**МІНІСТЕРСТВО ОСВІТИ І НАУКИ,
МОЛОДІ ТА СПОРТУ УКРАЇНИ
МІЖНАРОДНИЙ ЕКОНОМІКО-ГУМАНІТАРНИЙ
УНІВЕРСИТЕТ ІМЕНІ АКАДЕМІКА
СТЕПАНА ДЕМ'ЯНЧУКА**

**Р.М.ЛІТНАРОВИЧ**

# СУЧАСНІ ТЕХНОЛОГІЇ ІНФОРМАЦІЙНОЇ БЕЗПЕКИ

**Частина 1**

**НАВЧАЛЬНИЙ ПОСІБНИК**


**Рівне,2011**

**УДК 614.2.** Літнарович Р.М. Сучасні технології інформаційної безпеки. Частина 1. Навчальний посібник, МЕГУ, Рівне, 2011.-97 с.Litnarovich R. M. Modern technologies of informative safety. Part 1. Train aid. IEGU,Rivne, 2011.-97 p

Рецензенти: В.Г.Бурачек, доктор технічних наук, професор
Є.С. Парняков, доктор технічних наук, професор
В.О.Боровий , доктор технічних наук, професор
Відповідальний за випуск: Й.В. Джунь, доктор фізико-математичних наук, професор

Послідовно розглядаються основні поняття побудови сучасних технологій інформаційної безпеки. Навчально-методичний комплекс містить актуальний матеріал довідково-аналітичного характеру по наступних темах: поняття інформаційної безпеки; основні класифікації погроз інформаційній безпеці; програми з потенційно небезпечними наслідками; елементарні методи цифрового шифрування; симетричні системи захисту інформації; криптографічні системи з відкритим ключем; аутентифікація; методи криптоаналізу класичних шифрів.Для магістрантів-інформатиків.
**Ключові слова**: комп'ютерна безпека, інформаційна безпека, захист, інформація.

Последовательно излагаются основные понятия построения современных технологий информационной безопасности. Учебно-методический комплекс содержит актуальный материал справочно-аналитического характера по следующим темам: понятие информационной безопасности; основные классификации угроз информационной безопасности; программы с потенциально опасными последствиями; элементарные методы цифрового шифрования; симметричные системы защиты информации; криптографические системы с открытым ключом; аутентификация; методы криптоанализа классических шифров.Для магистрантов-информатиков.
**Ключевые слова**: компьютерная безопасность, информационная безопасность, защита, информация.
The basic concepts of construction of modern technologies of informative safety are consistently expounded. Train aid complex is contained by aktual material of certificate-analytical character on the followings themes: concept of informative safety; basic classifications of threats informative safety; programs with potentially hazard effects; elementary methods of digital encipherement; symmetric systems of priv; cryptographic systems with the opened key; authentification; methods of kriptoanaliz of classic codes.For magistrant-informatik.
**Keywords**: computer safety, informative safety, defence, information

**ЗМІСТ ……...Стор.**

## ВСТУП

В останнє сторіччя з'явилося багато галузей виробництва, які майже на 100% складаються з однієї інформації, наприклад дизайн, створення програмного забезпечення, реклама та ін. Яскраво демонструє підвищення ролі інформації у виробничих процесах поява такого заняття, як промислове шпигунство. Не матеріальні цінності, а чиста інформація стає об'єктом викрадення.

З появою і поширенням комп'ютерів і засобів автоматизованої обробки інформації виникла потреба в автоматизованих засобах захисту файлів та іншої збереженої комп'ютерами інформації. Особливо гостро стоїть потреба в засобах захисту і відчувається в багатокористувацьких системах, таких як системи з розділенням часу, а також в системах, до яких можна отримати доступ по простим телефонним лініям зв'язку або відкритим комп'ютерним мережам. При цьому для опису сукупності методів і засобів, призначених для захисту даних і протидії хакерам, став застосовуватися термін комп'ютерна безпека.

В результаті появи розподілених систем обробки даних і використання мереж і комунікаційного обладнання для обміну даними між користувачами терміналів і центральними комп'ютерами виникла потреба в забезпеченні захисту мережі, по якій передаються дані. При цьому слід зазначити, що термін "мережна безпека" має на увазі не одну, окремо взяту локальну мережу, а деяку сукупність мереж підприємств, урядових установ і навчальних закладів, пов'язаних між собою в об'єднану мережу обробки даних (Internet).

Між цими двома формами безпеки - комп'ютерної та мережевої - навряд чи можна провести чітку межу. Наприклад, одним із самих широко відомих типів втручання в роботу інформаційної системи є комп'ютерний вірус. Вірус може вноситися в систему як фізично, наприклад з дискети, з якої він зчитується і потрапляє в комп'ютер, так і через об'єднану мережу. Але як би там не було, після зараження комп'ютерної системи вірусом для його ідентифікації та видалення доводиться використовувати локальні засоби комп'ютерної безпеки.

Навчальний посібник, що представляє собою структуровану підбірку матеріалів, присвячених розгляду сучасних технологій інформаційної безпеки і методів захисту інформації,ставить за мету запропонувати систематично огляд теоретичних основ криптографії та її практичних застосувань в області захисту інформації. У даному посібнику висвітлюються актуальні питання захисту інформації при створенні і використанні розподілених корпоративних інформаційних систем. Особливу увагу приділено проблемам забезпечення інформаційної безпеки, елементарним методам цифрового шифрування, аутентифікації і методам криптоаналізу класичних шифрів.

Навчально-методичний комплекс з дисципліни "Інформаційна безпека та захист інформації" включає в себе навчальний посібник, робочу навчальну програму з дисципліни, методичні рекомендації до виконання лабораторних курсових робіт, а також тести. Запитання з варіантами відповідей для самоконтролю допоможуть краще розібратися в досліджуваному матеріалі і глибше зрозуміти його зміст, також ці питання можуть бути використані викладачами з метою контролю засвоєння матеріалу студентами.

Цикл, що складається з дев’яти лабораторних робіт, проводиться в рамках дисципліни "Технології комп’ютерної безпеки".

Серед усього спектру методів захисту даних від несанкціонованого доступу особливе місце займають

криптографічні алгоритми. На відміну від інших методів, вони спираються лише на властивості самої інформації і не використовують властивості її матеріальних носіїв, особливості вузлів її обробки, передачі та зберігання. Образно кажучи, криптографічні методи будують бар'єр між інформацією що захищається і реальним або потенційним зловмисником із самої інформації. Криптографічні методи захисту інформації в автоматизованих системах можуть застосовуватися як для захисту інформації, що обробляється в ЕОМ або що зберігається в різного типу вузлах, так і для закриття інформації, переданої між різними елементами системи по лініях зв'язку.

Основні напрямки використання криптографічних алгоритмів - передача конфіденційної інформації з каналів зв'язку (наприклад, електронна пошта), встановлення автентичності переданих повідомлень, зберігання інформації (документів, баз даних) на носіях у зашифрованому вигляді.

Виконання лабораторної роботи № 1 "Алгоритми обчислення найбільшого спільного дільника" дає студентам практичні навички реалізації алгоритмів Евкліда (класичного, бінарного, розширеного).Програми, що реалізують зазначені алгоритми, використовуються у двох наступних лабораторних роботах і, крім того,застосовуються в подальших роботах з дисципліни "Криптографічні методи та засоби забезпечення інформаційної безпеки" для обчислення найбільшого спільного дільника і звернення чисел і многочленів (наприклад,при виробленні параметрів криптосистеми табл).

При виконанні роботи № 2 "Імовірнісні алгоритми перевірки чисел на простоту" студенти повинні написати програми, що реалізують тести Ферма, Соловея-Штрассена





і Міллера-Рабіна, переконатися в працездатності алгоритмів (враховуючи імовірнісний характер тестування), дослідити алгоритми на кількість помилок, що допускаються ними в залежності від виду тестованого числа (розпізнавання чисел Кармайкла) і від кількості прогонів тесту, обробити отримані статистичні дані, а також виміряти і порівняти час роботи програм. Отримані навички використовуються в курсових роботах з дисципліни "Криптографічні методи та засоби забезпечення інформаційної безпеки" при виробленні параметрів криптосистем з відкритим ключом (для шифрування, цифрового підпису, алгоритмів на еліптичних кривих).

При виконанні лабораторних робіт № 3, № 4 та № 5 "Елементарні методи цифрового шифрування", "Симетричні системи захисту інформації" і "Системи захисту інформації з відкритим ключом. Асиметричні системи "студенти повинні написати програми, що реалізують ці алгоритми, протестувати їх і реалізувати програми, які допомагають розкрити один з алгоритмів шляхом криптографічного аналізу, досліджувати алгоритми на кількість помилок, обробити отримані статистичні дані,а також виміряти і порівняти час роботи програм.

Навчальний посібник, призначений як для академічної, так і професійної аудиторії, може виступати в якості основи курсу з інформаційної безпеки та захисту інформації для студентів, а також може використовуватися як довідкова література.

## 1. ПОНЯТТЯ ІНФОРМАЦІЙНОЇ БЕЗПЕКИ

Давно відомо, що інформація може бути справжнім скарбом. Саме тому часто багато зусиль витрачається як на охорону інформації, так і на добування її. Інформацію

потрібно захищати в тих випадках, коли є побоювання, що вона стане доступною стороннім, які можуть звернути її на шкоду законному користувачеві.
[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20].

Інформація - основне поняття наукових напрямів, що вивчають процеси передачі, переробки та зберігання різних даних. Суть поняття інформації зазвичай пояснюється на прикладах. Формальне визначення не дається, оскільки поняття інформації відноситься до таких фундаментальних понять, як матерія.

Інформація, яка потребує захисту, виникає в самих різних життєвих ситуаціях. У таких випадках кажуть, що інформація містить таємницю і є захищаємою, приватною, конфіденційною, таємною. Для найбільш типових ситуацій введені спеціальні поняття: державна таємниця, військова таємниця, комерційна таємниця, юридична таємниця, лікарська таємниця.

Інформація, що захищається має наступні ознаки:

- *Є певние коло законних користувачів, які мають право володіти цією інформацією;*
- *Є незаконні користувачі, які прагнуть оволодіти цією інформацією з тим, щоб звернути її собі на благо, а законним користувачам на шкоду.*

З боку незаконних користувачів існують різні види загроз для інформації, що захищається: загроза розголошення інформації, підміна інформації, імітація інформації та ін

Між людьми відбувається інтенсивний обмін інформацією, причому часто на великі відстані. Для забезпечення такого обміну інформацією існують різні види технічних засобів зв'язку: телеграф, телефон, радіо, телебачення. Нерідко виникає необхідність в обміні між віддаленими користувачами не просто інформацією, а захищеною інформацією. У цьому випадку незаконний користувач може спробувати перехопити інформацію з





загальнодоступного технічного каналу зв'язку.
Побоюючись цього, законні користувачі мають вжити додаткових заходів для захисту своєї інформації.
З підвищенням значущості і цінності інформації відповідно зростає і важливість її захисту.

З одного боку, інформація коштує грошей. Значить, витік або втрата інформації спричинить матеріальний збиток. З іншого боку, інформація - це управління. Несанкціоноване втручання в управління може привести до катастрофічних наслідків в об'єкті управління - виробництві, транспорті, військовій справі. Наприклад, сучасна військова наука стверджує, що повне позбавлення засобів зв'язку зводить боєздатність армії до нуля.

Захист інформації (ЗІ) в рамках цього курсу визначимо так: заходи для обмеження доступу до інформації для будь-яких осіб (категорій осіб), а також для посвідчення автентичності і незмінності інформації.

