МИНОБРНАУКИ РОССИИ

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

А. К. ПЛЕМЯННИКОВ

КРИПТОГРАФИЧЕСКИЕ МЕТОДЫ ЗАЩИТЫ ИНФОРМАЦИИ

Учебно-методическое пособие

Санкт-Петербург
Издательство СПбГЭТУ «ЛЭТИ»
2022

УДК 004.056.055(07)
ББК 3973.2-082.03 я7
П38

Племянников А. К.

П38 Криптографические методы защиты информации: учеб.-метод. пособие. СПб.: Изд-во СПбГЭТУ «ЛЭТИ», 2022. 51 с.

ISBN 978-5-7629-3111-3

Содержит описание лабораторных работ, в которых изучаются и исследуются классические и современные симметричные шифры, хеш-функции, асимметричные шифры, протоколы создания и проверки электронной подписи и цифровые сертификаты ключей для проверки подписей. Предназначено для студентов, обучающихся по специальностям 10.05.01 «Компьютерная безопасность», 01.03.02 «Прикладная математика и информатика».

УДК 004.056.055(07)
ББК 3973.2-082.03 я7

Рецензент д-р техн. наук, доцент Е. Б. Александрова (СПбГПУ Петра Великого).

Утверждено
редакционно-издательским советом университета
в качестве учебно-методического пособия

ISBN 978-5-7629-3111-3

Лабораторная работа 1.
ИЗУЧЕНИЕ КЛАССИЧЕСКИХ ШИФРОВ RAILFENCE, SCYTALE, CAESAR

Цель работы: исследовать шифры Rail Fence, Scytale, Caesar и получить практические навыки работы с ними, в том числе с использованием приложений CrypTool 1 и 2.

1.1. Шифр «Изгородь» (Rail Fence)

В этом шифре открытый текст вписывается в таблицу-шаблон, содержащую заданное количество строк – высоту изгороди. В каждую строку поочередно записывается одна буква со смещением от левого края шаблона, подобно изгороди. Зашифрованный текст создается объединением наборов символов из различных строк таблицы шаблона. Например, при помещении открытого текста *«0123456789»* в шаблон из 3 строк результат шифрования выглядит следующим образом:

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

Для увеличения криптостойкости этого шифра можно использовать смещение при записи открытого текста в шаблон. Выполним шифрование из примера выше со смещением 2:

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

Задание

1. Найти шифр в CrypTool 1: Encrypt/Decrypt –> Symmetric(Classis) –> Scytal/Rail Fence.

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. Выполнить самостоятельную работу:

1) создать шифровку для варианта *Offset=0* и *Number of Rows≤ 5* и обменяться с коллегой из учебной группы для дешифровки;

2) определить ключ методом «грубой силы» и расшифровать полученный от коллеги шифротекст.

Содержание раздела отчета

1. Задание.
2. Реализация в CrypTool 1 (скриншот, спецификация параметров).
3. Схема, поясняющая работу шифра.
4. Пример работы шифра для выбранных параметров.
5. Основные характеристики шифра:
   а) тип шифра (перестановка, замена, комбинированный);
   б) ключ шифра;
   в) оценка сложности атаки «грубой силы».

1.2. Шифр «Сцитала» (Scytale)

В криптографии шифр «Сцитала», известный также как шифр Древней Спарты, представляет собой прибор, используемый для осуществления перестановочного шифрования. Прибор состоит из граненого цилиндра (жезла) и узкой полоски пергамента, которая обматывается вокруг цилиндра по спирали. На гранях цилиндра записывалось сообщение. Иллюстрация, демонстрирующая работу данного шифра, представлена на рис. 1.1.

Рис. 1.1

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

Задание

1. Найти шифр в CrypTool 1: Encrypt/Decrypt –> Symmetric(Classic) Scytal/Rail Fence.
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. Выполнить самостоятельную работу: 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 (скриншот, спецификация параметров).
3. Схема, поясняющая работу шифра.
4. Пример работы шифра для выбранных параметров.
5. Основные характеристики шифра:
   а) тип шифра (перестановка, замена, комбинированный);
   б) ключ шифра;
   в) оценка сложности атаки «грубой силы».

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.

Шифрование основано на использовании таблицы замен: в одну строку таблицы записываются буквы алфавита, а в другую – тот же алфавит, но сдвинутый влево на выбранное значение смещения. Символ, находящийся под символом исходного алфавита, – это заменяющий символ в шифротексте.

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

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**

После замены получаем шифротекст: *НУМТХЖУГЧМВ*.

Задание

1. Найти шифр в CrypTool 1: Encrypt/Decrypt –> Symmetric(Classic) –> Caesar/Rot-13.

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 for Analysis.

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. Выполнить самостоятельную работу: обменяться шифровками с коллегой по группе для проведения подобной атаке по дешифрации сообщения.

Содержание раздела отчета

1. Задание.

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

3. Схема и математические формулы, поясняющие работу шифра.

4. Пример работы шифра для выбранных параметров.

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

6. Основные характеристики шифра:

   а) тип шифра (перестановка, замена, комбинированный);

   б) ключ шифра;

   в) 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 каждому разделу отчета 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 с использованием приложений CrypTool 1 и 2.

2.1. Шифр моноалфавитной подстановки (Substitution)

Шифрование основано на использовании таблицы замен: в одну строку таблицы записываются буквы алфавита языка исходного сообщения, а в другую строку – символы, на которые заменяются буквы исходного сообщения.

В CrypTool-реализации шифра моноалфавитной подстановки для задания таблицы замен используются алфавит исходного сообщения и параметры *Key* и *Offset*. Key – это кодовое слово, на основе которого формируется алфавит шифротекста. Первым шагом создания нового алфавита служит удаление всех повторяющихся букв, которые присутствуют в кодовом слове. Затем из алфавита удаляются все буквы кодового слова. На заключительном шаге кодовое слово внедряется в алфавит со смещением первого элемента кодового слова на величину параметра *Offset*.

Например, зашифруем текст «*CRYPTOGRAPHY*», используя кодовое слово «*PASSWORD*», смещение 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

После замены получаем шифротекст: *EMYKQJAMBKSY*.

Задание

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

2. Зашифровать и расшифровать текст, содержащий только вашу 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 шифротекста.

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 150 символов.

7. Изучить возможности CrypTool 1 для автоматизации выполнения ручного расшифрования для текстов размером менее 300 символов.

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

9. Дешифровать этот абзац, используя приложение Analysis –> Tools for Analysis и Analysis –> Symmetric Encryption(classic) –> Manual Analysis.

10. Выполнить самостоятельную работу:

а) зашифровать текст из 200 символов, сохранить ключ и обменяться шифровками с коллегой по учебной группе для дешифровки;

б) изучить одну из атак, реализованных в CrypTool 1 и 2, опираясь на Help и ссылки на статьи.

Содержание раздела отчета

1. Задание.
2. Реализация в CrypTool 1 (скриншот, спецификация параметров).
3. Пример работы шифра для выбранных параметров.
4. Основные характеристики шифра:
   а) тип шифра (перестановка, замена, комбинированный);
   б) ключ шифра;
   в) оценка сложности атаки «грубой силы».
5. Описание атаки на шифр с использованием утилит CrypTool 1.

2.2. Шифр двойной перестановки (Permutation/Transposition)

В основе шифра лежит перестановка матричного представления открытого текста. Перестановки можно выполнять по строкам или по столбцам, а также обоими способами.

Например, зашифруем текст «*ПРИМЕРМАРШРУТНЫЙШИФР*», с использованием сперва перестановки по столбцам (5, 4, 3, 1, 2) и далее – по строкам (2, 4, 3, 1):

1. Записать 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. Переставить столбцы:

	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. Переставить строки:

	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

Выписав текст строку за строкой, получаем шифротекст: *ФРИШЙМЕИРПНЫТУРРШАМР*.

Задание

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

2. Зашифровать и расшифровать текст, содержащий ваши 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. Выполнить самостоятельную работу:

а) зашифровать текст с произвольным сообщением в формате «DEAR message THANKS», используя только одинарную перестановку. Обменяться подобными шифровками с коллегой по учебной группе для дешифровки при условии, что формы обращения и завершения письма известны;

б) самостоятельно изучить атаку, реализованную в CrypTool 2, опираясь на Help и ссылки на статьи.

Содержание раздела отчета

1. Задание.
2. Реализация в CrypTool 1 (скриншот, спецификация параметров).
3. Пример работы шифра для выбранных параметров.
4. Основные характеристики шифра:

а) тип шифра (перестановка, замена, комбинированный);

б) ключ шифра;

в) оценка сложности атаки «грубой силы».

5. Описание атаки на шифр с использованием утилит CrypTool 1.

2.3. Шифр Виженера (Vigenere)

Шифр Виженера – способ многоалфавитного шифрования текста с использованием ключевого слова. Можно рассматривать шифр Виженера как состоящий из последовательности нескольких шифров Цезаря с различными значениями сдвига алфавитов.

Для зашифровывания используется таблица замен, в которой каждой букве алфавита языка исходного сообщения ставится в соответствие несколько вариантов букв для представления в шифротексте. Для этого выбирается кодовое слово длиной *n*, которое делит открытый текст на отрезки такой же длины. Далее составляется так называемая таблица Виженера: горизонтально записывается алфавит языка исходного сообщения, а вертикально под первым символом алфавита записывается кодовое слово. Заполнение таблицы осуществляется символами алфавита, начинающегося с очередной буквы кодового слова и циклически замыкающегося (т. е. применительно к латинице это выглядит так: …xyzabc…). Элемент шифротекста выбирается на пересечении столбца, соответствующего букве открытого текста, и строки, соответствующей букве кодового слова.

