

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.8382

Нечепуренко Н.А.

Преподаватель

Племянников А.К.

Санкт-Петербург

2021

Цели работы.

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

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 теоретические положения.

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 1):

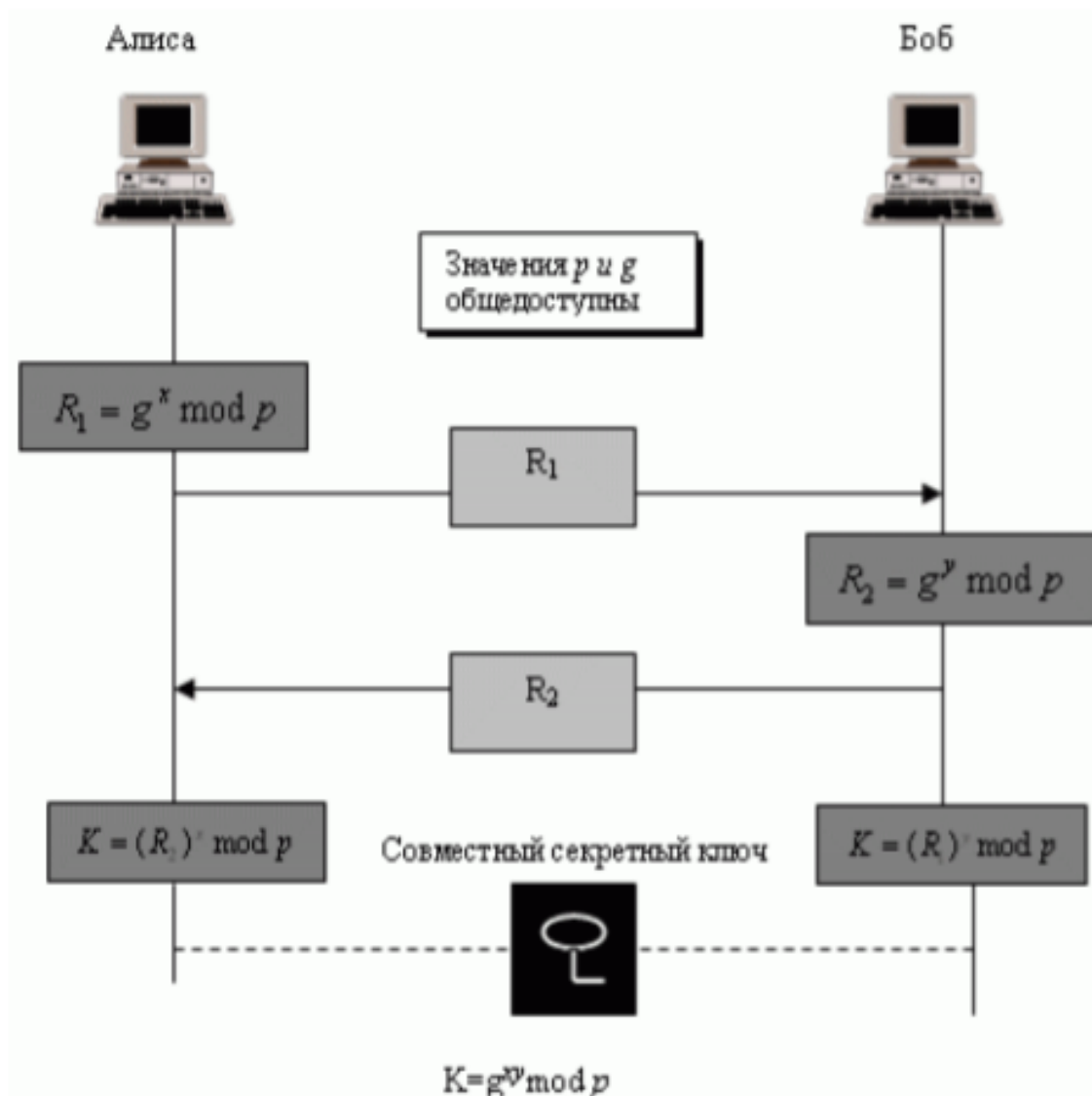


Рисунок 1 – Схема протокола Диффи-Хеллмана

1. Устанавливаются открытые параметры p , g :
 - p – большое простое число порядка 300 десятичных цифр (1024 бита),
 - 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:
 - $R_1 = g^x \bmod p$,

- $R_2 = g^y \mod p$

4. Стороны обмениваются открытыми ключами и вычисляют симметричный общий ключ К:

$$K = R_2^x \mod p = R_1^y \mod p$$

Реализация протокола Диффи-Хеллмана в CrypTool 1.

Запустим утилиту Diffie-Hellman Demonstration. С ее помощью сгенерируем p и g , а также закрытые ключи x и y (см. рис. 2).

