МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное
учреждение высшего образования
«СЕВАСТОПОЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

ИНСТИТУТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И УПРАВЛЕНИЯ
В ТЕХНИЧЕСКИХ СИСТЕМАХ

Кафедра «Информационные технологии и компьютерные системы»

КРИПТОГРАФИЧЕСКИЕ МЕТОДЫ ЗАЩИТЫ ИНФОРМАЦИИ

Методические указания
к выполнению лабораторных работ
по дисциплине «Зашита информации»
для студентов направления подготовки
09.03.01 «Информатика и вычислительная техника»

Севастополь
СевГУ
2019

УДК 004.056.55
К82

Р е ц е н з е н т:
К.В. Кротов – доцент кафедры «Информационные системы»,
канд. техн. наук, доцент

Составители: Н.Л. Корепанова, М.А. Лебедева

К82 Криптографические методы защиты информации: метод. указания к выполнению лабораторных работ по дисциплине «Защита информации» для студентов направления подготовки 09.03.01 «Информатика и вычислительная техника» / Сост. Н.Л. Корепанова, М.А. Лебедева. – Электрон. дан. – Севастополь: СевГУ, 2019 г. – Режим доступа: свободный после авторизации. – Загл. с экрана. – 31 с.

Целью методических указаний является оказание методической помощи бакалаврам при выполнении лабораторных работ по дисциплине «Защита информации».

Методические указания содержат:
- краткие теоретические сведения;
- постановку задачи для выполнения работы;
- содержание отчета;
- список литературы;
- индивидуальные задания для выполнения работы.

УДК 004.056.55

Методические указания рассмотрены и утверждены на заседании кафедры «Информационные технологии и компьютерные системы», протокол № 6 от 05 апреля 2019 г.

Текстовое (символьное) издание (1 Мб).
Системные требования: Intel, 3,4 GHz; 150 Мб; Windows XP/Vista/7; DVD-ROM; 1 Гб свободного места на жестком диске; программа для чтения pdf-файлов: Adobe Acrobat Reader, Foxit Reader.

СОДЕРЖАНИЕ

Введение

Развитие современных информационных технологий и многочисленные угрозы информации при ее хранении, обработке в компьютерных системах и передаче по компьютерным сетям, привели к необходимости развития методов и средств защиты информации. Наиболее действенными являются криптографические методы, которые широко используются для защиты информации от несанкционированного доступа и изменения. Криптография дает возможность обеспечить защиту информации путем изменения формы ее представления.

Цикл лабораторных работ по дисциплине «Защита информации» предназначен для изучения основных разделов криптографии: симметричного и асимметричного шифрования, алгоритмов электронной цифровой подписи, криптографических протоколов.

Лабораторная работа № 1. Шифрование данных методами подстановки, перестановки и полиалфавитными шифрами

Цель работы: Приобретение навыков шифрования информации с использованием простейших методов шифрования.

Криптографические методы защиты информации

Проблемой защиты информации путем ее преобразования занимается криптология (kryptos - тайный, logos - наука). Криптология разделяется на два направления - криптографию и криптоанализ. Цели этих направлений прямо противоположны:

– криптография занимается поиском и исследованием математических методов преобразования информации.

– сфера интересов криптоанализа - исследование возможности расшифровывания информации без знания ключей.

Криптография дает возможность преобразовать информацию таким образом, что ее прочтение (восстановление) возможно только при знании ключа. В качестве информации, подлежащей шифрованию и дешифрованию, рассматриваются тексты, построенные на некотором алфавите. Алфавит - конечное множество используемых для кодирования информации знаков. Примеры алфавитов, используемых в современных информационных системах:

– алфавит Z_{33} - 32 буквы русского алфавита и пробел;

– алфавит Z_{256} - символы, входящие в стандартные коды ASCII и КОИ-8;

– бинарный алфавит - $Z_2 = \{0,1\}$.

Шифрование – процесс преобразования исходного или открытого текста в зашифрованный. Выполняется на основе ключа и используется для защиты сообщений от несанкционированного прочтения. Дешифрование - обратный шифрованию процесс. На основе ключа шифрованный текст преобразуется в исходный.

Ключ - информация, необходимая для беспрепятственного шифрования и дешифрования текстов. Обычно ключ представляет собой последовательный ряд символов того же алфавита, в котором набрано информационное сообщение

По характеру используемого ключа криптографические методы делятся на:

- симметричные: для шифрования и дешифрования используется один и тот же секретный ключ;

- асимметричные: для шифрования и дешифрования используют разные ключи, открытый – для шифрования, секретный – для дешифрования.

К симметричным криптографическим алгоритмам относят простейшие методы шифрования (подстановки, перестановки), потоковые и блочные шифры.

Метод подстановки

Шифр подстановки или замены - наиболее простой вид преобразований, заключающийся в замене символов исходного текста на другие символы того же либо другого алфавита по определенному правилу.

Историческим примером шифра подстановки является шифр Цезаря, в котором каждый символ открытого текста заменяется другой буквой, которая определяется путем смещения по алфавиту от исходной буквы влево или вправо на *k* букв. При достижении конца алфавита выполняется циклический переход к его началу. Цезарь использовал шифр замены при смещении вправо при $k = 3$.

Для произвольного ключа *k* шифр имеет вид:

$$x_i \to y_j, \qquad i = (j + k) mod\ n, \qquad i = \overline{1,n} \tag{1.1}$$

где i – номер в алфавите символа открытого текста,

j –номер зашифрованного символа,

k –величина смещения - ключ,

n –количество букв в алфавите.

Обратная подстановка осуществляется по правилу

$$i = (j + n - k) mod\ n \tag{1.2}$$

Условием для успешной реализации этого метода является совпадение размера множеств открытого текста и шифротекста. Это условие в современных криптосистемах называется гомоморфизмом.

Другим вариантом метода подстановки является задание соответствия между буквами исходного алфавита и буквами подстановочного алфавита. Это позволяет заменять буквы в открытом тексте буквами из подстановочного алфавита Подстановочный алфавит может задаваться как множество символов, либо составляться по определенному правилу.

Пусть подстановочный алфавит составлен по следующему правилу:

$$y_{2k-1} = x_{2k}, y_{2k} = x_{33-2k} \quad k = \overline{1,16} \tag{1.3}$$

где *x*- исходный подстановочный алфавит; *y* - подстановочный алфавит;

В формуле (1.3) буквы с четными и нечетными номерами в алфавите, заменяются по разным правилам.

Воспользуемся новым алфавитом для шифрования фразы:

ОСНОВЫ ЗАЩИТЫ ИНФОРМАЦИИ

Каждая буква в этой фразе имеет порядковый номер в исходном алфавите. При шифровании методом подстановки необходимо заменить буквы исходного алфавита соответствующими буквами подстановочного алфавита (О - П, С - О, Н - Т и т.д.). Так буква О в исходном алфавите имеет номер 16, *k*=8. По правилу *x*(2·8)=*y*(33-2·8) буква О заменяется буквой с номером 17, т.е. П.

В шифрованном виде эта фраза примет следующий вид:

ПОТПГЭ ШБЖЙУЭ ЙТХПСНБЧЙЙ.

Шифрование простой подстановкой на коротких алфавитах обеспечивает слабую защиту открытого текста. Подстановочные криптограммы можно раскрыть, составляя частотные таблицы для букв, пар букв (биграмм) и троек букв (триграмм). Большие частоты появления одних букв и малые других, а также частые ассоциации гласных с согласными позволяют найти буквы открытого текста. С увеличением размера алфавита применение частотного анализа становится все более дорогим, однако, принцип подстановки теряет свою практическую значимость.

Метод перестановки

При шифровании этим методом переставляются не буквы алфавита, а буквы открытого текста в пределах группы, называемой таблицей перестановки. Например, сообщение разбито на группы знаков, включая пробелы, и в каждой группе буквы переставлены в соответствии с правилом:

⌈1 2 3 4⌉
⌊2 4 1 3⌋

В этом случае вторая буква исходного текста буде стоять на первом месте, четвертая – на втором и т.д. Если сообщение не кратно количеству символов в группе перестановки, последняя группа дополняется определенными символами, чаще всего пробелами.

Если задана фраза: ОСНОВЫ ЗАЩИТЫ ИНФОРМАЦИИ, то после шифрования она примет вид: СООНЫЗВ ЩТАИ НЫИОМФРИАИ.

В случае перестановки таблицы частот для пар и трех букв показывают наличие стандартных буквенных пар, позволяя реконструировать открытый текст путем поиска тех перестановок, которые их воссоединяют. Следовательно, ключ, используемый для преобразования открытого текста, может быть восстановлен по одной криптограмме. Используется, как правило, в сочетании с другими методами.

Многоалфавитные шифры

Слабая криптостойкость моноалфавитных подстановок преодолевается с применением подстановок многоалфавитных. Для защиты от частотного анализа были разработаны многоалфавитные шифры, в которых для шифрования сообщения периодически используется несколько различных подстановочных алфавитов. Если задано *r* подстановочных алфавитов, то исходное сообщение разбивается на группы по *r* символов, для шифрования *i*-го символа группы используется *i*-ый подстановочный алфавит. Например, для *r*=4 буквы с номерами 1,5,9,13, ... шифруются 1 алфавитом, буквы с номерами 2,7,10,14, ... - 2 алфавитом, и т.д.

Для получения открытого текста выделяются повторяющиеся группы знаков, и определяется период повторения. Предполагаемый период проверяется составлением частотного распределения для каждой n-й буквы зашифрованного текста. Если каждое из n частотных распределений имеет сильную неоднородность, характерную для моноалфавитной подстановки, то предполагаемый период является правильным. Затем задача решается как n различных простых подстановок.

Задание на лабораторную работу

1. Разработать алгоритм и составить программу, позволяющую закодировать любой текст одним из вышеизложенных методов и выполнить обратное преобразование. Метод, которым необходимо зашифровать исходную информацию, выбирается в соответствии с вариантом из таблиц 1.1, 1.2, 1.3. Язык программирования выбирается произвольно.
2. Осуществить вывод на экран или принтер полученной криптограммы.
3. Провести дешифрование данной криптограммы, в результате должен быть получен исходный текст.
4. Результаты работы оформить в виде отчета.