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

1. Разделить текст на отрезки:

п	р	и	м
к	л	ю	ч

е	р	ш	и
к	л	ю	ч

ф	р	а	в
к	л	ю	ч

и	ж	е	н
к	л	ю	ч

е	р	а	
к	л	ю	

2. Произвести замену (для 1-го 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	и
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. Аналогично заменяем все отрезки, получаем итоговый шифротекст: *ШЪЖВПЪЦЯЭЪЮЩТСГГПЪЮ*.

Задание

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

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

3. Провести 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. Воспроизвести эту атаку в автоматизированном режиме:

а) определить размер ключа с помощью приложения Analysis –> Tools for Analysis –> Autocorrelation;

б) выполнить перестановку текста с размером столбца, равным размеру ключа, приложением *Permutation/Transposition*;

в) определить очередную букву ключа приложением Analysis –> Symmetric Encryption(Classic) –> Cipher Text Only –> Caesar.

6. Выполнить самостоятельную работу: изучить атаку, реализованную в CrypTool 2, опираясь на Help и ссылки на статьи.

Содержание раздела отчета

1. Задание.
2. Схема и математические формулы, поясняющие работу шифра.
3. Пример работы шифра для выбранных параметров.
4. Основные характеристики шифра:

а) тип шифра (перестановка, замена, комбинированный);

б) ключ шифра;

в) оценка сложности атаки «грубой силы».

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

2.4. Содержание отчета по лабораторной работе

В отчете следует сформулировать цель работы, наполнить каждый раздел необходимым содержанием и сделать краткие выводы по каждому разделу отчета 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 с использованием приложения CrypTool 1 и 2.

3.1. Шифр Хилла (Hill)

Шифр Хилла основан на матричном преобразовании текста. Перед шифрованием необходимо каждому символу алфавита следует сопоставить код равный порядковому номеру символа в алфавите. Затем коды символов открытого текста записываются в матрицу размером $n \times m$ и создается шифрующая матрица $n \times n$. Для шифрования матрица открытого текста умножается на шифрующую матрицу и вычисляется остаток от деления значения элементов матрицы-произведения на число символов выбранного алфавита. Для расшифровки необходимо шифротекст умножить на матрицу, которая является мультипликативной инверсией по отношению к шифрующей для выбранного алфавита.

В качестве примера зашифруем текст «*HILLCIPHEREXAMPLES*»:

a	b	c	d	e	f	g	h	i	j	k	l	m	n
0	1	2	3	4	5	6	7	8	9	10	11	12	13

o	p	q	r	s	t	u	v	w	x	y	z
14	15	16	17	18	19	20	21	22	23	24	25

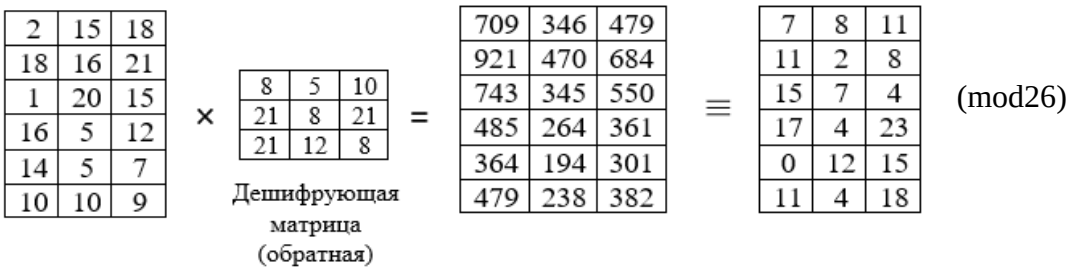
7	8	11		6	24	1		366	483	552		2	15	18	
11	2	8		13	16	10		252	432	151		18	16	21	
15	7	4		20	17	15		261	540	145		1	20	15	
17	4	23	×	Шифрующая матрица			=	614	863	402	≡	16	5	12	(mod26)
0	12	15						456	447	345		14	5	7	
11	4	18						478	634	321		10	10	9	

Шифротекст: *CPSSQVBUPQFMOFHKKJ*.

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

a	b	c	d	e	f	g	h	i	j	k	l	m	n
0	1	2	3	4	5	6	7	8	9	10	11	12	13

o	p	q	r	s	t	u	v	w	x	y	z
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 вашу фамилию (транслитерация латиницей), вручную и с помощью шифра с выбранным ключом 2 × 2. 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 3 × 3.

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

5. Выполнить самостоятельную работу: обменяться шифровками с коллегой по учебной группе для дешифрования при условии, что формы обращения и завершения сообщения известны. Размерность использованного ключа держать в секрете.

Содержание раздела отчета

1. Задание.

2. Схема и математические формулы, поясняющие работу шифра. Пример вычисления шифрующей и расшифровывающей матриц.

3. Реализация в CrypTool 1 (скриншот, спецификация параметров).

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

5. Основные характеристики шифра:

а) тип шифра (перестановка, замена, комбинированный) и размер блока данных;

б) ключ шифра;

в) оценка сложности атаки «грубой силы».

6. Описание атаки на шифр с использованием утилит Cryp Tool 1.

3.2. Комбинированный шифр ADFGVX

Шифр ADFGVX – один из самых известных шифров времен Первой мировой войны, который использовался немецкой армией. Особенность шифра заключается в том, что он основан на комбинации базовых операций замены и перестановки.

Шифрование осуществляется в два этапа. На первом этапе сначала задается матрица-ключ 6 × 6, заполненная произвольно символами алфавита и цифрами от 0 до 9. Индексами строк и столбцов этой матрицы служат буквы A, D, F, G, V, X. Далее каждый символ открытого текста кодируется парой буквенных индексов, на пересечении которых в матрице-ключе он находится. На втором этапе задается кодовое слово для перестановки столбцов, а затем ранее закодированный открытый текст переписывается построчно в матрицу с числом столбцов, равным длине ключевого слова. В завершение столбцы этой матрицы переставляются в соответствии с лексикографическим порядком букв ключевого слова и итоговый шифротекст образуется конкатенацией строк этой матрицы.

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

1. Составить матрицу и кодировать 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

Получится последовательность: *AFDFFGDDAVFXAVGXAAFAFGDXAV*.

2. Произвести перестановку:

o	**u**	**r**	**k**	**e**	**y**
3	**5**	**4**	**2**	**1**	**6**
a	f	d	f	f	g
d	d	a	v	f	x
a	v	g	x	a	a
f	a	f	g	d	x
a	v				

=>

e	**k**	**o**	**r**	**u**	**y**
1	**2**	**3**	**4**	**5**	**6**
f	f	a	d	f	g
f	v	d	a	d	x
a	x	a	g	v	a
d	g	f	f	a	x
		a		v	

Выписать текст по столбцам и сформировать шифротекст: *FFADFVXG ADAFADAGFFDVAVGXAX*.

Задание

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

2. Зашифровать и расшифровать текст, содержащий только вашу 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 150 символов.

6. Изучить инструмент автоматизации ручного расшифрования для текстов менее 300 символов.

7. Выполнить самостоятельную работу:

а) зашифровать текст из 200 символов, сохранить ключ, и обменяться шифровками с коллегой по группе для дешифровки;

б) самостоятельно изучить атаку по словарю, реализованную в CrypTool 2, опираясь на Help и ссылки на статьи.

Содержание раздела отчета

1. Задание.
2. Описание шифра.
3. Реализация в CrypTool 1 (скриншот, спецификация параметров).
4. Пример работы шифра для выбранных параметров.
5. Основные характеристики шифра:

а) тип шифра (перестановка, замена, комбинированный);

б) ключ шифра;

в) оценка сложности атаки «грубой силы».

6. Описание атаки на шифр с использованием утилит CrypTool 1.

3.3. Шифр Плейфера (Playfair)

Для работы алгоритма шифрования используется матрица 5 × 5 (для алфавита русского языка – 4 × 8), в которую в произвольном порядке записываются символы алфавита. Этот порядок можно задать кодовым словом. В таком случае в первую строку записывается кодовое слово (без повторения символов) слева направо или по спирали из верхнего левого угла к центру

матрицы, а оставшиеся клетки матрицы заполняются незадействованными буквами алфавита в их изначальном порядке.

Чтобы зашифровать текст, его необходимо разбить на пары символов (блоки). 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 символы, расположенные в ближайших от них ячейках справа.

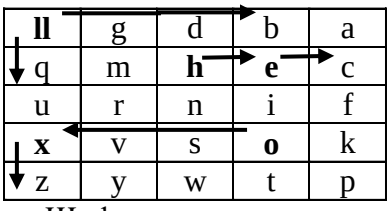
3. Если символы находятся в одном столбце, то они замещаются на расположенные ниже в ближайших от них клетках

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*



Шифрующая матрица

Шифротекст: ECQZBX.

Задание

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

2. Зашифровать и расшифровать текст, содержащий только вашу 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»:

а) познакомиться с методикой проведения атаки в разделе Work through the examples из Help;

б) познакомиться со спецификацией приложения для проведения атаки в разделе Analysis –> Symmetric Encryption(classic) –> Manual Analysis –> Playfair.

5. Выполнить самостоятельную работу: обменяться произвольными шифровками с коллегой по группе для дешифрования при условии, что форма обращения, используемая в сообщении, известна. Размер использованной матрицы (ключа) держать в секрете.

Содержание раздела отчета

1. Задание.
2. Описание шифра.
3. Реализация в Cryp Tool 1 (скриншот, спецификация параметров).
4. Пример работы шифра для выбранных параметров и текста сообщения.
5. Основные характеристики шифра:

а) тип шифра (перестановка, замена, комбинированный) и размер блока данных;

б) ключ шифра;

в) оценка сложности атаки «грубой силы».

6. Описание методики атаки на шифр с использованием утилиты CrypTool 1.

3.4. Содержание отчета по лабораторной работе

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

Лабораторная работа 4.
ИЗУЧЕНИЕ ШИФРА DES

Цель работы: исследовать шифры DES, 3DES, а также другие модификации шифра DES: DESX, DESL, DESXL и получить практические навыки работы с ними, в том числе с использованием приложения CrypTool версий 1 и 2.

