Міністерство освіти і науки, молоді та спорту України
Прикарпатський національний університет
імені Василя Стефаника
Фізико-технічний факультет
кафедра радіофізики і електроніки

Запухляк Р.І., Новосядлий С.П., Головатий Т.В.

ЗАХИСТ ІНФОРМАЦІЇ В КОМП'ЮТЕРНИХ СИСТЕМАХ

Методичні вказівки до виконання лабораторних робіт
для студентів спеціальності
«Комп'ютерна інженерія»

Івано-Франківськ
2012

УДК 681.322
ББК 32.973

Запухляк Р.І., Новосядлий С.П., Головатий Т.В. Захист інформації в комп’ютерних системах: методичні вказівки до виконання лабораторних робіт. – Прикарпатський національний університет імені Василя Стефаника, Івано-Франківськ, 2012 р. – 34 с.

Робоча програма дисципліни розроблена відповідно до вимог державних освітніх стандартів вищої професійної освіти з напряму підготовки дипломованого фахівця «Комп’ютерна інженерія».

Навчальний посібник містить методичні вказівки до виконання лабораторних робіт та завдання і приклади, що сприяють засвоєнню навчального матеріалу та дозволяють вирішувати проблеми, пов'язані із захистом інформації в комп'ютерних системах і мережах.

Розглянуто на засіданні кафедри радіофізики і електроніки (протокол № 7 від 28.03.2012); схвалено методичною комісією фізико-технічного факультету.

Затверджено вченою радою фізико-технічного факультету (протокол № 6 від 3 травня 2012 р.)

Рецензенти: Поплавський О.П., кандидат фізико-математичних наук, доцент.
Голота В.І., кандидат технічних наук, доцент;

ЗМІСТ

ЗАГАЛЬНІ ВКАЗІВКИ ДО ВИКОНАННЯ ЛАБОРАТОРНИХ РОБІТ

Ці методичні вказівки містять опис і порядок виконання лабораторних робіт з дисципліни «Захист інформації в комп'ютерних системах».

Метою лабораторних робіт є дослідження методів захисту інформації в комп'ютерних системах і мережах.

Завдання на виконання лабораторних робіт видаються викладачем. У лабораторії прийнятий індивідуальний метод виконання робіт. До чергової роботи студенти допускаються тільки після проходження співбесіди з основ захисту інформації, що стосуються відповідної лабораторної роботи.

Про готовність до роботи свідчать знання змісту роботи і основних теоретичних положень, що розглядаються в роботі. Звіти про виконану роботу мають бути складені технічно грамотно і закінчуватися самостійними висновками, оскільки студент повинен творчо підходити до отриманих результатів роботи, використовуючи свої практичні навички і теоретичні знання. Лабораторні роботи мають бути оформлені у вигляді звіту з вказівкою прізвища, ініціалів і шифру студента, теми та мети лабораторної роботи, теоретичних відомостей та результатів виконання самої лабораторної роботи, висновків. Перед захистом лабораторної роботи студент повинен здати оформлений звіт на перевірку викладачеві.

БІБЛІОГРАФІЧНИЙ СПИСОК

1. Галицкий А.В. Защита информации в сети - анализ технологий и синтез решений .-М.:ДМК Пресс,2004 .-616 с.
2. Гвоздева В.А. Правовая и программная защита компьютерной информации//Гвоздева Валентина Александровна.Введение в специальность программиста: Учебник..-М.:Форум,2005 .-С. 159-174.
3. Соколов А.В., Шаньгин В. Ф. Защита информации в распределенных корпоративных сетях и системах. –М.: ДМК Прес, 2002.- 656 с.
4. Роман Ю.В., Тимофеев П.А., Шаньгин В.Ф. Защита информации в компьютерных системах и сетях/ Под ред. В.Ф. Шаньгина. -М.: Радио и связь, 2001.-376 с.
5. Б. Шнайер. Прикладная криптография. Протоколы, алгоритмы, исходные тексты на языке Си. - М.: Издательство ТРИУМФ, 2002. - 816 с.
6. Петраков А.В. Основы практической защиты информации. -М.: Радио и связь, 2001. - 368 с.
7. Хоффман Л. Современные методы защиты информации. Пер. с англ. Казарова Е.А./ Л. Хоффман. –М. Сов. радио, 1980. –264 с.
8. Введение в криптографию: новые математические дисциплины. Учебник/под ред. В.В. Ященко. –СПб.: Питер, 2001. –287 с.
9. Иванов М.А. Криптографические методы защиты информации в компьютерных системах и сетях. –М.: Кудиц-образ, 2001. –363 с.
10. Анин Б. Защита компьютерной информации. –СПб.: Киев, М.: БВХ-Петербург, 2000. –368 с.

Лабораторна робота № 1

ШИФРУВАННЯ ДАНИХ МЕТОДАМИ ПІДСТАНОВКИ (ЗАМІНИ), ПЕРЕСТАНОВКИ І ПОЛІАБЕТНИМИ ШИФРАМИ

Мета роботи: придбання навичок шифрування інформації з використанням простих методів шифрування.

Основні теоретичні відомості

Проблемою захисту інформації шляхом її перетворення займається криптологія (kryptos – таємний, logos – наука). Криптологія розділяється на два напрями: криптографію і криптоаналіз.

Цілі цих напрямів прямо протилежні:

- криптографія займається пошуком і дослідженням математичних методів перетворення інформації.
- сфера інтересів криптоаналізу – дослідження можливості розшифрування інформації без знання ключів.

Термінологія. Криптографія дає можливість перетворити інформацію таким чином, що її прочитання (відновлення) можливе тільки при знанні ключа. В якості інформації, що підлягає шифруванню і дешифруванню, розглядаються тексти, побудовані на деякій абетці. Під цими термінами розуміється наступне:

Абетка – кінцева множина знаків, що використовується для кодування інформації.

Текст – впорядкований набір з елементів абетки. В якості прикладів абеток, що використовуються в сучасних інформаційних системах, можна навести наступні:

- абетка Z_{33} – 32 літери української абетки і пропуск;
- абетка Z_{256} – символи, що входять в стандартні коди ASCII і KOI-8;
- бінарна абетка – $Z_2=\{0,1\}$;
- вісімкова абетка або шістнадцяткова абетка.

Шифрування – процес перетворення: початковий текст, який носить також назву відкритого тексту, замінюється шифрованим текстом.

Дешифрування – зворотний шифруванню процес. На основі ключа шифрований текст перетвориться в початковий.

Ключ – інформація, необхідна для безперешкодного шифрування і дешифрування текстів.

Огляд використовуваних методів.

Метод підстановки (заміни)

Моноабеткова звичайна заміна Цезаря (шифр Цезаря). Як повідомляє історик Гай Светоній, римський імператор Гай Юлій Цезар користувався у своєму військовому та особистому листуванні шифром, сутність якого полягала у заміні кожної літери повідомлення на одну із інших літер 26-значного латинської абетки.

Щоб зрозуміти зашифроване повідомлення Цезаря, треба було кожну літеру в ній замінити третьою, що йде після неї в абетці. При досягненні кінця абетки виконувався циклічний перехід до його початку.

Такий метод шифрування можна відобразити шифрувальною таблицею, в якій вказано знаки заміни для кожного знаку криптограми. Використання таблиці очевидне: при шифруванні для кожного знаку відкритого тексту шукаємо відповідний знак криптограми, при дешифруванні – навпаки.

Отже, шифр Цезаря пов’язаний із використанням первинної абетки (для відкритого тексту) і вторинної абетки (для криптограми), циклічно зміщеної відносно

первинної на три знаки вперед.

Номер (код)	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
Знак Відкритого тексту	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
Знак криптограми	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W

Зокрема, Цезар використовував свій шифр у листуванні з Цицероном (близько 50 р. до н.е.). А його відоме послання VENI VIDI VICI – «Прийшов, побачив, переміг» своєму другові Амінтію (після перемоги над понтійським царем Фарнаком, сином Мітрідата при Золе в 47 р. до н.е.) у зашифрованому вигляді мало такий вигляд: SBKF SFAF SFZF.

Загальний випадок шифру Цезаря для первинної абетки деякої величини m полягає в тому, що вторинна абетка циклічно зміщується відносно первинної на K знаків вперед ($0 \le K < m$). Значення K є ключем цього шифру.

Наприклад, для абетки «АБВГДЕЖЗИК» об'ємом m=10 шифрувальна таблиця для K=6 має наступний вигляд:

Номер(код)	0	1	2	3	4	5	6	7	8	9
Знак відкритого тексту	А	Б	В	Г	Д	Е	Ж	З	И	К
Знак криптограми	Д	Е	Ж	З	И	К	А	Б	В	Г

Таким чином, відкритому тексту «КВИДА» відповідає криптограма «ГЖВИД» і навпаки.

Недоліки моноабеткової звичайної заміни Цезаря:

- вона не маскує частот появи різних знаків відкритого тексту і тому її легко зламати на підставі аналізу частот появи знаків у криптограмі;
- у вторинній абетці зберігається той же самий абетковий порядок знаків, що і впервинній;
- мала кількість можливих ключів (рівна величині абетки).

Метод перестановки

Метод перестановки, також нескладний метод криптографічного перетворення. Використовується, як правило, у поєднанні з іншими методами. При шифруванні цим методом переставляються не літери абетки, а літери відкритого тексту. Наприклад, повідомлення розбите на 4 групи знаків, включаючи пропуски, і в кожній групі літери переставлені відповідно до правила:

[1 2 3 4]
[2 4 1 3]

В цьому випадку фраза:

ОСНОВИ ЗАХИСТУ ІНФОРМАЦІЇ

буде представлена в наступному виді:

СООНИЗВ ХСАИУІТ ФРНОАІМЦЇ.

