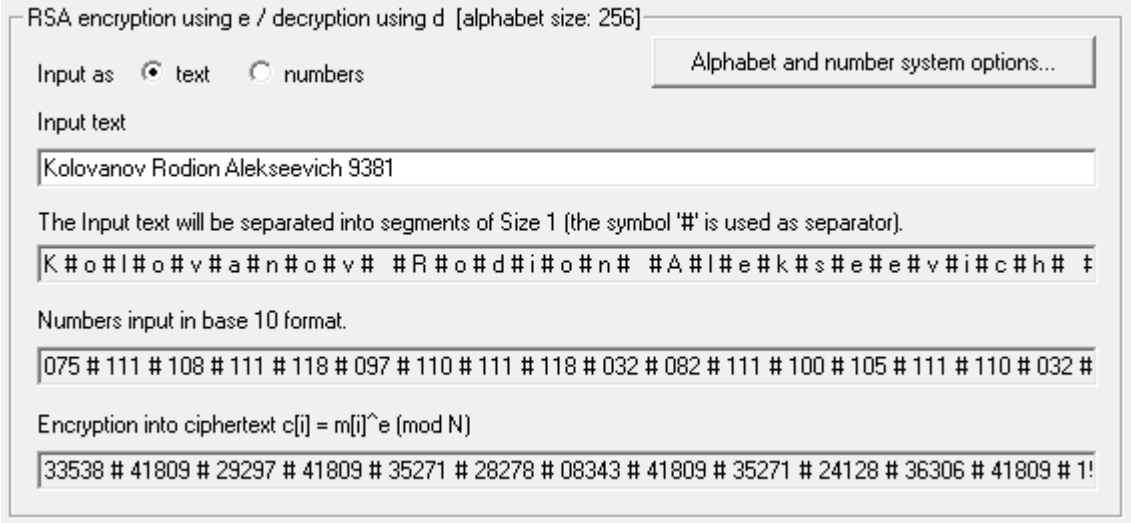
В качестве исходного текста был взят следующий текст:



Далее при помощи утилиты Indiv.Procedures -> RSACryptosystem -> RSA Demonstration исходный текст был зашифрован со следующими параметрами:

- p = 251;
- q = 197;
- n = 49447;
- $e = 2^{16} + 255$.

Результаты шифрования:



Коллеге была передана следующая информация:

- Шифротекст: 33538 # 41809 # 29297 # 41809 # 35271 # 28278 # 08343 # 41809 # 35271 # 24128 # 36306 # 41809 # 15653 # 30988 # 41809 # 08343 # 24128 # 15285 # 29297 # 45541 # 03798 # 36978 # 45541 # 45541 # 35271 # 30988 # 11942 # 21193 # 24128 # 11931 # 11813 # 40121 # 41655;
- n = 49447;
- $e = 2^{16} + 255$.