**Аспекти захисту інформації**

Крім усього вищесказаного, є ще одна важлива проблема: проблема співвідношення ціни інформації, витрат на її захист і витрат на її добування. При сучасному рівні розвитку техніки самі засоби зв'язку, а також розроблення засобів перехоплення інформації з них і засобів захисту інформації вимагає дуже великих витрат.

По-перше, хороший захист інформації обходиться дорого. Поганий же захист нікому не потрібний, тому що наявність в ній лише однієї "дірки" означає повну марність усього захисту в цілому (принцип суцільного захисту). Тому перш ніж вирішувати питання про захист інформації, слід визначити, чи варта вона того. Чи здатний можливий збиток від розголошення чи втрати інформації перевищити витрати на її захист? З цією ж метою потрібно максимально звузити коло, що захищається, щоб не витрачати зайвих грошей і часу.

По-друге, перш ніж захищати інформацію, не зайве визначити перечень ймовірних загроз, оскільки від всього на світі ви все одно не захиститесь. Можливий варіант, коли вам треба убезпечити дані від несанкціонованого доступу ззовні, наприклад з Інтернету. Можливо, однак, що хакерів ваші дані зовсім не цікавлять, і вам слід захищати інформацію тільки від власних співробітників. Можливо також, що викрадення або розголошення вашої інформації нікому не зашкодить, але от її підміна може завдати вам шкоди. У всіх трьох випадках методи захисту будуть сильно відрізнятися.

По-третє, при плануванні схеми ЗІ велике значення має не тільки її об'єктивна надійність, але і ставлення до захисту інших людей. У деяких випадках достатньо, щоб ви самі були впевнені в достатній надійності захисту. А в інших - це потрібно довести іншим людям (наприклад, замовникам ), які часто не розуміються у відповідних питаннях.

Таким чином, не існує абсолютного захисту. Всякий захист вимірюється часом злому.

Швидко розвиваються комп'ютерні інформаційні технології і вносять помітні зміни в наше життя. Інформація стала товаром, який можна придбати, продати, обміняти. При цьому вартість інформації часто в сотні разів перевершує вартість комп'ютерної системи, в якій вона зберігається.

Від ступеня безпеки інформаційних технологій в даний час залежить благополуччя, а часом і життя багатьох людей. Така плата за ускладнення і повсюдне поширення автоматизованих систем обробки інформації.

Під інформаційною безпекою розуміється захищеність інформаційної системи від випадкового або навмисного втручання, що наносить збиток власникам або користувачам інформації.





На практиці найважливішими є три аспекти інформаційної безпеки:

- *Доступність (можливість за розумний час отримати необхідну інформаційних послугу);*
- *Цілісність (актуальність і несуперечність інформації, її за хищеність від руйнування і несанкціонованої зміни);*
- *Конфіденційність (захист від несанкціонованого прочитання).*

Порушення доступності, цілісності і конфіденційності інформації можуть бути викликані різними небезпечними діями на інформаційні комп'ютерні системи.

Виходячи з усього згаданого в цій главі, можна прийти до висновку, що забезпечення інформаційної безпеки - це дуже важливе, надзвичайно велике і необхідне питання, рішення якого планується з цілком чітко визначених умов, має безліч варіантів,сукупне використання яких дає прийнятний результат.

## 2. ОСНОВНІ КЛАСИФІКАЦІЇ ЗАГРОЗ ІНФОРМАЦІЙНОЇ БЕЗПЕКИ

Сучасна інформаційна система являє собою складну систему, що складається з великої кількості компонентів різного ступеня автономності, які пов'язані між собою і обмінюються даними. Практично кожен компонент може піддатися зовнішньому впливу або вийти з ладу.

Забезпечення збереження конфіденційної інформації необхідно починати з визначення системи загроз, тобто негативних процесів, сприяючих витоку інформації.Саме загрози покликані усувати механізми забезпечення безпеки. Для успішного захисту ЗС необхідно знати весь перелік загроз безпеки. Тип загроз і їх кількість залежать від типу ПС, але якщо розробник має повний список загроз, то він зможе вибрати необхідні і застосувати потрібні для їх усунення засоби захисту. Існує безліч

класифікацій видів загроз за принципами і характером їх дії на систему, по використовуваних засобах , по цілях атаки і т.і. У даній роботі наводяться кілька загальних класифікацій загроз безпеки.

### 2.1. Класифікація загроз, запропонована Стівом Кентом

Для класифікації загроз був прийнятий підхід, запропонований Стівом Кентом. Ця класифікація не втратила актуальності і становить основу більшості описів загроз захисту (рис. 2.1).

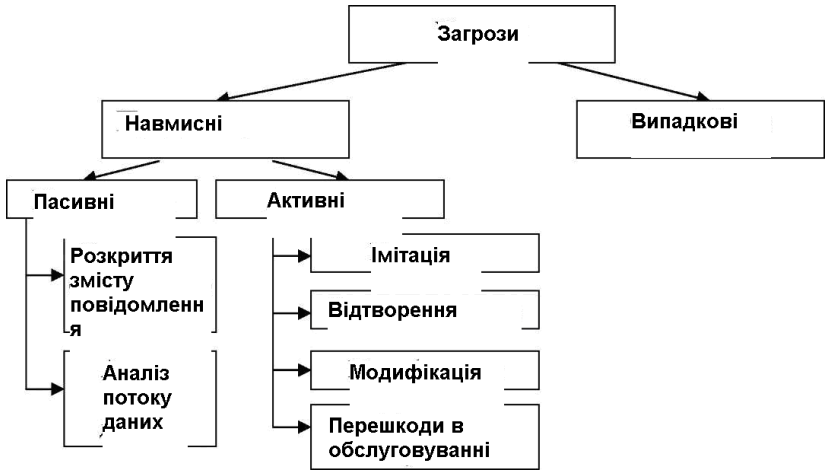



Рис.2.1. Загрози безпеки

Загрози підрозділяються на навмисні (пов'язані з цілеспрямованими діями порушника; в якості порушника можуть виступати службовці, відвідувачі, конкуренти, найманці і т.і.) і випадкові. Причинами випадокових впливів можуть бути аварійні ситуації через стихійні лиха і відключення електроживлення, відмови і збої апаратури, помилки в програмному забезпеченні, помилки в роботі





обслуговуючого персоналу і користувачів, перешкоди в лініях зв'язку через впливи зовнішнього середовища. Навмисні в свою чергу поділяються на пасивні і актив ні. Пасивні атаки носять характер перехоплення або моніторингу переданої інформації і не пов'язані з якою-небудь зміною інформації, їх можна умовно розділити на дві групи: розкриття вмісту повідомлення і аналіз потоку даних. Що таке розкриття повідомлення пояснювати не потрібно (телефонна розмова, електронна пошта).

Другий тип пасивних порушень - аналіз потоку даних - більш складний для розуміння. Припустимо, що ми використовуємо такий спосіб маскування вмісту повідомлення, що порушник, отримавши повідомлення, не може витягти з нього інформацію. Але навіть якщо шифрування надійно приховує зміст, у порушника залишається можливість спостерігати характерні знаки що передаються. Наприклад, можна обчислити вузли відправлення, відстежити частоту обміну повідомленнями та його довжину.

Пасивні порушення виявити дуже важко, але їх цілком реально попередити.

Активні загрози пов'язані зі зміною потоку даних або зі створенням фальшивих потоків і можуть бути поділені на чотири групи:

1. Імітація означає спробу одного об'єкту видати себе за інший. Зазвичай імітація виконується разом зі спробою активного порушення якого-небудь іншого типу. Наприклад, перехопивши потік даних аутентифікації, якими обмінюються системи, порушник може потім відтворити реальну послідовність аутентифікації, що дозволяє об'єкту з обмеженими повноваженнями розширити свої повноваження, імітувавши об'єкт, що має більш широкі повноваження.

2.Відтворення представляє собою пасивне перехоплення блоку даних і подальшу повторну передачу перехоплених даних з метою одержання несанкціонованого ефекту.

3.Модифікація означає або зміну частини легітимного повідомлення, або його затримку, або зміни порядку надходження повідомлення з метою отримання несанкціонованого ефекту. Наприклад, повідомлення "Дозволити доступ до секретного файлу Бюджет Івану Іванову" можна перетворити до вигляду "Дозволити доступ до секретного файлу Бюджет Петру Петрову".

4. Перешкоди в обслуговуванні створюють перешкоди в нормальному функціонуванні засобів зв'язку або управлінні ними. Наприклад, об'єкт може задержувати всі повідомлення, спрямовані певному адресату. Іншим прикладом перешкод в обслуговуванні є блокування роботи всієї мережі шляхом виводу мережі з ладу або шляхом навмисної її перевантаженні інтенсивним потоком повідомлень.

Активні загрози захисту мають характеристики, протилежні характеристикам пасивних порушень. Якщо пасивні порушення важко виявити, але існують методи, що дозволяють їх запобігти, то активні загрози повністю запобігти дуже непросто, оскільки це можна здійснити тільки в безперервному часі фізичним захистом всіх засобів зв’язку. Тому в разі активних порушень основною метою має бути виявивлення таких і швидке відновлення нормальної працездатності системи, яка може працювати повільніше або не працювати взагалі.

**2.2. Класифікація загроз безпеки за засобами впливу на систему**

По засобах впливу розрізняють три основні класи загроз:

*1. Втручання людини в роботу обчислювальної системи.* До цього класу належать організаційні засоби порушення безпеки (крадіжка носіїв інформації, несанкціонований





доступ до пристроїв зберігання і обробки інформації, псування устаткування і т.і.) І здійснення порушником несанкціонованого доступу до програмних компонентів системи (всі способи несанкціонованого проникнення в систему, а також способи отримання користувачем-порушником незаконних прав доступу). Заходи, що протистоять таким загрозам, носять організаційний характер, а також включають в себе вдосконалення систем розмежування доступу і системи виявлення спроб атак (наприклад, спроб підбору паролів).

*2. Апаратурно-технічне втручання в роботу обчислювальної системи*. Мається на увазі порушення безпеки та цілісності інформації за допомогою технічних засобів, наприклад отримання інформації по електромагнітному випромінюванні пристроїв, електромагнітні впливи на канали передачі інформації та інші методи. Захист від таких загроз, крім організаційних заходів, передбачає відповідні апаратні і програмні заходи.

*3. Руйнівний вплив на програмні компоненти системи за допомогою програмних засобів*. Такі засоби називаються руйнівними програмним засобами (РПС). До них відносяться комп'ютерні віруси, троянські коні, засоби проникнення у віддалені системи з використанням локальних і глобальних мереж. Засоби боротьби з подібними атаками складаються з програмно-і (рідше) апаратно-реалізованих систем захисту.

Розглянемо докладніше кожен із класів.

*Втручання людини*

• Диверсія (sabotage) - це фізичне або логічне пошкодження, свідомо наноситься зловмисником обладнанню або інформації. При фізичному пошкодженні зловмисник може, наприклад, викликати пошкодження жорсткого диску або вимкнути живлення при виконанні критичної дискової операції. Як приклади логічного

ушкодження можна навести зміну внутрішніх або зовнішніх міток і використання програмного забезпечення, яке змінює вміст файлу. Диверсії в основному здійснюють скривджені співробітники перед звільненням. Помста фірмі або конкретній особі - основний мотив диверсії.

• *Спотворення (misrepresentation)* - це використання комп'ютера для введення в оману або залякування з певною метою. При даній загрозі комп'ютер розглядається як об'єкт, предмет або інструмент правопорушення. Обман, навіть якщо з його допомогою витягується прибуток, не обов'язко є злочином, тому спотворення саме по собі згідно законодавства ще не комп'ютерний злочин. Головна небезпека спотворення полягає у введенні в оману осіб, відповідальних за безпеку, при приховуванні порушень.

