

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

ОТЧЕТ
по лабораторной работе №4
по дисциплине «Криптографические методы защиты информации»
Тема: Изучение асимметричных протоколов и шифров

Студентка гр. 9363

Труханова В.А.

Преподаватель

Племянников А.К.

Санкт-Петербург

2023

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.

Протокол Диффи-Хеллмана

7.1.1. Задание

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

7.1.2. Основные параметры и схема протокола

Протокол Диффи-Хеллмана является первым из опубликованных алгоритмов на основе открытых ключей. Обычно данный алгоритм называют обменом ключами по схеме Диффи-Хеллмана.

Цель схемы – обеспечить двум пользователям защищенную возможность получения симметричного секретного ключа.

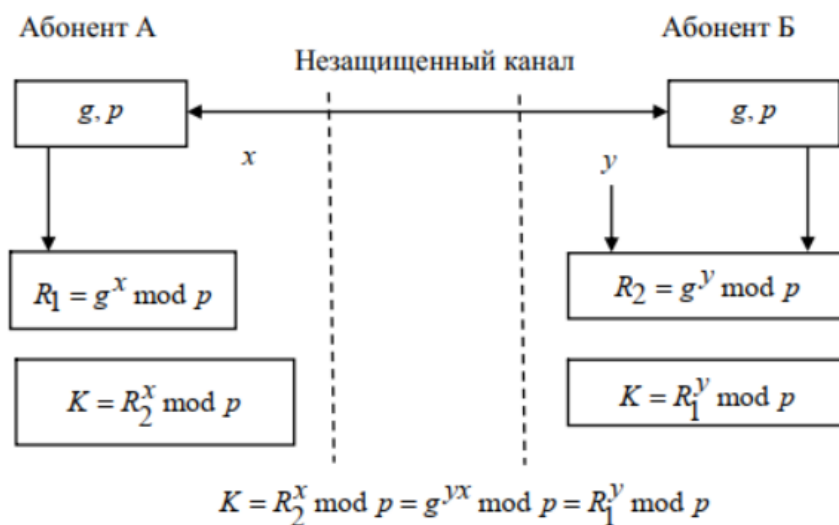


Рисунок 1- Схема протокола Диффи-Хеллмана

Протокол Диффи–Хеллмана состоит из следующих операций (рис. 7.1):

- Устанавливаются открытые параметры p, g :
 - а) p – большое простое число порядка 300 десятичных цифр (1024 бит);
 - б) g – первообразный корень по модулю p .
- Каждая из сторон генерирует закрытый ключ – большое число x и y соответственно.

- На каждой стороне вычисляется открытый ключ:

а) $R_1 = g^x \bmod p,$

б) $R_2 = g^y \bmod p.$

Стороны обмениваются открытыми ключами и вычисляют общие данные K для создания симметричного ключа: $K = R_2^x \bmod p = R_1^y \bmod p.$

7.1.3. Демонстрация работы протокола

Публичные параметры автоматически генерируются генератором с длиной 56 бит:

$$p = 78774425755274699, g = 25044468145950442;$$

$$x = 58841441568987774, y = 58968417830920481.$$

На каждой стороне вычисляется открытый ключ:

$$R_1 = 72408097483342572, R_2 = 20337679304053212.$$

И на последнем шаге генерируем симметричный ключ:

$$K = R_2^x \bmod p = R_1^y \bmod p = 563188396319348.$$

Схема протокола, реализованная в Cryptool

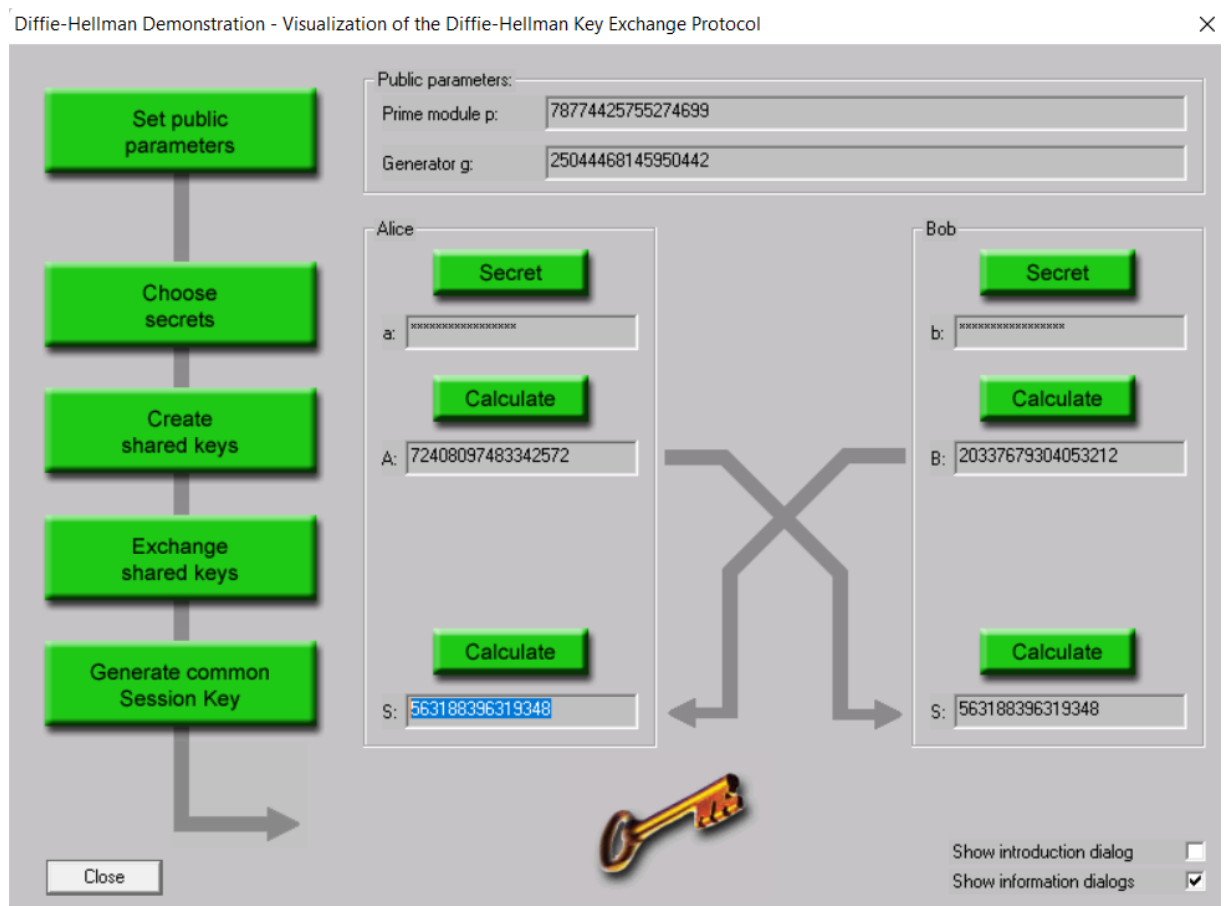
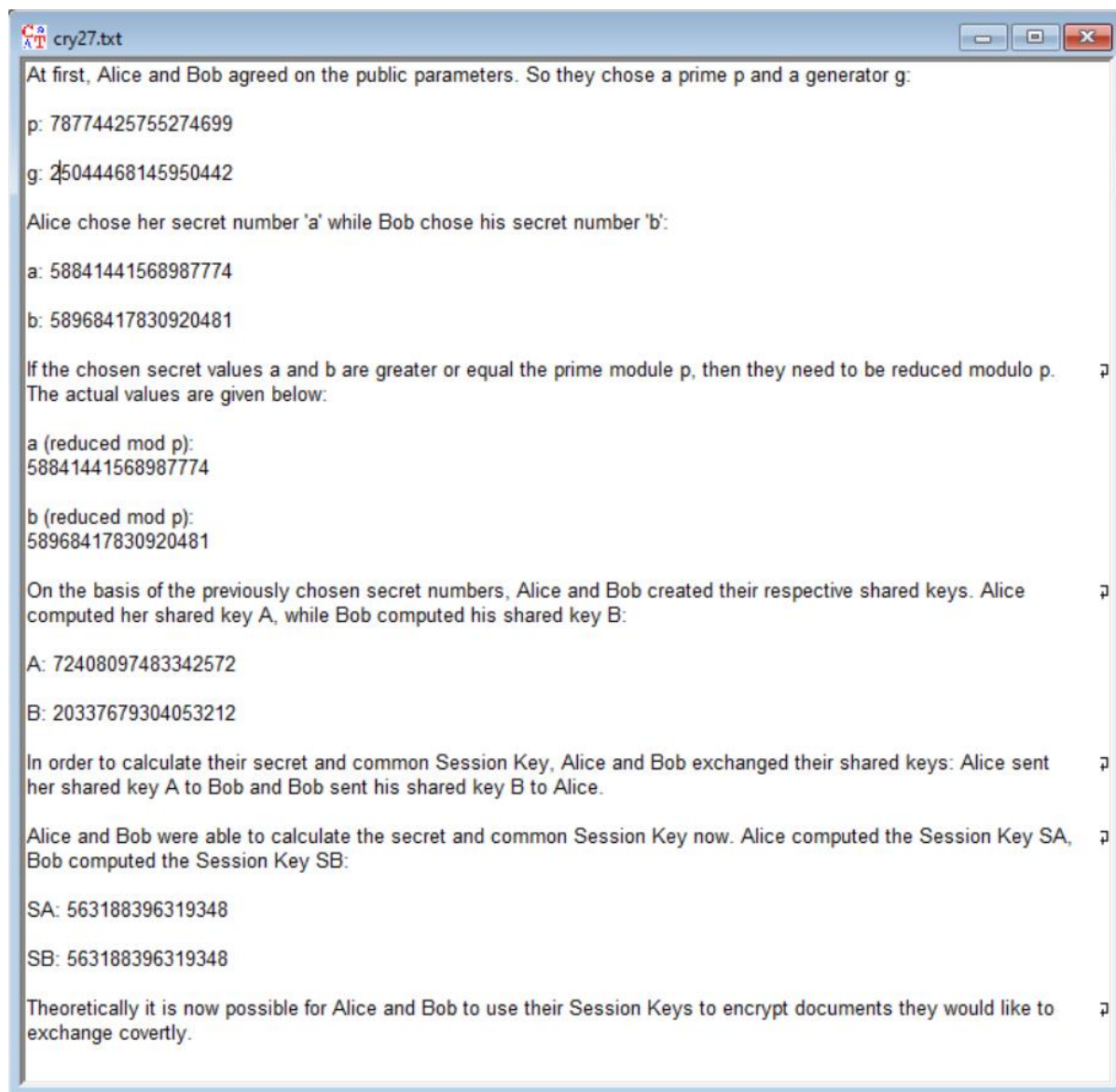


