МИНОБРНАУКИ РОССИИ

Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В. И. Ульянова (Ленина)

КРИПТОГРАФИЧЕСКИЕ МЕТОДЫ ЗАЩИТЫ ИНФОРМАЦИИ

Лабораторный практикум

Санкт-Петербург
СПбГЭТУ «ЛЭТИ»
2016

Авторы: А. К. Племянников, Е.О. Кузнецова

Криптографические методы защиты информации: лабораторный практикум. СПбГЭТУ «ЛЭТИ», 2016 55 с.

Содержит описание лабораторных работ, в которых изучаются и исследуются классические и современные симметричные шифры, хэш-функции, ассиметричные шифры, протоколы цифровой подписи и цифровые сертификаты. Предназначено для студентов, обучающихся по специальности 10.05.01 «Компьютерная безопасность».

Одобрено
Методической комиссией факультета компьютерных технологий и информатики в качестве учебно-методического пособия

Содержание

Лабораторная работа №1
Изучение классических шифров средствами RailFence, Scytale, Caesar.

Цель работы: исследовать шифры Rail Fence, Scytale, Caesar и получить практические навыки работы с ними, в том числе и в программном продукте Cryptool 1 и 2.

1.1 Шифр изгороди (Rail Fence)

В шифре Изгороди открытый текст разбивается на определенное количество строк. В каждой строке поочередно записывается одна буква подобно изгороди, зашифрованный текст составляется при чтении строки за строкой. Например, при разбиении открытого текста *«0123456789»* на 3 строки шифрование выглядит следующим образом:

Разбиение на строки						Шифротекст
0		4		8		
	1	3	5	7	9	=> 0481357926
		2		6		

Для увеличения криптостойкости используют смещение. Произведем шифрование из примера выше со смещением 2:

Разбиение на строки						Шифротекст
-		2		6		
	-	1	3	5	7	9 => 2613579048
		0		4		8

Задание

1. Найти шифр в CrypTool 1: *Encrypt/Decrypt-> Symmetric(Classis).*
2. Создать файл с открытым текстом, содержащим последовательность цифр.
3. Запустить шифр и выполнить зашифровку и расшифровку созданного текста несколько раз.

4. Установить, как влияют на шифрование параметры *Number of Rows* и *Offset.*

5. Зашифровать и расшифровать текст, содержащий только фамилию (транслитерация латиницей) вручную и с помощью шифра при *Number of Rows>2, Offset≥2*. Убедиться в совпадении результатов.

6. Создайте шифровку для варианта *Offset=0* и *Number of Rows≤* и передайте коллеге слева для расшифровки.

7. Определите ключ методом «грубой силы» и расшифруйте полученный от коллеги шифротекст.

Содержание раздела отчета

1. Задание.
2. Реализация в CrypTool 1.0 (скриншот, спецификация параметров).
3. Схема, поясняющая работу шифра.
4. Пример работы шифра для выбранных параметров.
5. Тип шифра (перестановка, замена, комбинированный).
6. Ключ шифра.
7. Оценка сложности атаки “грубой силы”.
8. Результат расшифровки перехваченного от коллеги текста.

1.2 Шифр «Сцитала» (Scytale)

В криптографии сцитала, известный также как *шифр Древней Спарты*, представляет собой прибор, используемый для осуществления перестановочного шифрования. Прибор состоит из цилиндра и узкой полоски пергамента, обматывавшейся вокруг него по спирали, на которой писалось сообщение. Иллюстрация, демонстрирующая работу данного шифра представлена на рисунке 1.1.

Рисунок 1.1

Для расшифровки использовался цилиндр такого же диаметра, на который наматывался пергамент, чтобы прочитать сообщение.

Задание

1. Найти шифр в CrypTool 1: *Encrypt/Decrypt-> Symmetric(Classic).*
2. Создать файл с открытым текстом, содержащим последовательность цифр.
3. Запустить шифр и выполнить зашифровку и расшифровку созданного текста несколько раз.
4. Установить, как влияют на шифрование параметры *Number of Edges* и *Offset.*
5. Зашифровать и расшифровать текст содержащий только фамилию (транслитерация латиницей) вручную и с помощью шифра при *Number of Edges > 2, Offset ≥ 2*. Убедиться в совпадении результатов.
6. Взять в CrypTool 2 шаблон атаки на шифр методом «грубой силы» и модифицировать этот шаблон, заменив блок с шифротекстом на блок ввода открытого текста и блок зашифрования. Изучить принципы этой автоматической атаки.

Содержание раздела отчета

1. Задание.
2. Реализация в CrypTool 1.0 (скриншот, спецификация параметров).
3. Схема, поясняющая работу шифра.
4. Пример работы шифра для выбранных параметров.
5. Тип шифра (перестановка, замена, комбинированный).
6. Ключ шифра.
7. Описание и оценка сложности атаки “грубой силы” на шифротекст, реализованной в CrypTool 2.

1.3 Шифр Цезаря (Caesar)

Шифр Цезаря — один из древнейших шифров. Шифр назван в честь римского императора Гая Юлия Цезаря, использовавшего его для секретной переписки.

Суть шифра в следующем: в одну строку записываются элементы алфавита, выбирается смещение, согласно которому ниже, символ под символом записывается используемый нами алфавит, сдвинутый влево на

величину смещения. Символ, находящийся под символом исходного алфавита в открытом тексте, является символом в шифротексте.

Например, зашифруем текст «*КРИПТОГРАФИЯ*», используя смещение равное 3 и русский алфавит:

а	б	в	**г**	д	е	ж	з	**и**	**к**	л	м	н	**о**	**п**	**р**	с	**т**	у	**ф**	х	ц	ч	ш	щ	ъ	ы	ь	э	ю	**я**
г	д	е	**ж**	з	и	к	л	**м**	**н**	о	п	р	**с**	**т**	**у**	ф	**х**	ц	**ч**	ш	щ	ъ	ы	ь	э	ю	я	а	б	**в**

Произведя замену получаем шифротекст: *НУМТХЖУГЧМВ*.

Если сопоставить каждой букве алфавита число, то математически шифрование и расшифровка производится по следующим формулам:

$$y = (x + k) mod\ n,$$

$$x = (y - k + n) mod\ n,$$

где x – открытый текст, y – полученный шифротекст, k – смещение, n – мощность алфавита (длина алфавита).

Задание

1. Найти шифр в CrypTool 1: *Encrypt/Decrypt-> Symmetric(Classic).*
2. Зашифровать и расшифровать текст, содержащий только фамилию (транслитерация латиницей) вручную и с помощью шифра с ключом, отличным от 0. Убедиться в совпадении результатов.
3. Построить гистограмму частот букв английского языка по эталонному файлу *English.txt* (папка *CrypTool/reference*), используя утилиту из *Analysis-> Tools foAnalysis.*
4. Зашифровать ключом отличным от 0 файл *CrypTool-en.txt* (папка *CrypTool/Examples*).
5. Построить гистограмму частот букв в зашифрованном тексте, сравнить визуально гистограммы и подтвердить ключ зашифрования.
6. Проверить гипотезу о значении ключа утилитой *Analysis-> Symmetric Encryption(Classic)->Cipher Text Only->Caesar.*
7. Передать шифровку соседу слева для проведения подобной атаки.

Содержание раздела отчета

1. Задание.
2. Реализация в CrypTool 1.0 (скриншот, спецификация параметров).

3. Схема, поясняющая работу шифра.
4. Пример работы шифра для выбранных параметров.
5. Описание выполненной процедуры атаки на шифротекст и результат (ключ) этой атаки.
6. Тип шифра (перестановка, замена, комбинированный).
7. Ключ шифра.
8. Оценка сложности атаки “грубой силы”.

1.4 Содержание отчета по лабораторной работе

В отчете следует сформулировать цель работы, наполнить каждый раздел необходимым содержанием и сделать краткие выводы по работе в заключении.

Лабораторная работа №2
Изучение классических шифров Substitution, Permutation/Transposition, Vigenere

Цель работы: исследовать шифры Substitution, Permutation/Transposition, Vigenere и получить практические навыки работы с ними, в том числе и в программном продукте Cryptool 1 и 2.

1.1 Шифр моноалфавитной подстановки(Substitution)

В шифре моноалфавитной подстановки используются два параметра Key и Offset. Key – это кодовое слово, на основе которого формируется алфавит шифрограммы. Первым шагом служит удаление всех элементов алфавита, которые присутствуют в кодовом слове. Затем все удвоенные элементы кодового слова сливаются в один. На втором шаге задается значение смещения первого элемента кодового слова используя параметр Offset. То есть по значению смещения определяется количество символов алфавита, полученного после удаления его элементов, которое будет предшествовать вставке кодового слова, после которого продолжится запись имеющегося алфавита.

Например, зашифруем текст «*CRYPTOGRAPHY*», используя кодовое слово «*PASSWORD*», смещение равное 5 и английский алфавит:

a	b	**c**	d	e	f	**g**	**h**	i	j	k	l	m	n	**o**	**p**	q	**r**	s	**t**	u	v	w	x	**y**	z

b	c	**e**	f	g	p	**a**	**s**	w	o	r	d	h	i	**j**	**k**	l	**m**	n	**q**	t	u	v	x	**y**	z

Произведя замену получаем шифротекст: *EMYKQJAMBKSY.*

Задание

1. Найти шифр в CrypTool 1: *Encrypt/Decrypt-> Symmetric(Classic).*

2. Зашифровать и расшифровать текст содержащий только фамилию (транслитерация латиницей) вручную и с помощью шифра с выбранным ключом и смещением *Offset*≠ 0. Убедиться в совпадении результатов.

3. Выполнить зашифрование и расшифрование с различными паролями и смещениями *Offset* и разобраться как формируется алфавит шифрограммы.

4. Выбрать абзац (примерно 600 символов) из файла *English.txt* (папка *CrypTool/reference*) и зашифровать его.

5. Выполнить атаку на шифротекст, используя приложение из *Analysis-> Symmetric Encryption(classic)-> Cipher Text Only.*

6. Повторить шифрование и атаку для тестов примерно в 300 и в 150 символов

7. Изучите ручное расшифрование для текстов менее 300 символов.

8. Выбрать новый абзац (примерно 600 символов) из файла *English.txt* (папка CrypTool/reference) и зашифровать его.

9. Расшифровать этот абзац, используя приложение *Analysis-> Tools for Analysis* и *Analysis-> Symmetric Encryption(classic)-> Manual Analysis.*

10. Зашифруйте текст из 200 символов, сохраните ключ, и передайте коллеге для расшифровки.

11. Самостоятельно изучите атаки, реализованные CrypTool 2, опираясь на Help и ссылки на статьи.

Содержание раздела отчета

1. Задание.

2. Реализация в CrypTool 1.0 (скриншот, спецификация параметров).

3. Пример работы шифра для выбранных параметров.
4. Тип шифра (перестановка, замена, комбинированный).
5. Ключ шифра.
6. Оценка сложности атаки “грубой силы”.
7. Описание атаки на шифр с использованием утилит CrypTool 1.0.
8. Результат расшифровки перехваченного от коллеги текста.
9. Описание атаки на шифр реализованной в CrypTool 2.0.

