МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДОНЕЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ВАСИЛЯ СТУСА

В. Г. Крижановський, Д. В. Чернов

ЛАБОРАТОРНИЙ ПРАКТИКУМ З КУРСУ «ЗАХИСТ ІНФОРМАЦІЇ В ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ СИСТЕМАХ»

Методичні вказівки

Вінниця
ДонНУ імені Василя Стуса
2023

УДК 004.056.5:[681.518::316.77](076.5)
К 822

Рекомендовано до друку вченою радою факультету інформаційних та прикладних технологій Донецького національного університету імені Василя Стуса (протокол № 9 від 24 квітня 2023 р.)

Автори:

В. Г. Крижановський – проф. кафедри прикладної математики та кібербезпеки;

Д. В. Чернов – старший викладач кафедри прикладної математики та кібербезпеки.

Рецензент:

П. К. Ніколюк, д-р фіз.-мат. наук, проф., проф. кафедри інформаційних технологій.

Н. Р. Кондратенко, канд. техн. наук, проф. кафедри захисту інформації Вінницького національного технічного університету.

Крижановський В. Г., Чернов Д. В.

К 822 Лабораторний практикум з курсу «Захист інформації в інформаційно-комунікаційних системах»: методичні вказівки. Вінниця: ДонНУ імені Василя Стуса, 2022. 72 с.

Наведено теоретичні відомості та інструкції з лабораторних робіт із курсу «Захист інформації в інформаційно-комунікаційних системах». Наведено теми затвердженої програми ЄДКІ для спеціальності 125 «Кібербезпека», які стосуються теми курсу. Лабораторні роботи відповідають частині напрямів ЄДКІ.

Посібник рекомендовано для студентів вищих навчальних закладів за напрямом «Кібербезпека» та може бути корисним студентам споріднених спеціальностей.

УДК 004.056.5:[681.518::316.77](076.5)

ЗМІСТ

ВСТУП

Інформаційно-комунікаційні системи (ІКС) – це нервова система сучасного постіндустріального суспільства, яка забезпечує численні функції керування виробництвом від малих підприємств до транснаціональних корпорацій, а також інформаційних супровід діяльності різноманітних суспільних інститутів. Захист інформації в таких системах складно переоцінити, і зрозуміло, що повною мірою вся підготовка бакалаврів спеціальності 125 «Кібербезпека» націлена на вирішення задач захисту інформації в ІКС. Різноманітність проблем захисту інформації в ІКС частково відображується в тематиці лабораторних робіт із курсу. Ці роботи стосуються питань захисту передачі інформації, захисту її зберігання та криптографічного захисту даних.

Роботи містять короткі теоретичні дані, опис процесу їх виконання, завдання, форми представлення звіту. Після представлення звіту відбувається захист роботи, під час якого студенти повинні відповісти на запитання щодо порядку виконання роботи, про достовірність отриманих даних та теорію процесів, які досліджуються в лабораторній роботі.

Лабораторна робота № 1
Зламування моноалфавітного підстановочного шифру методом частотної атаки

Мета роботи: ознайомитися на практиці з використанням частотної криптоатаки під час злому шифрів підстановки.

Вхідні дані:

Зашифрований текст; перелік літер, що найбільш часто зустрічаються в тексті; перелік літер, які найбільш часто використовуються в одній з мов.

Вихідні дані:

Розшифрований текст.

Теоретичні основи

Моноалфавітний шифр підстановки – шифр, у якому кожній літері вихідного алфавіту поставлена у відповідність одна літера шифру.

Наприклад, візьмемо слово «КУКУРУДЗА». Нехай букві «К» тексту відповідає буква «А» шифру, букві «У» тексту відповідає буква «Б» шифру, букві «Р» тексту відповідає буква «В» шифру, букві «З» тексту відповідає буква «Г» шифру, букві «А» тексту відповідає буква «Д» шифру, букві «Д» відповідає буква «К». Після підстановки літер шифру замість літер вихідного тесту слово «КУКУРУДЗА» у зашифрованому вигляді виглядатиме як «АБАБВБКГД».

Недоліком подібного шифрування є те, що якщо якась літера зустрічається у вихідному тексті найчастіше (наприклад, літера «О» в українському алфавіті), і відповідна їй буква шифру в зашифрованому тексті також зустрічається найчастіше.

У наведених нижче таблицях подані частоти появи літер в англійському та українському текстах:

Таблиця 1.1

Частота появи літер в англійському тексті (у відсотках)

Висока		Середня		Низька	
E	12,31	L	4,03	B	1,62
T	9,59	D	3,65	G	1,61
A	8,05	C	3,20	V	0,93
O	7,94	U	3,10	K	0,52
N	7,19	P	2,29	Q	0,20
I	7,18	F	2,28	X	0,20
S	6,59	H	2,25	J	0,10
R	6,03	W	2,03	Z	0,09
H	5,14	Y	1,88		

Таблиця 1.2

Ранжовані частоти використання українських букв

О	0,0942	р	0,0448	я	0,0248	ж	0,0093
А	0,0807	с	0,0424	з	0,0232	ю	0,0093
Н	0,0681	л	0,0369	б	0,0177	ц	0,0083
И	0,0626	к	0,0354	ь	0,0177	ш	0,0076
І	0,0575	д	0,0338	г	0,0155	ї	0,0065
В	0,0535	у	0,0336	ч	0,0141	є	0,0061
Т	0,0535	м	0,0303	й	0,0138	щ	0,0056
Е	0,0495	п	0,0290	х	0,0119	ф	0,0028

Знаючи частоти літер, що найбільше зустрічаються, і підрахувавши, які літери найчастіше зустрічаються в шифруванні, криптоаналітик може підібрати розшифровку для деяких літер тексту. Потім, аналізуючи короткі слова, знайти ще літери, справжні значення яких можна з високим рівнем упевненості передбачити. Наприклад, якщо вже розшифровано букву «О» і в тексті є слово «ОІО» (підкреслено вже розшифровані літери), то імовірніше, шифру «І» відповідає літера «К» у вихідному тексті («ОКО»). Чим далі текст розшифровується, тим легше проходить процес розшифровки.

Методичні вказівки

1. Запустити на виконання файл labw01.exe

На екрані з'явиться вікно виконання лабораторної роботи (рис. 1.1):

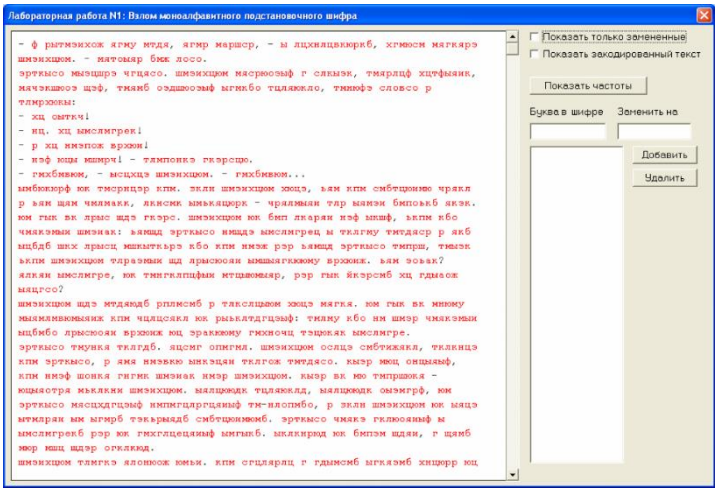


Рис. 1.1. Вікно виконання лабораторної роботи

У лівій частині вікна знаходиться зашифрований текст (літери, виділені червоним кольором). У процесі розшифрування розшифровані (правильно чи неправильно) літери тексту змінюють колір із червоного на чорний.

Щоб вказати для будь-якої букви шифру її справжнє (розшифроване) значення, потрібно в полі «Літера в шифрі» вказати значення букви, наприклад, «б», а в полі «Замінити на» – її справжнє значення, наприклад, «и», а потім натисніть кнопку «Додати». Результат такої дії наведено на рис. 1.2.

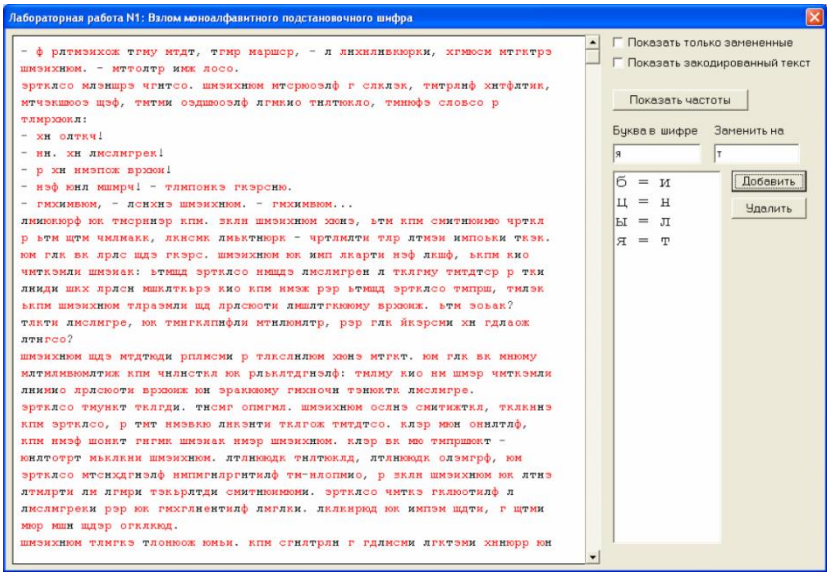


Рис. 1.2. Зміни вікна лабораторної роботи після розшифрування однієї літери

На рис. 1.3 наведено вікно виконання лабораторної роботи після додавання розшифровок кількох літер.

Щоб скасувати вказане розшифрування літери, потрібно в списку розшифровок мишкою вказати відповідну пару літер і натиснути кнопку «Видалити» (рис. 1.4).

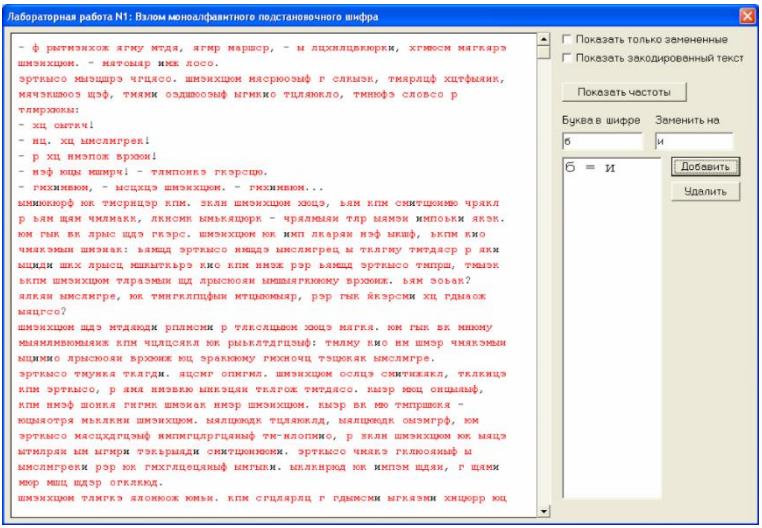


Рис. 1.3. Вікно лабораторної роботи після розшифрування кількох літер

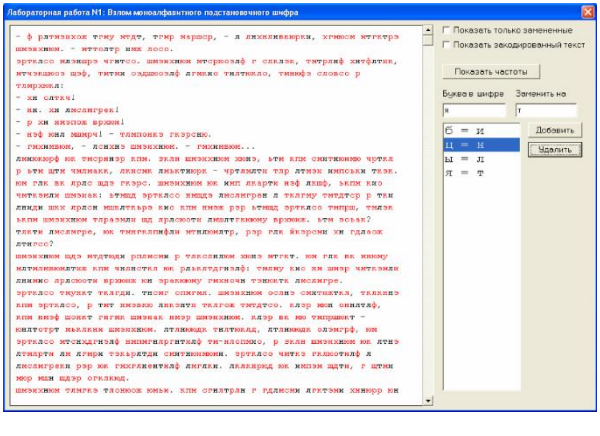


Рис. 1.4. Процедура видалення помилково вказаних розшифровок

Смуга вертикального скролінгу служить для навігації по тексту, що розшифровується.

2. Починається частотна атака з аналізу частот народження букв у шифруванні. Для цього у вікні виконання лабораторної роботи передбачена кнопка «Показати частоти». Під час її натискання на екран виводиться

перелік десяти літер, що найчастіше зустрічаються в шифрі, а також перелік літер, що найчастіше зустрічаються.

Слід пам'ятати, що для конкретного тексту частота літер може бути дещо іншою, ніж у середньому для обраної мови. Якщо загалом, наприклад, літера «т» зустрічається частіше, ніж буква «л», то в якомусь конкретному тексті буква «л» цілком може зустрічатися частіше за букву «т». Тому сліпо спиратися на дані частотного аналізу не слід.

3. У зашифрованому тексті здійснюється пошук коротких слів, зашифровані літери яких можна передбачити за вже розшифрованими літерами та частотною інформацією про букви.

Наприклад, є фрагмент тексту «зє», де «з» вже відомо.

Цей фрагмент може бути, імовірніше, словом «зі». У таблиці частот літера «є» шифру стоїть на 5-му місці, що відповідає позиції літери «і» української мови (4-е місце).

Далі повторюється пошук коротких слів, у яких можна здогадатися значення зашифрованих букв.

Коли багато літер уже відомо, зашифровані літери можуть заважати розумінню слів. Для полегшення подальшого аналізу у програмі передбачена можливість виставлення прапора «Показати лише замінені», під час виставлення якого всі зашифровані літери виводяться на екран у вигляді символів решітки.

Коли всі літери тексту розшифровані, на екран виводиться інформаційне вікно.

Контрольні питання

1. Що таке криптографія? Що таке шифр? У чому полягає мета шифрування?
2. Які основні вимоги висувають до шифру?
3. Що таке досконалий шифр?
4. Що таке криптографічна стійкість? Чим вона визначається?
5. У чому полягає суть шифрування за методом Цезаря?
6. Що таке частотний криптоаналіз?
7. Які переваги шифру Цезаря?
8. Які недоліки має шифр Цезаря?
9. Який шифр був побудований та вдосконалений на основі шифру Цезаря?
10. Шифр Атбаш.

Лабораторна робота № 2
Одноразові блокноти

Мета роботи: ознайомитися на практиці з принципом дії одноразового блокнота; використовуючи Matlab, досліджувати це завдання.

Перебіг виконання роботи

1. Згенеруйте одноразовий блокнот (масив випадкових чисел) завдовжки 10 000 000 байт.

Використовуйте для цього функцію «rand» матлаба.

2. Розтягніть отримані випадкові числа з інтервалу 0..1 на інтервал 0..255. Переведіть числа з формату з плаваючою комою до цілого формату. Скористайтеся функцією «uint8».

3. Побудуйте гістограму розподілу значень отриманого одноразового блокнота.

Скористайтеся функцією «hist». Побудуйте гістограму з 256 стовпчиками. Додайте гістограму до звіту про проведену роботу.

4. Збережіть одноразовий блокнот у файл (використовуйте функції «fopen», «fwrite», «fclose»). Спробуйте стиснути його програмою WinRAR чи WinZip. Який коефіцієнт стиснення було отримано? Додайте у звіт про проведену роботу точні розміри файла до та після стиснення. Поясніть результати.

5. Створіть змінну типу «рядок» і занесіть туди свої ПІБ. Наприклад, s = 'Пономаренко Микола Миколайович'; Перетворіть цей рядок на цілий формат (функція «uint16»).

Закодуйте отриманий масив сформованим одноразовим блокнотом (використовуйте функцію «bitxor» матлаба). Наведіть у звіті вихідний рядок, його цілочисельне подання та результат кодування одноразовим блокнотом. Декодуйте закодований рядок одноразовим блокнотом. Перетворіть отриманий масив у текстовий вигляд (використовуйте функцію «char» матлаба). Виведіть результат на екран. Чи відновився вихідний рядок?

6. Додайте до ПІБ ще й номер групи. Наприклад, s='Пономаренко Микола Миколайович 539'; Закодуйте та декодуйте цей рядок аналогічно діям, описаним у пункті 5, але з використанням іншого фрагмента одноразового блокнота. Додайте отримані результати до звіту.

7. Надайте викладачу звіт про виконану роботу в електронному або друкованому вигляді, а також текст програми.

Робота має бути виконана самостійно.

Контрольні питання

1. Чим симетричне шифрування відрізняється від асиметричного шифрування?
2. Що таке гібридне шифрування? Які проблеми вирішує гібридне шифрування?
3. У яких випадках використовується метод Вернама?
4. На якій операції базується метод Вернама?
5. Які властивості висуваються до ключа під час шифрування за методом Вернама?
6. Яке існує застереження під час використання методу Вернама?
7. Яка криптографічна стійкість методу Вернама?
8. У якому випадку повідомлення, зашифроване за допомогою методу Вернама, можна буде розшифрувати?

Лабораторна робота № 3
Мережа Файстеля

Мета роботи: запрограмувати мовою Матлаб мережу Файстеля.

Теоретичні основи

Мережею Файстеля називається метод оборотних перетворень тексту, за якого значення, обчислене від однієї з частин тексту, накладається на інші частини. Часто структура мережі виконується в такий спосіб, що для шифрування і розшифрування використовується один і той самий алгоритм – відмінність полягає тільки в порядку використання матеріалу ключа.

У схемі Файстеля кожний блок розбивається на ліву (l_0) і праву (r_0) частини, над якими здійснюються S раундів шифрування. Після завершення S раундів шифрування ліва (L_S) і права (R_S) частини міняються місцями (рис. 3.1):

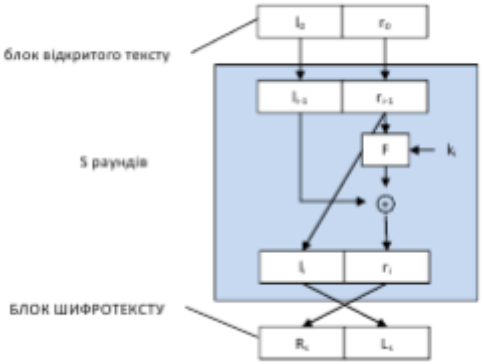


Рис. 3.1. Схема мережі Файстеля

Кожен раунд шифрування здійснюється за правилом:

$$l_i = r_i - 1;\ r_i = l_i - 1 + F\left(k_i, r_i - 1\right).$$

Процес розшифрування шифру Фейстеля принципово не відрізняється від процесу шифрування. Застосовується той самий алгоритм, але на вхід подається шифрований текст, а підключі використовуються у зворотній послідовності: для першого раунду береться підключ S-го раунду шифрування, для другого – (S-1)-ий, і так далі до тих пір, поки не буде введений ключ для останнього раунду. Ця властивість цієї схеми шифрування виявилась дуже зручною, тому що для розшифрування не потрібно вводити інший алгоритм, відмінний від алгоритму шифрування.

