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

ОТЧЕТ

по практической работе №7
по дисциплине «Криптографические методы защиты информации»
Тема: «Изучение асимметричных протоколов и шифров»

Студент гр. 9361 _____ Кисляков Н.

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.

1. Протокол Диффи–Хеллмана

1.1 Задание

1. Запустить утилиту Indiv.Procedures –> Protocols –> Diffie-Hellman demonstration… и установить все опции информирования в ON.

2. Выполнить последовательно все шаги протокола.

3. Сохранить лог-файл протокола для отчета (пиктограмма с изображением ключа).

4. Использовать полученные общие данные K для создания ключа зашифровки и расшифровки произвольного сообщения. Шифр выбрать самостоятельно.

1.2 Основные параметры и схема протокола

На рисунке 1 представлена схема протокола Диффи-Хеллмана. К основным параметрам протокола Диффи-Хеллмана относится:

- (p, g, R_1) и (p, g, R_2) – открытые ключи сторон;
- x, y – закрытые ключи сторон;
- $R_2^x \ mod \ p$ и $R_1^y \ mod \ p$ – 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: $g \ mod \ p, g^2 \ mod \ p, g^3 \ mod \ p \ \dots \ g^{p-1} \ mod \ p$ является различными целыми из $[1, p - 1]$;

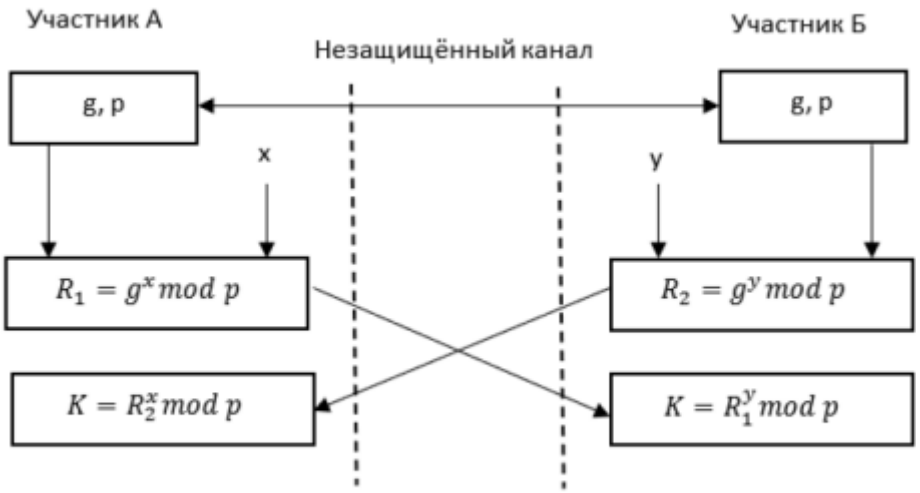
x, y – большие случайные числа такие, что $0 < xx < p - 1$, $0 < y < p - 1$;

- Поскольку:

$R_2^x \ mod \ p \ = \ (g^y \ mod \ p)^x \ mod \ p \ = \ g^{xy} \ mod \ p$;

$R_1^y mod\ p\ =\ (g^x\ mod\ p)^y\ mod\ p\ =\ g^{xy}\ mod\ p$;

- Стороны фактически создают симметричный ключ сеанса без центра распределения ключей (KDC).



$$K = {R_2}^x mod\ p = g^{yx} mod\ p = {R_1}^y mod\ p$$

Рисунок 1 – Протокол Диффи–Хеллмана

1.3 Скриншот демонстрации работы протокола, реализованной в CrypTool

В CrypTool 1, воспользуемся утилитой Indiv.Procedures -> Protocols -> Diffie-Hellman, была рассмотрена схема работы протокола Диффи-Хеллмана на рисунке 2.

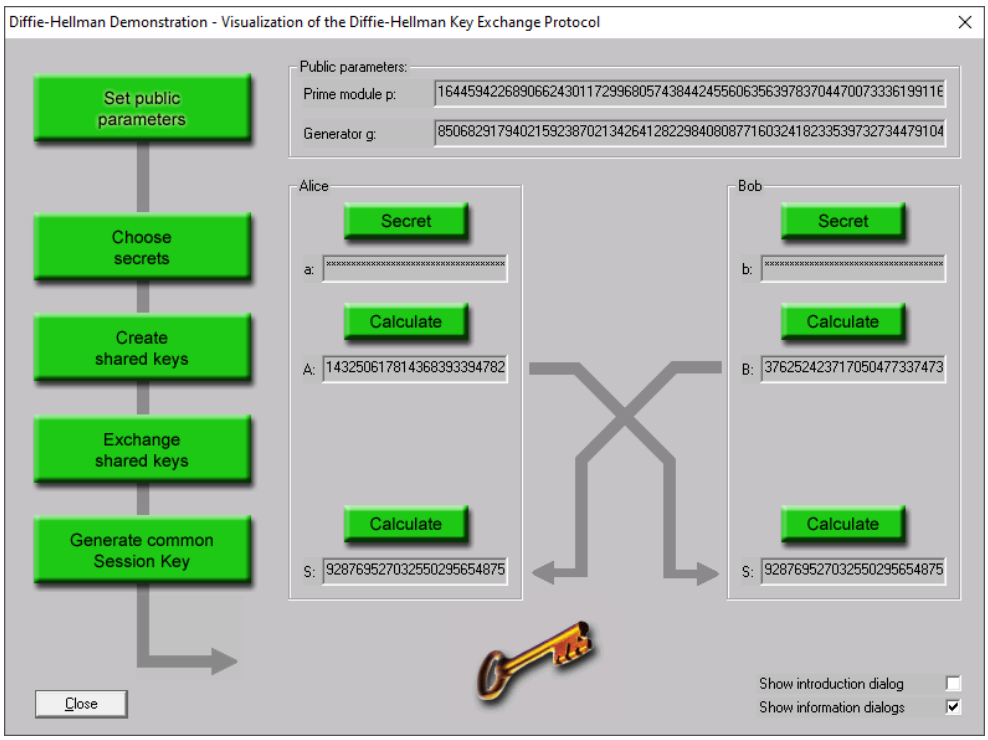


Рисунок 2 – Диффи-Хеллмана

На рисунке 3 представлен лог-файл.