4.1. Исследование преобразований DES

Стандарт шифрования данных DES [1] – блочный симметричный шифр, разработанный Национальным институтом стандартов и технологии (NIST – National Institute of Standards and Technology).

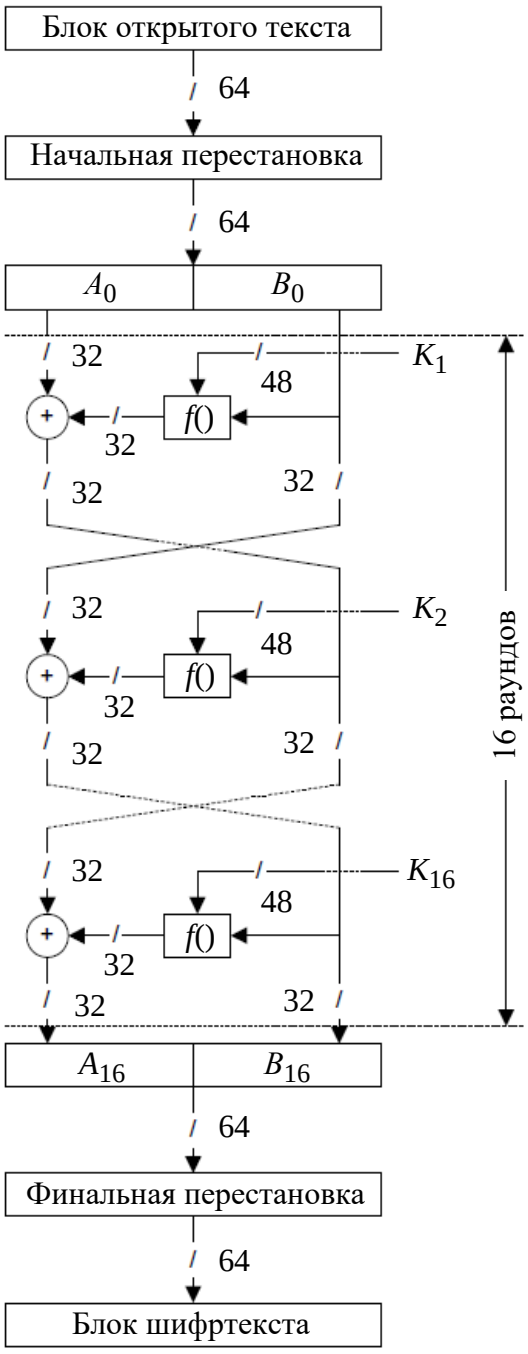


Рис. 4.1

Шифр DES основан на сети Фейстеля.

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, задаваемая таблично.

2. После начальной перестановки блок делится на 2 субблока по 32 бит (A_0 и B_0), над которыми проводятся 16 раундов преобразований:

$$A_i = B_{i-1};$$

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

где i – номер текущего раунда; K_i – ключ раунда; $\oplus$ – логическая операция XOR.

Схема работы функции раунда $f()$ представлена на рис. 4.2. Этапы раундового преобразования следующие:

а) расширяющая перестановка EP, которая преобразует входные 32 бит в 48 бит (рис. 4.3);

б) полученные 48 бит складываются с K_i операцией xor;

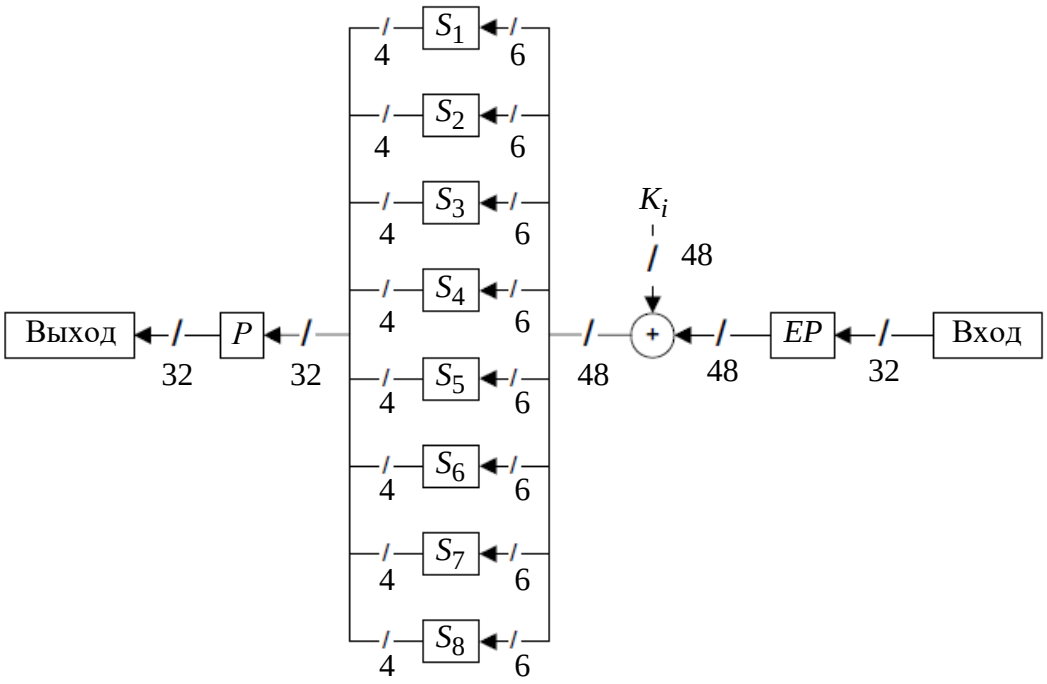


Рис. 4.2

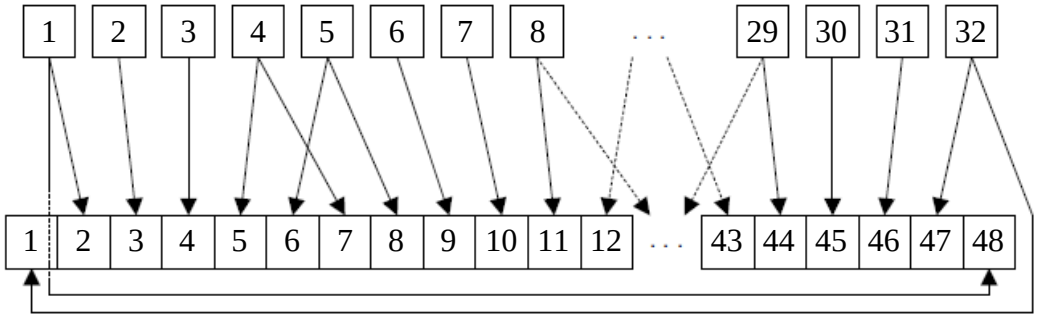


Рис. 4.3

в) результат сложения разбивается на 8 блоков по 6 бит. Каждый блок обрабатывается соответствующей таблицей замен;

г) над полученными 32 бит, после выполнения замен, выполняется перестановка (на рис. 4.2 обозначена *P*).

На последнем раунде алгоритма субблоки местами не меняются.

3. Полученные в итоге субблоки A_{16} и B_{16} образуют 64-битный блок, над которым производится конечная перестановка и в итоге получается результирующий блок шифротекста.

Процедура генерации раундовых ключей представлена на рис. 4.4. Из 64-битного ключа шифрования используется только 56 бит, каждый 8-й бит исключается. На рис. 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:

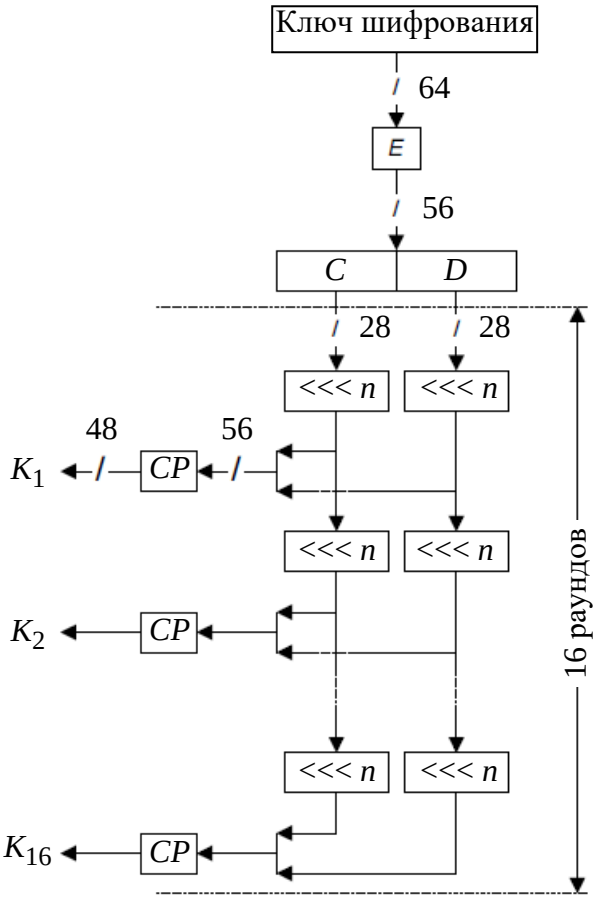


Рис. 4.4

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:

Indiv.Procedures –>
Visualization… –> DES…

2. Выполнить вручную преобразования первого раунда и вычисление раундовых ключей при следующих исходных данных:

а) открытый текст (не более 64 бит) – ваши фамилия_имя (транслитерация латиницей);

б) ключ (56 бит) – номер зачетной книжки и инициал отчества (всего 7 символов).

3. Выполнить вручную обратное преобразование зашифрованного сообщения.

4. Убедиться в совпадении результатов.

Содержание раздела отчета

1. Задание.