1.2 Шифр двойной перестановки(Permutation/Transposition)

В основе шифра лежит матричное расположение открытого текста и преобразование его посредством инверсии исходного ключа. Данный шифр можно проводить по строкам или по столбцам, а также с одной или двумя перестановками.

Например, зашифруем текст «*ПРИМЕРМАРШРУТНЫЙШИФР*»:

1. Записываем текст в матрицу:

	5	4	3	1	2
2	п	р	и	м	е
4	р	м	а	р	ш
3	р	у	т	н	ы
1	й	ш	и	ф	р

2. Производим перестановку столбцов:

	1	2	3	4	5
2	м	е	и	р	п
4	р	ш	а	м	р
3	н	ы	т	у	р
1	ф	р	и	ш	й

3. Производим перестановку строк:

	1	2	3	4	5
1	ф	р	и	ш	й
2	м	е	и	р	п
3	н	ы	т	у	р
4	р	ш	а	м	р

Выписываем текст строка за строкой и получаем шифротекст: *ФРИШЙМЕИРПНЫТУРРШАМР*

Задание

1. Найти шифр в CrypTool 1: *Encrypt/Decrypt-> Symmetric(Classic).*
2. Зашифровать и расшифровать текст, содержащий ФамилиюИмяОтчество (транслитерация латиницей) вручную и с помощью шифра с ключами для перестановки столбцов и строк. Убедиться в совпадении результатов.
3. Выполнить зашифрование и расшифрование с различными ключами и с различными вариантами перестановки матрицы с текстом по строкам и столбцам. Разобраться с параметрами утилиты.
4. Зашифровать текст, содержащий ФамилиюИмяОтчество и провести атаку, основанную на знании исходного текста *Analysis-> Symmetric Encryption(classic)-> Known Plaintext.*
5. Зашифровать текст с произвольным сообщением в формате «DEAR message THANKS», используя только одинарную перестановку.
6. Передайте шифровку соседу, для расшифрования при условии, что формы обращения и завершения письма известны.
7. Самостоятельно изучите атаки, реализованные в CrypTool 2, опираясь на Help и ссылки на статьи.

Содержание раздела отчета

1. Задание.
2. Реализация в CrypTool 1.0 (скриншот, спецификация параметров).
3. Пример работы шифра для выбранных параметров.
4. Тип шифра (перестановка, замена, комбинированный).
5. Ключ шифра.
6. Оценка сложности атаки “грубой силы”.
7. Описание атаки на шифр с использованием утилит CrypTool 1.0.

8. Результат расшифровки перехваченного от коллеги текста.

9. Описание атаки на шифр реализованной в CrypTool 2.0.

1.3 Шифр Виженера (Vigenere)

Шифр Виженера - метод полиалфавитного шифрования буквенного текста с использованием ключевого слова. Шифр Виженера состоит из последовательности нескольких шифров Цезаря с различными значениями сдвига. Для зашифровывания может использоваться таблица алфавитов, называемая tabula recta или квадрат (таблица) Виженера.

Выбирается кодовое слово длины n, которое делит открытый текст на отрезки данной длины. Каждой из получившихся последовательностей в соответствие ставится кодовое слово. Далее составляется, так называемая, таблица Вижинера. Горизонтально записывается алфавит, вертикально под первым символом алфавита записывается кодовое слово. Заполнение таблицы осуществляется символами алфавита, начинающегося с элемента кодового слова, и циклически замыкается (т.е. применительно к латинице это выглядит так: …xyzabc…). Элемент шифротекста получается пересечением соответствующего элемента алфавита из отрезка открытого текста и элемента кодового слова.

Например, зашифруем текст «*ПРИМЕРШИФРАВИЖЕНЕРА*», используя кодовое слово «*КЛЮЧ*» и русский алфавит:

1. Делим текст на отрезки:

п	р	и	м	е	р	ш	и	ф	р	а	в	и	ж	е	н	е	р	а	
к	л	ю	ч	к	л	ю	ч	к	л	ю	ч	к	л	ю	ч	к	л	ю	

2. Производим замену (для 1ого отрезка):

а	б	в	г	д	е	ж	з	**и**	к	л	**м**	н	о	**п**	**р**	с	т	у	ф	х	ц	ч	ш	щ	ъ	ы	ь	э	ю	я
К	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	**ш**	щ	ъ	ы	ь	э	ю	я	а	б	в	г	д	е	ж	з	а
Л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	**ъ**	ы	ь	э	ю	я	а	б	в	г	д	е	ж	з	и	к
Ю	я	а	б	в	г	д	е	**ж**	з	и	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ъ	ы	ь	э
Ч	ш	щ	ъ	ы	ь	э	ю	я	а	б	**в**	г	д	е	ж	з	и	к	л	м	н	о	п	р	с	т	у	ф	х	ц

Получаем шифротекст для первого отрезка: *ШЪЖВ*

3. Производим аналогичную замену всех отрезков, получаем итоговый шифротекст: *ШЪЖВПЪЦЯЭЪЮЩТСГГПЪЮ*

Задание

1. Найти шифр в CrypTool 1: *Encrypt/Decrypt-> Symmetric(Classic).*

2. Зашифровать и расшифровать текст, содержащий только фамилию (транслитерация латиницей) вручную и с помощью шифра с выбранным ключом. Убедиться в совпадении результатов.

3. Произвести атаку на шифротекст, используя приложение *Analysis-> Symmetric Encryption(Classic)-> Cipher Text Only->Vigenere.*

4. Повторить атаку для фрагмента текста из файла *English.txt* (папка CrypTool/reference). Размер текста не менее 1000 символов.

5. Воспроизведите эту атаку в автоматизированном режиме:

   a. Определите размер ключа с помощью приложения *Analysis-> Tools for Analysis-> Autocorrelation*

   b. Выполните перестановку текста с размером столбца равным размеру ключа приложением *Permutation/Transposition*

   c. Определите очередную букву ключа приложением *Analysis-> Symmetric Encryption(Classic)-> Cipher Text Only->Caesar*.

6. Самостоятельно изучите атаки, реализованные CrypTool 2, опираясь на Help и ссылки на статьи.

Содержание раздела отчета

1. Задание (как в описании работы).

2. Схема и формулы, поясняющие работу шифра.

3. Пример работы шифра для выбранных параметров.

4. Тип шифра (перестановка, замена, комбинированный).

5. Ключ шифра.

6. Оценка сложности атаки “грубой силы”.

7. Описание выполненной процедуры атаки на шифротекст и результат (ключ) этой атаки.

8. Описание атаки на шифр реализованной в CrypTool 2.0.

1.4 Содержание отчета по лабораторной работе

В отчете следует сформулировать цель работы, наполнить каждый раздел необходимым содержанием и сделать краткие выводы по работе в заключении.

Лабораторная работа №3
Изучение классических шифров Hill, ADFGVX, Playfair

Цель работы: исследовать шифры Hill, ADFGVX, Playfair и получить практические навыки работы с ними, в том числе и в программном продукте CrypTool 1 и 2.

2.1 Шифр Хилла (Hill)

Шифр Хилла подразумевает работу с матрицами. Перед шифрованием необходимо каждому символу открытого текста сопоставить число, записать полученные значения в матрицу размера *n*m* и сформировать шифрующую матрицу *n*n*. Шифрующая матрица должна быть обратимой для выбранного модуля, по которому производятся вычисления (обычно для английского алфавита этот модуль 26). Для шифрования производится умножение матрицы открытого текста на шифрующую матрицу. Для дешифрации необходимо шифротекст умножить на обратную к шифрующей матрицу.

В качестве примера шифрования, зашифруем текст «*HILLCIPHEREXAMPLES*»:

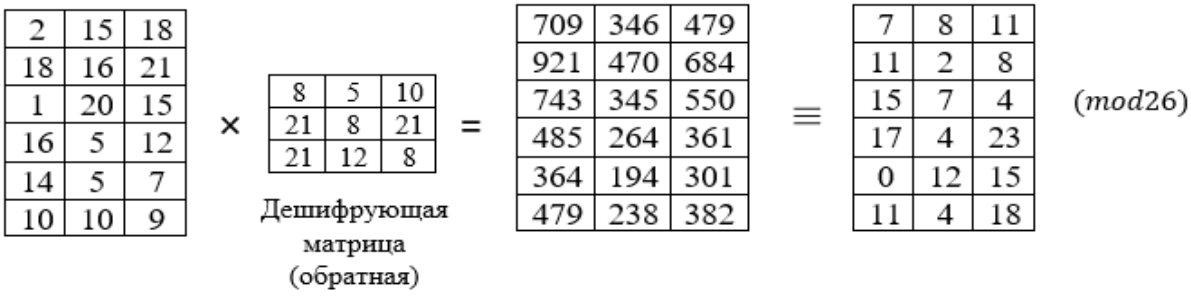
a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

$$\begin{pmatrix} 7 & 8 & 11 \\ 11 & 2 & 8 \\ 15 & 7 & 4 \\ 17 & 4 & 23 \\ 0 & 12 & 15 \\ 11 & 4 & 18 \end{pmatrix} \times \underbrace{\begin{pmatrix} 6 & 24 & 1 \\ 13 & 16 & 10 \\ 20 & 17 & 15 \end{pmatrix}}_{\text{Шифрующая матрица}} = \begin{pmatrix} 366 & 483 & 552 \\ 252 & 432 & 151 \\ 261 & 540 & 145 \\ 614 & 863 & 402 \\ 456 & 447 & 345 \\ 478 & 634 & 321 \end{pmatrix} \equiv \begin{pmatrix} 2 & 15 & 18 \\ 18 & 16 & 21 \\ 1 & 20 & 15 \\ 16 & 5 & 12 \\ 14 & 5 & 7 \\ 10 & 10 & 9 \end{pmatrix} (mod 26)$$

Шифротекст: *CPSSQVBUPQFMOFHKKJ*

Для демонстрации дешифровки, расшифруем полученный шифротекст «*CPSSQVBUPQFMOFHKKJ*»:

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25



Получаем открытый текст: *HILLCIPHEREXAMPLES.*

Задание

1. Найти шифр в CrypTool 1: *Encrypt/Decrypt-> Symmetric(Classic).*

2. Зашифровать и расшифровать текст содержащий только фамилию (транслитерация латиницей) вручную и с помощью шифра с выбранным ключом 2x2. Убедиться в совпадении результатов. Проверить обратимость шифрующей матрицы (ключа).

3. Зашифровать текст с произвольным сообщением в формате «DEAR MR *ФАМИЛИЯ ИМЯ ОТЧЕСТВО* THANK YOU VERY MUCH», используя транслитерацию латиницей и шифрующую матрицу 3x3.

4. Выполнить атаку на основе знания открытого текста, используя приложение из *Analysis-> Symmetric Encryption(classic)-> Known Plaintext.*

5. Удалить из сообщения и шифротекста фрагменты с *ФАМИЛИЯ ИМЯ ОТЧЕСТВО* и повторить атаку. Убедиться, что полученный ключ (матрица) совпадает с исходным.