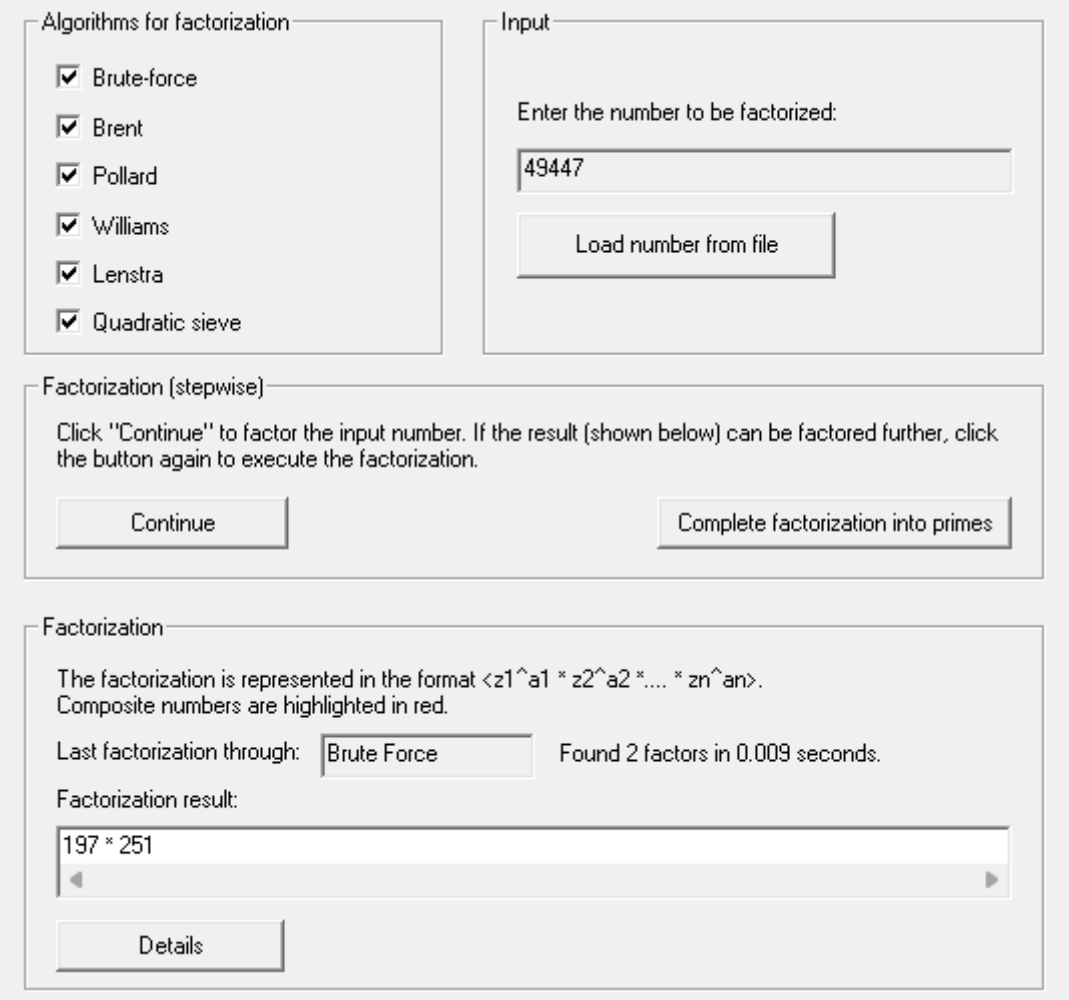
Было представлено, что автор данной лабораторной работы является сам себе является коллегой. От коллеги была получена следующая информация:

- Шифротекст: 33538 # 41809 # 29297 # 41809 # 35271 # 28278 # 08343 # 41809 # 35271 # 24128 # 36306 # 41809 # 15653 # 30988 # 41809 # 08343 # 24128 # 15285 # 29297 # 45541 # 03798 # 36978 # 45541 #

45541 # 35271 # 30988 # 11942 # 21193 # 24128 # 11931 # 11813 # 40121 # 41655;

- n = 49447;
- $e = 2^{16} + 255$.

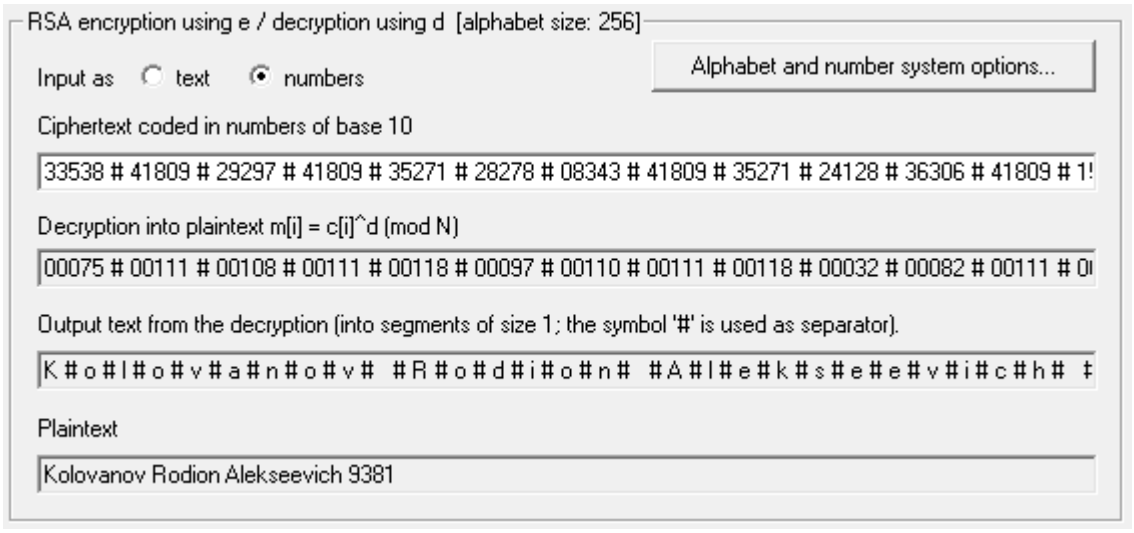
Далее в утилите Indiv.Procedures -> RSACryptosystem -> RSADemonstration был переключен режим на «For data encryption…», после чего была выполнена факторизация полученного модуля n:



Отсюда были получены параметры p и q:

- p = 197;
- q = 251;

При помощи известных e, p и q можно найти d, после чего осуществить расшифровку полученного от коллеги шифротекста:



Как видно из результатов, расшифрованное сообщение совпадает с исходным. Атака грубой силы на RSA прошла успешно.