Рисунок 2 – Окно «Diffie-Hellman demonstration»



```
At first, Alice and Bob agreed on the public parameters. So they chose a prime p and a generator g:
p: 78774425755274699
g: 25044468145950442

Alice chose her secret number 'a' while Bob chose his secret number 'b':
a: 58841441568987774
b: 58968417830920481

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):
58841441568987774

b (reduced mod p):
58968417830920481

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: 72408097483342572
B: 20337679304053212

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: 563188396319348
SB: 563188396319348

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

Рисунок 3 – Лог-файл протокола Диффи-Хеллмана

7.1.4. Таблица соответствия демонстрации протокола и параметров протокола

Таблица 1. Соответствие параметров

Параметр протокола	Cryptool
Открытые параметры	p, g
Открытый ключ	R_1, R_2
Секрет (Закрытый ключ)	x, y
Общий секретный ключ	K

7.1.5. Исходный, зашифрованный и расшифрованный тексты

$$K = 563188396319348 = 02\ 00\ 37\ 84\ 4D\ E2\ 74.$$

The screenshot displays two windows from the CrypTool 1 (CT1) application. The top window, titled 'startingexample-en.txt', shows the results of a DES (ECB) encryption. It includes a hex dump of the encrypted data, which is a 16-byte block. The bottom window, titled 'DES (ECB) decryption of <DES (ECB) encryption of <startingexamp...', shows the results of the decryption. It displays the original text: 'Starting example for the CrypTool version family 1.x (CT1)' and a remark about successor versions CT2, JCT, and CTO.

Top Window: startingexample-en.txt

Starting example for the CrypTool version family 1.x (CT1)

Remark:
The successor versions of CT1 (called CT2, JCT and CTO) now offer a significantly wider range of functionality than CT1. In CT1 only errors will be corrected. Please use the newer versions of CrypTool little by little.

DES (ECB) encryption of <startingexample-en.txt>, key <00 02 00 3...

00000000	C5 DC 72 2B 04 29 59 29 51 E6 C0	..r+.)Y)Q..
0000000B	D6 29 79 8F 38 B5 E6 7D 81 C6 96	.)y.8...}...
00000016	ED E1 25 E0 F2 68 78 37 F8 F9 49	..%..hx7...I
00000021	B6 6B 21 E3 94 27 72 BB 80 FA B8	.k!...'r....
0000002C	D1 6F 83 90 5A 6E 2E DA 4A ED 6E	..o...2n...J.n

Bottom Window: DES (ECB) decryption of <DES (ECB) encryption of <startingexamp...

Starting example for the CrypTool version family 1.x (CT1)

Remark:
The successor versions of CT1 (called CT2, JCT and CTO) now offer a significantly wider range of functionality than CT1. In CT1 only errors will be corrected. Please use the newer versions of CrypTool little by little.

CrypTool 1 (CT1) is a comprehensive and free educational program about cryptography and cryptanalysis offering extensive online help and many visualizations.

Рисунок 4 – Результат шифровки и дешифровки

Шифр RSA

7.2.1. Задание

- Запустите утилиту *Indiv. Procedures* \rightarrow *RSA Cryptisystem* \rightarrow *RSA Demonstration...*;