6. Передайте произвольную шифровку коллеге для расшифрования при условии, что формы обращения и завершения сообщения известны. Размер использованного ключа держать в секрете.

Содержание раздела отчета

1. Исходное описание шифра (как в лекции). Пример вычисления шифрующей и расшифровывающей матрицы.

2. Реализация в CrypTool 1.0 (скриншот, спецификация параметров).

3. Пример работы шифра для выбранных параметров и текста сообщения.

4. Тип шифра (перестановка/замена/комбинированный, блочный/потоковый).

5. Ключ шифра.

6. Оценка сложности атаки “грубой силы”.

7. Описание атаки на шифр с использованием утилит Cryp Tool 1.0.

8. Результат атаки и расшифровка перехваченного от коллеги текста.

2.2 Комбинированный шифр ADFGVX

Шифр ADFGVX — один из самых известных шифров времён Первой мировой войны, который использовался немецкой армией. Особенность шифра заключается в том, что он построен на соединении базовых операций замены и перестановки.

Шифрование осуществляется в два этапа. На первом этапе задается матрица, заполненная символами алфавита, а также цифрами от 0 до 9. Далее каждый символ кодируется парой символов, на пересечении которых он находится. На втором этапе производится перестановка столбцов, заданная кодовым словом.

Например, зашифруем текст «*CIPHEREXAMPLE*» с кодовым словом «OURKEY»:

1. Составляем матрицу и кодируем каждый символ открытого текста

	a	d	f	g	v	x
a	a	b	**c**	d	e	f
d	g	h	i	j	k	l
f	m	n	o	p	q	r
g	s	t	u	v	w	x
v	y	z	0	1	2	3
x	4	5	6	7	8	9

Получаем последовательность: *AFDFFGDDAVFXAVGXAAFAFGDXAV*

2. Производим перестановку:

o	**u**	**r**	**k**	**e**	**y**		**e**	**k**	**o**	**r**	**u**	**y**

3	**5**	**4**	**2**	**1**	**6**		**1**	**2**	**3**	**4**	**5**	**6**
a	f	d	f	f	g		f	f	a	d	f	g
d	d	a	v	f	x	=>	f	v	d	a	d	x
a	v	g	x	a	a		a	x	a	g	v	a
f	a	f	g	d	x		d	g	f	f	a	x
a	v								a		v	

Выписываем текст по столбцам и формируем шифротекст: FFADFVXGADAFADAGFFDVAVGXAX

Задание

1. Найти шифр в CrypTool 1: *Encrypt/Decrypt-> Symmetric(Classic).*

2. Зашифровать и расшифровать текст содержащий только фамилию (транслитерация латиницей) вручную и с помощью шифра с выбранным ключом. Убедиться в совпадении результатов.

3. Выбрать абзац (примерно 600 символов) из файла *English.txt* (папка CrypTool/reference) и зашифровать его.

4. Выполнить атаку на шифротекст, используя приложение из *Analysis-> Symmetric Encryption(classic)-> Cipher Text Only.*

5. Повторить шифрование и атаку для тестов примерно в 300 и в 150 символов.

6. Изучите ручное расшифрование для текстов менее 300 символов.

7. Зашифруйте текст из 200 символов, сохраните ключ, и передайте соседу для расшифровки.

8. Самостоятельно изучите атаку по словарю, реализованную в CrypTool 2, опираясь на Help и ссылки на статьи.

Содержание раздела отчета

1. Исходное описание шифра (как в лекции).

2. Реализация в CrypTool 1.0 (скриншот, спецификация параметров).

3. Пример работы шифра для выбранных параметров.

4. Тип шифра (перестановка, замена, комбинированный).

5. Ключ шифра.

6. Оценка сложности атаки “грубой силы”.
7. Описание атаки на шифр с использованием утилит CrypTool 1.0.
8. Результат расшифровки перехваченного от коллеги текста.
9. Описание атаки на шифр реализованной в CrypTool 2.0.

2.3 Шифр Плейфера (Playfair)

Для работы алгоритма используется матрица 5*5 (если используется русский алфавит, то 4 * 8). В первую строку записывается кодовое слово (без повторения символов) слева направо или по спирали из верхнего левого угла к центру матрицы. Оставшиеся клетки матрицы заполняются незадействованными буквами алфавита в своем изначальном порядке.

Чтобы зашифровать текст его необходимо разбить на пары символов. Процесс шифрования подчиняется следующим правилам:

1. Если два символа совпадают или остался один символ, то к первому символу добавляется *X* и шифруется уже эта пара.
2. Если символы находятся в одной строке, то они замещаются на расположенные в ближайших от них справа символы.
3. Если символы в одном столбце, то они замещаются на расположенные ниже в ближайших от них клетках
4. Если символы находятся в разных углах образуемого ими прямоугольника, то они заменяются на символы, стоящие в противоположных углах этого прямоугольника, в тех же строках.

Расшифровка сообщения происходит инверсией данных правил.

Пример шифрования:

Открытый текст: *HELLO => HE LL O => HE LX LO*

ll	g	d	b	a
q	m	**h**	**e**	c
u	r	n	i	f
x	v	s	**o**	k
z	y	w	t	p

Шифрующая матрица

Шифротекст: ECQZBX.

Задание

1. Найти шифр в CrypTool 1: Encrypt/Decrypt-> Symmetric(Classic).

2. Зашифровать и расшифровать текст содержащий только фамилию (транслитерация латиницей) вручную и с помощью шифра с выбранной ключевой матрицей. Убедиться в совпадении результатов.

3. Зашифровать текст с произвольным сообщением в формате «DEAR ALL THANK YOU FOR ПРОИЗВОЛЬНЫЙ ТЕКСТ», используя выбранную шифрующую матрицу.

4. Выполнить атаку на основе знания части открытого текста, используя приложение из *Analysis-> Symmetric Encryption(classic)->Manual Analysis.* В качестве известного фрагмента текста использовать «DEAR ALL THANK YOU FOR»:

   a. Познакомьтесь с методикой проведения атаки в разделе Work through the examples из Help

   b. Познакомьтесь со спецификацией приложения для проведения атаки в разделе Analysis-> Symmetric Encryption(classic)->Manual Analysis->Playfair

5. Передайте произвольную шифровку соседу для расшифрования при условии, что форма обращения, используемая в сообщении, известна. Размер использованной матрицы (ключа) держать в секрете.

Содержание раздела отчета

1. Исходное описание шифра (как в лекции).

2. Реализация в Cryp Tool 1.0 (скриншот, спецификация параметров).

3. Пример работы шифра для выбранных параметров и текста сообщения.

4. Тип шифра (перестановка/замена/комбинированный, блочный/потоковый).

5. Ключ шифра.

6. Оценка сложности атаки “грубой силы”.

7. Описание методики атаки на шифр с использованием утилиты CrypTool 1.0.

8. Результат атаки и расшифровка перехваченного от коллеги текста.

2.4 Содержание отчета по лабораторной работе

В отчете следует сформулировать цель работы, наполнить каждый раздел необходимым содержанием и сделать краткие выводы по работе в заключении.

Лабораторная работа №4
Изучение шифра DES

Цель работы: исследовать шифры DES, 3DES, а также модификаций DESX, DESL, DESXL и получить практические навыки работы с ними, в том числе и в программном продукте Cryptool 1 и 2.

3.1 Исследование преобразований DES

Стандарт шифрования данных (DES) —блочный шифр с симметричными ключами, разработан Национальным Институтом Стандартов и Технологии (*NIST –National Instituteof Standards and Technology*).

Шифр DES основан на сети Фейстеля.

Алгоритм DES шифрует информацию блоками по 64 бита с помощью 64-битного ключа шифрования.

Шифрование выполняется следующим образом (рис. 4.1):

1. Над 64-битными блоками производится начальная перестановка согласно таблице.

2. Результат предыдущей операции делится на 2 субблока по 32 бита (A_0 и B_0), над которыми производятся 16 раундов преобразований:

$$A_i = B_{i-1};$$
$$B_i = A_{i-1} \oplus f(B_{i-1}, K_i),$$

Где: i- номер текущего раунда, K_i- ключ раунда, $\oplus$ - логическая операция xor.

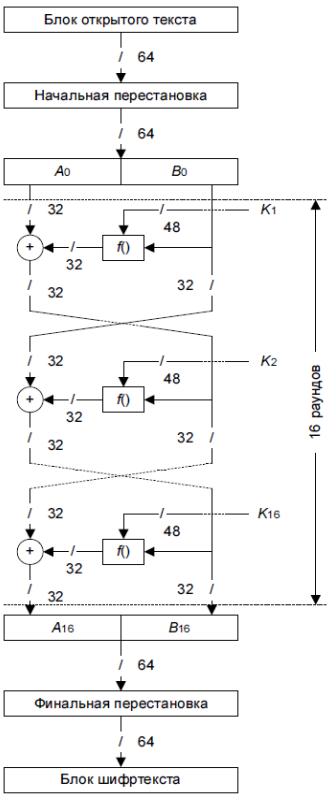


Рисунок 4.1

Структура функции раунда *f()* представлена на рисунке 4.2. Этапы функции:

a) Расширяющая перестановка *EP*, которая преобразует входные 32 бита в 48 бит (рис. 4.3).

b) Полученные 48 бит складываются с K_i операцией xor.

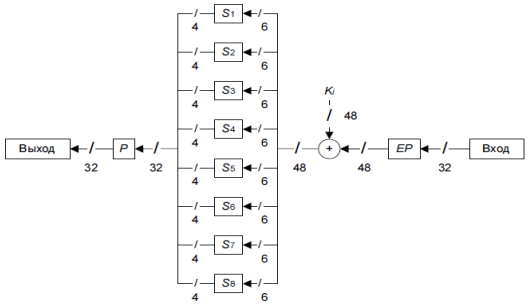


Рисунок 4.2

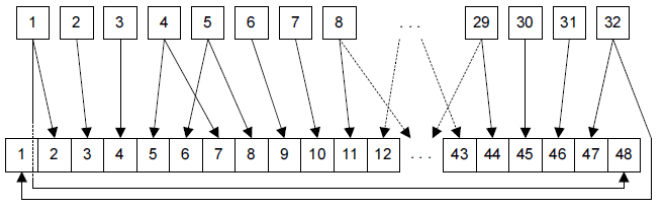


Рисунок 4.3

c) Результат сложения разбивается на 8 блоков по 6 битов. Каждый блок обрабатывается соответствующей таблицей замен.