2. Основные характеристики и описание DES.

3. Ручной расчет субблоков и раундовых ключей шифра для первого раунда. Сравнение с результатами демо-приложения.

4. Ручной расчет обратного преобразования шифровки.

4.2. Исследование DES в режимах ECB и CBC

Режимы использования симметричных блочных шифров предназначены для зашифрования больших файлов данных. В режиме ECB шифр DES используется независимо для каждого 64-битного блока исходного файла данных. Схема использования шифра в режиме ECB представлена на рис. 4.5.

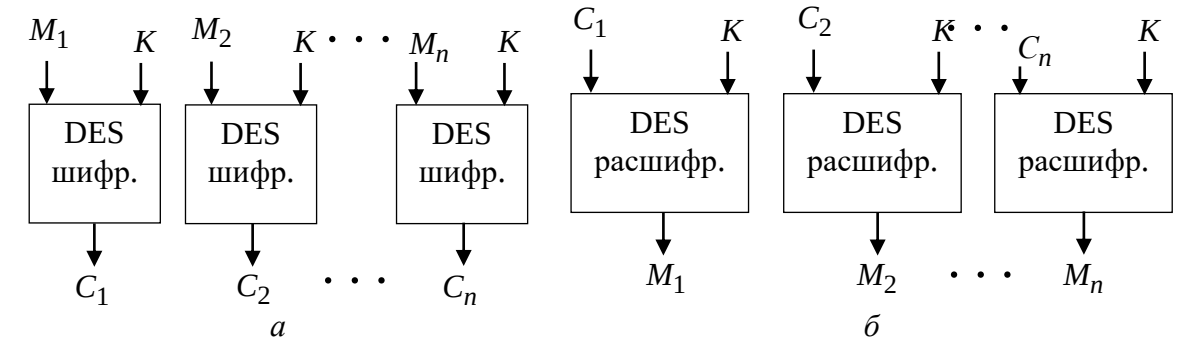


Рис. 4.5

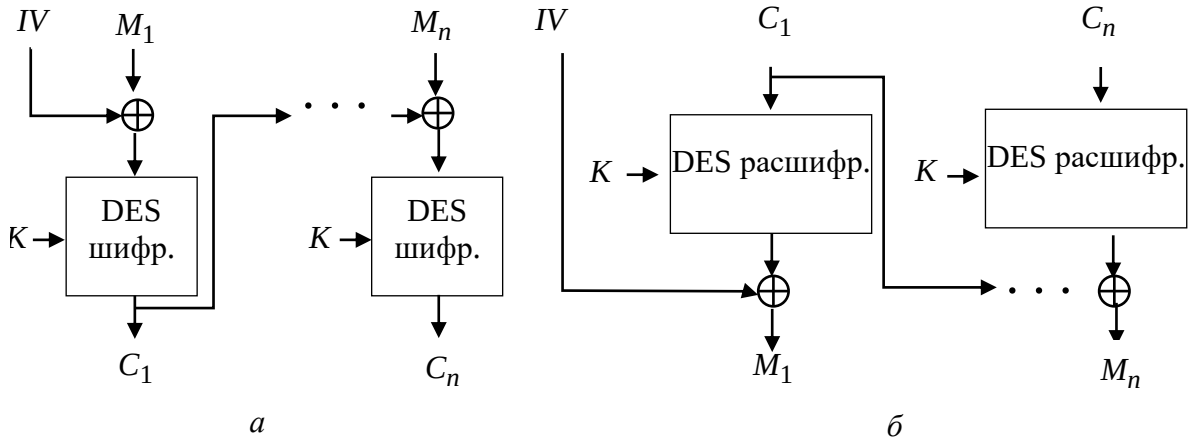


Рис. 4.6

В режиме CBC перед запуском DES для зашифрования каждого очередного блока открытого текста происходит побитовое XOR-сложение этого блока с блоком зашифрованного текста из предыдущего шага.

Схема использования шифра в режиме CBC представлена на рис. 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 шифровки в виде картинок для отчета.
5. Сжать исходную и две 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 оценить 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. Задание.

2. Схемы использования DES в режимах ECB и CBC.

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.

4.3. Исследование 3-DES

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

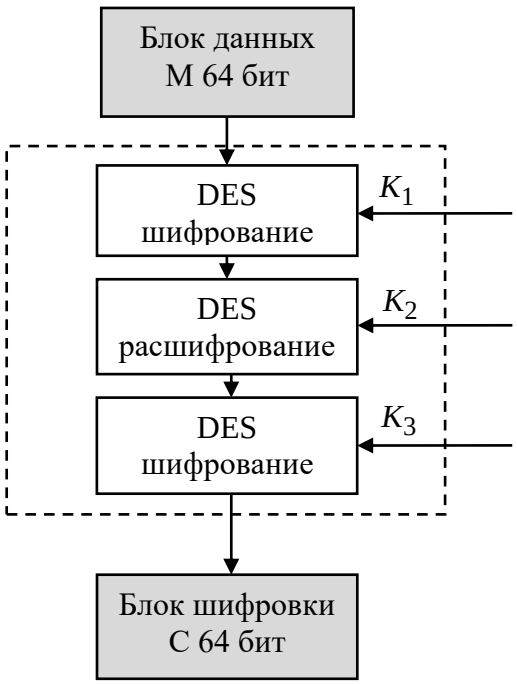


Рис. 4.7

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 используется один и тот же ключ.

На текущий момент наиболее популярны версии шифра 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 характеристики этого файла.

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

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

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

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

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

Содержание раздела отчета

1. Задание.

2. Основные параметры и обобщенная схема шифра.

3. Скриншоты частотной и автокорреляционной характеристик исходного текста и шифровки в режимах ECB и CBC.

4. Схема реализации в 3-DES в CrypTool 1 и подтверждающие скриншоты.

4.4. Исследование модификаций DESX, DESL, DESXL шифра DES

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

$$\mathrm{DESX}(M) = K_2 \oplus \mathrm{DES}_K(M \oplus K_1).$$

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

Алгоритм DESL – это сильно облегченная версия алгоритма DES. Данный алгоритм был разработан в 2006 г. для RFID-меток. Алгоритм предполагает отказ от входной и выходной перестановок блока текста, так как они не несут криптографической сложности, а также восемь S-блоков заменяются на один, но более стойкий, чем все восемь стандартных блоков DES.

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

Задание

1. Изучить алгоритмы зашифрования и расшифрования DESX, DESL, DESXL.

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

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

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

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

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

Содержание раздела отчета

1. Задание.
2. Основные параметры и обобщенные схемы шифров.
3. Таблица зависимости энтропии шифротекста от используемого шифра.
4. Таблица зависимости времени подбора ключа от используемого шифра.

4.5. Содержание отчета по лабораторной работе

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

Лабораторная работа 5.
ИЗУЧЕНИЕ ШИФРА AES

Цель работы: исследовать характеристики шифра Rijndael и других финалистов конкурса AES, а также изучить атаку предсказанием дополнения на симметричные блочные шифры в режиме использования CBC. Получить практические навыки работы с шифрами и алгоритмом проведения атаки, в том числе с использованием приложения CrypTool 1 и 2.

5.1. Исследование преобразований AES

Шифр AES (в прежнем Rijndael) [1] работает на основе перестановочно-подстановочной сети (SP-сеть). Обобщенная схема работы алгоритма представлена на рис. 5.1. Шифр принимает на вход блок данных 128 бит и ключ с вариантами длиной 128, 192 и 256 бит, выполняя раундовое преобразование 10, 12 и 14 раз соответственно.

В версии с наименьшей длиной ключа алгоритм AES получает на вход блок открытого текста размером 128 бит (16 байт) и ключ того же размера. Значения блока записываются в столбцы матрицы состояний размером 4 × 4 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 рис. 5.2:

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

2. Полученная матрица состояний проходит через преобразование подстановки SubBytes.