- Задайте в качестве обрабатываемого сообщения своё Ф.И.О.;
- Сгенерируйте открытый и закрытый ключи;
- Зашифруйте сообщение. Сохраните скриншот результата;
- Расшифруйте сообщение. Сохраните скриншот результата;
- Убедитесь, что расшифрование произошло корректно;

7.2.2. Обобщенная схема протокола шифрования RSA

Алгоритм RSA представляет собой асимметричный блочный шифр, в котором блоки открытого и зашифрованного сообщений представляются целыми числами из диапазона от 0 до $n - 1$ для блока размером $\log_2 n$ бит.

Алгоритм шифрования RSA состоит из следующих операций:

1. Вычисление ключей:

а) генерируются два больших простых числа p и q (держатся в секрете);

б) вычисляется $n = p \times q$;

в) выбирается произвольное число e ($e < n$), взаимно простого с $\varphi(n)$ (функцией Эйлера);

г) вычисляется число d : $d \times e \equiv 1 \mod \varphi(n)$;

д) числа (e, n) составляют открытый ключ, d – закрытый ключ, p и q уничтожаются.

2. Зашифрование:

а) открытый текст разбивается на блоки (числа) m_i : $m_i < n$;

б) каждый блок открытого текста преобразуется в шифротекст по формуле:

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

3. Расшифрование:

а) шифротекст представляется блоками (числами) c_i : $c_i < n$;

б) каждый блок шифротекста преобразуется в открытый текст по формуле:

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

Обобщенная схема шифра RSA представлена на рисунках 5.

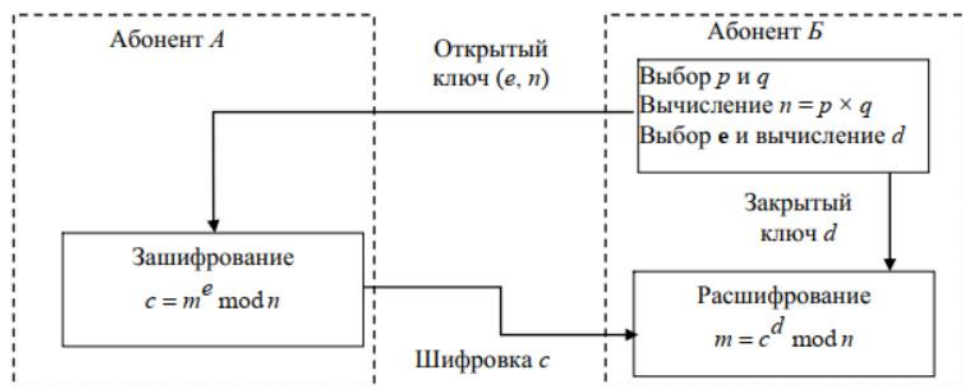


Рисунок 5 – Обобщенная схема шифра RSA

7.2.3. Результат генерации ключей, зашифровки и расшифровки

Запущена утилита *RSA Demonstration*.... Диалоговое окно представлено на рисунке 6.

The screenshot shows the 'RSA Demonstration' dialog box. It has a title bar with a close button. The main area contains two radio buttons for selecting the operation: 'Choose two prime numbers p and q...' (selected) and 'For data encryption or certificate verification...'. Below this is the 'Prime number entry' section with fields for 'Prime number p' and 'Prime number q', and a 'Generate prime numbers...' button. The 'RSA parameters' section has fields for 'RSA modulus N', 'phi(N) = (p-1)(q-1)', 'Public key e' (containing '2^16+1'), and 'Private key d'. There are 'Update parameters' and 'Close' buttons. The bottom section is for 'RSA encryption using e / decryption using d' with a text input area and 'Encrypt' and 'Decrypt' buttons.

Рисунок 6 – Окно «*RSA Demonstration*»

На рисунке 7 представлен результат генерации ключей.

This screenshot shows the same 'RSA Demonstration' dialog box after key generation. The 'Prime number p' field now contains '139' and 'Prime number q' contains '239'. The 'RSA modulus N' field contains '33221' and is labeled '(public)'. The 'phi(N) = (p-1)(q-1)' field contains '32844' and is labeled '(secret)'. The 'Public key e' field still contains '2^16+1'. The 'Private key d' field now contains '11093'. The 'Update parameters' button is visible. The encryption/decryption section at the bottom remains empty.