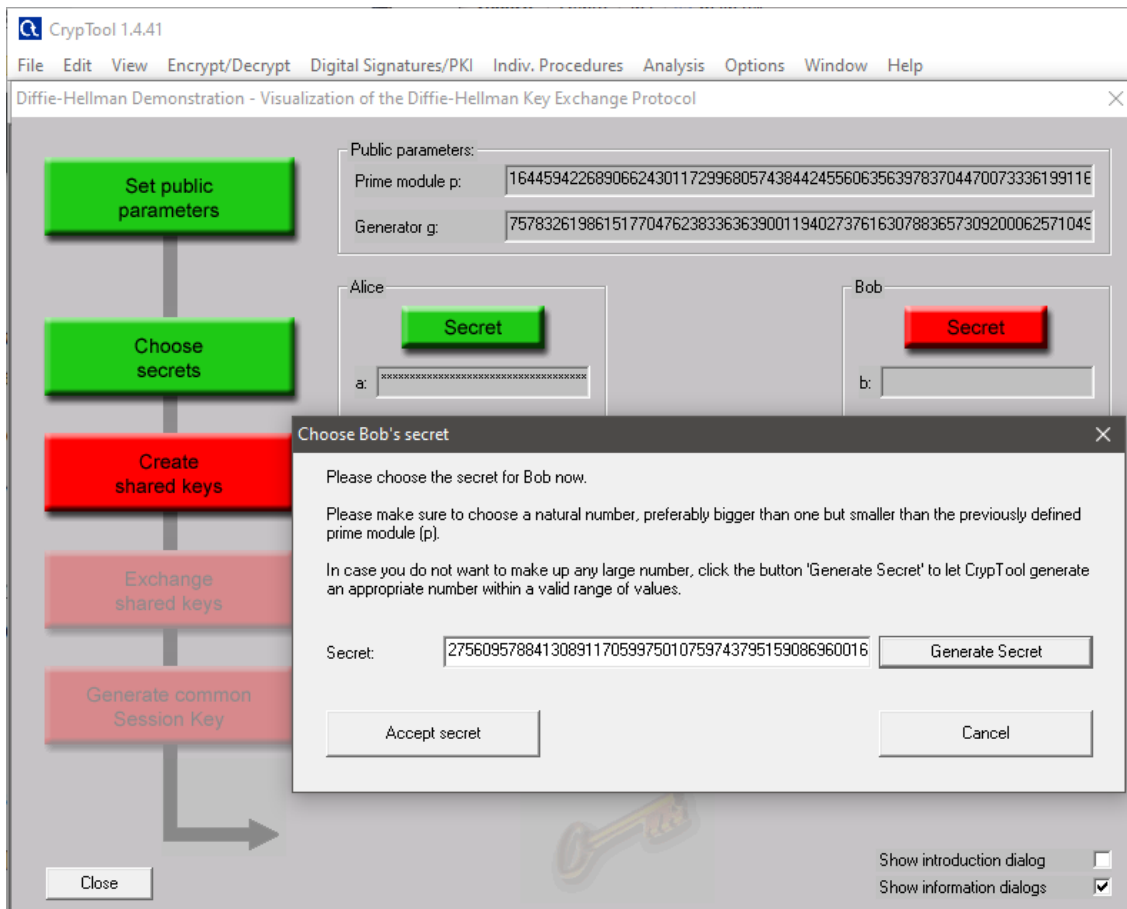


Рисунок 2 – Интерфейс Diffie-Hellman Demonstration

Выполнив все шаги протокола получаем общий ключ (см. рис. 3).

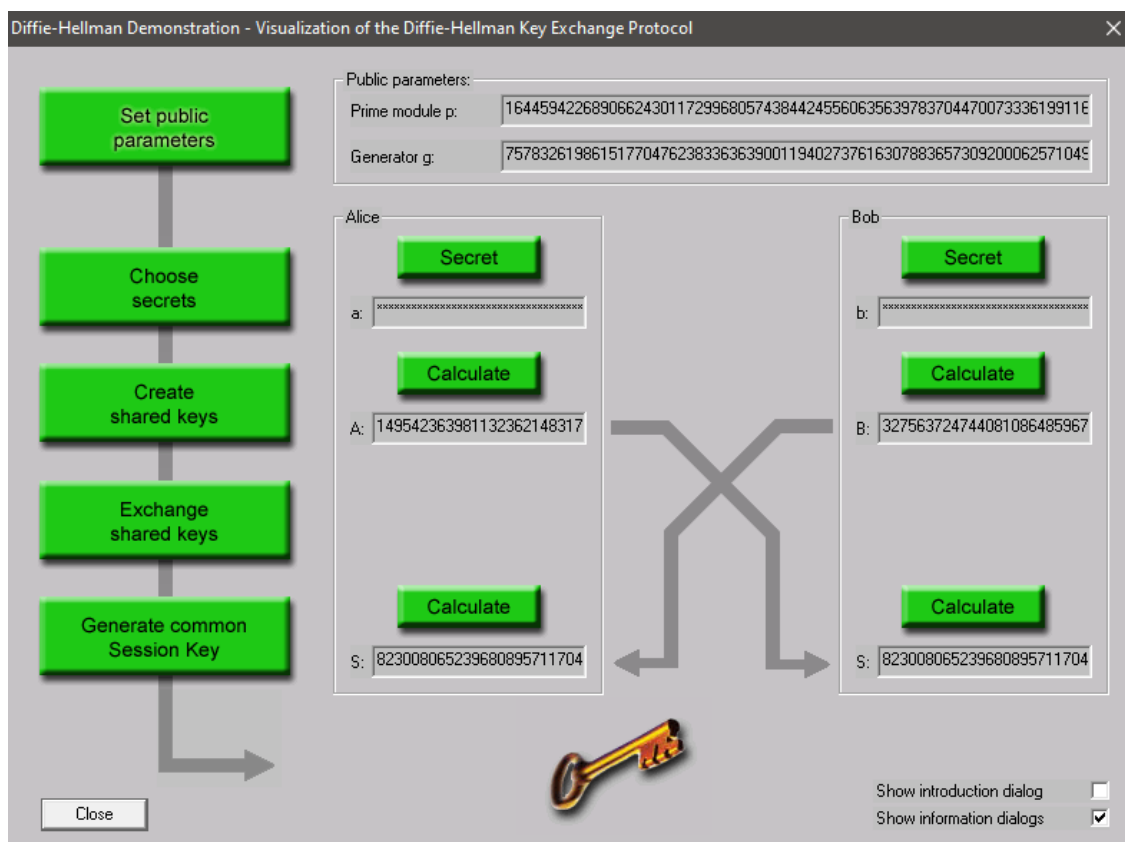
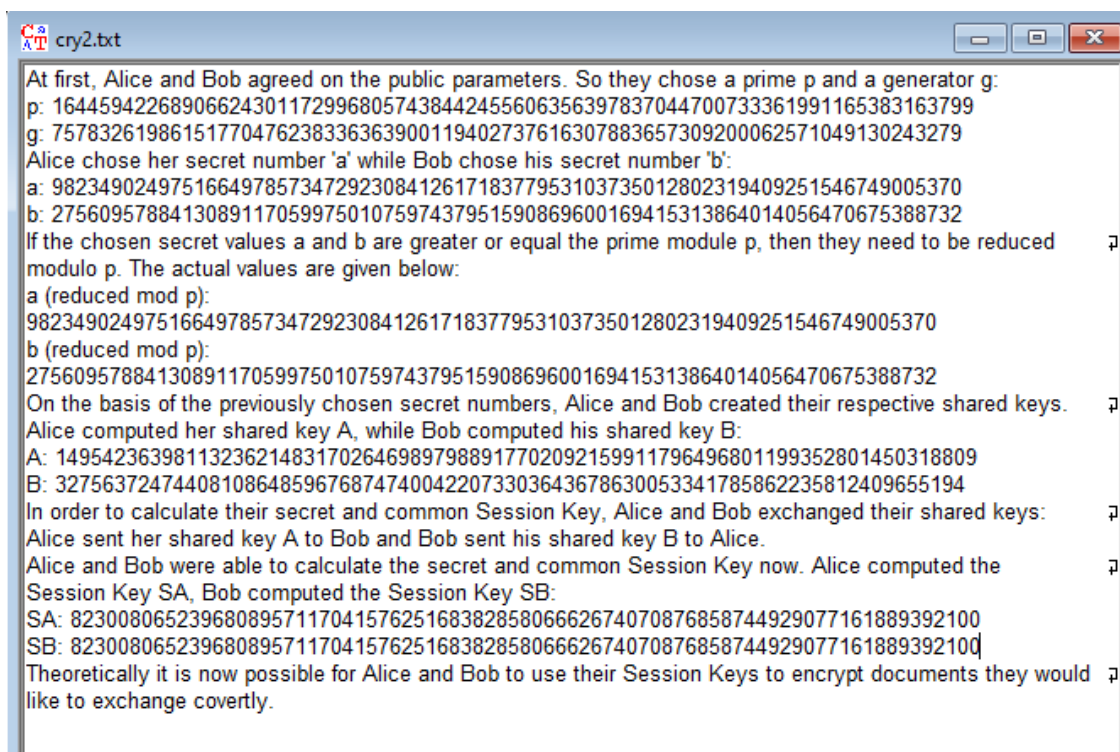