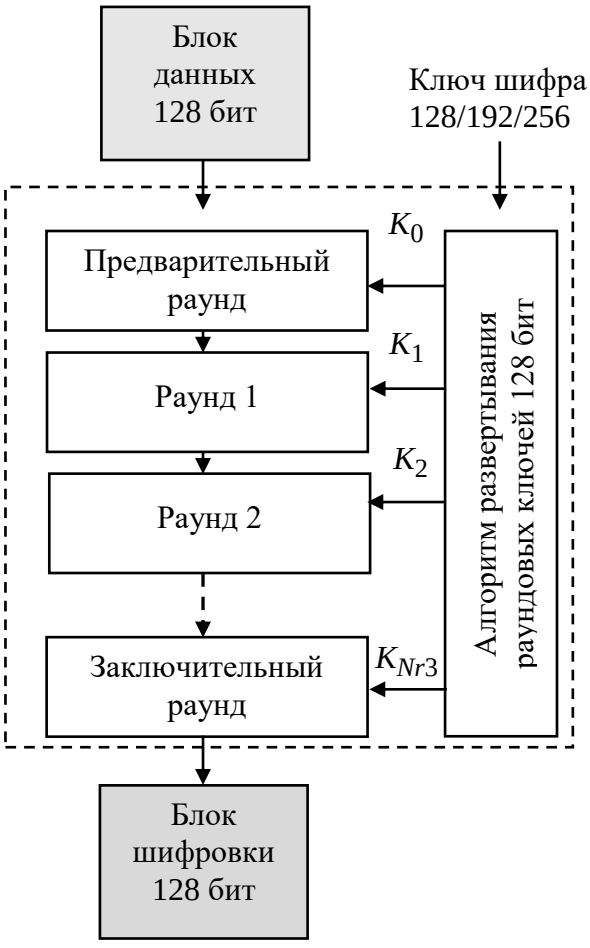


Рис. 5.1

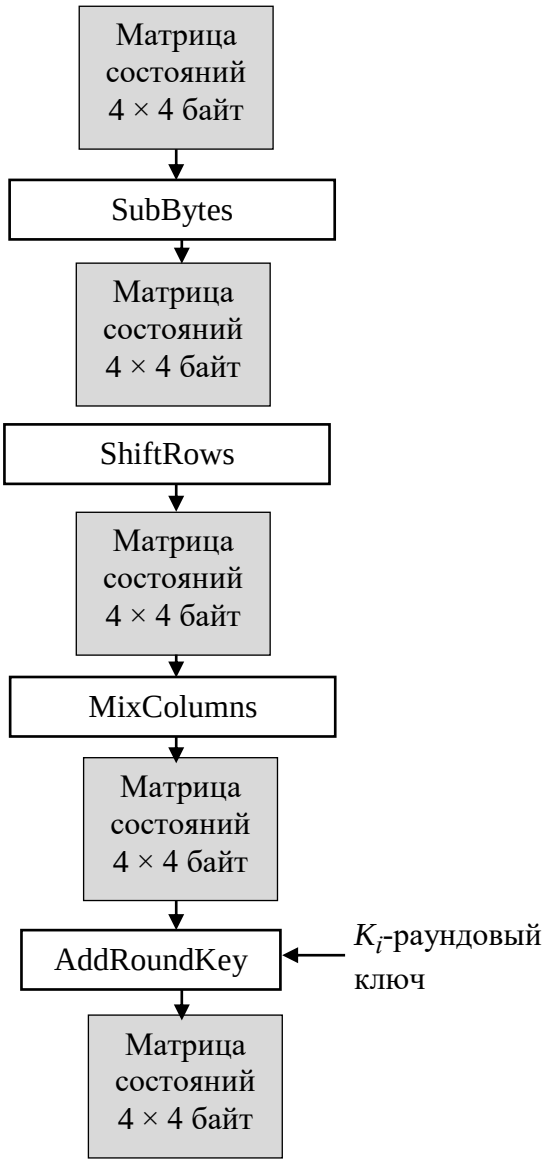


Рис. 5.2

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

4. Смешивание столбцов матрицы состояний путем ее умножением на матрицу констант в конечном поле $GF(2^8)$ выполняется преобразованием MixColumn.

5. Сложение полученных столбцов матрицы состояний с раундовым ключом операцией xor – преобразование AddRoundKey.

6. Действия 2–5 повторяются в каждом раунде, за исключением последнего.

7. Последний раунд не включает в себя смешивание столбцов.

Расшифровывание выполняется применением обратных операций и подстановкой раундовых ключей 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:

а) открытый текст – свои фамилия_имя (транслитерация латиницей);

б) 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. Задание.

2. Основные характеристики и описание AES с примерами скриншотов из демо-приложения.

3. Расчет матрицы состояний и раундового ключа шифра для одного раунда. Сравнение с результатами демо-приложения.

4. Скриншоты наблюдений потоковой модели шифра и сопутствующие выводы.

5.2. Исследование финалистов конкурса AES (Rijndael, MARS, RC6, Serpent, Twofish)

Победителем конкурса AES стал алгоритм Rijndael (ставший AES), так как по совокупности всех характеристик этот алгоритм оптимален, но допускает распараллеливание раундовых преобразований. Более подробную информацию о шифрах, участвовавших в конкурсе, можно найти в справочнике [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 оценить 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. Задание.

2. Исходные данные для экспериментов:

   а) исходный текст;

   б) секретный ключ.

3. Таблица с результатами качества зашифрования исследованными шифрами.

4. Таблица с результатами трудоемкости атаки «грубой силы» для исследованных шифров.

5.3. 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 со 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 и задействуя 1 ядро процессора. Зафиксировать затраты времени.

7. Выполнить атаку повторно со средним и максимальным количествами процессорных ядер. Зафиксировать затраты времени.

Содержание раздела отчета

1. Задание.

2. Исходные данные для экспериментов:

   а) исходный текст;

   б) 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.

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

Цель атаки – дешифровка блоков сообщения без знания ключа. При проведении этой атаки предполагается, что нарушитель может модифицировать и отправлять блоки зашифрованного сообщения серверу для расшифровки, а также распознавать ответы сервера о корректности дополнения последнего блока. Расшифровка сообщения нарушителем начинается с последнего блока 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 рис. 5.3, где P_i – блок открытого текста; C_i – блок шифротекста; I_i – промежуточное состояние; K – ключ; D_K – функция расшифровки; T_i – формируемое дополнение.

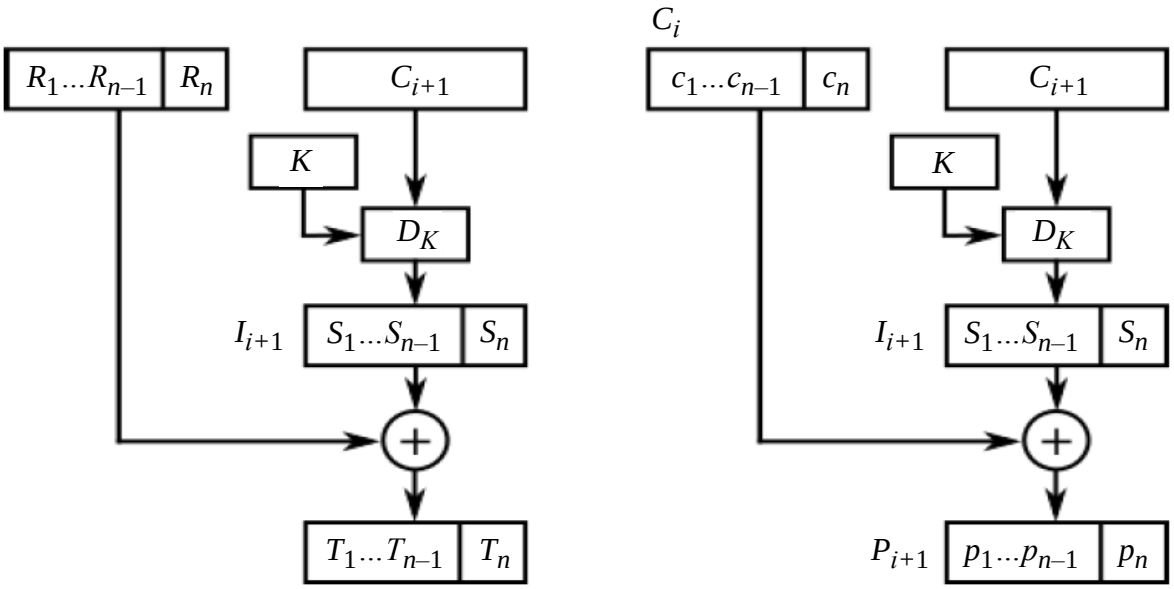


Рис. 5.3

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 блок p_{i+1} полностью.

Более подробное описание Padding Oracle Attack можно найти в [2].

В CryoTool 2 атака предсказанием дополнения реализована в три фазы:

1. Нахождение длины дополнения.
2. Подбор дополнения.
3. Расшифровка текста.

Задание

1. Найти и запустить шаблон атаки в CrypTool 2: Padding Oracle Attack on AES.
2. Подготовиться к атаке теоретически, т. е. изучить:
   а) комментарии к шаблону;
   б) действия атакующего злоумышленника [2].
3. Внедрить во второй блок исходного текста коды символов своего имени.
4. Выполнить 3 фазы атаки и сохранить итоговые скриншоты по окончании каждой фазы.
5. Убедиться, что атака удалась.

Содержание раздела отчета

1. Задание.
2. Исходные данные для экспериментов:
   а) исходный текст;
   б) секретный ключ.
3. Шаблон атаки «Padding Oracle Attack» из CrypTool 2.
4. Описание атаки «Padding Oracle Attack».
5. Результаты 3 фаз атак в виде итоговых скриншотов ПО.

5.5. Содержание отчета по лабораторной работе

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

Лабораторная работа 6.
ИЗУЧЕНИЕ ХЕШ-ФУНКЦИЙ

Цель работы: исследование хеш-функций MD5, SHA-256, SHA-512, SHA-3, кода аутентификации HMAC для контроля целостности и анализ атак дополнительной коллизии на хеш-функцию. Получить практические навыки работы с хеш-функциями и алгоритмом атаки дополнительной коллизии, в том числе с использованием приложения CrypTool 1 и 2.

6.1. Исследование лавинного эффекта MD5, SHA-1, SHA-256, SHA-512

Хеш-функцией (hash function) называется математическая или иная функция $H()$, которая для строки бит M произвольной длины вычисляет некоторое целое значение или некоторую другую строку бит $h = H(M)$, но фиксированной длины.

Хеш-значение может также называться хеш-кодом (hash code), дайджестом (digest) или отпечатком (fingerprint) сообщения.

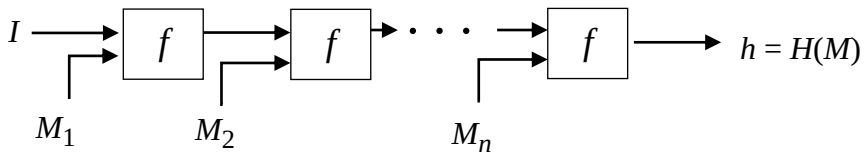
Криптографические хеш-функции имеют дополнительные свойства однонаправленности, позволяющие отличать их от обычных функций, которые также вычисляют значение фиксированной длины по входным данным произвольной длины:

1. Зная M, легко вычислить $h(M)$.

2. Зная значение h, вычислительно трудно определить M, для которого $H(M) = h$ (устойчивость к 1-му прообразу).

3. Зная сообщение M, вычислительно трудно определить другое сообщение, M', для которого $H(M) = H(M')$ (устойчивость ко 2-му прообразу).

4. Вычислительно трудно вообще найти любую пару сообщений M и M', для которых $H(M) = H(M')$ (устойчивость к коллизиям).