У випадку перестановки, таблиці частот для двох і трьох літер, показують наявність стандартних літерних пар, дозволяючи реконструювати відкритий текст шляхом пошуку тих перестановок, які їх поєднують. Отже, ключ, що використовується для перетворення відкритого тексту, може бути відновлений за одною криптограмою.

Багатоабеткові шифри

Слабка криптостійкість моноабеткових підстановок (замін) долається з

застосуванням багатоабеткових підстановок (замін). Для захисту від частотного аналізу були розроблені багатоабеткові шифри, в яких для шифрування повідомлення періодично використовується декілька різних абеток підстановки. При шифруванні інформації, букви з номерами 4N+i шифруються i-ою абеткою (1,5,9,13, ... - 1 абеткою, 2,7,10,14, ... – 2 абеткою, і так далі).

Для отримання відкритого тексту виділяються групи знаків, що повторюються, і визначається період повторення. Передбачуваний період перевіряється складанням частотного розподілу для кожної n-ї літери зашифрованого тексту. Якщо кожен з n частотних розподілів має сильну неоднорідність, характерну для багатоабеткової підстановки, то передбачуваний період є правильним. Потім завдання вирішується як n різних простих підстановок.

Методика виконання роботи

1. Розробити алгоритм і скласти програму, що дозволяє закодувати будь-який текст одним з вищевикладених методів і виконати зворотне перетворення. Метод, яким необхідно зашифрувати початкову інформацію, вибирається відповідно до варіанту з таблиць 1,2,3. Мова програмування вибирається довільно.
2. Здійснити вивід на екран або принтер отриману криптограму.
3. Провести дешифрування цієї криптограми, в результаті має бути отриманий початковий текст.
4. Результати роботи оформити у вигляді звіту та захистити його.

Зміст звіту

1. Опис методу, що використовується. Опис початкових даних.
2. Алгоритм роботи програми (блок-схема), текст програми, результати роботи програми.
3. Аналіз результатів роботи, висновки.

Варіанти індивідуальних завдань

Таблиця 1. Методи шифрування

Номер вар.	Метод шифрування	Таблиця	Номер завдання в таблиці	Представлення початкового тексту
1	Підстановка	2	3	Англійська абетка
2	Перестановка	3	1	ASCII -код
3	Багатоабеткові шифри	2	1, 2, 5	Українська абетка
4	Перестановка	3	2	Українська абетка
5	Підстановка	2	4	ASCII -код
6	Багатоабеткові шифри	2	1, 3	Українська абетка
7	Підстановка	2	1	ASCII -код
8	Багатоабеткові шифри	2	2, 3, 4, 5	Англійська абетка
9	Перестановка	3	3	Англійська абетка
10	Підстановка	2	2	Українська абетка
11	Перестановка	3	4	Англійська абетка
12	Багатоабеткові шифри	2	1, 3, 4	Українська абетка
13	Підстановка	2	5	Англійська абетка
14	Багатоабеткові шифри	2	1, 5	Українська абетка
15	Перестановка	3	5	ASCII -код

Таблиця 2. Абетки підстановки

Номер з/п	Вихідна абетка		Абетка підстановки (вар.)									
			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		Я	D	Н	W
3	В	C	Г	X	У	A	М	.	И	A	О	X
4	Г	D	И	Y	М	B	Н	X	Е	B	П	Y
5	Ґ	E	Є	Z	К	H	Х	Y	Ь	H	Р	Z
6	Д	F	Ь		Х	I	Л	,	Ґ	I	С	
7	Е	G	З	.	Ч	J	И	!	Ш	J	Т	.
8	Є	H	Ш	,	I	E	I	S	Щ	E	У	,
9	Ж	I	И	!	Щ	F	Ж	T	Ц	F	Ф	!
10	З	J	Ц	:	Ж	G	З	:	Ч	G	Х	:
11	И	K	Л	;	Ґ	O	Д	;	Ф	O	Ц	;
12	І	L	Ф	?	Д	P	Е	Q	Х	P	Ч	?
13	Ї	M	Н	-	Э	Q	В	R	Т	Q	Ш	-
14	Й	N	Т	K	В	R	Г	?	У	R	Щ	K
15	К	O	П	L	Я	K	А	-	Р	K	Ґ	L
16	Л	P	Р	M	А	L	Б	N	С	L	Ь	M
17	М	Q	С	N	Б	M	Ю	O	О	M	И	N
18	Н	R	О	O	Ю	N	Я	P	П	N	Е	O
19	О	S	У	P	Г	U	Є	L	М	U	Ю	P
20	П	T	М	Q		V	Е	M	Н	V	Я	Q
21	Р	U	Х	R	Е	W	Ь	U	К	W		R
22	С	V	К	S	Ь	:		V	Л	:	А	S
23	Т	W	Ч	T	З	S	Ш	W		S	Б	T
24	У	X	І	U	Ш	T	Щ	A	Й	T	В	U
25	Ф	Y	Щ	A	І	Z	Ц	B	Ж	Z	Г	A
26	Х	Z	Ж	B	Ц		Ч	C	З		Д	B
27	Ц		Ґ	C	Ї	X	Ф	D	Д	X	Є	C
28	Ч	.	Д	D	Ф	Y	К	E	Є	Y	Ї	D
29	Ш	,	Е	E	Н	;	Т	F	В	;	Ж	E
30	Щ	!	В	F	Т	?	У	G	Г	?	З	F
31	Ь	:	Я	G	П	-	Р	H	А	-	І	G
32	Ю	;		H	Р	.	С	I	Б	.	Й	H
33	Я	?	А	I	И	,	Й	J	Ї	,	К	I
34		-	Ї	J	Л	!	Ї	K	І	!	Л	J

Таблиця 3. Групи перестановок.

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

Контрольні запитання

1. Чому метод підстановки має слабку надійність?
2. Що таке частотний аналіз?
3. Що є криптографічним ключем в методі перестановки?
4. Як пов'язані метод підстановки і багатоабеткові шифри?

Лабораторна робота №2

ШИФРУВАННЯ ДАНИХ ЗА ДОПОМОГОЮ ГЕНЕРАТОРА ПСЕВДОВИПАДКОВИХ ЧИСЕЛ

Мета роботи: освоєння принципів шифрування гамуванням, вивчення властивостей генератора псевдовипадкових чисел, програмна реалізація методу гамування.

Основні теоретичні відомості

Принцип шифрування гамуванням полягає в генерації гами шифру за допомогою датчика псевдовипадкових чисел і накладенні отриманої гами шифру на відкриті дані оборотним чином (наприклад, використовуючи операцію додавання за модулем 2). Процес дешифрування зводиться до повторної генерації гами шифру при відомому ключі і накладенні такої ж гамми на зашифровані дані.

Отриманий зашифрований текст є досить важким для розкриття у тому випадку, якщо гама шифру не містить бітових послідовностей, що повторюються і змінюється випадковим чином для кожного шифрованого слова. Якщо період гами перевищує довжину всього зашифрованого тексту і невідома жодна частина початкового тексту, то шифр можна розкрити тільки прямим перебором (підбором ключа). В цьому випадку криптостійкість визначається розміром ключа.

Метод гамування стає безсилим, якщо відомий фрагмент вихідного тексту і шифрограма, що відповідає йому. В цьому випадку простим відніманням за модулем 2 виходить відрізок псевдовипадкової послідовності і по ньому відновлюється вся ця послідовність.

Шифрування даних за допомогою датчика псевдовипадкових чисел (ПВЧ)

Лінійні конгруентні датчики ПВЧ

Щоб отримати лінійні послідовності елементів гами, довжина яких не перевищує розмір шифрованих даних, використовують датчики ПВЧ. Одним з хороших конгруентних генераторів є лінійний конгруентний датчик ПВЧ. Він виробляє послідовності псевдовипадкових чисел T(i), що описуються співвідношенням

$$T(i+1) = (A * T(i) + C) \bmod M,$$

де A і C – константи, T(0) – початкова величина, що вибрана в якості генеруючого числа. Очевидно, що ці три величини і утворюють ключ.

Такий датчик ПВЧ генерує псевдовипадкові числа з визначеним періодом повторення, що залежить від вибраних значень A і C. Значення M зазвичай встановлюється рівним 2^b, де b – довжина машинного слова у бітах. Необхідно вибирати числа A і C так, щоб період M був максимальним.

Як показано Д. Кнуттом, лінійний конгруентний датчик має максимальну довжину M тоді, коли C непарне і A mod 4 = 1. Ще одними підходом до вибору відповідних чисел є: числа A і M та C і M мають бути взаємно простими (НСД(A,M)=1, НСД(C,M)=1), а $0 \le T(i) < M$.

Як приклад використання лінійного конгруентного датчика ПВЧ розглянемо процес шифрування початкового тексту "абв". Нехай b = 5, тоді у відповідності з номером у абетці: літера "а" має двійковий код 00001; літера "б" має двійковий код 00010; літера "в" має двійковий код 00011. Початковий текст буде представлений у вигляді послідовності 00001 00010 00011.

Для формування гами шифру виберемо параметри датчика ПВЧ : A=5; C=3; T(0)=7; $M=2^b$; b=5; $M=2^5=32$. Сформуємо три псевдовипадкові 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).