Рисунок 3 – Результат исполнения протокола Diffie-Hellman

Логи протокола приведены на рисунке 4.



```
At first, Alice and Bob agreed on the public parameters. So they chose a prime p and a generator g:
p: 164459422689066243011729968057438442455606356397837044700733361991165383163799
g: 75783261986151770476238336363900119402737616307883657309200062571049130243279
Alice chose her secret number 'a' while Bob chose his secret number 'b':
a: 98234902497516649785734729230841261718377953103735012802319409251546749005370
b: 27560957884130891170599750107597437951590869600169415313864014056470675388732
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):
98234902497516649785734729230841261718377953103735012802319409251546749005370
b (reduced mod p):
27560957884130891170599750107597437951590869600169415313864014056470675388732
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: 149542363981132362148317026469897988917702092159911796496801199352801450318809
B: 32756372474408108648596768747400422073303643678630053341785862235812409655194
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: 82300806523968089571170415762516838285806662674070876858744929077161889392100
SB: 82300806523968089571170415762516838285806662674070876858744929077161889392100
Theoretically it is now possible for Alice and Bob to use their Session Keys to encrypt documents they would
like to exchange covertly.
```

Рисунок 4 – Логи исполнения протокола Diffie-Hellman

Зашифруем и расшифруем произвольный текст порядка 1000 символов шифром AES, используя префикс общего ключа. Результат приведен на рисунке ниже.

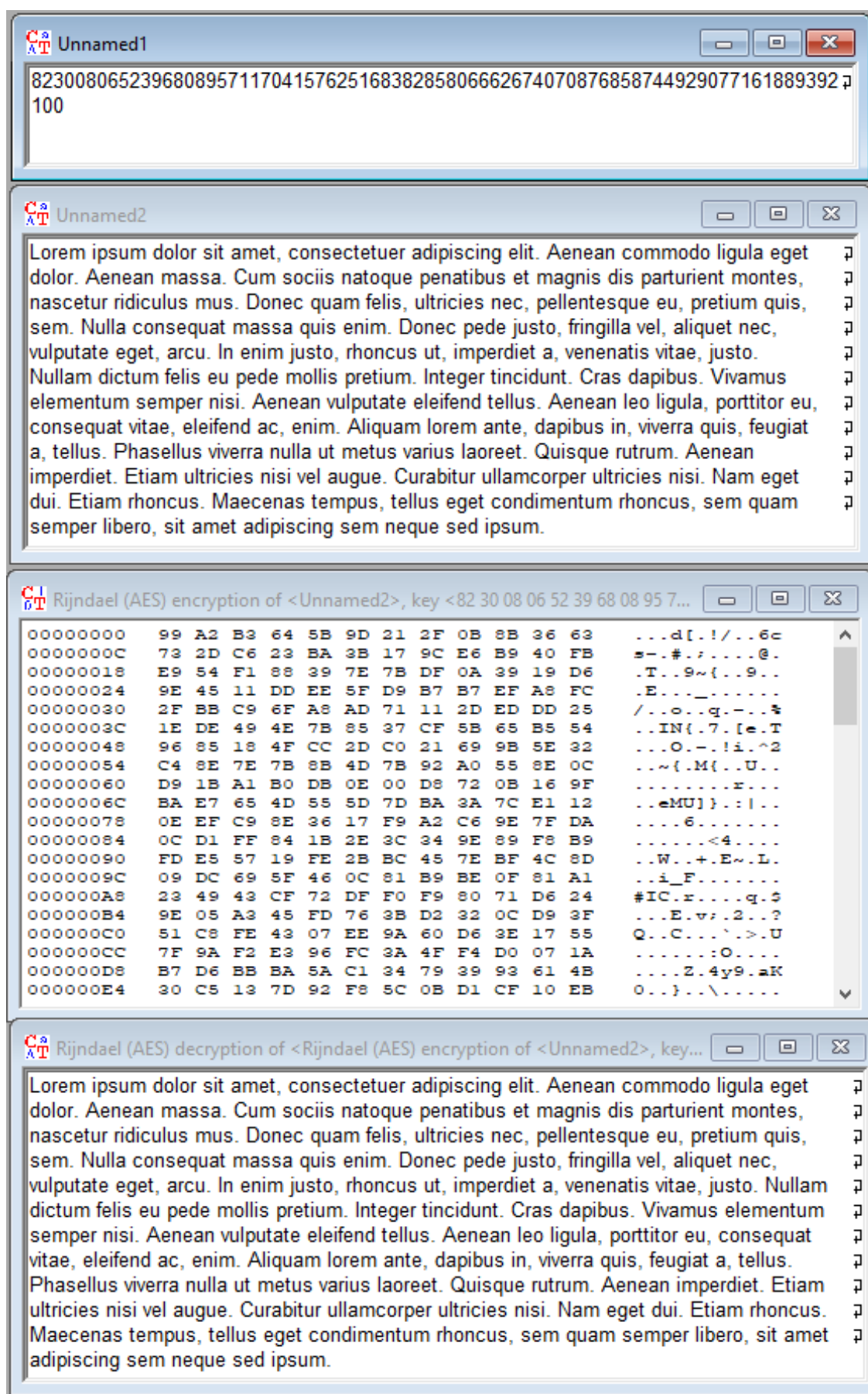


Рисунок 5 – Пример зашифровки и расшифровки текста с использованием общего ключа

Вывод.

Протокол Диффи-Хеллмана позволяет используя свойства малой теоремы Ферма обмениваться открытыми ключами, генерируя общий закрытый ключ шифрования. Средствами Cryptool 1 и утилиты Diffie-Hellman Demonstration было сгенерировано большое простое число p , его первообразный корень g , а также две закрытые части ключа. Затем был вычислен общий ключ, с помощью которого был зашифрован, а затем расшифрован некоторый открытый текст.

Шифр RSA.

Задание.

1. Запустите утилиту Indiv.Procedures -> RSACryptosystem
-> 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.

Основные теоретические положения.