Таблица 1.1 - Методы шифрования

Ном вар.	Метод шифрования	Таблица	Номер задания в таблице	Представление исходного текста
1	Подстановка	2	3	Английский алфавит
2	Перестановка	3	1	ASCII-код
3	Многоалфавитные шифры	2	1, 2, 5	Русский алфавит
4	Перестановка	3	2	Русский алфавит
5	Подстановка	2	4	Английский алфавит
6	Многоалфавитные шифры	2	1, 3	Русский алфавит
7	Подстановка	2	1	Английский алфавит
8	Многоалфавитные шифры	2	2, 5	Английский алфавит
9	Перестановка	3	3	ASCII-код

Продолжение таблицы 1.1				
10	Подстановка	2	2	Русский алфавит
11	Перестановка	3	4	ASCII-код
12	Многоалфавитные шифры	2	1, 3, 4	Русский алфавит

Таблица 1.2 – Подстановочные алфавиты

Ном симв	Исходный алфавит		Подстановочный алфавит									
			1		2		3		4		5	
1	А	A	Б	V	С	C	О	Z	Ю	C	М	V
2	Б	B	Ю	W	О	D	П	ппро-ббел	Я	D	Н	W
3	В	C	Г	X	У	A	М	.	Ы	A	О	X
4	Г	D	Ы	Y	М	B	Н	X	Э	B	П	Y
5	Д	E	Е	Z	К	H	Х	Y	Ь	H	Р	Z
6	Е	F	Ь	про-бел	Х	I	Л	,	Ъ	I	С	про-бел
7	Ё	G	З	.	Ч	J	И	!	Ш	J	Т	.
8	Ж	H	Ш	,	И	E	Й	S	Щ	E	У	,
9	З	I	Й	!	Щ	F	Ж	T	Ц	F	Ф	!
10	И	J	Ц	:	Ж	G	З	:	Ч	G	Х	:
11	Й	K	Л	;	Ъ	O	Д	;	Ф	O	Ц	;
12	К	L	Ф	?	Д	P	Е	Q	Х	P	Ч	?
13	Л	M	Н	-	Э	Q	В	R	Т	Q	Ш	-
14	М	N	Т	K	В	R	Г	?	У	R	Щ	K
15	Н	O	П	L	Я	K	А	-	Р	K	Ъ	L
16	О	P	Р	M	А	L	Б	N	С	L	Ь	M
17	П	Q	С	N	Б	M	Ю	O	О	M	Ы	N
18	Р	R	О	O	Ю	N	Я	P	П	N	Э	O
19	С	S	У	P	Г	U	Ы	L	М	U	Ю	P
20	Т	T	М	Q		V	Э	M	Н	V	Я	Q
21	У	U	Х	R	Е	W	Ь	N	К	W	про-бел	R
22	Ф	V	К	S	Ь	:	про-бел	O	Л	:	А	S
23	Х	W	Ч	T	З	S	Ш	P	про-бел	S	Б	T
24	Ц	X	И	U	Ш	T	Щ	A	Й	T	В	U
25	Ч	Y	Щ	A	Й	Z	Ц	B	Ж	Z	Г	A
26	Ш	Z	Ж	B	Ц	про-бел	Ч	C	З	про-бел	Д	B
27	Щ	про-бел	Ъ	C	Ё	X	Ф	D	Д	X	Е	C
28	Ъ	.	Д	D	Ф	Y	К	E	Е	Y	Ё	D
29	Ь	,	Э	E	Н	;	Т	F	В	;	Ж	E
30	Ы	!	В	F	Т	?	У	G	Г	?	З	F
31	Э	:	Я	G	П	-	Р	H	А	-	И	G
32	Ю	;	про-бел	H	Р	.	С	I	Б	.	Й	H
33	Я	?	А	I	Ы	,	Ъ	J	Ё	,	К	I
34	про-бел	-	Ё	J	Л	!	Ё	K	И	!	Л	J

Таблица 1.3 - Группы перестановок

Номер вар.	Группа перестановки	Номер вар.	Группа перестановки
1	$\begin{bmatrix}1&2&3&4&5&6\\3&5&2&6&1&4\end{bmatrix}$	4	$\begin{bmatrix}1&2&3&4&5&6\\2&6&3&5&1&4\end{bmatrix}$
2	$\begin{bmatrix}1&2&3&4&5\\5&4&1&2&3\end{bmatrix}$	5	$\begin{bmatrix}1&2&3&4&5\\2&5&4&3&1\end{bmatrix}$
3	$\begin{bmatrix}1&2&3&4&5&6\\2&5&3&4&1&6\end{bmatrix}$	6	$\begin{bmatrix}1&2&3&4&5&6\\3&5&2&6&1&4\end{bmatrix}$

Содержание отчета:

- цель работы, постановка задачи,
- описание исходных данных,
- алгоритм работы программы,
- текст программы,
- результаты работы программы,
- анализ результатов
- выводы.

Контрольные вопросы

1. Почему метод подстановки имеет слабую надежность?
2. Что такое частотный анализ?
3. Что является криптографическим ключом в методе перестановки?
4. Как связаны метод подстановки и многоалфавитные шифры?
5. В чем отличие криптографии от криптоанализа?
6. По какому признаку шифры делят на симметричные и асимметричные?

Лабораторная работа №2. Шифр гаммирования

Цель работы: Освоение принципов шифрования гаммированием, изучение свойств генератора псевдослучайных чисел, программная реализация метода гаммирования.

Теоретические основы метода гаммирования

Принцип шифрования гаммированием заключается в генерации гаммы шифра с помощью датчика псевдослучайных чисел и наложении полученной гаммы шифра на открытые данные обратимым образом (используя операцию сложения по модулю 2).

$$y_i = x_i \oplus g_i \tag{2.1}$$

где x_i - бит исходного текста;

y_i - бит зашифрованного текста;

g_i - бит гаммы.

Процесс дешифрования сводится к повторной генерации гаммы шифра при известном ключе и наложении такой же гаммы на зашифрованные данные.

Гамма шифра генерируется независимо от исходного текста.

Полученный зашифрованный текст является достаточно трудным для раскрытия в том случае, если гамма шифра не содержит повторяющихся битовых последовательностей и изменяется случайным образом для каждого шифруемого слова. Если период гаммы превышает длину всего зашифрованного текста и неизвестна никакая часть исходного текста, то шифр можно раскрыть только прямым перебором (подбором ключа). В этом случае криптостойкость определяется размером ключа.

Метод гаммирования становится бессильным, если известен фрагмент исходного текста и соответствующая ему шифрограмма. В этом случае простым сложением по модулю 2 получается отрезок псевдослучайной последовательности и по нему восстанавливается вся эта последовательность.

Линейные конгруэнтные датчики ПСЧ

Чтобы получить линейные последовательности элементов гаммы, длина которых не превышает размер шифруемых данных, используют датчики ПСЧ. Одним из хороших конгруэнтных генераторов является линейный конгруэнтный датчик ПСЧ. Он вырабатывает последовательности псевдослучайных чисел $T(i)$, описываемые соотношением

$$T_i = (A \cdot T_{i-1} + C) mod\ M, \quad (2.2)$$

где A, C, M - константы, T_0 - исходная величина, выбранная в качестве порождающего числа. Очевидно, что эти три величины и образуют ключ.

Такой датчик ПСЧ генерирует псевдослучайные числа с определенным периодом повторения, зависящим от выбранных значений A и C. Значение M обычно устанавливается равным 2^b, где b -длина машинного слова в битах. Необходимо выбирать числа A и C так, чтобы период M был максимальным.

Как показано Д.Кнуттом, линейный конгруэнтный датчик имеет максимальную длину M тогда, когда C нечетное и A mod 4 = 1.

В качестве примера использования линейного конгруэнтного датчика ПСЧ рассмотрим процесс шифрования исходного текста «абв». Пусть b = 5, т.е. для представления буквы исходного текста используется 5 двоичных разрядов. В соответствии с номером в алфавите буква «а» имеет двоичный код 00001; буква «б» имеет двоичный код 00010; буква «в» имеет двоичный код 00011. Исходный текст будет представлен в виде последовательности 00001 00010 00011.

Для формирования гаммы шифра выберем параметры датчика ПСЧ: A=5; C=3; T(0)=7; $M=2^b$; b=5; $M=2^5$=32. Сформируем три псевдослучайных числа:

T(1) = (5·7+3) mod 32 = 6 (00110);

T(2) = (5·6+3) mod 32 = 1 (00001);

T(3) = (5·1+3) mod 32 = 8 (01000).

Полученная гамма шифра 00110 00001 01000. Зашифрованный текст получается путем наложения гаммы шифра на исходный текст (путем сложения по модулю 2):

```
00001 00010 00011
00110 00001 01000
-----------------
00111 00011 01011
```

что соответствует шифрограмме «жвк», «ж» (седьмая буква в алфавите) имеет код 00111, «в» (третья буква в алфавите) имеет код 00011, «к» (одиннадцатая буква в алфавите) имеет код 01011.

Дешифрование производится путем наложения той же гаммы на зашифрованный текст с помощью операции сложения по модулю 2. В результате получаем исходный текст «абв».

```
00111 00011 01011
00110 00001 01000
-----------------
00001 00010 00011
```

Метод гаммирования с обратной связью

При использовании обратной связи значение зашифрованного символа зависит не только от гаммы, но и от предыдущих символов.

Для получения сегмента гаммы можно использовать контрольную сумму определенного участка шифруемых данных. Процесс шифрования в этом случае представляется следующими шагами:

1. Генерация сегмента гаммы $H(1)$ и наложение его на соответствующий участок шифруемых данных.

2. Подсчет контрольной суммы участка, соответствующего сегменту гаммы $H(1)$.

3. Генерация с учетом контрольной суммы уже зашифрованного участка данных следующего сегмента гамм $H(2)$.

4. Подсчет контрольной суммы участка данных, соответствующего сегменту данных $H(2)$ и т.д.

Под контрольной суммой понимают функцию $f(t(1), \ldots t(n))$, где $t(i)$ - i-е слово шифруемых данных.

Зашифруем исходный текст «абв», представленный в виде последовательности 00001 00010 00011. Пусть A=5; C=3; b=5; M=32;T(0)=7. Тогда T(1)=(5·7+3) mod 32 = 6 (00110).

В качестве контрольной суммы участка данных, выберем количество единиц на этом участке. Тогда сегменту H(1) соответствует участок 00001, количество единиц равно 1.

T(2)=(5·1+3) mod 32 = 8 (01000).