Отримана гама шифру 00110 00001 01000. Зашифрований текст виходить шляхом накладення гами шифру на початковий текст (шляхом додавання за модулем 2):

00001 00010 00011
<u>00110 00001 01000</u>
00111 00011 01011

що відповідає шифрограмі "еви", "е" (сьома буква в абетці) має код 00111, "в" (третя буква в абетці) має код 00011, "и" (одинадцята буква в абетці) має код 01011.

Дешифрування проводиться шляхом накладення тієї ж гамми на зашифрований текст:

00111 00011 01011
<u>00110 00001 01000</u>
00001 00010 00011

Метод гамування із зворотним зв'язком

Полягає в тому, що для отримання сегменту гами використовується контрольна сума певної ділянки шифрованих даних. Наприклад, якщо розглядати гаму шифру як об'єднання множин H(j), що не перетинаються, то процес шифрування можна представити наступними кроками:

1. Генерація сегменту гами H(1) і накладення його на відповідну ділянку шифрованих даних.
2. Підрахунок контрольної суми ділянки, що відповідає сегменту гами H(1).
3. Генерація з врахуванням контрольної суми вже зашифрованої ділянки даних наступного сегменту гам H(2).
4. Підрахунок контрольної суми ділянки даних, що відповідає сегменту даних H(2) і так далі.

Під контрольною сумою розуміють функцію f(t(1), ..., t(n)), де t(i) – і-е слово шифрованих даних.

Зашифруємо початковий текст "абв", представлений у вигляді послідовності 00001 00010 00011. Нехай A=5; C=3; b=5; M=32;T(0)=7. Тоді:

T(1) = (5*7+3) mod 32 = 6 (00110).

В якості контрольної суми ділянки даних, виберемо кількість одиниць на цій ділянці. Тоді сегменту H(1) відповідає ділянка 00010, кількість одиниць дорівнює 1.

T(2) = (5*1+3) mod 32 = 8 (01000).

Контрольна сума наступної ділянки (00010) дорівнює 1.

T(3) = (5*1+3) mod 32 = 8 (01000).

Отримана шифрограма:

00001 00010 00011
<u>00110 01000 01000</u>
00111 01010 01011

що відповідає тексту "ези".

Методика виконання роботи

1. Вибрати в таблиці параметри генератора псевдовипадкових чисел: A, C, T(0), b.
2. Розробити програму шифрування і дешифрування тексту.

3. Провести шифрування початкового тексту, отримати шифрограму, виконати її дешифрування і порівняння з початковим текстом.
4. Провести зміну одного або декілька параметрів генератора випадкових чисел, здійснити отримання шифрограми і порівняння її з попереднім варіантом.
5. Результати роботи оформити у вигляді звіту.

Зміст звіту

1. Опис використовуваного методу, опис початкових даних.
2. Алгоритм роботи програми, текст програми, результати роботи програми.
3. Аналіз результатів, висновки.

Варіанти індивідуальних завдань

Таблиця 1. Генератори ПВЧ

№ варіанту	Вид генератора ПВЧ	Кількість розрядів b
1	Лінійні конгруентні датчики ПВЧ	6
2	Гамування із зворотним зв'язком	7
3	Лінійні конгруентні датчики ПВЧ	8
4	Гамування із зворотним зв'язком	6
5	Лінійні конгруентні датчики ПВЧ	7
6	Гамування із зворотним зв'язком	8
7	Лінійні конгруентні датчики ПВЧ	6
8	Гамування із зворотним зв'язком	7
9	Лінійні конгруентні датчики ПВЧ	8
10	Гамування із зворотним зв'язком	6
11	Лінійні конгруентні датчики ПВЧ	7
12	Гамування із зворотним зв'язком	8
13	Лінійні конгруентні датчики ПВЧ	6
14	Гамування із зворотним зв'язком	7
15	Лінійні конгруентні датчики ПВЧ	8
16	Гамування із зворотним зв'язком	6
17	Лінійні конгруентні датчики ПВЧ	7
18	Гамування із зворотним зв'язком	8
19	Лінійні конгруентні датчики ПВЧ	6
20	Гамування із зворотним зв'язком	7
21	Лінійні конгруентні датчики ПВЧ	8
22	Гамування із зворотним зв'язком	6
23	Лінійні конгруентні датчики ПВЧ	7
24	Гамування із зворотним зв'язком	8

Контрольні запитання

1. Які параметри конгруентного генератора необхідно вибрати для отримання максимальної довжини послідовності псевдовипадкових чисел?
2. Від чого залежить довжина псевдовипадкової послідовності?

3. Принцип дії генераторів із зворотним зв'язком?
4. Які стандарти використовують методи гамування?

Лабораторна робота № 3

МЕРЕЖІ ФЕЙШТЕЛЯ

Мета роботи: вивчити принципи роботи мережі Фейштеля, навчитися шифрувати інформацію за допомогою використання блокового криптоалгоритму.

Основні теоретичні відомості

Основи криптоалгоритмів на базі мережі Фейштеля

Мережа Фейштеля отримала широке поширення, оскільки забезпечує виконання вимоги про багатократне використання ключа і матеріалу вихідного блоку інформації. Класична мережа Фейштеля має наступну структуру:

Незалежні потоки інформації, породжені з початкового блоку, називаються вітками мережі. У класичній схемі їх дві. Величини V_i називаються параметрами мережі, звичайно це функції від матеріалу ключа. Функція F називається твірною. Дія, що складається з одноразового обчислення твірної функції, і подальшого накладення її результату на іншу вітку з обміном їх місцями, називається циклом або раундом (англ. round) мережі Фейштеля. Оптимальне число раундів K - від 8 до 32. Часто кількість раундів не фіксується розробниками алгоритму, а лише вказуються розумні межі (обов'язково нижній, і не завжди – верхній) цього параметра.

Ця схема є оборотною. Мережа Фейштеля має ту властивість, що навіть якщо в якості твірної функції F буде використане безповоротне перетворення, то і в цьому випадку увесь ланцюжок буде відновлюваний. Це відбувається внаслідок того, що для зворотного перетворення мережі Фейштеля не потрібне обчислення функції F^{-1}.

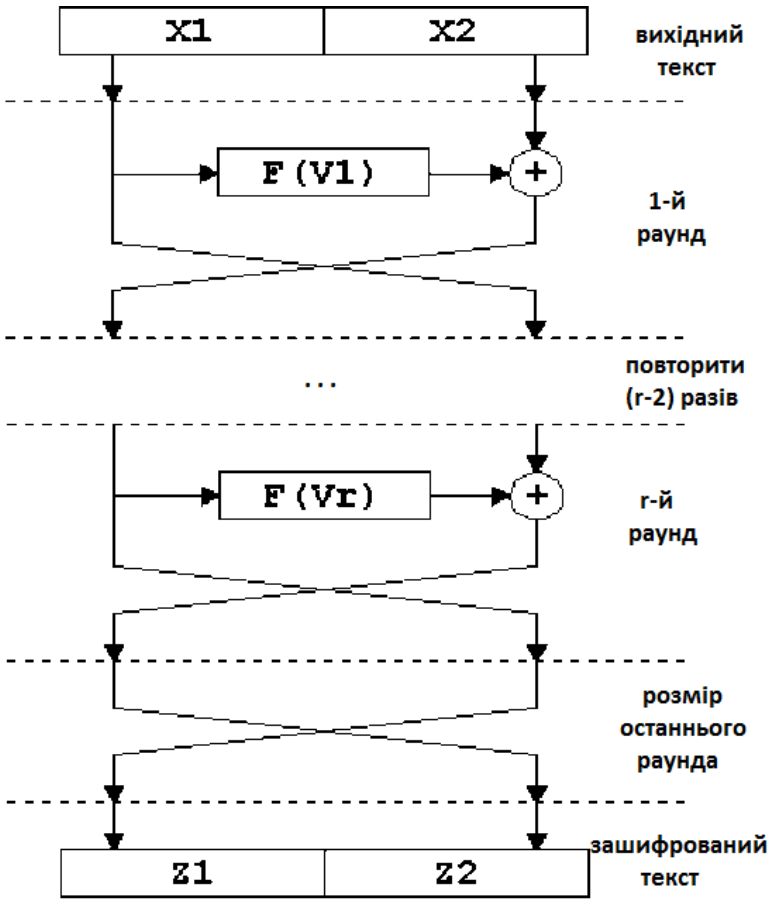

Рис. 1. Класична структура мережі Фейштеля.

Мережа Фейштеля симетрична за рахунок використання операції XOR і для її оборотності не має значення чи є число раундів парним або непарним числом.

Використання модифікації мережі Фейштеля для більшого числа віток пов'язане з тим, що при великих розмірах кодованих блоків (128 і більше біт) стає незручно працювати з математичними функціями за модулем 64 і вище. Основні одиниці інформації, що обробляються процесорами на сьогодні – це байт і подвійне машинне слово 32 біта. Буде логічно розбивати початкові блоки не на дві, а на 4 частини. В цьому випадку мережа Фейштеля може набирати такого вигляду:

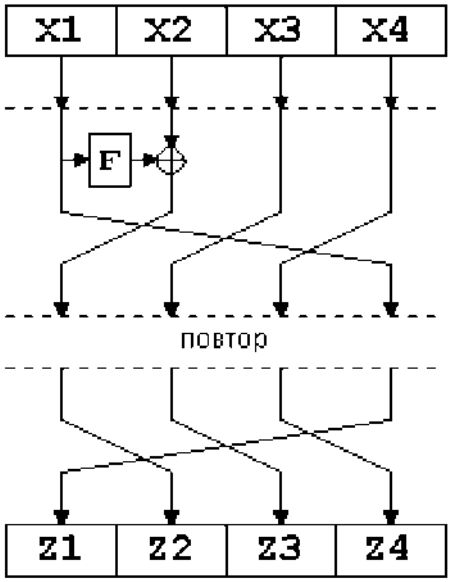

Рис. 2. Структура модифікованої мережі Фейштеля.