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 6):

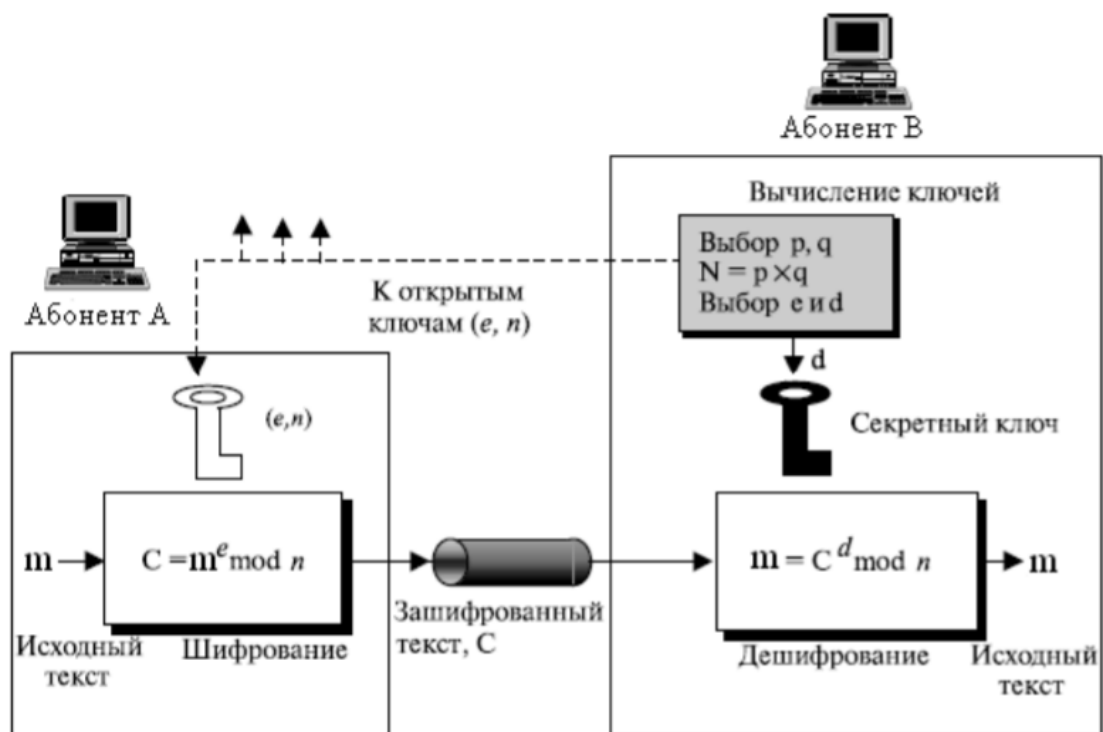


Рисунок 6 – Схема алгоритма RSA

1. Вычисление ключей:

- Генерация двух больших простых чисел p и q (p и q держаться в секрете).
- Вычисление $n = p * q$
- Выбор произвольного e ($e < n$), взаимно простого с $\phi(n)$ – функцией Эйлера
- Вычисление $d : e \cdot d = 1 \bmod \phi(n)$.
- Числа (e, n) – открытый ключ, d – закрытый ключ, p и q уничтожаются

2. Шифрование:

- Открытый текст разбивается на блоки $m_i : m_i < n$.
- Каждый блок открытого текста преобразуем в шифротекст по фор-

муле:

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

3. Расшифровка:

- Шифротекст представляется блоками $c_i : c_i < n$.
- Каждый блок шифротекста преобразуется в открытый текст по фор-

муле:

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

Шифр RSA в Cryptool 1.

С помощью утилиты RSA Demonstration в Crytool 1 сгенерируем открытый и закрытый ключи. Зашифруем сообщение NECHEPURENKO NIKITA ALEKSANDROVICH, результат представлен на рисунке 7.

RSA Demonstration

RSA using the private and public key -- or using only the public key

- ☒ Choose two prime numbers p and q. The composite number $N = pq$ is the public RSA modulus, and $\phi(N) = (p-1)(q-1)$ is the Euler totient. The public key e is freely chosen but must be coprime to the totient. The private key d is then calculated such that $d = e^{-1} \pmod{\phi(N)}$.
- ☐ For data encryption or certificate verification, you will only need the public RSA parameters: the modulus N and the public key e.

Prime number entry

Prime number p	<input type="text" value="193"/>	<button>Generate prime numbers...</button>
Prime number q	<input type="text" value="241"/>	

RSA parameters

RSA modulus N	<input type="text" value="46513"/>	(public)
$\phi(N) = (p-1)(q-1)$	<input type="text" value="46080"/>	(secret)
Public key e	<input type="text" value="2^16+1"/>	
Private key d	<input type="text" value="28673"/>	

Update parameters

RSA encryption using e / decryption using d [alphabet size: 256]

Input as ☒ text ☐ numbers Alphabet and number system options...

Input text

The Input text will be separated into segments of Size 1 (the symbol '#' is used as separator).

```
N # E # C # H # E # P # U # R # E # N # K # O #   # N # I # K # I # T # A #   # A # L # E # K # S # A # N # I
```

Numbers input in base 10 format.

```
078 # 069 # 067 # 072 # 069 # 080 # 085 # 082 # 069 # 078 # 075 # 079 # 032 # 078 # 073 # 075 # 073 #
```

Encryption into ciphertext $c[i] = m[i]^e \pmod{N}$

```
16202 # 25159 # 31333 # 42918 # 25159 # 15588 # 11882 # 15032 # 25159 # 16202 # 04047 # 14901 # 1!
```

Encrypt
Decrypt
Close

Рисунок 7 – Шифрование ФИО с помощью RSA

Теперь возьмем шифротекст и проведем дешифровку. Результат представлен на рисунке 8.

ния строки, состоящей из ФИО.

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 шифра RSA в Cryptool 1.

Для выполнения поставленных задач был сгенерирован рыбный текст длиной примерно в 1000 символов. Текст приведен на рисунке 9.

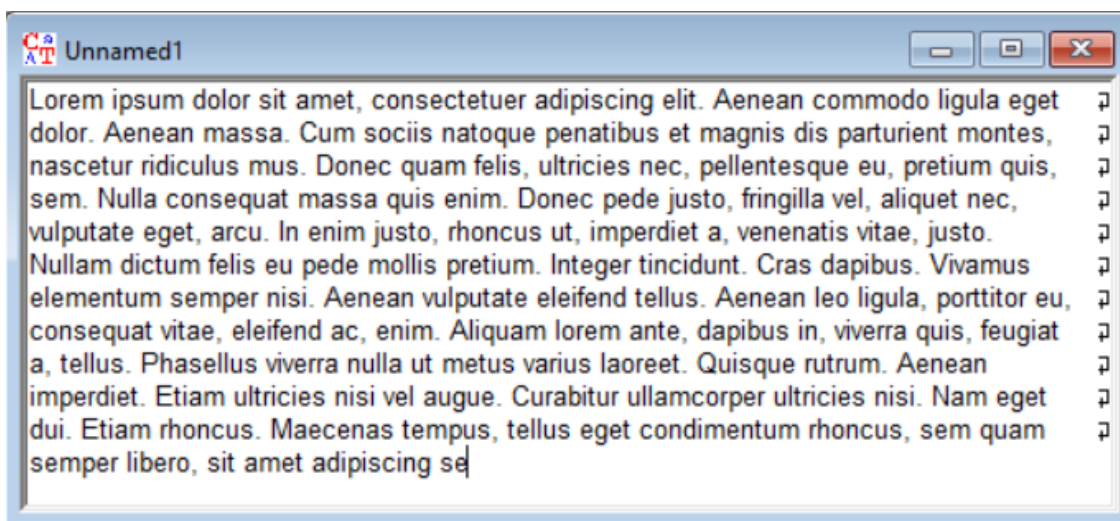


Рисунок 9 – Сгенерированный для задания текст

Интерфейс утилиты генерации ключей приведен на рисунке 10.