Рисунок 7 – Генерация ключей

С помощью данной утилиты было проведено зашифрование сообщения при заданных параметрах. Параметры и результат шифрования приведены на рисунке 8.

The screenshot shows the 'RSA Demonstration' window with the following sections:

- Header:** RSA Demonstration (with a close button 'X')
- Instructions:**
 - ☒ 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: 131
 - Prime number q: 223
 - Generate prime numbers... button
- RSA parameters:**
 - RSA modulus N: 29213 (public)
 - $\phi(N) = (p-1)(q-1)$: 28860 (secret)
 - Public key e: $2^{16}+1$
 - Private key d: 15233
 - Update parameters button
- RSA encryption using e / decryption using d [alphabet size: 27]:**
 - Input as: ☒ text ☐ numbers
 - Alphabet and number system options... button
 - Input text: TRUKHANOVA VERONIKA
 - The Input text will be separated into segments of Size 3 (the symbol '#' is used as separator).
 - TRU # KHA # NOV # A V # ERO # NIK # A
 - Numbers input in base 10 format.
 - 15087 # 08236 # 10633 # 00751 # 04146 # 10460 # 00729
 - Encryption into ciphertext $c[i] = m[i]^e \pmod{N}$
 - 18559 # 11564 # 23930 # 00512 # 15969 # 09918 # 27051
- Buttons:** Encrypt, Decrypt, Close

Рисунок 8 – Результат шифрования

На рисунке 9 представлен результат дешифрования текста.

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: 131

Prime number q: 223

Generate prime numbers...

RSA parameters

RSA modulus N: 29213 (public)

$\phi(N) = (p-1)(q-1)$: 28860 (secret)

Public key e: $2^{16}+1$

Private key d: 15233

Update parameters

RSA encryption using e / decryption using d [alphabet size: 27]

Input as ☐ text ☒ numbers

Alphabet and number system options...

Ciphertext coded in numbers of base 10

18559 # 11564 # 23930 # 00512 # 15969 # 09918 # 27051

Decryption into plaintext $m[i] = c[i]^d \pmod{N}$

15087 # 08236 # 10633 # 00751 # 04146 # 10460 # 00729

Output text from the decryption (into segments of size 3; the symbol '#' is used as separator).

TRU # KHA # NOV # A V # ERO # NIK # A

Plaintext

TRUKHANOVA VERONIKA

Encrypt Decrypt Close

Рисунок 9 – Результат дешифрования

Исследование шифра RSA

7.3.1. Задание

- Выбрать текст на английском языке (не менее 1000 знаков) и сохранить в файле формата *.txt;
- Сгенерировать пары ассиметричных RSA-ключей утилитой *Digital Signatures* \rightarrow *PKI* \rightarrow *Generate/Import Keys* с различными длинами (4 варианта);
- Зашифровать текст (примерно 1000 символов) различными открытыми ключами. Зафиксировать время зашифровки;
- Расшифровать текст различными закрытыми ключами. Зафиксировать время зашифровки;

- Проверить корректность расшифровки. Зафиксировать скриншоты результата.

7.3.2. Выбранный текст

На Рисунке 10 представлен текст на английском языке в формате .TXT.
(Размер текста: 1115 символов)

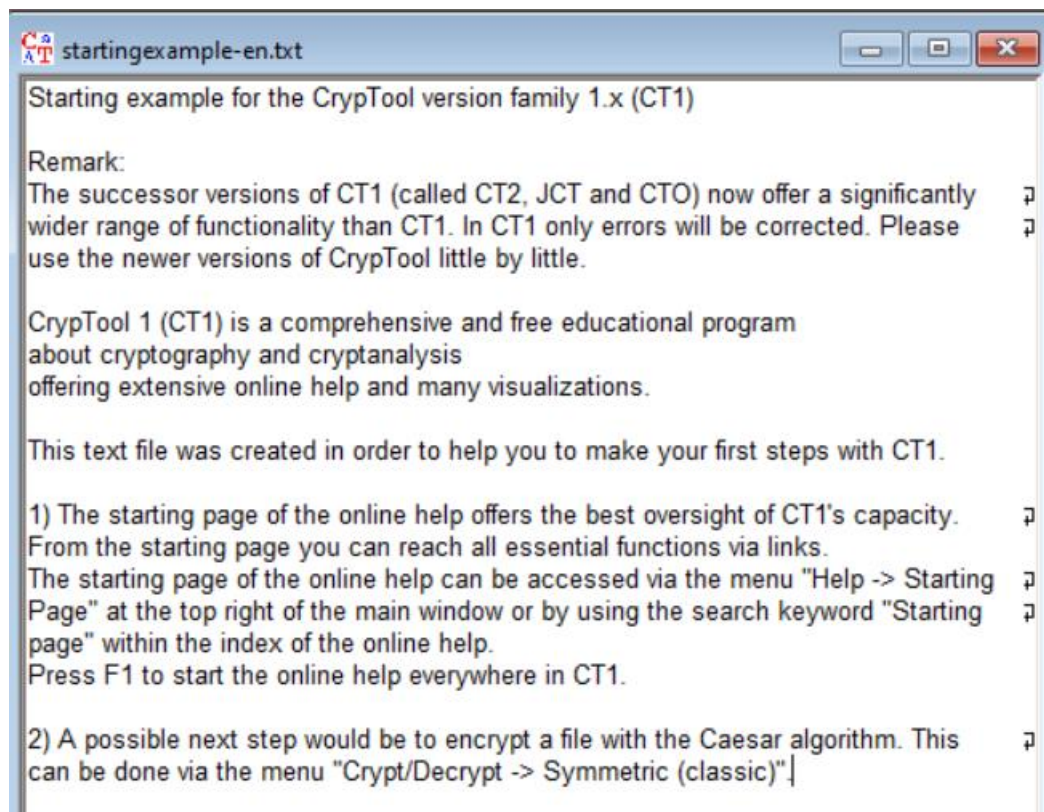


Рисунок 10 – Исходный текст

7.3.3. Результаты генерации ключевых пар различной длины