Алгоритм призначений для шифрування і дешифрування інформації, що представляється у вигляді слів, розрядністю 128 біт на основі 64-бітового ключа. Операції шифрування і дешифрування є інверсними і використовують один і той же ключ.

Розглянемо шифрування одного блоку.

Позначимо X1X2X3X4 конкатенацію послідовностей X1, X2, X3 і X4, в якій біти послідовностей X1, X2, X3, X4 слідують один за одним. Розмірність послідовності дорівнює сумі розмірностей всіх складових. Символом «+» позначимо операцію побітового складання за модулем 2.

Ітеративний процес шифрування описується наступними формулами:

$X1(i) = X2(i-1)+F(V_i)$, i = 1, 2, ..., n;

X2(i) = X3(i-1), i = 1, 2, ..., n;

X3(i) = X4(i-1), i = 1, 2, ..., n;

$X4(i) = X1(i-1)$, i = 1, 2, ..., n; де $F(V_i)$ – твірна функція;

n – кількість раундів, може змінюватися, залежно від вимог з швидкодії і криптостійкості (n= 8 - 128);

V_i =X1(i-1)+h(K) – параметр мережі;

h(K)= K1 ROL i + K2 ROR i,

K1 і K2 – ліва і права частині ключа K,

ROL і ROR – операції циклічного зсуву вліво і вправо відповідно.

Пропонований алгоритм має ряд достоїнств. В першу чергу – простота реалізації і висока швидкодія, яка досягається за рахунок використання операцій, що мають високу швидкість виконання.

Дешифрування блоку інформації проводиться тією ж мережею Фейштеля, але з інверсним порядком параметрів мережі. У явному виді ключ в алгоритмі не використовується, що підвищує його криптостійкість. При знанні ключа, але відсутності інформації про кількість раундів криптоаналітику буде досить складно дешифрувати зашифровану інформацію.

Методика виконання роботи

1. Вибрати в таблиці параметри для мережі Фейштеля
2. Розробити програму шифрування і дешифрування тексту.
3. Провести шифрування початкового тексту.
4. Виконати дешифрування отриманої шифрограми і порівняти результат з початковим текстом.
5. Результати роботи оформити у вигляді звіту.

Зміст звіту

1. Опис використовуваного методу, опис початкових даних.
2. Алгоритм роботи програми, текст програми, результати роботи програми.
3. Аналіз результатів, висновки.

Варіанти індивідуальних завдань

Варіанти завдань представлені в таблиці 1, номер варіанту вибирається відповідно до номера студента в списку групи.

Таблиця 1. Параметри мережі Фейштеля.

Номер вар.	Кількість раундів	Твірна функція
1	8	Додавання
2	10	Виключаюче АБО
3	12	Множення за модулем 2^N+1
4	14	Множення за модулем 2^N
5	10	Арифметичний зсув вправо
6	18	Арифметичний зсув вліво
7	20	Додавання
8	8	Множення за модулем 2^N+1
9	24	Виключаюче АБО
10	20	Додавання
11	18	Множення за модулем 2^N+1
12	28	Виключаюче АБО
13	12	Додавання
14	14	Додавання
15	24	Виключаюче АБО
16	22	Додавання
17	8	Додавання
18	10	Множення за модулем 2^N
19	22	Виключаюче АБО
20	14	Додавання

Контрольні запитання

1. Представте класичну структуру мережі Фейштеля.
2. Що називається раундом в мережі Фейштеля?
3. Які властивості має мережа Фейштеля?
4. Яким чином використовується матеріал ключа при шифруванні?

Лабораторна робота №4

ДОСЛІДЖЕННЯ КРИПТОАЛГОРИТМУ ШИФРУВАННЯ RSA

Мета роботи: дослідження структури алгоритму і методики практичної реалізації криптосистеми шифрування RSA.

Основні теоретичні відомості

Як відомо, алгоритми симетричного шифрування використовують ключі відносно невеликої довжини і тому можуть швидко шифрувати великі об'єми даних.

При використанні алгоритму симетричного шифрування відправник і одержувач застосовують для шифрування і розшифрування даних один і той же секретний ключ. Таким чином, алгоритми симетричного шифрування ґрунтуються на припущенні про те, що зашифроване повідомлення не зможе прочитати ніхто, крім того хто має ключ для його розшифрування. При цьому, якщо ключ не скомпрометований, то при розшифруванні автоматично виконується аутентифікація відправника, оскільки тільки він має ключ, за допомогою якого можна зашифрувати повідомлення. Таким чином, для симетричних криптосистем актуальна проблема безпечного розподілу симетричних секретних ключів. У зв'язку з цим, без ефективної організації захищеного розподілу ключів, використання звичайної системи симетричного шифрування в обчислювальних мережах практично неможливе.

Вирішенням цієї проблеми є використання асиметричних алгоритмів шифрування, що називаються криптосистемами з відкритим ключем. У них для шифрування даних використовується один ключ, що називається «відкритим», а для розшифрування, - інший, що називається «закритим або секретним». Слід мати на увазі, що ключ розшифрування не може бути визначений з ключа шифрування.

У асиметричних криптосистемах відритий ключ і криптограма можуть бути відправлені по незахищених каналах. Концепція таких систем заснована на застосуванні однонапрямлених функцій.

Як приклад однонапрямленої функції може служити цілочисельне множення. Пряме завдання – обчислення добутку двох великих цілих чисел p і q, n=p*q. Це відносно нескладне завдання для ЕОМ.

Зворотне завдання – факторизація або розкладання на множники великого цілого числа практично нерозв'язна при досить великих значеннях n.

Наприклад, якщо $p \approx q$, а їх добуток $n \approx 2^{664}$, то для розкладання цього числа на множники знадобиться 2^{23} операцій, що практично неможливо виконати за прийнятний час на сучасних ЕОМ.

Іншим прикладом однонапрямленої функції є модульна експонента з фіксованою основою і модулем.

Наприклад, якщо $y = a^x$, то природно можна записати, що $x = \log_a(y)$.

Завдання дискретного логарифмування формулюється таким чином. Для відомих цілих *a*, *n*, *y* слід знайти таке число *x*, при якому $a^x(mod\ n) = y$. Наприклад, якщо $a=2^{664}$ і $n=2^{664}$ знаходження показника степеня *x* для відомого *y* вимагає близько 10^{26} операцій, що також неможливо виконати на сучасних ЕОМ.

У зв'язку з тим, що на даний час не вдалося довести, що не існує ефективного алгоритму обчислення дискретного логарифма за прийнятний час, то модульна експонента також умовно віднесена до однонапрямлених функцій.

Іншим важливим класом функцій, що використовуються при побудові криптосистем з відкритим ключем є, так звані, однонапрямлені функції з секретом. Функція відноситься до цього класу за умови, що вона є однонапрямленою і, крім того,

можливе ефективне обчислення зворотної функції, якщо відомий секрет.

У цій лабораторній роботі досліджується криптосистема RSA, що використовує модульну експоненту з фіксованим модулем і показником степеня (тобто однонапрямлену функцію з секретом).

Методика виконання роботи

Завдання на виконання лабораторної роботи видається викладачем після проходження студентами співбесіди з основ криптосистем з відкритим ключем.

Порядок виконання роботи відповідає, наведеній нижче, криптосистемі шифрування даних за схемою RSA.

Схема алгоритму шифрування даних RSA
Визначення відкритого «e» і секретного «d» ключів

1. Вибір двох взаємно простих великих чисел p і q.
2. Визначення їх добутку: n=p*q.
3. Визначення функції Ейлера: $\varphi(n) = (p-1)(q-1)$.
4. Вибір відкритого ключа **e** з врахуванням умов:
$$1 < e < \varphi(n), \text{ НСД } (e, \varphi(n)) = 1$$
5. Визначення секретного ключа d, що задовольняє умові
$$e \cdot d \equiv 1(mod\ \varphi(n)), \text{ де } d<n$$

Алгоритм шифрування повідомлення M (дії відправника)

1. Розбиває початковий текст повідомлення на блоки M_1, M_2,..., M_n.
(M_i = 0,1, 2,..., n)
2. Шифрує текст повідомлення у вигляді послідовності блоків:
$$C_i = M_i^e (mod\ n)$$
3. Відправляє одержувачеві криптограму: C_1, C_2,..., C_n.
4. Одержувач розшифровує криптограму за допомогою секретного ключа d за формулою: $M_i = C_i^d (mod\ n)$.