M_i – блок данных размером m

Рис. 6.1

Алгоритм вычисления значений хеш-функций MD5, SHA-1, SHA-256, SHA-512 основан на схеме Меркла–Дамгарда (рис. 6.1) [3].

Доказано, что если используемая в схеме функция сжатия $f()$ устойчива к коллизиям, то и сама хеш-функция также устойчива к коллизиям.

Задание

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

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

3. Для каждой хеш-функции повторить следующие действия:

а) изменить (добавлением, заменой, удалением символа) исходный файл;

б) зафиксировать количество измененных битов в дайджесте модифицированного сообщения;

в) вернуть сообщение в исходное состояние.

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

Содержание раздела отчета

1. Задание.

2. Основные параметры и обобщенная схема хеш-функций MD5, SHA-1.

3. Таблица с фактическими и усредненными параметрами лавинного эффекта для исследованных хеш-функций.

6.2. Хеш-функция SHA-3

В отличие от вышерассмотренных хеш-функций в основе SHA-3 (в прежнем Keccak) лежит конструкция под названием Sponge – губка. Сам алгоритм состоит из 2 этапов:

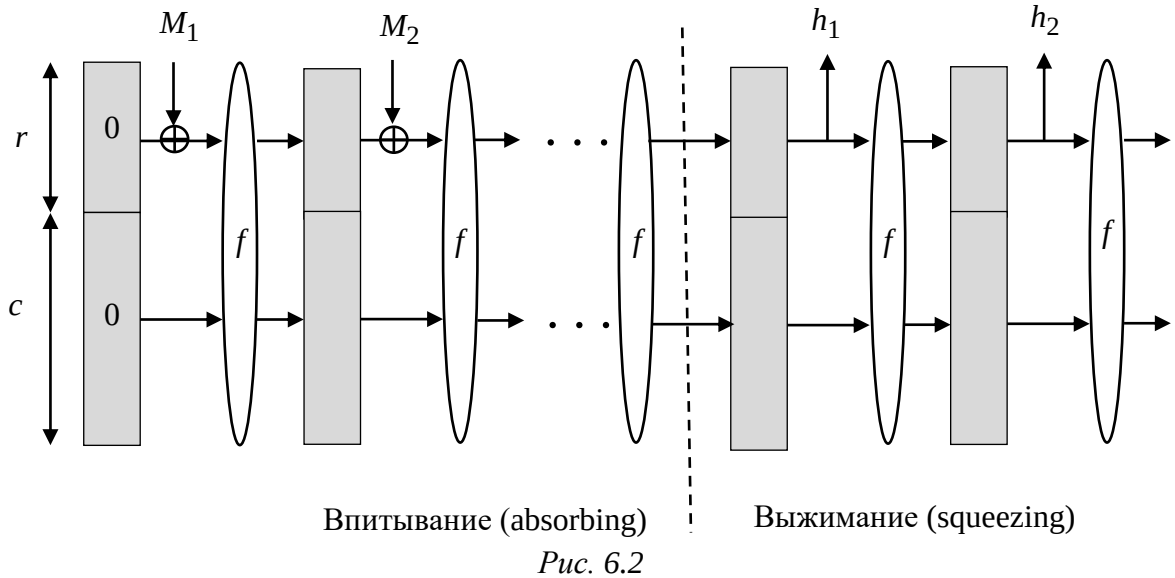


Рис. 6.2

1. *Впитывание* (*Absorbing*). 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. *Выжимание* (*Squeezing*). Чтобы получить хеш, 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).

Обобщенная схема работы алгоритма представлена на рис. 6.2.

Задание

1. Открыть шаблон *Keccak Hash (SHA-3)* в CrypTool 2.
2. В модуле Keccak сделать следующие настройки:
   а) Adjust manually=ON;
   б) Keccak version= SHA3-512.
3. Загрузить файл из предыдущего задания
4. Запустить проигрывание шаблона в режиме ручного управления:
   а) сохранить скриншоты преобразований первого раунда;
   б) сохранить скриншот заключительной фазы;
   в) 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. Задание.
2. Основные параметры, обобщенная схема и перестановки хеш-функции Keccak (SHA-3), на основе изученной презентации.
3. Описание средства оценивания лавинного эффекта.
4. Таблица с фактическими параметрами лавинного эффекта.

6.3. Контроль целостности по коду HMAC

Код аутентификации HMAC – один из механизмов проверки целостности информации, позволяющий гарантировать то, что данные, передаваемые или хранящиеся в ненадежной среде, не были изменены посторонними лицами [3]. Два абонента, использующие HMAC, разделяют общий секретный

ключ K и используют одинаковую хеш-функцию $H()$. Алгоритм HMAC можно записать в виде следующей формулы:

$$\text{HMAC}_K(\text{text}) = H\left\{(K \oplus \text{opad}) \| H\left[(K \oplus \text{ipad}) \| \text{text}\right]\right\},$$

где $\oplus$ – операция xor; $\|$ – конкатенация; K – секретный ключ; ipad – блок вида (0x36 0x36 0x36 ... 0x36), где байт 0x36 повторяется b раз; H – хеш-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 ваши 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 и модифицированный тексты коллеге, не раскрывая, какой текст корректен. Попросить коллегу определить это самостоятельно.

Содержание раздела отчета

1. Задание.

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.

6.4. Атака дополнительной коллизии на хеш-функцию

Модель атаки коллизии на хеш-функцию основана на одном из парадоксов дней рождений. Оказывается, что в группе, состоящей всего из 23 = $= c\sqrt{365}$ или более человек, вероятность совпадения дней рождения (по числу и месяцу) хотя бы у двоих из группы превышает 50 %. Применительно к хеш-функции это означает, что сложность атаки, цель которой состоит в

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

Задание

1. Сформировать два текста на английском языке – истинный и фальсифицированный. Сохранить тексты в файлах формата .txt.

2. Утилитой Analysis –> Attack on the hash value... модифицировать сообщения 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. Задание.

2. Описание атаки в терминах парадокса «дней рождения».

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

4. Таблица с оценками временной сложности атаки.

6.5. Содержание отчета по лабораторной работе

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

Лабораторная работа 7.
ИЗУЧЕНИЕ АСИММЕТРИЧНЫХ ПРОТОКОЛОВ И ШИФРОВ

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.

7.1. Протокол Диффи–Хеллмана

Протокол Диффи–Хеллмана (DH) – первое из опубликованных криптопреобразований на основе открытых ключей [4]. Поэтому этот протокол еще называют обменом ключами по схеме Диффи–Хеллмана.

Цель протокола – обеспечить двум пользователям возможность получения симметричного секретного ключа путем обмена данными по незащищенному каналу связи.

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

1. Устанавливаются открытые параметры p, g:

   а) p – большое простое число порядка 300 десятичных цифр (1024 бит);

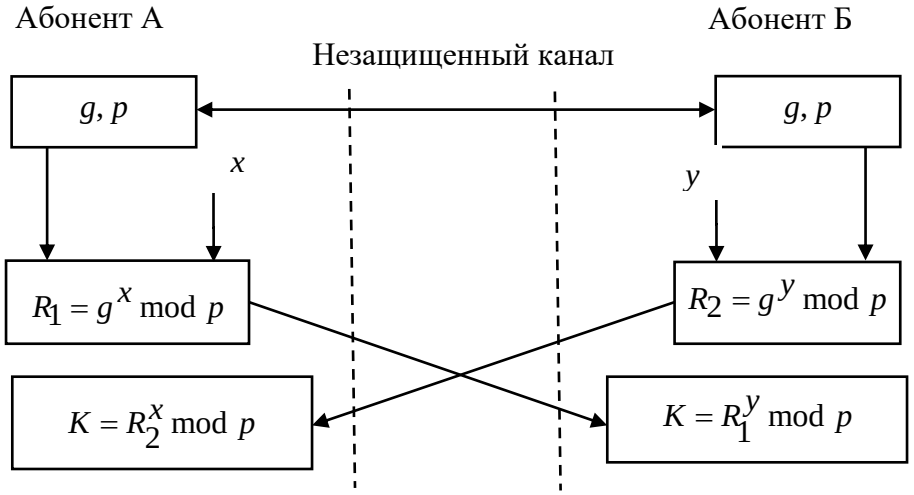
   б) g – первообразный корень по модулю p.

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

3. На каждой стороне вычисляется открытый ключ:

а) $R_1 = g^x \bmod p$,

б) $R_2 = g^y \bmod p$.



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

Рис. 7.1

Стороны обмениваются открытыми ключами и вычисляют общие данные K для создания симметричного ключа:

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

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.

Содержание раздела отчета

1. Задание.

2. Основные параметры и схема протокола.

3. Скриншот демонстрации работы протокола, реализованной в CrypTool.

4. Таблица соответствия демонстрации протокола (CrypTool) и параметров протокола.