Generation of Asymmetric Key Pair

Algorithm

☒ RSA
 Bit length of RSA modulus: 512

☐ DSA
 Bit length of DSA prime number: 1024

☐ Elliptic curves
 Identifier (bit length and curve parameter): prime239v1

User data

The key pair will be put in an encrypted PSE with the name shown below. The key pair will be protected by your PIN code.

Last name: Nечepurenko

First name: Nikita

Key identifier (optional):

PIN:

PIN verification:

The domain parameter of the selected elliptic curve will be shown below.

Parameters	Value of the parameter	Bit len...

Base for presentation of numbers

☐ Octal
 ☒ Decimal
 ☐ Hexadecimal

Generate new key pair...

PKCS #12 Import

Show key pair...

Close

Рисунок 10 – Интерфейс утилиты генерации ключей

Сгенерируем 4 пары ключей разной длины: 512, 768, 1024 и 2048 битов. Выполним шифровку и дешифровку, зафиксируем время.

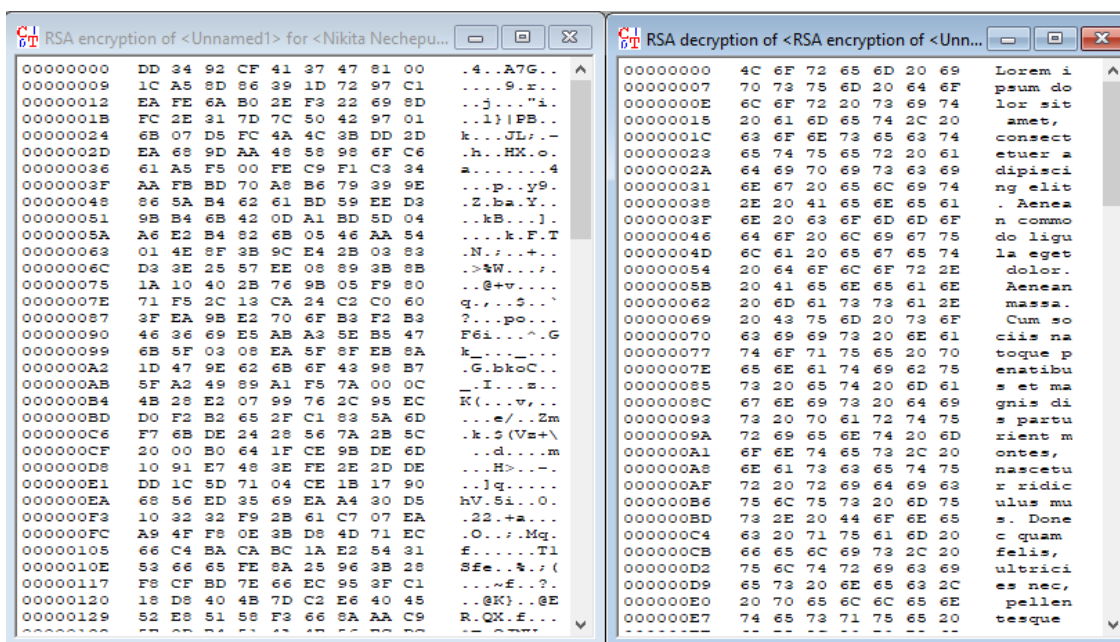


Рисунок 11 – Шифровка и дешифровка текста парой ключей длиной 512 бит

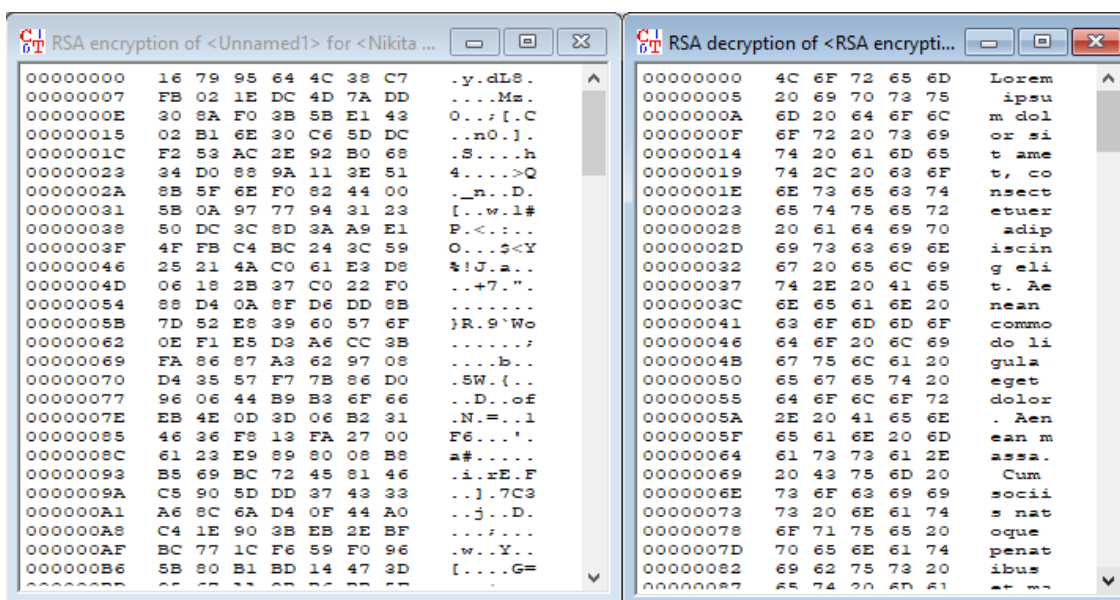


Рисунок 12 – Шифровка и дешифровка текста парой ключей длиной 768 битов

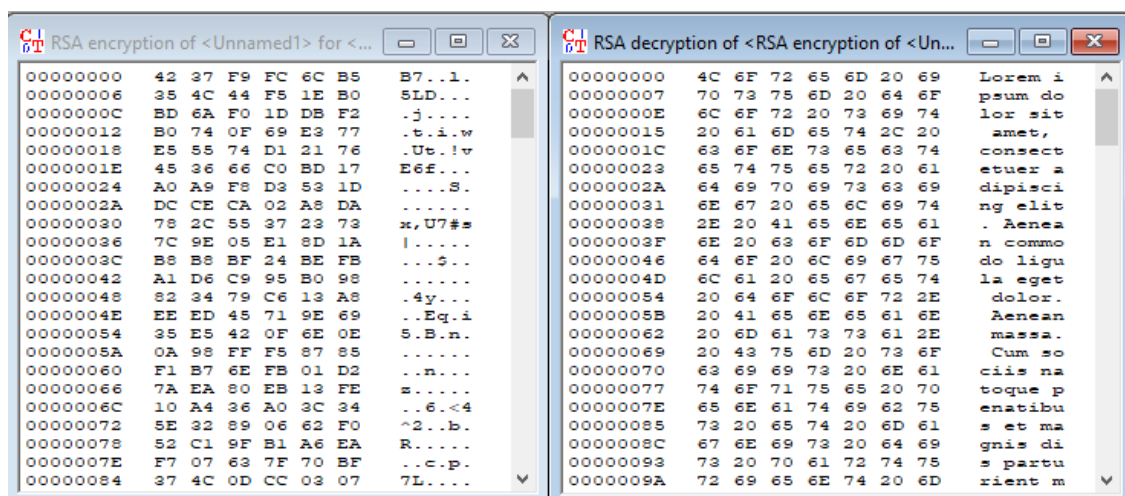


Рисунок 13 – Шифровка и дешифровка текста парой ключей длиной 1024 бита

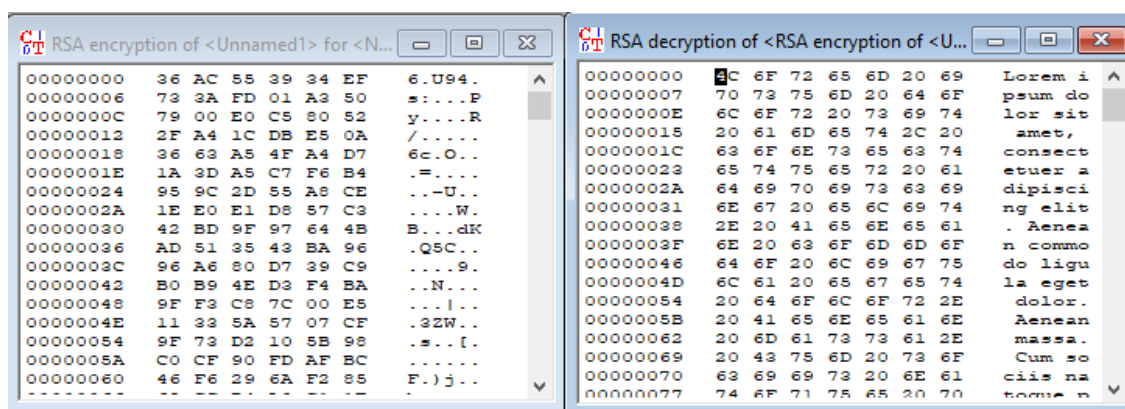