Процедуру шифрування даних розглянемо на наступному прикладі (для простоти і зручності розрахунків в цьому прикладі використані числа малої розрядності):

1. Вибираємо два прості числа p і q, p=3, q=11;
2. Визначаємо їх добуток (модуль) n=p*q=33;
3. Обчислюємо значення функції Ейлера $\varphi(n) = (p-1)(q-1)$
$$\varphi(n) = 2 \cdot 10 = 20$$
4. Вибираємо випадковим чином відкритий ключ з урахуванням виконання умов $1 < e \leq \varphi(n)$ і $\text{НСД}(e, \varphi(n)) = 1, e = 7$;
5. Обчислюємо значення секретного ключа d, що задовольняє умові $e \cdot d \equiv 1(mod\ \varphi(n))$, $7 \cdot d \equiv 1(mod\ 20)$; $d = 3$;
6. Відправляємо одержувачеві пару чисел (n=33, d=3);

Представляємо шифроване повідомлення M як послідовність цілих чисел 312.

7. Розбиваємо початкове повідомлення на блоки M_1=3, M_2=1, M_3=2;
8. Шифруємо текст повідомлення, представлений у вигляді послідовності блоків: $C_i = M_i^e (mod\ n)$:
$C_1 = 3^7 (\text{mod } 33) = 2187 (\text{mod } 33) = 9$,
$C_2 = 1^7 (\text{mod } 33) = 1 (\text{mod } 33) = 1$,

$C_3=2^7(\text{mod } 33)=128(\text{mod } 33)=29.$

9. Відправляємо криптограму $C_1=9$, $C_2=1$, $C_3=29$.
10. Одержувач розшифровує криптограму за допомогою секретного ключа d за формулою: $M_i = C_i^d(mod\ n)$.

$M_1=9^3 \text{ (mod } 33) = 729 \text{ (mod } 33) = 3$

$M_2 = 1^3 \text{ (mod } 33) = 1 \text{ (mod } 33) = 1$

$M_3 = 29^3 \text{ (mod } 33) = 24389 \text{ (mod } 33) = 2.$

Отримана послідовність чисел 312 є початковим повідомленням M.

Зміст звіту

1. Скласти блок-схему і програму алгоритму шифрування RSA.
2. Лістинг програми шифрування заданого повідомлення M з використанням алгоритму RSA.
3. Висновки: переваги і недоліки алгоритму шифрування RSA.

Лабораторна робота №5

ДОСЛІДЖЕННЯ ЕЛЕКТРОННОГО ЦИФРОВОГО ПІДПИСУ (ЕЦП) RSA

Мета роботи: дослідження структури алгоритму і методики практичної реалізації (ЕЦП) RSA.

Основні теоретичні відомості

Технологія застосування системи ЕЦП припускає наявність мережі абонентів, що обмінюються підписаними електронними документами. При обміні електронними документами по мережі значно знижуються витрати, пов'язані з їх обробкою, зберіганням і пошуком.

Одночасно при цьому виникає проблема, як аутентифікації автора електронного документу, так і самого документу, тобто встановлення достовірності автора і відсутності змін в отриманому електронному повідомленні.

У алгоритмах ЕЦП як і в асиметричних системах шифрування використовуються однонапрямлені функції. ЕЦП використовується для аутентифікації текстів, що передаються по телекомунікаційних каналах.

ЕЦП є відносно невеликим об'ємом додаткової цифрової інформації, що передається разом з підписаним текстом.

Концепція формування ЕЦП базується на оборотності асиметричних шифрів, а також на взаємозв'язку вмісту повідомлення, самого підпису і пари ключів. Зміна хоч би одного з цих елементів зробить неможливим підтвердження достовірності підпису, який реалізується за допомогою асиметричних алгоритмів шифрування і хеш-функцій.

Система ЕЦП включає дві процедури:

- формування цифрового підпису;
- перевірку цифрового підпису.

У процедурі формування підпису використовується секретний ключ відправника повідомлення, в процедурі перевірки підпису – відкритий ключ відправника.

Безпека системи RSA визначається обчислювальною трудністю розкладання на множники великих цілих чисел. Недоліком алгоритму цифрового підпису RSA є вразливість її до мультиплікативної атаки. Іншими словами, алгоритм ЕЦП RSA дозволяє хакерові без знання секретного ключа сформувати підписи під тими документами, в яких результат хешування можна обчислити як добуток результату хешування вже підписаних документів.

Узагальнена схема формування і перевірки електронного цифрового підпису приведена на рис.1.

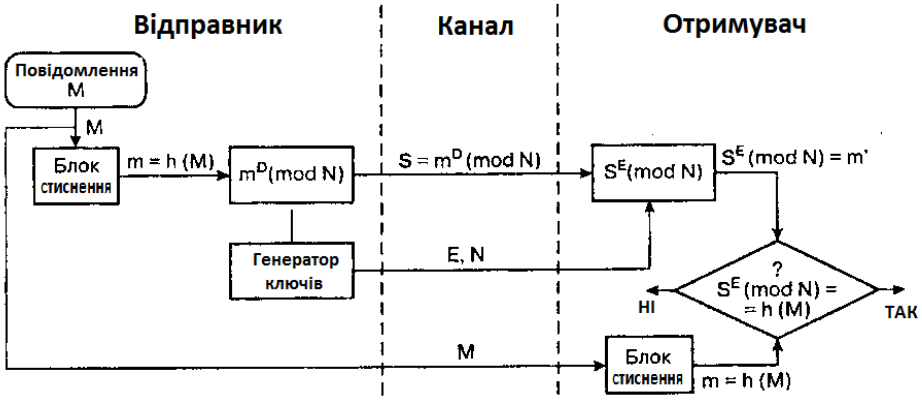

Рис. 1. Схема електронного цифрового підпису RSA.

Методика виконання роботи

Алгоритм електронного цифрового підпису (ЕЦП) RSA

Визначення відкритого «e» і секретного « d» ключів (дії відправника)

1. Вибір двох взаємно простих великих чисел p і q.
2. Визначення їх добутку $n = p \cdot q$.
3. Визначення функції Ейлера: $\varphi(n) = (p-1)(q-1)$.
4. Вибір секретного ключа d з врахуванням умов: $1 < d \leq \varphi(n)$, $\text{НСД}\big(d, \varphi(n)\big) = 1$.
5. Визначення значення відкритого ключа e: e< n,

$$e \cdot d \equiv 1 (mod\ \varphi(n)).$$

Формування ЕЦП

1. Обчислення хеш-значення повідомлення M: m=h(M).
2. Для отримання ЕЦП шифруємо хеш-значення m за допомогою секретного ключа d і відправляємо одержувачеві цифровий підпис $S = m^d (mod\ n)$ і відкритий текст повідомлення M.

Аутентифікація повідомлення – перевірка достовірності підпису

1. Розшифрування цифрового підпису S за допомогою відкритого ключа e і обчислення її хеш-значення $m' = S^e (mod\ n)$.
2. Обчислення хеш-значення прийнятого відкритого тексту M

m=h(M).

3. Порівняння хеш-значень m і m', якщо m=m', то цифровий підпис S – достовірний.

Завдання на виконання лабораторної роботи видається викладачем після проходження студентами співбесіди з основ аутентифікації даних і концепції формування електронного цифрового підпису.

Порядок виконання роботи відповідає, приведеному вище алгоритму формування ЕЦП за схемою RSA.

Процедуру формування ЕЦП повідомлення M розглянемо на наступному простому прикладі:

1. Обчислення хеш-значення повідомлення M: m=h(M).

Хешуюче повідомлення M представимо як послідовність цілих чисел 312. Відповідно до приведеного вище алгоритму формування ЕЦП RSA вибираємо два взаємно простих числа p=3, q=11, обчислюємо значення $n = p \cdot q = 3 \cdot 11 = 33$, вибираємо значення секретного ключа d=7 і обчислюємо значення відкритого ключа e=3. Вектор ініціалізації H_0 вибираємо рівним 6 (вибирається випадковим чином).

Хеш-код повідомлення M=312 формується таким чином:

$H_1 = (M_1 + H_0)^2 \pmod n = (3 + 6)^2 \pmod{33} = 81 \pmod{33} = 15;$

$H_2 = (M_2 + H_1)^2 \pmod n = (1 + 15)^2 \pmod{33} = 256 \pmod{33} = 25;$

$H_3 = (M_3 + H_2)^2 \pmod n = (2 + 25)^2 \pmod{33} = 729 \pmod{33} = 3;$ m = 3

2. Для отримання ЕЦП шифруємо хеш-значення m за допомогою секретного ключа d і відправляємо одержувачеві цифровий підпис

$S = m^d\ (mod\ n)$ і відкритий текст повідомлення M,

$S = 3^7 (mod\ 33) = 2187\ (mod\ 33) = 9$

3. Перевірка достовірності ЕЦП

Розшифрування S (тобто обчислення її хеш-значення m') проводиться за допомогою відкритого ключа e.

$$m' = S^e (mod\ n) = 9^3 (mod\ 33) = 729\ (mod\ 33) = 3$$

4. Якщо порівняння хеш-значень m' і m показує їх рівність, тобто m = m', то підпис достовірний.

Зміст звіту

1. Скласти блок-схему алгоритму і програму формування ЕЦП RSA.
2. Лістинг програми розрахунку ЕЦП RSA відповідно до завдання.
3. Висновки переваги і недоліки ЕЦП RSA.

Лабораторна робота № 6

ДОСЛІДЖЕННЯ КРИПТОАЛГОРИТМУ ШИФРУВАННЯ ЕЛЬ-ГАМАЛЯ

Мета роботи: дослідження структури алгоритму і методики практичної реалізації криптосистеми шифрування Ель Гамаля.

Основні теоретичні відомості

Схема шифрування Ель Гамаля може бути використана як для формування цифрових підписів, так і шифрування даних.