Сгенерированы пары ассиметричных RSA-ключей утилитой *Digital Signatures* → *PKI* → *Generate/Import Keys* с различными длинами (512, 768, 1024, 2048 бит). Результаты представлены на рисунке 11-14.

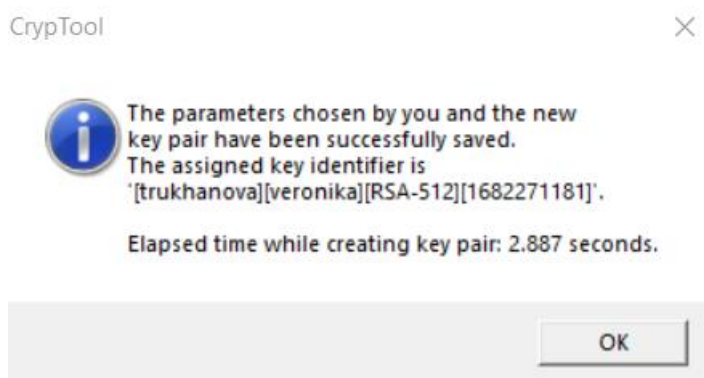


Рисунок 11 – Генерация ключа длиной 512 бит

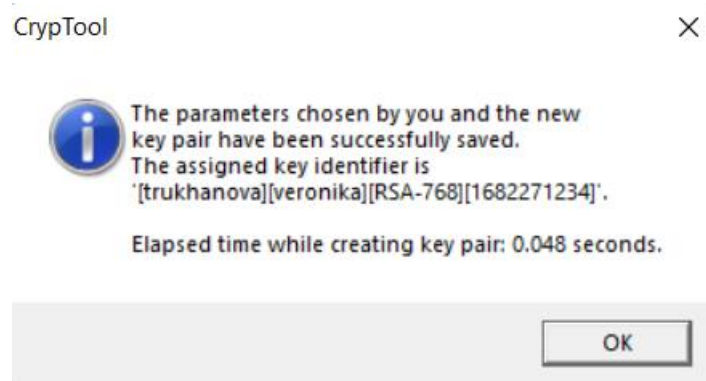


Рисунок 12 – Генерация ключа длиной 768 бит

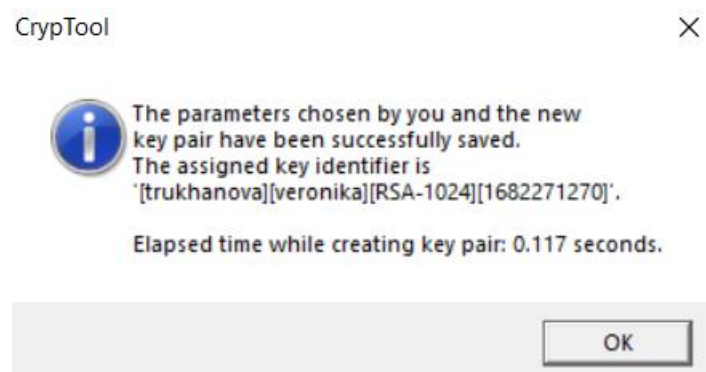


Рисунок 13 – Генерация ключа длиной 1024 бит

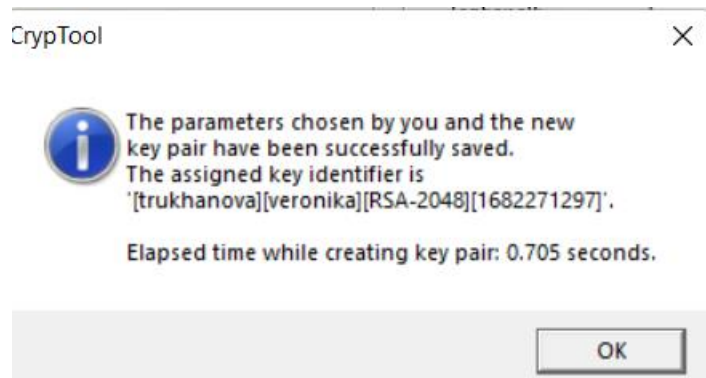


Рисунок 14 – Генерация ключа длиной 2048 бит

7.3.4. Шифровка и расшифровка текста ключами разной длины

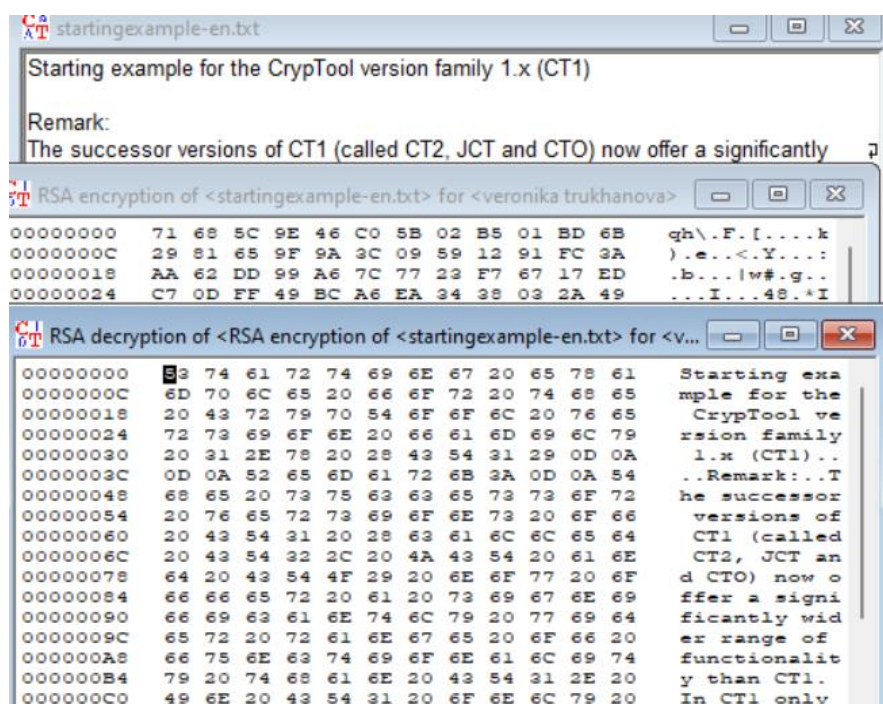


Рисунок 15 – Результаты шифровки и расшифровки исходного текста
(512)

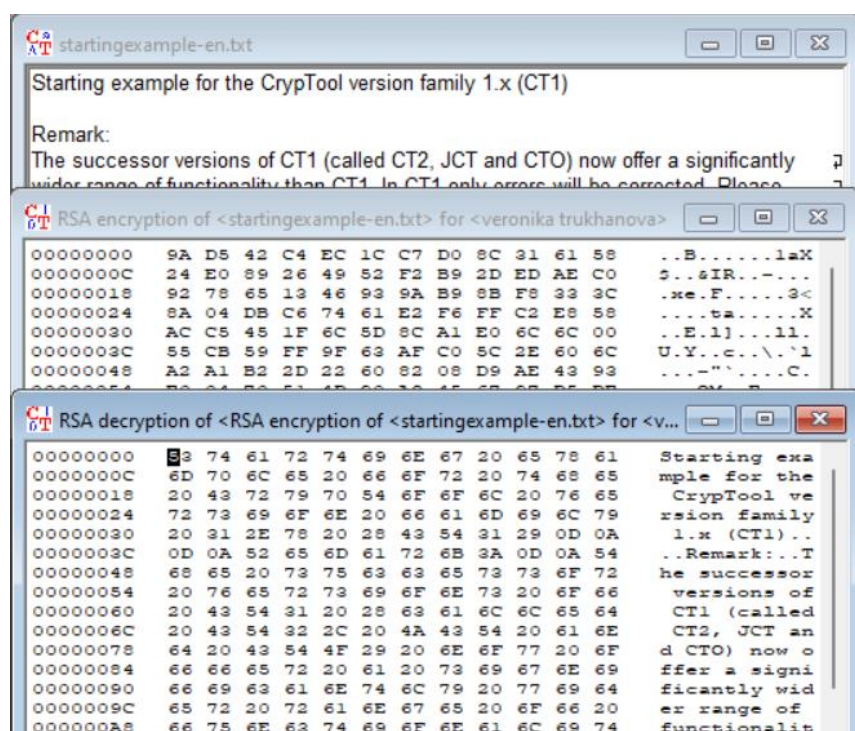


Рисунок 16 – Результаты шифровки и расшифровки исходного текста
(768)

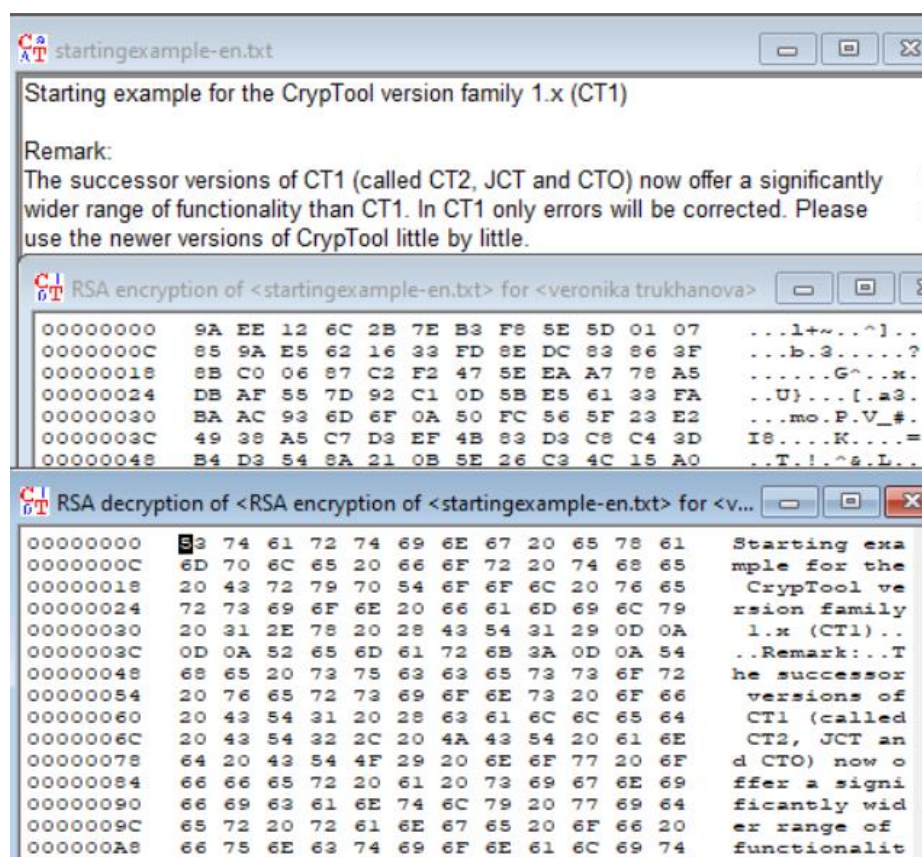


Рисунок 14 – Результаты шифровки и расшифровки исходного текста (1024)

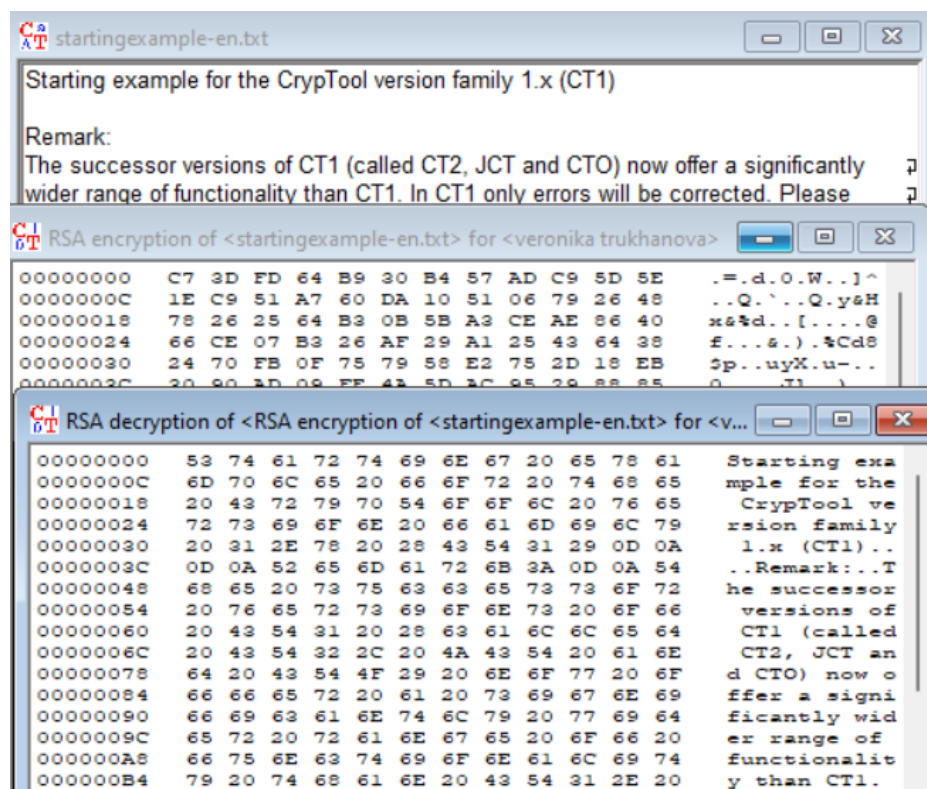


Рисунок 15 – Результаты шифровки и расшифровки исходного текста (2048)

7.3.5. Временные затраты на зашифровку и расшифровку

Таблица 2. Временные затраты

Длина ключа	Время на зашифровку, сек	Время на расшифровку, сек
512	0.000	0.006
768	0.000	0.011
1024	0.000	0.014
2048	0.002	0.060

Атака «грубой силы» на RSA

7.4.1. Задание

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

7.4.2. Исходные данные для атаки

От коллеги были получены следующие данные:

$n = 299, e = 101$.

Шифротекст:

076 # 101 # 116 # 039 # 115 # 032 # 103 # 111 # 032 # 098 # 097 # 107 # 101
032 # 115 # 111 # 109 # 101 # 032 # 115 # 117 # 103 # 097 # 114 # 032 # 099 # 111
111 # 107 # 105 # 101 # 115 # 033

020 # 173 # 116 # 026 # 046 # 288 # 155 # 076 # 288 # 128 # 067 # 074 # 173
288 # 046 # 076 # 148 # 173 # 288 # 046 # 234 # 155 # 067 # 160 # 288 # 112 # 076
076 # 074 # 261 # 173 # 046 # 245

7.4.2. Результат факторизации

Выполнена факторизация модуля n . Результат представлен на рисунке 16.

Factorization of a Number

Algorithms for factorization

- ☒ Brute-force
- ☒ Brent
- ☒ Pollard
- ☒ Williams
- ☒ Lenstra
- ☒ Quadratic sieve

Input

Enter the number to be factorized:

Factorization (stepwise)

Click "Continue" to factor the input number. If the result (shown below) can be factored further, click the button again to execute the factorization.

Factorization

The factorization is represented in the format $\langle z_1^{a_1} * z_2^{a_2} * \dots * z_n^{a_n} \rangle$.
Composite numbers are highlighted in red.

Last factorization through: Found 2 factors in 0.004 seconds.

Factorization result:

Рисунок 16 – Результат факторизации

7.4.3. Расшифрованное сообщение

Используя полученный результат была проведена расшифровка сообщения. Результат представлен на рисунке 17.

Имитация атаки на гибридную криптосистему

Модель гибридной криптосистемы, асимметричная составляющая которой использует асимметричный шифр (например, RSA), представлена на рисунке 18.