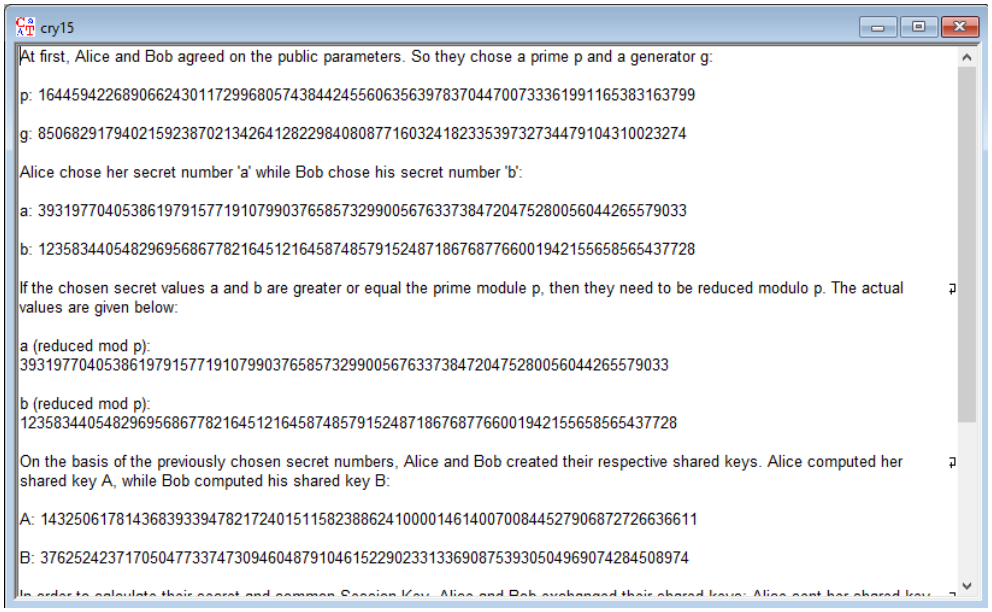


Рисунок 3 – Лог-файл

1.4 Таблица соответствия демонстрации протокола (CrypTool) и параметров протокола

Сопоставим параметры из демонстрационного приложения в CrypTool 1 с параметрами протокола Диффи-Хеллмана. Результаты сопоставления занесены в таблицу 1.

Таблица 1

Параметр протокола Диффи-Хеллмана	Параметр из демонстрационного приложения	Описание параметра
p	p	Большое простое число порядка 300 десятичных цифр (1024 бита)
g	g	Порождающий элемент циклической группы (генератор)
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 секретный ключ

1.5 Скриншот исходного, зашифрованного и расшифрованного текстов, полученных с помощью выбранного шифра и ключа, созданного на основе протокола DH

Используя полученный общий секретный ключ из демонстрационного приложения CrypTool 1, было зашифровано и расшифровано произвольное сообщение. В качестве шифра был выбран шифр AES. На рисунке 4 представлен исходный текст. На рисунке 5 и 6 представлена шифровка и расшифровка исходного текста.

Секретный ключ:

«92876952703255029565487543808587398484298249606752768274309221474681603554527» (CD 56 7E 0A 3D 7C 9F 6B FD 48 0F 9A 38 49 2C 04 71 D2 87 8E 85 96 97 6E 7F AB 9E 57 FE AC FC DF).

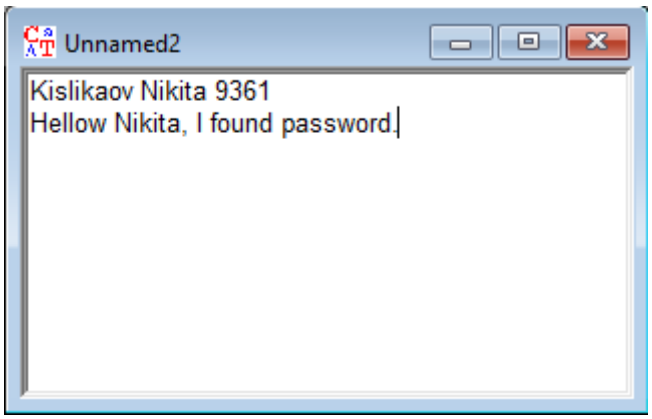


Рисунок 4 – Исходный текст

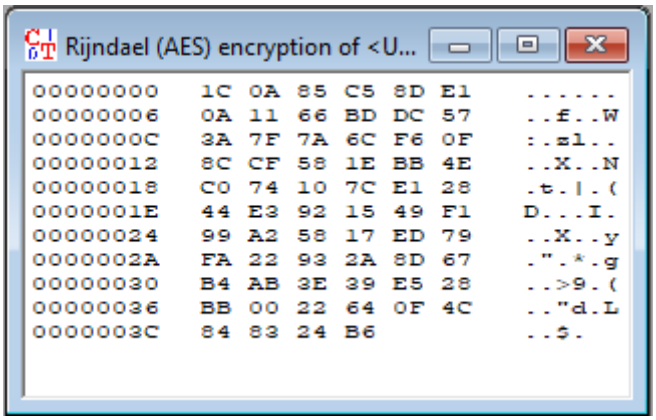


Рисунок 5 – Шифровка

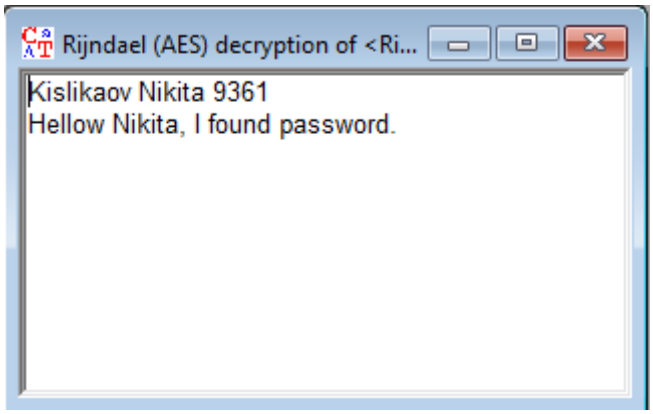


Рисунок 6 – Расшифровка

2. Шифр RSA

2.1 Задание

1. Запустить утилиту Indiv.Procedures –> RSACryptisystem –> RSA Demonstration.

2. Задать в качестве обрабатываемого сообщения свои Ф.И.О.

3. Сгенерировать открытый и закрытый ключи.

4. Зашифровать сообщение. Сохранить скриншот результата.

5. Расшифровать сообщение. Сохранить скриншот результата.

6. Убедиться, что расшифрование произошло корректно

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:

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

b) вычисляется $n = p \times q$;

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

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

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

2. Зашифрование:

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

b) каждый блок открытого текста преобразуется в шифротекст по формуле: $c_i = m_i^e \ mod \ n$.

3. Расшифрование:

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

b) каждый блок шифротекста преобразуется в открытый текст по формуле: $m_i = c_i^d \ mod \ n$.

Обобщенная схема протокола шифрования RSA представлена на рисунке 7.