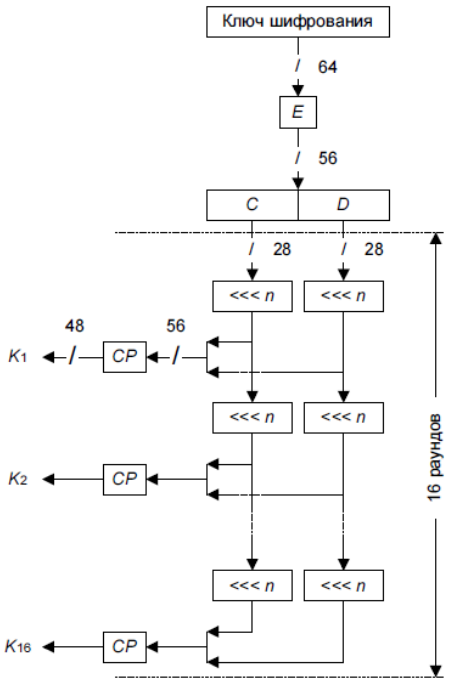


Рисунок 4.4

d) Над полученными 32 битами, после выполнения замен, выполняется перестановка (на рисунке 4.2 обозначена *P*).

На последнем раунде алгоритма субблоки не меняются местами.

3. Полученные субблоки A_{16} и B_{16} образуют 64-битный блок, над которым производится перестановка. Результатом перестановки является шифротекст.

Процедура генерации раундовых ключей представлена на рисунке 4.4. Из 64-битного ключа шифрования используется только 56 бит, каждый 8-й бит исключается. На рисунке 4.4 операция сжатия ключа и перестановка обозначена как *E*.

После перестановки блок в 56 бит делится на два 28-битных блока (*C* и *D*). Затем выполняются 16 раундов преобразований:

1. Текущие *C* и *D* циклически сдвигаются влево на определенное количество бит.
2. *C* и *D* объединяются в 56-битное значение, к которому применяется сжимающая перестановка. На выходе получаем 48-битный раундовый ключ.

Расшифровывание данных алгоритмом DES происходит при прохождении всех шагов алгоритма в обратном порядке.

Задание

1. Изучить преобразования шифра DES с помощью демонстрационного приложения из Cryptool 1.
    a. Indiv.Procedures-> Visualization…-> DES…
2. Выполнить вручную преобразования одного раунда и вычисление раундовых ключей при следующих исходных данных:
    a. Открытый текст (не более 64 бит) – фамилия_имя (транслитерация латиницей)
    b. Ключ (56 бит) – номер зачетной книжки II инициал (всего 7 символов)
3. Выполнить вручную обратное преобразование зашифрованного сообщения
4. Убедиться в совпадении результатов

Содержание раздела отчета

1. Формулировка задания и содержание этой части отчета.
2. Описание DES с примерами скриншотов из демоприложения.
3. Ручной расчет субблоков и раундовых ключей шифра для первых двух раундов.
4. Ручной расчет обратного преобразования шифровки.

3.2 Исследование DES в режимах ECB и CBC

Режим ECB шифра DES работает независимо с каждым 64-битным блоком шифруемых данных. Структура работы режима ECB представлена на рисунке 4.5.

Режим CBC шифра DES перед запуском шифрования каждого очередного блока складывает его с предыдущим операцией xor. Структура работы режима CBC представлена на рисунке 4.6.

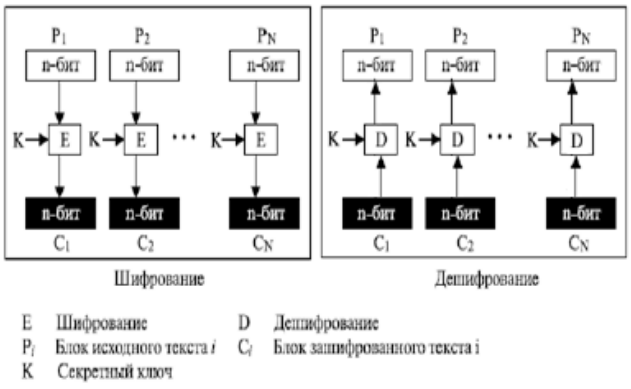


Рисунок 4.5

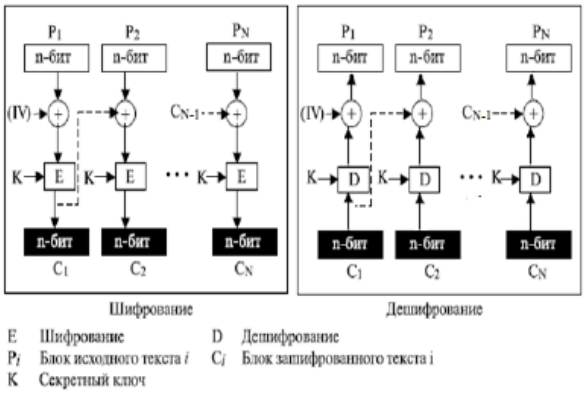


Рисунок 4.6

Задание

1. Создать картинку со своими ФИО (формат bmp).
2. Зашифровать картинку шифром DES в режиме ECB.
3. Зашифровать картинку шифром DES в режиме CBC с тем же ключом.
4. Сохранить скриншоты картинок для отчета.
5. Сжать исходную и 2 зашифрованных картинки средствами CrypTool. Зафиксировать размеры полученных файлов в таблице.
6. Выбрать случайный текст на английском языке (не менее 1000 знаков) и зашифровать его DES в режиме ECB.
7. Для одного и того же шифротекста оцените время проведения атаки «грубой силы» в случаях, когда известно n-4, n-6, n-8,.., 2 байт секретного ключа. Зафиксировать результаты измерений в таблице.
8. Повторить подобные измерения для DES в режиме CBC.

Содержание раздела отчета

1. Формулировка задания и содержание этой части отчета.
2. Основные параметры и обобщенная схема шифров (как в лекции).
3. Скриншоты исходного и шифрованных изображений в разных режимах работы шифров.
4. Таблица сравнений результатов сжатия исходного и шифрованных изображений.
5. Таблица зависимости оценки времени атаки грубой силы от размера известной части ключа.

3.3 Исследование 3-DES

Шифр 3-DES (рисунок 4.7) состоит в трехкратном применении обычного DES. Существует 4 основные версии данного шифра:

1. DES-EEE3 – шифрование происходит 3 раза независимыми ключами
2. DES-EDE3 – операции шифровка-расшифровка-шифровка с тремя разными ключами
3. DES-EEE2 – то же что и DES-EEE3, но на первом и последнем шаге одинаковый ключ
4. DES-EDE2 – то же что и DES-EDE3, но на первом и последнем шаге одинаковый ключ

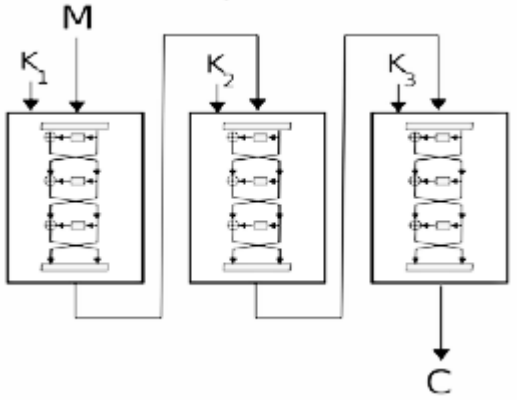


Рисунок 4.7

На текущий момент самыми популярными разновидностями шифра являются DES-EDE3 и DES-EDE2.

Задание

1. Выбрать случайный текст на английском языке (не менее 1000 знаков).
2. Создать бинарный файл с этим текстом, зашифровав и расшифровав его DES на 0-м ключе.

3. Снять и сохранить частотную и автокорреляционную характеристику этого файла.

4. Зашифровать бинарный файл шифром 3-DES в режиме ECB.

5. Снять и сохранить частотную и автокорреляционную характеристику файла с шифровкой.

6. Зашифровать исходный бинарный файл 3-DES в режиме CBC с тем же ключом.

7. Снять и сохранить частотную и автокорреляционную характеристику файла с шифровкой.

8. Определить экспериментальным путем по какой схеме работает реализация 3-DES в CrypTool. Сохранить подтверждающие скриншоты.

Содержание раздела отчета

1. Формулировка задания и содержание этой части отчета.

2. Основные параметры и обобщенная схема шифра (как в лекции).

3. Скриншоты исходного и зашифрованного текста в бинарном представлении.

4. Скриншоты частотной и автокорреляционной характеристик исходного текста и шифровки.

5. Таблица зависимости времени атаки грубой силы от размера известной части ключа.

6. Схема реализации в CrypTool 1 и подтверждающие скриншоты.

3.4 Исследование модификаций DESX, DESL, DESXL шифра DES

Алгоритм DESX использует на входе ключ длиной 184 бита, который делится на 3 56-битные части. Процесс шифрования происходит по следующей схеме:

$$DESX(M) = K_2 \oplus DES_K(M \oplus K_1)$$

Если $K_1 = K_2 = 0$, то данный алгоритм сводится к стандартному DES.

Алгоритм DESL является облегченной версией алгоритма DES. Данный алгоритм был создан в 2006 году для RFID-меток. Алгоритм предполагает отказ от входной и выходной перестановки блока текста, т.к. они не несут криптографической сложности, а также 8 S-блоков заменяется на 1, но более стойкий чем все 8 стандартных блока DES.

Алгоритм DESXL использует те же оптимизации что и DESL, но производит шифрование по алгоритму DESX.

Задание

1. Выбрать случайный текст на английском языке (не менее 1000 знаков).

2. Создать бинарный файл с этим текстом, зашифровав и расшифровав его DES на 0-м ключе.

3. С помощью CrypTool зашифровать текст с использованием шифров DESX, DESL, DESXL.

4. Средствами CrypTool вычислить энтропию исходного текста и шифротекстов, полученных в итоге. Зафиксировать результаты измерений в таблице.

5. Средствами CrypTool оцените время проведения атаки «грубой силы» при полном отсутствии информации о секретном ключе

Содержание раздела отчета

1. Формулировка задания и содержание этой части отчета.

2. Основные параметры и обобщенные схемы шифров (самостоятельно).

3. Таблица зависимости энтропии шифротекста от используемого шифра.

4. Таблица зависимости времени подбора ключа от используемого шифра.

3.5 Содержание отчета по лабораторной работе

В отчете следует сформулировать цель работы, наполнить каждый раздел необходимым содержанием и сделать краткие выводы по работе в заключении.

Лабораторная работа №5
Изучение шифра AES

Цель работы: исследовать шифр AES, финалистов конкурса AES, атаку предсказанием дополнения и получить практические навыки работы с шифрами и атакой, в том числе и в программном продукте Cryptool 1 и 2.

4.1 Исследование преобразований AES

Шифр Rijndael (AES) работает на основе перестановочно-подстановочной сети (SP-сеть). Обобщенная схема работы алгоритма представлена на рисунке 5.1.