Контрольная сумма следующего участка (00010) равна 1.

T(3)=(5·1+3) mod 32 = 8 (01000).

Полученная шифрограмма: соответствует тексту «жик».

00001 00010 00011
00110 01000 01000
00111 01010 01011

Задание на лабораторную работу

1. Выбрать в таблице 2.1 параметры генератора ПСЧ: A, C, T_0, b в соответствии с вариантом.
2. Разработать программу шифрования и дешифрования текста.
3. Произвести шифрование исходного текста, получить шифрограмму, осуществить ее дешифрование и сравнение с исходным текстом. Рекомендуется для представления символов исходного текста использовать стандартную кодировку символов.
4. Произвести изменение одного или несколько параметров генератора случайных чисел, осуществить получение шифрограммы и сравнение ее с предыдущим вариантом.
5. Результаты работы оформить в виде отчета.

Таблица 2.1 – Генераторы ПСЧ

№ варианта	Вид генератора ПСЧ	Количество разрядов b
1	Линейные конгруэнтные датчики ПСЧ	6
2	Гаммирование с обратной связью	7
3	Линейные конгруэнтные датчики ПСЧ	8
4	Гаммирование с обратной связью	6
5	Линейные конгруэнтные датчики ПСЧ	7
6	Гаммирование с обратной связью	8
7	Линейные конгруэнтные датчики ПСЧ	6
8	Гаммирование с обратной связью	7
9	Линейные конгруэнтные датчики ПСЧ	8
10	Гаммирование с обратной связью	6
11	Линейные конгруэнтные датчики ПСЧ	7
12	Гаммирование с обратной связью	8
13	Линейные конгруэнтные датчики ПСЧ	6
14	Гаммирование с обратной связью	7
15	Линейные конгруэнтные датчики ПСЧ	8

Содержание отчета:

- цель работы, постановка задачи,
- описание используемого метода,
- описание исходных данных,
- алгоритм работы программы,
- текст программы,
- результаты работы программы,
- анализ результатов
- выводы.

Контрольные вопросы

1. Какие параметры конгруэнтного генератора необходимо выбрать для получения максимальной длины последовательности псевдослучайных чисел?
2. От чего зависит длина псевдослучайной последовательности?
3. Каков принцип действия генераторов с обратной связью?
4. Какую операцию используют для шифрования в методе гаммирования?
5. Каковы достоинства и недостатки метода гаммирования?
6. Что является ключом в шифрах гаммирования?

Лабораторная работа № 3. Сеть Фейштеля

Цель работы: изучить принципы работы сети Фейштеля, научиться шифровать информацию посредством использования блочного криптоалгоритма.

Криптографические алгоритмы на базе сети Фейштеля

Сеть Фейштеля - один из методов построения блочных шифров, который преобразовывает *n*-битный блок исходного текста в *n*-битный блок зашифрованного текста. Шифрование и дешифрование осуществляется на основе криптографического ключа *K*.

Классическая сеть Фейштеля имеет следующую структуру. Входной блок делится на несколько равной длины подблоков, называемых ветвями сети. В классической схеме их две (рисунок 3.1).

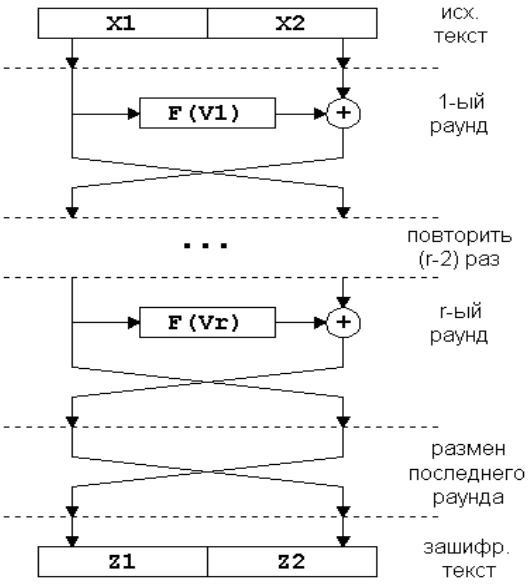

Рисунок 3.1 - Классическая структура сети Фейштеля

Величины V_i называются параметрами сети, обычно это функции от материала ключа. Функция F называется образующей. Действие, состоящее из однократного вычисления образующей функции и последующего наложения (сложения по модулю 2) ее результата на другую ветвь с обменом их местами, называется циклом или раундом (англ. round) сети Фейштеля. Оптимальное число раундов R – от 8 до 32. Часто количество раундов не фиксируется разработчиками алгоритма, а лишь указываются разумные пределы (обязательно нижний, и не всегда – верхний) этого параметра.

Данная схема является обратимой. Сеть Фейштеля обладает тем свойством, что даже если в качестве образующей функции F будет использовано необратимое преобразование, то и в этом случае вся цепочка будет восстановима. Это происходит вследствие того, что для обратного преобразования сети Фейштеля не нужно вычислять функцию F^{-1}.

Сеть Фейштеля симметрична за счет использования операции XOR и для ее обратимости не имеет значение является ли число раундов четным или нечетным числом.

Использование модификации сети Фейштеля для большего числа ветвей связано с тем, что при больших размерах кодируемых блоков (128 и более бит) становится неудобно работать с математическими функциями по модулю 64 и выше. Основные единицы информации обрабатываемые процессорами на сегодняшний день – это байт и двойное машинное слово 32 бита. Будет логично разбивать исходные блоки не на две, а на 4 части. В этом случае сеть Фейштеля может принимать следующий вид:

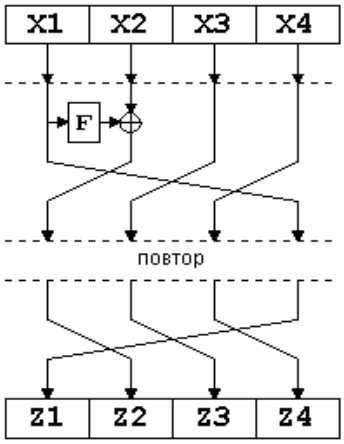

Рисунок 3.2 - Структура модифицированной сети Фейштеля

Алгоритм предназначен для шифрования и дешифрования информации, представленной в виде слов, разрядностью 128 бит на основе 64-битового ключа. Операции шифрования и дешифрования являются инверсными и используют один и тот же ключ.

Рассмотрим шифрование одного блока для сети с 4 ветвями.

Обозначим $X1X2X3X4$ конкатенацию последовательностей $X1$. $X2$, $X3$ и $X4$, в которой биты последовательностей $X1$, $X2$, $X3$, $X4$ следуют друг за другом. Размерность последовательности равна сумме размерностей всех составляющих. Символом + обозначим операцию побитового сложения по модулю 2.

Итеративный процесс шифрования описывается следующими формулами:

$X1(i) = X2(i\text{-}1)+F(V_i)$, $i = 1, 2, \ldots, n$;

$X2(i) = X3(i\text{-}1)$, $i = 1, 2, \ldots, n$;

$X3(i) = X4(i\text{-}1)$, $i = 1, 2, \ldots, n$;

$X4(i) = X1(i\text{-}1)$, $i = 1, 2, \ldots, n$;

где $F(V_i)$ - образующая функция;

n - количество раундов, может изменяться, в зависимости от требований по быстродействию и криптостойкости (n= 8 ÷128).

Функция F является основной характеристикой алгоритма, построенного на основе сети Фейштеля. Эта функция использует подключ раунда и одну ветвь входного блока для вычисления результата. Пример вычисления образующей приведен ниже.

$F_i = X1(i\text{-}1) + V_i(K)$

$V_i(K) = K1$ ROL $i + K2$ ROR i - параметр сети;

$K1$ и $K2$ - левая и правая части ключа K,

ROL и ROR - операции циклического сдвига влево и вправо соответственно.

Предлагаемый алгоритм имеет ряд достоинств. В первую очередь - простота реализации и высокое быстродействие, которое достигается за счет использования операций, имеющих высокую скорость выполнения.

Дешифрование блока информации производится той же сетью Фейштеля, но с инверсным порядком параметров сети. В явном виде ключ в алгоритме не используется, что повышает его криптостойкость. При знании ключа, но отсутствии информации о количестве раундов криптоаналитику будет достаточно сложно дешифровать зашифрованную информацию.

Задание на лабораторную работу

1. Выбрать из таблицы 3.1 параметры сети Фейштеля в соответствии с вариантом.
2. Разработать программу шифрования и дешифрования текста блоками, В программе предусмотреть ввод криптографического ключа, вычисление образующей функции, зависящей от материала ключа и части блока.
3. Произвести шифрование исходного текста, получить шифрограмму, осуществить ее дешифрование и сравнение с исходным текстом.
4. Результаты работы оформить в виде отчета.

Таблица 3.1 – Параметры сети Фейштеля

Номер вар.	Количество раундов	Образующая функция
1	8	Сложение
2	10	Исключающее ИЛИ
3	12	Циклический сдвиг вправо
4	14	Умножение по модулю 2^N
5	10	Арифметический сдвиг вправо
6	18	Арифметический сдвиг влево
7	20	Сложение
8	8	Умножение по модулю 2^N
9	24	Исключающее ИЛИ
10	20	Сложение
11	18	Умножение по модулю 2^N
12	28	Исключающее ИЛИ
13	12	Сложение
14	14	Циклический сдвиг влево
15	24	Исключающее ИЛИ

Содержание отчета:

- цель работы, постановка задачи,
- описание используемого метода,
- описание исходных данных,
- алгоритм работы программы,
- текст программы,
- результаты работы программы,

– анализ результатов
– выводы.

Контрольные вопросы

1. Какова структура классической сети Фейштеля?
2. Что называется раундом в сети Фейштеля?
3. Какими свойствами обладает сеть Фейштеля?
4. Каким образом используется материал ключа при шифровании?
5. В чем отличие процессов шифрования и дешифрования?
6. Назовите достоинства и недостатки блочных шифров.

Лабораторная работа №4. Изучение алгоритма RSA

Цель работы: Освоить механизм шифрования и дешифрования данных в криптографической системе с открытыми ключами RSA.

Теоретические основы криптосистем с открытым ключом

Главная проблема использования одноключевых (симметричных) криптосистем заключается в распределении ключей. Для того, чтобы был возможен обмен информацией между двумя сторонами, ключ должен быть сгенерирован одной из них, а затем в конфиденциальном порядке передан другой.