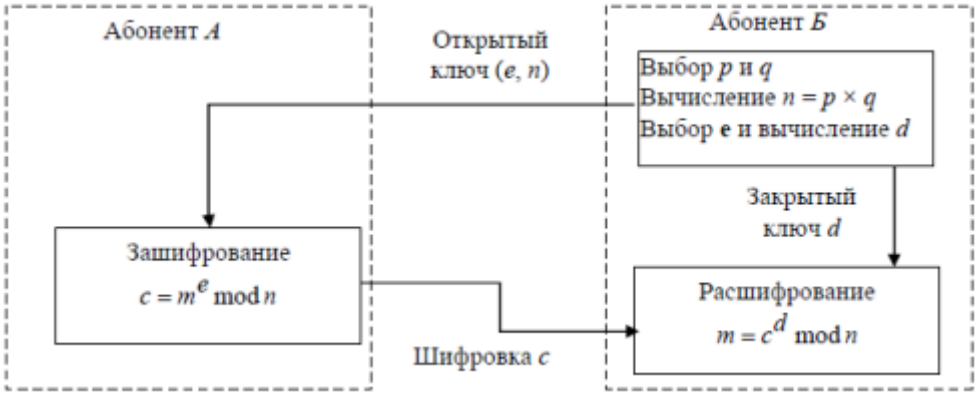


Рисунок 7 – Схема RSA

2.3 Скриншот результата генерации ключей

В CrypTool 1 используем утилитой Indiv.Procedures -> RSACryptisystem -> RSA Demonstration, зашифруем и расшифруем текст «Kisliakov Nikita». Чтоб начать использовать утилиту, нужно сгенерировать открытый и акрытый ключ на рисунке 8. В итоге мы получили следующие ключи:

Открытый ключ: $e = 2^{16} + 1$ и $n = 29737$;

Закрытый ключ: $d = 3013$;

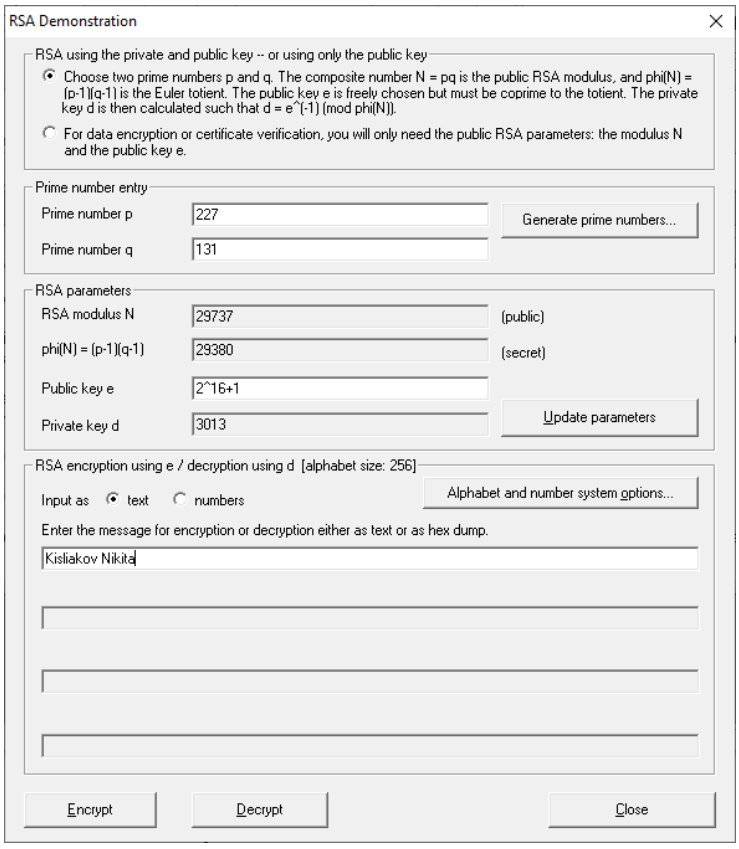


Рисунок 8 – Генерация ключей

2.4 Скриншот результата зашифровки

На рисунке 9 представлен результат шифрования.

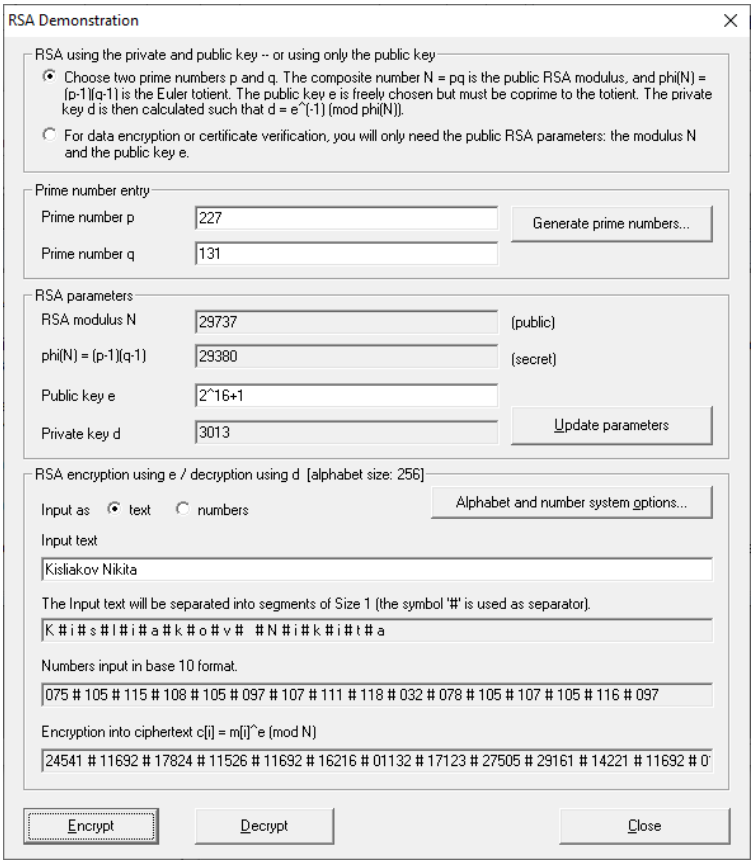


Рисунок 9 – Результат шифрования

2.5 Скриншот результата расшифровки

На рисунке 10 представлена расшифровка сообщения.

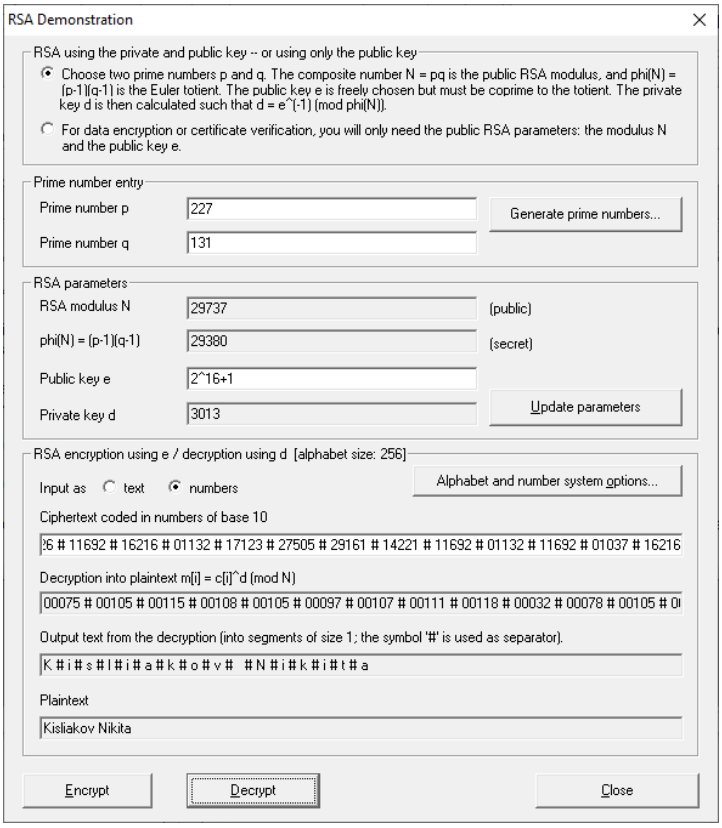


Рисунок 10 – Результат расшифровки

3. Исследование шифра RSA

3.1 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.