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

7.2. Шифр RSA

Алгоритм RSA [4] 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 (рис. 7.2):

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: $e \times d = 1 \bmod \varphi(n)$;

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

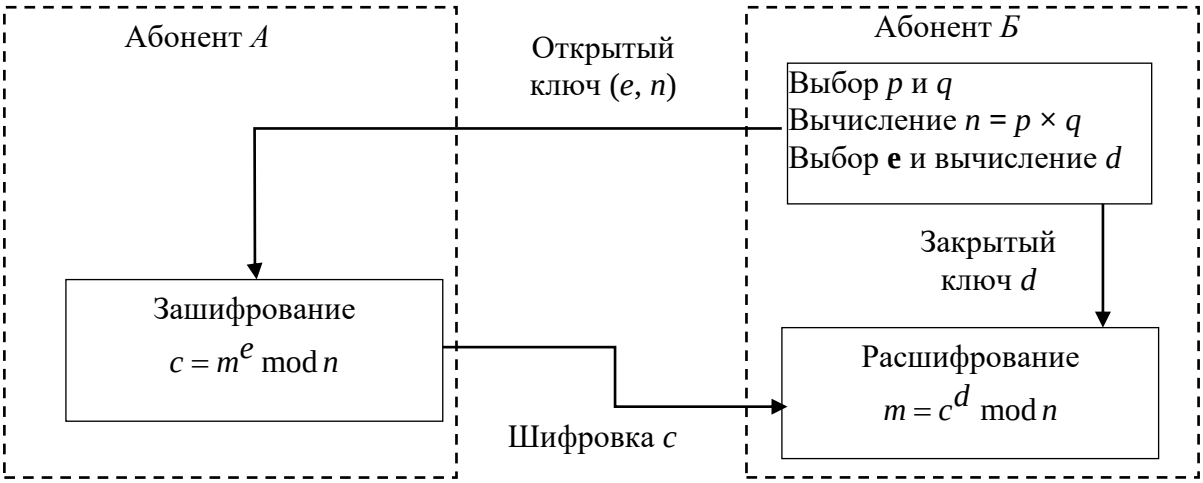


Рис. 7.2

2. Зашифрование:

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

б) каждый блок открытого текста преобразуется в шифротекст по формуле

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

3. Расшифрование:

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

б) каждый блок шифротекста преобразуется в открытый текст по формуле

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

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.

Содержание раздела отчета

1. Задание.

2. Обобщенная схема протокола шифрования RSA.

3. Скриншот результата генерации ключей.

4. Скриншот результата зашифровки.

5. Скриншот результата расшифровки.

7.3. Исследование шифра RSA

Задание

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

2. Сгенерировать пары асимметричных RSA-ключей утилитой Digital Signatures –> PKI –> Generate/Import Keys с различными длинами (4 варианта).

3. Зашифровать текст (примерно 1000 символов) различными открытыми ключами. Зафиксировать время зашифровки.

4. Расшифровать текст различными закрытыми ключами. Зафиксировать время зашифровки.

5. Проверить корректность расшифровки. Зафиксировать скриншоты результата.

Содержание раздела отчета

1. Задание.

2. Выбранный текст.

3. Результаты генерации ключевых пар различной длины.

4. Размер исходного текста.

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

7.4. Атака «грубой силы» на RSA

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.

Содержание раздела отчета

1. Задание.
2. Исходные данные для атаки, полученные от коллеги.
3. Результат факторизации (скриншот).
4. Расшифрованное в итоге сообщение (скриншот).

7.5. Имитация атаки на гибридную криптосистему

Модель гибридной криптосистемы, асимметричная составляющая которой использует асимметричный шифр (например, RSA), представлена на рис. 7.3.

Шифрование в рамках модели осуществляется следующим образом:

1. Сообщение шифруется симметричным секретным ключом.
2. Секретный ключ шифруется открытым ключом получателя.
3. Зашифрованное сообщение и ключ объединяются в цифровой конверт, который отправляется получателю.
4. Получатель сначала расшифровывает секретный ключ своим закрытым ключом, а затем расшифровывает этим секретным ключом шифровку сообщения.

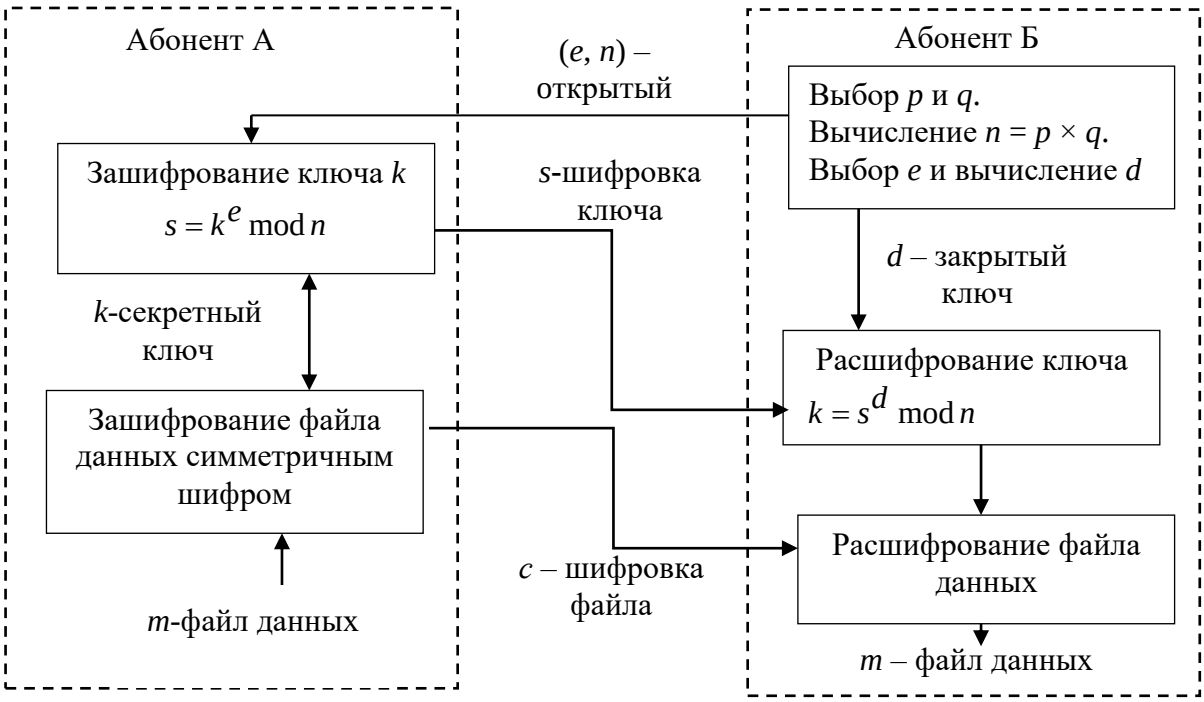


Рис. 7.3

Атака на модель гибридной криптосистемы основана на том, что злоумышленник сначала перехватывает цифровой конверт, содержащий зашифрованные сообщение и секретный ключ, затем специальным 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.

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.

Содержание раздела отчета

1. Задание.

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

3. Алгоритм действий злоумышленника.

4. Текст передаваемого сообщения.

5. Лог-файлы участников протокола.

7.6. Содержание отчета по лабораторной работе

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

Лабораторная работа 8.
ИЗУЧЕНИЕ ЭЛЕКТРОННОЙ ПОДПИСИ

Цель работы: исследовать алгоритмы создания и проверки электронной подписи, алгоритмы генерации ключевых пар для алгоритмов электроной подписи RSA, DSA, ECDSA и получить практические навыки работы с ними, в том числе с использованием приложения CrypTool версий 1 и 2.

8.1. Генерация ключевых пар

Генерация ключевых пар для алгоритма RSA (двух больших простых чисел p и q (держатся в секрете)):

1. Вычисление $n = p \times q$.

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

3. Вычисление $d : e \times d = 1 \bmod \varphi(n)$.

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

Генерация ключевых пар для алгоритма DSA:

1. Выбирается число p: длина – [512, 1024] бит, число бит в p должно быть кратно 64.

2. Выбирается число q, которое имеет тот же размер в битах, что и размер дайджеста используемой хеш-функции (160 бит для SHA-1) и удовлетворяющее условию $(p-1) = 0 \bmod q$.

3. Выбирается $e_1 : e_1^q = 1 \bmod p$.

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

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

Генерация ключевых пар для алгоритма ECDSA:

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

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

3. Выбирается простое число q – порядок одной из циклических подгрупп группы точек эллиптической кривой: $q \times (x_1, y_1) = 0$.

4. Выбирается закрытый ключ d.

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. Зафиксировать 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. Задание.
2. Описание алгоритмов генерации ключевых пар.
3. Таблица с фактическими временем генерации ключевых пар.
4. Скриншоты со значениями открытых ключей.

8.2. Процессы создания и проверки электронной подписи

Электронная подпись – это некоторая информация в электронной форме (код), которая присоединена к другой информации (файлу данных) с целью подтверждения авторства и контроля целостности файла данных. Обобщенная схема, поясняющая работу протокола подписания документа и проверки электронной подписи, представлена на рис. 8.1.

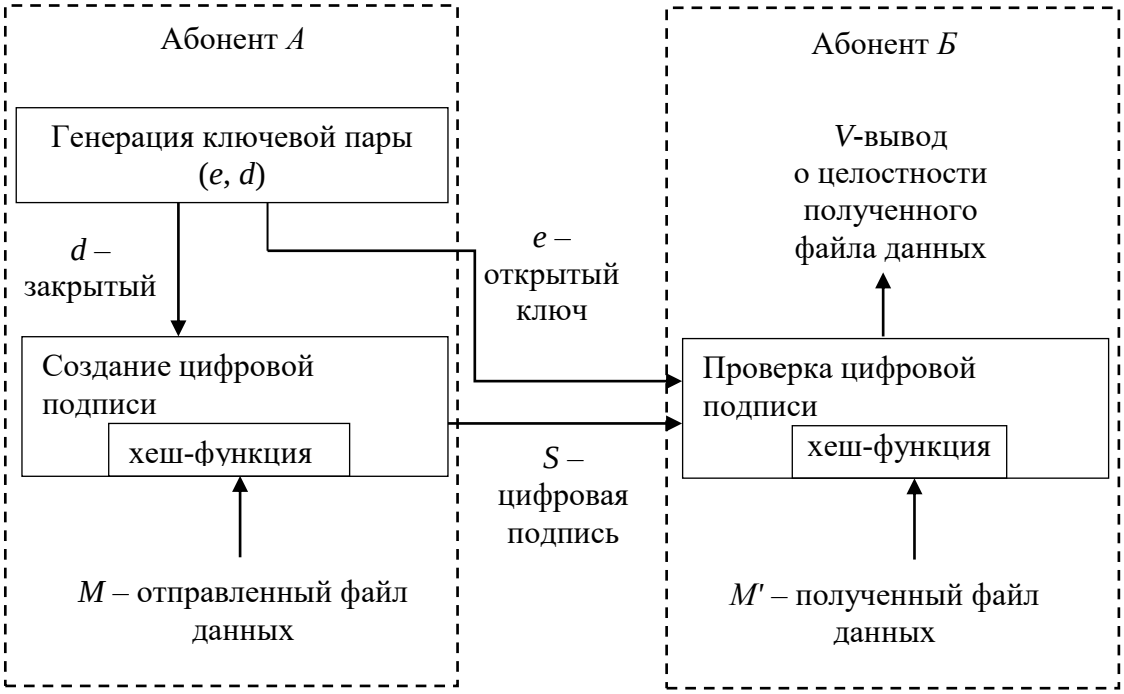


Рис. 8.1

Следует отметить, что самом общем случае на стороне отправителя запускается процедура создания электронной подписи (процедура подписания), а на стороне получателя – процедура проверки электронной подписи (процедура верификация). Подпись создается на основе хеш-кода отправляемого файла данных и закрытого ключа отправителя, а при проверке подписи используется хеш-код от полученного файла данных, сопровождающая его подпись и открытый ключ отправителя, доставленный получателю.