• *Крадіжка (theft)* може бути трьох типів: крадіжка обладнання, крадіжка інформації і крадіжка послуг.

Крадіжка обладнання - найчастіше викликає найбільші побоювання. Адже втрата комп'ютерів означає втрату продуктивності фірми, втрату всієї зберігаємої в викрадених комп'ютерах інформації, яка після цього втратить свою таємність, доступність і може бути цілісність.

Крадіжка інформації - для того щоб викрасти інформацію, не потрібно виносити з будівлі комп'ютер. Термін "витік даних" (data leakage) застосовується для опису дій, які полягають в таємному копіюванні інформації і винесення її за межі організації. Цілісність наражається на небезпеку тільки в тому випадку, коли відбувається крадіжка єдиної копії інформації. Крадіжка інформації може залишатися непоміченою.

Крадіжка послуг - може виявлятися в різних формах: від ігор на службовому комп'ютері до виготовлення з його допомогою книг. Використання продукції фірми та її





ресурсів в особистих цілях є крадіжка послуг. Цей вид крадіжки представляє небезпеку доступності.

• *Розкрадання (embezzlement)* зазвичай має відношення до внутрішньої роботи, коли крадіжка грошей або ресурсів роботодавця виконується його ж працівником. Будучи одним з найстаріших видів злочинів взагалі, розкрадання є одним з найбільш поширених видів комп'ютерних злочинів зокрема. Найлегшим способом розкрадання даних є їх підміна (data diddling) - процес зміни даних перед введенням або в процесі введення. Розкрадачі можуть скористатися таким багатим набором різних прийомів, що розповісти про всі практично неможливо.

• *Шахрайство (fraud)* - це будь-яке використання інформаційної системи при спробі обману організації або отримання її ресурсів. Способи шахрайства настільки ж різноманітні, як і способи розкрадання.

• *Недбалість (bumbling)* - помилки людини (human errors), випадковості (accidents), помилки (errors of omission), прояви некомпетентності (errors of commission).

• *Неточна або застаріла інформація.*

• *Різні версії*.Програмне забезпечення постійно оновлюється, тому завжди необхідно стежити за тим, які версії програм виконуються і своєчасно їх оновлювати.

• *Піггібекінг (piggybacking)* - це нелегальне проникнення куди-небудь услід за особою, яка має легальний доступ. Електронний піггібекінг - це отримання нелегального доступу після того, як легальний користувач, ввівши пароль і підключившись до системи, некоректно завершив сеанс роботи або завершив сеанс роботи, але не відключився від системи. Фізичний піггібекінг - це безпосереднє проникнення в закриту зону після особи, яка має до неї доступ.

• *Самозванство (impersonation)* - це використання коду доступу іншої людини для проникнення в систему з метою

вивчення даних, використання програм або відведеного йому комп'ютерного часу.

• *Збір сміття (scavenging)* або підглядання (browsing) часто пов'язані з необхідністю покопатися у відходах (dumpster diving), щоб знайти листинги, стрічки, диски, інформацію про кредитні картки, використані копірки та інші відомості. Стосовно до комп'ютерів збір сміття може означати відновлення за допомогою відповідних утиліт файлів, видалених користувачем. Крім того, джерелом витоку інформації може стати принтер, коли через затримки друку користувач створює більше число копій, котрі можуть потрапити в чужі руки.

• *Умисне пошкодження даних або програм.*

*Апаратурно-технічне втручання*

При апаратурному втручанні у функціонування НД можливі два варіанти порушення безпеки: випадкове порушення (апаратні збої, перевантаження) і умисне (випромінювання, перехоплення), при якому апаратура ви ¬ ступає інструментом порушників. Розглянемо всі ці випадки.

• Апаратні збої, які можуть порушити таємність, якщо вони проходять у пристрої керування доступом або якщо відновлення работоспроможності системи вимагає пониження рівня захисту. Апаратний збій може завдати шкоди цілісності. Іноді навіть збій однієї з мікросхем пам'яті може привести до втрати інформації, якщо він відбудеться, наприклад, при копіюванні. Доступність також може постраждати від апаратного збою. Якщо який-небудь пристрій вийде з ладу, то може бути тимчасова або постійна втрата доступу до надаваних їм ресурсів.

• Перевантаження. При великих навантаженнях зламається все що завгодно. Коли система перевантажується, безпека мережі наражається на ризик.





• Випромінювання (emanations) - це випускання електромагнітних сигналів, що є одним зі слабких місць у комп'ютерному захисті. Як кабелі, так і спільні з їх допомогою пристрої випромінюють певні сигнали. Чутлива антена дозволить на відстані прочитати дані навіть при незначному рівені випромінювання. Слабкими місцями в системі безпеки через випромінювання можуть бути: комп'ютери, принтери, модеми, монітори, клавіатура, перехідники, підсилювачі, розподільні коробки, місця з'єднання коаксиального кабелю.

• *Перехоплення (wiretapping*) може виконуватися як із застосуванням елементарних затискачів типу "крокодил", так і шляхом спостереження за випромінюванням або супутниковими передачами за допомогою антен. Дуже часто встановити підслуховуючий пристрій на одному кінці кабелю простіше, ніж підключитися десь посередині.

*Вплив на програмні компоненти обчислювальної системи*

Існує три види використання програмних продуктів для атаки на обчислювальну систему. Або використовуються впроваджувані руйнівні програмні засоби (віруси, хробаки, троянські коні), або використовується установлене програмне

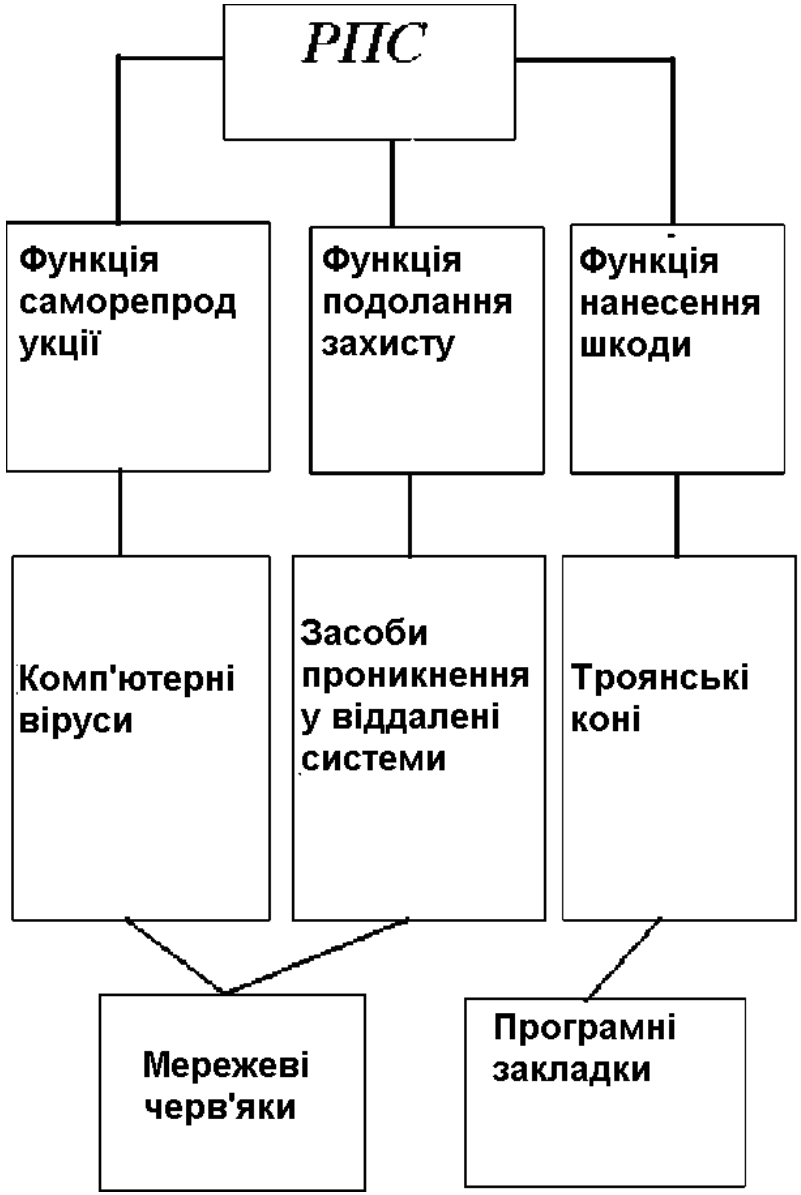



Рис 2.2. Типи РПС





забезпечення (логічні бомби, помилки програмування, неправильна маршрутизація, потаємні ходи та лазівки). Третім видом є використання особливих утиліт і програм (мережеві аналізатори, суперзаппінг).

Руйнівні програмні засоби (РПС) - це клас програм, призначених для атаки на ПС і містять деструктивні функції.РПС бувають несамовідтворюваними (троянські коні) і само відтворюваними (віруси, хробаки). Класифікація РПС представлена на рис 2.2.

Встановлене програмне забезпечення служить джерелом потенційних проломів в безпеці. Потенційні проломи в безпеці (flaw) - це недоліки програмного забезпечення, закладені на етапі його розробки та впровадження. За цей тип загроз несуть відповідальність розробники, які допускають помилки в програмуванні, залишають у готовому продукті засоби налагодження і приховані лазівки. Помилки в ході експлуатації також призводять до потенційних проломів в безпеці, наприклад неправильна маршрутизація.

Особливі утиліти є засобом порушення безпеки в руках зловмисників. Прослуховування мережевого трафіку, здійснення цілого класу віддалених атак можливо лише за допомогою спеціального програмного забезпечення.

Дуже коротко дамо поняття руйнівних програмних засобів, більш детально про них буде викладено далі.

Класифікація РПС представлена на рис. 2.2.

• *Віруси* - це програми, які можуть заражати інші програми, модифікуючи їх допомогою додавання своїх, можливо змінених копій. За статистикою основними способами проникнення вірусів є принесені дому носії та програмне забезпечення, яке розповсюджується по глобальних мережах. Для вірусів, на відміну від інших РПС, необхідний носій, з яким вони проникають в систему і заражають інші файли.

• *Троянський кінь (troyan horse)* - це будь-яка програма, що містить у собі деяку руйнівну функцію, яка активізується при настанні деякої умови спрацьовування. Троян - це вірус, який, маскуючись під корисну програму, антивірус, upgrade до Windows і т.і., заражає ваш ком ¬ п'ютер.

• *Мережевими хробаками* називають віруси, які розповсюджуються із глобальних мереж, вражаючи цілі системи, а не програми. З появою глобальної мережі Internet цей вид порушення безпеки представляє найбіль ¬ шу загрозу. Зазвичай метою злому мереж є придбання нелегальних прав на користування ресурсами системи.

• *Логічна бомба (logic bomb)* - це модифікація комп'ютерної програми, в результаті якої дана програма може виконуватися декількома способами в залежності від певних обставин. При перевірці в звичайних умовах бомба ніяк не виявляється, але при певній події програма працює за алгоритмом, відмінним від заданї. Логічні бомби можуть використовуватись для розкрадань, можуть випадковим чином змінювати чи знищувати дані.

• *Потайні ходи (back door)* - це додатковий спосіб проникнення в систему, часто навмисно створюваний розробником мережі, хоча іноді він може виникнути і випадково. Лазівка (trap door) - різновид потайного ходу. Так зазвичай називають допоміжні засоби, які програмісти використовують при створенні, тестуванні або підтримці комплексних програм. Потаємні ходи зазвичай не документуються.

