**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.***

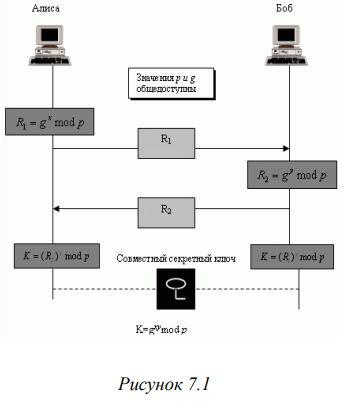
Протокол Диффи-Хеллмана является первым из опубликованных криптопреобразований на основе открытых ключей. Поэтому этот протокол ещё называют обменом ключами по схеме Диффи-Хеллмана.

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*:
   1. *p* – большое простое число порядка 300 десятичных цифр (1024 бита),
   2. *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:
   1. ,
   2. .
4. Стороны обмениваются открытыми ключами и вычисляют симметричный общий ключ K:





***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*.

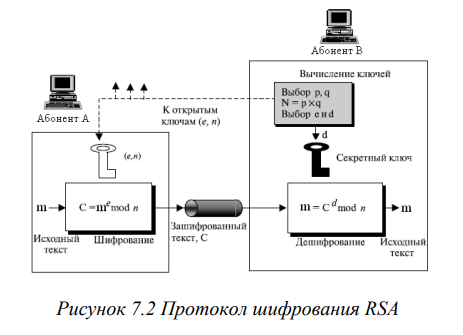
Алгоритм шифрования RSA состоит из следующих операций (рисунок 7.2):

1. Вычисление ключей:
   1. Генерация двух больших простых чисел *p* и *q* (*p* и *q* держаться в секрете);
   2. Вычисление *n* = *p* *\** *q*;
   3. Выбор произвольного *e* (*e < n*), взаимно простого с 𝜑(𝑛) – функцией Эйлера;
   4. Вычисление 𝑑: 𝑒 ∗ 𝑑 = 1 𝑚𝑜𝑑 𝜑(𝑛);
   5. Числа (*e*, *n*) – открытый ключ, *d* – закрытый ключ, *p* и *q* уничтожаются.
2. Шифрование:
   1. Открытый текст разбивается на блоки 𝑚*i* : 𝑚*i* < 𝑛;
   2. Каждый блок открытого текста преобразуем в шифротекст по формуле:



1. Расшифровка:
   1. Шифротекст представляется блоками *сi*: *сi* < *n*;
   2. Каждый блок шифротекста преобразуется в открытый текст по формуле:





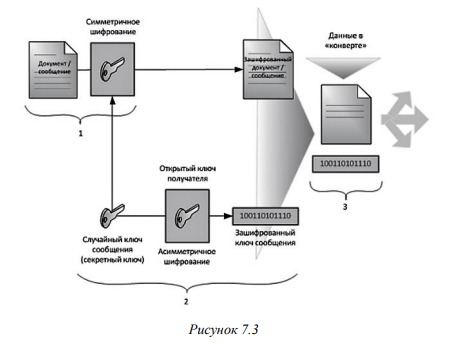
***Имитация атаки на гибридную криптосистему.***

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.



**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*, *R1*) и (*p*, *g*, *R2*) – открытые ключи сторон;
* *x*, *y* – закрытые ключи сторон;
*  и  – односторонние функции с секретом (TOWF).

Математическая модель протокола Диффи-Хеллмана:

* *р* – большое простое число порядка 300 десятичных цифр (1024 бита);
* *g* – порождающий элемент циклической группы (генератор) порядка *р*, для которого справедливо: 𝑔 𝑚𝑜𝑑 𝑝, 𝑔2 𝑚𝑜𝑑 𝑝, 𝑔3 𝑚𝑜𝑑 𝑝, … , 𝑔*p-1* 𝑚𝑜𝑑 𝑝 являются различными целыми из [1, p - 1];
* *х*, *у* – большие случайные числа такие, что *0 < x < p – 1*, *0 < y < p – 1*;
* Поскольку:

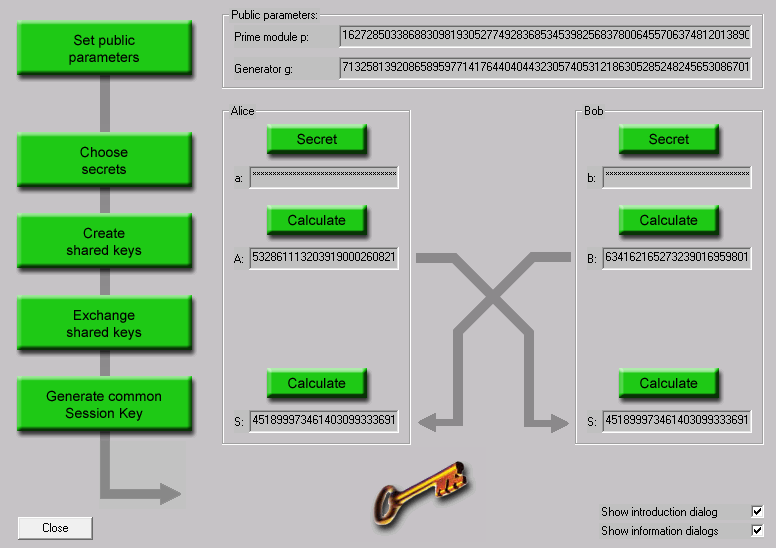




* Стороны фактически создают материал для симметричного ключа сеанса без Центра распределения ключей (KDC).

*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. |
| *R1* | *A* | Открытый ключ Алисы. |
| *R2* | *B* | Открытый ключ Боба. |
| *K* | *S* | Общий симметричный ключ. |

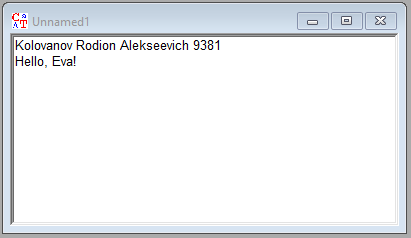
*Шифровка и расшифровка произвольного сообщения.*

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

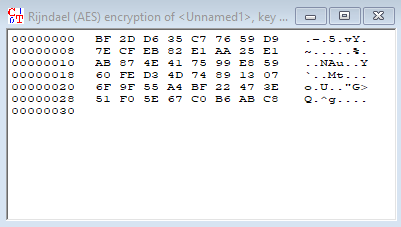
Общий ключ K:

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).

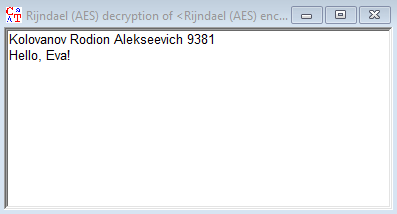
Исходное сообщение:



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* (*р* и *q* – простые числа) на множители является очень трудной, если мы знаем только *n*, а *р* и *q* – большие числа (задача факторизации).

Генерация ключей:

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



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

Зашифрование:

* Открытый текст разбивается на блоки 𝑚𝑖 размером 𝑘 ≤ [𝑙𝑜𝑔2𝑛] бит. Блоки интерпретируются, как числа из диапазона (0, 2𝑘 - 1);
* Ключ шифрации – пара чисел (*e*, *n*) (открытый ключ);
* Каждый блок открытого текста преобразуется в шифротекст по формуле:



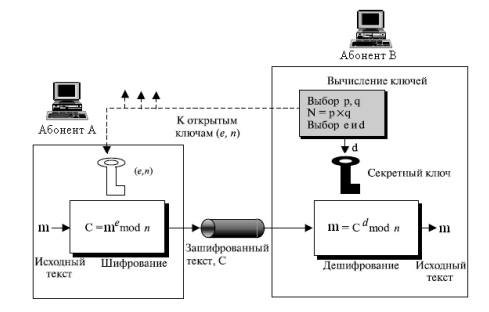
Расшифрование:

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



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





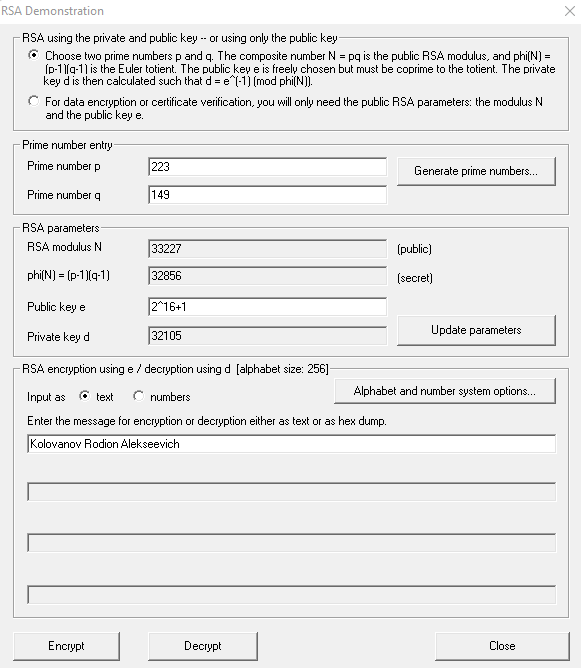
*Шифровка и расшифровка сообщения.*

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.

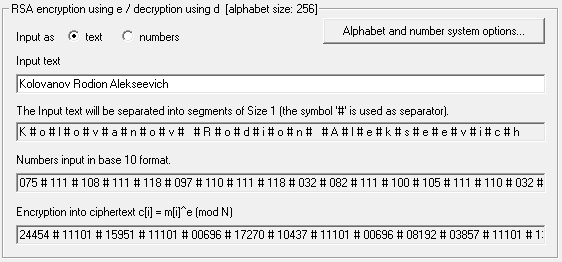
Открытый ключ: *e* = 216 + 1 и *n* = 33227.

Закрытый ключ: *d* = 32105.

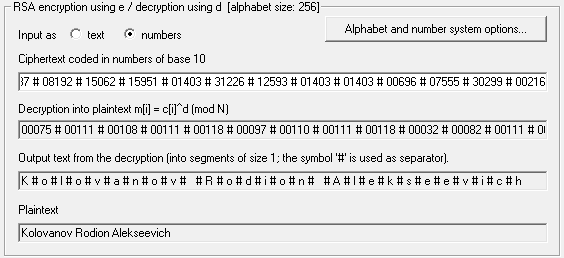
Скриншот результата генерации ключей:



Скриншот результата шифрации:



Скриншот результата расшифровки:



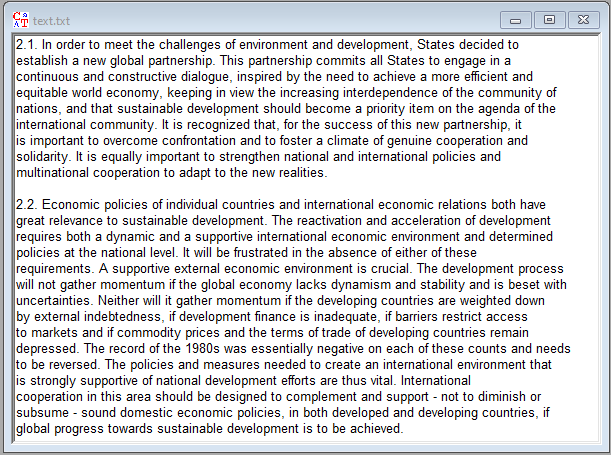
***Исследование шифра RSA.***

*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:

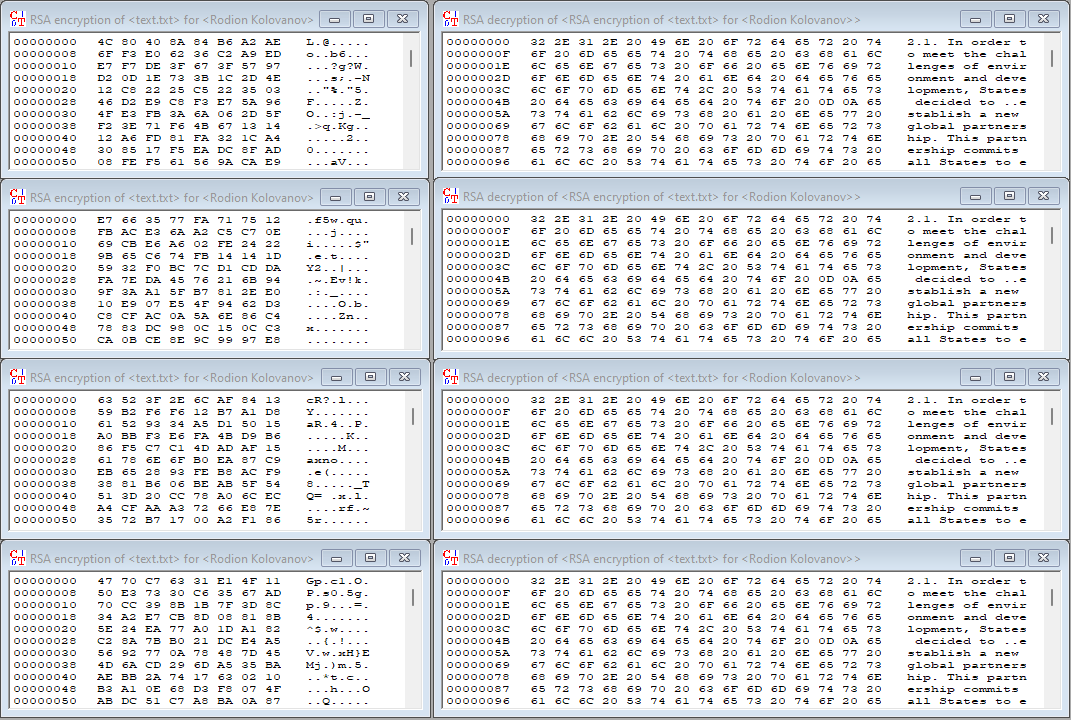
|  |  |  |
| --- | --- | --- |
| Длина ключа | Экспонента | Модуль |
| 512 | 65537 | 1340214050183598652934880408668166408882 9601975528719947844144202637396257687344 3446394788640465305028377059592930984248 53685064842214466402682833095015179 |
| 768 | 65537 | 1532374822079986942630238572837271761588  3259378919116367672536162334774244303946  5635063370764201056085774634150064609331  4617736998004037174946587740302276568883  7853299881824738121883650749122101955831  16093375211057331786923945201953 |
| 1024 | 65537 | 1796469680797948529344990509355864061517  4148795977428840206847716020630470302679  8653065812592497064604160292699265203433  2179619795358478587298922005205737194516  4823068604493163327285541474266365893111  6996832034387361611549281954994825922526  0933236612926519132772816211468821119972  55381093561420527683087423349 |
| 2048 | 65537 | 320120372237524641674723314732306457200 103273393054960242100981680240835590995 899238420909346523639812570147142750104 911666223316316108746267951981605404962 367999461817572603302642908092879906458 721925853943187702824763947061916143947 452074829996876077576154587787891873307 914125625309854229990791500637259617475 476763060650517736080506618547632872513 551128568015395900193267521770584952789 067958810937844087670862922576535599411 461713076051971196009517931024078037188 697711596800064045833018737480363576650 174638043568651949729684817686750662528 073161248939964612230547461648235188918 29585913460136510318884738565527 |



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.

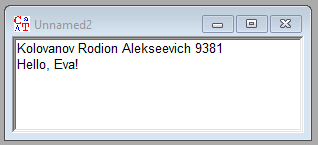
***Атака грубой силы на RSA.***

*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.*

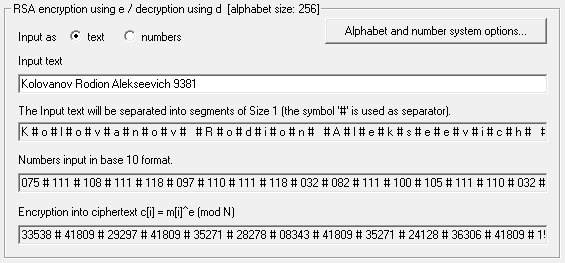
В качестве исходного текста был взят следующий текст:



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

* *p* = 251;
* *q* = 197;
* *n =* 49447;
* *e* = 216 + 255.