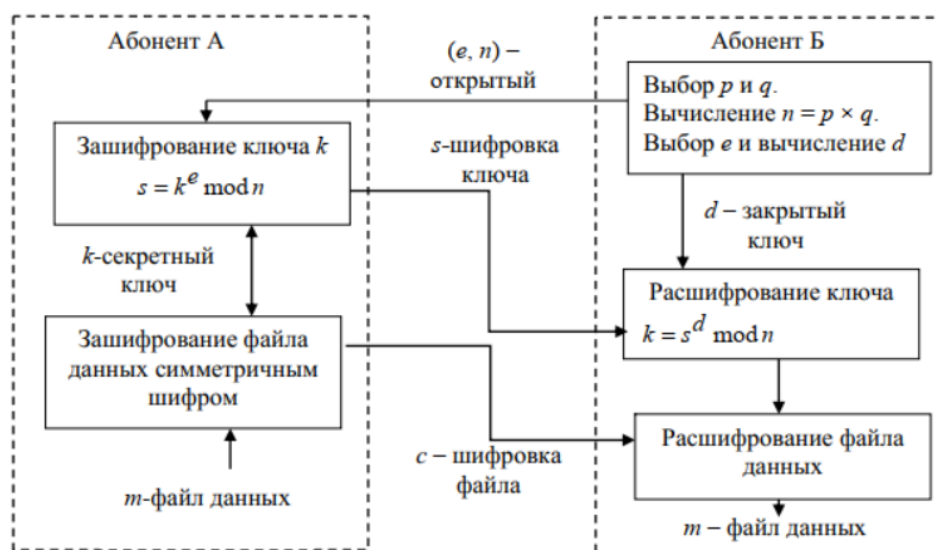


Рисунок 18 – Модель гибридной криптосистемы

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.

Цель атак – определить симметричный секретный ключ, зашифрованный открытым ключом криптосистемы.

Условия атаки:

- Нарушитель может перехватывать сообщения, адресованные серверу;
- Нарушитель может модифицировать сообщения и направлять их серверу;
- Сервер не определяет, от кого был получен конверт;

- Нарушитель может классифицировать ответы сервера на ПРИНЯТО/ОТКЛОНЕНО, т.е. случаи успешной и неуспешной расшифровки (по распознаванию ключевого слова).

7.5.1. Задание

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

7.5.2. Описание цели атаки, модель злоумышленника, схема атакуемого протокола гибридного шифрования

Описание цели атаки:

Цель атаки – определить симметричный секретный ключ, зашифрованный открытым ключом криптосистемы, при условии, что:

- Нарушитель может перехватывать сообщения, адресованные серверу;
- Нарушитель может модифицировать сообщения и направлять их серверу;
- Нарушитель может классифицировать ответы сервера на ПРИНЯТО/ОТКЛОНЕНО.

На Рисунке 16 представлен исходный текст.

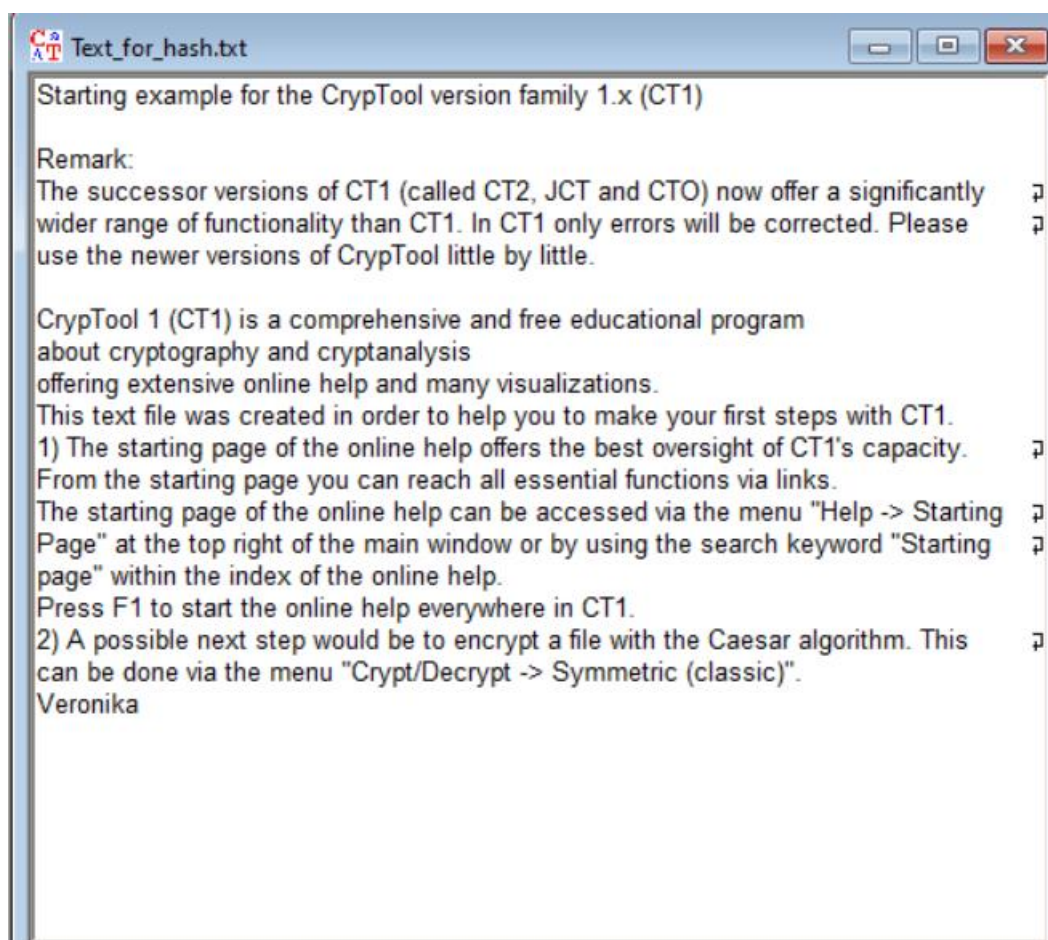



Рисунок 19 – Исходный текст

На Рисунках 20, 21 и 22 представлены лог-файлы участников протокола: Алисы, Боба и Труды соответственно.



Action log:


- Alice has composed a message for Bob
- Alice chose a random session key
- Alice has encrypted the message symmetrically with the session key
- Alice chose Bob's public RSA key e
- Alice encrypted the session key with Bob's public RSA key
- Alice sent the hybrid encrypted file to Bob

Randomly chosen session key:

D7F31074CB98E8C1850FBF1218056B51

OK

Рисунок 20 – Лог-файл Алисы



Action log:

- Bob could successfully decrypt 61 of 131 messages
- Bob received 131 messages up to now

Actually, Bob cannot decide whether the messages he received were sent by Alice or Trudy. However, given a certain keyword, Bob can decide if a message was sent by Alice. Please specify the keyword below:

Keyword:

Received session keys and decryption results:

N...	Decryption:	Decrypted session key (hexadecimal):
1	Correct	D7F31074CB98E8C1850FBF1218056B51
2	Correct	D7F31074CB98E8C1850FBF1218056B51
3	Incorrect	E9E815A8E0ED91DC95F975198B5AD72C
4	Incorrect	E9E815A8E0ED91DC95F975198B5AD72C
5	Incorrect	E9E815A8E0ED91DC95F975198B5AD72C
6	Correct	D7F31074CB98E8C1850FBF1218056B51
7	Incorrect	E9E815A8E0ED91DC95F975198B5AD72C
8	Correct	D7F31074CB98E8C1850FBF1218056B51
9	Correct	D7F31074CB98E8C1850FBF1218056B51
10	Incorrect	E9E815A8E0ED91DC95F975198B5AD72C

OK

Рисунок 21 – Лог-файл Боба



Action log:

- Trudy has intercepted the message Alice sent to Bob
- Trudy has isolated the encrypted session key from the message
- Trudy has created 130 modified session keys up to now
- 60 of 130 modified messages were successfully decrypted by Bob's server

Intercepted, encrypted session key:

```
08D8D48A6D6F3FD3A45AE7CCDD9E7EC68D73E87F97211A8FF9763DABA54AD65777F3E8D5C1391BF08B1
```

Modified and encrypted session keys:

Modified and encrypted session key (hexadecimal):

```
0EA3FA2D7FCF47D8BF55B4627AEA26872C293EE94969BC6515FA35ED60D7567C4E5B436261905ED...
C1CBE00562C90FBC4143FF316C31F2C98371F80A6E2C0E231847EA10B081A032CFDD24FC8D98E81...
D959A4E80EA12B547223EF7C92E1E4512A622E25C47B5E65EE34C12FDED3CC751440AA157C18319...
D49E422DC12453D875B2350F7CCC1C8B44D741218D43CBD79C252CD2385AFC6DC4F9045114FDF1...
05F125D2A32265C6C63D8F9553DF797BA434672888A48F944D6F066AF64617413110AD773070D94
```

Decrypted session key (calculated by Trudy, based on Bob's responses):

```
D7F31074CB98E8C1850FBF1218056B51
```

Message (calculated by Trudy using the decrypted session key):

Starting example for the CrypTool version family 1.x (CT1)

Remark:

The successor versions of CT1 (called CT2, JCT and CTO) now offer a significantly wider range of functionality than CT1. In CT1 only errors will be corrected. Please use the newer versions of CrypTool little by

OK

Рисунок 22 – Лог-файл Труди

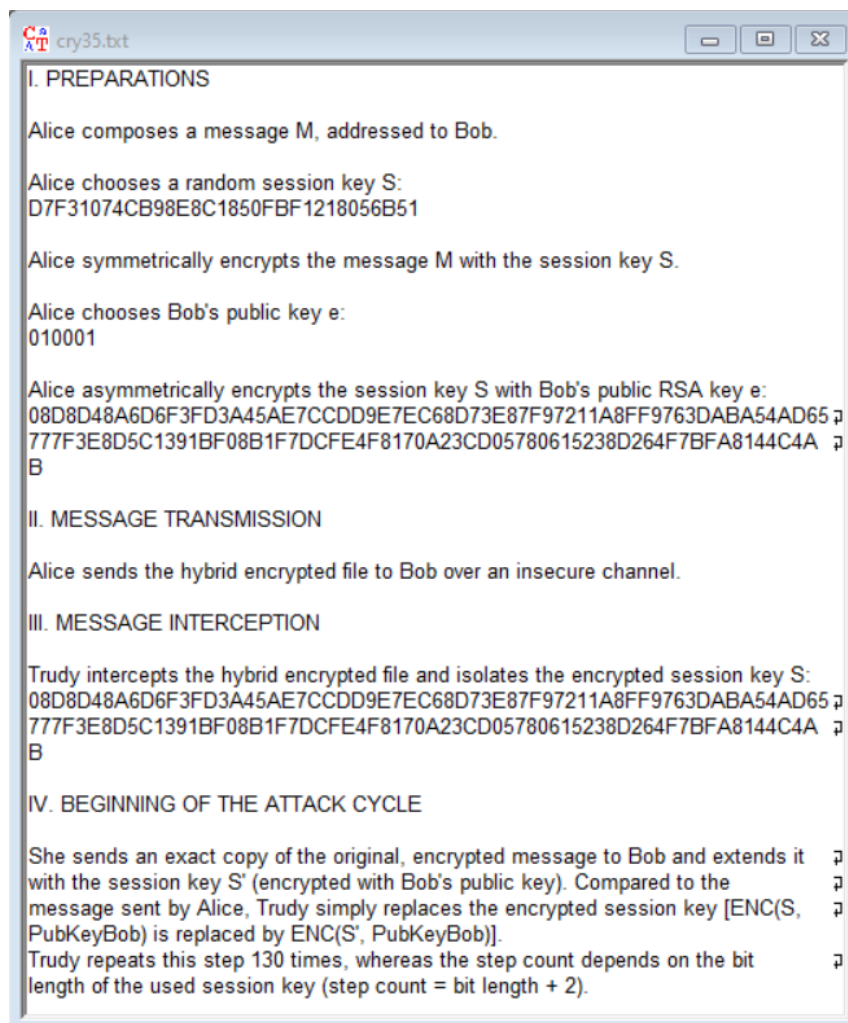


Рисунок 22 – Лог-файл атаки

Заключение

По итогу выполнения данной работы были сделаны следующие выводы:

1. При помощи протокола Диффи-Хеллмана стороны могут обмениваться данными по незащищенному каналу, так как в его основе используется математически-сложная задача дискретного логарифмирования. Также при помощи данного протокола можно создавать ключи для других шифров.

2. Алгоритм RSA – асимметричный блочный шифр (с длиной блока $\log_2 n$ бит). Принимающая сторона генерирует закрытый и открытый ключи, открытый ключ отправляется отправляющей стороне и используется для зашифрования сообщения, после шифровка отправляется принимающей стороне и при помощи закрытого ключа происходит расшифрование. С увеличением длины ключа данное время на зашифровку и на расшифровку возрастает.

На алгоритм RSA можно применить атаку грубой силы если факторизовать часть открытого ключа – модуль n , особенно если n небольшое число.

3. На гибридную модель можно провести атаку «сторонним каналом», основанную на том, что злоумышленник перехватывает цифровой конверт с зашифрованным сообщением и зашифрованным секретным ключом. Была проведена атака на гибридную модель, основанная на том, что злоумышленник перехватывает цифровой конверт с зашифрованным сообщением и зашифрованным секретным ключом. Модифицируя полученные данные и анализируя ответы сервера, можно побитово восстановить целиком секретный ключ. Модифицируя полученные данные и анализируя ответы сервера, можно побитово восстановить секретный ключ.