Суть шифрования с открытым ключом заключается в том, что для шифрования данных используется один ключ, а для расшифрования другой (поэтому такие системы часто называют ассиметричными).

Первый ключ, которым шифруется исходное сообщение, называется открытым и может быть опубликован для использования всеми пользователями системы. Расшифрование с помощью этого ключа невозможно. Второй ключ, с помощью которого дешифруется сообщение, называется секретным (закрытым) и должен быть известен только законному получателю закрытого сообщения.

Алгоритмы шифрования с открытым ключом используют так называемые необратимые или односторонние функции. Эти функции обладают следующим свойством: при заданном значении аргумента $\boldsymbol{x}$ относительно просто вычислить значение функции $f(x)$, однако, если известно значение функции $\boldsymbol{y} = f(x)$, то нет простого пути для вычисления значения аргумента **x**.

Система открытого распространения ключей позволяет двум сторонам сформировать совместную часть некоторой распределенной секретной информации. Однако, ни одна из сторон не имеет никакого непосредственного влияния на то, какой окажется эта информация.

Криптосистема RSA

RSA – криптографическая система с открытым ключем, обеспечивающая такие механизмы защиты как шифрование и цифровая подпись (аутентификация – установление подлинности).

Алгоритм RSA работает следующим образом:

Пусть p и q - два больших различных простых числа, и пусть $n = p \cdot q$ и e некоторое целое, взаимно простое с $(p-1)\cdot(q-1)$.

Пространства открытых текстов M_k и зашифрованных сообщений C_k представляют собой множество неотрицательных целых чисел Z_n, меньших n. Если исходное сообщение окажется слишком длинным, чтобы принадлежать Z_n, его необходимо разбить на части, равные m.

Соответствующая ключу k функция шифрования E_k: M_k -> C_k определяется как

$$E_k(m) = m^e \, mod \, n \quad (4.1)$$

Для того, чтобы полностью определить алгоритм ее вычисления, достаточно записать *e* и *n* в открытый справочник. Такая пара называется открытым ключом.

E_k является кандидатом на однонаправленную функцию с потайным ходом. Эффективный алгоритм вычисления *Dk* легко получить, задав дополнительную секретную информацию *p* и *q*. С этой целью, используя обобщенные алгоритмы Евклида для нахождения наибольшего общего делителя, чтобы вычислить целое число *d*, такое что $e \cdot d = 1 \mod \phi(n)$, где $\phi(n) = (p-1)\cdot(q-1)$ – функция Эйлера. По теореме Эйлера $m^{(ed)} = m \mod(n)$ для любого целого числа *m* и, следовательно, $(m^e)^d \mod(n) = m$, при условии что $0 <= m < n$, что обеспечивается, когда *m* принадлежит M_k.

Функция дешифрования D_k: C_k -> M_k в связи с этим определяется как $D_k(c) = c^d \mod(n)$, и эффективный алгоритм для модульного возведения в степень также может быть использован и для ее вычисления. Тогда каждый пользователь криптосистемы RSA должен выбрать целые числа *p*, *q* и *e* и вычислить с их помощью *d*. После чего он делает свой открытый ключ доступным в пользовательском справочнике, тогда как *d* сохраняет в секрете. Это дает возможность любому другому пользователю шифровать посылаемые ему сообщения, которые только он и может расшифровать. Для того чтобы эта идея была реализована практически, решающим является удовлетворение требование, чтобы генерация больших случайных простых чисел и вычисление *d* были легко осуществимы.

Например, пусть $p = 19$ и $q = 23$, тогда $n = 437$ и $\phi(n) = 396$. Пусть также $e = 13$, и поэтому $d = 61$, так как $13 \cdot 61 = 793 = 2\phi(n)+1$. Тогда сообщение в открытом тексте $m = 123$ будет зашифровано как $c = 123^{13} \mod(437) = 386$. Действительно, $386^{61} \mod(437) = 123$.

Если бы существовали эффективные методы разложения на сомножители (факторинга), то, разложив *n* на сомножители (факторы) *p* и *q*, можно было бы получить секретный (private) ключ *d*. Таким образом надежность криптосистемы RSA основана на трудноразрешимой задаче разложения *n* на сомножители (то есть на невозможности факторинга *n*) так как в настоящее время эффективного способа поиска сомножителей не существует.

Алгоритм шифрования и дешифрования RSA.

1. Выбирается два больших простых числа *p* и *q*.
2. Определяется $n=p \cdot q$.
3. Выбирается большое случайное число *e*. Это число должно быть взаимно простым с результатом $(p-1)\cdot(q-1)$.
4. Определяется такое число *d*, для которого является истинным соотношение $(e \cdot d) \mod((p-1)\cdot(q-1))=1$.
5. Открытым ключом являются числа *e* и *n*, а секретным ключом - числа *d*, *p* и *q*.

После произведенного выбора открытого и секретного ключей для шифрования данных необходимо выполнить следующие действия:

– разбить шифруемый текст на блоки, каждый из которых может быть представлен в виде числа $M_i = 0,1,...,n-1$;

– зашифровать текст, рассматриваемый как последовательность чисел $M(i)$ по формуле:

$$C_i = M_i^{\,e} \, mod \, n \quad (4.2)$$

Чтобы расшифровать эти данные, используется секретный ключ $\{d, n\}$ и выполняются следующие вычисления:

$$M_i = C_i^{\,d} \, mod \, n \quad (4.2)$$

В результате получают исходный текст M_i.

Задание на лабораторную работу

1. Разработать программу, осуществляющую шифрование и дешифрование сообщения алгоритмом RSA. Ключи генерируются на основе чисел *p* и *q*, значения которых

выбирается из таблицы 4.1 в соответствии с вариантом. При выборе числа *e* использовать минимально возможное

2. Исходное сообщение M может состоять из символов. как русского, так и любого другого алфавита.
3. Обеспечить вывод ключей и зашифрованного текста.
4. В программе предусмотреть проверку, являются ли два числа взаимно простыми.
5. Результаты работы оформить в виде отчета.

Таблица 4.1 –Данные для генерации ключей в методе RSA

Номер вар.	*p, q*	Номер вар	*p, q*
1	193, 353	8	163, 409
2	211, 317	9	227, 307
3	157, 433	10	233, 293
4	179, 383	11	241, 307
5	223, 317	12	167, 401
6	199, 337	13	271, 277
7	181, 367	14	137, 479

Содержание отчета:

- цель работы, постановка задачи,
- описание используемого метода,
- описание исходных данных,
- алгоритм работы программы,
- текст программы,
- результаты работы программы,
- анализ результатов
- выводы.

Контрольные вопросы

1. Что такое однонаправленные функции?
2. Основные свойства однонаправленных функций с потайным ходом.
3. Какие числа называются взаимно простыми?
4. Как реализуется программное возведение в степень для больших чисел?
5. На чем основана криптостойкость алгоритма RSA?
6. Каковы достоинства и недостатки асимметричных алгоритмов?

Лабораторная работа № 5. Создание электронной подписи в документе

Цель работы: разработка процедур выработки и проверки электронной цифровой подписи (ЭЦП) сообщений на базе асимметричного криптографического алгоритма с применением функции хеширования.

Теоретические положения

Схема цифровой подписи - набор алгоритмов и протоколов, позволяющих построить информационное взаимодействие между двумя и более участниками таким образом, чтобы факт авторства переданного массива данных, «подписанного одним из участников», мог быть надежно подтвержден или опровергнут третьей стороной.

Любая схема цифровой подписи предполагает добавление к подписываемому массиву данных дополнительного кода - цифровой подписи, выработать которую может только автор сообщения, обладающий секретным ключом подписи, а все остальные могут лишь проверить соответствие этой подписи подписанным данным.

Процедура ЭЦП на базе асимметричного криптографического алгоритма включает в себя процедуры выработки и проверки подписи под данным сообщением. Цифровая подпись, состоящая из двух целых чисел, вычисляется с помощью определенного набора правил.

ЭЦП обеспечивает:

Защиту от изменений документа. При любом случайном или преднамеренном изменении документа (или подписи) подпись станет недействительной, потому что вычислена она на основании исходного состояния документа и соответствует лишь ему.

Невозможность отказа от авторства. Так как создать корректную подпись можно лишь, зная закрытый ключ, а он известен только владельцу, то владелец не может отказаться от своей подписи под документом.

Алгоритм ЭЦП ГОСТ Р34.10-94

Российским стандартом на процедуры выработки и проверки электронно-цифровой подписи до 2001 года являлся ГОСТ Р34.10-94, основанный на задаче дискретного логарифмирования. После подписывания сообщения M к нему дописывалась цифровая подпись размером 512 бит, состоящая из двух числ.

Выбор параметров системы ЭЦП

Параметры системы ЭЦП - числа p,q,a. Эти числа не являются секретными. Конкретный набор их значений может быть общим для группы пользователей.

1. p - простое число (по ГОСТ $2^{509}<p<2^{512}$)

2. q –простое число, q должно быть простым делителем числа p-1(по ГОСТ $2^{254}<q<2^{256}$)

3. a – целое число, $1<a<p-1$, при этом $a^q \pmod p = 1$

4. k - целое число, $1<k<q$, k – секртгный сеансовый ключ, генерируется в поцессе формирования подписи, после подписывания сообщении уничтожается.

4. x - секретный ключ для формирования подписи $1<x<q$

5. y - открытый ключ для проверки подписи, $y=a^x \bmod p$.

Генерация электронно-цифровой подписи

Процесс генерации электронно-цифровой подписи состоит из нескольких этапов:

1. Вычисляется хэш-код сообщения m: $h=H(m)$, если $h(m) \pmod q = 0$,то $h(m)$ присваивается значение 1.

2. Из диапазона $[1,q]$ случайным образом выбирается значение k

3. Вычисляется $r= (a^k \bmod p)$, $r1=r \pmod q$;

если $r1=0$, следует вернуться к предыдущему этапу и выработать другое значение k.

4. Вычисляется:

$s= (x \cdot r1+k \cdot h(m)) \pmod q$;

если s=0, то необходимо вернуться к п.2 и выработать другое значение k.

Значения $r1,s$ являются электронно-цифровой подписью сообщения m и передаются вместе с ним по каналам связи.