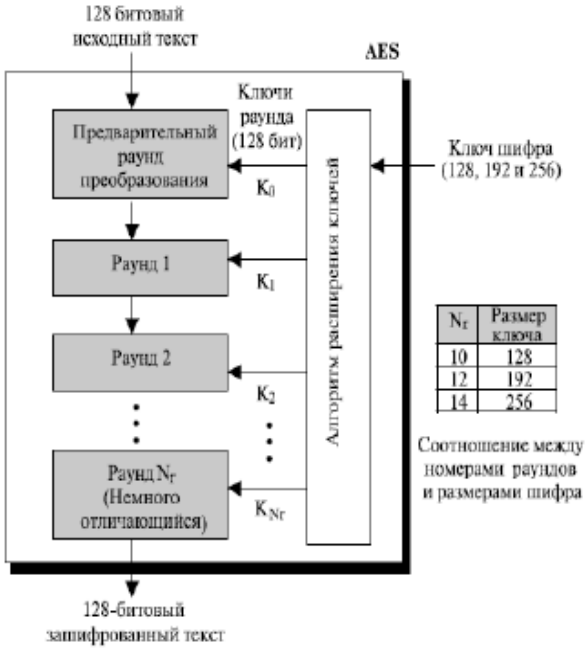


Рисунок 5.1

В версии с наименьшей длиной ключа алгоритм AES получает на вход блок открытого текста размером 16 байт и 16 байт ключа. Значения блока записывается в столбцы матрицы состояний размером 4x4 байт.

Процедура расширения ключей ExpandKey создает последовательно (слово за словом) 128 битные раундовые ключи от единственного входного ключа шифра.

После того, как сформированы раундовые ключи, начинается раундовая обработка матрицы состояний. В каждом раунде алгоритма выполняются следующие преобразования, представленные на рисунке 5.2:

1. Столбцы матрицы состояний складываются с ключом шифра операцией xor.

2. Полученный текст проходит через преобразование подстановки SubBytes.

3. Циклический сдвиг влево всех строк матрицы состояний преобразованием ShiftRows .

4. Смешивание столбцов матрицы состояний путем ее умножения на матрицу констант в конечном поле $GF(2^8$– преобразовние MixColumns Сложение полученных столбцов матрицы состояний с раундовым ключом операцией xor – преобразование AddRoundKey

5. Действия 2-5 повторяются в каждом из 10-ти раундов

6. Последний раунд не включает в себя смешивание столбцов.

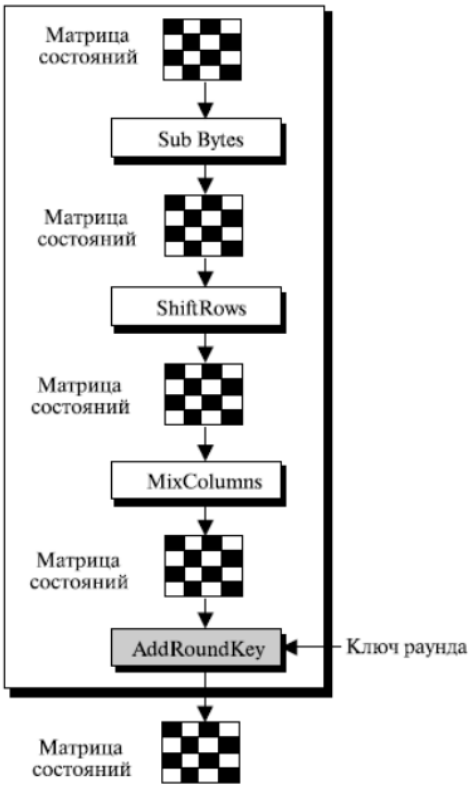


Рисунок 5.2

Расшифровывание выполняется применением обратных операций в обратной последовательности.

Задание

1. Изучить преобразования шифра AES с помощью демонстрационного приложения из Cryptool 1: *Indiv.Procedures->Visualization…->AES->Rijndael Animation.*

2. Выполнить вручную преобразования для одного раунда и вычисление раундового ключа при следующих исходных данных:

   a. Открытый текст – фамилия_имя (транслитерация латиницей)

   b. Ключ – номер группы_отчество

3. Проверить полученные результаты с помощью приложения-инспектора: *Indiv.Procedures->Visualization…->AES->Rijndael Inspector.*

4. Провести наблюдения в потоковой модели шифра AES с помощью демонстрационного приложения из CrypTool 1 для 0-текста и 0-ключа: *Indiv.Procedures->Visualization…->AES->Rijndael Flow Visualisation*

Содержание раздела отчета

1. Формулировка задания.

2. Описание AES с примерами скриншотов из демоприложения.

3. Расчет матрицы состояний и раундового ключа шифра для одного раунда.

4. Скриншоты приложения-инспектора, подтверждающие корректность расчетов.

5. Выводы.

4.2 Исследование финалистов конкурса AES (AES, MARS, RC6, Serpent, Twofish)

Победителем конкурса AES я валяется алгоритм Rijndael (AES), т.к по всем характеристикам алгоритм Rijndael (AES), не уступает остальным алгоритмам-финалистам. Остальные финалисты конкурса (Serpent, Twofish, MARS и RC6), практически равнозначны по совокупности характеристик, за исключением алгоритма MARS, имеющего существенно больше недостатков, в том числе алгоритм практически нереализуем в условиях ограниченных ресурсов. Более подробную информацию можно найти в лекциях и [1].

Задание

1. Выбрать текст на английском языке (не более 120 знаков)

2. Создать бинарный файл с этим текстом, зашифровав и расшифровав его шифром AES на 0-м ключе

3. С помощью Cryptool 1 зашифровать с ключом отличным от 0 текст с использованием шифров AES, MARS, RC6, Serpent и Twofish

4. Приложением из Cryptool 1 вычислить энтропию исходного текста и шифротекстов, полученных в итоге. Зафиксировать результаты измерений в таблице

5. Приложением из Cryptool 1 оцените время проведения атаки «грубой силы» всех шифров для одного и того же шифротекста в случаях, когда известно n-2, n-4, n-6,..., 2 байт секретного ключа. Зафиксировать результаты измерений в таблице.

Содержание раздела отчета

1. Формулировка задания.
2. Исходные данные для экспериментов:
    a. Исходный текст
    b. Секретный ключ
3. Таблица с результатами качества зашифрования исследованными шифрами.
4. Таблица с результатами трудоемкости атаки «грубой силы» для исследованных шифров.
5. Выводы.

4.3 Атака «грубой силы» на AES

Задание

1. Найти и запустить шаблон атаки в CrypTool 2: *AES Analysis using Entropy(2).*
2. Выбрать открытый текст (примерно 1000 знаков) и загрузить его в шаблон.
3. Провести атаку «грубой силы» когда известно n-2, n-4, n-6 байт секретного ключа, используя в качестве оценочной функции энтропию и задействовав 1 ядро процессора. Зафиксировать затраты времени.
4. Выполнить атаку повторно с средним и максимальным количеством процессорных ядер. Зафиксировать затраты времени.
5. Сформировать текст с произвольным сообщением в формате «DEAR SIRS message THANKS» и загрузить его в шаблон.
6. Провести атаку «грубой силы» когда известно n-2, n-4, n-6 байт секретного ключа, используя в качестве оценочной функции

словосочетание DEAR SIRS задействовав 1 ядро процессора. Зафиксировать затраты времени.

7. Выполнить атаку повторно с средним и максимальным количеством процессорных ядер. Зафиксировать затраты времени.

Содержание раздела отчета

1. Формулировка задания.
2. Исходные данные для экспериментов:
    a. Исходный текст
    b. Секретный ключ
3. Шаблон атаки «грубой силы» из Cryptool 2.
4. Таблица с результатами трудоемкости энтропийной атаки «грубой силы» для различных вариантов знаний о ключе и количестве задействованных процессорных ядер.
5. Таблица с результатами трудоемкости текстовой атаки «грубой силы» для различных вариантов знаний о ключе и количестве задействованных процессорных ядер.
6. Выводы.

4.4 Атака предсказанием дополнения на шифр AES в режиме CBC (Padding Oracle Attack)

При атаке Padding Oracle предполагается, что мы можем отправлять сообщения серверу на расшифровку, который может возвращать ответ, корректно ли выполнено дополнение последнего блока. Расшифровка сообщения происходит с последних блоков шифротекста.

Рассмотрим расшифровку блока C_{i+1}.

1. Формируем R – все биты, кроме последнего, случайные значения. Перебираем байт R_n от 0x00 до 0xFF, каждый раз посылая на сервер $[R||C_{i+1}]$. Если при некотором R_n сервер «одобряет», то $T_n = 01, S_n = R_n \oplus 0x01, p_n = S_n \oplus C_n$. Схема первого этапа представлена на рисунке 5.3.

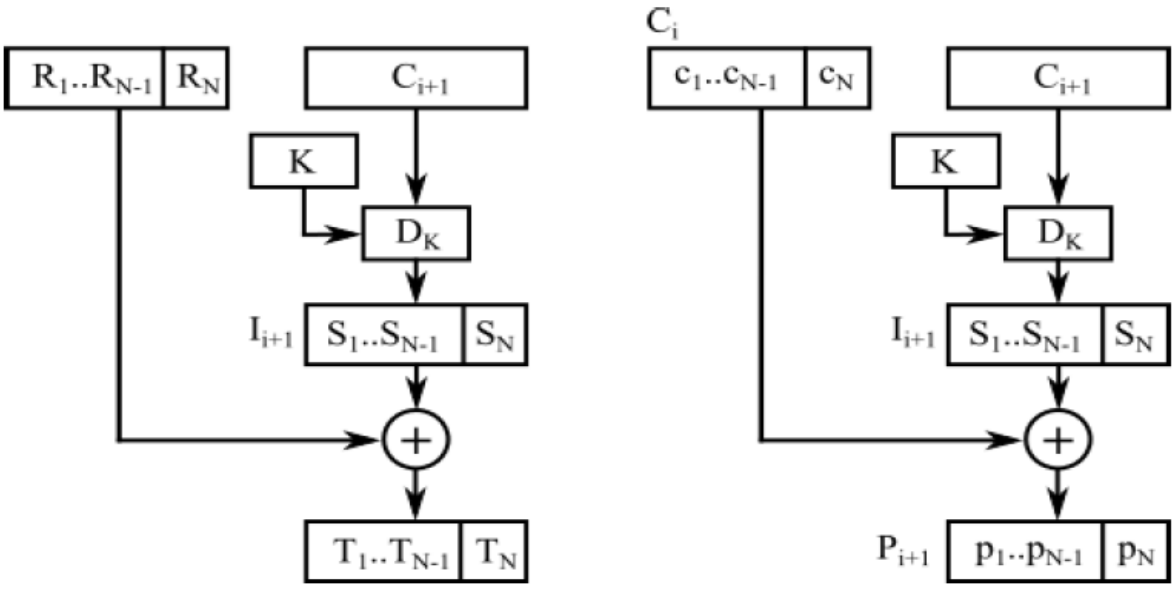


Рисунок 5.3

P_i – *открытый текст,* C_i *– шифротекст,* I_i *– дополнение,* K *– ключ,* D_i *– функция расшифровки,* T_i *– дополнение.*

2. *Формируем* R – все биты, кроме двух последних, случайные значения. $R_n = S_n \oplus 0x02$, чтобы $T_n = 02$. Перебираем байт R_{n-1} от 0x00 до 0xFF, каждый раз посылая на сервер $[R||C_{i+1}]$. Если при некотором R_{n-2} сервер «одобряет», то $T_{n-1} = 02, S_n = R_{n-1} \oplus 0x02, p_{n-1} = S_{n-1} \oplus C_{n-1}$.