3.2 Выбранный текст

В CrypTool 1 выбрали файл «english», данный текст представлен на рисунке 11

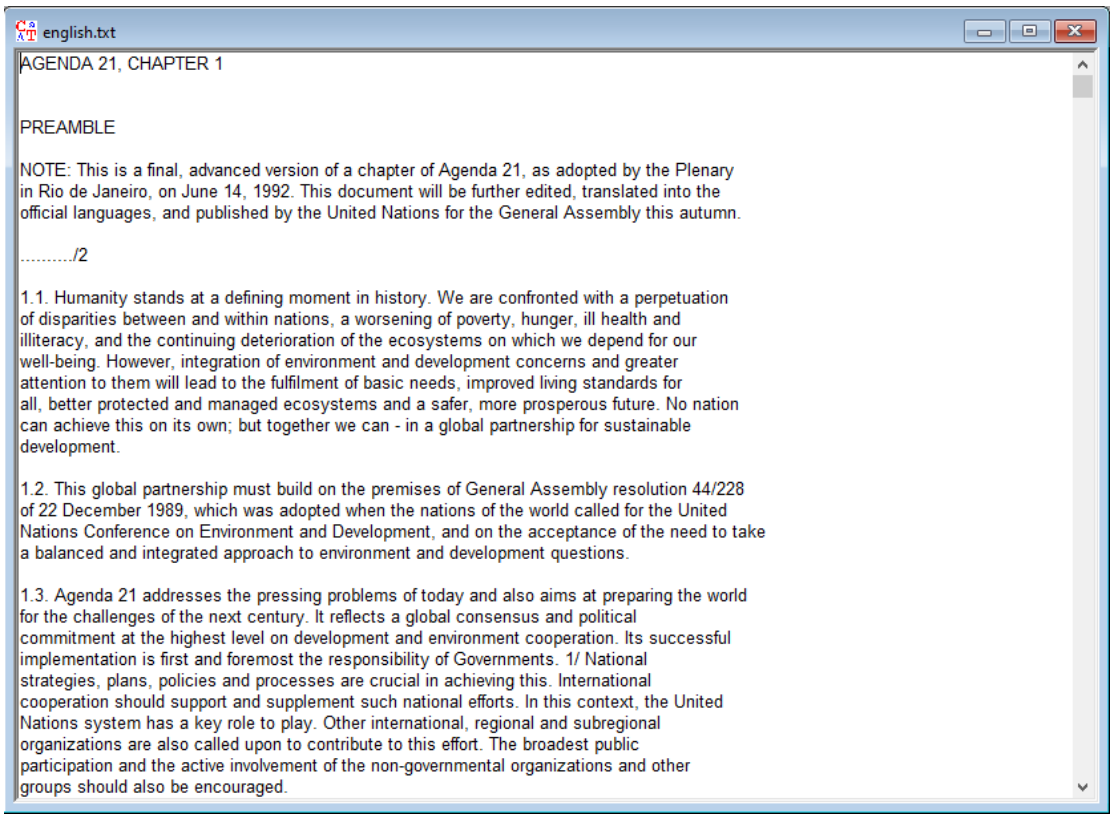


Рисунок 11 – Выбранный текст

3.3 Результаты генерации ключевых пар различной длины

Воспользуемся утилитой из CrypTool 1, Digital Signatures -> PKI -> Generate/Import Keys, чтобы сгенерировать пары ассиметричных RSA-ключей длиной 512, 768, 1024 и 2048 бит. Сгенерированные ключи представлены в таблице 2.

Таблица 2

Длина ключа	Экспонента	Модуль
512	65537	13380028276324062090959881626440459145022537710179750541672551662006074843582494407969080540056790913098686387383818855564924270301511236753601867746791473
768	65537	154567825250658869941806491232034659907370376481122781883694289749103101735663881334725167831758646854649772964575159165596488872011594121707223543178172214620683447124892869843604574699812153309455577996836082011108762073896729656 7

1024	65537	176916946062052870062358849577501702238561525757365104906118699500404099391441777036232625134494874934809936461915206422215161736557852245470990561164238773620541385955586853350726904118221288306686114559587665108353219432220351517215452862002382664931883549579727826786720301233748490310661816231948199469693
2048	65537	3165611213713777199351627791204169317678413047639368243315039093891487688023689639466177757497731244911353100935241443034267008579545735942594446644025485432650720227319762837819678612491778006101967080127008926154020952281704378847204275802809491546181133181179483350189692338555351966746219241100436542250502759700183192159496739206631795564161722713472223622596821345069057418175410093088577151652186113484681875999073614298471352515562475979768322335668491950309972453805058235805439101785625882297018282632654202076737286983176382301648346810231075804260011974909786862500676684054143024420339579754517732927454 7

3.4 Размер исходного текста

Файл «english» состоит из 143253 символа вместе с пробелами.

3.5 Таблица затрат времени на зашифровку и расшифровку при использовании ключей разной длины

После генерации ассиметричных RSA-ключей, зашифруем и расшифруем наш исходный текст. В таблице 3 представлено время шифрования и расшифрования исходного текста.

Таблица 3

Длина ключа	Время шифрования	Время расшифрования
512	0,022 секунд	0,508 секунд
786	0,038 секунд	0,804 секунд
1024	0,053 секунд	1,259 секунд
2048	0,099 секунд	4,240 секунд

На рисунке 12, 13, 14 и 15, представлены результаты шифрования исходного текста с помощью ассиметричных RSA-ключей длиной 512, 768, 1024 и 2048 бит