Мережа Файстеля надійно зарекомендувала себе як крипостійка схема проведення криптоперетворень, і її можна знайти практично в будь-якому сучасному блоковому шифрі.

Цікава особливість шифру Файстеля полягає в тому, що функція раунду є оберненою незалежно від властивості функції *F*.

Для створення криптостійкого шрифту залишилося визначити:

- спосіб генерування підключів k_i;
- кількість раундів *S*;
- функцію *F*.

Відповіді на ці питання можна знайти, наприклад, під час вивчення шифру DES.

Завдання

1. Напишіть алгоритмічною мовою системи Matlab функцію, яка обчислює виходи мережі Файстеля.

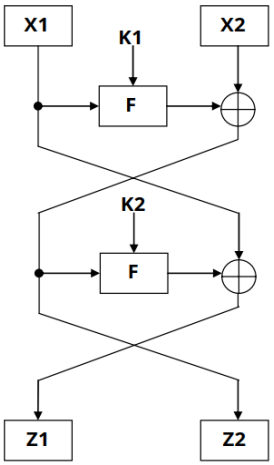


Рис. 3.2. Структура процесу шифрування у мережі Файстеля

Функція повинна мати вигляд:

function [z1,z2] = Mykolay(x1,x2,k1,k2,nr)
% тут має бути написаний
% вами текст програми, який обчислює
% вихід мережі Файстеля

Mykolay – ім’я функції (придумайте собі якесь інше).

На вхід функції подаватимуться:

x1, x2 – вихідні дані, які потрібно зашифрувати (тип даних – uint32);

nr – число раундів (повторів) мережі Файстеля;

k1, k2 – масиви з матеріалом ключа (довжина кожного з масивів дорівнює r, тип даних – uint32).

На виході функції мають бути:

z1, z2 – вихідні зашифровані дані (тип даних – uint32).

Функція F(x,k):

Число x зсувається праворуч на 3 біти, якщо раунд – парний, або вліво на 2 біти, якщо непарний (функція «bitshift» Матлаба).

Результат зсуву додається сумою за модулем два з k (функція «bitxor» Матлаба).

2. Напишіть мовою Матлаб функцію, яка декодує раніше закодовані дані z1 та z2.

Щоб декодувати один раунд мережі Файстеля (останній раунд r), запишемо вирази для z1 та z2:

z1 = x1 xor F(z2, k2(nr))

z2 = x2 xor F(x1, k1(nr))

Звідси можна знайти спочатку x1 (тут xor – сума за модулем два):

x1 = z1 xor F(z2, k2(nr))

а потім і x2:

x2 = z2 xor F(x1, k1(nr))

Ця дія має повторюватися у циклі nr разів.

Приклад циклу, що проходить у зворотному порядку і виводить на екран числа від nr до 1:

```
for kkk=nr:-1:1
disp(kkk);
end
```

3. Спробуйте перевірити працездатність двох функцій. Надайте викладачу для перевірки тексти програм в електронному вигляді.

Контрольні питання

1. У чому полягає ідея складових блокових шифрів?
2. Що таке SP-мережа?
3. У чому полягає дифузія та конфузія?
4. Дайте визначення мережі Файстеля та лавиноподібного ефекту.
5. Що називають раундом та гілками мережі Файстеля?
6. У чому полягають процеси шифрування та дешифрування за допомогою мережі Файстеля?
7. Що буде, якщо збільшити чи зменшити кількість раундів мережі Файстеля?
8. Який може бути утворююча функція *F*?

Лабораторна робота № 4
Метод шифрування з відкритим ключем RSA

Мета роботи: ознайомитися на практиці з методом шифрування RSA та особливостями його практичної реалізації (зведення великих чисел великою мірою з використанням модулярного множення за Монтгомері та методом двійкового зведення у степінь).

Вихідні дані:

Зашифрована кількість C, закритий ключ D, частина відкритого ключа N.

Вихідні дані:

Потрібно декодувати число M.

Теоретичні основи

Для розшифрування числа M у методі RSA досить зашифроване число C звести в степінь D і знайти залишок від поділу на N.

Основна проблема, що виникає під час практичного використання методу RSA, полягає в тому, що числа C, D і N дуже великі. Використовувати традиційні методи множення чисел під час зведення числа C у степінь D не вийде через переповнення обчислювального пристрою. На практиці використовують метод бінарного зведення в степінь для скорочення кількості множень, а також метод модулярного множення Монтгомері для того, щоб не виходити з розрядності чисел C, D і N у процесі множення більше, ніж на 1 розряд.

Методичні вказівки

1. Скористайтеся двійковим методом зведення в степінь, щоб визначити послідовність множень (поточний результат множиться сам на себе або на вихідне число), яка мінімізує загальну кількість операцій множення.

2. Переведіть число C на модулярну форму.

3. Зведіть C у степінь D двійковим методом, користуючись модулярним множенням Монтгомері (використовуйте на вихідне число C, а його модулярну форму!).

4. Переведіть число із модулярної форми у звичайну. Для цього помножте його на 1 за Монтгомері.

5. Перевірте розшифроване повідомлення M (ввівши його в рядок «Декодоване повідомлення» та натиснувши кнопку «Перевірити»).

Вікно програми виконання лабораторної роботи

Ця програма є калькулятором, у якому реалізовані всі необхідні для виконання лабораторної роботи функції: переклад із звичайної форми

в модулярну, переведення числа з десяткової системи числення у двійкову, а також множення за Монтгомері.

На рис. 4.1 наведено вікно програми.

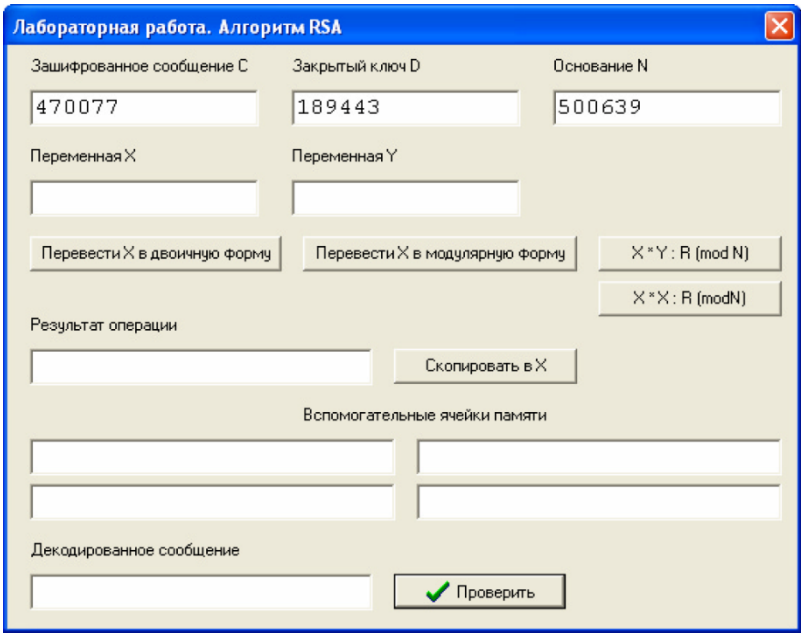


Рис. 4.1. Вікно програми виконання лабораторної роботи

У верхніх трьох рядках введення є вихідні дані для виконання роботи. Вміст будь-якого рядка введення в цій програмі можна виділити і скопіювати (права клавіша миші або комбінація клавіш «Ctrl-C») і вставити в інший рядок введення (права клавіша миші або комбінація клавіш «Ctrl-V»).

Копка «Перевести X у двійкову форму» переводить вміст рядка введення «Змінна X» у двійковий вигляд і поміщає результат у рядок введення «Результат операції».

Кнопка «Перевести X у модулярну форму» переводить вміст рядка введення «Змінна X» у модулярну форму та поміщає результат у рядок введення «Результат операції».

Кнопка «X*X : R (mod N)» виконує множення за Монтгомері вмісту рядка введення «Змінна X» самого на себе і поміщає результат у рядок введення «Результат операції».

Кнопка «X*Y : R (mod N)» виконує множення Монтгомері вмісту рядка введення «Змінна X» на вміст рядка вводу «Змінна Y» і поміщає результат у рядок введення «Результат операції».

Кнопка «Скопіювати в X» містить результат останнього обчислення (вміст рядка введення «Результат операції») у рядок введення «Змінна X».

Чотири рядки введення, що знаходяться нижче, є допоміжними і використовують для зберігання будь-якої інформації, наприклад, числа D у двійковому вигляді.

Кнопка «Перевірити» призначена для перевірки правильності отриманого результату, який має бути попередньо поміщений у рядок введення «Декодоване повідомлення».

Контрольні питання

1. У чому полягає алгоритм RSA?
2. На чому заснована безпека RSA?
3. Назвіть основні параметри алгоритму, як обчислюються ключі?
4. Як відбувається шифрування та дешифрування в цьому алгоритмі? На основі яких формул?
5. Опишіть основні методи злому RSA?
6. Яких необхідно вжити заходів, щоб уникнути розкриття алгоритму?
7. Для чого і чому використовують комбіновані криптоалгоритми?
8. У чому полягають переваги та недоліки асиметричних алгоритмів?
9. У чому полягають переваги та недоліки симетричних алгоритмів?

Лабораторна робота № 5
Прихована передача інформації у JPEG-зображеннях

Мета роботи: ознайомитися з візуальними спотвореннями внаслідок впровадження інформації у різні компоненти JPEG зображення.

Під час роботи у зображення 800 × 600 пікселів впроваджується приблизно 700 Кбайт інформації (роман Френка Герберта «Дюна» в одному з форматів електронних книг). Завдання студентів – забезпечити, щоб візуальні спотворення зображення були якомога менш помітні.

Теоретичні основи

На відміну від криптографії, стеганографія призначена для приховування самого факту наявності інформації.

Саме стеганографія (грец. – тайнопис) вивчає методи та засоби приховування інформації. Методи та засоби приховування інформації в електронних файлах належать до комп'ютерної стеганографії.

Основними стеганографічними поняттями є повідомлення й контейнер. Повідомленням $m \in M$ називають секретну інформацію, наявність якої необхідно приховати, де M – множина всіх повідомлень. Контейнером $b \in B$ називають несекретну інформацію, яку використовують для приховування повідомлень, де B – множина всіх контейнерів. Пустий контейнер (контейнер-оригінал) – це контейнер b, що не містить у собі повідомлень, заповнений контейнер (контейнер-результат) b_m – це контейнер b, що містить повідомлення m.

Стеганографічним перетворенням вважають залежності виду F і F^{-1}:

$$F : M \times B \times K \to B,\ F^{-1} : B \times K \to M\ , \qquad (5.1)$$

які відповідають трійці (повідомлення, пустий контейнер, ключ із множини K), що приводить до контейнеру-результату та пари (заповнений контейнер, ключ із множини K), яка відтворює вхідне повідомлення, тобто:

$$F\left(m,b,k\right) = b_{m,k}\,, \quad F^{-1}\left(b_{m,k}\right) = m\,, \qquad (5.2)$$

де $m \in M,\ \ b, b_m \in B,\ k \in K$.

Стеганографічною системою називають (F, F^{-1}, M, B, K) – співвідношення повідомлень, контейнерів та перетворень, що їх поєднують.

Аналіз практично застосованих методів комп'ютерної стеганографії дає змогу виділити такі основні класи:

1. Методи, що базуються на наявності вільних проміжків у представленні / збереженні даних.

2. Методи, що базуються на принципі надлишкового представлення / збереження даних.

3. Методи, що базуються на застосуванні спеціально розроблених форматів представлення / збереження даних.

Варто зауважити, що методи внесення прихованої інформації в об'єкти залежать насамперед від призначення й типу об'єкта, а також від формату, в якому представлені дані. Тобто для будь-якого формату представлення комп'ютерних даних можуть бути запропоновані власні стеганографічні методи.

Наразі наявне програмне забезпечення не має функціональних можливостей приховування інформації до зображень типу *.jpg, крім програми Steghide. Проте до недоліків Steghide варто віднести:

- консольний інтерфейс користувача;
- перевантаження ключами налаштування роботи;
- відсутність алгоритму кодування інформації під час занесення до контейнера;
- не передбачено використання динамічних крипто-ключів та їх захист;
- додаток не підтримує платформу MacOS X.

Методичні вказівки

Під час виконання роботи можна вказувати програмі, які компоненти JPEG зображення впровадити більше інформації, а які менше. Сумарна кількість інформації за таких умов залишається незмінною і змінюється тільки її розподіл між компонентами зображення.

Після будь-яких змін у налаштуваннях натискайте на напис «Зображення з впровадженим повідомленням», і тоді на екран буде виведено це зображення, а праворуч від нього виводитиметься кількісна оцінка якості в дБ. Для виконання лабораторної роботи необхідно, щоб якість була більша за 43 дБ. Та підгрупа, у якої значення PSNR (пікове співвідношення сигнал / шум) буде максимальним, отримує заохочення.

За допомогою повзунка можна вибирати баланс розподілу інформації між компонентами кольору та яскравості. Визначте, який компонент спотворення помітніший.

За допомогою повзунка «Маскування в текстурних ділянках» можна перерозподіляти інформацію між однорідними (небо) ділянками зображення та текстурними (листя, бриж на воді).

У восьми рядках введення можна кожному з коефіцієнтів ДКП (під час JPEG-стиску виконується дискретне косинусне перетворення у блоках 8 × 8 пікселів і у кожному блоці здійснюється квантування) окремо задати частку впровадження туди інформації. Ці коефіцієнти мають

бути позитивними числами, більшими за нуль. 0-й коефіцієнт відповідає найнижчій частоті, а 9-й – найвищій.

Чим більший коефіцієнт буде задано, тим більше інформації буде запроваджено у відповідні частоти зображення. Визначте, на яких частотах людське око краще виявляє спотворення.

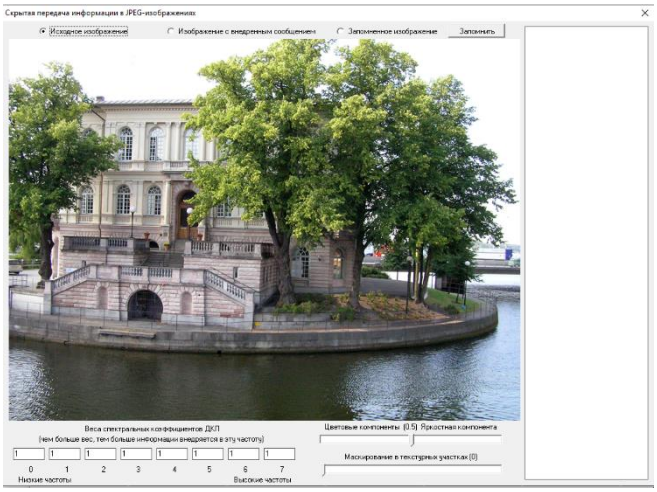


Рис. 5.1. Робоче вікно програми

Контрольні питання

1. Яка мета методів стеганографії?
2. Які методи найчастіше використовуються у цифровій стеганографії?
3. Як виконується приховування даних методом найменш значимого біта LSB?
4. Як залежить візуальна якість зображення-контейнера від номера використаного біта (бітів) для приховування даних за методом LSB?
5. Які дані можна приховати методом LSB?
6. Як визначити максимальний обсяг даних, які приховати у файлі-контейнері за методом LSB?
7. Як можна підвищити скритність даних у методі LSB?

Лабораторна робота № 6
Використання хеш-функцій з прикладу MD5. Оцінка стійкості пароля до злому

Мета роботи: ознайомитися на практиці з роботою хеш-функцій. Написати на Matlab програму для оцінювання стійкості пароля до злому.

Теоретичні основи

Оцінимо стійкість U пароля зі злому[1].

1. Нехай L – довжина пароля.

Якщо довжина пароля $L \leq 4$, то $U = 0$;

якщо $5 \leq L \leq 7$, то $U = 6$;

якщо $8 \leq L \leq 15$, то $U = 12$;

якщо 16 L, то $U = 18$.

2. Якщо в паролі є літери, але тільки в одному (нижньому або верхньому) регістрі, то $U = U + 5$;

якщо в паролі є літери у нижньому і верхньому регістрах, то $U = U + 7$.

3. Нехай N – число цифр у паролі.

Якщо число цифр у паролі $1 \leq N \leq 2$, то $U = U + 5$;

якщо $3 \leq N$, то $U = U + 7$.

4. Нехай S – число спецсимволів (# $ % @) у паролі.

Якщо $1 \leq S < 2$, то $U=U+5$;

якщо $2 \leq S$, то $U = U + 10$.

5. Якщо в паролі є літери в обох регістрах, спецсимволи та цифри, то $U = U + 6$;

якщо чогось одного з цього немає, $U = U + 4$.

Якщо $U < 16$, пароль дуже слабкий;

якщо $15 < U < 25$ – слабкий;

якщо $24< U < 35$ – середній;

якщо $34< U < 45$ – сильний;

якщо $44 < U$ – дуже сильний.

Методичні вказівки

1. Створіть на своєму комп'ютері якийсь робочий підкаталог і скопіюйте туди *.m файли із каталогу лабораторної роботи. Запустіть Matlab та виберіть створений підкаталог як «Current Directory».

[1] Оцінка стійкості паролю за допомогою ентропії є точнішою, спробуйте порівняти ці результати оцінки якості паролю з результатами обрахування ентропії паролю.

2. Ознайомтеся з функцією обчислення хеш-функції, що додається до лабораторної роботи MD5 (файл md5.m). Наберіть у Matlab команду help md5.

3. Надайте текстовій змінній своє ПІБ. Наприклад, s=’Пономаренко Микола Миколайович'; Обчисліть значення MD5 для цієї змінної. Змініть якусь одну букву у своєму ПІБ. Обчисліть значення MD5 для зміненого тексту. Поясніть отриманий результат.

4. Поясніть, як хеш-функція може використовуватися під час аутентифікації для того, щоб забезпечити автентифікацію сеансу зв’язку та приховати пароль, що передається каналами зв’язку від користувача до сервера.

5. Напишіть програму для оцінювання складності пароля (заданого у вигляді текстового рядка). У підкаталозі лабораторної роботи викладено функції, що полегшують написання цієї програми:

isbigl(s) – повертає 1, якщо s – літера у верхньому регістрі;
issml(s) – повертає 1, якщо s – літера в нижньому регістрі;
iscif(s) – повертає 1, якщо s – цифра;
isspec(s) – повертає 1, якщо s – спецсимвол.

Ознайомтеся з текстами цих функцій та використовуйте їх під час написання своєї програми.

6. Продемонструйте на прикладах роботу своєї програми. Наведіть приклади середнього та сильного паролів.