На третьем шаге пытаемся получить дополнение 030303, на четвертом – 04040404. После N шагов получаем полностью блок p_{i+1}.

Более подробное описание атаки Padding Oracle можно найти в статье [4].

В CryoTool 2 атака предсказанием дополнения реализована в три фазы:

1. *Фаза 1*. Нахождение длины дополнения, т.е. последний байт
2. *Фаза 2*. Подбор дополнения
3. *Фаза 3*. Расшифровка текста

Задание

1. Найти и запустить шаблон атаки в CrypTool 2: *Padding Oracle Attack on AES.*
2. Подготовьтесь к атаке теоретически:
   a. Изучите комментарии к шаблону

b. Изучите публикацию [4]

3. Внедрите во второй блок исходного текста коды символов своего имени.

4. Выполните 3 фазы атаки и сохраните итоговые скриншоты по окончанию каждой фазы.

5. Убедитесь, что атака удалась.

Содержание раздела отчета

1. Формулировка задания.
2. Исходные данные для экспериментов:
   a. Исходный текст
   b. Секретный ключ
3. Шаблон атаки «Padding Oracle Attack» из CrypTool 2.
4. Описание атаки «Padding Oracle Attack».
5. Результаты 3-х фазы атак в виде итоговых скриншотов ПО.
6. Выводы.

4.5 Содержание отчета по лабораторной работе

В отчете следует сформулировать цель работы, наполнить каждый раздел необходимым содержанием и сделать краткие выводы по работе в заключении.

Лабораторная работа №6
Изучение хэш-функций

Цель работы: исследование хэш-функций MD5, SHA-256, SHA-512, SHA-3, кода контроля целостности HMAC и анализ атак дополнительной коллизии на хэш-функцию. Получить практические навыки работы с хэш-функциями и атакой на них, в том числе и в программном продукте Cryptool 1 и 2.

5.1 Исследование лавинного эффекта MD5, SHA-1, SHA-256, SHA-512

Хэш-функцией (hashfunction) называется математическая или иная функция, которая для строки произвольной длины вычисляет некоторое целое значение или некоторую другую строку фиксированной длины.

Хэш-значение может также называться дайджестом (*digest*) или отпечатком (*fingerprint*) сообщения.

Алгоритмы хэширования также называют бесключевыми дайджестами сообщений (*nonkeyedmessagedigest*).

Однонаправленная функция *H(M)* применяется к сообщению длины M и возвращает значение фиксированной длины h. ($h = H(M)$, где h имеет длину m). Однонаправленные функции имеют дополнительные свойства, позволяющие отличать их от обычных функций, которые вычисляют значение фиксированной длины по входным данным:

1. Зная M, легко вычислить h.
2. Зная H, трудно определить M, для которого $H(M) = h$,
3. Зная M, трудно определить другое сообщение, M1, для которого $H(M) = H(M1)$.

Задание

1. Открыть текст не менее 1000 знаков. Добавить свое ФИО последней строкой. Перейти к утилите Indiv.Procedures->Hash->Hash Demonstration..

2. Задать хэш-функцию, подлежащую исследованию: MD5, SHA-1, SHA-256, SHA-512.?,

3. Для каждой хэш-функции повторить следующие действия:

   a. Измените (добавлением, заменой, удалением символа) исходный файле

   b. Зафиксировать количество измененных битов в дайджесте модифицированного сообщения.

   c. Вернуть сообщение в исходное состояние.

4. Выполните процедуру 3 раза (добавлением, заменой, удалением символа) и подсчитайте среднее количество измененных бит дайджеста. Зафиксировать результаты в таблице.

Содержание раздела отчета

1. Формулировка задания
2. Основные параметры и обобщенная схема хэш-функций MD5, SHA-1 (см. лекцию)
3. Таблица с фактическими и усредненными параметрами лавинного эффекта для исследованных хэш-функций
4. Выводы

5.2 Хэш-функция SHA-3

В основе Keccak (SHA-3) лежит конструкция под названием Sponge – губка. Сам алгоритм состоит из 2-х этапов:

1. *Впитывание – Absording.* На каждом шаге очередной блок сообщения p_i длиной r подмешивается к части внутреннего состояния *S*, которая затем целиком модифицируется функцией *f*-многораундовой бесключевой псевдослучайной перестановкой.

2. *Отжатие – Squeezing.* Чтобы получить хэш, функция *f* многократно применяется к состоянию, и на каждом шаге сохраняется кусок размера *r* до тех пор, пока не получим выход *Z* необходимой длины (путем конкатенации).

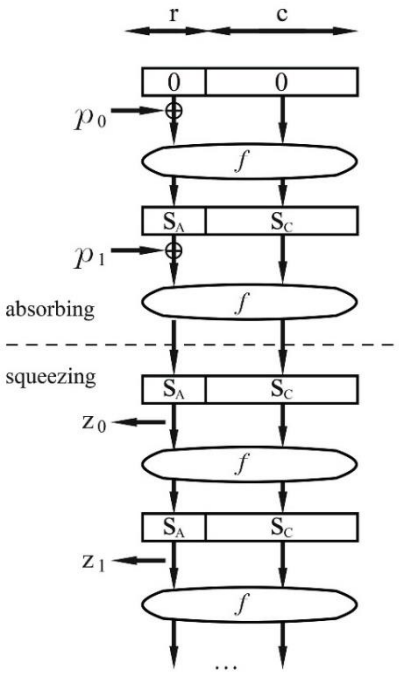


Рисунок 6.1

Обобщенная схема работы алгоритма представлена на рисунке 6.1.

Задание

1. Открыть шаблон *Keccak Hash (SHA-3)* в Cryptool 2
2. В модуле Keccak сделать следующие настройки:
   a. Adjust manually=ON
   b. Keccak version= SHA3-512

3. Загрузить файл из предыдущего задания

4. Запустить проигрывание шаблона в режиме ручного управления:

   a. Сохранить скриншоты преобразований первого раунда

   b. Сохранить скриншот заключительной фазы

   c. Сохранить значение дайджеста

5. Вычислить значения дайджеста для модифицированных текстов из предыдущего задания

6. Подсчитать лавинный эффект с помощью самостоятельно разработанной автоматизированной процедуры

Содержание раздела отчета

1. Формулировка задания

2. Основные параметры и обобщенная схема хэш-функции Keccak Hash, (SHA-3) на основе изученной презентации

3. Описание средства оценивания лавинного эффекта

4. Таблица с фактическими параметрами лавинного эффекта

5. Выводы

5.3 Контроль целостности по коду HMAC

HMAC - один из механизмов проверки целостности информации, позволяющий гарантировать то, что данные, передаваемые или хранящиеся в ненадёжной среде, не были изменены посторонними лицами. Механизм HMAC использует MAC — стандарт, описывающий способ обмена данными и способ проверки целостности передаваемых данных с использованием секретного ключа. Два клиента, использующие HMAC, как правило, разделяют общий секретный ключ. HMAC — надстройка над MAC.

Алгоритм HMAC можно записать в виде одной формулы:

$$HMAC_K(text) = H\big((K \oplus opad)||H((K \oplus ipad)||text)\big),$$

где $\oplus$ — операция xor, ‖- конкатенация, K— секретный ключ, $ipad$ — блок вида (0x36 0x36 0x36 ... 0x36), где байт 0x36 повторяется b раз, H — хэш-

функция, *opad* — блок вида (0x5c 0x5c 0x5c ... 0x5c), где байт 0x5c повторяется *b* раз.

Задание

1. Выбрать текст на английском языке (не менее 1000 знаков), добавить собственное ФИО и сохранить в файле формата .TXT

2. Придумать пароль и сгенерировать секретный ключ утилитой *Indiv.Procedures->Hash-> Key Generation* из Cryptool 1. Сохранить ключ в файле формата .TXT. Прочитать Help к этой утилите.

3. Сгенерировать HMAC для имеющегося текста и ключа с помощью утилиты *Indiv.Procedures->Hash-> Generation of HMACs*. Сохранить HMAC в файле формата .TXT. Прочитать Help к этой утилите.

4. Передать пароль, HMAC (и его характеристики), исходный текст и модифицированный текст коллеге, не раскрывая, какой текст является корректным. Попросите коллегу определить это самостоятельно.

Содержание раздела отчета

1. Формулировка задания.

2. Выбранная схема генерации ключа и ее параметры.

3. Выбранная схема создания HMAC.

4. Описание действий передающий стороны на примере выполненного задания.

5. Описание действий принимающей стороны на примере выполненного задания.

6. Выводы.

5.4 Атака дополнительной коллизии на хэш-функцию

Атака дополнительной коллизии основана на одной из проблем парадокса дня рождений. Он заключается в следующем: в группе, состоящей из 23=c*√365 или более человек, вероятность совпадения дней рождения (число и месяц) хотя бы у двух людей превышает 50 %. Применительно к хэш-функции это означает, что сложность атаки, целью которой является поиск

двух сообщений с одинаковыми значением хэш-функции, пропорционально $\sqrt{2^N}$, где N – длина хэш-кода.

Задание

1. Сформировать два текста на английском языке - один истинный, а другой фальсифицированный. Сохранить тексты в файлах формата *.txt

2. Утилитой *Analysis-> Attack on the hash value...* произвести модификацию сообщений для получения одинакового дайджеста. В качестве метода модификации выбрать *Attach characters-> Printable characters.*

3. Проверить, что дайджесты сообщений действительно совпадают с заданной точностью.

4. Сохранить исходные тексты, итоговые тексты и статистику атаки для отчета.

5. Зафиксировать временную сложность атаки для 8, 16, 32,40, 48, … бит совпадающих частей дайджестов.

Содержание раздела отчета

1. Формулировка задания.

2. Описание атаки в терминах парадокса «дня рождения» (см. лекцию).

3. Представление результатов атаки: исходные и модифицированные тексты, статистика, дайджесты исходных и модифицированных сообщений.

4. Таблица с оценками временной сложности атаки.

5. Выводы.

5.5 Содержание отчета по лабораторной работе

В отчете следует сформулировать цель работы, наполнить каждый раздел необходимым содержанием и сделать краткие выводы по работе в заключении.

Лабораторная работа №7
Изучение ассиметричных шифров

Цель работы: исследовать протокол Диффи-Хеллмана, шифр RSA и получить практические навыки работы с ними, в том числе и в программном продукте CrypTool 1.

6.1 Протокола Диффи-Хеллмана

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.