Проверка электронно-цифровой подписи

Проверка электронно-цифровой подписи осуществляется с использованием открытого ключа и происходит следующим образом:

1. Проверяется выполнение условий $0<r1<q$, $0<s<q$, и если хотя бы одно из них нарушено, подпись отвергается.

2. Вычисляется хэш-код полученного сообщения m_1 $h=H(m_1)$;

Если $h(m_1)(\text{mod } q) = 0$, то $h(m_1)=1$

3. Вычисляется значение $v=(h(m_1))^{q-2}(\text{mod } q)$.

4. Вычисляется значения $z_1=s \cdot v(\text{mod } q)$; $z_2=(q-r1)v(\text{mod } q)$.

5. Вычисляется значение $u=(a^{z1} \cdot y^{z2}(\text{mod } p))(\text{mod } q)$

6. проверяется равенство $u = r1$.Если равенство выполняется, то подпись принимается. В противном подпись считается недействительной.

Криптографические хэш-функции

Хэширование – преобразование входного массива данных произвольной длины в выходную битовую строку фиксированной длины. Такие преобразования также называются хэш-функциями или функциями свёртки, а их результаты называют хэшем, хэш-кодом или дайджестом сообщения.

Однонаправленная функция $h=H(m)$ должна обладать свойствами:

1. Хэш-функция h должна применяться к блоку данных любой длины.
2. Хэш-функция h создает выход фиксированной длины.
3. $H(m)$ относительно легко (за полиномиальное время) вычисляется для любого значения M.
4. Для любого данного значения хэш-кода h вычислительно невозможно найти M такое, что $H(M) = h$.
5. Для любого данного x вычислительно невозможно найти y, не равный x, что $H(y) = H(x)$.
6. Вычислительно невозможно найти произвольную пару (x, y) такую, что $H(y) = H(x)$.

Однонаправленные хэш-функции строятся на идее функции сжатия. Входами функции сжатия являются блок сообщения и выход предыдущего блока текста. Выход представляет собой хэш-значение всех блоков до этого момента. Т.е. хэш-значение блока равно $h_i=H(m_i, h_{i-1})$.

В криптографии применяют криптографически стойкие функции:

- MD4, MD5;
- SHA1, SHA224, SHA256, SHA384, SHA512;
- RIPEMD160, RIPEMD256, RIPEMD320.

Для алгоритма ГОСТ Р34.10-94 вычисление функции хэширования установлено в ГОСТ Р 34.11.

Простейшими примерами хэш-функций могут служить контрольные суммы. Контрольные суммы – это несложные, крайне быстрые и легко реализуемые алгоритмы, используемые для защиты от непреднамеренных искажений, в том числе ошибок аппаратуры. Недостатком контрольных сумм является низкая криптостойкость. Примером таких алгоритмов служат деление сообщения на 32- или 16-битные слова и их суммирование, что применяется, например, в TCP/IP или CRC32, применяемый в аппаратуре Ethernet и в формате упакованных файлов ZIP.

Задание на лабораторную работу

1. Выбрать из таблицы 5.1 в соответствии с вариантом алгоритм вычисления хэш-функции (контрольной суммы).
2. Реализовать программную реализацию алгоритма создания и проверки электронно-цифровой подписи.
3. Подписать текстовое сообщение
4. Проверить правильность ЭЦП.
5. Внести изменения в сделанную подпись. Убедится, что подпись не является подлинной.
6. Результаты работы оформить в виде отчета.

Таблица 5.1.- Варианты контрольных сумм

Номер вар.	Алгоритм вычисления контрольных сумм
1	Количество 1 в битовом представлении символов исходного текста
2	Сложение кодов символов исходного текста модулю 2^{16}
3	Разбиение исходного текста на блоки по 32 бита и выполнение над ними операции сложения по модуля 2.
4	Умножение кодов символов исходного текста по модулю 2^{16}
5	Разбиение исходного текста на блоки по 16 бит и выполнение над ними операции сложения по модулю 2 и циклического сдвига на 1 разряд вправо при каждой операции XOR
6	Количество 1 в битовом представлении символов исходного текста
7	Разбиение исходного текста на блоки по 16 бит и выполнение над ними операции сложения по модулю 2 и циклического сдвига на 1 разряд влево при каждой операции XOR
8	Разбиение исходного текста на блоки по 16 бит и выполнение над ними операции сложения по модулю 2 и арифметического сдвига на 1 разряд вправо при каждой операции XOR
9	Количество 1 в битовом представлении символов исходного текста
10	Разбиение исходного текста на блоки по 16 бит и выполнение над ними операции сложения по модулю 2 и арифметического сдвига на 1 разряд влево при каждой операции XOR
11	Умножение кодов символов исходного текста по модулю 2^{16}
12	Количество 1 в битовом представлении символов исходного текста

Содержание отчета:

- цель работы, постановка задачи,
- описание используемого метода,
- описание исходных данных,
- алгоритм работы программы,
- текст программы,
- результаты работы программы,
- анализ результатов
- выводы.

Контрольные вопросы

1. Какие криптоалгоритмы используются для создания электрооной цифровой подписи?
2. Что такое криптографическая хэш-функция, какими свойствами она должна обладать?
3. Как содержание сообщение влияет на электронную цифровую подпись?
4. Где используется ЭЦП?
5. В каком случае электронная цифровая подпись при проверке отвергается?
6. От каких угроз информации защищает ЭЦП?

Лабораторная работа №6. Защита графического файла с помощью цифрового водяного знака

Цель работы: Изучение стеганографических методов защиты информации. Реализация программы с использованием стеганографических принципов защиты информации.

Стеганография

Стеганография — в переводе с греческого дословно означает «тайнопись». Это наука о скрытой передаче информации путём сохранения в тайне самого факта передачи. В отличие от криптографии, которая скрывает содержимое секретного сообщения, стеганография скрывает само его существование.

В настоящее время под стеганографией чаще всего понимают скрытие информации в графических, аудио либо текстовых файлах путём использования специального программного обеспечения.

Из рамок цифровой стеганографии вышло наиболее востребованное легальное направление — встраивание цифровых водяных знаков (ЦВЗ) (watermarking), являющееся основой для систем защиты авторских прав и DRM систем. Методы этого направления настроены на встраивание скрытых маркеров, устойчивых к различным преобразованиям контейнера (атакам).

Требования к цифровым водяным знакам:

1. ЦВЗ должен легко (вычислительно) извлекаться законным пользователем.

2. ЦВЗ должен быть устойчивым либо неустойчивым к преднамеренным и случайным воздействиям (в зависимости от приложения). Если ЦВЗ используется для подтверждения подлинности, то недопустимое изменение контейнера должно приводить к разрушению ЦВЗ (хрупкий ЦВЗ). Если же ЦВЗ содержит идентификационный код, логотип фирмы и т. п., то он должен сохраниться при максимальных искажениях контейнера, конечно, не приводящих к существенным искажениям исходного сигнала. Например, у изображения могут быть отредактированы цветовая гамма или яркость, у аудиозаписи— усилено звучание низких тонов и т. д.

3. Должна иметься возможность добавления к стего дополнительных ЦВЗ.

Объект (сообщение), в который происходит вложение информации, называется контейнером. Если контейнером является изображение, то общую схему встраивания информации в изображение можно представить так, как это сделано на рисунке 6.1.

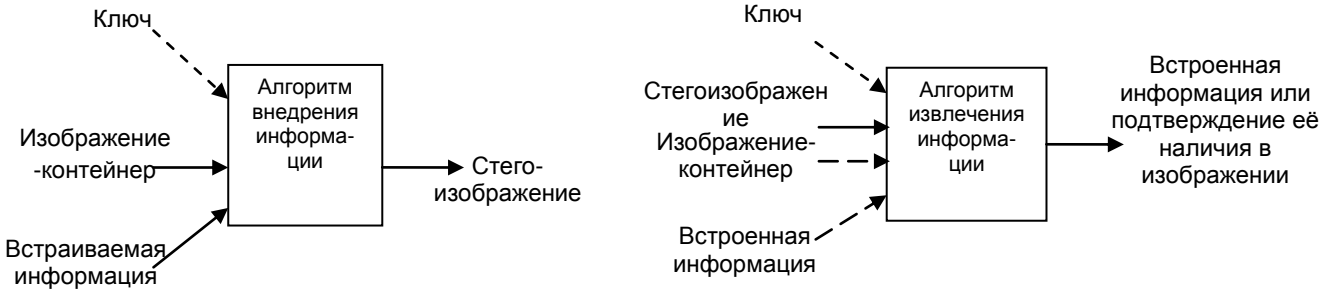

Рисунок 6.1 - Общая схема встраивания и извлечения информации

----> означает, что компонента может отсутствовать. Это зависит от используемого алгоритма внедрения информации и поставленной прикладной задачи.

Причины распространённости использования в качестве контейнера неподвижного изображения:

- размер контейнера заранее известен;
- отсутствуют ограничения режима передачи в реальном времени;

- возможность внедрения информации большого объёма;
- слабая чувствительность глаза человека к некоторым изменениям характеристик изображения.

LSB-алгоритм

Цифровые изображения представляют собой матрицу пикселей. Пиксель – это единичный элемент изображения. Он имеет фиксированную разрядность двоичного представления. Например, пиксели полутонового изображения кодируются 8 битами (значения яркости изменяются от 0 до 255).

Младший значащий бит (LSB) изображения несет в себе меньше всего информации. Известно, что человек обычно не способен заметить изменение в этом бите. Фактически, он является шумом. Поэтому его можно использовать для встраивания информации. Таким образом, для полутонового изображения объем встраиваемых данных может составлять 1/8 объема контейнера. Например, в изображение размером 512x512 можно встроить 32 килобайта информации. Если модифицировать два младших бита (что также почти незаметно), то можно скрытно передать вдвое больший объем данных.

Встраивание ЦВЗ происходит путем модификации Допустим, имеется 8-битное изображение в градациях серого. 00h (00000000b) обозначает чёрный цвет, FFh (11111111b) — белый. Всего имеется 256 градаций (2^8). Пусть сообщение состоит из 1 байта — например, 01101011b. При использовании 2 младших бит в описаниях пикселей, потребуется 4 пикселя. Допустим, они чёрного цвета. Тогда пиксели, содержащие скрытое сообщение, будут выглядеть следующим образом: 00000001 00000010 00000010 00000011. Тогда цвет пикселей изменится: первого — на 1/255, второго и третьего — на 2/255 и четвёртого — на 3/255. Такие градации, мало того что незаметны для человека, могут вообще не отобразиться при использовании низкокачественных устройств вывода.