7. Поясніть, як хеш-функція може використовуватися під час аутентифікації для того, щоб забезпечити справжність сеансу зв’язку та приховати пароль, що передається каналами зв’язку від користувача до сервера.

8. Напишіть програму для оцінювання складності пароля (заданого у вигляді текстового рядка). У підкаталозі лабораторної роботи викладено функції, що полегшують написання цієї програми:

isbigl(s) – повертає 1, якщо s – літера у верхньому регістрі;
issml(s) – повертає 1, якщо s – літера в нижньому регістрі;
iscif(s) – повертає 1, якщо s – цифра;
isspec(s) – повертає 1, якщо s – спецсимвол.

Ознайомтеся з текстами цих функцій і використовуйте їх під час написання своєї програми.

9. Продемонструйте на прикладах роботу своєї програми. Наведіть приклади середнього та сильних паролів.

Контрольні питання

1. Що таке хеш-функція? Коли вона криптографічно стійка? Що таке лавинний ефект?
2. Які основні властивості має задовольняти хеш-функція?
3. Алгоритм MD5, основні етапи його виконання.
4. Довжина вхідного блоку в бітах функції стиснення алгоритму MD5.
5. Назвати число раундів у алгоритмі MD5.
6. Назвати кількість кроків у кожному раунді алгоритму MD5.
7. Які операції використовуються у функції стиснення алгоритму MD5?
8. Перерахувати основні постійні дані, що використовуються в алгоритмі MD5.
9. Скільки примітивних функцій використовують в алгоритмі MD5?

Лабораторна робота № 7
Цифровий підпис

Мета роботи: набути уміння створювати й перевіряти підпис повідомлення за допомогою алгоритмів RSA та Ель-Гамаля; навчитись створювати власні ключі криптографічного захисту даних, обмінюватися ними з іншими користувачами, шифрувати та підписувати повідомлення за допомогою системи GNU Privacy Guard.

Матеріально-технічне забезпечення: ПК зі встановленим програмним забезпеченням GNU Privacy Guard, інструкції до лабораторної роботи, текстові повідомлення для шифрування та підписування згідно з варіантом.

Теоретичні основи

Використання новітніх ІТ-технологій дає змогу досягнути істотного і швидкого зменшення непродуктивних витрат. Одним із важливих шляхів є перехід до безпаперових технологій роботи з документами та впровадження безпечних технологій дистанційного надання послуг. Вирішення цього завдання тісно пов'язане з регулюванням використання електронного підпису в Україні.

Так, наприклад, з осені 2017 року в Україні відбувся початок видачі внутрішніх паспортів у вигляді ID-картки одразу з електронним цифровим підписом (ЕЦП) і впровадження MobileID (ЕЦП на SIM-картці).

Правовий статус електронного цифрового підпису в Україні визначається Законом України «Про електронний цифровий підпис». Порядок застосування електронного цифрового підпису органами державної влади, органами місцевого самоврядування, підприємствами, установами та організаціями державної форми власності визначається Кабінетом Міністрів України. Порядок застосування електронного підпису, зокрема електронного цифрового підпису в банківській системі України та суб'єктами переказу коштів, визначається Національним банком України.

Слід розрізняти поняття «електронний підпис» та «електронний цифровий підпис».

Під електронним підписом (ЕП) розуміються дані в електронній формі, які додаються до інших електронних даних або логічно з ними пов'язані та призначені для ідентифікації підписувача цих даних.

Наприклад, електронним підписом є комбінація логіна і пароля, що вводяться користувачем під час реєстрації в системі.

Під час підтвердження здійснення банківського платежу в електронній формі електронним підписом стане введення користувачем отриманого від банку коду для перевірки і підтвердження його введення натисканням клавіші Enter.

Під електронним цифровим підписом (ЕЦП) розуміється такий електронний підпис, що був отриманий внаслідок криптографічного перетворення набору електронних даних.

Електронний цифровий підпис дає змогу підтвердити цілісність підписаного з його допомогою документа та ідентифікувати підписувача.

Електронний цифровий підпис призначений для забезпечення діяльності фізичних і юридичних осіб, яка здійснюється з використанням електронних документів. Укладення юридичними особами господарських угод в електронній формі з використанням сторонами ЕЦП є повністю правомірним.

У випадках, коли відповідно до законодавства необхідне засвідчення дійсності підпису на документах та відповідності копій документів оригіналам на електронний документ накладається ще один електронний цифровий підпис юридичної особи (наприклад, ЕЦП нотаріуса), спеціально призначений для таких цілей. Такий додатковий ЕЦП називається електронною печаткою.

Електронний цифровий підпис накладається за допомогою особистого ключа та перевіряється за допомогою відкритого ключа.

Особистий ключ – параметр електронного цифрового підпису, доступний тільки підписувачу.

Відкритий ключ – параметр електронного цифрового підпису, доступний усім суб'єктам відносин у сфері використання електронного цифрового підпису.

Відкритий ключ електронного цифрового підпису використовується для перевірки електронного цифрового підпису.

У тих випадках, коли ЕЦП використовується для підпису зашифрованого документа, з метою безпеки використовується дві пари ключів. Одна пара ключів використовується для шифрування документа (т. з. ключі для протоколу розподілу), а друга пара ключів – для накладення на документ електронного цифрового підпису.

Сертифікатом відкритого ключа є документ, який засвідчує чинність і належність відкритого ключа ЕЦП підписувачу. Сертифікати відкритих ключів можуть розповсюджуватися в електронній формі або у формі документа на папері.

Нотаріуси, державні реєстратори прав на нерухоме майно, державні реєстратори юридичних осіб, фізичних осіб-підприємців та громадських формувань повинні використовувати тільки захищені носії особистих ключів (наприклад, смарт-карти, електронні ключі, криптомодулі), що забезпечують захист записаних на нього даних від несанкціонованого доступу. Для кожного особистого ключа потрібен окремий носій.

Органи сертифікації

Відповідно до Закону України «Про електронний цифровий підпис», в Україні існує п'ять видів органів, пов'язаних із сертифікацією ключів:

1. Центри сертифікації ключів (ЦСК).
2. Акредитовані центри сертифікації ключів (АЦСК).
3. Центральний засвідчувальний орган (ЦЗО).
4. Засвідчувальний центр органу виконавчої влади або іншого державного органу (ЗЦ).
5. Контролюючий орган (КО) Держспецзв'язок.

Центром сертифікації ключів (ЦСК) може бути юридична особа, незалежно від форми власності, або фізична особа, яка є суб'єктом підприємницької діяльності, що надає послуги електронного цифрового підпису та засвідчила свій відкритий ключ у центральному засвідчувальному органі (ЦЗО) або засвідчувальному центрі (ЗЦ).

Акредитованим центром сертифікації ключів (АЦСК) є Центр сертифікації ключів (ЦСК), акредитований відповідно до «Порядку акредитації центру сертифікації ключів», що був затверджений постановою № 903 Кабінету Міністрів України від 13 липня 2004 р. АЦСК повинен використовувати для надання послуг у сфері цифрового підпису лише надійні засоби ЕЦП, що мають сертифікат відповідності або позитивний експертний висновок за результатами державної експертизи у сфері криптографічного захисту інформації.

Центральний засвідчувальний орган (ЦЗО) видає посилені сертифікати ключів центрам сертифікації ключів (АЦСК) та засвідчувальним центрам (ЗЦ). Центральний засвідчувальний орган визначається Кабінетом Міністрів України. Постановою Кабінету Міністрів України «Про затвердження Положення про центральний засвідчувальний орган» з жовтня 2011 року виконання функцій Центрального засвідчувального органу було покладено на Міністерство юстиції України.

Технічне й технологічне забезпечення виконання функцій центрального засвідчувального органу здійснюється державним підприємством «Інформаційний центр» Міністерства юстиції України, яке визначено адміністратором інформаційно-телекомунікаційної системи центрального засвідчувального органу.

Щорічну планову перевірку Центрального засвідчувального органу (ЦЗО) здійснила комісія у складі працівників *Державної служби спеціального зв'язку та захисту інформації (ДССЗЗІ).*

Засвідчувальний центр центрального органу виконавчої влади (ЗЦ) визначається Кабінетом Міністрів України для забезпечення реєстрації, засвідчення чинності відкритих ключів та акредитації групи центрів сертифікації ключів (ГЦСК), які надають послуги електронного цифрового

підпису цьому органу і підпорядкованим йому підприємствам, установам та організаціям.

Інші державні органи за необхідності, за погодженням із Кабінетом Міністрів України визначають свої засвідчувальні центри, призначені для виконання функцій, зазначених у частині першій цієї статті. Так, свій засвідчувальний центр був створений Національним банком України.

Засвідчувальний центр (ЗЦ) відносно групи центрів сертифікації ключів (ГЦСК) має ті самі функції і повноваження, що й центральний засвідчувальний орган (ЦЗО) стосовно центрів сертифікації ключів (ЦСК).

Контролюючий орган (КО) – спеціально уповноважений центральний орган виконавчої влади з питань організації спеціального зв'язку та захисту інформації.

Функції контролюючого органу виконує Державна служба спеціального зв'язку та захисту інформації (ДССЗЗІ).

Порядок видачі сертифікатів

Сертифікат відкритого ключа видається центром сертифікації ключів (ЦСК), який засвідчує чинність і належність відкритого ключа підписувачу. ЦСК завіряє сертифікат відкритого ключа своїм підписом.

Генерація особистого та відкритого ключів для органів державної влади, органів місцевого самоврядування, підприємств, установ та організацій державної форми власності здійснюється підписувачем в акредитованому центрі сертифікації ключів (АЦСК), що обслуговує установу, або безпосередньо в установі з використанням надійних засобів електронного цифрового підпису. Згенерований особистий ключ підписувача захищається паролем та записується на носій ключової інформації.

Генерація особистого та відкритого ключів для фізичних або юридичних осіб недержавної форми власності здійснюється в центрі сертифікації ключів (ЦСК) після ідентифікації заявника та отриманих від нього даних, необхідних для формування сертифіката. З метою збереження секретності ключа генерація особистого ключа проводиться безпосередньо самим користувачем в офісі ЦСК або АЦСК. За необхідності користувач може згенерувати ключову пару самостійно, використовуючи спеціальне програмне забезпечення. У такому разі він повинен буде відіслати в ЦСК / АЦСК запити на формування сертифіката з відкритим ключем ЕЦП (а за необхідності – і на формування сертифіката з відкритим ключем протоколу розподілу) разом з усіма необхідними документами для реєстрації.

Особистий ключ підписувача повинен відповідати відкритому ключу, зазначеному в сертифікаті.

Центр сертифікації ключів має право встановлювати під час формування сертифіката ключа належність відкритого ключа та відповідного особистого ключа підписувачу.

Центр сертифікації ключів має право надати допомогу під час генерації особистих ключів.

Зберігання особистих ключів підписувачів та ознайомлення з ними в центрі сертифікації ключів забороняється.

Підписувач зобов'язаний зберігати особистий ключ у таємниці та надавати центру сертифікації ключів дані для засвідчення чинності відкритого ключа і своєчасно надавати центру сертифікації ключів інформацію про зміну даних, відображених у сертифікаті ключа.

На виданий сертифікат установлюється термін його дії. Після закінчення цього терміну власнику видається новий сертифікат.

ПОНЯТТЯ ЦИФРОВОГО ПІДПИСУ

Із широким розповсюдженням у сучасному світі електронних форм документів, зокрема і конфіденційних, та засобів їх обробки, особливо актуальним є питання автентифікації, ідентифікації та неспростовності електронної документації. Для захисту від підробки, перевірки цілісності даних та достовірності джерела повідомлення використовують цифровий підпис.

Електронний підпис – електронні дані, які додаються підписувачем до інших електронних даних або логічно з ними пов'язуються і використовуються ним як підпис.

Цифровий підпис – вид електронного підпису, отриманого за результатом криптографічного перетворення набору електронних даних, який додається до цього набору або логічно з ним поєднується і дає змогу підтвердити його цілісність та ідентифікувати підписувача.

Існує декілька алгоритмів побудови цифрового підпису (ЦП). Найбільш ефективним та найпоширенішим у застосуванні наразі є алгоритм ЦП на основі асиметричних криптосистем із використанням хеш-функцій. Хеш-функція являє собою функцію, математичну або іншу, що отримує на вхід рядок змінної довжини і перетворює його в рядок фіксованої, зазвичай меншої, довжини. Такі перетворення ще називають функціями згортки, а їх результати – хешем, хеш-значенням або дайджестом повідомлення.

Хеш-функція H, яка використовується у алгоритмі ЦП, призначена для того, щоб стиснути повідомлення M довільної довжини до двійкового хеш-значення $h(M)$ фіксованої довжини.

Основні властивості криптографічної хеш-функції:

1) детермінованість – для однакових повідомлень M функція має повертати однакові хеш-значення h;

2) односторонність – за значенням h неможливо відновити M;

3) наявність лавинного ефекту – будь-які, навіть незначні, зміни у повідомленні M призводять до значних змін у хеш-значенні h;

4) відсутність колізій (унікальність хеша) – ймовірність співпадіння хеш-значень двох різних повідомлень повинна бути надзвичайно малою;

5) висока швидкість роботи.

ЕТАПИ ЦИФРОВОГО ПІДПИСУ

1. Генерація пари ключів. За допомогою алгоритму генерації ключів створюється пара ключів – закритий (для створення підпису) та відкритий (для перевірки підпису).

2. Формування підпису. Для заданого електронного документа за допомогою деякої хеш-функції обчислюється хеш-значення, після чого воно зашифровується із використанням закритого ключа підписувача. Зашифрований дайджест і є ЦП для цього документа.

3. Перевірка (верифікація) підпису. Для отриманого документа одержувач знову обчислює його хеш-значення, після чого за допомогою відкритого ключа підписувача дешифрує ЦП. Якщо хеші рівні – підпис справжній.

Управлінням ключами займаються центри сертифікації ключів (ЦСК), що забезпечують:

- доступ користувача до справжнього відкритого ключа іншого користувача;
- захист ключів від підміни зловмисником;
- організацію відкликання ключа у випадку його компрометації.

Сертифікат відкритого ключа – електронний документ, який засвідчує належність відкритого ключа фізичній або юридичній особі.

АЛГОРИТМ ЦИФРОВОГО ПІДПИСУ RSA

Для створення підпису повідомлення M спочатку необхідно за допомогою деякої хеш-функції обчислити хеш-значення $h(M)$. Далі за алгоритмом RSA генеруються ключі (e, n) і (d, n). ЦП повідомлення $h(M)$ буде мати вигляд: $S = h(M)^d \bmod n$.

Тепер кожен, хто має відкритий ключ підписувача повідомлення, може перевірити дійсність підпису. Для цього необхідно знайти результат хешування прийнятого повідомлення M за допомогою тієї самої хеш-функції $h'(M)$ та порівняти його із $s^e \bmod n = h(M)$. Якщо дайджести рівні – підпис дійсний.

Приклад 7.1

З використанням алгоритму RSA підписати та перевірити підпис повідомлення M, хеш-значення якого $h(M) = 88$.

Оберемо $p = 17$ і $q = 11$, тоді $n = p \cdot q = 17 \cdot 11 = 187$.

Обчислимо $\varphi(187) = (p - 1)\times(q - 1)=16 \cdot 10 = 160$.

Виберемо взаємно просте число e як відкритий ключ $e = 7$ та перевіримо виконання умов: $1 < 7 <160$, НСД(7, 160) =1.

Знайдемо закритий ключ $d = 23$ за розширеним алгоритмом Евкліда з рівняння $7d \equiv 1 \bmod 160$.

Обчислимо підпис за допомогою закритого ключа підписувача:

$$s = h(M)^d \bmod n = 88^{23} \bmod 187 = 11.$$

Для перевірки підпису повідомлення M одержувачу потрібно знову обчислити його хеш-значення $h(M) = 88$ та порівняти зі значенням, отриманим за допомогою відкритого ключа підписувача:

$$s^e = h(M)^d \bmod n = 88^{23} \bmod 187 = 11.$$

У цьому разі будемо вважати, що підпис справжній.

АЛГОРИТМ ЦИФРОВОГО ПІДПИСУ ЕЛЬ-ГАМАЛЯ

1. Визначення відкритого «y» секретного «x» ключів.

1.1. Вибір двох взаємно простих великих чисел p і q, $q < p$.

1.2. Вибір значення секретного ключа x, $x < p$.

1.3. Визначення значення відкритого ключа y з виразу:

$$y = q^x \bmod p .$$

2. Формування ЕЦП

2.1. Обчислення хеш-значення повідомлення M: $m = h(M)$; $1 < m < p - 1$.

2.2. Вибір випадкового числа k, $0 < k < p - 1$ і найбільший спільний дільник – НСД(k, p – 1) = 1.

2.3. Визначення значення a з виразу: $a = q^k \bmod p$.

2.4. Визначення значення b з виразу:

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

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

3. Аутентифікація повідомлення – перевірка дійсності підпису

3.1. Обчислення хеш-значення прийнятого відкритого тексту повідомлення M $m' = h(M)$ визначення b з виразу:

$$m' = (xa + kb) \bmod (p - 1).$$

3.2. Підпис вважається достовірним, якщо $a < p$, $m = m'$ і виконується умова:

$$y^a a^b \bmod p = q^{m'} \bmod p .$$

4. У якості процедури формування ЕЦП розглянемо наступний приклад (для зручності розрахунків у цьому прикладі використані числа малої розрядності)

4.1. Вибираємо просте число p і два випадкових числа q і x (q і $x < p$), $p = 11$, $q = 2$ і секретний ключ $x = 8$.

4.2. Обчислюємо значення відкритого ключа y $y = q^x \bmod p = 2^8 \bmod 11 = 3$.

4.3. Визначаємо хеш-значення вихідного повідомлення M (312) $m = h(M)$, у цьому прикладі приймаємо $m = 3$.

4.4. Вибираємо випадкове ціле число k, взаємно просте з $p-1$.

Приймаємо k=9, НСД(9,10) = 1.

4.5. Для формування ЕЦП обчислюємо елементи підпису a і b $a = q^k \bmod p = 2^9 \bmod 11 = 6$.

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

$$m = (xa + kb) \bmod (p-1);$$

$$3 = (8*6 + 9*b) \bmod 10;\ 9*b = -45 \bmod 10;$$

$$b = 5 .$$

Тобто цифровим підписом є пара чисел a = 6, b = 5.

Цифровий підпис $S = (a,b)$ і відкритий текст повідомлення M відправляються одержувачу. Для контролю цілісності повідомлення й вірогідності ЕЦП одержувач обчислює хеш-значення m' прийнятого відкритого тексту повідомлення M. Водночас відправник і одержувач використовує одну й ту саму хеш-функцію h.