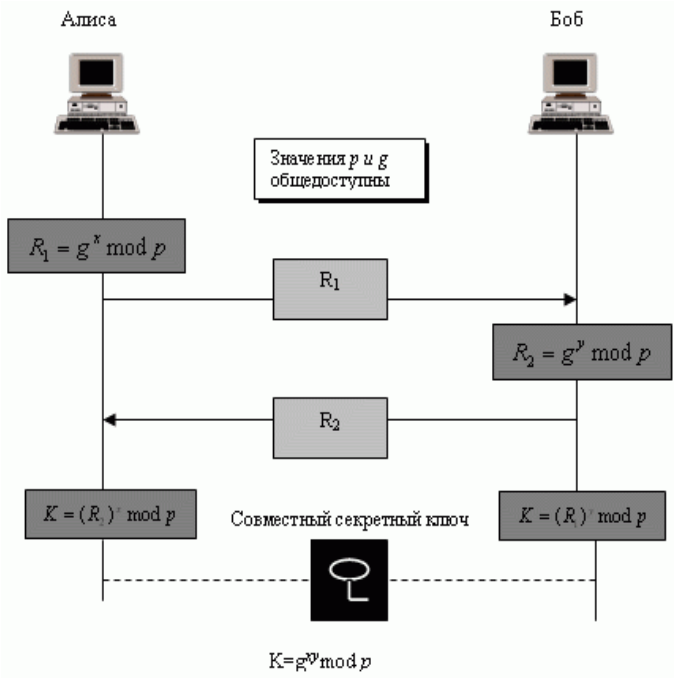


Рисунок 7.1

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

1. Устанавливаются открытые параметры p, g:

a) p – большое простое число порядка 300 десятичных цифр (1024 бита),

b) g – первообразный корень по модулю p.

2. Каждая из сторон генерирует большое число x и y соответственно.

3. На каждой стороне вычисляется открытый ключ:

a) $R_1 = g^x mod\ p$,

b) $R_2 = g^y mod\ p$.

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

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

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.

Содержание раздела отчета

1. Основные параметры протокола.

2. Скриншот схемы протокола, реализованной в CrypTool.

3. Таблица соответствия схемы протокола (CrypTool) и параметров протокола.

4. Скриншот исходного, зашифрованного и расшифрованного текстов.

6.2 Шифр RSA

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

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

1. Вычисление ключей:

   a) Генерация двух больших простых чисел p и q (p и q держаться в секрете).

   b) Вычисление n = p * q

   c) Выбор произвольного e (e <n), взаимно простого с $\varphi(n)$.

   d) Вычисление $d: e * d = 1\ mod\ \varphi(n)$.

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

2. Шифрование:

a) Открытый текст разбивается на блоки $m_i: m_i < n$.

b) Каждый блок открытого текста преобразуем в шифротекст по формуле:

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

3. Расшифровка:

a) Шифротекст разбивается на блоки $c_i: c_i < n$.

b) Каждый блок шифротекста преобразуем в открытый текст по формуле:

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

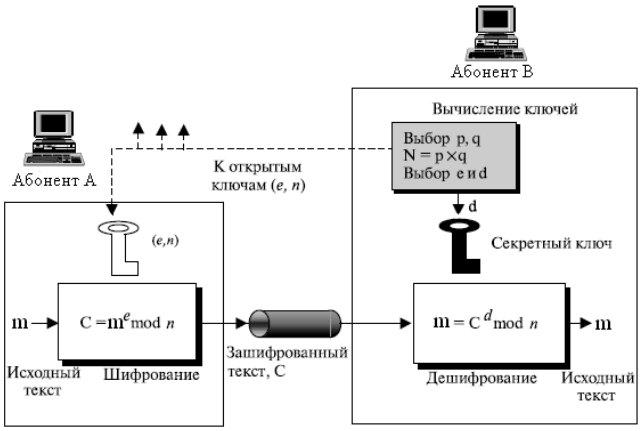


Рисунок 7.2

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.

Содержание раздела отчета

1. Обобщенная схема шифра.
2. Скриншот результата генерации ключей.
3. Скриншот результата шифрации.
4. Скриншот результата расшифровки.

6.3 Исследование шифра RSA

Задание

1. Выбрать текст на английском языке (не менее 1000 знаков) и сохранить в файле формата *.txt

2. Сгенерировать пары ассиметричных 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.

Содержание раздела отчета

1. Выбранный текст.
2. Результаты генерации ключевых пар различной длины.
3. Размер исходного текста.
4. Таблица затрат времени на зашифровку и расшифровку при использовании ключей разной длины.

6.4 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 *c*, *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.

Содержание раздела отчета

1. Исходные данные для атаки, полученные от коллеги.

2. Результат факторизации (скриншот).

3. Расшифрованное в итоге сообщение.

6.5 Имитация атаки на гибридную криптосистему

Модель гибридной криптосистемы представлена на рисунке 7.3. Шифрование при помощи гибридной модели осуществляется следующим образом:

1. Сообщение шифруется симметричным секретным ключом.

2. Секретный ключ шифруется открытым ключом.

3. Зашифрованное сообщение и ключ составляют цифровой конверт, который отправляется получателю.

4. Получатель сначала расшифровывает секретный ключ, затем расшифровывает секретным ключом шифротекст сообщения.

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.

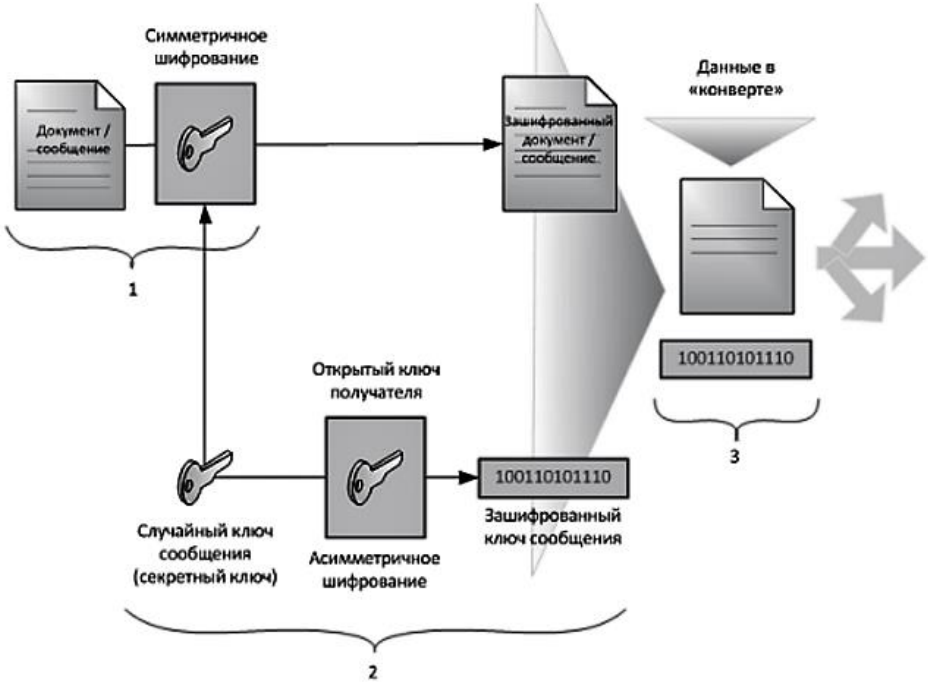


Рисунок 7.3

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.

Содержание раздела отчета

1. Описание цели атаки
2. Текст передаваемого сообщения
3. Лог-файлы участников протокола

6.6 Содержание отчета по лабораторной работе

В отчете следует сформулировать цель работы, наполнить каждый раздел необходимым содержанием и сделать краткие выводы по работе в заключении.

Лабораторная работа №8
Изучение цифровой подписи

Цель работы: исследовать алгоритмы создания и проверки цифровой подписи, алгоритмы генерации ключевых пар RSA, DSA, ECDSA и получить практические навыки работы с ними, в том числе и в программном продукте CrypTool 1.

7.1 Генераторов ключевых пар

Генерация ключевых RSA

Генерация двух больших простых чисел p и q (p и q держаться в секрете).

1. Вычисление n = p * q
2. Выбор произвольного e (e <n), взаимно простого с $\varphi(n)$.
3. Вычисление $d: e * d = 1 \, mod \, \varphi(n)$.
4. Числа (e, n) – открытый ключ, d – закрытый ключ, p и q уничтожаются.

Генерация ключевых пар DSA

1. Выбирается число p: длина - [512,1024] битов и число битов в p должно быть кратно 64.
2. Выбирается число q, которое имеет тот же самый размер дайджеста 160 битов, такое, что: $(p-1) = 0 \, mod \, q$.
3. Выбирается $e_1: {e_1}^q = 1 \, mod \, p$.

4. Выбирается целое число $d < q$ и вычисляется $e_2 = {e_1}^d \ mod \ p$.

5. Числа (e_1, e_2, p, q)- открытый ключ, d – закрытый ключ.

Генерация ключей ECDSA

1. Выбирается эллиптическая кривая $E_p(a, b), p -$ простое число.

2. Выбирается простое число q – порядок одной из циклических подгрупп группы точек эллиптической кривой: $q \times (x_0, y_0) = O$.

3. Выбирается закрытый ключ d.

4. Выбирается точка на кривой $e_1 = (x_1, y_1)$.

5. Вычисляется точка на кривой $e_2 = d \times e_1$.

6. Открытый ключ - (a, b, q, p, e_1, e_2).

Задание

1. Перейти к утилите *«Digital Signatures/PKI->PKI/Generate…»*.

2. Сгенерировать ключевые пары по алгоритмам RSA-2048, DSA-2048, EC-239. Зафиксируйте время генерации в таблице.

3. С помощью утилиты *«Digital Signatures/PKI-> PKI/Display…»* вывести сгенерированный открытый ключ и сохранить соответствующий скриншот.

Содержание раздела отчета

1. Описание алгоритмов генерации (как в лекции).

2. Таблица с фактическими временем генерации ключевых пар.

3. Скриншоты со значениями открытых ключей.

7.2 Процессы создания и проверки цифровой подписи

Обобщенные схемы подписания и проверки цифровой подписи представлены на рисунке 8.1.