Обнаружение LSB-кодированного стего осуществляется по аномальным характеристикам распределения значений диапазона младших битов отсчётов цифрового сигнала.

Достоинства рассматриваемого метода заключаются в его простоте и сравнительно большом объеме встраиваемых данных. Однако, он имеет серьезные недостатки. Во-первых, скрытое сообщение легко разрушить. Во-вторых, не обеспечена секретность встраивания информации. Нарушителю точно известно местоположение всего секретного сообщения.

Выделяются два метода внедрения: LSB-Replacement и LSB-Matching. Первый из них (LSB-R) состоит в простой замене LSB на бит внедряемого сообщения, будь то 0 или 1. При последовательном внедрении в пиксели сообщения 1001, они меняются следующим образом - 1001: 51 80 121 62 $\rightarrow$ 51 80 120 63. Второй метод (LSB-M, называемый также ±1-внедрение) немного сложнее. Если бит внедряемого сообщения равен LSB, то ничего не делается. В противном случае с одинаковой вероятностью производится прибавление либо вычитание единицы из значения пикселя (в исключительных случаях, когда это значение равно 0 либо 255, используется метод LSB-R). Пример внедрения сообщения 1001 выглядит так: 51 80 121 62 $\rightarrow$ 51 80 122 (либо 120) 63 (либо 61). При использовании и того и другого метода биты внедряемого сообщения оказываются в LSB (т.е. метод извлечения информации один и тот же).

Задание к лабораторной работе

1. Написать программу внедрения и извлечения скрытой информации в BMP-файлы с использованием стеганографических (LSB) алгоритмов. В качестве контейнера использовать графический формат BMP. В алгоритме LSB для четных вариантов число используемых младших бит - 2 бита, использовать метод LSB-R, для нечетных вариантов число используемых младших бит - 1 бит, использовать метод LSB-M.
2. Результаты работы оформить в виде отчета.

Содержание отчета:

- цель работы, постановка задачи,
- описание используемого метода,
- описание исходных данных,
- алгоритм работы программы,
- текст программы,
- результаты работы программы,
- анализ результатов
- выводы.

Контрольные вопросы

1. Что собой представляет стеганография?
2. Перечислите области применения стеганографических алгоритмов.
3. Каковы требования к цифровым водяным знакам?
4. В чем суть LSB-алгоритма?
5. От чего зависит стойкость стегосистем?
6. Каковы особенности встраивания и извлечения информации из стегоконтейнера?

Лабораторная работа № 7. Парольная защита

Цель работы: изучение принципов организации парольной защиты программ, ознакомление с видами паролей, реализация парольной защиты.

Парольная идентификация

Стандартность архитектурных принципов построения, оборудования и программного обеспечения персональных компьютеров, мобильность программного обеспечения определяют сравнительно легкий доступ к информации, находящейся в персональном компьютере. Несанкционированный доступ к информации персонального компьютера - незапланированное ознакомление, обработка, копирование, применение различных вирусов, модификация или уничтожение информации в нарушение правил доступа.

Под защитой информации понимают создание организованной совокупности средств, методов и мероприятий, предназначенных для предупреждения искажения, уничтожения или несанкционированного использования защищаемой информации. К ним относятся аппаратные и программные средства, криптографическое закрытие информации, физические меры, организационные мероприятия и законодательные меры. Один из методов защиты - парольная идентификация, ограничивающая доступ несанкционированного пользователя.

Включение защиты в программу связано с разработкой программ с запросом информации, т.е. требующих для своей работы ввода дополнительной информации, такой как пароли или номера ключей. Однако такая проверка доступа к программам или системам не должна существенно сказываться на быстродействии программы или требовать от пользователя сложных дополнительных действий.

Пароль - это код, используемый для получения доступа к системам или файлам, оснащенным парольной защитой. Пароли обеспечивают сохранение целостности программного обеспечения в составе вычислительной системы, но для поддержания паролей требуется высокая дисциплинированность. При первой регистрации пользователя администратор определяет круг полномочий для получения и изменения информации или выполнения определенных управляющих действий в системе, руководствуясь его профессиональными обязанностями и должностными инструкциями. Затем пользователю прелагается ввести свой пароль согласно правилам, принятым в данной системе. Метод паролей требует, чтобы вводимый пользователем пароль (строка символов) сравнивался с

тем, который хранится в вычислительной системе для данного пользователя. Если пароль верен, система должна вывести на экран терминала дату и время последнего входа в систему этого пользователя. Затем пользователю предоставляется возможность пользоваться всей информацией, доступ к которой ему разрешен (пароли можно также использовать независимо от пользователя для защиты файлов, записей, полей данных внутри записей и т.д.). Процедура установления подлинности пользователей с помощью пароля приведена на рисунке 7.1.

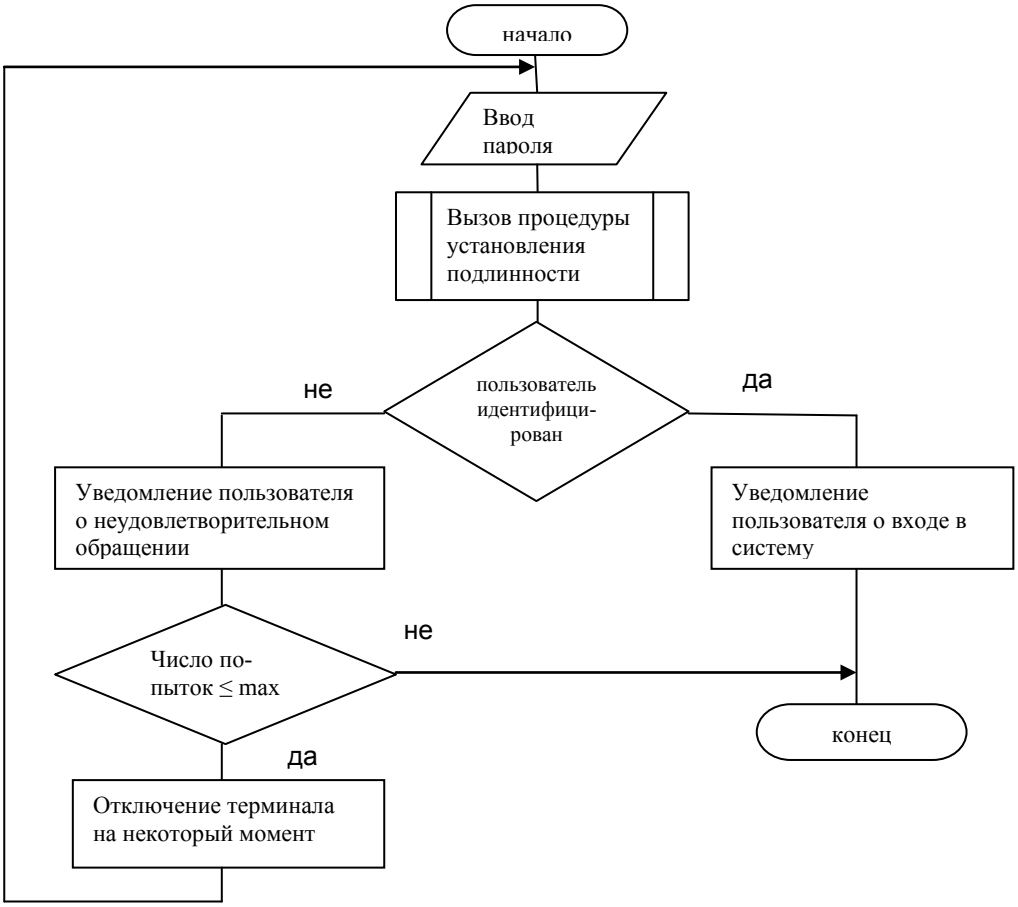

Рисунок 7.1 - Схема установления подлинности пользователя.

Парольная защита является достаточно эффективной, если:
- сохранять пароль в тайне;
- просматривать систему для поиска резидентных программ или троянских коней, предназначенных для перехвата паролей; установить защиты в системе от таких программ;
- установить требования к минимальной длине и множеству символов в паролях;
- при наличии средств использовать интеллектуальные карты, опознавательные знаки, биометрические устройства управления доступом;
- осуществлять периодическое изменение паролей и контроль их сроков действия.

Система не должна отображать вводимые пользователем пароли, либо на месте ввода выводить последовательность случайных символов. Не следует хранить пароли в открытом в виде или на носителе. Немедленно после ввода пароля производить шифрование пароля и очистку памяти, содержащую открытый текст пароля. Для предотвращения угадывания пароля рекомендуется использовать пароли, генерируемые компьютером, а также

производить блокировку после определенного количества попыток ввода неправильного пароля.

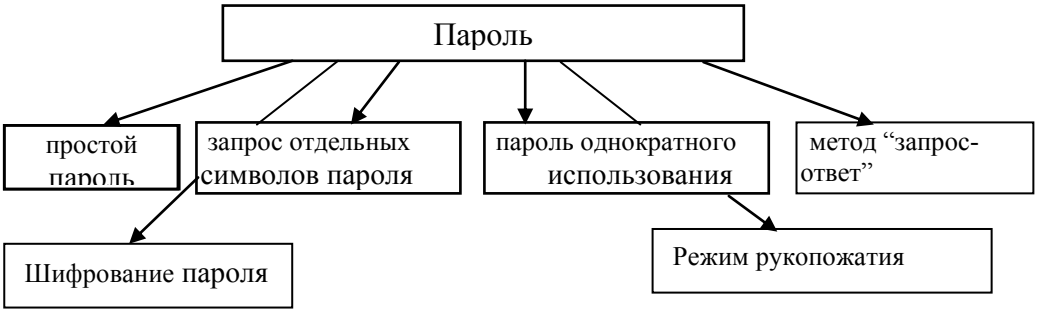

Рисунок 7.2 - Виды паролей

Простой пароль

Простой пароль - вводимая пользователем с клавиатуры строка символов. В схеме с простым паролем пользователю разрешается самому выбирать пароль таким образом, чтобы его было легко запомнить. Иногда в ряду символов пароля и в конце его оставляют пробелы. Отличие действительного пароля от кажущегося (без пробелов) повышает защищенность системы.