Отримавши підписане повідомлення й відкритий ключ y = 3, одержувач для перевірки дійсності підпису перевіряє виконання умови:

$$y^a a^b \bmod p = q^{m'} \bmod p ;$$

$$3^6 * 6^5 \bmod 11 = 2^3 \bmod 11;$$

$$5668704 \bmod 11 = 8 \bmod 11;$$

$$8 \bmod 11 = 8 \bmod 11;$$

тому якщо умова виконується, то прийняте одержувачем повідомлення вважається справжнім.

Отже, процедура встановлення дійсності прийнятого повідомлення полягає в перевірці відповідності автентифікатора повідомлення.

Варто мати на увазі те, що кожен підпис за схемою Ель-Гамаля вимагає нового значення *k*. Випадкове значення *k* повинне зберігатися в секреті.

Контрольні питання

1. Перерахувати число параметрів у криптографічній системі Ель-Гамаля.
2. Перелічити секретні параметри системи Ель-Гамаля.
3. Перерахувати відкриті параметри системи Ель-Гамаля.
4. На якому досить важкому завданні з теорії чисел базується криптографічна система Ель-Гамаля?
5. Описати схему формування ЕЦП з використанням алгоритму Ель-Гамаля.
6. Описати схему перевірку ЕЦП з використанням алгоритму Ель-Гамаля.
7. Описати схему формування цифрового підпису із застосування алгоритму RSA.
8. Описати схему перевірки цифрового підпису із застосування алгоритму RSA.
9. Що спільного між звичайним та цифровим підписами? Чим вони різняться?
10. Які завдання дає змогу вирішити цифровий підпис?
11. У чому полягає принципова складність практичного застосування систем цифрового підпису?

РОБОТА ІЗ СИСТЕМОЮ GNU PRIVACY GUARD ІЗ ВИКОРИСТАННЯМ ОБОЛОНКИ KLEOPATRA

GNU Privacy Guard, GnuPG – вільно поширюване програмне забезпечення, що використовує криптографію з відкритим ключем. Перша версія проєкту, створена Вернером Кохом (Werner Koch) та профінансована німецьким урядом, вийшла у світ у 1999 році під ліцензією GNU General Public. Функції GnuPG дають змогу шифрувати та підписувати повідомлення за допомогою цифрового підпису, а також керувати списками відкритих ключів респондентів.

Звичним інтерфейсом для GnuPG є командний рядок, проте наразі існують різні зовнішні оболонки, які роблять доступною функціональність цієї програми через графічний інтерфейс користувача, наприклад «Kleopatra» для Windows або «GNU Privacy Assistant» (GPA) для Linux.

У GnuPG використовуються різні криптографічні алгоритми: симетричні шифри, шифрування з відкритим ключем і змішані (гібридні) алгоритми.

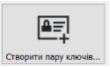
Гібридна (змішана, комбінована) криптосистема – це криптосистема, в якій розподіл ключів здійснюється за допомогою асиметричних криптоалгоритмів, а процес шифрування даних – за допомогою симетричних (отже, симетричний ключ використовується для шифрування даних, а асиметричний для шифрування самого симетричного ключа). Гібридні криптосистеми поєднують у собі зручність розподілу секретних ключів та високу швидкість шифрування.

Зазвичай під час гібридного шифрування створюється одноразовий секретний сеансовий ключ – це псевдовипадкове число, яке генерується на основі випадкових рухів миші, натискань клавіш клавіатури тощо. Такий ключ використовується лише один раз для шифрування повідомлення з використанням деякого надійного та швидкого симетричного алгоритму.

Сеансовий ключ зашифровується відкритим ключем одержувача та додається до шифротексту. Під час дешифрування процедури виконуються у зворотному порядку.

Створення пари ключів

Під час першого запуску «Kleopatra» (рис. 7.1) потрібно створити власну зв’язку ключів. Для цього необхідно виконати такі дії:

1) натиснути кнопку 

або скористатися меню «Файл → Створити пару ключів»;

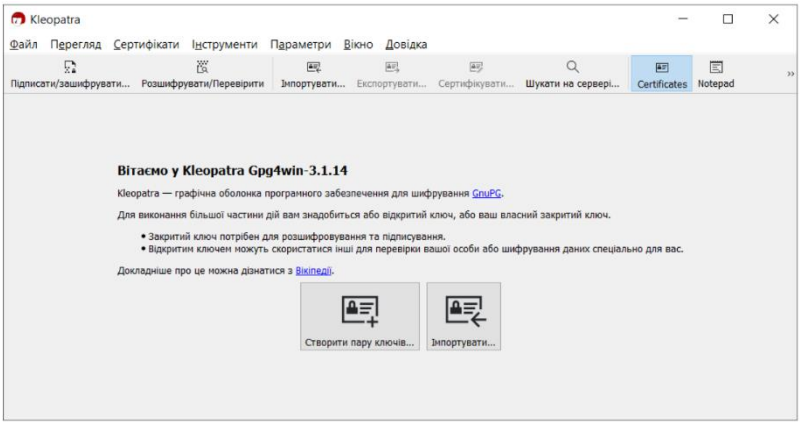


Рис. 7.1. Стартове вікно оболонки Kleopatra

2) у вікні Майстра створення ключів (рис. 7.2) потрібно ввести відомості про себе у відповідні поля (ім'я, електронну адресу); кнопка Додаткові параметри... дає змогу вибрати тип ключа, його довжину, строк дії тощо. Основною особливістю GnuPG є система ключів. У GnuPG користувач створює декілька ключів, причому кожен служить для окремої дії (і використовує різні алгоритми). Один із ключів, що створюється першим, є головним ключем, решта ключів йому підпорядковані – це підключі (субключі).

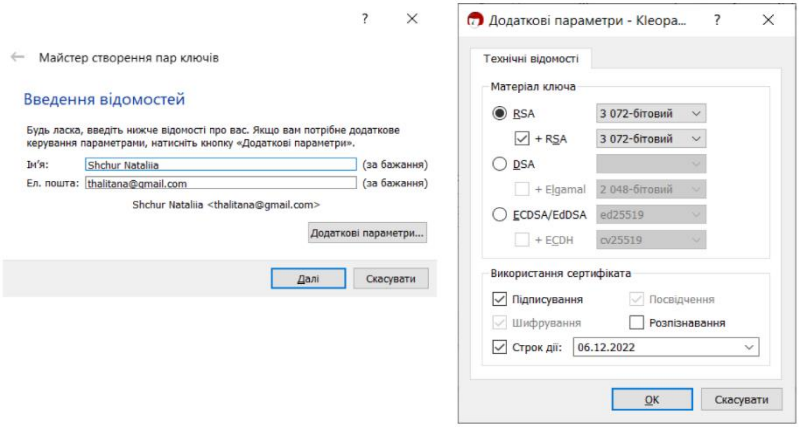


Рис. 7.2. Створення пари ключів за допомогою майстра

3) у наступному вікні необхідно натиснути «Створити та ввести пароль для захисту нового ключа» (рис. 7.3);

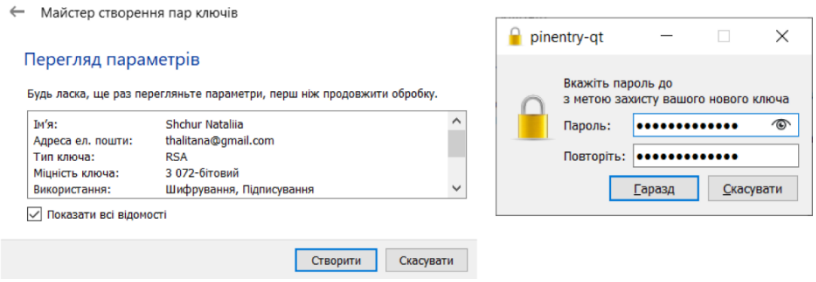


Рис. 7.3. Введення паролю для захисту нового ключа

4) у наступному вікні майстер має повідомити про успішне створення ключів (рис. 7.4), після чого потрібно натиснути кнопку «Завершити».

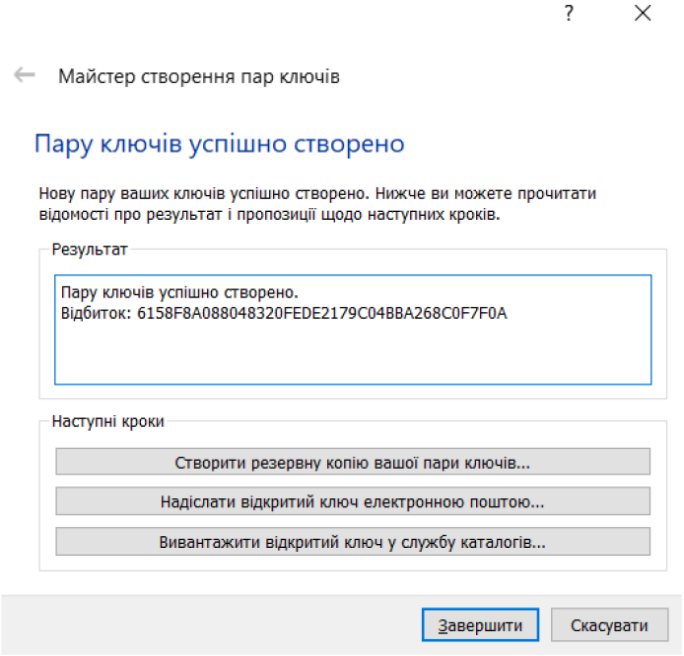


Рис. 7.4. Повідомлення про успішне створення ключів

Усі функції управління ключами здійснюються у вікні «Kleopatra» (рис. 7.5), в якому висвітлюються всі ключі, створені користувачем для власного користування, а також усі імпортовані публічні ключі його кореспондентів.

Ключі зберігаються у зашифрованій формі у вигляді двох файлів, які називаються зв'язками ключів («keyrings»). Ці файли записуються у папках на диску відповідно до поточних налаштувань.

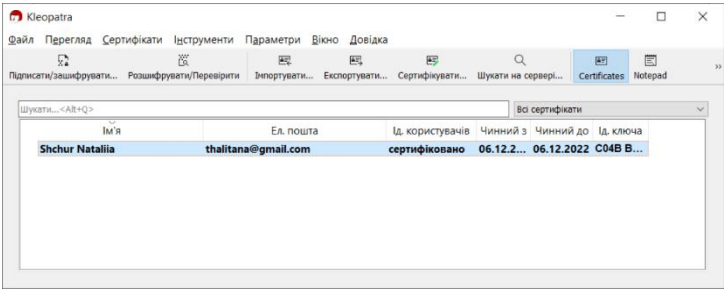
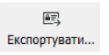


Рис. 7.5. Список наявних ключів у вікні оболонки Kleopatra

Експорт ключів

До початку обміну повідомленнями з іншими користувачами GPG варто обмінятись з ними публічними ключами.

Для експорту ключа потрібно:

1) у вікні «Kleopatra» натиснути кнопку 

або у контекстному меню електронного ключа вибрати пункт «Експортувати», або використати меню «Файл → Експортувати»;

2) обрати папку для збереження ключа, ввести його ім'я та натиснути «Зберегти».

Імпорт ключів

Імпортувати відкриті ключі інших користувачів можна, виконавши такі дії:

3) у вікні «Kleopatra» натиснути кнопку 

або у контекстному меню електронного ключа вибрати пункт Імпортувати, або використати меню «Файл → Імпортувати»;

4) обрати ключ на диску, ввести його ім'я та натиснути «Відкрити»;

5) також варто погодитися із перевіркою сертифіката ключа (рис. 7.6).

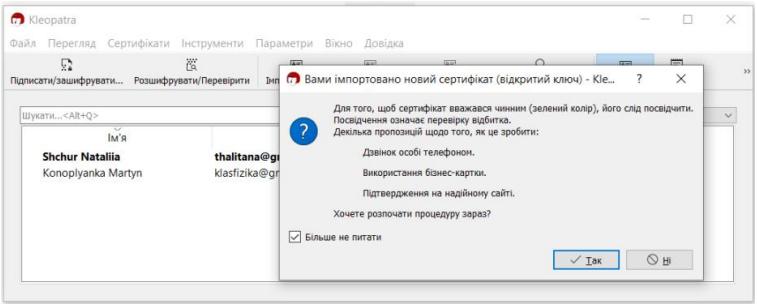


Рис. 7.6. Перевірка сертифіката ключа, що імпортується

Шифрування та (або) підписування файлів

Для шифрування та (або) підписування файла необхідно натиснути кнопку  або використати меню «Файл→Підписати / зашифрувати»;

Відкриється діалогове вікно «Підписати / зашифрувати файли» (рис. 7.7), у якому потрібно обрати необхідну дію та обрати відкриті ключі одержувача(-ів) повідомлення, натиснувши на піктограму  .

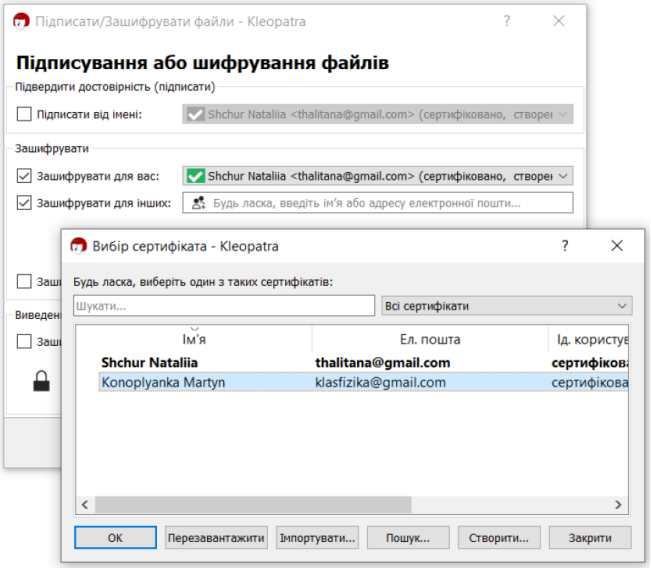


Рис. 7.7. Діалогове вікно Підписати / зашифрувати файли

Підписати файл за допомогою свого відкритого ключа дає змогу опція:

Також існує можливість виконати дві описані вище операції одночасно.

Розшифрування та (або) перевірка підпису файлів

Для розшифрування / перевірки цифрових підписів файлів використовуються кнопка 

та пункт меню «Файл → Розшифрувати/Перевірити».

Під час розшифрування на екрані з'явиться вікно перевірки пароля. Файл буде розшифрований після введення правильного пароля за умови, що його було зашифровано з використанням відкритого ключа отримувача (рис. 7.8). Очевидно також, що розшифрування файла можливе тільки за умов наявності у середовищі вікна «Kleopatra» закритого ключа отримувача. Розшифрованому файла автоматично присвоюється назва файла-оригіналу (файла, який було зашифровано).

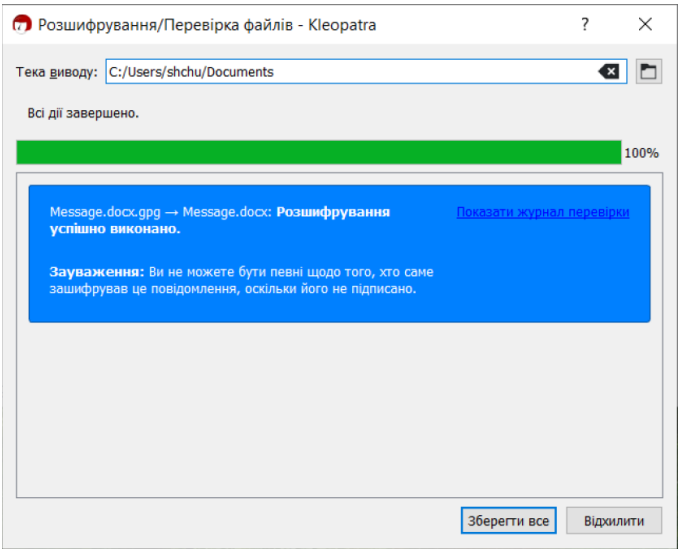


Рис. 7.8. Вікно розшифрування файла

Якщо файл має підпис, на екрані з'являється вікно з повідомленням, яке містить назву файла, відомості про особу, яка підписала файл, дату

і час накладання підпису та позначку, чи залишається підпис дійсним (рис. 7.9).

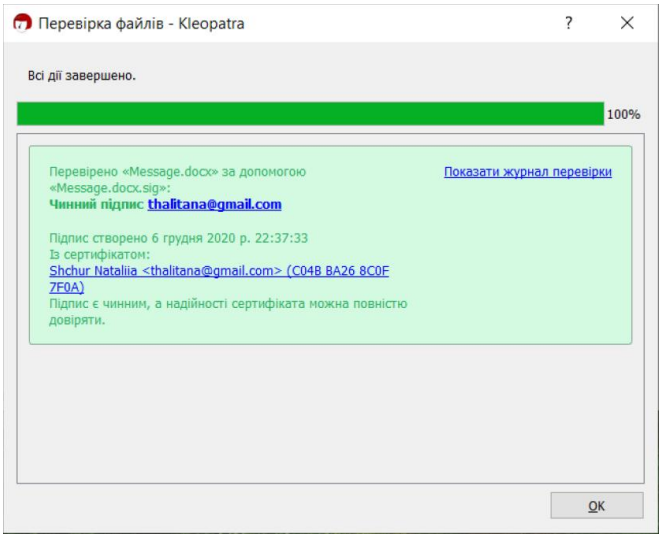


Рис. 7.9. Вікно перевірки підпису

Доступ до функцій GPG

Для забезпечення зручного виконання операцій шифрування, підписування, дешифрування, перевірки підпису тощо у контекстному меню файла (рис. 7.10) можна обрати «Sign and encrypt» або «More GpgEX option».

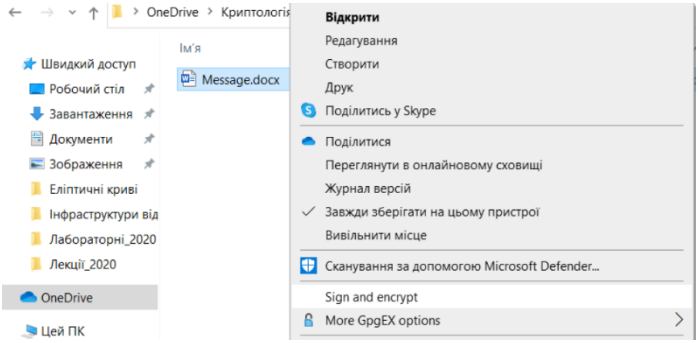


Рис. 7.10. Вибір у контекстному меню документа команд GPG

ЗАВДАННЯ ДО ЛАБОРАТОРНОЇ РОБОТИ

Завдання 1

Виконати створення та перевірку ЦП повідомлення згідно з варіантом (визначається номером студента у журналі: непарний – 1 варіант, парний – 2 варіант). Усі кроки алгоритму описати у звіті.