Имитация атаки на гибридную криптосистему.

Задание.

1. Подготовьте текст передаваемого сообщения на английском с вашим именем в конце;
2. Запустите утилиту Analysis -> Asymmetric Encryption -> Side-Channel attack on «Textbook RSA»;
3. Настройте сервер, указав в качестве ключевого слова ваше имя, используемое в конце текста;
4. Выполните последовательно все шаги протокола;
5. Сохраните лог-файлы участников протокола для отчета.

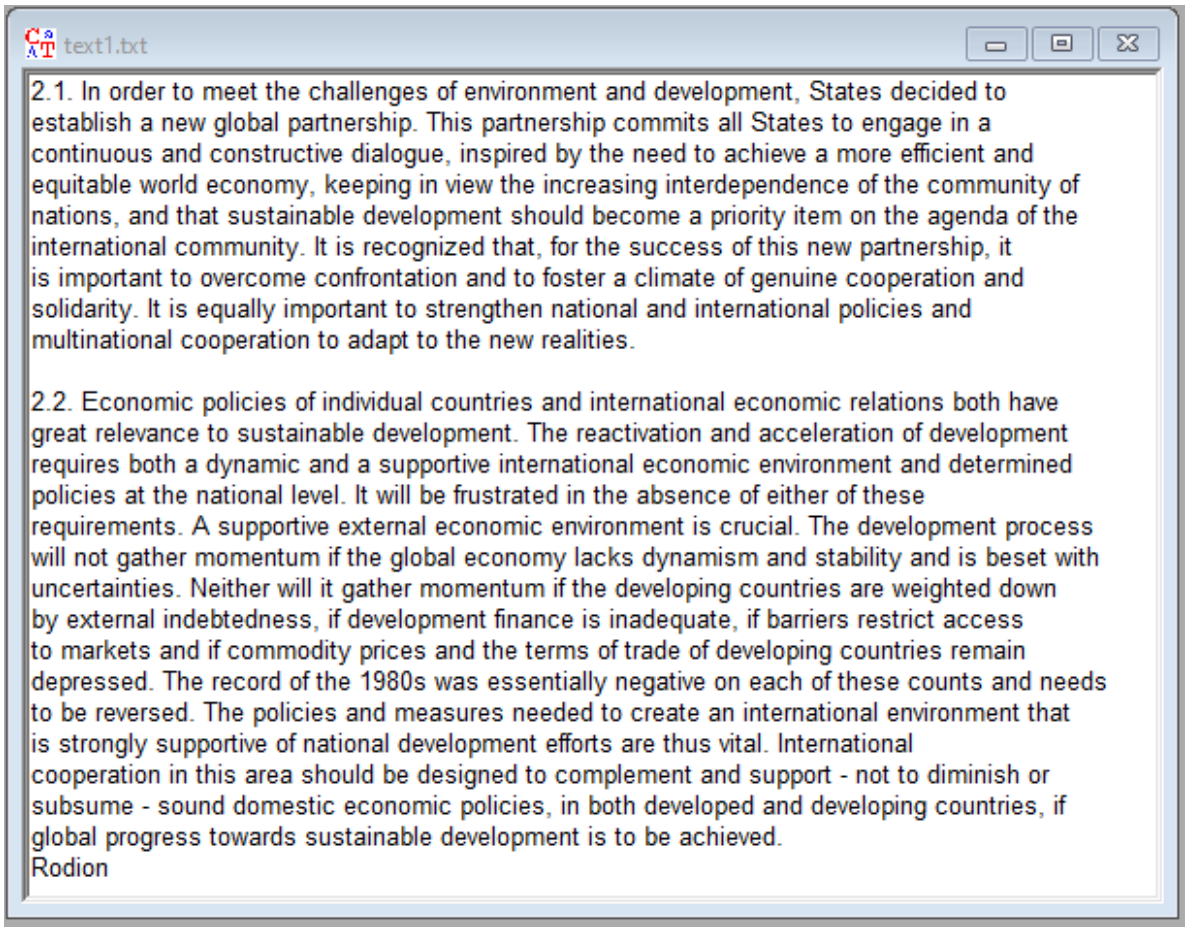
Цель атаки.