Подбор пароля путем простого перебора комбинаций предполагает перебор всех возможных сочетаний символов в пароле. Время, необходимое для разгадывания пароля методом простого перебора, является геометрической прогрессией от длины пароля, но есть различные кривые, зависящие от размера алфавита, на основе которого был создан пароль и от размера набора символов, по отношению к которым рассматриваются различные пароли.

Согласно формуле Андерсена:

$$4{,}32 \cdot 10^4 \cdot R \cdot M / E \cdot P \leq A^S \qquad (7.1)$$

где R - скорость передачи в линии связи (симв./с);

E - число символов в каждом передаваемом сообщении при попытке получить доступ;

S - длина пароля;

A - число символов в алфавите, из которого составлен пароль;

P - вероятность правильного отгадывания пароля.

Наибольшее влияние на вероятность P раскрытия пароля оказывает величина S. Увеличение пароля на один символ значительно увеличивает время для раскрытия этого пароля. Поэтому применение очень длинных паролей может быть обосновано. Борьба с перебором комбинаций заключается в использовании программного обеспечения, ограничивающего минимальную длину пароля и использовании более обширного алфавита (256 символов).

Выборка символов

Использование в качестве пароля отдельных символов условного слова (например, 1 и 5 буква) предотвращает ситуацию, когда целое слово может быть случайно услышано. Запрашиваемые символы пароля изменяются при каждой новой попытке доступа. Позиции запрашиваемых символов можно получить с помощью некоторой процедуры преобразования, привязанной к показаниям часов ЭВМ или выработать генератором псевдослучайных чисел. Однако пароль следует изменять достаточно часто, поскольку он может быть составлен из отдельных символов.

Пароль однократного использования

В схеме однократного использования пароля пользователю выдается список из N паролей. Такие же N паролей хранятся в ЭВМ (в зашифрованном виде). Данная схема обеспечивает большую степень безопасности, но она является и более сложной. После

использования пароля пользователь вычеркивает его из списка. При дальнейшей работе система на этот пароль реагировать не будет, поскольку ожидает следующий по списку пароль.

Пароли однократного использования могут применяться также для установления подлинности подтверждения об отключении ЭВМ от обслуживания пользователя и подтверждения подлинности требования пользователя об отключении от ЭВМ. Всякий раз, когда получено требование пользователя об окончании работы, ЭВМ немедленно передает ему свой пароль однократного использования и прерывает связь. Если пользователь отключается и не получает истинного пароля от ЭВМ, ему следует принять меры предосторожности.

Недостатки паролей однократного использования:

- пользователь должен помнить весь список паролей и следить за текущим паролем;
- в случае ошибки в процессе передачи пользователь не знает, следует ли ему передать тот же пароль или послать следующий;
- если пароли определены путем использования линейной последовательности псевдослучайных чисел, то первоначальная последовательность может быть восстановлена на основании нескольких перехваченных паролей.

Метод “запрос-ответ”

В методе “запрос-ответ” набор ответов на m стандартных и n ориентированных на пользователя вопросов хранится в ЭВМ и управляется операционной системой. Когда пользователь делает попытку включиться в работу, операционная система случайным образом выбирает и задает ему некоторые (или все) из этих вопросов. Пользователь должен дать правильный ответ на все вопросы, чтобы получить разрешение на доступ к системе. Иногда пользователям задается большое количество стандартных вопросов и от них требуются ответы на те, которые они выберут.

Шифрование паролей

Шифрование пароля повышает безопасность системы. Этот метод предполагает, что пароль, вводимый при входе в систему, шифруется и сравнивается с зашифрованным паролем, хранящимся в базе данных. Для шифрования пароля можно использовать простой метод обратимого шифрования или более сложный метод “необратимой беспорядочной сборки”, когда несколько паролей в явной форме преобразуются в одинаковый зашифрованный пароль. В этом случае не существует никакой схемы для возвращения к оригиналу пароля. Система просто шифрует каждый пароль пользователя во время процесса регистрации и сверяет его с зашифрованным паролем, хранящимся в собственном файле пользователя.

Пример этого метода - полиномиальное необратимое представление:

$$f(x) = (x^n + a_1 \cdot x^{n-1} + \cdots + a_{n-1} \cdot x + a_0) mod\ P \qquad (7.2)$$

где P- большое, a_i и n - целые числа; x -пароль в явной форме; $f(x)$ - зашифрованный пароль.

Режим “рукопожатия”

Операционная система может потребовать, чтобы пользователь установил свою подлинность с помощью корректной обработки алгоритмов, которую называют режимом “рукопожатия” и она может быть выполнена как между двумя ЭВМ, так и между пользователем и ЭВМ.

ЭВМ для установления подлинности могла дать пользователю число, выбранное случайным образом, а затем запросить от него ответ. Для подготовки ответа пользователь “u” применяет собственное заранее подготовленное преобразование t_u. Информацией, на основе которого принимается решение, здесь является не пароль, а преобразование t_u. ЭВМ посылает значение x, а пользователь отвечает значением $t_u(x)$. Даже в случае знания значений x и $t_u(x)$

угадать функцию преобразования невозможно. Функция преобразования может быть различна для каждого пользователя.

Порядок выполнения работы

1. Изучить существующие методы парольной защиты
2. Выбрать метод парольной защиты в соответствии с заданным вариантом из таблицы 7.1.
3. Разработать алгоритм и программную реализацию выбранного метода парольной защиты с использованием демонстрационных возможностей выбранного языка программирования.
4. Оформить отчет.

Таблица 7.1 – Виды парольной защиты

№ варианта	Вид парольной защиты
1	Выборка символов
2	Пароль однократного использования
3	Шифрование паролей
4	Метод “запрос-ответ”
5	Режим “рукопожатия”
6	Выборка символов
7	Простой пароль
8	Шифрование паролей
9	Метод “запрос-ответ”
10	Режим “рукопожатия”
11	Выборка символов
12	Простой пароль
13	Пароль однократного использования
14	Шифрование паролей
15	Выборка символов
16	Метод “запрос-ответ”
17	Простой пароль
18	Режим “рукопожатия”
19	Выборка символов

Содержание отчета:

- цель работы, постановка задачи,
- описание используемого метода,
- описание исходных данных,
- алгоритм работы программы,
- текст программы,
- результаты работы программы,
- анализ результатов
- выводы.

Контрольные вопросы

1. При соблюдении каких условий парольная защита является эффективной?
2. Каковы недостатки парольной защиты?
3. Что такое метод рукопожатия?
4. Какие операционные системы имеют встроенную парольную защиту?
5. Сравните медоды простой парольной защиты и выборку символов.
6. Как реализуют пароли однократного использования.

Лабораторная работа № 8. Реализация протокола Диффи-Хеллмана на эллиптических кривых

Цель работы: изучение особенностей реализации криптографических протоколов распределения ключей, асимметричной криптографии на эллиптических кривых, разработка системы распределения криптографических ключей.

.Криптографические протоколы распределения ключей

Протоколом называют совокупность правил, регламентирующих: последовательность шагов, предпринимаемых двумя или большим количеством сторон для совместного решения некоторой задачи, форматы сообщений, пересылаемых между участниками обмена, действия при возникновении сбоев. Криптографический протокол — протокол, процессе выполнения которого участники используют криптографические алгоритмы.

Основной задачей протоколов распределения ключей является выработка участниками общего ключа на основе действий пользователей по созданию защищенного канала связи, заключающаяся в генерации и обмене сеансовыми ключами и аутентификации сообщений.

Одним из самых распространенных способов ключевой генерации и обмена является протокол Диффи-Хеллмана, основанный на асимметричной криптографии. Стойкость протокола базируется на сложности решения задачи дискретного логарифмирования. Пользователи предварительно договариваются о параметрах системы n и g. В процессе выполнения шагов протокола каждый пользователь генерирует случайное число (x – первый пользователь, y – второй) и обмениваются сообщениями $g^x \bmod n$, $g^y \bmod n$. После обмена данными пользователи вычисляют общий ключ $g^{xy} \bmod n$. Данные, передаваемые по сети, не позволяют злоумышленнику восстановить ключ, это требует нахождения дискретного логарифма, а это задача не решается за приемлемое время.

Криптография на основе эллиптических кривых

Криптография на основе эллиптических кривых – подход асимметричной криптографии, при котором открытые и закрытые ключи обозначаются как точки на математическом объекте, называемом эллиптической кривой. При использовании таких алгоритмов на полагается, что не существует субэкспоненциальных алгоритмов для решения задачи дискретного логарифмирования в группах таких точек. При этом порядок группы точек эллиптической кривой определяет сложность задачи. Считается, что для достижения такого же уровня криптостойкости как и в RSA, требуются группы меньших порядков, что уменьшает затраты на хранение и передачу информации.

В криптографии применяется уравнение эллиптической кривой E вида:

$$y^2 = x^3 + a \cdot x + b \ (mod\ p), \tag{8,1}$$

Где p - простое, a и b должны удовлетворять условию

$$4 \cdot a^3 + 27 \cdot b^2 \neq 0 \ (mod\ p) \tag{8.2}$$

Такая кривая обозначается E_p (a,b) (рисунок 8.1)

На эллиптической кривой определены две операции: сложение точек и удвоение точки. Нулем является точка O, также называемая «бесконечно удаленная точка». В этой точке сходятся все вертикальные прямые. Сумма трех точек лежащих на одной прямой равна O. Если P, Q и $R(x, y)$ лежат на одной прямой, то $P + Q = (x, –y)$.

Правила сложения можно записать в виде:

$$(x, y) + O = (x, y),\ O + O = O\ .\ (x, y) + (x, -y) = O. \tag{8.3}$$

Суммой двух точек $P(x_1, y_1)$ и $Q(x_2, y_2)$ называется точка $–R(x_3, y_3)$, обратная точке R пересечения эллиптической кривой и прямой, проходящей через точки P и Q.

Координаты x_3, y_3 определяются по формулам:

$$x_3 = \lambda^2 - x_1 - x_2 (mod\ p), \quad y_3 = \lambda \cdot (x_1 - x_3) - y_1 (mod\ p) \tag{8.4}$$

где

$$\lambda = \frac{y_2 - y_1}{x_1 - x_2} mod\, p$$

.