1. З використанням алгоритму RSA створіть та перевірте підпис повідомлення, якщо його хеш $h(M) = 7$, а параметри $p = 13$ та $q = 17$. Самостійно оберіть відкритий ключ e та обчисліть закритий ключ d.

За алгоритмом Ель-Гамаля виконайте формування та перевірку підпису повідомлення, якщо його хеш $h(M) = 8$, а параметри $p = 23$ та $g = 5$. Оберіть закритий ключ x, сесійний ключ k та обчисліть відкритий ключ y.

2. З використанням алгоритму RSA створіть та перевірте підпис повідомлення, якщо його хеш $h(M) = 6$, а параметри $p = 11$ та $q = 13$. Самостійно оберіть відкритий ключ e та обчисліть закритий ключ d.

За алгоритмом Ель-Гамаля виконайте формування та перевірку підпису повідомлення, якщо його хеш $h(M) = 7$, а параметри $p = 19$ та $g = 3$. Оберіть закритий ключ x, сесійний ключ k та обчисліть відкритий ключ y.

Завдання 2

Виконати завдання у системі GPG та додати до звіту скріншоти вікна GPG на кожному кроці: створення ключів, експортування / імпортування ключів, шифрування / підписування, дешифрування / перевірки підпису, а також скріншот дешифрованого текстового повідомлення від викладача.

2.1. Створити ключі у діалоговому вікні «Kleopatra» на основі алгоритму RSA, довжиною 3072 біт. Заповнити поля «Ім'я» та «Елек. пошта» (латинськими літерами).

2.2. Експортувати свій публічний ключ у свою робочу папку. Відповідний файл повинен мати назву за шаблоном, наприклад Shchur Nataliia_0x8C0F7F0A_public.asc.

2.3. Відправити свій публічний ключ на пошту викладача.

2.4. Імпортувати публічний ключ викладача до середовища Kleopatra.

2.5. За допомогою текстового редактора створити файл, вказати у ньому своє прізвище, ім'я, по батькові. Присвоїти файлу назву Enc_N.docx, де N – номер студента за списком групи, впорядкованим за алфавітом (наприклад, Enc _12.docx).

2.6. Із використанням відкритого ключа викладача зашифрувати Enc_N.docx за допомогою GPG. Схема зашифрування повідомлення із використанням GnuPG представлена на рис. 7.11.

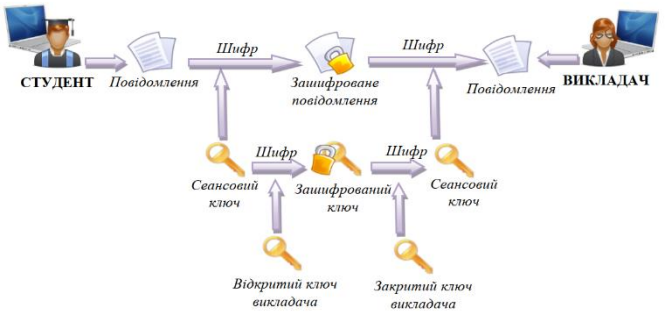


Рис. 7.11. Схема зашифровування повідомлення із використанням GnuPG

2.7. За допомогою текстового редактора створити файл, вказати у ньому свій варіант, курс, групу. Присвоїти файлу назву Enc_Sign_N.docx, де N – номер студента за списком групи, впорядкованим за алфавітом (наприклад, Enc_Sign _12.docx).

2.8. Із використанням свого ключа підписати Enc_Sign_N.docx та зашифрувати за допомогою ключа викладача. Схема алгоритму створення та перевірки підпису з використанням GnuPG представлена на рис. 7.12.

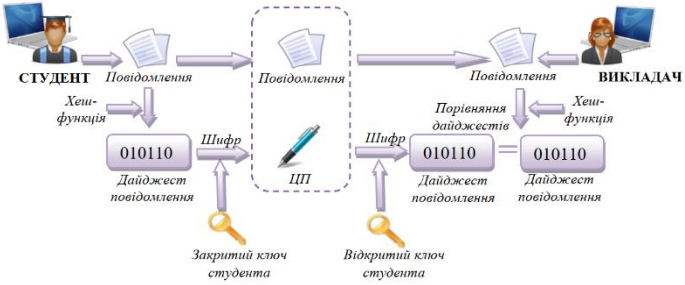


Рис. 7.12. Схема створення та перевірки підпису з використанням GnuPG

2.9. Відправити два файли викладачу: зашифрований Enc_N.docx та підписаний / зашифрований Enc_Sign_N.docx.

2.10. Отримати від викладача зашифроване повідомлення, підписане його цифровим підписом.

2.11. Розшифрувати повідомлення викладача та перевірити дійсність його підпису у системі GPG.

Контрольні питання

1. Для чого потрібен цифровий підпис?
2. Дайте визначення поняттям «хешування», «хеш-функція».
3. Опишіть схему створення і перевірки ЦП.
4. Який порядок використання відкритого та закритого ключів під час створення і перевірки ЦП?
5. Які схеми цифрового підпису існують?
6. Як здійснюється підпис RSA? Яка відмінність підпису RSA від шифру RSA?
7. Як здійснюється підпис Ель-Гамаля?
8. Як здійснюється перевірка на дійсність підпису Ель-Гамаля?

Лабораторна робота № 8
Інформаційна безпека на рівні операційної системи Windows

Мета роботи: ознайомитися з принципами побудови архітектури підсистеми безпеки сучасних операційних систем; вивчити моделі безпеки операційної системи Windows та здобути практичні навички використання засобів забезпечення її безпеки.

Теоретичні основи

8.1. Аналіз захищеності сучасних операційних систем

Під час оцінки ступеня захищеності операційних систем діє нормативний підхід, згідно з яким сукупність завдань, що виконується системою безпеки, повинна задовольняти певні вимоги. Їх перелік визначається загальноприйнятими стандартами, наприклад, TCSEC («Помаранчева книга» [7]) або «Загальні критерії оцінки безпеки інформаційних технологій». В Україні також діють подібні критерії, які визначені в НД ТЗІ 2.5-004-99 «Критерії оцінки захищеності інформації в комп'ютерних системах від несанкціонованого доступу (НСД)». Такі стандарти складають основу політики безпеки системи передбачає відповіді на питання: яку інформацію захищати, якого виду загрози можуть бути реалізовані в системі та які саме засоби планується використовувати для захисту від кожного типу атак.

Сьогодні до сучасних популярних операційних систем прийнято відносити два сімейства: Windows та Linux. Так, наприклад, зі сторони сімейства Windows після еволюції від однокористувацької моделі до багатокористувацької розробники операційної системи приділили серйозну увагу забезпеченню безпеки роботи користувачів. Це підтверджується категоріями, присвоєними різним версіям цієї операційної системи за тими чи іншими міжнародними і національними критеріями оцінки безпеки. Так, за класифікацією «Помаранчевої книги» ОС Windows NT 4 ще в 1999 році отримала клас безпеки С2, за стандартом ISO/IEC 15408 Common Criteria for Information Technology Security Evaluation (Загальні критерії оцінки безпеки інформаційних технологій) клієнтські і серверні версії від Windows 2000 до Windows 10, від Windows Server 2008 до Windows Server 2013 отримали рівень безпеки EAL4+. А взагалі, необхідно відзначити, що обидва сімейства операційних систем переважно задовольняють вимоги класу С2 TCSEC, згідно з якими система повинна забезпечувати:

- безпечний вхід у систему, який забезпечує точну ідентифікацію користувачів ОС і надає їм можливість доступу до ресурсів комп'ютера тільки після проходження процедури аутентифікації. У Windows за ідентифікацію та аутентифікацію користувачів відповідають процеси Winlogon.exe і Lsass.exe;

- управління доступом – надання користувачам можливості захисту приналежних їм даних, що дає змогу власнику ресурсу (файла, розділу реєстру, об'єкта ядра та ін.) визначити, хто має право на доступ до ресурсу, а також уточнити суть цих прав (читання, зміна, запуск тощо). Під час використання дискреційної моделі доступу для ущільнення матриці доступу власник може наділяти правами, які надають різні види доступу до об'єкта, і окремого користувача, і групи користувачів. Безпечний доступ в ОС Windows реалізується за допомогою компонента «Security Reference Monitor» виконавчої системи Ntoskrnl.exe;

- системний аудит – здатність системи проводити докладний аудит дій, виконуваних користувачами і самою операційною системою;

- аудит безпеки, який дає змогу реєструвати всі події, що належать до питань безпеки. Ідентифікація користувачів під час входу в систему дає змогу прив'язувати всі події безпеки в системі до конкретного користувача. У Windows аудит підтримується SRM і Lsass.exe;

- захист об'єктів від повторного використання – здатність системи запобігати доступу користувача до інформаційних ресурсів, із якими до цього працював інший користувач, тобто система не дозволяє користувачам переглядати дані, видалені іншим користувачем, а також не дозволяє звертатися до пам'яті, яка раніше була використана, а потім звільнена іншим користувачем. У Windows звільнена пам'ять очищується системним потоком обнулення сторінок, який працює під час простою системи (з нульовим пріоритетом);

- захист самої системи від зовнішнього впливу або нав'язування, як-от модифікація завантаженої системи або системних файлів, що зберігаються на диску.

8.2. Підсистема захисту в ОС Windows

Вивчення структури системи захисту допомагає зрозуміти особливості її функціонування. Незважаючи на слабку документованість ОС Windows за непрямими джерелами, можна розглядати особливості її функціонування.

Для захисту даних Windows використовує такі основні механізми:

- аутентифікація і авторизація користувачів;
- аудит подій в системі;
- шифрування даних;
- підтримка інфраструктури відкритих ключів;
- вбудовані засоби мережного захисту.

Ці механізми підтримуються такими підсистемами ОС Windows як LSASS (Local Security Authority Subsystem Service, локальна підсистема безпеки) – слідкує за процесом аутентифікації, доступом користувачів

та аудитом в системі, SAM (Security Account Manager, менеджер локальних записів безпеки) – забезпечує підтримку аутентифікаційної бази даних SAM, SRM (Security reference Monitor, монітор контролю безпеки) – перехоплює звернення користувачів до об'єктів захисту і передає їх на обробку LSASS. SRM виконується в режимі ядра ОС, Active Directory (служба каталогів), EFS (Encrypting File System, файлова система шифрування) та ін. Для більш детального розгляду, на рис. 8.1. схематично відображено структуру системи захисту ОС сімейства Windows, яка фактично складається з таких компонентів:

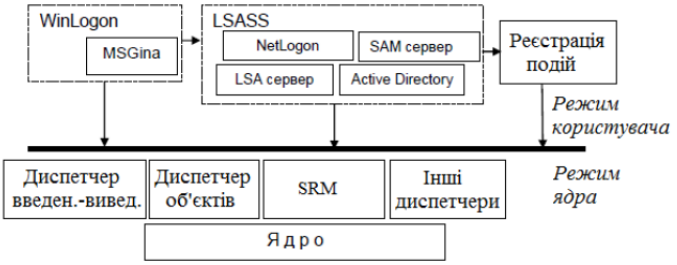


Рис. 8.1. Система захисту ОС Windows

Процедура реєстрації (Logon Processes), яка обробляє запити користувачів на вхід у систему. Вона запускає початкову інтерактивну процедуру діалогу із користувачем на екрані і віддалену процедуру входу, яка дає змогу віддаленим користувачам отримати доступ із робочої станції мережі до серверних процесів Windows. Процес Winlogon реалізований у файлі Winlogon.exe і виконується як процес режиму користувача. Стандартна бібліотека аутентифікації Gina реалізована у файлі Msgina.dll.

Локальний адміністратор безпеки (Local Security Authority, LSA), який гарантує, що користувач має дозвіл на доступ у систему. Цей компонент – центральний для системи захисту ОС Windows. Він формує маркери доступу, керує локальною політикою безпеки і надає користувачам аутентифікаційні послуги. LSA також контролює політику аудиту і веде журнал, у якому зберігаються повідомлення, що формуються диспетчером доступу. Основна частина функціональності реалізована в Lsasrv.dll.

Менеджер локальних записів безпеки (Security Account Manager, SAM) керує базою даних обліку користувачів, яка містить інформацію про всіх користувачів ОС, зокрема і про групи користувачів. Ця служба реалізована в Samsrv.dll і виконується у процесі LSASS.

Монітор контролю безпеки (Security Reference Monitor, SRM), який перевіряє, чи має користувач право на доступ до об'єкта, а також право на виконання тих дій, які він намагається зробити. Цей компонент

забезпечує легалізацію доступу і політику аудиту, що визначаються LSA. Він надає послуги для програм супервізорного режиму і режиму користувача та гарантує, що користувачі і процеси, які здійснюють спроби доступу до об'єкта, мають необхідні права. Також він формує повідомлення служби аудиту, коли це необхідно. Це компонент ядра системи: Ntoskrnl.exe.

Усі компоненти активно використовують базу даних LSASS, що містить параметри політики безпеки локальної системи, яка зберігається в розділі HKLM\SECURITY реєстру.

Як уже зазначалося раніше, захист об'єктів і аудит дій з ними в ОС Windows організовані на основі виборчого (дискреційного) доступу, коли права доступу (читання, запис, видалення, зміна атрибутів) суб'єкта до об'єкта відкрито задаються у спеціальній матриці доступу. Для укрупнення матриці користувачі можуть об'єднуватися в групи. Взагалі необхідно відзначити, що реалізація моделі дискреційного доступу пов'язана з функціонуванням SRM, який, згідно з описом, забезпечує також управління ролевим і привілейованим доступом. Під час спроби суб'єкта (одного з потоків процесу, запущеного від його імені) отримати доступ до об'єкта вказується, які операції користувач збирається виконувати з об'єктом. Якщо подібний тип доступу дозволений, потік отримує специфікатор (дескриптор безпеки) об'єкта і всі потоки процесу можуть виконувати операції з ним. Подібна схема доступу, очевидно, вимагає аутентифікації кожного користувача, який отримує доступ до ресурсів та його надійну ідентифікацію в системі, а також механізмів опису прав користувачів і груп користувачів у системі, опису та перевірки дискреційних прав доступу користувачів до об'єктів. Тому в наступному підрозділі розглянемо, як в ОС Windows організована ідентифікація, аутентифікація та авторизація користувачів.

8.2.1. Ідентифікація та аутентифікація користувача. Вхід у систему

Усі чинні в системі суб'єкти (користувачі, групи, локальні комп'ютери, домени) ідентифікуються у Windows не за іменами, унікальність яких не завжди вдається досягти, а за ідентифікаторами безпеки (Security Identifiers, SID). SID являє собою числове значення змінної довжини:

S – R – I – S0 – S1–...– Sn – RID;

S – незмінний ідентифікатор строкового подання SID;

R – рівень ревізії (версія). На сьогодні 1.

I – (identifier-authority) ідентифікатор повноважень. Являє собою 48-бітний рядок, що ідентифікує комп'ютер або мережу, який(а) видав SID об'єкту. Можливі значення:

- 0 (SECURITY_NULL_SID_AUTHORITY) – використовуються для порівнянь, коли невідомі повноваження ідентифікатора;

- 1 (SECURITY_WORLD_SID_AUTHORITY) – застосовуються для конструювання ідентифікаторів SID, які представляють усіх користувачів. Наприклад, ідентифікатор SID для групи Everyone (Усі користувачі) – це S-1-1-0;

- 2 (SECURITY_LOCAL_SID_AUTHORITY) – використовуються для побудови ідентифікаторів SID, що представляють користувачів, які входять на локальний термінал;

- 5 (SECURITY_NT_AUTHORITY) – сама операційна система. Тобто цей ідентифікатор випущений комп'ютером або доменом.

Sn – 32-бітові коди (кількістю 0 і більше) субагентів, яким було передано право видати SID. Значення перших підлеглих повноважень загальновідомо.

Вони можуть мати значення:

- 5 – ідентифікатори SID присвоюються сеансам реєстрації для видачі прав будь-якому додатку, що запускається під час певного сеансу реєстрації. У таких ідентифікаторах SID перші підлеглі повноваження встановлені як 5 і приймають форму S-1-5-5-x-y;

- 6 – коли процес реєструється як служба, він отримує спеціальний ідентифікатор SID у свій маркер для позначення даної дії. Цей ідентифікатор SID має підпорядковані повноваження 6 і завжди буде мати вигляд S-1-5-6;

- 21 (SECURITY_NT_NON_UNIQUE) – позначають ідентифікатор SID користувача та ідентифікатор SID комп'ютера, які не є унікальними в глобальному масштабі;

- 32 (SECURITY_BUILTIN_DOMAIN_RID) – позначають вбудовані ідентифікатори SID. Наприклад, відомий SID для вбудованої групи адміністраторів S-1-5-32-544;

- 80 (SECURITY_SERVICE_ID_BASE_RID) – позначають ідентифікатор SID, який належить службі.

Інші підлеглі повноваження ідентифікатора спільно позначають домен або комп'ютер, який видав ідентифікатор SID.

RID – 32-бітний відносний ідентифікатор. Він є ідентифікатором унікального об'єкта безпеки в області, для якої був визначений SID. Наприклад, 500 – позначає вбудований обліковий запис Administrator, 501 – позначає вбудований обліковий запис Guest, а 502 – RID для квитка на отримання квитків протоколу Kerberos [8].

Під час генерації SID Windows використовує генератор випадкових чисел, щоб забезпечити унікальність SID для кожного користувача. Зокрема, якщо видалити користувача в системі, а потім створити його під тим самим ім'ям, то SID створеного користувача буде вже іншим. Як приклад, для деякого довільного користувача SID може виглядати так:

S-1-5-21-1690090054-2308632580-4048739682-1000

Визначеним користувачам і групам Windows видає характерні SID, що складаються з SID комп'ютера або домену та зумовленого RID. У таблиці 8.1 наведено перелік деяких загальновідомих SID.

Таблиця 8.1

Загальновідомі SID Windows

SID	Назва	Опис
S-1-1-0	Всі	Група, в яку входять усі користувачі
S-1-5-2	Мережа	Група, в яку входять усі користувачі, що зареєструвалися в системі з мережі
S-1-5-7	Анонімний вхід	Група, в яку входять усі користувачі, що увійшли в систему анонімно
S-1-5-домен-500	Адміністратор	Обліковий запис адміністратора системи. За замовчуванням цей запис забезпечує повний контроль системи
S-1-5-домен-501	Гість	Обліковий запис користувача-гостя

Повний список загальновідомих SID можна подивитися в документації Platform SDK. Дізнатися SID конкретного користувача в системі, а також SID груп, у які він включений, можна, використовуючи консольну команду **whoami:**

whoami /user або **whoami /groups.**

Відповідність імені користувача і його SID можна відстежити також у ключі реєстру HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\ ProfileList.