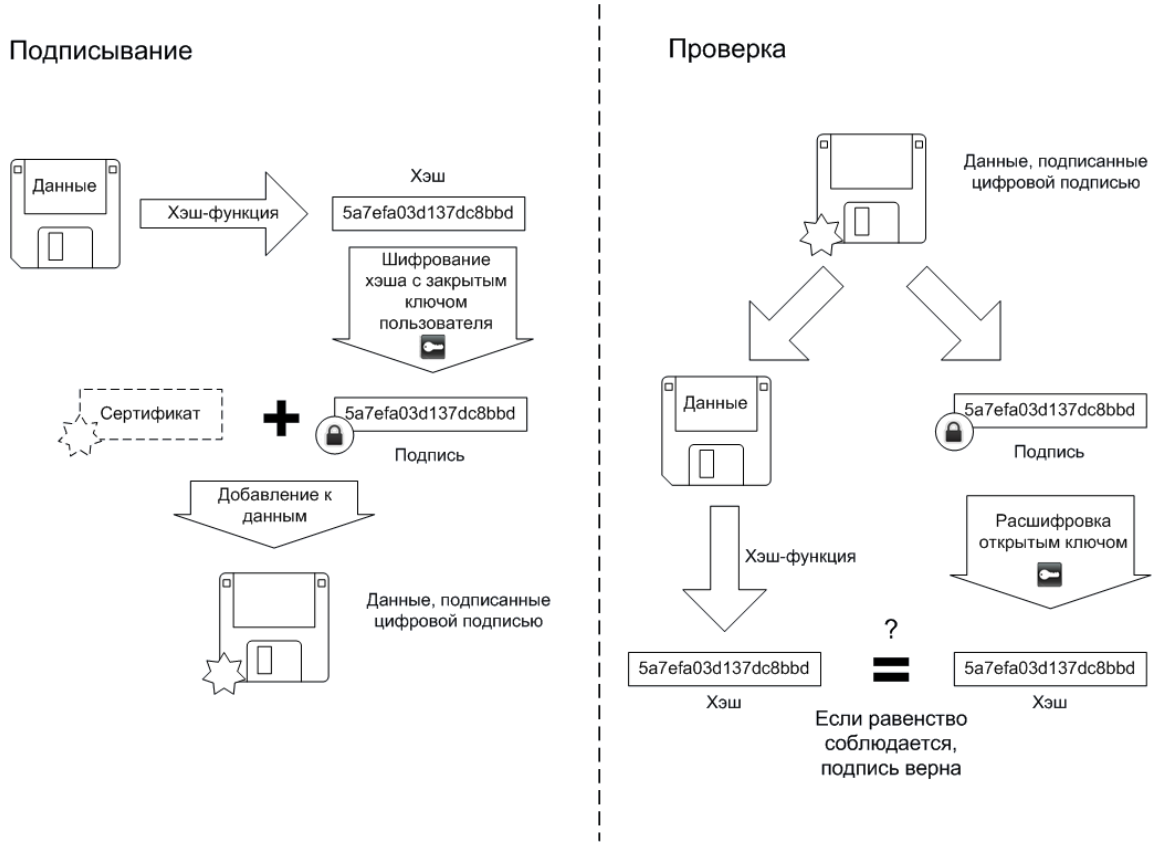


Рисунок 8.1

Задание

1. Открыть текст не менее 5000 знаков. Перейти к приложению *Digital Signatures/PKI-> Sign Document…*

2. Задайте хэш-функцию, и другие параметры цифровой подписи.

3. Создайте подпись ключами, сгенерированными в предыдущем задании. Зафиксируйте время создания цифровой подписи для каждого ключа.

4. Сохраните скриншот цифровой подписи с помощью приложения *Digital Signatures/PKI-> Extract Signature*.

5. Выполните процедуру проверки подписи *Digital Signatures/PKI-> Verify Signature* для случаев сохранения и нарушения целостности исходного текста. Сохраните скриншоты результатов.

Содержание раздела отчета

1. Обобщенная схема создания и проверки цифровой подписи.

2. Таблица с фактическими временем генерации цифровой подписи.
3. Скриншоты со значениями цифровой подписи.
4. Скриншоты с результатами проверки цифрой подписи.

7.3 Схемы цифровой подписи на эллиптических кривых

Схема цифровой подписи ECDSA (рисунок 8.2)

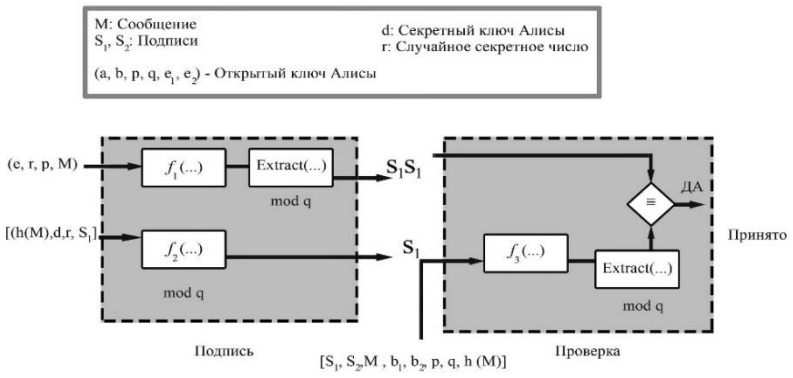


Рисунок 8.2

В процессе подписания две функции f_1 и f_2 и экстрактор Extract создают две части подписи. В процессе проверки (верификации) обрабатывают выход одной функции f_2 (после прохождения через экстрактор) и сравнивают ее с первой частью подписи.

После того, как сгенерирована ключевая пара (закрытый ключ - d, и открытый ключ - (a, b, q, p, e_1, e_2) (см. раздел 8.1), осуществляется подписание документа, затем на принимающей стороне осуществляется проверка (рисунок 8.3).

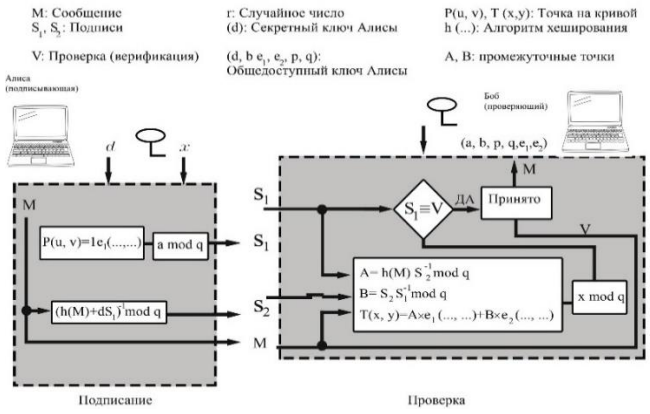


Рисунок 8.3

Алгоритм подписания ECDSA состоит из следующих операций:

1. Выбирается секретное случайное число $r: r \in (1, q-1)$
2. Выбирается третья точка на кривой: $P(u, v) = r \times e_1$
3. Вычисляется первая часть подписи по формуле:

$$S_1 = u \; mod \; q \; ,$$

где *u*- абсцисса.

4. Вычисляется вторая часть подписи по формуле:

$$S_2 = (h(M) + d \times S_1) \times r^{-1} \; mod \; q \; ,$$

где $h(M)$- дайджест сообщения, *d*- закрытый ключ.

Алгоритм проверки цифровой подписи ECDSA включает следующие операции:

1. Вычисляем промежуточные результаты *A* и *B*:

$$A = h(M) \times {S_2}^{-1} mod \; q$$

$$B = {S_2}^{-1} \times S_1 \; mod \; q$$

2. Восстанавливаем третью точку:

$$T(x, y) = A \times e_1 + B \times e_2$$

3. Верификатор $V = x \; mod \; q$ сравнивается с первой частью цифровой подписи S_1.

Задание

1. Выполните процедуру создание подписи *«Digital Signatures/PKI-> Sign Document…»* алгоритмом ECSP-DSA в пошаговом режиме (*Display inter. results=ON*). Зафиксируйте скриншоты последовательности шагов.

2. Выполните процедуру проверки подписи ECSP-DSA для случаев сохранения и нарушения целостности исходного текста. Сохраните скриншоты результатов.

3. Проверить лекционный материал по ECDSA, выполнив создание и проверку подписи сообщения *M* (принять *M=h(M)*) приложением *Indiv.Procedures->Number Theory…->Point Addition on EC.*

Содержание раздела отчета

1. Описание алгоритма формирования и проверки подписи ECDSA.

2. Результаты (скриншоты) пошагового выполнения ECDSA в CrypTool 1. Сравнение лекционной версии и реализации.

3. Результаты проверки лекционного материала по ECDSA с использованием приложения *Indiv.Procedures->Number Theory…->Point Addition on EC.*

7.4 Демонстрация процесса подписи в среде PKI

Инфраструктура открытых ключей *(ИОК, PKI –Public Key Infrastructure)* —набор средств (технических, материальных, организационных и т. д.), распределённых служб и компонентов, в совокупности используемых для поддержки решения криптозадач на основе закрытого и открытого ключей, а именно:

1. Обеспечение конфиденциальности информации;

2. Обеспечение целостности информации;

3. Обеспечение аутентификации пользователей и ресурсов, к которым обращаются пользователи;

4. Обеспечение возможности подтверждения совершенных пользователями действий

Одним из решений криптозадач на основе открытого и закрытого ключей является использование сертификатов. Сертификат—это электронный документ, который содержит:

1. Открытый ключ пользователя (открытый ключ)

2. Информацию о пользователе, которому принадлежит сертификат

3. Информацию о сроке действия сертификата

4. Другие атрибуты

5. Цифровую подпись удостоверяющего центра, выдавшего сертификат

Существует несколько вариантов использования сертификатов электронной подписи:

1. Для зашифрования и расшифрования электронных документов
2. Для подписания электронного документа и проверки подписи
3. Для подписания и шифрования электронных документов.

Задание

1. Запустить демонстрационную утилиту *«Digital Signatures/PKI-> Signature Demonstration…»*.

2. Получите сертификат на ранее сгенерированную ключевую пару RSA-2048.

3. Выполните и сохраните скриншоты всех этапов создания цифровой подписи документа.

4. Сохраните скриншот сертификата для проверки этой цифровой подписи.

Содержание раздела отчета

1. Сравнение структуры сертификата (как в лекции) и сертификата из CrypTool 1.0.

2. Схема процедуры подписания из CrypTool.

7.5 Подписание своего отчета

Задание

1. Сконвертируйте отчет в формат pdf.

2. Экспортируйте ранее созданный сертификат ключевой пары *RSA Digital Signatures/PKI->PKI/Generate…->Export PSE(#PKCS12).*

3. Откройте pdf-версию отчета и попытайтесь подписать с использованием этого сертификата.

4. Создайте собственный самоподписанный сертификат в среде Adobe Reader и используйте его для подписи отчета.

5. Сохраните скриншоты свойств подписи и сертификата.

6. Внесите изменения (маркеры, комментарии) в отчет и проверьте подпись.

Содержание раздела отчета

1. Скриншот титульного листа с цифровой подписью.
2. Скриншоты свойств подписи и сертификата.
3. Скриншот результата проверки после внесения изменений в отчет.

7.6 Содержание отчета по лабораторной работе

В отчете следует сформулировать цель работы, наполнить каждый раздел необходимым содержанием и сделать краткие выводы по работе в заключении.

Рекомендуемая литература

1. Сергей Панасенко. Алгоритмы шифрования. Специальный справочник.

2. Венбо Мао. Современная криптография: теория и практика.

3. Вильям Столлингс. Криптография и безопасность сетей: принципы и практика.

4. https://habrahabr.ru/post/247527/

5. Брюс Шнайер. Прикладная криптография.

6. Н. Н. Токарева. Симметричная криптография. Краткий курс.

Лабораторный практикум

Племянников Александр Кимович,
Кузнецова Екатерина Олеговна.

Криптографические методы защиты информации

Издание публикуется в авторской редакции

СПбГЭТУ «ЛЭТИ»
197376, Санкт-Петербург, ул. Проф. Попова, 5