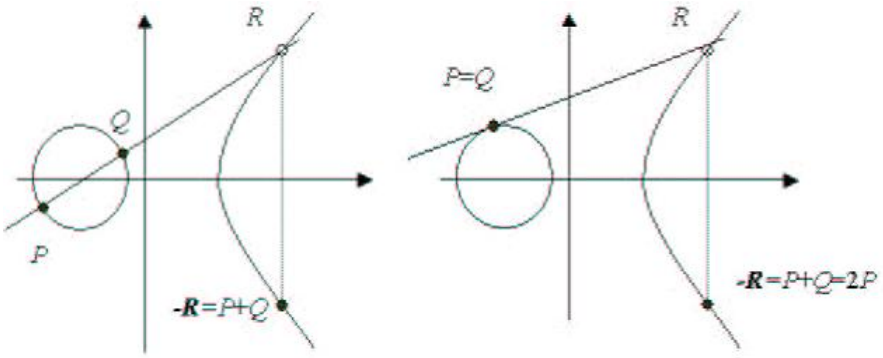

Рисунок 8.1 - Эллиптические кривые

В случае, если необходимо удвоить точку $P(x_1, y_1)$, найти $P+P$, то в этой точке проводится касательная к кривой. Результат удвоения – точка $-R(x_3, y_3)$, обратная точке R пересечения эллиптической кривой и касательной к точке P.

Координаты x_3, y_3 определяются по формулам:

$$x_3 = \lambda^2 - 2 \cdot x_1 (mod\, p), \quad y_3 = \lambda \cdot (x_1 - x_3) - y_1 (mod\, p) \tag{8.5}$$

где

$$\lambda = \frac{3 \cdot x_1^2 + a}{2 \cdot y_1} mod\, p$$

При вычислении координат используются правила модульной арифметики: Все действия выполняются по модулю p, операция деления числителя на знаменатель заменяется на операцию умножения числителя на число, обратное к знаменателю по модулю p, отрицательные результаты приводят к положительным последовательным сложением с модулем p. Например,

$$\frac{-6}{7} mod\ 20 = -6 \cdot 3\ mod\ 20 = 2$$

Число, обратное к 7 по модулю 20 равно 3, так как 7·3 *mod* 20 =1.

Пример операций над точками эллиптической кривой

Рассмотрим кривую E_7 (2,6):

$$y^2 = x^3 + 2 \cdot x + 6\ (mod\ 7)$$

.

Проверим условие

$$4 \cdot 2^3 + 27 \cdot 6^2\ (mod\ 7) = 3$$

.

Найдем точку (5,1) + (4,6).

$$\lambda = \frac{6-1}{4-5} = \frac{5}{6} = 5 \cdot 6^{-1} = 2 mod\ 7$$

$$x_3 = 2^2 - 5 - 4 = 2(mod\ 7),\ y_3 = 2(5-2) - 1 = 5(mod\, p)$$

Координаты полученной точки (2,5)

В результате вычислений должны получаться целочисленные значения. Число $x = 6^{-1}$, обратное к данному (6) по модулю 7 удовлетворяет условию $(x \cdot 6) mod\ 7 = 1$. Откуда x=6. Для вычисления мультипликативно обратного элемента для числа x можно вос- пользоваться расширенным алгоритмом Евклида или с помощью обобщения Эйлера для малой теоремы Ферма.

Алгоритм Диффи-Хеллмана на эллиптических кривых

Для установления защищенной связи два пользователя А и В совместно выбирают эллиптическую кривую E и точку $G(x,y)$ на ней. На первом этапе пользователь А выбирает свое секретное целое число k_1, вычисляет произведение $k_1 \cdot G$ и посылает результат абоненту В. Пользователь В генерирует свое секретное большое число k_2, вычисляет произведение $k_2 \cdot G$ и пересылает его получателю А.

При этом параметры самой кривой, координаты точки на ней и значения произведений являются открытыми и могут передаваться по незащищенным каналам связи. Предполагается, что злоумышленник может получить оба этих значения, но не модифицировать их.

На втором этапе абонент А на основе имеющегося у пользователя числа и полученного по сети значения вычисляет ключ $K= k_1 \cdot k_2 \cdot G$. Абонент В аналогично вычисляет значение $K= k_2 \cdot k_1 \cdot G$. В силу свойств операции умножения на число оба пользователя получат общее секретное значение (координаты точки), которое они могут использовать для получения ключа шифрования. Секретное значение представляет собой пару чисел, для получения ключа симметричного шифрования из пары получают одно значение.

Стойкость шифрования с помощью эллиптических кривых базируется на сложности нахождения множителя k точки P по их произведению.

Умножение точки на скаляр

Умножение точки на число реализуется последовательностью сложений и удвоений точки эллиптической кривой.

Вычисление m-кратной композиции точек ЭК.

Вход: точка P, число , представленное в двоичном виде $m=(m_t , m_{t-1} ,\dots m_1)$

Выход: $Q = [m]P$.

1. $Q = \mathbf{O}$.

2. Для каждого $i = t, t–1, \dots, 1$ выполнить

2.1. $Q= [2]Q$.

2.2. Если $m_i = 1$, то $Q= Q + P$.

3. Результат Q.

Данный алгоритм требует не более t сложений и t удвоений точек.

Пример

Работа алгоритма вычисления m-кратной композиции на примере вычисления точки $29P$. Здесь $29 = (11101)_2$, $t= 5$. На каждой итерации цикла алгоритма:

$[i = 5\ m_5 = 1] : Q \leftarrow O , \quad Q \leftarrow Q + P = P ;$

$[i = 4 m_4 = 1] : Q \leftarrow 2Q = 2P , Q \leftarrow Q + P = 3P ;$

$[i = 3 m_3 = 1] : Q \leftarrow 2Q = 6P , Q \leftarrow Q + P = 7P ;$

$[i = 2 m_2 = 0] : Q \leftarrow 2Q = 14P ;$

$[i = 1 m_1 = 1] : Q \leftarrow 2Q = 28P; \quad Q \leftarrow Q+P = 29P.$

Кратная точка вычислена с применением 5 умножений и 4 сложений точек.

Порядок выполнения работы

1. Выбрать коэффициенты a,b и модуль p эллиптической кривой, координаты x, y точки G, а также секретные значения k_1, k_2 абонентов из таблицы 8.1 в соответствии с вариантом.

3.Разработать программную реализацию метода Диффи-Хеллмана. Предусмотреть проверку эллиптической кривой по формуле (8.2). Исходными данными являются параметры кривой, координаты точки и секретные значения каждого участника обмена. Результат работы программы – координаты произведения точки G на число, которые должны совпасть у каждого из участников.

4. Оформить отчет.

Таблица 8.1 – Исходные данные протокола Диффи-Хеллмана

№ вар.	a	b	p	$G(x,y)$	k_1	k_2
1	-1	1	29	(9,27)	4	17
2	1	1	23	(7,11)	5	16
3	2	3	97	(3,6)	6	10
4	-1	3	37	(2,3)	7	12
5	3	5	17	(1,3)	8	11
6	1	1	23	(6,4)	8	9
7	1	0	23	(9,5)	10	8
8	9	17	23	(16,5)	11	7
9	2	3	97	(3,6)	12	6
10	-1	3	37	(2,3)	13	5
11	3	5	17	(1,3)	3	13
12	1	1	23	(6,4)	2	14
13	1	0	23	(9,5)	14	4
14	2	3	97	(3,6)	15	3
15	1	1	23	(3,13)	16	2

Содержание отчета:

- цель работы, постановка задачи,
- описание используемого метода,
- описание исходных данных,
- алгоритм работы программы,
- текст программы,
- результаты работы программы,
- анализ результатов
- выводы.

Контрольные вопросы

1. Цель применения протокола Диффи-Хеллмана.
2. Что представляет собой эллиптическая кривая?
3. Какие операции определены на эллиптической кривой при использовании в криптографических приложениях?
4. Как выполнить умножение точки эллиптической кривой на число?
5. Как вычислить число. Обратное к данному по заданному модулю?
6. Что является нулем эллиптической кривой?

Библиографический список

1. Адаменко, М.В. Основы классической криптологии: секреты шифров и кодов / М.В. Адаменко. - 2-е изд., испр. и доп. - Москва : ДМК Пресс, 2016. - 296 с. - ISBN 978-5-97060-166-2. - Режим доступа: http://znanium.com/catalog/product/1027822

2. Бабенко, Л. К. Криптографическая защита информации: симметричное шифрование : учеб. пособие для вузов / Л. К. Бабенко, Е. А. Ищукова. — Москва : Издательство Юрайт, 2019. — 220 с. — (Серия : Университеты России). — ISBN 978-5-9916-9244-1. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: https://biblio-online.ru/bcode/437667 (дата обращения: 24.04.2019).

3. Введение в криптографию. Курс лекций / В.А. Романьков. — 2-е изд., испр. и доп. — М. : ФОРУМ : ИНФРА-М, 2017. — 240 с. — (Высшее образование). - Режим доступа: http://znanium.com/catalog/product/914480.

4. Криптографическая защита информации : учеб. пособие / С.О. Крамаров, О.Ю. Митясова, С.В. Соколов [и др.]; под ред. проф. С.О. Крамарова. — М. : РИОР : ИНФРА-М, 2018. — 321 с. — (Высшее образование). — DOI: https://doi.org/10.12737/1716-6 - Режим доступа: http://znanium.com/catalog/product/901659Иванов М.А. Криптографические методы защиты информации в компьютерных системах и сетях./ М.А. Иванов.- М.: Кудиц-образ, 2001.-363с.

5. Рябко, Б.Я. Основы современной криптографии и стеганографии [Электронный ресурс] : монография / Б.Я. Рябко, А.Н. Фионов. — Электрон. дан. — Москва : Горячая линия-Телеком, 2016. — 232 с. — Режим доступа: https://e.lanbook.com/book/111098. — Загл. с экрана.

6. Швечкова, О.Г. Базовые криптофафические ачгоритмы зашиты информации : учеб. пособие / О.Г. Швечкова, А.Н. Пылькин, Д.В. Марчев. - М. : КУРС, 2018. - 168 с. - ISBN 978-5-906923-83-7. - Режим доступа: http://znanium.com/catalog/product/1016955

КРИПТОГРАФИЧЕСКИЕ МЕТОДЫ ЗАЩИТЫ ИНФОРМАЦИИ

Методические указания
к выполнению лабораторных работ

Составители: **Корепанова** Наталия Леонидовна,
Лебедева Марина Анатольевна

В авторской редакции

Изд. № 180/19. Объём 2 п.л.
РИИЦМ ФГАОУ ВО "Севастопольский государственный университет"