Рисунок 14 – Шифровка и дешифровка текста парой ключей длиной 2048 битов

Зависимость временных затрат на шифровку и дешифровку приведена в таблице 1.

Таблица 1 – Время шифрования/дешифрования в зависимости от размера ключа

Размер ключа, бит	Шифровка, секунд	Дешифровка, секунд
512	0.0	0.0

768	0.0	0.010
1024	0.0	0.010
2048	0.0	0.031

Выводы.

Были сгенерированы 4 пары ключей: 512, 768, 1024 и 2048 битов длиной. Исходный текст длиной примерно 1000 символов был успешно зашифрован и дешифрован. Временные затраты на шифровку оказались меньше точности представления чисел во встроенной в Cryptool 1 утилите. Дешифровка заняла чуть больше времени, но все равно это порядок сотых долей секунды, что приемлемо для использования данного шифра.

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 $n=pq > 256$.
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 -> RSA Cryptosystem -> RSA Demonstration 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 RSA в Cryptool 1.

Коллеге был передан открытый ключ $e = 2^{16} + 1$, модуль $N = 35237$ и шифротекст: 32371 # 18289 # 17453 # 32371 # 06570 # 31629 # 25208 # 00161 # 05924 # 10136 # 21969 # 06260 # 32371 # 17453 # 02556.

Интерфейс утилиты атаки методом грубой силы с использованием факторизации N приведен на рисунке 15.