Безпека схеми Ель Гамаля обумовлена складністю обчислення дискретних логарифмів в кінцевому полі.

Нині найбільш перспективними системами криптографічного захисту є системи з відкритим ключем. У таких системах для шифрування повідомлення використовується закритий ключ, а для розшифрування – відкритий.

Відкритий ключ не є секретним і може бути опублікований для використання усіма користувачами системи, які зашифровують дані. Розшифрування даних за допомогою відкритого ключа неможливе.

Для розшифрування даних одержувач зашифрованої інформації використовує секретний ключ, який не може бути визначений з відкритого ключа.

При використанні алгоритму шифрування Ель Гамаля довжина шифротексту вдвічі більша довжини початкового відкритого тексту M.

У реальних схемах шифрування необхідно використати в якості модуля n велике просте число, що має в двійковому представлені довжину 512...1024 біт.

Слід зазначити, що формування кожного підпису за цим методом вимагає нового значення k, причому це значення повинне вибиратися випадковим чином. Якщо порушник розкриє значення k, повторно використовуване відправником, то може розкрити і секретний ключ x відправника.

Алгоритм шифрування даних за схемою Ель Гамаля приведений нижче.

Методика виконання роботи

Завдання на виконання лабораторної роботи видається викладачем після проходження студентами співбесіди з основ криптографічного захисту інформації.

Порядок виконання роботи відповідає приведеній нижче криптосистемі шифрування даних за схемою Ель Гамаля.

Схема алгоритму шифрування даних Ель Гамаля

Визначення відкритого "y" і секретного "x" ключів

1. Вибір двох взаємно простих великих чисел p і q, q < p.
2. Вибір значення секретного ключа x, x <p.
3. Визначення значення відкритого ключа y з виразу:

$$y = q^x (mod\ p).$$

Алгоритм шифрування повідомлення M

1. Вибір випадкового числа k, що задовольняє умові:

$$1 \le k < p - 1 \text{ і } \text{НСД}(k, p-1) = 1.$$

2. Визначення значення a із виразу: $a = q^k (mod\ p)$.
3. Визначення значення b з виразу: $b = y^k M (mod\ p)$.
4. Криптограма C, що складається з a і b, відправляється одержувачеві.
5. Одержувач розшифровує криптограму за допомогою виразу:

$$Ma^x \equiv b(mod\ p).$$

Процедуру шифрування даних розглянемо на наступному прикладі (для зручності розрахунків в цьому прикладі використані числа малої розрядності)

1. Вибираємо два взаємно простих числа p=11 і q=2;
2. Вибираємо значення секретного ключа x, (x<p), x=8;
3. Обчислюємо значення відкритого ключа y із виразу:
$$y = q^x(mod\ p) = 2^8(mod\ 11) = 256\ (mod\ 11) = 3.$$
4. Вибираємо значення відкритого повідомлення M = 5;
5. Вибираємо випадкове число k = 9; НСД(9,10)=1;
6. Визначаємо значення *a* із виразу:
$$a = q^k(mod\ p) = 2^9(mod\ 11) = 512\ (mod\ 11) = 6;$$
7. Визначаємо значення b з виразу:
$$b = y^k M(mod\ p) = 3^9 \cdot 5\ (mod\ 11) = 98415\ (mod\ 11) = 9.$$

Таким чином, отримуємо зашифроване повідомлення як (a,b) = (6,9) і відправляємо одержувачеві.

8. Одержувач розшифровує цей шифротекст, використовуючи секретний ключ x і вирішуючи наступне порівняння:
$$M \cdot a^x \equiv b\ (mod\ p) = 5 \cdot 6^8 \equiv 9\ (mod\ 11) = 8398080 \equiv 9\ (mod\ 11).$$
Обчислене значення повідомлення M=5 є заданим початковим повідомленням.

Зміст звіту

1. Скласти блок-схему і програму алгоритму шифрування Ель Гамаля.
2. Лістинг програми шифрування заданого повідомлення з використанням алгоритму Ель Гамаля.
3. Висновки.

Лабораторна робота № 7

ДОСЛІДЖЕННЯ ЕЛЕКТРОННОГО ЦИФРОВОГО ПІДПИСУ (ЕЦП) ЕЛЬ ГАМАЛЯ

Мета роботи: дослідження структури алгоритму і методики практичної реалізації (ЕЦП) Ель Гамаля.

Основні теоретичні відомості

Загальновизнані прийоми встановлення достовірності фізичного підпису під документом абсолютно не придатні при обробці документів в електронній формі. Вирішенням цього питання є алгоритм, так званої системи електронної підписки документів. Для гарантії достовірності авторства і цілісності інформаційного повідомлення необхідно зашифрувати його вміст. При використанні цифрового підпису інформація не шифрується і залишається доступною будь-якому користувачеві, що має до неї доступ.

При обміні електронними документами по мережі значно знижуються витрати, пов'язані з їх обробкою, зберіганням і пошуком.

Одночасно при цьому виникає проблема, як аутентифікації автора електронного документу, так і самого документу, тобто встановлення достовірності автора і відсутності змін в отриманому електронному повідомленні.

ЕЦП використовується для аутентифікації текстів, що передаються по телекомунікаційних каналах. Функціонально він аналогічний звичайному рукописному підпису і має основні її властивості:

- засвідчує, що підписаний текст виходить від особи, що поставила підпис;
- не дає цій самій особі можливості відмовитися від зобов'язань, пов'язаних з підписаним текстом;
- гарантує цілісність підписаного тексту.

ЕЦП є відносно невеликим об'ємом додаткової цифрової інформації, що передається разом з підписаним текстом.

Концепція формування ЕЦП за схемою Ель Гамаля також базується на оборотності асиметричних шифрів і на взаємозв'язку вмісту повідомлення, самого підпису і пари ключів.

Ідея алгоритму цифрового підпису Ель Гамаля базується на тому, що для обгрунтування практичної неможливості фальсифікації цифрового підпису в ній використано складніше обчислювальне завдання дискретного логарифмування, ніж розкладання на множники великого цілого числа. Основною гідністю такої схеми цифрового підпису є можливість вироблення ЕЦП для великого числа повідомлень з використанням одного секретного ключа.

Безпека схеми Ель Гамаля обумовлена складністю обчислення дискретних логарифмів в кінцевому полі.

Опис схеми формування ЕЦП Ель Гамаля представлений нижче.

Методика виконання роботи

Завдання на виконання лабораторної роботи видається викладачем після проходження студентами співбесіди з основ аутентифікації даних і концепції формування електронного цифрового підпису за схемою Ель Гамаля.

Схема формування ЕЦП Ель Гамаля
Визначення відкритого "y" і секретного "x" ключів
(дії відправника)

1. Вибір двох взаємно простих великих чисел p і q, q<p.
2. Вибір значення секретного ключа x, x < p.
3. Визначення значення відкритого ключа y з виразу:

$$y = q^x (mod\ p).$$

Формування ЕЦП

1. Обчислення хеш-значення повідомлення M: m=h(M).
2. Вибір випадкового числа k, $0 < k < p - 1$ і $\text{НСД}(k, p-1) = 1$.
3. Визначення значення *a* з виразу: $a = q^k (mod\ p)$.
4. Визначення значення b з виразу:

$$m = (xa + kb)(mod(p-1)).$$

5. Цифровий підпис $S = (a, b)$ і відкритий текст повідомлення M відправляються одержувачеві.

Аутентифікація повідомлення – перевірка достовірності підпису (дії одержувача)

1. Обчислення хеш-значення прийнятого відкритого тексту повідомлення M

$$m' = h(M).$$

2. Підпис вважається достовірним, якщо a < p, m = m' і виконується умова

$$y^a a^b (mod\ p) = q^{m'} (mod\ p).$$

В якості процедури формування ЕЦП розглянемо наступний приклад (для зручності розрахунків в цьому прикладі використані числа малої розрядності):

1. Вибираємо просте число p і два випадкові числа q і x (q і x <p), p = 11, q = 2 і секретний ключ x = 8;
2. Обчислюємо значення відкритого ключа y

$$y = q^x (mod\ p) = 2^8 (mod\ 11) = 3;$$

3. Визначаємо хеш-значення початкового повідомлення M, (312) m = h(M), в цьому прикладі приймаємо m = 3 (методика визначення хеш-значення повідомлення M приведена в роботі 2).
4. Вибираємо випадкове ціле число k, взаємно просте з p-1. Приймаємо k=9, НСД(9,10)=1.
5. Для формування ЕЦП обчислюємо елементи підпису *a* і b

$$a = q^k (mod\ p) = 2^9 (mod\ 11) = 6.$$

Елемент b визначаємо за допомогою розширеного алгоритму Евкліда з наступного співвідношення:

$$m = (xa + kb)\big(mod(p-1)\big);\ 3 = (8 \cdot 6 + 9 \cdot b)(mod\ 10) = 9 \cdot b = -45(mod\ 10);$$

$$b = 5.$$

У цьому прикладі цифровим підписом є пара чисел *a*=6, b=5.

Цифровий підпис S=(*a*,b) і відкритий текст повідомлення M відправляють одержувачеві. Для контролю цілісності повідомлення і достовірності ЕЦП одержувач обчислює хеш-значення m' прийнятого відкритого тексту повідомлення M. При цьому відправник і одержувач використовує одну і ту ж хеш-функцію h(·).