• *Помилки програмування (bugs)*. Помилки не є результатом злого умислу, але їх наявність в програмному продукті наражає на небезпеку самі різні аспекти функціонування системи.

• При роботі в мережах дуже часто виникає проблема неправильної маршрутизації, тобто інформація передається не тому, кому вона призначена. Це може бути пов'язано зі





збігом номерів вузлів глобальної мережі або з дуже педантичним проходженням інструкції, коли кілька користувачів дають своїм вузлам імена, вказані в інструкції по замовчуванню.

• Мережеві аналізатори - це програми, що дозволяють вести аналіз сітьового трафіку. Використовуючи апаратні і програмні засоби, більшість з них можуть зчитувати будь-які параметри потоку даних, включаючи будь-який незашифрований текст.

• *Суперзаппінг (superzapping)* - це несанкціоноване використання утиліт для модифікації, знищення, копіювання, розкриття, вставки, при псуванні або забороні застосування комп'ютерних даних. Ніякими програмними засобами суперзаппінг зазвичай виявити неможливо. Крім того, навіть використовуючи системні журнали, довести суперзаппінг дуже складно, так як порушник може відредагувати ці журнали.

**2.3. Загрози безпеки в розподільчих системах**

Розглянемо типові загрози, характерні для сучасних розподільчих систем. Класифікуючи загрози, виділимо фундаментальні, первинні, що ініціюють базові загрози.

*До фундаментальних загроз належать:*

витік інформації - розкриття інформації неавторизованому користувачеві або процесу;

порушення цілісності - компрометація узгодженості (не протиріччя ) даних шляхом цілеспрямованого створення, підміни і руйнування даних;

відмова в послузі - навмисне блокування легального доступу до інформації чи інших ресурсів (наприклад, за допомогою перевантаження сервера потоком запитів);

незаконне використання - використання ресурсів незаконним чином, неавторизованим об'єктом або суб'єктом. Наприклад,використання вилученого комп'ютера з метою "злому" інших комп'ютерів мережі.

Реалізація фундаментальних загроз багато в чому залежить від реалізації первинних погроз. Первинні загрози ініціюють фундаментальні загрози. Первинні загрози поділяються на загрози проникнення і загрози впровадження.

*До загроз проникнення відносяться:*

маскарад. Користувач (або інша сутність - процес, підсистема і т.д.) маскується і намагається видати себе за іншого користувача. Дана загроза, як правило, пов'язана зі спробами проникнення всередину периметра безпечності і часто реалізується хакерами;

обхід захисту - використання слабких місць системи безпеки для обходу захисних механізмів з метою отримання законних прав та привілеїв;

порушення повноважень - використання ресурсів не за призначенням. Дана загроза пов'язана з діями внутрішнього порушника.

*До загроз впровадження відносяться:*

троянські програми - програми, що містять прихований або явний програмний код, при виконанні якого порушується функціонування системи безпеки;

потаємні ходи - деякі додаткові можливості, таємно вбудовані в систему або її компоненти, що порушують функціонування системи безпеки при введенні специфічних даних.

Подібні загрози, як правило, реалізуються за допомогою спеціальних агентів впровадження, активізуються після деякого періоду латентності.

Розглядаючи фундаментальні загрози, слід враховувати також загрози базові. Наприклад, витік інформації пов'язаної з такими базовими погрозами, як підслуховування, аналіз трафіку, персональна необережність, "копання у смітті".





**2.4. Взаємозв'язок різних видів загроз**

Взаємозв'язок різних загроз може бути досить складним (рис. 2.3).Так, маскарад є загрозою, що ініціює фундаментальні загрози, в тому числі витік інформації. Однак маскарад сам по

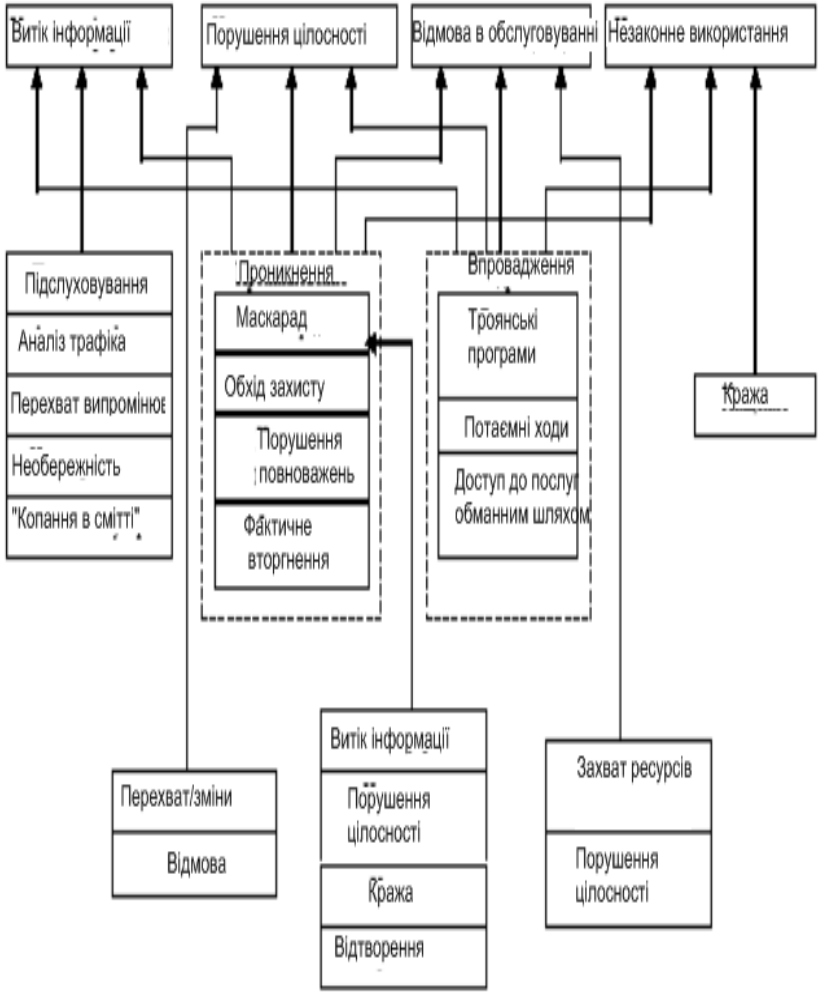



Рис.2.3. Взаємозв’язок загроз

собі також може залежати від витоку інформації. Наприклад, розкриття пароля може ініціювати загрозу маскараду.

Аналіз понад трьох тисяч комп'ютерних злочинів показав, що найчастіше виникають такі загрози (у порядку убування): порушення повноважень, маскарад, обхід захисту, троянські програми або потаємні ходи, "копання у смітті".

Розглянуті в цьому розділі загрози безпеці досить багаточислені і різноманітні. Звичайно, розробнику необхідно знати і орієнтуватися у все зростаючому обсязі загроз, але для обчислювальної системи різного призначення список актуальних загроз різний. Щоб визначити, які загрози для даної системи більш актуальні, а які зовсім не важливі, існують методи аналізу небезпек - це визначення ймовірності загроз і потенціальних втрат, які можуть статися внаслідок вад системи. Основуючись на результатах аналізу, можна вибрати ряд засобів контролю або заходів забезпечення безпеки, що володіють високою економічною ефективністю і забезпечують необхідний рівень захисту.

## 3. ПРОГРАМИ З ПОТЕНЦІЙНО НЕБЕЗПЕЧНИМИ НАСЛІДКАМИ

Програмою з потенційно небезпечними наслідками назвемо програму або частину програми, яка здатна виконати одну з таких дій:

• сховати ознаки своєї присутності в програмному середовищі ПЕОМ;

• самодубліруватися, асоціювати себе з іншими програмами та / або переносити свої фрагменти в будь-якій області оперативної або зовнішньої пам'яті, що не належить програмі;





• змінювати код програм в оперативній або зовнішній пам'яті;

• зберігати фрагменти інформації з оперативної пам'яті в деяких областях зовнішньої пам'яті (локальних або віддалених);

• спотворювати довільним чином, блокувати та / або підміняти виводи в зовнішню пам'ять або канал зв'язку масив інформації, що утворився в результаті роботи прикладних програм, або вже знаходиться у зовнішній пам'яті масива даних.

**Програми з потенційно небезпечними наслідками можна умовно підрозділити:**

• на класичні програми-"віруси";

• програми типу "програмний черв'як" або "троянський кінь" і фрагменти програм типу "логічний люк";

• програми типу "логічна бомба";

• програмні закладки - узагальнений клас програм з потенційно небезпечними наслідками.

Крім того, такі програми можна класифікувати за методом і місцем їх впровадження і застосування (тобто по "способу доставки" в систему):

• закладки, пов'язані з програмно-апаратним середовищем (BIOS);

• закладки, пов'язані з програмами первинного завантаження;

• закладки, пов'язані з драйвером DOS, командним інтерпретатором, мережними драйверами, тобто із завантаженням і роботою операційного середовища;

• закладки, пов'язані з прикладним програмним забезпеченням загального призначення (вбудовані в клавіатурні та екранні драйвери, програми тестування ПЕОМ, утиліти, файлові оболонки);

• виконувані модулі, що містять тільки код закладки (як правило, впроваджувані в пакетні файли типу BAT);

• модулі-імітатори, що збігаються за зовнішнім виглядом з легальними програмами, які вимагають введення конфіденційної інформації;
• закладки, масковані під програмні засоби оптимізаційного призначення (архіватори, прискорювачі і т.і.);
• закладки, масковані під програмні засоби ігрового та розважаю чого призначення (як правило, використовуються для первинного впровадження інших закладок; умовна назву - **"дослідник").**

### 3.1. Комп'ютерні віруси

Перші дослідження саморозмножуваних штучних конструкцій проводилися в середині минулого століття: у роботах фон Неймана, Вінера й інших їм дано визначення і приведений математичний аналіз кінцевих автоматів, у тому числі само відтворюваних.. Термін "комп'ютерний вірус" появився пізніше - офіційно вважається, що його вперше ужив співробітник Лехайського університету (США) Фред Коен в 1984 р. на 7-й конференції з безпеки інформації, яка проводилася в США.

Що таке комп'ютерний вірус?Формальне визначення цього поняття досі не придумано, і є серйозні сумніви, що воно взагалі може існувати. Численні спроби дати «сучасне» визначення вірусу не привели до успіху. Щоб відчути всю складність проблеми, попробуйте, наприклад, дати визначення поняття "редактор". Ви або придумаєте щось дуже загальне, або почнете перераховувати всі відомі типи редакторів. І те, й інше навряд чи можна вважати прийнятним. Тому ми обмежимося розглядом деяких властивостей комп'ютерних вірусів, які дозволяють говорити про них як про деякий певний клас програм.

Комп'ютерний вірус - це програма, здатна до самовідтворення. Така здатність є єдиним засобом, притаманним всім типам вірусів. Але не тільки віруси





здатні до самовідтворення. Будь-яка операційна система і ще безліч програм здатні створювати власні копії. Копії ж вірусу не тільки не зобов'язані повністю збігатися з оригіналом, але й можуть взагалі з ним не співпадати! Вірус не може існувати в "повній ізоляції": сьогодні не можна представити собі вірус, який не використовує код інших програм, інформацію про файлову структуру або навіть просто імена інших програм. Причина зрозуміла : вірус повинен якимось способом забезпечити передачу собі управління. Програма, всередині якої знаходиться вірус, називається "**зараженою**". Коли така програма починає роботу, то спочатку, як правило, управління отримує вірус. Вірус знаходить і "заражає" інші програми або виконує якісь шкідливі функції: псує файли чи таблицю розміщення файлів на диску, "засмічує" оперативну пам'ять, змінює адресацію звернень до зовнішніх пристроїв і т.і. Більше того, заражені програми можуть бути перенесені на інший комп'ютер за допомогою дискет або локальної мережі.