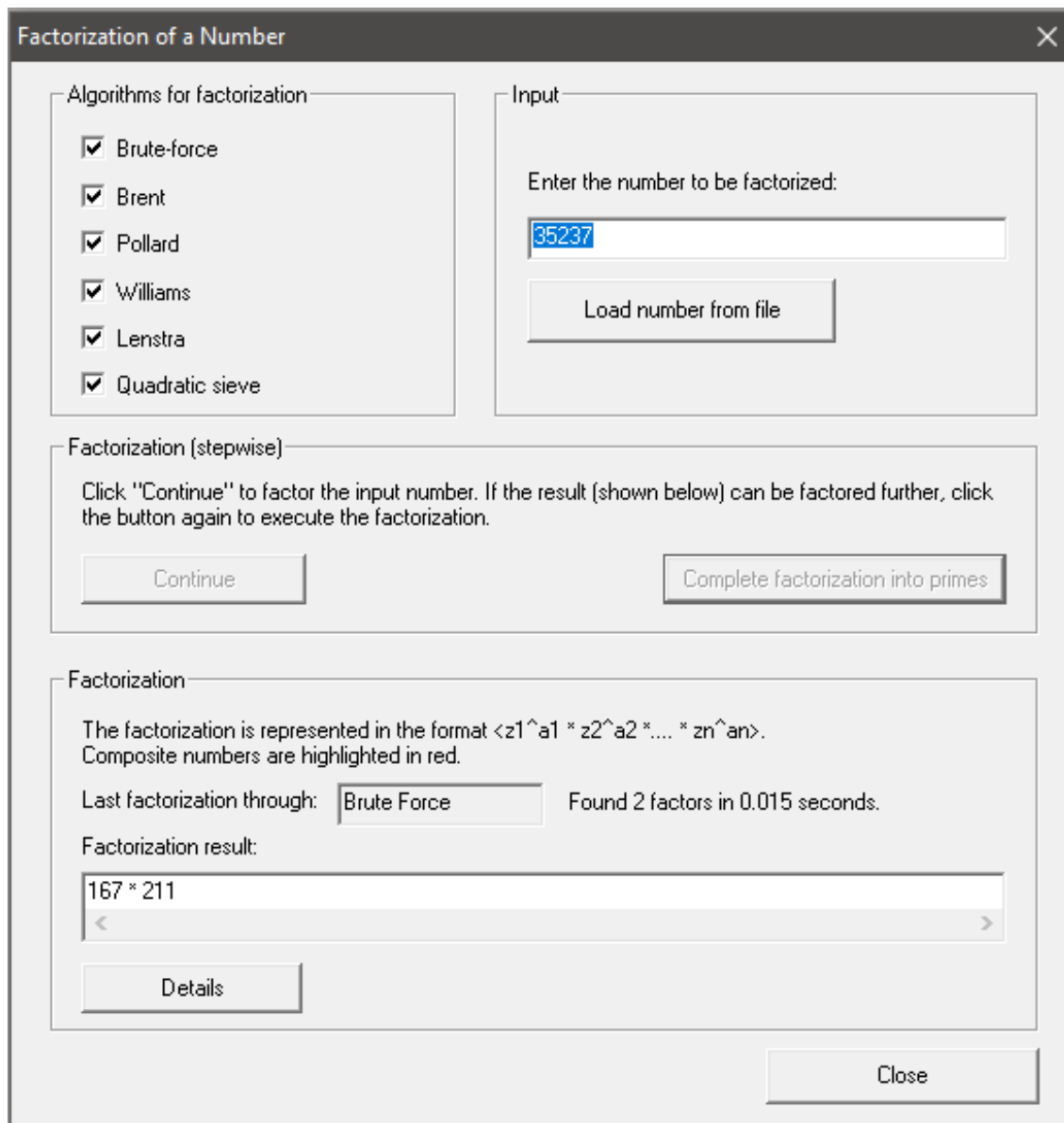


Рисунок 15 – Атака методом грубой силы, факторизация N

С помощью утилиты были получены p и q , равные 167 и 211 соответ-

ственно. Дешифруем полученное от коллеги сообщение (см. рис. 16).

The screenshot shows the 'RSA Demonstration' application window. It has a title bar with a close button. The main content area is divided into several sections:

- RSA using the private and public key -- or using only the public key**: Two radio buttons. The first is selected: 'Choose two prime numbers p and q. The composite number $N = pq$ is the public RSA modulus, and $\phi(N) = (p-1)(q-1)$ is the Euler totient. The public key e is freely chosen but must be coprime to the totient. The private key d is then calculated such that $d = e^{-1} \pmod{\phi(N)}$ '. The second option is 'For data encryption or certificate verification, you will only need the public RSA parameters: the modulus N and the public key e.'
- Prime number entry**: Two input fields for 'Prime number p' (value: 167) and 'Prime number q' (value: 211). A 'Generate prime numbers...' button is to the right.
- RSA parameters**: Four input fields: 'RSA modulus N' (35237, labeled '(public)'), ' $\phi(N) = (p-1)(q-1)$ ' (34860, labeled '(secret)'), 'Public key e' ($2^{16}+1$), and 'Private key d' (27893). An 'Update parameters' button is to the right.
- RSA encryption using e / decryption using d [alphabet size: 256]**:
 - 'Input as' with radio buttons for 'text' and 'numbers' (selected).
 - 'Alphabet and number system options...' button.
 - 'Ciphertext coded in numbers of base 10': A text area containing '3 # 32371 # 06570 # 31629 # 25208 # 00161 # 05924 # 10136 # 21969 # 06260 # 32371 # 17453 # 02556'.
 - 'Decryption into plaintext $m[i] = c[i]^d \pmod{N}$ ': A text area containing '00067 # 00082 # 00065 # 00067 # 00075 # 00077 # 00069 # 00073 # 00070 # 00089 # 00079 # 00085 # 0'.
 - 'Output text from the decryption (into segments of size 1; the symbol '#' is used as separator)': A text area containing 'C # R # A # C # K # M # E # I # F # Y # O # U # C # A # N'.
 - 'Plaintext': A text area containing 'CRACKMEIFYOUCAN'.

At the bottom, there are three buttons: 'Encrypt', 'Decrypt' (highlighted with a dashed border), and 'Close'.

Рисунок 16 – Дешифровка полученного шифротекста

Атака была успешно проведена.

Выводы.

Была успешно проведена атака методом грубой силы на шифротекст с использованием факторизации модуля. Для небольших сомножителей эта

операция может выполняться за приемлемое время, поэтому в современных системах используются большие простые числа, порядка 300 десятичных цифр, что делает атаку методом грубой силы практически невозможной за приемлемое время.

Имитация атаки на гибридную криптосистему.

Задание.

1. Подготовьте текст передаваемого сообщения на английском с вашим именем в конце.
2. Запустите утилиту Analysis -> Asymmetric Encr...-> 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.

Основные теоретические положения.

Модель гибридной криптосистемы, асимметричная составляющая которой использует асимметричный шифр (например RSA) представлена на рисунке 17.

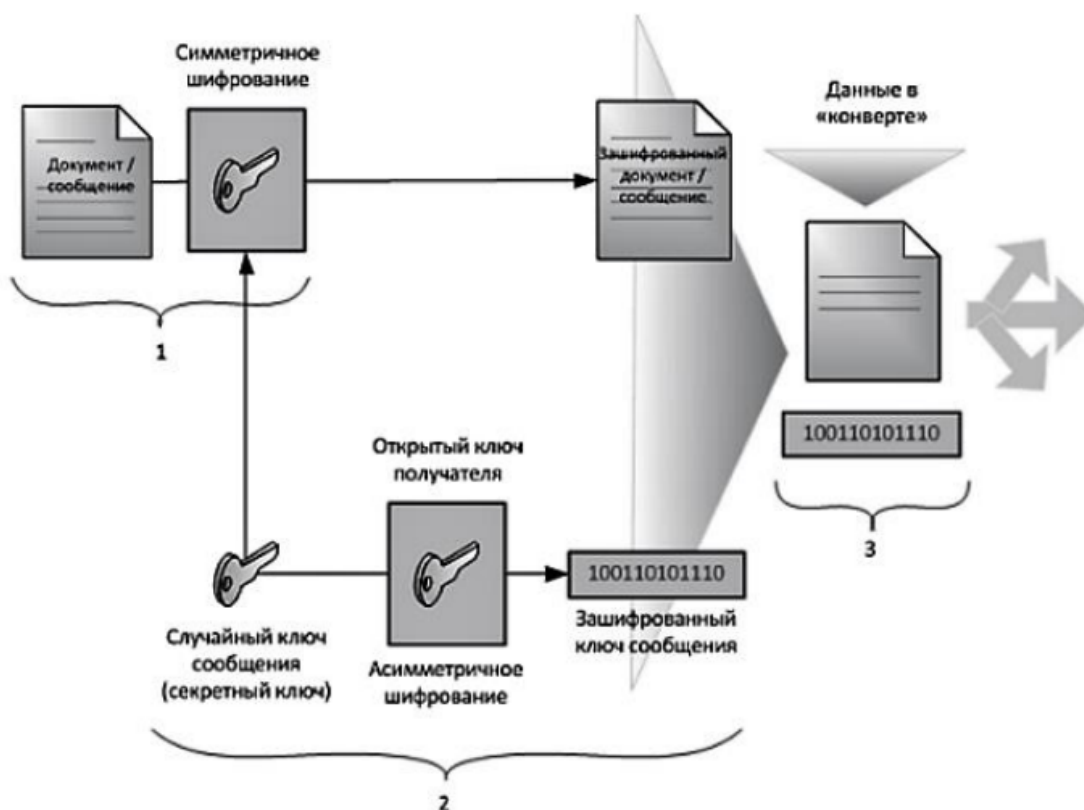


Рисунок 17 – Модель гибридной криптосистемы

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.