Згідно з політикою безпеки для доступу до комп'ютера користувач повинен пройти процедуру аутентифікації. Ця процедура ініціюється комбінацією клавіш «CTRL+ALT+DEL». Ця комбінація клавіш, відома як SAS (secure attention sequence), завжди перехоплюється драйвером клавіатури, який викликає при цьому справжню (а не «троянського коня») програму аутентифікації. Процес користувача не може сам перехопити цю комбінацію клавіш або відмінити чи скасувати її обробку драйвером. Кажучи мовою стандартів, у системі реалізована функціональність захищеного каналу (trusted path functionality). Ця особливість відповідає вимогам захисту рівня В «Помаранчевої книги».

Процедурою аутентифікації користувача в системі управляє програма WinLogon.exe, яка за допомогою інтерактивної процедури відображає початковий діалог із користувачем на екрані. Процес WinLogon активно взаємодіє з бібліотекою GINA (Graphical Identification and Authentication – графічною бібліотекою ідентифікації і аутентифікації).

Одержавши ім’я і пароль користувача від GINA, WinLogon викликає модуль LSASS для аутентифікації цього користувача. У разі успішного входу в систему Winlogon отримує з реєстру профіль користувача, визначає тип оболонки і запускає її.

Комбінація SAS може бути одержана системою не лише на етапі входу користувача в систему. Якщо ж користувач уже увійшов до системи, то після натиснення клавіш «CTRL+ALT+DEL» він отримує такі можливості: подивитися список активних процесів, ініціювати перезавантаження або вимкнення комп’ютера, змінити свій пароль і заблокувати робочу станцію. Зі свого боку, якщо робоча станція заблокована, то після введення SAS користувач має можливість її розблокування. Іноді може бути здійснене примусове виведення користувача з системи з подальшим входом у неї адміністратора.

Після аутентифікації користувача процесом Winlogon усі процеси, запущені від імені цього користувача, будуть ідентифікуватися спеціальним об’єктом, званим маркером доступу (access token). Під час формування маркера використовуються ключі SECURITY і SAM реєстру. Перший ключ визначає загальну політику безпеки, а другий ключ містить інформацію про захист для індивідуальних користувачів. Якщо процес користувача запускає дочірній процес, то його маркер успадковується, тому маркер доступу уособлює користувача для системи в кожному запущеному від його імені процесі.

Основні елементи маркера представлені на рис. 8.2.

SID користувача	SID1 ... SIDn Ідентифікатори груп користувача	DACL за замовчуванням	Привілеї	Інші параметри

Рис. 8.2. Узагальнена структура маркера доступу

Маркер доступу містить ідентифікатор доступу самого користувача та всіх груп, у які він включений. У маркер включений також DACL за замовчуванням – список дискреційного контролю доступу, який приєднується до створюваних користувачем об’єктів.

Ще одна важлива для визначення прав користувача в системі частина маркера – список його привілеїв, призначення і відкликання яких є прерогативою локального адміністратора безпеки LSA. **Привілеї** – це права довіреного об’єкта на здійснення будь-яких дій відносно всієї системи. У таблиці 8.2 перераховані деякі привілеї, які можуть бути надані користувачеві.

Управління привілеями користувачів здійснюється в оснащенні «*Групова політика*», розділ *Конфігурація Windows/Локальні політики/ Призначення прав користувача*.

Таблиця 8.2

Привілеї, якими можуть бути наділені користувачі

Ім'я та ідентифікатор привілею	Опис привілею
SeIncreaseBasePriorityPrivilege (Збільшення пріоритету диспетчерування)	Користувач, що володіє цим привілеєм, може змінювати пріоритет диспетчерування процесу за допомогою інтерфейсу Диспетчера завдань
SeLockMemoryPrivilege (Закріплення сторінок у пам'яті)	Процес отримує можливість зберігати дані фізичної пам'яті, не вдаючись до кешування даних у віртуальній пам'яті на диску
SeAuditPrivilege (Управління аудитом та журналом безпеки)	Користувач отримує можливість вказувати параметри аудиту доступу до об'єкта для окремих ресурсів, як-от файли, об'єкти Active Directory, ключі реєстру
SeTakeOwnershipPrivilege (Оволодіння файлами або іншими об'єктами)	Користувач отримує можливість ставати власником будь-яких об'єктів безпеки системи, включно з об'єктами Active Directory, файлами і папками NTFS, принтерами, розділами реєстру, службами, процесами і потоками
SeShutdownPrivilege (Завершення роботи системи)	Користувач отримує можливість завершувати роботу операційної системи на локальному комп'ютері
SeChangeNotifyPrivilege (Обхід перехресної перевірки)	Використовується для обходу перевірки дозволів для проміжних каталогів під час проходження багаторівневих каталогів

Щоб подивитися привілеї користувача, можна також використовувати команду **«whoami»:**

whoami /all

Інші параметри маркера мають інформаційний характер і визначають, наприклад, яка підсистема створила маркер, унікальний ідентифікатор маркера безпеки, час його дії. Необхідно також відзначити можливість створення обмежених маркерів (restricted token), які відрізняються від звичайних тим, що з них видаляються деякі привілеї та його SID-ідентифікатори перевіряються тільки на заборонні правила.

Принцип мінімальних привілеїв рекомендує виконання всіх операцій із мінімальними привілеями, необхідними для досягнення результату. Це дає змогу зменшити втрати від спроб навмисного збитку й уникнути випадкових втрат даних. Наприклад, користувачу не рекомендується реєструватися як адміністратор системи без необхідності. Отже, обмежені маркери використовуються для процесів, які підміняють клієнта і виконують небезпечний код.

Створити обмежений маркер можна програмно, використовуючи API-функцію CreateRestrictedToken, а можна запустити процес з обмеженим маркером, використовуючи пункт контекстного меню Windows «Запуск від імені іншого користувача» (рис. 8.3).

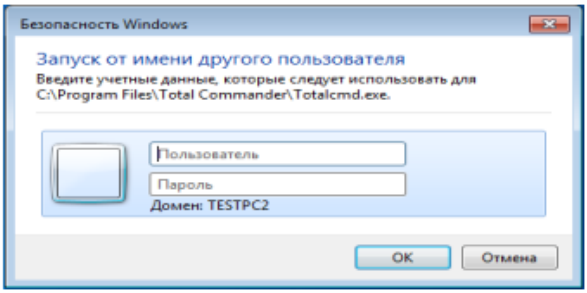


Рис. 8.3. Запуск процесу з обмеженим маркером

Також необхідно відзначити, що маркер доступу може бути створений не тільки при первинному вході користувача в систему. Windows надає можливість запуску процесів від імені інших користувачів, створюючи для цих процесів відповідний маркер. Для цих цілей можна використовувати:

- API-функції CreateProcessAsUser, CreateProcessWithLogon;
- віконний інтерфейс (рис. 8.3), який ініціюється при виборі пункту контекстного меню «Запуск від імені іншого користувача»;
- консольну команду «runas»:

runas /user: username program,

де «username» – ім'я облікового запису користувача, який буде використаний для запуску програми в форматі користувач@домен або домен\користувач;

«program» – команда або програма, яка буде запущена за допомогою облікового запису, зазначеного в параметрі /user.

У будь-якому варіанті запуску процесу від імені іншого облікового запису потрібно задати його пароль.

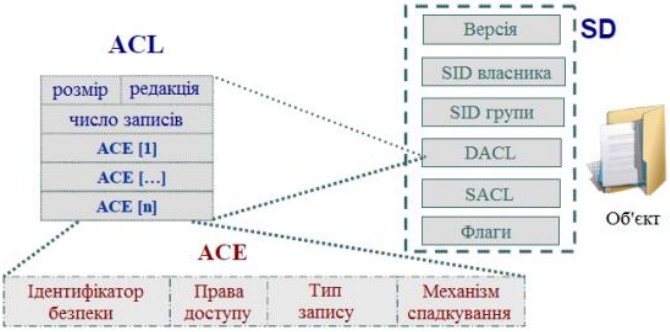


Рис. 8.4. Структура дескриптора безпеки об'єкта ОС Windows

8.2.2. Захист об'єктів системи

Маркер доступу ідентифікує суб'єктів-користувачів системи. З іншого боку, кожен об'єкт системи, що потребує захисту, містить опис прав доступу до нього користувачів. Для цих цілей використовується дескриптор безпеки (Security Descriptor, SD). Кожному об'єкту системи, включно з файлами, принтерами, мережними службами, контейнерами Active Directory та ін., присвоюється дескриптор безпеки, який визначає права доступу до об'єкта і містить такі основні атрибути (рис. 8.4):

- SID власника, що ідентифікує обліковий запис користувача – власника об'єкта;
- список дискреційного контролю доступу (Discretionary Access Control List, DACL), який дає змогу відстежувати права та обмеження, встановлені власником цього об'єкта. DACL може бути змінений користувачем, який вказаний як поточний власник об'єкта.
- список системного контролю доступу (System Access Control List, SACL), що визначає перелік дій над об'єктом, які підлягають аудиту;
- прапорці, які визначають атрибути об'єкта.

Авторизація Windows заснована на зіставленні маркера доступу суб'єкта з дескриптором безпеки об'єкта. Керуючи властивостями об'єкта, адміністратори можуть встановлювати дозволи, призначати право володіння і відстежувати доступ користувачів.

Список дискреційного контролю доступу містить набір записів ACE (Access Control Entries). У DACL кожен ACE складається з чотирьох частин: у першій зазначаються користувачі або групи, до яких належить цей запис, у другій – права доступу, а третя інформує про те, надаються ці права чи відбираються. Четверта частина являє собою набір прапорців, що визначають, як цей запис буде успадковуватися вкладеними об'єктами (актуально, наприклад, для папки файлової системи, розділів реєстру).

Якщо список ACE в DACL порожній, до нього немає доступу ні в одного користувача (тільки у власника на зміну DACL). Якщо відсутній сам DACL у SD об'єкта, в такому разі всі користувачі мають повний доступ до нього.

Якщо який-небудь потік запросив доступ до об'єкта, підсистема SRM здійснює перевірку прав користувача, що запустив потік, на даний об'єкт, переглядаючи його список DACL. Перевірка здійснюється до появи дозвільних прав на всі запитані операції. Якщо зустрінеться забороняюче правило хоча б на одну запитану операцію, доступ не буде наданий.

Докладніше розглянемо приклад на рис. 8.5. Процес намагається отримати доступ до об'єкта з заданим DACL. У маркері процесу вказані SID користувача, який запустив його, а також SID груп, у які він входить. У списку DACL об'єкта присутні правила, що дають змогу здійснювати читання для користувача з SID = 100, і запис для групи з SID = 205. Однак

у доступі користувачу буде відмовлено, оскільки раніше зустрічається забороняюче запис правило для групи з SID = 201.

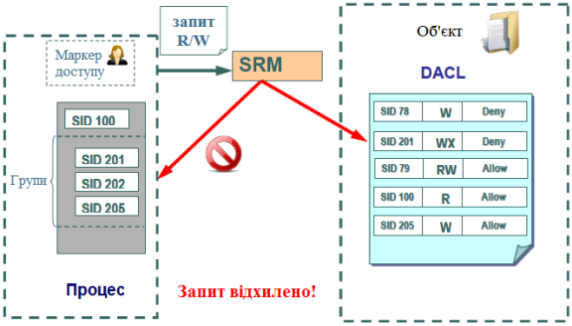


Рис. 8.5. Перевірка прав доступу користувача до об'єкта

Необхідно зазначити, що забороняюче правило поміщено у списку DACL на рисунку невипадково. Забороняючі правила завжди розміщуються перед дозвільними, тобто є домінуючими під час перевірки прав доступу.

Для визначення і перегляду прав доступу користувачів до ресурсів можна використовувати і графічні засоби контролю, і консольні команди. Стандартне вікно властивостей об'єкта файлової системи (диску, папки, файла) на вкладці Безпека (рис. 8.6) дає змогу переглянути поточні дозволи для користувачів і груп користувачів, редагувати їх, створювати нові або видаляти наявні.

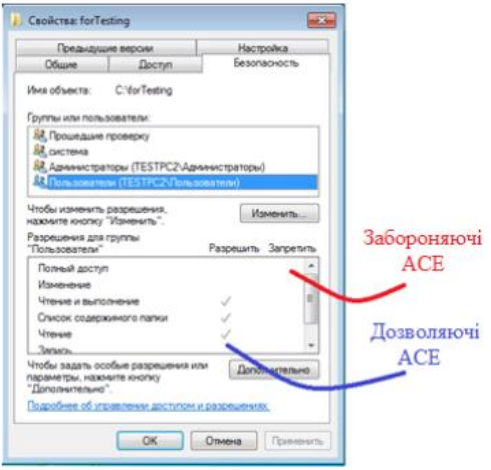


Рис. 8.6. GUI-інтерфейс Windows для зміни прав доступу до об'єктів

Під час визначення прав доступу до об’єктів можна задати правила їх успадкування в дочірніх контейнерах. У вікні додаткових параметрів безпеки на вкладці Дозволу під час вибору опції «Додавати дозволи, які успадковуються від батьківських об’єктів» (рис. 8.7) можна успадкувати дозволи і обмеження, задані для батьківського контейнера, поточного об’єкта.

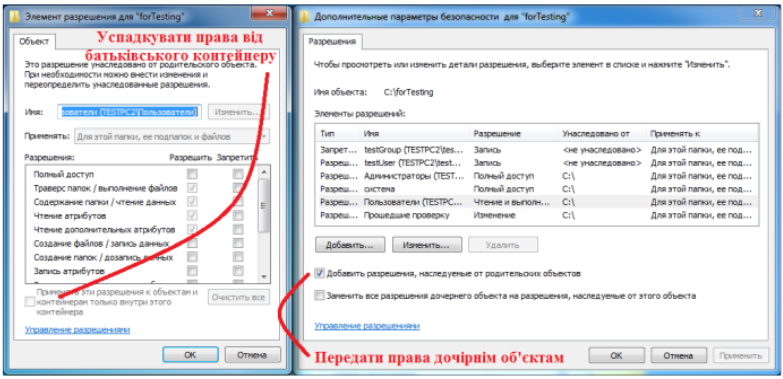


Рис. 8.7. Визначення параметрів успадкування прав доступу до об’єктів

Під час вибору опції «Застосовувати ці дозволи до об’єктів і контейнерів тільки усередині цього контейнера» дозволяється передача визначених для об’єкта-контейнера правил доступу його дочірнім об’єктам.

У цьому самому вікні на вкладці «Власник» можна дізнатися власника об’єкта і замінити його. Власник об’єкта має право на зміну списку його DACL, навіть якщо до нього заборонений будь-який тип доступу. Адміністратор має право ставати власником будь-якого об’єкта.

З урахуванням можливості входження користувача у різні групи і незалежності визначення прав доступу до об’єктів для груп і користувачів часто буває складно визначити кінцеві права користувача на доступ до об’єкта: потрібно переглянути забороняючі правила, визначені для самого об’єкта, для всіх груп, у які він включений, потім те саме зробити для дозвільних правил.

Автоматизувати процес визначення дозволених користувачеві видів доступу до об’єкта можна з використанням вкладки «Чинні дозволи» вікна додаткових параметрів безпеки об’єкта (рис. 8.8).

Для перегляду та зміни прав доступу до об’єктів у режимі командного рядка призначена команда «icacls».

ICACLS і’мя [/grant[:r] Sid:perm[...]] [/deny Sid:perm [...]] [/remove[:g|:d]] Sid[...]] [/T] [/C] [/L] [/Q] [/setintegritylevel Level:policy[...]]

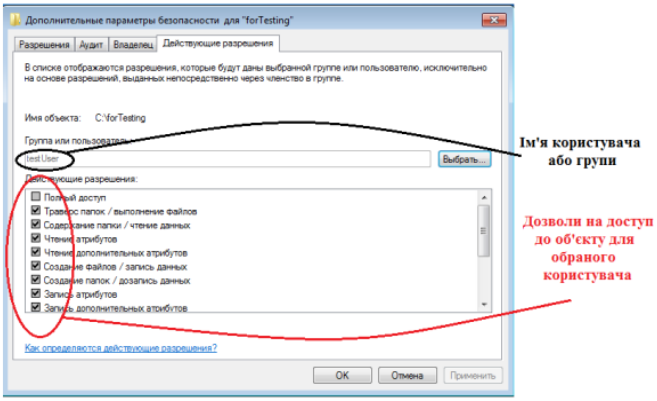


Рис. 8.8. Визначення ефективних прав доступу користувача (групи) до об'єкта

Призначення параметрів команди наведені в таблиці 8.3.

Таблиця 8.3

Параметри команди «icacls»

<ім'я>	Задає файл або папку, права доступу до якої необхідно переглянути / змінити (допустимо використовувати шаблони з символами * та ?)
/grant[:r] Sid:perm	Надання зазначених прав доступу користувача. З параметром *:r* ці дозволи замінюють усі раніше надані дозволи. Без параметра *:r* дозволи додаються до будь-яких раніше наданих дозволів
/deny Sid:perm	Відгук зазначених прав доступу користувача. Додається ACE відкликання для заявлених дозволів з видаленням цих дозволів у будь-якому представленні
/remove[:[g|:d]] Sid	Видалення всіх входжень SID в ACL. Із параметром :g видаляються всі входження наданих прав у цьому SID. Із параметром :d видаляються всі входження відкликаних прав у цьому SID
/setintegritylevel	Додавання ACE рівня цілісності до всіх відповідних файлів

Для вказівки прав, які додаються або віднімаються, використовуються такі значення:

F – повний доступ;
DE – видалення;
WD – запис;
RC – читання;
N – немає доступу.

Розглянемо кілька прикладів:

icacls c:\test – видасть список DACL для папки test.

icacls c:\test /deny ім'я_комп'ютера \ім'я_користувача:(WD) – заборонить запис до об'єкта для зазначеного користувача.

icacls c:\test /grant ім'я_комп'ютера \і'мя_групи:(F) – надасть повний доступ до папки c:\test і її підпапок усім членам зазначеної групи.

Для програмного перегляду і зміни списків DACL можна використовувати API-функції AddAccessAllowedAce, AddAccessDeniedAce, SetSecurityInfo.

Розглянуті способи роботи зі списком дискреційного доступу ілюструють реалізацію у Windows моделі довільного доступу. Але починаючи з Windows Vista, фірма Microsoft реалізувала елементи мандатного доступу для контролю доступу до об'єктів. За цей рівень забезпечення безпеки відповідає Windows Integrity Control (WIC). Концепція WIC вторить раніше згаданим принципам примусового (мандатного) управління доступом і заснована на побудові довірчих відносин між об'єктами і управлінні діями з ними користувачів на основі їх рівня довіри. Базовим поняттям WIC є рівень «цілісність об'єкта» (integrity level). WIC присвоює контрольованим об'єктам один з шести доступних рівнів цілісності:

1. Untrusted – анонімні процеси автоматично потрапляють у цю категорію.

2. Low – стандартний рівень під час роботи з інтернетом. Якщо браузер Internet Explorer запущений у захищеному режимі, всі файли і процеси, асоційовані з ним, призначаються в цю категорію. Деякі папки, як-от Temporary Internet Folder, також за замовчуванням наділяються *Низьким рівнем довіри*.