В даний час **відомо більше тридцяти тисяч вірусів**. Умовно вони поділяються на класи за такими ознаками:
• середовище проживання вірусу;
• спосіб зараження середовища проживання;
• деструктивні можливості;
• особливості алгоритму.

**За середовищем існування** розрізняють віруси мережеві, файлові, завантажувальні і спеціальні. Мережеві віруси поширюються по комп'ютерній мережі, файлові впроваджуються у виконувані файли, тобто у файли, що мають розширення COM і EXE. Файлові віруси можуть впроваджуватися й у інші типи файлів, але, як правило, записані в таких файлах, вони ніколи не отримують управління і, отже, втрачають здатність до розмноження. Завантажувальні віруси впроваджуються в завантажувальний сектор диска (Boot-сектор) або в сектор,

утримуючиий програму завантаження системного диска (Master Boot Record). Спеціальні орієнтовані на конкретні особливості програмного забезпечення (ПО), наприклад вірус, що заражає документи редактора Word. Існують сполучення - наприклад, файлово-завантажувальні віруси, що заражають і файли, і загрузочний сектор дисків. Крім того, по мережі можуть поширюватися віруси будь-яких типів.

**За способом зараження** віруси поділяються на резидентні і нерезидентні. Резидентний вірус при зараженні (інфікуванні) комп'ютера залишає в оперативній пам'яті свою резидентну частину, яка потім перехоплює звернення операційної системи до об'єктів зараження (файлів, завантажувальних секторів дисків і т.і.) і впроваджується в них. Резидентні віруси знаходяться в пам'яті і є активними аж до вимикання або перезавантаження комп’ютера. Нерезидентні віруси не заражають пам'ять комп'ютера і є активними обмежений час. Деякі віруси залишають в оперативній пам’яті невеликі резидентні програми, які не поширюють вірус. Такі віруси вважаються нерезидентними.

**За деструктивним можливостями** віруси можна розділити:

• на нешкідливі, ніяк не впливають на роботу комп'ютера (крім зменшення вільної пам'яті на диску в результаті свого поширення);

• безпечні, вплив яких обмежується зменшенням вільної пам'яті на диску і графічними, звуковими та іншими ефектами;

• небезпечні віруси, які можуть привести до серйозних збоїв у роботі комп'ютера;

• дуже небезпечні віруси, які можуть привести до втрати програм, знищити дані, сприяти прискореному зносу або





пошкодження частин механізмів (наприклад, головок вінчестерів).

**За особливостями алгоритму** віруси важко класифікувати через їх велику різноманітність. Найпростіші віруси - паразитичні, вони змінюють вміст файлів і секторів диска і можуть бути досить легко виявлені і знищені. Можна відзначити **віруси-реплікатори, звані хробаками**, які поширюються по комп'ютерних мережах, обчислюють адреси мережних комп'ютерів і записують за цими адресами свої копії. Відомі віруси-невидимки, звані стелс-вірусами, які дуже важко виявити і знешкодити, так як вони перехоплюють звертання операційної системи до уражених файлів і секторів дисків і підставляють замість свого тіла незараженіе ділянки диска. Найважче знайти віруси-мутанти, що підтримують алгоритми шифрування-розшифрування, завдяки яким копії одного і того ж вірусу не мають жодного повторюваного ланцюжка байтів. Маються і так звані квазівірусні або "троянські» програми, які хоч і не здатні до само поширення, але дуже небезпечні, тому що, масируючись під корисну програму, руйнують завантажувальний сектор і файлову систему дисків.

### 3.1.1. Основні види вірусів і схеми їх функціонування

Серед усього розмаїття вірусів можна виділити такі основні групи:

• завантажувальні;

• файлові;

• файлово-завантажувальні.

Тепер детальніше про кожну з цих груп.

Завантажувальні віруси

Розглянемо схему функціонування дуже простого завантажувального вірусу, який заражає дискети. Що відбувається, коли ви вмикаєте комп'ютер? Насамперед управління передається програмі початкового завантаження, яка зберігається в постійно

запам'ятовуючому пристрої (ПЗУ), тобто ПНЗ ПЗУ. Ця програма тестує обладнання та при успішному завершенні перевірок намагається знайти дискету в дисководі А: \. Всяка дискета розмічена на так звані сектори і доріжки. Сектори об'єднуються в кластери, але це для нас несуттєво. Серед секторів є кілька службових, використовуваних операційною системою для власних потреб (у цих секторах не можуть розміщуватися ваші дані). Серед службових секторів нас поки цікавить один - сектор початкового завантаження (boot-sector). У секторі початкового завантаження зберігається інформація про дискету - кількість поверхонь, кількість доріжок, кількість секторів і т.і.

Тепер розглянемо вірус. У завантажувальних вірусах виділяють дві частини - голову і хвіст. Хвіст, взагалі кажучи, може бути порожнім.

Нехай у вас є чиста дискета і заражений комп'ютер, під яким ми розуміємо комп'ютер з активним резидентним вірусом. Як тільки цей вірус виявить, що в дисководі з'явилася відповідна жертва - в нашому випадку не захищена від запису і ще не заражена дискета, він приступає до зараження. Заражаючи дискету, вірус виробляє наступні дії:

• виділяє деяку область диска і позначає її як недоступну операційній системі, це можна зробити по-різному, у найпростішому і традиційному випадку зайняті вірусом сектори позначаються як збійні (bad);

• копіює у виділену область диска свій хвіст і оригінальний (здоровий) завантажувальний сектор;

• заміщає програму початкового завантаження в завантажувальному секторі (дійсному) своєю головою, організовує ланцюжок передачі управління.

Таким чином, голова вірусу тепер першою одержує управління, вірус встановлюється в пам'ять і передає





управління оригінальному завантажувальному сектору, в ланцюжку

ППЗ (ПЗУ) - ПНЗ (диск) - СИСТЕМА з'являється нова ланка:

ППЗ (ПЗУ) - ВІРУС - ПНЗ (диск) - СИСТЕМА

**Мораль зрозуміла: ніколи не залишайте (випадково) дискет в дисководі А.**

Ми розглянули схему функціонування простого завантажувального вірусу, що живе у завантажувальних секторах дискет. Як правило, віруси здатні заражати не тільки завантажувальні сектори дискет, але і завантажувальні сектори вінчестерів. При цьому на відміну від дискет на вінчестері є два типи загрузочних секторів, що містять програми початкового завантаження, які отримують управління. При завантаженні комп'ютера з вінчестера перший бере на себе управління програмою початкового завантаження в MBR (Master Boot Record – головний завантажувальний запис). Якщо ваш жорсткий диск розбитий на кілька розділів, то лише один з них позначений як завантажувальний (boot).Програма початкового завантаження в MBR знаходить завантажувальний розділ вінчестера і передає управління на програму початкового завантаження цього розділу. Коди останньої збігаються з кодами програми початкового завантаження, що містяться на звичайних дискетах, а відповідні завантажувальні сектори відрізняються тільки таблицями параметрів. Таким чином, на вінчестері є об'єкт атаки завантажувальних вірусів: програма початкового завантаження в MBR і програма початкового завантаження в бут-секторі завантажувального диска.

**Файлові віруси**

Розглянемо тепер схему роботи простого файлового вірусу. На відміну від завантажувальних вірусів, які практично завжди резидентні, файлові вируси зовсім не

обов'язково резидентні. Розглянемо схему функціонування нерезидентного файлового вірусу. Нехай у нас є інфікований виконуваний файл. При запуску такого файлу вірус отримує управління, проводить деякі дії і передає управління "господарю" (тобто самій програмі).

Які ж дії виконує вірус? Він шукає новий об'єкт для зараження - який підходить за типом файла, що ще не заражений (у тому випадку, якщо вірус «пристойний", а то трапляються такі, що заражають відразу, нічого не перевіряючи). Заражаючи файл, вірус впроваджується в його код, щоб отримати управління при запуску цього файлу. **Окрім своєї основної функції - розмноження**, вірус цілком може зробити що-небудь хитромудре (сказати, запитати, зіграти) - це вже залежить від фантазії автора вірусу. Якщо файловий вірус резидентний, то він встановиться в пам'ять і одержить можливість заражати файли і проявляти інші здібності лише під час роботи зараженого файла. Заражаючи виконуваний файл, вірус завжди змінює його код - отже, зараження виконуваного файлу завжди можна виявити. Але, змінюючи код файла, вірус не обов'язково вносить інші зміни:

- не міняє довжину файлу;
- не використовує ділянки коду;
- не змінює початок файлу.

Нарешті, до файлових вірусів часто відносять віруси, які "мають деяке відношення до файлів", але не зобов'язані впроваджуватися в їх код. Розглянемо як приклад схему функціонування вірусів відомого сімейства Dir-II. Не можна не визнати, що з'явившись у 1991 р., ці віруси стали причиною справжньої епідемії комп'ютерної чуми в Росії. Розглянемо модель, на якій ясно видно основну ідею вірусу. Інформація про файли зберігається в каталогах. Кожен запис каталогу включає в себе ім'я файлу, дату і час ство ¬ данія, деяку додаткову інформацію, номер першого





кластера фай ¬ ла і так звані резервні байти. Останні залишені «про запас» і са ¬ мій MS-DOS не використовуються.

При запуску виконуваних файлів система зчитує із запису в каталозі перший кластер файлу і далі всі інші кластери. Віруси сімейства Dir-II виробляють наступну "реорганізацію" файлової системи: сам вірус записується в деякі вільні сектори диска, які він позначає як збійні. Крім того, він зберігає інформацію про перші кластери виконуємих файлів в резервних бітах, а на місце цієї інформації записує посилання на себе.

Таким чином, при запуску будь-якого файлу вірус отримує управління (операційна система запускає його сама), резидентно встановлюється в пам'яті і передає управління викликаному файлу.

**Завантажувально-файлові віруси**

Ми не станемо розглядати модель завантажувально-файлового вірусу, тому що ніякої нової інформації ви при цьому не дізнаєтеся. Але тут представляється нагода коротко обговорити вкрай "популярний" в один час завантажувально-зочний-файловий вірус OneHalf, що заражає головний завантажувальний сектор (MBR) і виконувані файли. Основна руйнівна дія - шифрування секторів вінчестера. При кожному запуску вірус шифрує чергову порцію секторів, а зашифрувавши половину жорсткого диска, радісно повідомляє про це нам. Основна проблема при лікуванні даного вірусу полягає в тому, що недостатньо просто видалити вірус з MBR і файлів, треба розшифрувати зашифровану їм інформацію. Найбільш "смертельна" дія - просто переписати новий здоровий MBR. Головне - не панікуйте. Зважте все спокойно, порадьтеся з фахівцями.

**Поліморфні віруси**

Більшість питань пов'язано з терміном "поліморфний вірус". Цей вид комп'ютерних вірусів представляється на сьогоднішній день **найбільш небезпечним**. Пояснимо ж, що це таке.