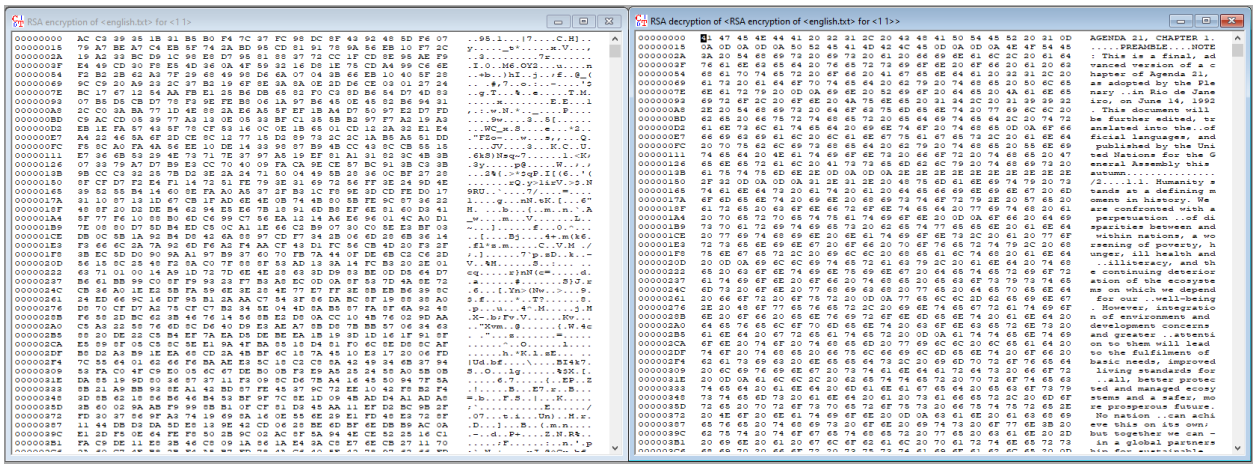


Рисунок 12 – Результат с длиной 512 бит

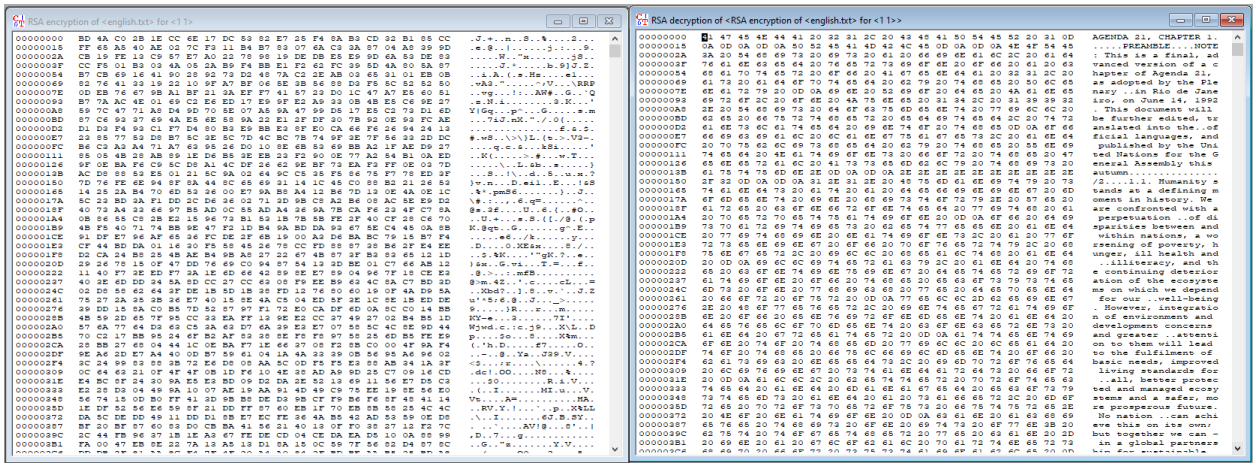


Рисунок 13 – Результат с длиной 768 бит

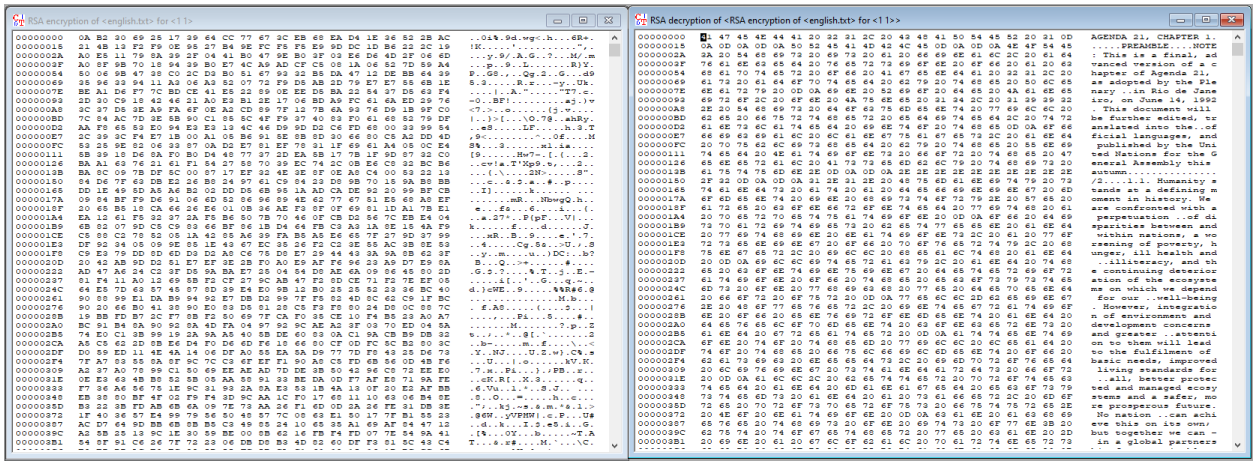


Рисунок 14 – Результат с длиной 1024 бит

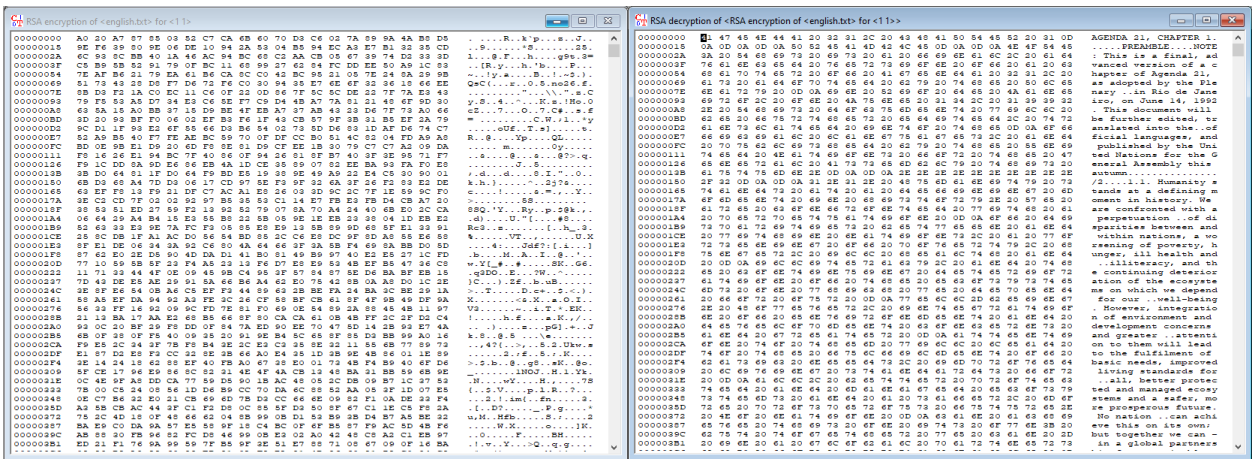


Рисунок 15 – Результат с длиной 2048 бит

4. Атака «грубой силы» на RSA

4.1 Задание

1. Запустить утилиту Indiv.Procedures –> RSACryptosystem –> RSA Demonstration.

2. Установить переключатель в режим «Choose two prime…».