3. Medium – у цьому контексті працює більшість об'єктів. Ординарні користувачі отримують *Середній рівень*, і якщо не вказано який-небудь інший, тоді всім об'єктам присвоюється цей рівень доступу.

4. High – рівень, асоційований у системі з *Адміністраторами*. Об'єкти *Високого рівня* недоступні звичайним користувачам.

5. System – рівень для роботи ядра операційної системи та її служб.

6. Installer – вершина в ієрархії рівнів WIC. Його об'єкти можуть редагувати і видаляти файли всіх попередніх рівнів.

Контроль за рівнями цілісності під час доступу до об'єкта також здійснюється на основі правил ACE. Але це спеціалізовані ACE, які починаючи з ОС Windows Vista зберігаються у списку SACL дескриптора безпеки об'єкта поряд із правилами аудиту. Рівень цілісності користувача (процесу, що виконується від його імені) зберігається в його токені безпеки. Під час доступу процесу до об'єкта монітор безпеки порівнює рівень цілісності в токені з рівнем цілісності в дескрипторі об'єкта (у SACL). Система видає права доступу залежно від того, вищий чи нижчий рівень цілісності суб'єкта відносно об'єкта, а також залежно від прапорців

політики цілісності у відповідному ACE об’єкта. Рівні цілісності (IL) користувача описуються в його ідентифікаторі безпеки, а саме в його RID-частині:

SID = S-1-16-0x0 – рівень Untrusted;
SID = S-1-16-0x1000 – рівень Low;
SID= S-1-16-0x2000 – рівень Medium;
SID= S-1-16-0x3000 – рівень High;
SID= S-1-16-0x4000 – рівень системи.

Для зміни рівня цілісності об’єктів можна використовувати такі інструменти:

– уже розглянуту команду «icacls» із ключем /setintegitylevel. Наприклад, ось так можна присвоїти файлу низький (L) рівень цілісності:

icacls c:\forTesting /setintegritylevel L

– використовуючи спеціальні утиліти «Chml» («change mandatory label») для зміни рівня цілісності файлів та папок і «Regil» («Registry integrity levels») для роботи з рівнями цілісності ключів реєстру.

Змінити рівень цілісності процесу можна, наприклад, запустивши його утилітою psexec.exe з відповідним ключем. Ось як можна запустити блокнот із високим рівнем цілісності:

psexec –h notepad.exe

Очевидно, що змінювати рівень цілісності процесів, що виконуються, – потенційно небезпечна операція, тому її можуть запускати тільки процеси, у яких у маркері доступу встановлений привілей SeRelabelPrivilege.

Дізнатися, який рівень цілісності має процес можна, наприклад, запустивши утиліту «ProcessExplorer» з набору Sysinternals (рис. 8.9).

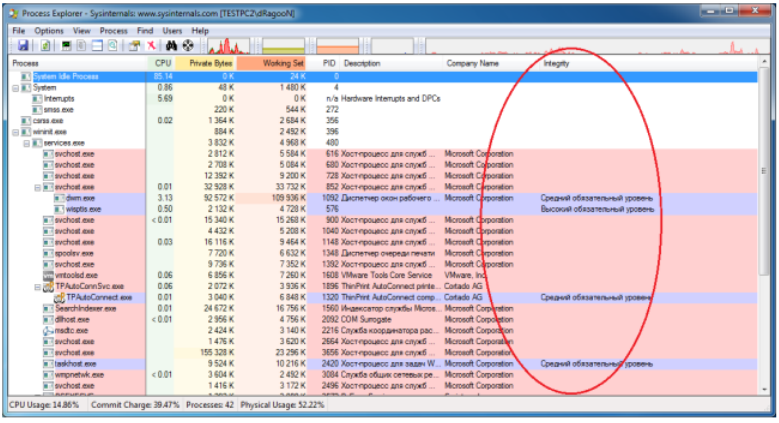


Рис. 8.9. Рівень цілісності запущених процесів в інтерфейсі ProcessExplorer

Необхідно зазначити, що контроль рівнів цілісності має більш високий пріоритет під час перевірки прав доступу до об'єкта перед дискреційної таблицею.

8.2.3. Підсистема аудиту

Навіть найкраща система захисту рано чи пізно буде зламана, тому виявлення спроб вторгнення – найважливіше завдання системи захисту. Основним інструментом виявлення вторгнень є аудит подій у системі, що є важливим елементом політики безпеки. Окремі дії користувачів протоколюються, а одержаний протокол використовується для виявлення вторгнень. ОС Windows веде аудит подій за 9 категоріями:

1. Аудит подій входу в систему.
2. Аудит управління обліковими записами.
3. Аудит доступу до служби каталогів.
4. Аудит входу в систему.
5. Аудит доступу до об'єктів.
6. Аудит зміни політики.
7. Аудит використання привілеїв.
8. Аудит відстеження процесів.
9. Аудит системних подій.

Розглянемо більш докладно, які події відстежує кожна з категорій.

Аудит подій входу в систему. Аудит спроб користувача увійти в систему з іншого комп'ютера або вийти з неї, за умови, що цей комп'ютер використовується для перевірки справжності облікового запису.

Аудит управління обліковими записами. Аудит подій, пов'язаних з управлінням обліковими записами на комп'ютері: створення, зміна або видалення облікового запису користувача або групи; перейменування, відключення або включення облікового запису користувача; встановлення або зміна пароля.

Аудит доступу до служби каталогів. Аудит подій доступу користувача до об'єкта каталогу Active Directory, для якого заданий власний список системного контролю доступу (SACL).

Аудит входу в систему. Аудит спроб користувача увійти в систему з комп'ютера або вийти з неї.

Аудит доступу до об'єктів. Аудит подій доступу користувача до об'єкта – наприклад, файла, папки, розділу реєстру, принтера і под., – для якого заданий власний список системного контролю доступу (SACL).

Аудит зміни політики. Аудит фактів зміни політик, призначення прав користувачів, політик аудиту або політик довірчих відносин.

Аудит використання привілеїв. Аудит спроб користувача скористатися наданим йому правом.

Аудит відстеження процесів. Аудиту таких подій, як-от активізація програми, завершення процесу, повторення дескрипторів і непрямий доступ до об'єкта.

Аудит системних подій. Аудит подій перезавантаження або вимикання комп'ютера, а також подій, які впливають на системну безпеку або на журнал безпеки.

Рішення про аудит конкретного типу подій безпеки приймаються відповідно до політики аудиту локальної системи. Політика аудиту, також звана локальною політикою безпеки (local security policy), є частиною політики безпеки, підтримуваної LSASS у локальній системі, і налаштовується за допомогою редактора локальної політики безпеки (Оснащення gpedit.msc, Конфігурація комп'ютера – Конфігурація Windows – Параметри безпеки – Локальні політики – Політика аудиту, рис. 8.10).

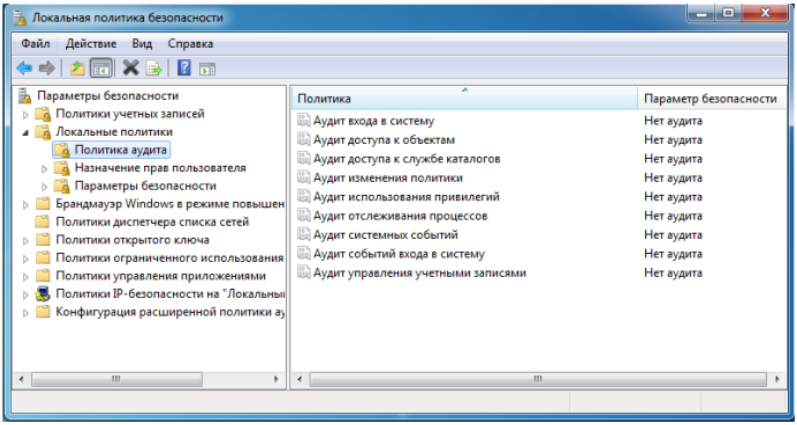


Рис. 8.10. Конфігурація політики аудиту редактора локальної політики безпеки

Для кожного об'єкта в SD міститься список SACL, що складається із записів ACE, які регламентують запис у журнал аудиту вдалих або невдалих спроб доступу до об'єкта. Ці ACE визначають, які операції, що виконуються над об'єктами конкретними користувачами або групами, підлягають аудиту. Інформація аудиту зберігається в системному журналі аудиту. Аудиту можуть підлягати і успішні, і невдалі операції. Подібно до записів ACE DACL правила аудиту об'єктів можуть успадковуватися дочірніми об'єктами. Процедура спадкування визначається набором прапорців, які є частиною структури ACE.

Налаштування списку SACL може бути здійснене у вікні додаткових властивостей об'єкта (пункт «Додатково», закладка «Аудит», рис. 8.11).

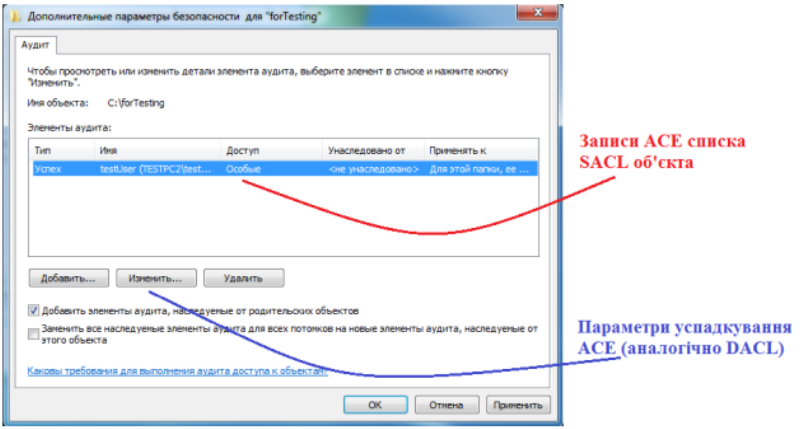


Рис. 8.11. Інтерфейс редагування правил аудиту для об'єкта

Для програмного перегляду і зміни списків SACL можна використовувати API-функції GetSecutityInfo і SetSecutityInfo.

Під час ініціалізації системи і зміни політики LSASS посилає SRM повідомлення, які інформують його про поточну політику аудиту. LSASS відповідає за прийом записів аудиту, які генеруються на основі подій аудиту від SRM, їх редагування і передачу Event Logger (реєстратору подій). SRM посилає записи аудиту LSASS через своє LPC-з'єднання. Після цього Event Logger заносить записи в журнал безпеки.

Починаючи з Windows Vista, підтримуються дві категорії журналів подій: **«журнали Windows»** і **«журнали додатків і служб».** Журнали Windows – реєструють загальносистемні події і ведуться самою ОС. Журнали додатків і служб – індивідуальні для конкретних типів додатків і компонентів (Internet Explorer, MediaCenter, PoerShell та ін.). Події аудиту записуються в журнали Windows таких типів (на прикладі Windows 7):

1. Журнал додатків. У журналі додатків містяться дані, що стосуються роботи додатків і програм.

2. Журнал безпеки. Журнал безпеки містить записи про події, як-от успішні і невдалі спроби доступу в систему, а також про події, що стосуються використання ресурсів.

3. Журнал системи. У журналі містяться події системних компонентів Windows. Наприклад, у журналі системи реєструються збої при завантаженні драйвера або інших системних компонентів під час запуску системи.

4. Журнал установки. Фіксує події, пов'язані зі встановленням або видаленням компонентів системи.

5. Журнал перенаправлення. Фіксує події, перенаправлені з сусідніх комп'ютерів.

Перегляд журналу безпеки здійснюється у вікні «Перегляд подій» (eventvwr.msc, рис. 8.12). Самі журнали зберігаються у файлах з розширенням evtx в папці %SystemRoot%\System32\Winevt\Logs\.

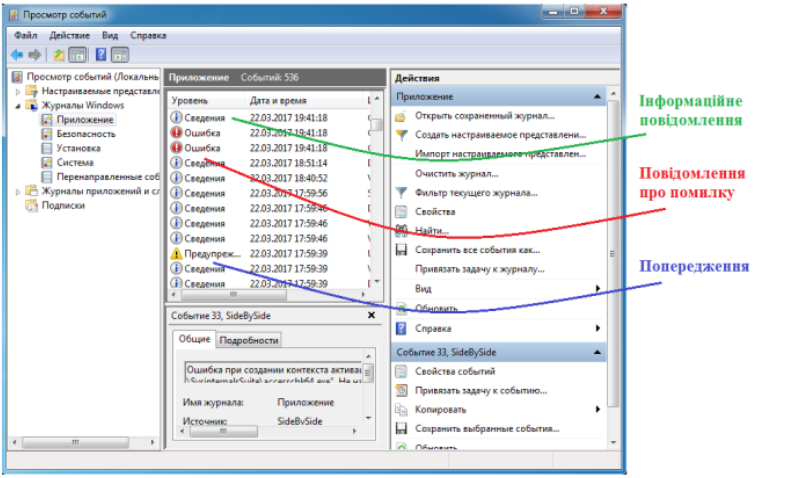


Рис. 8.12. Вікно Windows «Перегляд подій»

У журналі реєструються події різних типів:

Відомість – сигналізує про зміну в додатку або компоненті, наприклад, успішний доступ до ресурсу, запуск програми або служби.

Попередження – сигналізує про потенційно небезпечні події, що виникли в додатку або компоненті, які не заважають його роботі, але можуть стати причиною проблем у майбутньому.

Помилка – сигналізує про проблему, яка впливає на програму або компоненту.

Критична помилка – відповідає збою, критичному для додатка або компонента, після якого вони не можуть продовжувати роботу.

8.2.4. Файлова система шифрування

Починаючи з версії Windows 2000, в операційних системах лінійки Windows NT підтримується шифрування даних на розділах файлової системи NTFS з використанням файлової системи шифрування (Encrypted File System, EFS). Основна її перевага полягає в забезпеченні конфіденційності даних на дисках комп'ютера за рахунок використання надійних симетричних алгоритмів шифрування даних у режимі реального часу.

Для шифрування даних EFS використовує симетричний алгоритм шифрування (AES або DESX) з випадковим ключем для кожного файла (File Encryption Key, FEK). За замовчуванням дані шифруються у Windows 2000 і Windows XP за алгоритмом DESX, а в Windows XP з Service Pack 1 (або вище) та Windows Server 2003 – за алгоритмом AES. У версіях Windows, дозволених для експорту за межі США, драйвер EFS реалізує 56-бітний ключ шифрування DESX, тоді як у версії, що підлягає використанню тільки у США, і у версіях із пакетом для 128-бітного шифрування довжина ключа DESX дорівнює 128 бітам. Алгоритм AES у Windows використовує 256-бітові ключі.

Водночас для забезпечення секретності самого ключа FEK шифрується асиметричним алгоритмом RSA відкритим ключем користувача, результат шифрування FEK – Data Decryption Field, DDF – додається в заголовок зашифрованого файла (рис. 8.13). Такий підхід забезпечує надійне шифрування без втрати ефективності процесу шифрування: дані шифруються швидким симетричним алгоритмом, а для гарантії секретності симетричного ключа використовується асиметричний алгоритм шифрування.

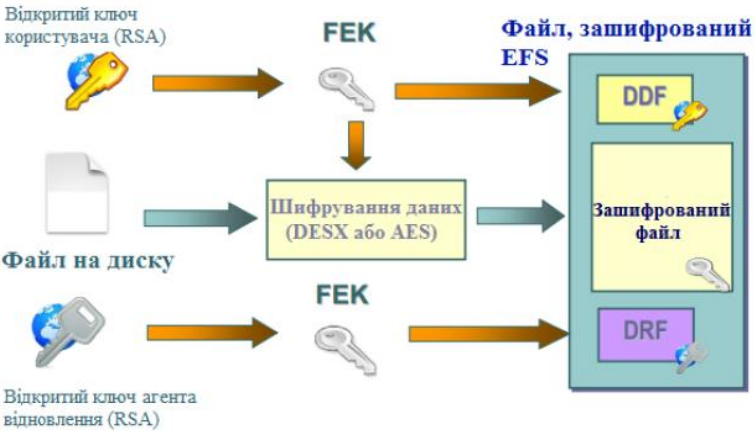


Рис. 8.13. Схема шифрування файлу в EFS

Для шифрування файлів із використанням EFS можна використовувати графічний інтерфейс або команду **«cipher».**

Графічний інтерфейс доступний у стандартному вікні властивостей об’єкта після натиснення кнопки «Додатково» (рис. 8.14). Зашифровані об’єкти у стандартному інтерфейсі Windows Explorer відображаються зеленим кольором, а під час спроби відкрити зашифрований файл іншим користувачем відбувається «Відмова в доступі».

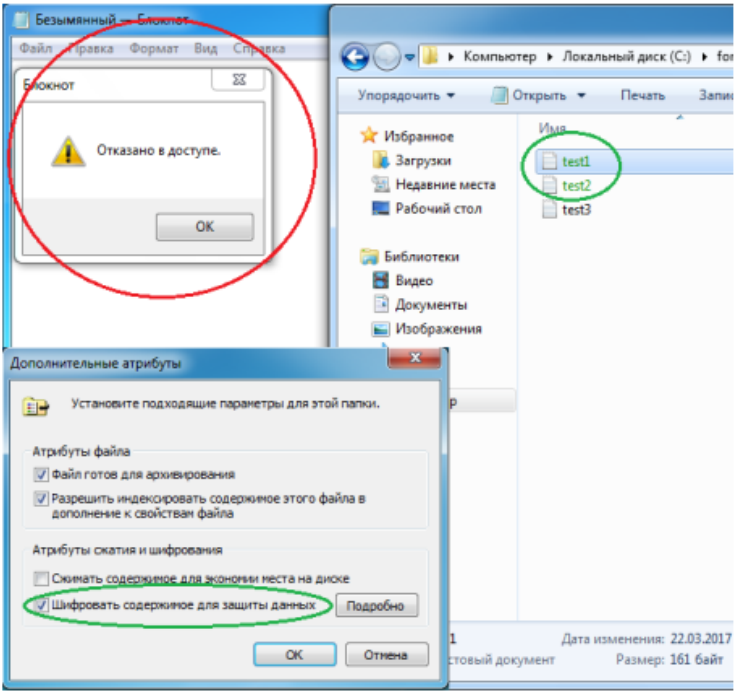


Рис. 8.14. Графічний інтерфейс шифрування файла з використанням EFS

Необхідно зазначити, що EFS дозволяє розділяти зашифрований файл між декількома користувачами. У цьому випадку FEK шифрується відкритими ключами всіх користувачів, яким дозволений доступ до файла, і кожен результат шифрування додається в DDF.