Результаты шифрования:



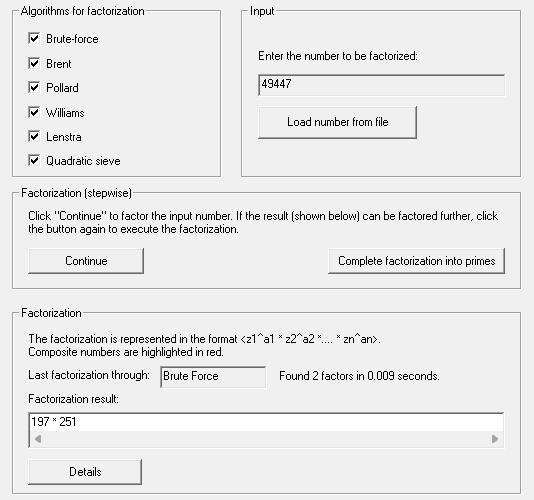
Коллеге была передана следующая информация:

* 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* = 216 + 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* = 216 + 255.

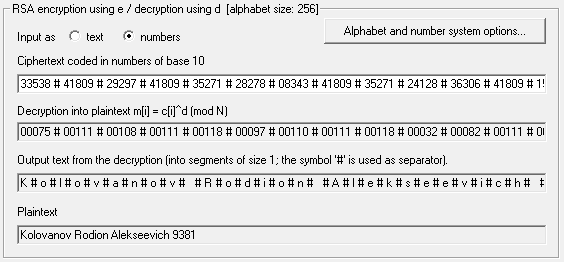
Далее в утилите Indiv.Procedures -> RSACryptosystem -> RSADemonstration был переключен режим на «For data encryption…», после чего была выполнена факторизация полученного модуля n:



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.*

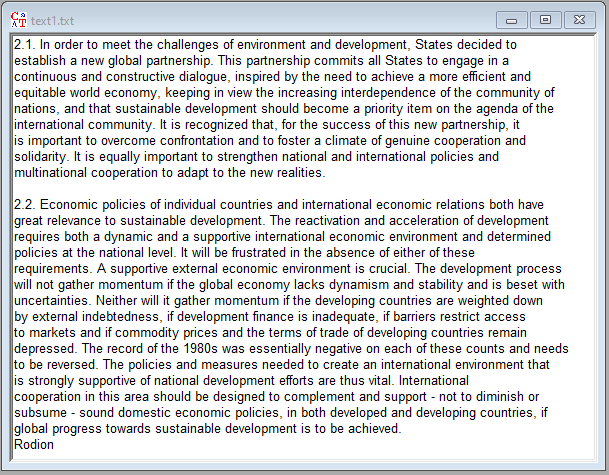
1. Подготовьте текст передаваемого сообщения на английском с вашим именем в конце;
2. Запустите утилиту Analysis -> Asymmetric Encryption -> Side-Channel attack on «Textbook RSA»;
3. Настройте сервер, указав в качестве ключевого слова ваше имя, используемое в конце текста;
4. Выполните последовательно 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:
   1. 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*, *R1*) и (*p*, *g*, *R2*) – открытые ключи сторон, *x*, *y* – закрытые ключи сторон,  и  – односторонние функции с секретом. *р* – большое простое число порядка 300 десятичных цифр, *g* – порождающий элемент циклической группы (генератор) порядка *р*, *х* и *у* – большие случайные числа. Симметричный ключ вычисляется следующим образом: .
2. Шифр RSA:
   1. 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* (*р* и *q* – простые числа) на множители является очень трудной, если мы знаем только *n*, а *р* 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: , блок шифротекста преобразуется в открытый текст по формуле: .
   2. 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:
   1. 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:
   1. 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: 71325813920865895977141764404044323057405312186305285248245653086701796657256  Alice chose her secret number 'a' while Bob chose his secret number 'b':  a: 18143708636600263833191610642538413919470565844964360814584688548383821403901  b: 9139741642413585371976300801836032993358503415815293010214918211586385643975  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):  18143708636600263833191610642538413919470565844964360814584688548383821403901  b (reduced mod p):  9139741642413585371976300801836032993358503415815293010214918211586385643975  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: 53286111320391900026082142294290476333546708536290559839760632324129831014128  B: 63416216527323901695980193507117239608422349122066889142631695044607304980111  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: 45189997346140309933369168087575158857529838206572205305769286638838484283893  SB: 45189997346140309933369168087575158857529838206572205305769286638838484283893  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 elice 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 keyhe 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). |