3. Выбрать параметры p и q так, чтобы n = pq > 256.

4. Задать открытый ключ e.

5. Зашифровать произвольное сообщение и передать его вместе с открытым ключом (n, e) коллеге. В ответ получить аналогичные данные.

6. Запустить утилиту Indiv.Procedures –> RSACryptosystem –> RSADemonstration и установить переключатель в режим «For data encryption…».

7. Выполнить факторизацию модуля n командой Factorize…

8. Использовать полученный результат для расшифровки сообщения, полученного от коллеги. Проверить корректность.

4.2 Исходные данные для атаки, полученные от коллеги

Предположим, наш коллега отправил нам следящие исходные данные:

- Шифротекст: «31442 # 07428 # 41521 # 41521 # 34310 # 09394 # 29564 # 20714 # 30861 # 20714 # 10710 # 18504 # 47242 # 09394 # 37508 # 09394 # 32522 # 34310 # 22842 # 47010 # 42283 # 09394 # 02206 # 34310 # 22842 #

08293 # 09394 # 06415 # 18504 # 06205 # 06205 # 06415 # 34310 # 08293 # 42283 # 04897»;

- $n = 49163$;
- $e = 2^{16} + 1$;

4.3 Результат факторизации (скриншот)

В CrypTool 1 воспользуемся утилитой Indiv.Procedures -> RSACryptosystem -> RSA Demonstration. Выполним факторизацию полученного модуля n. Результат факторизации представлен на рисунке 16. В конечном результате были получены параметры $p = 211$ и $q = 233$.

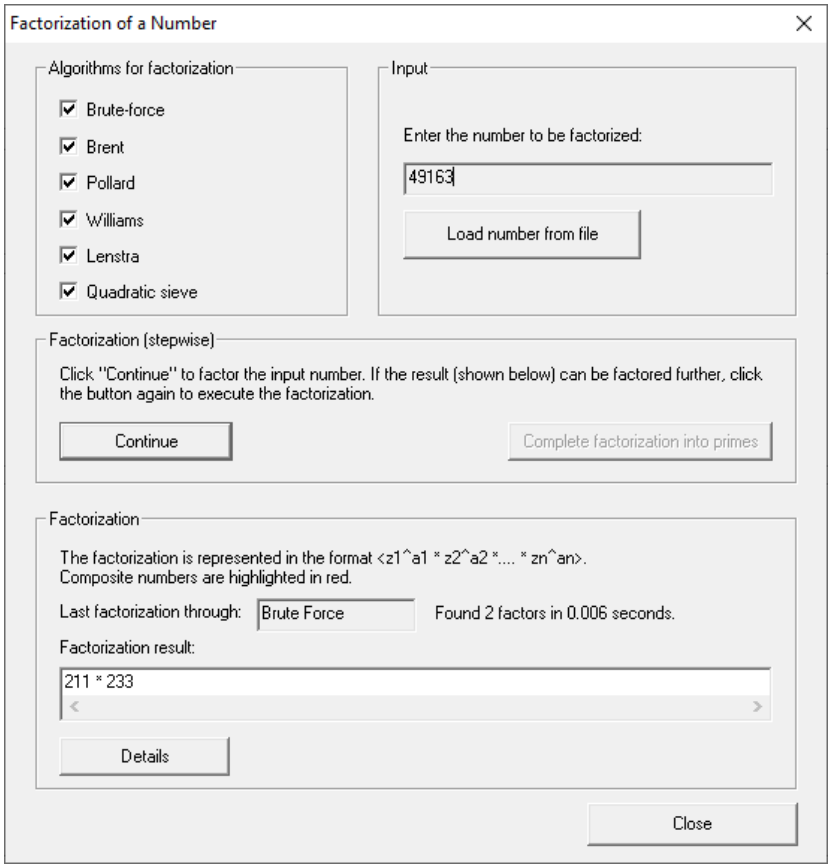


Рисунок 16 – Факторизация модуля n

4.4 Расшифрованное в итоге сообщение (скриншот)

Имея достаточно параметров, мы можем расшифровать сообщение нашего коллеги. В итоге мы получили «Hello Nikita, I found your passpord.». Результат представлен на рисунке 17.

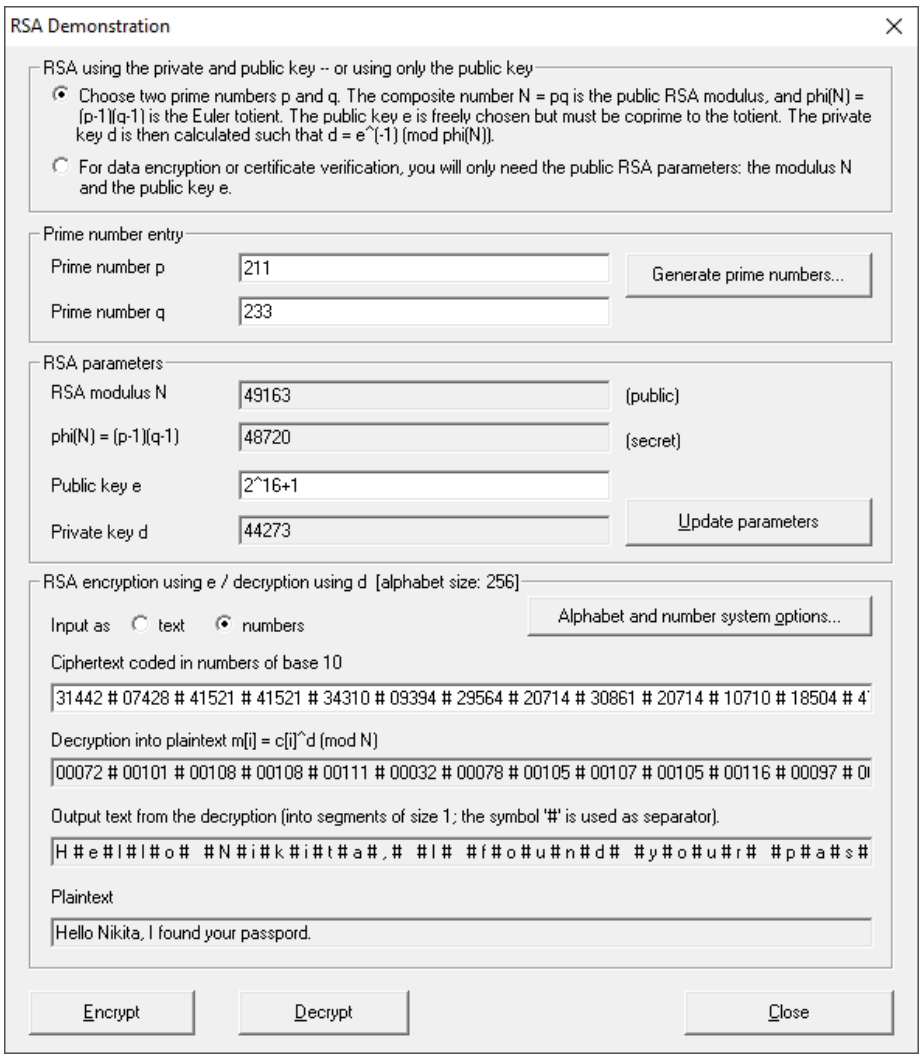


Рисунок 17 – Расшифрованное сообщение

5. Имитация атаки на гибридную криптосистему

5.1 Задание

1. Подготовить текст передаваемого сообщения на английском с вашим именем в конце.