Шифрування файла з використанням EFS захищає файл комплексно: користувачу, який не має права на розшифрування файла, не дозволені зокрема операції видалення, перейменування і копіювання файла. Необхідно пам'ятати, що EFS є частиною файлової системи NTFS, і в разі копіювання захищеного файла авторизованим користувачем на інший том із файловою системою, яка не підтримує EFS (наприклад, FAT32), він буде розшифрований і збережений на цільовому томі у відкритому вигляді.

Консольна команда «cipher» може бути використана для шифрування / розшифрування файлів із командного рядка або в bat-сценарії.

cipher [{/e|/d}] [/s:каталог] [/a] [/i] [/f] [/q] [/h] [/k] [/u[/n]] [шлях [...]] | [/r:i'мя_файла_без_розширення]

Призначення параметрів команди наведені в таблиці 8.4.

Таблиця 8.4

Параметри команди «cipher»

/e	Шифрує вказані папки. Папки позначаються в такий спосіб, щоб файли, які будуть додаватися в папку пізніше, також шифрувалися
/d	Розшифровує вказані папки. Папки позначаються в такий спосіб, що файли, які будуть додаватися в папку пізніше, не будуть шифруватися
/s: каталог	Виконує обрану операцію над зазначеною папкою і всіма підпапками в ній
/a	Виконує операцію над файлами і каталогами
/i	Продовження виконання зазначеної операції навіть після виникнення помилок. За замовчуванням виконання «cipher» припиняється після виникнення помилки
/f	Виконання повторного шифрування або розшифрування вказаних об'єктів. За замовчуванням уже зашифровані або розшифровані файли пропускаються командою «cipher»
/k	Створення ключа шифрування файла для користувача, який виконав команду «cipher». Якщо використовується цей параметр, усі інші параметри команди «cipher» не враховуються
/u	Оновлення ключа шифрування файла користувача або ключа агента відновлення на поточні ключі до всіх зашифрованих файлів на локальному диску (якщо ці ключі були змінені). Цей параметр використовується тільки разом з параметром /n
/n	Заборона оновлення ключів. Цей параметр слугує для пошуку всіх зашифрованих файлів на локальних дисках. Цей параметр використовується тільки разом з параметром /u
шлях	Вказує шаблон, файл або папку
/r: і'мя_файла	Створення нового сертифіката агента відновлення і закритого ключа з наступним їх записом у файлі з іменем, вказаним в параметрі ім'я_файла (без розширення). Якщо використовується цей параметр, усі інші параметри команди «cipher» не враховуються

Наприклад, щоб визначити, зашифрована якась папка чи ні, необхідно використовувати команду:

cipher шлях\ім'я_папки

Команда cipher без параметрів виводить статус (зашифрований чи ні) для всіх об'єктів поточної папки.

Для шифрування файлу необхідно використовувати команду:

cipher /e /a шлях\ім'я_файла

Для розшифрування файла відповідно використовується команда:

cipher /d /a шлях\ім'я_файла

Допустиме шифрування / розшифрування групи файлів за шаблоном:
cipher /e /a d:\work*.doc

Пара «відкритий і закритий ключ для шифрування FEK» створюється для користувача автоматично під час першого шифрування файла з використанням EFS.

Якщо деякий користувач або група користувачів зашифрували файл з використанням EFS, то його вміст доступно тільки їм. Це призводить до ризиків втрати доступу до даних в зашифрованих файлах у разі втрати пароля даними користувачами (працівник забув пароль, звільнився тощо). Для запобігання подібних проблем адміністратор може визначити деякі облікові записи в якості агентів відновлення.

Агенти відновлення (Recovery Agents) визначаються в політиці безпеки **Encrypted Data Recovery Agents** (Агенти відновлення зашифрованих даних) на локальному комп'ютері або в домені. Ця політика доступна через оснащення **Групова політика (gpedit.msc)** розділ **«Параметри безпеки» >> »Політика відкритого ключа» >> »Файлова система EFS».** Пункт меню **«Дія» >> »Додати агент відновлення даних»** відкриває «майстер додавання нового агента».

Додаючи агентів відновлення, можна вказати, які криптографічні пари (позначені їх сертифікатами) можуть використовувати ці агенти для відновлення зашифрованих даних (рис. 8.15). Сертифікати для агентів відновлення створюються командою «cipher» з ключем /r (див. табл. 8.4). Для користувача, який буде агентом відновлення, необхідно імпортувати закритий ключ агента відновлення із сертифіката, створеного командою «cipher». Це можна зробити у вікні «майстра імпорту сертифікатів», що автоматично завантажується під час подвійного клацання по файлу ***.pfx.**

EFS створює **DRF (Data Recovery Field)** – елементи ключів для кожного агента відновлення, використовуючи провайдер криптографічних сервісів, зареєстрований для EFS-відновлення. DRF додається в зашифрований файл і може бути використаний як альтернативний засіб вилучення FEK для розшифрування вмісту файла.

Windows зберігає закриті ключі в підкаталозі **Application Data\ Microsoft\Crypto\RSA** каталогу профілю користувача. Для захисту закритих ключів Windows шифрує всі файли в папці RSA на основі симетричного ключа, що генерується у випадковий спосіб; такий ключ називається **майстер-ключем користувача.** Майстер-ключ має довжину в 64 байти і створюється стійким генератором випадкових чисел. Майстер-ключ також зберігається у профілі користувача в каталозі **Application Data\Microsoft\Protect** і шифрується за алгоритмом 3DES з допомогою ключа, який частково заснований на паролі користувача. Коли користувач змінює свій пароль, майстер-ключі автоматично розшифровуються, а потім заново зашифровуються з урахуванням нового пароля.

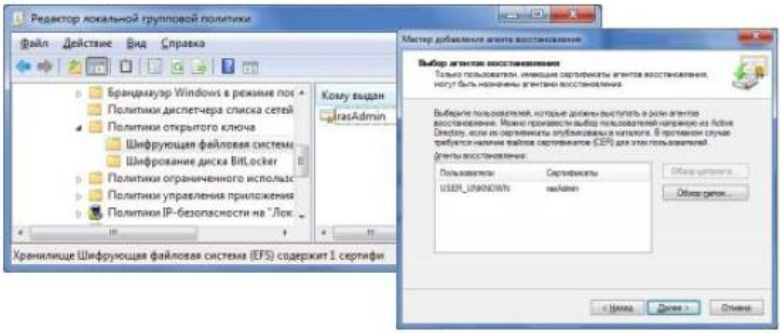


Рис. 8.15. Додавання нового агента відновлення EFS

Для розшифрування FEK EFS використовує функції **Microsoft CryptoAPl (CAPI).** CryptoAPI складається з DLL провайдерів криптографічних сервісів (cryptographic service providers, CSP), які забезпечують програмам доступ до різних криптографічних сервісів (шифрування, розшифрування і хешування). EFS спирається на алгоритми шифрування RSA, що надаються провайдером **Microsoft Enhanced Cryptographic Provider.**

Шифрування та розшифрування файлів можна здійснювати програмно, використовуючи API-функції **EncryptFile** і **DecryptFile.**

Порядок виконання лабораторної роботи

1. Увімкнути ПК.
2. Ознайомитися з короткими теоретичними відомостями.
3. Виконати завдання:

1. Під час виконання лабораторної роботи на комп'ютерах у навчальній лабораторії використовуйте раніше створену віртуальну машину **Test PC1 Win** або створіть та запустіть нову віртуальну машину. Увійдіть у систему під обліковим записом адміністратора. Усі дії в наступних підпунктах виконуйте в системі, що працює на віртуальній машині.

2. Створіть обліковий запис нового користувача, наприклад, testUser розділ «Керування комп'ютером» (compmgmt.msc). Під час створення нового облікового запису забороніть користувачеві зміну пароля і зніміть обмеження на термін дії його пароля. Створіть нову групу, наприклад, «testGroup», і включіть у неї нового користувача, натисніть «Видалити користувача з усіх інших груп». Створіть на диску C: папку «forTesting». Створіть або скопіюйте в папку кілька текстових файлів (*.txt).

3. З допомогою команди «runas» запустити сеанс командного рядка (cmd.exe) від імені новоствореного користувача. Командою whoami подивіться SID користувача і всіх його груп, а також поточні привілеї користувача.

Рядок запуску і результат роботи цієї і ВСІХ наступних консольних команд записати у файл протоколу лабораторної роботи або відобразити у вигляді скріншотів.

4. Переконайтеся у відповідності імені користувача та отриманого SID у реєстрі Windows (використовуйте ключ реєстру HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\ProfileList).

5. Командою whoami визначте перелік поточних привілеїв користувача testUser. У сеансі командного рядка користувача спробуйте змінити системний час командою «time». Щоб надати користувачу такий привілей, запустіть оснащення «Локальні параметри безпеки» (secpol.msc). Додайте користувача в список параметрів політики «Зміна системного часу» розділу «Локальні політики >> Призначення прав користувача». Після цього перезапустіть ваш сеанс командного рядка від імені користувача, переконайтеся, що у списку привілеїв додався SeSystemtimePriviege. Спробуйте змінити системний час командою «time».

6. Переконайтеся, що привілей «Завершення роботи системи» (SeShutdown-Privilege) наданий користувачу testUser. Після цього спробуйте завершити роботу системи з сеансу командного рядка користувача командою shutdown –s. Додайте йому привілей «Примусове віддалене завершення» (SeRemoteShutdownPrivilege). Спробуйте завершити роботу консольною командою ще раз (скасувати команду завершення до її безпосереднього виконання можна командою shutdown –a).

7. Ознайомтеся з довідкою по консольній команді «icacls». Використовуючи цю команду, перегляньте дозволи на папку c:\forTesting. Поясніть всі позначення в описах прав користувачів і груп у видачі команди.

8. Дозвольте користувачу testUser запис у папку forTesting, але забороніть запис для групи testGroup. Спробуйте записати файли чи папки в forTesting від імені користувача testUser. Поясніть результат. Подивіться надані дозволи користувача testUser до папки forTesting у вікні властивостей папки.

9. Використовуючи стандартне вікно властивостей папки, виберіть для користувача testUser такі права доступу до папки, щоб він міг записувати інформацію в папку forTesting, але не міг переглядати її вміст. Переконайтеся, що папка forTesting є тепер для користувача testUser «сліпою», запустивши, наприклад, від його імені файловий менеджер і спробувавши записати файли в папку, переглянути її вміст, видалити файл з папки.

10. Для вкладеної папки forTesting\Docs зніміть наслідування ACL і дозвольте користувачеві перегляд, читання і запис у папку. Перевірте, що для користувача папка forTesting\Docs перестала бути «сліпою» (наприклад, зробіть її поточною в сеансі роботи файлового менеджера від імені користувача і створіть у ній новий файл).

11. Зніміть заборону на читання папки forTesting для користувача testUser. Використовуючи команду «icacls», забороніть цьому користувачу доступ до файлів із розширенням txt у папці forTesting. Переконайтеся в недоступності файлів для користувача.

12. Командою «icacls» забороніть користувачу всі права на доступ до папки forTesting і дозвольте повний доступ до вкладеної папки forTesting\Docs. Переконайтеся, що папка forTesting\Docs є доступною для користувача. Поясніть результат.

13. Від імені користувача testUser зашифруйте який-небудь файл на диску. Переконайтеся, що після цього був створений сертифікат користувача, запустивши налаштування certmgr.msc від імені користувача (розділ Особисті). Перегляньте основні параметри сертифіката відкритого ключа користувача testUser (термін дії, використовувані алгоритми).

14. Створіть у папці forTesting нову папку Encrypt. У папці Encrypt створіть або скопіюйте в неї текстові файли. Зашифруйте папку Encrypt і весь її вміст із меню властивості папки від імені адміністратора. Спробуйте переглянути або скопіювати який-небудь файл цієї папки від імені користувача testUser. Поясніть результат.

15. Скопіюйте зашифрований файл у незашифровану папку (наприклад, forTesting). Переконайтеся, що він залишився зашифрованим. Додайте користувача testUser у список користувачів, які мають доступ до файла у вікні властивостей шифрування файла. Повторіть спробу отримати доступ до файлу від імені користувача testUser.

16. Оформити звіт згідно з вимогами.

17. Відповісти на контрольні питання та підготуватися до опитування.

Зміст звіту

1. Титульний лист.

2. Постановка завдання.

3. Протокол виконання лабораторної роботи, що містить результати роботи всіх консольних команд або відповідні скріншоти з поясненням отриманих результатів.

4. Висновки та відповіді на контрольні питання.

Контрольні питання

1. До якого класу безпеки нажежить ОС Windows за різними критеріями оцінки?
2. У який спосіб користувачі ідентифікуються в ОС Windows?
3. Що таке списки DACL і SACL?
4. Як відбувається перевірка прав доступу користувача до ресурсів ОС Windows?
5. Що таке маркер безпеки і яка його роль у моделі безпеки Windows?
6. Що таке рівень цілісності? Як він впливає на права доступу суб'єктів до об'єктів ОЗ? Як можна дізнатися і задати рівень цілісності для об'єктів і суб'єктів?
7. Які події підлягають аудиту в ОС Windows?
8. У який спосіб зашифровуються файли у файловій системі EFS? Що таке FEK? DDF?
9. Які алгоритми шифрування використовуються в EFS?

Література

1. Інформаційна безпека: навч. пос. / за заг. ред. д-ра техн. наук, проф. Ю. Я. Бобала та д-ра техн. наук, доц. І. В. Горбатого; Ю. Я. Бобало, І. В. Горбатий, М. Д. Кіселичник, А. П. Бондарєв, та інші Львів: Видавництво Львівської політехніки, 2019. 580 с.
2. Крижановський В. Г., Сергієнко С. П. Апаратно-програмні засоби захисту інформації у корпораціях: навчально-методичний посібник. Вінниця: ДонНУ ім. Василя Стуса, 2019. 36 с.
3. Методичні рекомендації до лабораторних робіт з дисципліни «Безпека програмного забезпечення комп'ютерних систем» для здобувачів освітнього ступеня бакалавра спеціальності 123 »Комп'ютерна інженерія» (напрям підготовки 6.050102 «Комп'ютерна інженерія») усіх форм навчання / упоряд.: В. Г. Бабенко, Т. В. Миронюк, Т. А. Стабецька; М-во освіти і науки України, Черкас. держ. технол. ун-т. Черкаси: ЧДТУ, 2018. 42 с.
4. Пономаренко М. М. Захист інформації у телекомунікаційних системах: навч. посіб. / М. М. Пономаренко. Харків: Нац. аерокосм. ун-т ім. М. Є. Жуковського «Харк. авіац. ін-т», 2015. 40 с.
5. Телекомунікаційні та інформаційні мережі: підручник [для ВНЗ] / П. П. Воробієнко, Л. А. Нікітюк, П. І. Резніченко Київ: Самміт-Книга, 2010. 708 с.
6. Тарнавський Ю. А. Технології захисту інформації: підручник для студ. спеціальності 122 «Комп'ютерні науки», спеціалізацій «Інформаційні технології моніторингу довкілля», «Геометричне моделювання в інформаційних системах» / КПІ ім. Ігоря Сікорського. Київ: КПІ ім. Ігоря Сікорського, 2018. 162 с.
7. Критерії оцінки безпеки комп'ютерних систем МО США («Помаранчева книга») / TCSTC (Department of Defense Trusted Computer System Evaluation Criteria, DoD 5200.28-STD, 1983.
8. Глоба Л. С. Розробка інформаційних ресурсів та систем Т. 2. Київ. 2013.
9. Бурячок В. Л. Основи інформаційної та кібернетичної безпеки: навч. пос. / В. Л. Бурячок, Р. В. Киричок, П. М. Складанний. Київ, 2018. 320 с.

Додаток А
Витяг з програми ЄДКІ зі спеціальності 125 «Кібербезпека»
за темою «Захист інформації в ІКС»

Когнітивні рівні, необхідні для відповіді на запитання за темою:
Рівень А. Знання.
Рівень В. Знання, розуміння.
Рівень С. Знання, розуміння, застосування.
Рівень D. Знання, розуміння, застосування та аналіз / синтез / оцінка.

Код	Найменування розділу / підрозділу / теми	Питома вага, %	Когнітивний рівень
3.	**БЕЗПЕКА ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ СИСТЕМ**	**15–25**	
3.1.	Захист інформації, що обробляється та зберігається в ІКС	**1,5–2,5**	
3.1.1.	Процедури ідентифікації, автентифікації, авторизації користувачів		B
3.1.2.	Резервування інформації та компонентів ІКС		B
3.2.	**Програмні та програмно-апаратні комплекси ЗЗІ**	**5–7**	
3.2.1.	Антивіруси, міжмережеві екрани (призначення, архітектура, функції)		B
3.2.2.	IPS, IDS (призначення, архітектура, функції)		B
3.2.3.	Системи контролю та управління доступом в ІКС (Active Directory, ACL)		B
3.3.	**Відновлення функціонування ІКС після реалізації загроз, здійснення кібератак, збоїв та відмов різних класів і походження**	**2,5–3,5**	
3.3.1.	Організаційно-технічні заходи відновлення функціонування ІКС		B
3.3.2.	Журнал аудиту подій		B
3.3.3.	Політики резервного копіювання даних		B
3.4.	**Моніторинг процесів функціонування ІКС**	**2,5–3,5**	
3.4.1.	Джерела інформації про події та типи подій, що аналізуються в системах моніторингу		B
3.4.2.	Система візуалізації та управління подіями (SIEM)		B
3.4.3.	Аналіз подій		B
3.5.	**Механізми безпеки комп'ютерних мереж**	**5–7**	
3.5.1.	Протоколи безпеки на канальному рівні		B
3.5.2.	Протоколи безпеки на мережному рівні (IPSec)		B
3.5.3.	Протоколи безпеки на транспортному / сеансовому рівні (SSL / TLS)		B
3.5.4.	Протоколи безпеки прикладного рівня (HTTPS)		B
3.5.5.	Протоколи автентифікації прикладного рівня (RADIUS)		B
3.5.6.	Віртуальні приватні мережі (VPN)		B

Навчальне видання

Крижановський Володимир Григорович
Чернов Дмитро Вікторович

ЛАБОРАТОРНИЙ ПРАКТИКУМ З КУРСУ «ЗАХИСТ ІНФОРМАЦІЇ В ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ СИСТЕМАХ»

Методичні вказівки

Редактор О. А. Солдатова
Технічний редактор Т. О. Важеніна-Гопрак

Підписано до друку 16.05.2023
Формат 60 х 84/16. Папір офсетний.
Друк – цифровий. Умовн. друк. арк. 4,18
Тираж 10 прим. Зам. 16.

Донецький національний університет імені Василя Стуса
21021, м. Вінниця, 600-річчя, 21
Свідоцтво про внесення суб'єкта видавничої справи
до Державного реєстру серія ДК № 5945 від 15.01.2018