Определить симметричный секретный ключ, зашифрованный открытым ключом асимметричной криптосистемы. Атака на гибридную модель основана на том, что злоумышленник перехватывает цифровой конверт, содержащий

зашифрованное сообщение и зашифрованный секретный ключ. Затем, модифицируя полученные данные, побитово восстанавливает зашифрованный секретный ключ, анализируя положительные и отрицательные ответы сервера.

Проведение атаки.

В качестве исходного текста был взят следующий текст:



Для рассмотрения атаки была открыта утилита Analysis -> Asymmetric Encryption -> Side-Channel attack on «Textbook RSA». В качестве ключевого слова для сервера было выбрано слово «Rodion».

Далее были выполнены все шаги атаки, в результате которой был получен исходный текст:

Как видно из результатов, атака была проведена успешно. Файлы логов проведения атаки на гибридную криптосистему представлены в Приложении А.

Выводы.

В ходе выполнения данной лабораторной работы были исследованы протокол Диффи-Хеллмана и шифр RSA, а также были исследованы атака «грубой силы» на шифр RSA и атака на гибридную криптосистему.

1. Протокол Диффи-Хеллмана:
    a. Была рассмотрена работа протокола Диффи-Хеллмана. Было определено, что протокол позволяет двум пользователям получить материал для генерации одинакового симметричного секретного ключа путем обмена данными по незащищенному каналу связи. (p, g, R_1) и (p, g, R_2) – открытые ключи сторон, x, y – закрытые ключи сторон, $R_2^{\,x} \bmod p$ и $R_1^{\,y} \bmod p$ – односторонние функции с секретом. p – большое простое число порядка 300 десятичных цифр, g – порождающий элемент циклической группы (генератор) порядка p, x и y – большие случайные числа. Симметричный ключ вычисляется следующим образом: $K = R_2^{\,x} \bmod p = R_1^{\,y} \bmod p$.
2. Шифр RSA:
    a. Была рассмотрена работа шифра RSA. Было определено, что RSA является асимметричным блочным шифром. Он базируется на том, что задача разложения чисел вида $n = p * q$ (p и q – простые числа) на множители является очень трудной, если мы знаем только n, а p и q – большие числа (задача факторизации). (e, n) – открытый ключ, d – закрытый ключ. p и q после генерации ключей уничтожаются. Каждый блок открытого текста преобразуется в шифротекст по формуле: $c_i = (m_i^e) \bmod n$, блок шифротекста преобразуется в открытый текст по формуле: $m_i = (c_i^{\,d}) \bmod n$.
    b. Было исследовано время шифрования и расшифрования шифром RSA в зависимости от длины ключа. Было определено, что шифр

RSA работает довольно медленно: при шифровании текста размера 2148 символов при шифровке и расшифровке в сумме используется 0.02 секунд при длине ключа 1024, и 0.08 секунд при длине ключа 2048. Отсюда применение шифра RSA для больших объемов данных нецелесообразно, лучше использовать гибридное шифрование.

3. Атака «грубой силы» на шифр RSA:
    a. Была рассмотрена и проведена атака «грубой силы» на шифр RSA. Для этого осуществлялась факторизация модуля n на простые множители p и q. Далее по известным p, q и e был найден закрытый ключ d. Атака прошла успешно, поскольку используемые при генерации ключей значения p и q были недостаточно большими, поэтому факторизация числа $n = p * q$ выполнилась относительно быстро.
4. Атака на гибридную криптосистему:
    a. Была рассмотрена имитации атаки на гибридную криптосистему. Было определено, что целью рассматриваемой атаки является определение симметричного секретного ключа, зашифрованного открытым ключом асимметричной криптосистемы. Атака основана на том, что злоумышленник перехватывает цифровой конверт, содержащий зашифрованное сообщение и зашифрованный секретный ключ. Затем, модифицируя полученные данные, побитово восстанавливает зашифрованный секретный ключ, анализируя положительные и отрицательные ответы сервера.

Были получены практические навыки работы с рассматриваемыми протоколами и шифрами, и атакой на них с использованием приложения CrypTool 1 и 2.