2. Запустить утилиту Analysis –> Asymmetric Encr… –> Side-Channel attack on «Textbook RSA»…

3. Настроить сервер, указав в качестве ключевого слова ваше имя, используемое в конце текста.

4. Выполнить последовательно все шаги протокола.

5. Сохранить лог-файлы участников протокола для отчета.

5.2 Описание цели атаки, модель (возможности) злоумышленника, схема атакуемого протокола гибридного шифрования

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

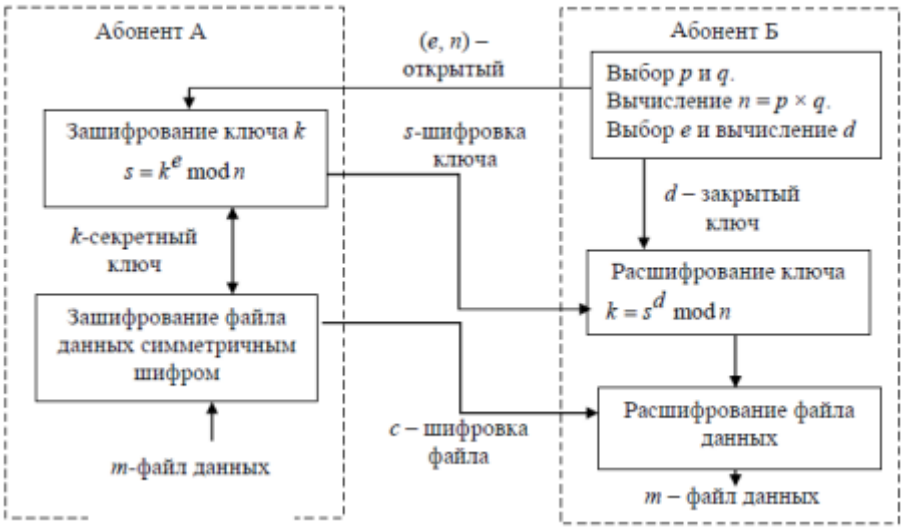


Рисунок 18 – Модель гибридной криптосистемы

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.

Атака на модель гибридной криптосистемы основана на том, что злоумышленник сначала перехватывает цифровой конверт, содержащий зашифрованные сообщение и секретный ключ, затем специальными образом модифицирует шифровку ключа из конверта и восстанавливает бит за битом зашифрованный секретный ключ, анализируя положительные и отрицательные ответы сервера, которые злоумышленник получает по побочным каналам.

Цель атаки – 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).

5.3 Алгоритм действий злоумышленника

Длина в битах модуля n, используемого в RSA, существенно больше, чем длина в битах секретного ключа. При расшифровке конверта сервер использует только младшие биты расшифрованного сообщения в качестве секретного ключа.

Модификация на первом шаге выполняется путем замены старших бит конверта шифровкой ключа, сдвинутой на один бит влево. Анализируется ответ сервера: если ПРИНЯТО, то бит, следующий за старшим битом конверта – нулевой, а если ОТКЛОНЕНО, то бит равен единице. Продолжая действовать подобным образом, можно бит за битом восстановить целиком секретный ключ.

5.4 Текст передаваемого сообщения

На рисунке 19 представлен исходный текст.

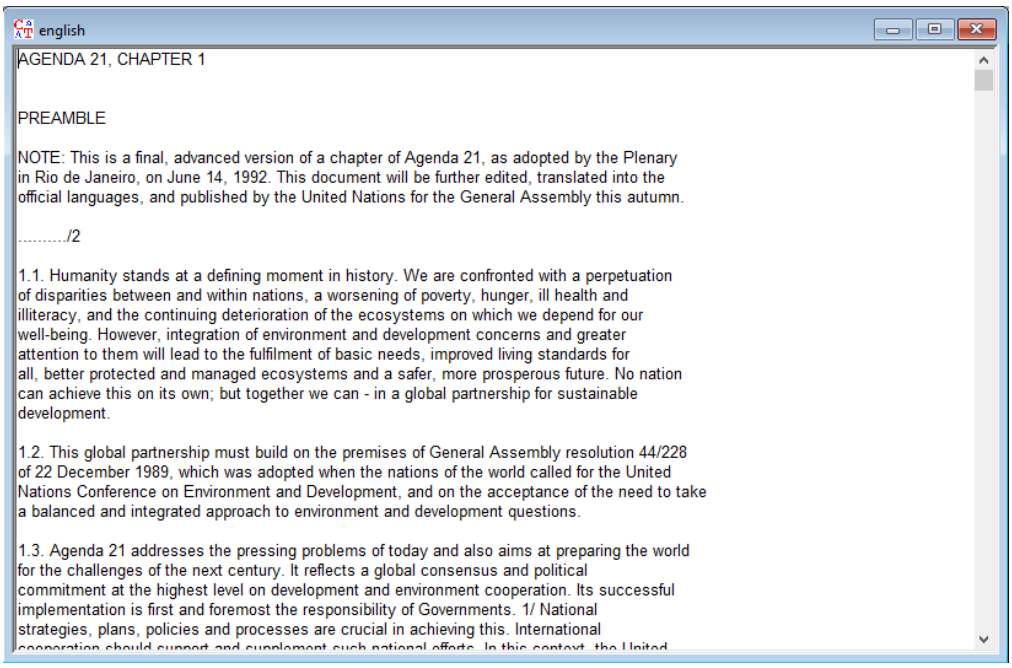


Рисунок 19 – Исходный текст

5.5 Лог-файлы участников протокола

В CrypTool 1, воспользуемся утилитой Analysis -> Asymmetric Encr… -> Side-Channel attack on «Textbook RSA». Используем ключевое слово «Nikita». На рисунке 20 представлен результат работы, после выполнение всех шагов атаки.