Поліморфні віруси - віруси, що модифікують свій код в заражених програмах таким чином, що два примірники одного і того ж вірусу можуть не збігатися ні в одному біті. Такі віруси не тільки шифрують свій код, використовуючи різні шляхи шифрування, але й містять код генерації шифровщика і розшифровувача, що відрізняє їх від звичайних шифрувальних вірусів, які також можуть шифрувати ділянки свого коду, але мають при цьому постійний код шифрувальника і розшифровувача. Поліморфні віруси - це віруси з самомоінфікуючим розшифровщиком. Мета такого шифрування: маючи заражений і оригінальний файли, ви все одно не зможете проаналізувати його код за допомогою звичайного дизассемблирування. Цей код зашифрований і являє собою безглуздий набір команд. Розшифровка виробляється самим вірусом вже безпосередньо під час виконання. При цьому можливі варіанти: він може розшифрувати себе всього відразу, а може виконати таку розшифровку «по ходу» де, може знову шифрувати вже відпрацьовані ділянки. Все це робиться заради труднощів аналізу коду вірусу.

### 3.1.2. Шляхи проникнення вірусів у комп'ютер і механізм розподілу вірусних програм

Основними шляхами проникнення вірусів у комп'ютер є зйомні диски (гнучкі й лазерні), а також комп'ютерні мережі. Зараження жорсткого диска вірусами може відбутися при завантаженні програми з дискети, що містить цей вірус. Таке зараження може бути випадковим, наприклад, якщо дискету не вийняли з дисководу А і перезавантажили комп'ютер, при цьому дискета може бути





і не системною. Заразити дискету набагато простіше. На неї вірус може потрапити, навіть якщо дискету просто вставили в дисковод зараженого комп’ютера і, наприклад, прочитали її зміст.

Вірус, як правило, впроваджується в робочу програму таким чином, щоб при її запуску управління спочатку передалося йому і тільки після виконання всіх його команд знову повернулося до робочої програми. Отримавши доступ до управління, вірус насамперед переписує сам себе в іншу робочу програму і заражає її. Після запуску програми, що містить вірус, стає можливим зараження інших файлів. Найчастіше вірусом заражаються завантажувальний сектор диска і виконувані файли, що мають розширення EXE, COM, SYS, BAT. **Вкрай рідко заражаються текстові файли.**

Після зараження програми вірус може виконати якусь диверсію, не дуже серйозну, щоб не привернути уваги. І нарешті, не забуває повернути керування тій програмі, з якої був запущений. Кожне виконання зараженої програми переносить вірус у наступну. Таким чином , заразиться все програмне забезпечення.

### 3.1.3. Ознаки появи вірусів

При зараженні комп'ютера вірусом важливо його виявити. Для цього слід знати про основні ознаки прояву вірусів. До них можна віднести такі:

- Припинення роботи або неправильна робота раніше успішно функціонувавших раніше програм;
- Повільна робота комп'ютера;
- Неможливість завантаження операційної системи;
- Зникнення файлів і каталогів чи спотворення їх вмісту;
- Зміна дати і часу модифікації файлів;
- Зміна розмірів файлів;
- Несподіване значне збільшення кількості файлів на диску;

- Істотне зменшення розміру вільної оперативної пам'яті;
- Виведення на екран непередбачених повідомлень або зображень;
- Подача непередбачених звукових сигналів;
- Часті зависання і збої в роботі комп'ютера.

Слід зазначити, що перелічені вище явища необов'язково викликають присутністю вірусу, а можуть бути наслідком інших причин. Тому завжди утруднена правильна діагностика стану комп'ютера.

Отже, можна навести масу фактів, які свідчать про те, що загроза інформаційного ресурсу зростає з кожним днем, піддаючи паніці відповідальних осіб у банках, на підприємствах і в компаніях в усьому світі. І при цьому загроза ця виходить від комп'ютерних вірусів, які спотворюють або знищують життєво важливу, цінну інформацію, що може призвести не тільки до фінансових втрат, але й до людських жертв.

**Комп'ютерний вірус** - спеціально написана програма, здатна мимовільно приєднуватися до інших програм, створювати свої копії та впроваджувати їх у файли, системні області комп'ютера і в обчислювальні мережі з метою порушення роботи програм, псування файлів і каталогів,створення різноманітних перешкод у роботі комп'ютера.

В даний час відомо більше **40 000** програмних вірусів, число яких безперервно зростає. Відомі випадки, коли створювалися навчальні посібники що, допомагають у написанні вірусів.

Нагадаємо основні види вірусів: **завантажувальні, файлові,файлово-завантажувальні.** Найбільш небезпечний вид вірусів - поліморфні.

З історії комп'ютерної вірусології ясно, що будь-яка оригінальна комп'ютерна розробка змушує творців





антивірусів пристосовуватися до нових технологій, постійно удосконалюючи антивірусні програми.

Причини появи і розповсюдження вірусів приховані, з одного боку, в психології людини, з іншого боку, з відсутністю засобів захисту у операціойної системи.

Основні шляхи проникнення вірусів - знімні диски та комп'ютерні мережі. Щоб цього не сталося, дотримуйтесь заходів щодо захисту. Також для виявлення , видалення і захисту від комп'ютерних вірусів розроблено декілька видів спеціальних програм, які називаються антивірусними. Якщо ви все ж виявили в комп'ютері вірус, то за традиційним підходом краще покликати професіонала, щоб той далі розібрався.

Але деякі властивості вірусів спантеличують навіть фахівців. Ще зовсім недавно важко було собі уявити, що вірус може пережити холодне перезавантаження або розповсюджуватися через файли документів. У таких умовах не можна не надавати значення хоча б початковоій антивірусній освіті користувачів.

### 3.2. Люки

Люком називається не описана в документації на програмний продукт можливість роботи з цим програмним продуктом. Сутність використання люків полягає в тому, що при виконанні користувачем деяких не описаних в документації дій він отримує доступ до можливостей і даних, які в звичайних умовах для нього закриті (зокрема, вихід в привілейований режим).

Люки найчастіше є результатом забудькуватості розробників. У процесі розробки програми розробники часто створюють тимчасові механізми, що полегшують ведення налагодження за рахунок прямого доступу до відладжуваної частини продукту. По закінченню налагодження більшість люків забирається з програми; але

люди є люди - часто вони забувають про існування якихось дрібних "лючків".

Одним з найбільш показових прикладів використання "забутих" люків є, мабуть, широко відомий в комп'ютерному світі інцидент з вірусом Морріса. Однією з причин, що зумовили можливість поширення цього вірусу, була помилка розробника програми електронної пошти, що входить до складу однієї з версій операційної системи UNIX, яка призвела до появи малопомітного лючка. Для вас, напевно, буде корисно знати, що американські фахівці оцінюють збитки, завдані в результаті цього інциденту, більш ніж в 100 мільйонів доларів.

Люки можуть утворюватися також в результаті часто практикуємих технологій розробки програмних продуктів "зверху вниз". При цьому програміст приступає відразу до написання керуючої програми,замінюючи майбутні підпрограми так званими "заглушками". У теорії моментом завершення розробки кінцевої програми за такою технологією можна вважати момент заміни останньої заглушки реальною підпрограмою.

Насправді справа дещо складніша. Вся біда в тому, що автори часто залишають заглушки в кінцевому програмному продукті, який передається в експлуатацію. Роблять це часом ненавмисно: наприклад, на ранніх стадіях розробки передбачалася наявність в кінцевому програмному продукті деякої підпрограми, проте в процесі розробки з'ясувалося, що ця підпрограма в силу яких-небудь причин не потрібна. Але заглушка-то залишилася! Видалити заглушку, не замінюючи її підпрограмою, буває досить складно. Це може спровокувати програміста залишити заглушку "до кращих часів".

Можливий варіант, коли заглушки залишаються у кінцевій програмі свідомо, в розрахунку на підключення надалі до працюючої програми нових підпрограм, що реалізовують





деякі нові можливості,або передбачаючи можливе підключення до програми тестуючих засобів для більш точного налаштування програми. Хто може дати гарантію, що в один прекрасний день такою заглушкою хто-небудь не скористається для підключення до програми зовсім іншої підпрограми, що працює в інтересах цього "когось", а не законного власника?

Нарешті, ще одним поширеним джерелом люків є так званий "невизначене введення". Не так вже рідкісна ситуація, коли програми створюються недосвідченими програмістами, що виходять з припущення, що користувачі будуть працювати з його програмою завжди коректно. У цьому випадку реакція на невизначене введення може бути в кращому разі непередбаченою. Набагато гірше, якщо програма у випадку однакового невизначеного введення виконує деякі повторювані дії - це дає потенційно загарбникові можливість планувати свої дії з порушенням безпеки.

Таким чином, люк (або люки) може бути присутнім в програмі з огляду на те, що програміст:

1) забув видалити його;

2) навмисне залишив його в програмі для забезпечення тестування або виконання частини налагодження;

3) навмисне залишив його в програмі в інтересах полегшення кінцевого складання програмного продукту;

4) навмисне залишив його в програмі з тим, щоб мати приховане середовище доступу до програми вже після того, як вона увійшла до складу кінцевого продукту.

### 3.3. Троянські коні

Існують програми, що реалізують, крім функцій, описаних у документації, і деякі інші функції, в документації не описані. Такі програми називаються "троянськими кіньми".

### 3.4. Логічна бомба

Логічною бомбою зазвичай називають програму або навіть ділянку коду в програмі, який реалізує деяку функцію при виконанні певної умови.

Світова комп'ютерна громадськість досить добре знайома з логічними бомбами. Логічна бомба є одним з улюблених способів помсти програмістів компаніям, які їх звільнили або чим-небудь образили. При цьому найчастіше спрацьовування бомби ставиться в залежність від установки в системі дати - так звані "вартові" бомби. Це дуже зручно: допустимо, програміст знає, що його звільнять 1 березня; в такому випадку він може встановити "годинну" бомбу на вибух, припустимо, 6 липня або навіть на Різдво, коли сам він буде вже поза межами досяжності для постраждалої компанії.

У цьому відношенні цікава висловлена одним з адміністраторів систем думка, що при звільненні системного програміста буде краще, якщо глава фірми проводить його до дверей офісу, чемно попрощається і подасть пальто.

### 3.5. Програмні закладки

Для того щоб закладка змогла виконати будь-які функції по відношенню до іншої прикладної програми, вона повинна отримати управління на себе, тобто процесор повинен почати виконувати інструкції, пов'язані з кодом закладки. Це можливо тільки при одночасному виконанні двох умов: закладка повинна перебувати в оперативній пам'яті до початку роботи програми, на яку спрямовано її вплив; закладка повинна активізуватися по деякій загальній для закладки і для прикладної програми події.

Це досягається шляхом аналізу і обробки закладкою загальних щодо неї та прикладної програми подій, наприклад переривань. Причому дані події повинні





супроводжувати роботу прикладної програми або роботу всієї ПЕОМ.

Виконання коду закладки може бути супроводжено операціями несанкціонованого запису (НСЗ), наприклад для збереження деяких фрагментів інформації, і несанкціонованого зчитування (НЗЧ), яке може проходити окремо від операцій читання прикладної програми або спільно з нею. При цьому операції зчитування і запису, можливо, не пов'язані з отриманням конфіденційної інформації, наприклад зчитування параметрів пристрою або його ініціалізація (закладка може використовувати для своєї роботи і такі операції, зокрема,для ініціювання збійних ситуацій або переназначення вводу-виводу).

Несанкціонований запис закладкою може відбуватися:

- в масив даних, який не збігається з користувацькою інформацією, - збереження інформації;
- в масив даних, що співпадає з користувацькою інформацією або її підмножиною, - спотворення, знищення або нав'язування інформації закладкою.