ПРИЛОЖЕНИЕ А

Лог-файл работы протокола Диффи-Хеллмана в *CrypTool 1*:

```
At first, Alice and Bob agreed on the public parameters. So they chose a prime
p and a generator g:

p:
162728503386883098193052774928368534539825683780064557063748120138904672906063

g:
713258139208658959771417644040443230574053121863052852482456530867017966572 56

Alice chose her secret number 'a' while Bob chose his secret number 'b':

a:
181437086366002638331916106425384139194705658449643608145846885483838214039 01

b:
913974164241358537197630080183603299335850341581529301021491821158638564397 5

If the chosen secret values a and b are greater or equal the prime module p,
then they need to be reduced modulo p. The actual values are given below:

a (reduced mod p):
181437086366002638331916106425384139194705658449643608145846885483838214039 01

b (reduced mod p):
913974164241358537197630080183603299335850341581529301021491821158638564397 5

On the basis of the previously chosen secret numbers, Alice and Bob created
their respective shared keys. Alice computed her shared key A, while Bob computed
his shared key B:

A:
532861113203919000260821422942904763335467085362905598397606323241298310141 28

B:
634162165273239016959801935071172396084223491220668891426316950446073049801 11

In order to calculate their secret and common Session Key, Alice and Bob
exchanged their shared keys: Alice sent her shared key A to Bob and Bob sent
his shared key B to Alice.

Alice and Bob were able to calculate the secret and common Session Key now.
Alice computed the Session Key SA, Bob computed the Session Key SB:

SA:
451899973461403099333691680875751588575298382065722053057692866388384842838 93

SB:
451899973461403099333691680875751588575298382065722053057692866388384842838 93

Theoretically it is now possible for Alice and Bob to use their Session Keys to
encrypt documents they would like to exchange covertly.
```

Файлы логов проведения атаки на гибридную криптосистему:

I. PREPARATIONS

Alice composes a message M, addressed to Bob.

Alice chooses a random session key S:
ED43DD9EB63FE3D4CED3C9F0A9B12821

Alice symmetrically encrypts the message M with the session key S.

Alice chooses Bob's public key e:
010001

Alice asymmetrically encrypts the session key S with Bob's public RSA key e:
67C106C51A1A2E7931FC139984444215319BC0074E96D317323054F358C38FBD86780B5A9099A2
A4E5364F72C45EC40AD379829D20BAF646169C6E8D4D6DDDA5AB426A3E39FF0DA47729DD8B0FC9
FEA8586A91F470E53DD72A29CC9071DA763A1C84E5E4E5997CFBB79766BA22ABB94F4546D20339
1D46411AB68759834F80C17EB6C112A87C824E72D6D64CB7E9181AC54FCAD7597942068DBD20F5
F3705EC632D5412F7A92F11FC692D02B100F24559C8496C746DDF5ECE3E728CA7F0E016AE197AF
628C21231505EB2B71A2BA3FE52BBC0927ED1405DC403F60A24F2BA083220143B10AD63D72CBFB
16EF7BE13D1C8E4F4CEBB3553DE7E6D6AAAEC4094DFD

II. MESSAGE TRANSMISSION

Alice sends the hybrid encrypted file to Bob over an insecure channel.

III. MESSAGE INTERCEPTION

Trudy intercepts the hybrid encrypted file and isolates the encrypted session key S:
67C106C51A1A2E7931FC139984444215319BC0074E96D317323054F358C38FBD86780B5A9099A2
A4E5364F72C45EC40AD379829D20BAF646169C6E8D4D6DDDA5AB426A3E39FF0DA47729DD8B0FC9
FEA8586A91F470E53DD72A29CC9071DA763A1C84E5E4E5997CFBB79766BA22ABB94F4546D20339
1D46411AB68759834F80C17EB6C112A87C824E72D6D64CB7E9181AC54FCAD7597942068DBD20F5
F3705EC632D5412F7A92F11FC692D02B100F24559C8496C746DDF5ECE3E728CA7F0E016AE197AF
628C21231505EB2B71A2BA3FE52BBC0927ED1405DC403F60A24F2BA083220143B10AD63D72CBFB
16EF7BE13D1C8E4F4CEBB3553DE7E6D6AAAEC4094DFD

IV. BEGINNING OF THE ATTACK CYCLE

She sends an exact copy of the original, encrypted message to Bob and extends it with the session key S' (encrypted with Bob's public key). Compared to the message sent by Alice, Trudy simply replaces the encrypted session key [ENC(S, PubKeyBob) is replaced by ENC(S', PubKeyBob)].
Trudy repeats this step 130 times, whereas the step count depends on the bit length of the used session key (step count = bit length + 2).