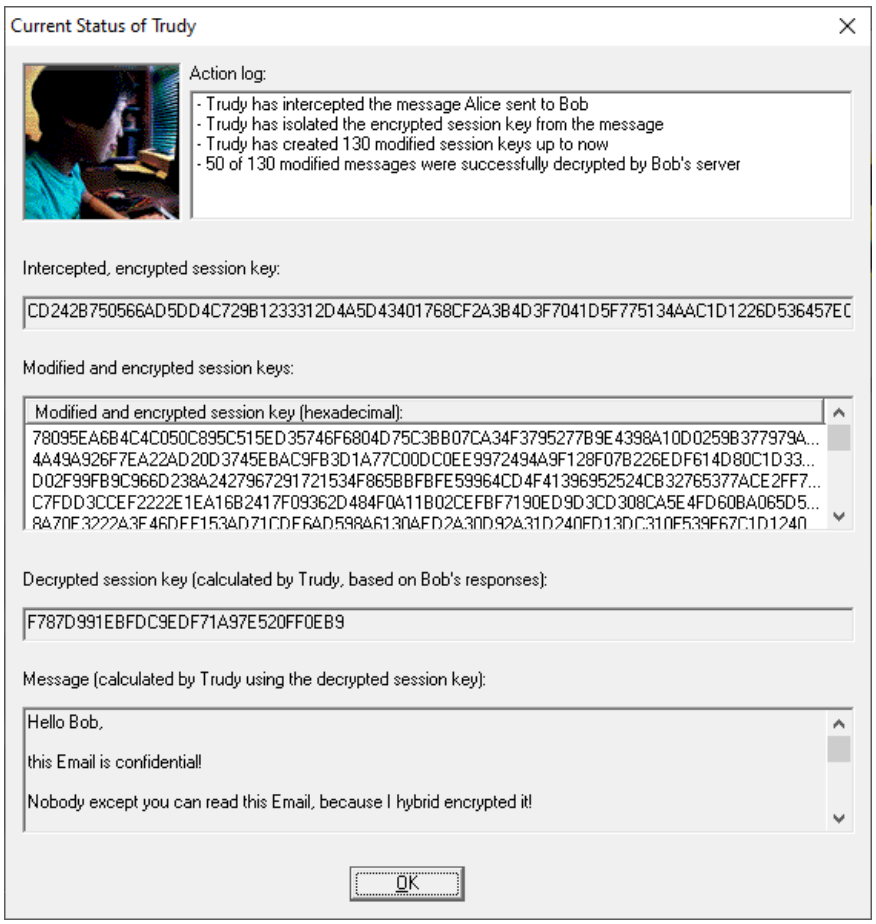


Рисунок 20 – Результат атаки

На рисунке 21 представлен лог-файл.

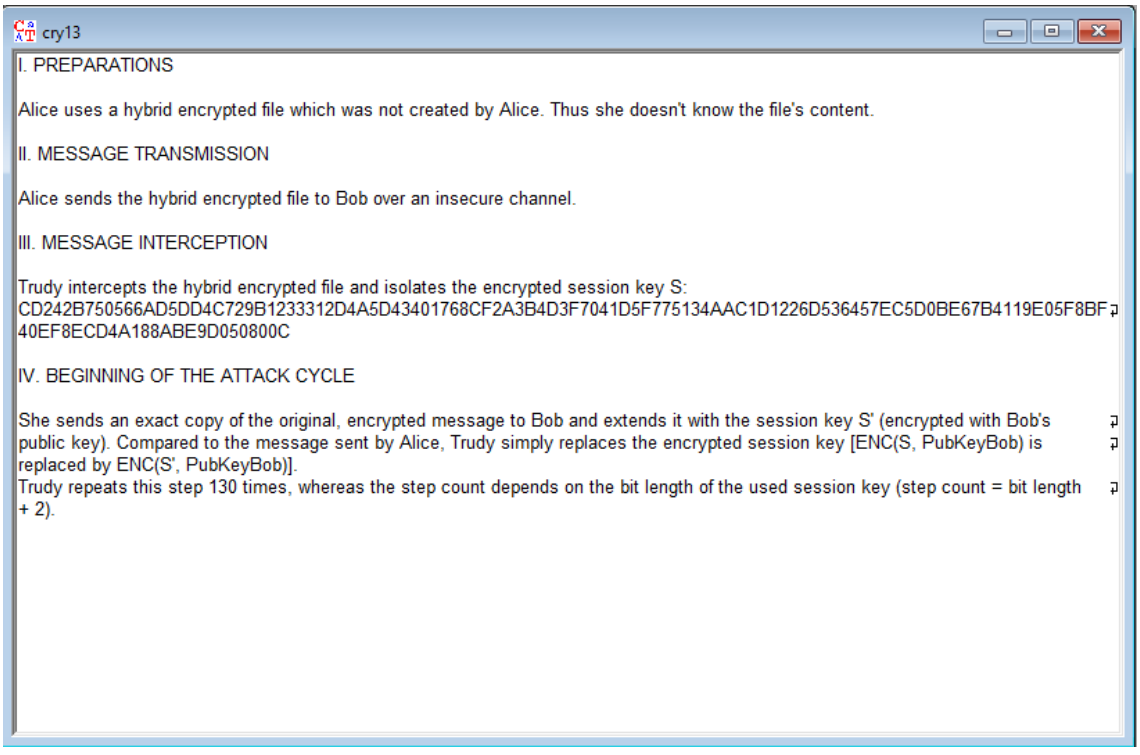


Рисунок 21 – Лог-файл

Вывод

В данной работе, были изучены принцип работы протоколов Диффи-Хелмана, шифр RSA, атака «грубой силы» на шифр RSA и атака на гибридную криптосистему. Данные протоколы были изучены в такой программе ка CrypTool 1.

1. Протокол Дифф-Хелмана

В данной главе, ознакомились с работой протокола Диффи-Хелмана. Основные параметры протокола ДиффиХеллмана: $(p,\ g,\ R_1)$ и $(p,\ g,\ R\_2\ )$ – открытые ключи сторон; $x,\ y$ – закрытые ключи сторон; $R_2^x\ \mathrm{mod}\ p$ и $R_1^y\ \mathrm{mod}\ p$ – односторонние функции с секретом (TOWF); p – большое простое число порядка 300 десятичных цифр (1024 бита); g – порождающий элемент циклической группы (генератор) порядка pp; $x,\ y$ – большие случайные числа такие, что $0 < x < p - 1$, $0 < y < p - 1$. Секретный ключ вычисляется следующим образом: $K\ =\ R_2^x\ mod\ p\ =\ R_1^y\ mod\ p$.

2. Шифр RSA

В данной главе, мы изучили принцип работы протокола RSA. Протокол RSA является симметричным блочным шифром. В котором блоки открытого и зашифрованного сообщений представляются целыми числами из диапазона от 0 до $n - 1$ для блока размером $\log_2 n$ бит.

При помощи утилиты из CrypTool 1, мы сгенерировали ключ, по определенным параметрам и после это зашивроали наш исходный текст, а после этого расшифровали наше сообщение.

Также пришли к выводу что шифр RSA для больших объемов данных использовать нецелесообразно. Лучше использовать гибридное шифрование.

3. Атака «грубой силой» на шифр RSA.

В этой главе мы изучили и применили атаку «грубой силой» на шифр RSA. Мы сгенерировали 4 пары ассиметричных RSA-ключей длиной 512, 768, 1024 и 2048 бит с экспонентой 65537. Затем шифровали и расшифровали текст, который состоит из 143253 символов. Время, которое уделялось для

шифрования и расшифрования записали в таблицу. По данной таблице можно сделать вывод, чем больше длина ключа, тем больше времени он будет тратить на это.

4. Атака на гибридную криптосистему.

Была изучена имитация атаки на гибридную криптосистему. Целью 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 ответы сервера.