Отже, можна розглядати три основні групи деструктивних функцій, які можуть виконуватися закладками:

- збереження фрагментів інформації, що виникає при роботі користувачів, прикладних програм, введенні-виведенні даних, у зовнішній пам'яті мережі (локальної або віддаленої) або виділеної ПЕОМ, у тому числі різних паролей, ключів та кодів доступу, власне конфіденційних документів в електронному вигляді;
- зміна алгоритмів функціонування прикладних програм (тобто цілеспрямований вплив у зовнішній або оперативній пам'яті), наприклад програма розмежування доступу стане пропускати користувачів з будь-яким паролем ;
- нав'язування деякого режиму роботи (наприклад, при знищенні інформації - блокування запису на диск, при цьому інформація, природно, не знищується) або заміна

записуваної інформації інформацією, нав'язаної закладкою (наприклад,при виведенні на екран слова "невірно" замінюється словом "вірно",а "рубль" - "долар" і т.і.).

Наведемо кілька важливих прикладів. Припустимо, що програмний засіб виробляє деякі файлові операції. Для цього відкривається файл, частина його зчитується в буфер оперативної пам'яті, обробляється і після записується у файл з колишнім або новим ім'ям.

Тепер уявіть собі, що вводяться вами документи і не записуються на диск або записуються в спотвореному вигляді, або суто конфіденційна інформація банку крім запису в базу даних доповнено до файлу, послань в мережу. Або ви дали команду зашифрувати файл для передачі, а файл відправили "в дорогу" в незашифрованому вигляді. . Такими є лише деякі з широко відомих негативних дій, що можуть проводитися закладками з файлами-документами.

Розглянуті дії особливо небезпечні для програм підтвердження достовірності електронних документів ("електронний цифровий підпис" - ЕЦП). При зчитуванні приготованого для підпису файла - документа може відбутися зміна імені автора, дати, часу, цифрових даних, заголовка документа (наприклад, зміна суми платежу у платіжних дорученнях тощо).

Дані такого типу можуть, наприклад, нав'язувати істинність електронного підпису, навіть якщо файл був змінений.

Широко відома і використовувалася в багатьох банках система ЕЦП Pretty Good Privace (PGP). Багато хто став жертвою найпростішої програмної закладки проти цієї системи.

**Розглянемо процес "електронного підписування",** реалізований в програмі PGP. Програма зчитує файл для обчислення хеш-функції блоками по 512 байт, причому





завершенням процесу читання є зчитування блоку меншої довжини.

Робота закладки була заснована на нав'язуванні довжини файлу. Закладка дозволяє програмі ЕЦП вважати тільки перший 512-байтний блок і обчислювати підпис тільки на його основі.

Така ж схема діє і при перевірці підпису. Отже, основна частина файлу може бути довільним чином викривлена. Практика застосування ЕЦП в системах автоматизованого фінансового документообігу показала, що саме програмна реалізація ЕЦП найбільш сильно схильна до впливу з боку програмних закладок, які дозволяють здійснювати проводки завідомо фальшивих фінансових документів і втручатися в порядок вирішення спорів за фактом застосування ЕЦП.

Відзначимо чотири основні методи впливу програмних закладок на ЕЦП:

1. Метод нав'язування вхідної інформації - пов'язаний з перекручуванням поступаю чого на підпис файлу.
2. Метод нав'язування результату перевірки - пов'язаний з впливом на ознаку правильності підпису незалежно від результатів роботи.
3. Метод нав'язування довжини повідомлення - пред'явлення програмі ЕЦП електронного документа меншої довжини, отже, виробляється підпис тільки частини документа.
4. Метод спотворення програми ЕЦП - пов'язаний зі зміною виконуваного коду самої програми ЕЦП.

Завдання боротьби з програмними закладками вельми складна і многопланова. Можна розглядати наступні умови її вирішення:

1. Невідома наявність в якій-небудь безлічі програм фрагментів закладок, ставиться задача визначення факту їх

наявності або відсутності;при цьому програми не виконуються (статична задача).

2. В умовах п. 1 програми використовуються за своїм призначенням, ставиться те ж завдання виявлення закладки, але в даному випадку - за результатами роботи (динамічна задача).

3. Відбувається обмін програмним продуктом (у просторі - передача по каналу зв'язку або на матеріальному носії, у часі - зберігання), вільний від потенційно небезпечних дій, програмний продукт не виконуючи, завдання захисту (статична) ставиться в трьох варіантах: не допустити поза закладки; виявити впроваджений код закладки; видалити впроваджений код закладки.

4. В умовах п. 3 вирішується динамічна задача - захист від впливу закладки в ході роботи програм.

5. В умовах потенційної можливості впливу закладок вирішується завдання боротьби з їх підсумковим впливом, тобто закладки присутні в системі, але або не активні при виконанні критичних дій прикладних програм, або результат їх впливу не конструктивний.

Далі розглянуті задачі будемо згадувати як завдання 1-5.

З іншого боку, методи боротьби з впливом закладок можна розділити на класи і пов'язати з проблемою захисту програмного забезпечення взагалі:

1. Загальні методи захисту програмного забезпечення, які вирішують завдання боротьби з випадковими збоями обладнання та несанкціонованим доступом:

**а.**Контроль цілісності системних областей, що запускаються прикладних програм і використовуваних даних (рішення Завдання 3).

**б.** Контроль критичних для безпеки системи подій (рішення задачі 2).





Методи а і б дієві лише тоді, коли контрольні елементи не можуть бути чутливими до закладок і руйнівний вплив входить до контролює мого класу. Так, наприклад, система контролю за викликом переривань не буде відслідковувати обіг на рівні портів. Контроль може бути обійдений шляхом:

• нав'язування кінцевого результату перевірок;

• впливу на процес зчитування інформації;

• зміни хеш-функцій, що зберігаються в загальнодоступних файлах або в оперативній пам'яті.

Важливо, що включення процесу контролю має бути виконано до початку впливу закладки, або контроль повинен здійснюватися повністю апаратними засобами з програмами управління, що містяться в ПЗУ.

**в.** Створення безпечного та ізольованого операційного середовища (рішення задачі 4).

**г.** Запобігання результуючого впливу вірусу або закладки (наприклад, запис на диск тільки в зашифрованому на рівні контролера вигляді де - тим самим збереження інформації закладкою не має сенсу, або заборона запису на диск на апаратному рівні) (рішення завдання 5).

**2.Спеціальні методи виявлення програм з потенційно** небезпечними наслідками: пошук фрагментів коду за характерними наслідками (сигнатурами), властивим закладкам, або, навпаки,дозвіл на виконання або впровадження в ланцюжок переривань тільки програмам з відомими сигнатурами (рішення задач 1,).Пошук критичних ділянок коду методом Семантичного аналізу, тобто аналізу фрагментів коду на виконувані ними функції, наприклад виконання НСЗ часто пов'язане з дізасемблюванням або емуляцією виконання (рішення задач 1, 2).

Дуже важливим є комплекс організаційно-технічних заходів захисту від вірусів і програмних закладок.

Заходи захисту можна поділити на дві основні групи:
I. Заходи захисту на етапі розробки програмного забезпечення (ПЗ) системи.
II. Заходи захисту на етапі експлуатації.
До групи I входять:
1.Заходи захисту на етапі розробки прикладного ПЗ, що містить внутрішній захист від несанкціонованого доступу (НСД).
Вони спрямовані на виявлення у вихідних текстах програм комуникацій і доступу деяких фрагментів або підпрограм, які полегшують або не реєструють доступ розробників програм (вхід за фіксованими паролями, безпарольний доступ після натискання деяких клавіш,обхід регістра дії користувачів з фіксованими іменами і т.і.). Наявність таких фрагментів фактично зведе нанівець весь комплекс інформаційної безпеки системи, оскільки доступ через них можливий як людині, так і програмній закладці.
2.Заходи захисту при розробці ПЗ захисту від НСД (повинні бути передбачені заходи з перевірки цілісності збережених на зовнішніх носіях програмних засобів захисту, контроль цілісності їх в оперативній пам'яті і т.і.).
До групи II входять:
1.Регулярні заходи захисту та контролю, що застосовуються постійно з фіксованими часовими інтервалами.
2. Епізодичні захисні заходи (на додаток до п. 1 в період підвищення небезпеки вірусного нападу).
3.Локалізаційно-відновлювальні заходи, що застосовуються у разі проникнення і виявлення закладок та заподіяння ними негативних наслідків.
До загальних способів захисту системи відносяться:
- Обмеження фізичного доступу до програм та обладнання шляхом встановлення відповідного організаційного





режиму і застосування апаратних чи програмних засобів обмеження доступу до ПЕОМ та її компоненту;
- При активізації прикладного ПО контроль його цілісності, цілісності областей DOS, BIOS і CMOS шляхом прорахунку контрольних сум (обчислення хеш-функцій) і порівняння їх з еталонними значеннями для кожної ПЕОМ;
- Максимальне обмеження і контроль за передачею по мережі виконуємих файлів (типу.EXE і. COM),. SYS-і. BIN-файлів з метою запобігання розповсюдження файлових вірусів, вірусів типу Driver, завантажувально-файлових вірусів, а також розмноження закладок по мережі; використання фільтрів і шлюзів при передачі даних;
- Організація вибіркового і раптового контролю роботи операторів ПЕОМ з метою виявлення фактів використання нерегламентованої ПЗ;
- Захист від запису на магнітних носіях (дискетах), облік і надійне зберігання архівних копій;
- Негайне знищення цінної інформації одразу після закінчення потреби в ній;
- Періодична оптимізація зовнішніх носіїв (вінчестерів) на предмет виявлення збійних або псевдозбійних кластерів і стирання фрагментів конфіденційної інформації за допомогою засобів типу SPEED DISK.
Засоби захисту, що враховують специфіку роботи фрагментів системи:
а) для комунікаційних підсистем: засоби та методи підвищення загальної надійності системи (програмне або апаратне дублювання, використання "гарячого резерву" і т.і.);
б) для серверів локальних мереж (СЛС): контроль складу та порядку використування ПЗ, що знаходиться на СЛС; дублювання стандартних засобів захисту від НСД в ПО мережі Novell та інших різновидів мережевого ПО.

Передачі засобами захисту;заборона запису на загальний диск файла-сервера локальної мережі виконуваних файлів, що не мають відношення до обробки інформації в мережі.

Що ж стосується заходів захисту від вірусів і закладок в процесі розробки самих програм захисту (п.2 групи I), то в даному випадку необхідно передивитися: вбудований самоконтроль ПЗ системи захисту, встановлений на мережі, шляхом прорахунку контрольних сум по файлах і коду програм в оперативній пам'яті;перевизначення "на себе" істотно важливих переривань (int01h, 03h, 08h, 10h, 13h, 21h) для запобігання перехоплення введення ключів і паролів і їх збереження закладкою на зовнішньому носії,а також блокіровка проникнення в логіку роботи програм захисту за допомогою стандартних налагоджувальних засобів; захист від перенесення встановленого ПО захисту комунікації на іншу ПЕОМ, що проводиться з метою детального вивчення ПЗ та пошуку обхідних шляхів для подолання захисту;це може бути досягнуто "прив'язкою" ПО до індивідуальних параметрів ПЕОМ - тим самим работоздатність ПЗ буде забезпечуватися тільки на даній ЕОМ мережі.

### 3.6. Атака "салямі"

А тепер поговоримо про «бич» банківських комп'ютерних систем - атаці "салямі".

Щоб зрозуміти сенс такої атаки, корисно згадати технологію виготовлення відомого сорту ковбаси, яка створюється шляхом з'єднання в єдине ціле безлічі дрібних шматочків м'яса. Виходить досить смачно.

При розробці банківських систем встановлюється правило округлення (або усічення), що використовується при виконанні всіх операцій. Вся хитрість заключається в тому, як запрограмувати обробку заокруглень. Можна, звичайно, просто видаляти неіснуючі величини. Але можна й не видаляти, а накопичувати на якомусь спеціальному





рахунку. Там пів цента, тут півтора цента ... - А в сумі? Як свідчить практика, сума, складена буквально з нічого, за пару років експлуатації "хитрою" програмою в середньому за розміром банку може обчислюватися тисячами доларів. Можна сказати, що атака "салямі" - комп'ютерна реалізація відомої приказки "З миру по нитці - голому сорочка".