Задание

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

2. Задать хеш-функцию и другие параметры электронной подписи.

3. Создать подписи, используя закрытые ключи, сгенерированные в предыдущем задании. Зафиксировать время создания электронной подписи для каждого ключа (опция Display signature time должна быть включена)

4. Сохранить скриншот любой электронной подписи с помощью приложения Digital Signatures/PKI –> Extract Signature.

5. Выполнить процедуру проверки любой подписи Digital Signatures/ PKI –> Verify Signature для случаев сохранения и нарушения целостности исходного текста. Сохранить скриншоты результатов.

Содержание раздела отчета

1. Задание.

2. Обобщенная схема создания и проверки электронной подписи.

3. Таблица с фактическими временами создания электронной подписи различными алгоритмами.

4. Скриншот со значением электронной подписи.

5. Скриншоты с результатами проверки электронной подписи.

8.3. Создание и проверка электронной подписи на основе эллиптических кривых

С использованием ключевой пары (закрытый ключ – d, и открытый ключ – (a, b, q, p, e_1, e_2), сгенерированной в 8.1 для алгоритма ECDSA [4], осуществляется подписание документа, а затем на принимающей стороне – верификация подписи (рис. 8.2).

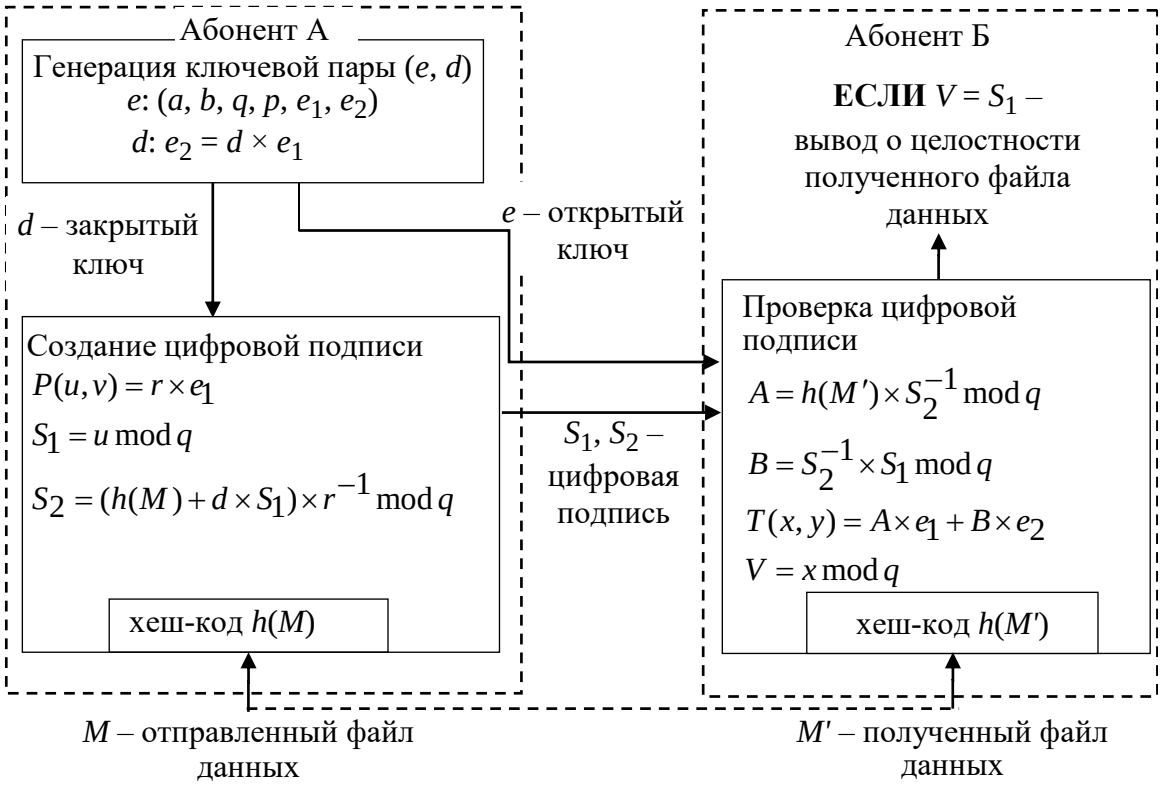


Рис. 8.2

Алгоритм создания электронной подписи 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

$$S_1 = u \bmod q,$$

где u – абсцисса.

4. Вычисляется вторая часть подписи по формуле

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

где $h(M)$ – хеш-код сообщения; d – закрытый ключ.

Алгоритм проверки электронной подписи ECDSA включает следующие операции:

1. Вычисляются промежуточные результаты A и B:

$$A = h(M') \times S_2^{-1} \bmod q;$$
$$B = S_2^{-1} \times S_1 \bmod q.$$

2. Восстанавливается третья точка:

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

Верификатор $V = x \bmod q$ сравнивается с первой частью электронной подписи S_1.

Задание

1. Выполнить процедуру создания подписи Digital Signatures/PKI –> Sign Document… алгоритмом ECSP-DSA в пошаговом режиме (*Display inter. results = ON*). Зафиксировать скриншоты последовательности шагов.

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

3. Проверить лекционный материал по ECDSA, создав и проверив подпись сообщения M (принять $M = h(M)$) приложением Indiv.Procedures –> Number Theory… –> Point Addition on EC.

Содержание раздела отчета

1. Задание.

2. Описание алгоритма формирования и проверки подписи ECDSA.

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

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

8.4. Демонстрация процесса подписи в среде PKI

Инфраструктура открытых ключей (ИОК, PKI – Public Key Infrastructure) – набор средств (технических, материальных, организационных и т. д.), распределенных служб и компонентов, в совокупности используемых для поддержки решения основных задач криптографии, т. е. обеспечения:

1) конфиденциальности информации;

2) целостности информации;

3) аутентификации пользователей и ресурсов, к которым обращаются пользователи;

4) возможности подтверждения совершенных пользователями действий.

Решение перечисленных задач основано на использовании сертификатов открытых ключей. Сертификат открытого ключа – это электронный документ, который содержит:

1) открытый ключ пользователя;

2) информацию о пользователе, которому принадлежит сертификат;

3) информацию о сроке действия сертификата;

4) информацию об издателе сертификата;

5) другие атрибуты;

6) электронную подпись этих данных, созданную удостоверяющим центром, издавшим этот сертификат.

Существует несколько вариантов использования сертификатов открытых ключей:

1) для зашифрования и расшифрования электронных документов;

2) для подписания электронного документа и проверки подписи;

3) для аутентификации отправителя документа.

Задание

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

2. Получить сертификат ключа проверки электронной подписи (открытого ключа) на ранее сгенерированную ключевую пару RSA-2048.

3. Выполнить и сохранить скриншоты всех этапов создания электронной подписи документа.

4. Сохранить скриншот полученного сертификата ключа проверки этой электронной подписи.

Содержание раздела отчета

1. Задание.

2. Описание структуры сертификата из CrypTool 1.

3. Схема процедуры подписания из CrypTool 1.

8.5. Подписание своего отчета

Задание

1. Сконвертировать отчет в формат pdf.

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

3. Открыть pdf-версию отчета и попытаться подписать с использованием этого сертификата.

4. Создать собственный самоподписанный сертификат в среде Adobe Reader и использовать его для подписи отчета.

5. Сохранить скриншоты свойств подписи и сертификата.

6. Внести изменения (маркеры, комментарии) в отчет и проверить подпись.

Содержание раздела отчета

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

8.6. Содержание отчета по лабораторной работе

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

Список литературы

1. Панасенко С. П. Алгоритмы шифрования: спец. справ. М.: BHV, 2009. 564 с.

2. Padding Oracle Attack или почему криптография пугает. URL: https://habrahabr.ru/post/247527.

3. Шнайер Б. Прикладная криптография. Протоколы, алгоритмы, исходные тексты на языке Си: пер. с англ. М.: Триумф, 2003. 815 с.

4. Коржик В. И., Яковлев В. А. Основы криптографии: учеб. пособие. СПб.: Интермедия, 2017. 294 с.

Содержание

Племянников Александр Кимович

Криптографические методы защиты информации

Учебно-методическое пособие

Редактор Н. В. Лукина
Компьютерная верстка И. С. Беляевой

Подписано в печать 05.12.22. Формат 60×84 1/16.
Бумага офсетная. Печать цифровая. Печ. л. 3,25.
Гарнитура «Times New Roman». Тираж 53 экз. Заказ .

Издательство СПбГЭТУ «ЛЭТИ»
197022, С.-Петербург, ул. Проф. Попова, 5Ф