Имитация атаки на гибридную криптосистему в Cryptool 1.

Протокол работы атаки приведен на рисунке 18.

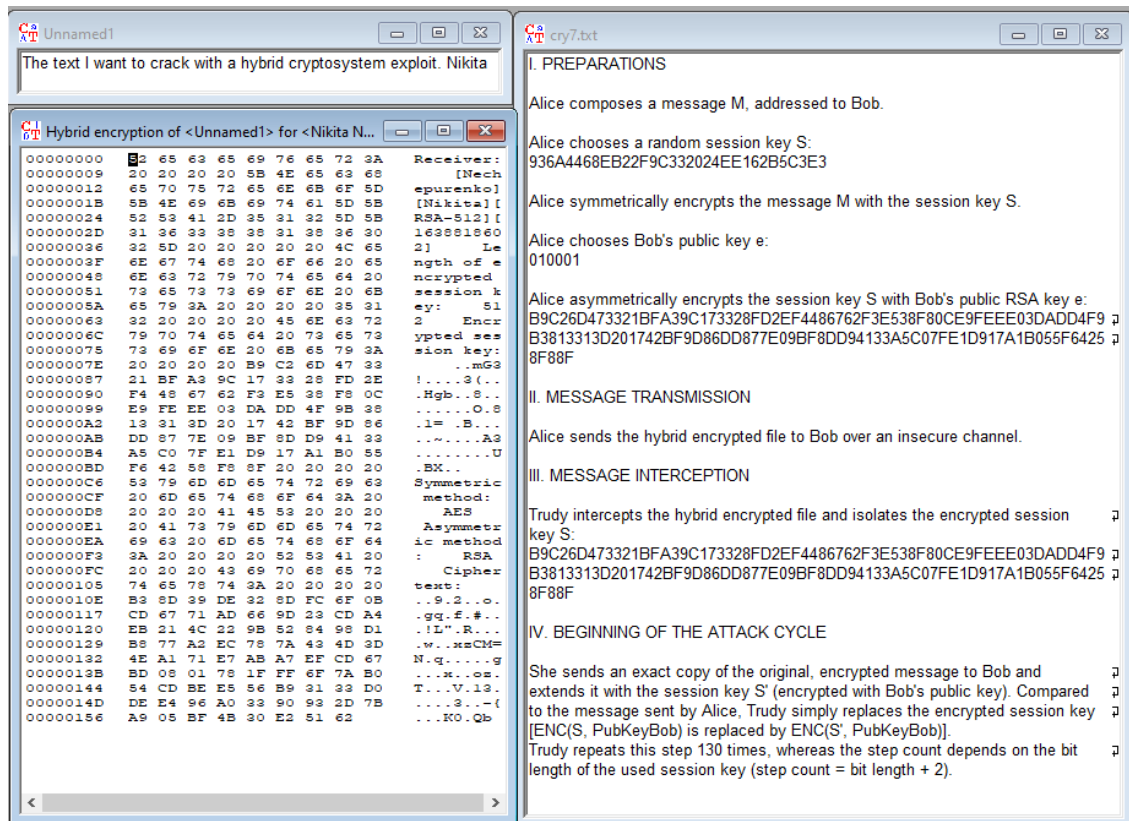


Рисунок 18 – Атака на гибридную криптосистему в Cryptool 1.

Заключение.

1. Был изучен протокол Диффи-Хеллмана, позволяющий используя свойства малой теоремы Ферма обмениваться открытыми ключами по незащищенному каналу, генерируя общий закрытый ключ шифрования. Для работы протокола выбираются открытые параметры: большое простое число p , g - первообразный корень по модулю p . Стороны генерируют числа x и y , используя которые вычисляются открытые ключи сторон: $R_1 = g^x \bmod p$ и $R_2 = g^y \bmod p$ соответственно. Общий закрытый ключ вычисляется как $K = R_1^y \bmod p = R_2^x \bmod p$. Средствами Cryptool 1 было сгенерировано большое простое число p , его первообразный корень g , а также две закрытые части ключа. Затем был вычислен общий ключ, с помощью которого был зашифрован, а затем расшифрован некоторый открытый текст.

2. Был изучен асимметричный блочный шифр RSA. Исходный текст представляется числами от 0 до $n-1$ для некоторого n . Генерируются 2 больших числа p и q , вычисляется $n = pq$. Затем выбирается произвольное $e (e < n)$, взаимно простое с $\phi(n)$ – функцией Эйлера. Вычисляется закрытый ключ $d : e \cdot d = 1 \bmod \phi(n)$. Числа (e, n) – открытый ключ, p и q уничтожаются. Для шифровки открытый текст разбивается на блоки $m_i : m_i < n$ и преобразуется по формуле $c_i = m_i^e \bmod n$. Для дешифровки используется формула $m_i = c_i^d \bmod n$. С помощью утилиты Cryptool 1 были сгенерированы открытый и закрытый ключи, а также проведено шифрование и дешифрование строки, состоящей из ФИО.

3. Были сгенерированы 4 пары ключей: 512, 768, 1024 и 2048 битов длиной. Исходный текст длиной примерно 1000 символов был успешно зашифрован и дешифрован. Временные затраты на шифровку оказались меньше точности представления чисел во встроенной в Cryptool 1 утилите. Дешифровка заняла чуть больше времени, но все равно это порядок сотых долей секунды,

что приемлемо для использования данного шифра.

4. Была успешно проведена атака методом грубой силы на шифротекст с использованием факторизации модуля. Для небольших сомножителей эта операция может выполняться за приемлемое время, поэтому в современных системах используются большие простые числа, порядка 300 десятичных цифр, что делает атаку методом грубой силы практически невозможной за приемлемое время.

5. Средствами Cryptool 1 была успешно произведена атака на гибридную криптосистему. В таких системах сообщение шифруется симметричным ключом, а симметричный ключ открытым ключом получателя. После этого происходит передача конверта, который состоит из зашифрованного сообщения и зашифрованного симметричного ключа. Атака на модель гибридной криптосистемы основана на том, что злоумышленник сначала перехватывает конверт, затем, модифицирует шифровку ключа из конверта и побитово восстанавливает зашифрованный секретный ключ, анализируя положительные и отрицательные ответы сервера.