Отримавши підписане повідомлення і відкритий ключ y=3, одержувач для перевірки достовірності підпису перевіряє виконання умови

$$y^a a^b (mod\ p) = q^{m'} (mod\ p),$$

$$3^6 \cdot 6^5 (mod\ 11) = 2^3 (mod\ 11),$$

$$5668704\ (mod\ 11) = 8\ (mod\ 11),$$

оскільки умова виконується, то прийняте одержувачем повідомлення визнається

справжнім.

Таким чином, процедура встановлення достовірності прийнятого повідомлення полягає в перевірці відповідності аутентификатора повідомлення.

Слід мати на увазі, що кожен підпис за схемою Ель Гамаля вимагає нового значення k. Випадкове значення k повинно зберігатися в секреті.

Зміст звіту

1. Скласти блок-схему алгоритму і програму формування ЕЦП Ель Гамаля на будь-якій зручній для студента мові.
2. Лістинг програми розрахунку ЕЦП Ель Гамаля відповідно до завдання.
3. Висновки: переваги і недоліки ЕЦП Ель Гамаля.

Лабораторна робота № 8

ПАРОЛЬНИЙ ЗАХИСТ

Мета роботи: вивчення принципів організації парольного захисту програм, ознайомлення з видами паролів, реалізація парольного захисту.

Основні теоретичні відомості

Стандартність архітектурних принципів побудови, устаткування і програмного забезпечення персональних комп'ютерів, мобільність програмного забезпечення визначають порівняно легкий доступ до інформації, що знаходиться в персональному комп'ютері. Несанкціонований доступ до інформації персонального комп'ютера – незапланованe ознайомлення, обробка, копіювання, застосування різних вірусів, модифікація або знищення інформації через порушення правил доступу.

Під захистом інформації розуміють створення організованої сукупності засобів, методів і заходів, призначених для попередження спотворення, знищення або несанкціонованого використання інформації, що захищається. До них відносяться апаратні і програмні засоби, криптографічне закриття інформації, фізичні заходи, організаційні заходи і законодавчі заходи. Один з методів захисту – парольна ідентифікація, що обмежує доступ несанкціонованого користувача.

Парольний захист програм

Включення захисту в програму пов'язане з розробкою програм із запитом інформації, тобто що вимагають для своєї роботи введення додаткової інформації, такої як паролі або номери ключів. Проте така перевірка доступу до програм або систем не повинна істотно позначатися на швидкодії програми або вимагати від користувача складних додаткових дій.

Пароль – це код, що використовується для отримання доступу до систем або файлів, оснащених парольним захистом. Паролі забезпечують збереження цілісності програмного забезпечення у складі обчислювальної системи, але для підтримки паролів потрібно високу дисциплінованість. При першій реєстрації користувача адміністратор визначає коло повноважень для отримання і зміни інформації або виконання певних керуючих дій в системі, керуючись його професійними обов'язками і посадовими інструкціями. Потім користувачеві пропонується ввести свій пароль згідно з правилами, прийнятими в цій системі. Метод паролів вимагає, щоб пароль (рядок символів), що вводиться користувачем, порівнювався з тим, який зберігається в обчислювальній системі для цього користувача. Якщо пароль вірний, система повинна вивести на екран терміналу дату і час останнього входу в систему цього користувача. Потім користувачеві надається можливість користуватися усією інформацією, доступ до якої йому дозволений (паролі можна також використовувати незалежно від користувача для захисту файлів, записів, полів даних всередині записів і так далі). Процедура встановлення достовірності користувачів за допомогою пароля приведена на рис.1. Парольний захист є досить ефективним, якщо:

- зберігати пароль в таємниці;
- переглядати систему для пошуку резидентних програм або троянських коней, призначених для перехоплення паролів; встановити захист в системі від таких програм;
- встановити вимоги до мінімальної довжини і множини символів в паролях;
- за наявності засобів використати інтелектуальні карти, розпізнавальні знаки, біометричні облаштування управління доступом;

- здійснювати періодичну зміну паролів і контроль їх термінів дії.

Система не повинна відображати паролі, що вводяться користувачем, або на місці вводу виводити послідовність випадкових символів. Не слід зберігати паролі у відкритому вигляді або на носії. Негайно після вводу пароля виконувати шифрування пароля і очищення пам'яті, що містить відкритий текст пароля. Для запобігання вгадуванню пароля рекомендується використати паролі, що генеруються комп'ютером, а також проводити блокування після визначеної кількості спроб вводу неправильного пароля.

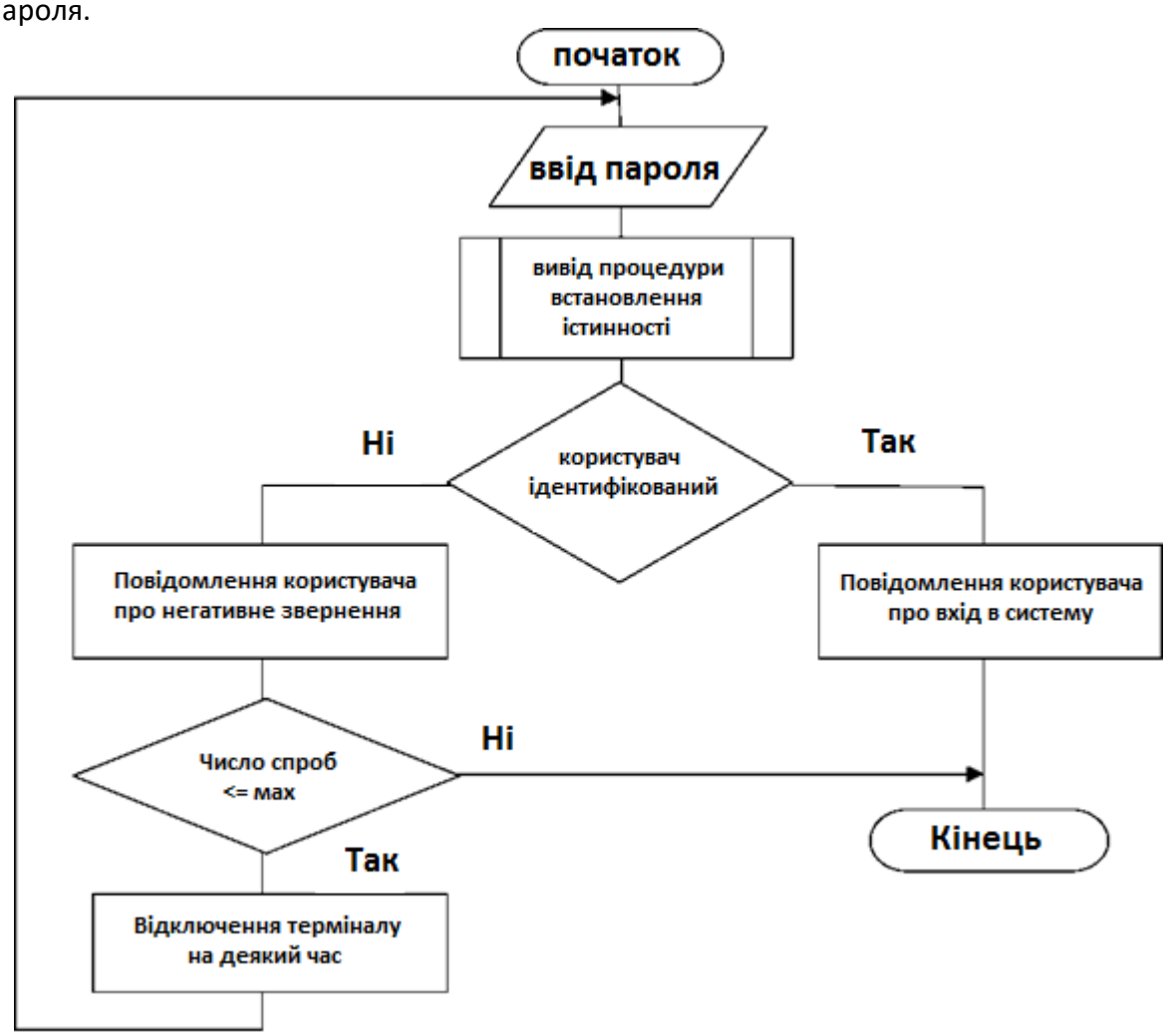

Рис. 1. Схема встановлення достовірності користувача.

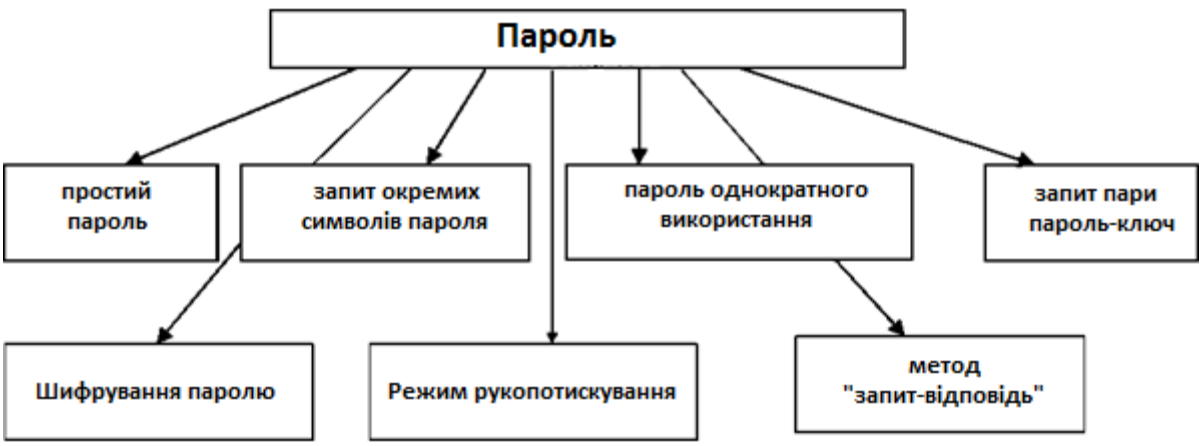

Рис. 2. Види паролів.

Простий пароль

Простий пароль – це рядок символів, що вводиться користувачем з клавіатури. У схемі з простим паролем користувачеві дозволяється самому вибирати пароль так, щоб його було легко запам'ятати. Іноді у ряді символів пароля і в його кінці залишають пропуски. Відмінність дійсного пароля від того, що здається (без пропусків) підвищує захищеність системи.

Підбір пароля шляхом простого перебору комбінацій припускає перебір усіх можливих поєднань символів в паролі. Час, необхідний для розгадування пароля методом простого перебору, є геометричною прогресією від довжини пароля, але є різні криві, залежні від розміру абетки, на основі якої був створений пароль і від розміру набору символів, по відношенню до якого розглядаються різні паролі.

Згідно з формулою Андерсена:

$$\mathbf{4,32 \times 10^4 \frac{RM}{EP} \leq A^3}$$

де R – швидкість передачі в лінії зв'язку (симв./с);

E – число символів, що передається в кожному повідомленні, яке передається при спробі отримати доступ;

S – довжина пароля;

A – число символів в абетці, з якого складений пароль;

P – імовірність правильного відгадування пароля.

Найбільший вплив на імовірність P розкриття пароля робить величина S. Збільшення пароля на один символ значно збільшує час для розкриття цього пароля. Тому застосування дуже довгих паролів може бути обґрунтоване. Боротьба з перебором комбінацій полягає у використанні програмного забезпечення, що обмежує мінімальну довжину пароля і використанні більш великої абетки (256 символів).

Вибірка символів

Використання в якості паролю окремих символів умовного слова (наприклад, 1 і 5 буква) запобігає ситуації, коли ціле слово може бути випадкове почуте. Запрошені символи паролю змінюються при кожній новій спробі доступу. Позиції запрошених символів можна отримати за допомогою деякої процедури перетворення, що прив'язана до показів годинників ЕОМ або виробити генератором псевдовипадкових чисел. Проте пароль слід змінювати досить часто, оскільки він може бути складений з окремих символів.

Пароль одноразового використання

У схемі одноразового використання пароля користувачеві видається список з N паролів. Такі ж N паролів зберігаються в ЕОМ (у зашифрованому вигляді). Ця схема забезпечує велику міру безпеки, але вона є і більше складною. Після використання пароля користувач викреслює його зі списку. При подальшій роботі система на цей пароль реагувати не буде, оскільки чекає наступний за списком пароль.

Паролі одноразового використання можуть застосовуватися також для встановлення достовірності підтвердження про відключення ЕОМ від обслуговування користувача і підтвердження достовірності вимоги користувача про відключення від ЕОМ. Кожен раз, коли отримана вимога користувача про закінчення роботи, ЕОМ негайно передає йому свій пароль одноразового використання і перериває зв'язок. Якщо користувач відключається і не отримує істинного пароля від ЕОМ, йому слід вжити запобіжні заходи.

Недоліки паролів одноразового використання :

користувач повинен пам'ятати увесь список паролів і стежити за поточним паролем;

у разі помилки в процесі передачі користувач не знає, чи слід йому передати той же пароль або надіслати наступний;
якщо паролі визначені шляхом використання лінійної послідовності псевдовипадкових чисел, то первинна послідовність може бути відновлена на підставі декількох перехоплених паролів.

Метод "запит-відповідь"

У методі "запит-відповідь" набір відповідей на m стандартних і n орієнтованих на користувача запитань зберігається в ЕОМ і керується операційною системою. Коли користувач робить спробу включитися в роботу, операційна система випадковим чином вибирає і задає йому деякі (чи всі) з цих питань. Користувач повинен дати правильну відповідь на усі запитання, щоб отримати дозвіл на доступ до системи. Іноді користувачам задається велика кількість стандартних запитань і від них потрібно відповіді на ті, які вони виберуть.

Шифрування паролів

Шифрування пароля підвищує безпеку системи. Цей метод передбачає, що пароль, який вводиться при вході в систему, шифрується і порівнюється із зашифрованим паролем, що зберігається у базі даних. Для шифрування пароля можна використати простий метод оборотного шифрування або складніший метод "безповоротної безладної зборки", коли декілька паролів в явній формі перетворюються в однаковий зашифрований пароль. В цьому випадку не існує жодної схеми для повернення до оригіналу пароля. Система просто шифрує кожен пароль користувача під час процесу реєстрації і звіряє його із зашифрованим паролем, що зберігається у власному файлі користувача.

Приклад цього методу – поліноміальне безповоротне представлення:

$$f(x) = (x^n + a_1 x^{n-1} + \cdots + a_{n-1}x + a_n) mod\ P,$$

де P – велике, a_i і n – цілі числа; x – пароль в явній формі; f(x) – зашифрований пароль.

Режим «рукостискання»

Операційна система може зажадати, щоб користувач встановив свою достовірність за допомогою коректної обробки алгоритмів, яку називають режимом "рукостискання" і вона може бути виконана як між двома ЕОМ, так і між користувачем і ЕОМ.

ЕОМ для встановлення достовірності могла дати користувачеві число, вибране випадковим чином, а потім запросити від нього відповідь. Для підготовки відповіді користувач "u" застосовує власне заздалегідь підготовлене перетворення t_u. Інформацією, на основі якого приймається рішення, тут є не пароль, а перетворення t_u. ЕОМ відсилає значення x, а користувач відповідає значенням $t_u(x)$. Навіть у разі знання значень x і $t_u(x)$ вгадати функцію перетворення неможливо. Функція перетворення може бути різна для кожного користувача.

Запит пари пароль-ключ

Паролі можна використати не лише для встановлення достовірності користувача по відношенню до систем, але і для зворотного встановлення достовірності. Це важливо, наприклад, в мережах ЕОМ, коли користувач хоче взаємодіяти тільки з цією ЕОМ і тому бажає переконатися в достовірності обчислювальної установки.

Користувач вводить з клавіатури своє ім'я, яке в незашифрованому вигляді відсилається по лініях зв'язку в ЕОМ. Замість секретного пароля в ЕОМ зберігається

ключ перетворення секретності, пов'язаний з кожним ім'ям. Потім користувач відсилає свою копію ключа перетворення. Якщо ключі ідентичні, то це означає, що встановлення достовірності як користувача, так і ЕОМ пройшло успішно. Ключ перетворення не повинен передаватися по лініях зв'язку. Якщо зв'язок припиняється, то кожна сторона, що бере участь в передачі повідомлень, буде негайна попереджена про ситуацію, що створилася.

Методика виконання роботи

Завдання на виконання лабораторної роботи видається викладачем після проходження студентами співбесіди з основ парольного захисту.

Порядок виконання роботи відповідає, наведеній нижче, схемі.

1. Вивчення існуючих методів парольного захисту.
2. Вибір методу парольного захисту відповідно до заданого варіанту.
3. Розробка алгоритму і програмна реалізація вибраного методу парольного захисту з використанням демонстраційних можливостей вибраної мови програмування.
4. Оформлення звіту.

Зміст звіту

4. Скласти блок-схему і програму вибраного методу парольного захисту з використанням демонстраційних можливостей вибраної мови програмування.
5. Лістинг програми.
6. Висновки: переваги і недоліки вибраного методу парольного захисту.

Варіанти індивідуальних завдань

Таблиця 1. Види парольного захисту

№ варіанту	Вид парольного захисту
1	Вибірка символів
2	Пароль одноразового використання
3	Шифрування паролів
4	Метод "запит-відповідь"
5	Режим "рукостискання"
6	Вибірка символів
7	Простий пароль
8	Запит пари пароль-ключ
9	Метод "запит-відповідь"
10	Режим "рукостискання"
11	Вибірка символів
12	Простий пароль
13	Пароль одноразового використання
14	Шифрування паролів
15	Вибірка символів
16	Метод "запит-відповідь"
17	Простий пароль
18	Режим "рукостискання"
19	Вибірка символів
20	Запит пари пароль-ключ

№ варіанту	Вид парольного захисту
21	Шифрування паролів
22	Режим "рукостискання"
23	Пароль одноразового використання
24	Метод "запит-відповідь"

Контрольні запитання

1. При дотриманні яких умов парольний захист є ефективним?
2. Які недоліки парольного захисту?
3. Що таке пароль?
4. Які операційні системи мають вбудований парольний захист?

Руслан Ігорович Запухляк
Степан Петрович Новосядлий
Тарас Володимирович Головатий

Захист інформації в комп'ютерних системах: методичні вказівки до виконання лабораторних робіт. – Прикарпатський національний університет імені Василя Стефаника, Івано-Франківськ, 2012 р. – 34 с.

Підписано до друку 15.02.2002 р.
Формат 60x84/$_{16}$
Умов. друк. арк.. , видавн. арк. .

Прикарпатський університет
імені Василя Стефаника
вул. Шевченка, 57
м. Івано-Франківськ
76025, Україна
тел. (03422) 59-60-82