Програмні закладки при хаотичній "інформатизації" фінансової сфери стають потужним деструктивним фактором у її розвитку. Труднощі виявлення закладок і боротьби з їх впливом без перебільшення дає можливість назвати їх інформаційною зброєю.

Легко зрозуміти здивування і засмучення банкірів, які бачать завідомо фальшиві платіжні доручення, які системи електронного підпису вважають справжніми. Щоб уникнути подібних розчарувань, необхідно постійно пам'ятати про інформаційну загрозу і ще ... легенду про троянського коня.

## 4. ЗАРОДЖЕННЯ КРИПТОГРАФІЇ

Поняття "безпека" охоплює широке коло інтересів як окремих осіб, так і цілих держав. У наш мобільний час чільне місце Відводиться проблемі інформованої безпеки, забезпечення захисту конфіденціальной інформації від ознайомлення з нею конкуруючих груп. Недаром великий психолог Вільям Шекспір у "Королі Лірі" говорив: "Щоб думку ворога дізнатися, серця розкривають, а не те що листи".

Про важливість збереження інформації в таємниці знали вже в давні часи, коли з появою писемності з'явилася і небезпека прочитання її небажаними особами.

Існували три основні способи захисту інформації. Один з них передбачав захист її суто силовими методами: охорона документа - носія інформації - фізичними особами, передача його спеціальним кур'єром і т.і. Другий спосіб отримав назву "стеганографія" латино-грецьке сполучення

слів, що означають в сукупності "тайнопис"). Він полягав у прихованні самого факту наявності інформації. В даному випадку використовувалися так звані симпатичні чорнила. При відповідному "прояві" бумаги текст стає видимим. Один із прикладів приховування інформації приведені в працях давньогрецького історика Геродота. На голові раба, яка голилися наголо, записувалося потрібне повідомлення. І коли волосся його достатньо відростало, раба відправляли до адресата, який знову голив його голову і зчитував отримане повідомлення.

Третій спосіб захисту інформації полягав у перетворенні сенсового тексту в якийсь набір хаотичних знаків (або букв алфавіту). Одержувач даного донесення мав можливість перетворити його в те ж саме осмислене повідомлення, якщо володів ключом до його побудови. Цей спосіб захисту інформації називається криптографічним. Криптографія - слово грецьке і в перекладі означає "тайнопис". За твердженням ряду фахівців, криптографія за віком - ровесник єгипетських пірамід. У документах древ ¬ ніх цивілізацій - Індії, Єгипту, Месопотамії - є відомості про системи та способи складання шифрованих листів.

У давньоіндійських рукописах описані 64 способи письма. Один з найстаріших шифрованих текстів з Месопотамії є табличка, написана клинописом і містить рецепт для виготовлення глазурі для гончарних виробів. Для написання його були використані рідко вживаємі клинописні знаки, ігнорувалися деякі голосні і приголосні і вживалися числа замість імен.

Шифровані тексти Стародавнього Єгипту - це найчастіше релігійні тексти і медичні рецепти.

Абсолютно відсутні відомості про використання шифрів в Древньому Китаї, що пояснюється, очевидно, складністю використовуваного ієрогліфічного листа.





Найбільш повні і достовірні відомості про шифри відносяться до Стародавньої Греції.

Основне поняття криптографії - шифр (від арабського "цифра";араби першими стали замінювати літери на цифри з метою захисту вихідного тексту). Секретний елемент шифру, недоступний стороннім, називається ключом шифру. Як правило, в стародавні часи використовувалися так звані шифри заміни і шифри перестановки. Історичним прикладом шифру заміни є шифр Цезаря (I століття до н.е.), описаний істориком Стародавнього Риму Светонієм.. Гай Юлій Цезар використав у своєму листуванні шифр власного винаходу. Стосовно до сучасної російської мови він полягав у наступному. Виписувався алфавіт: А, Б, В, Г, Д, Е, ...; потім під ним виписувався той же алфавіт, але з зсувом на 3 літери вліво:

| А | Б | В | Г | Д | Е | ё | Ж | З | И | й | К | Л | М | Н | О | П | Р | С | Т | У | Ф | Х | Ц | Ч | Ш | щ | ы | Ь | Ъ | Э | Ю | Я |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Г | Д | Е | ё | Ж | З | И | й | К | Л | М | Н | О | П | Р | С | Т | У | Ф | Х | Ц | Ч | ш | щ | ы | ь | ъ | э | Ю | Я | А | Б | В |

При зашифровці літера А замінялася буквою Г, Б замінялася на Д, В - на Е і т.д. Так, наприклад, слово "РИМ" перетворювалося на слово "УЛП". Отримувач повідомлення "УЛП" шукав ці букви в нижньому рядку і по буквах над ними встановлював вихідне слово "РИМ". Ключем у шифрі Цезаря є величина зсуву третій нижнього рядка алфавіту. Наступник Юлія Цезаря - Цезар Август - використовував той же шифр, але з ключем - зрушення 4. Слово "РИМ" він у цьому випадку зашифрував би в буквосполучення "ФМР".

У художній літературі класичним прикладом шифру заміни являється відомий шифр "Танцюючі чоловічки" (К. Дойля). У ньому букви тексту замінювалися на символічні фігурки людей. Ключом такого шифру були пози чоловічків, які замінюють літери ...

Для прикладу шифру перестановки виберемо ціле позитивне число, скажімо 5; розташуємо числа від 1 до 5 у дворядковий запис

| | | | | |
|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 |
| 3 | 2 | 5 | 1 | 4 |

Зашифруємо фразу "СВЯЩЕННАЯ РИМСКАЯ ИМПЕРИЯ». У цій фразі 23 літери. Доповнимо її двома довільними літерами (наприклад, Ь, Е) до найближчого числа, кратного 5, тобто 25. Випишемо цю доповнену фразу без пропусків, одночасно розбивши її на п'ятизначні групи:
СВЯЩЕННАЯР ИМСКА ЯІМПЕ РІЯЬЕ. Букви кожної групи переставимо відповідно до зазначеного двострочного запису за наступним правилом: перша буква стає на третє місце, друга - на друге, третя на п'яте, четверта на перше і п'ята на четверте. Отриманий текст виписується без пропусків:
ЩВ СЕЯЯННРАКМИ АСПИЯЕМЬИРЭЯ. При розшифруванні текст 1 розбивається на групи по 5 букв і букви переставляются у зворотному порядку: 1 на 4 місце, 2 на 2, 3 на 1, 4 на 5 та 5 на 3. Ключом шифру є вибране число 5 і порядок розташування чисел у нижньому ряду дворядкового запису.
Одним з перших приладів, що реалізують шифр перестановки, є так званий прилад СЦИТАЛЛА. Він був винайдений в стародавній "варварській" Спарті за часів Лікурга; Рим швидко скористався цим приладом. Для зашифрування тексту використовувався циліндр заздалегідь обумовленого діаметра. На циліндр намотувався тонкий ремінь з пергаменту, і текст виписувався порядково по твірній циліндра (уздовж його осі). Потім ремінь згортався і вирушав до одержувачуа





повідомлення. Останній намотував його на циліндр того ж діаметру і читав текст по осі циліндра.У цьому прикладі ключом такого шифру був діаметр циліндра і його довжина, які, по суті, породжують дворядковий запис, зазначений вище.
Цікаво, що винахід дешифрувального пристрою "АНТИСЦИТАЛЛА" приписується великому Аристотеля. Він запропонував для цього використовувати конусоподібний "спис", на який намотувався перехоплений ремінь, який пересувався по осі до того стану, поки не з'являвся осмислений текст.
Були й інші способи захисту інформації, розроблені в античні часи. Давньогрецький полководець Еней Тактика в IV столітті до н.е. запропонував пристрій, названий згодом "диском Енея". Принцип його був простий. На диску діаметром 10-15 см і товщиною 1-2 см висвердлюються отвори по числу букв алфавіту. У центрі диска містилася "котушка" з намотаною на ній ниткою достатньої довжини. При шифруванні нитка "витягаючись" з котушки і послідовно простягалася через отвори, відповідно з буквами шифруємого тексту. Диск і був посланням. Одержувач посланя послідовно витягав нитку з отворів, що дозволяло йому одержувати передане повідомлення, але в зворотньому порядку проходження букв. При перехопленні диска недоброзичливець мав можливість прочитати повідомлення тим же чином, що і одержувач. Але Еней передбачив можливість легкого знищення переданого повідомлення при загрозі захоплення диска. Для цього було достатньо висмикнути "котушку" із закріпленим на ній кінцем нитки до повного виходу всієї нитки з усіх отворів диска.
Ідея Енея була використана в створенні інших оригінальних шифрів заміни. Скажімо, в одному з варіантів замість диска використовувалася лінійка з числом отворів,

рівних кількості букв алфавіту. Кожний отвір позначався своєї буквою; букви по отворах розташовувалися в довільному порядку. До лінійки була прикріплена котушка з намотаною на неї ниткою. Поряд з котушкою був проріз. При шифруванні нитка простягалася через проріз, а потім через отвір, відповідно першій букві шифруємого тексту, при цьому на нитці зав'язувався вузлик у місці проходження її через отвір; потім нитка поверталася в проріз і аналогічно зашифровувалася друга літера тексту і т.д. Після закінчення шифрування нитка видалялась і пе ¬ редавалась одержувачу повідомлення. Той, маючи ідентичну лінійку, простягав нитку через проріз до отворів, які визначаються вузлами, і відновлював текст по буквах отворів.

Цей пристрій одержав назву "лінійка Енея". Шифр, реалізований лінійкою Енея, є одним із прикладів шифру заміни: коли літери заміюються на відстані між вузликами з урахуванням проходження через проріз. Ключом шифру був порядок розташування літер по отворах в лінійці. Сторонній, отримавший нитку (навіть маючи лінійку, але без нанесених на ній літер), не зможе прочитати передане повідомлення.

Аналогічно "лінійці Енея" "вузликове письмо" отримало поширення в індіанців Центральної Америки. Свої повідомлення вони також передавали у вигляді нитки, на якій зав'язувалися різнокольорові вузлики, що визначали зміст отриманого повідомлення.

Помітним внеском Енея в криптографію є запропонований ним так званий книжковий шифр, описаний у творі "Про оборону укріплених місць". Еней запропонував проколювати малопомітні дірки в книзі або в іншому документі над літерами (або під ними) секретного повідомлення. Інтересно зазначити, що в Першій світовій війні німецькі шпигуни використовували аналогічний





шифр, замінивши дірки на точки, що наносяться симпатичними чорнилами на букви газетного тексту. Книжковий шифр в сучасному його вигляді має дещо інший вигляд. Суть цього шифру полягає в заміні букв на номер рядка та номер цієї букви в рядку і заздалегідь обумовленої сторінці деякої книги. Ключом такого шифру є книга і використовувана сторінка в ній. Цей шифр виявився "довгожителем" і застосовувався навіть у часи Другої світової війни.

Ще один винахід стародавніх греків - так званий квадрат Полібія. Стосовно до сучасного латинському алфавіту з 26 букв шифрування по цьому квадрату полягало в наступному. У квадрат розміром 5х6 клітин виписуються всі букви алфавіту, при цьому літери І, І не розрізняються (І ототожнюється з буквою І):

| | | | | | |
|---|---|---|---|---|---|
| А | В | С | Б | Е | |
| А | А | В | С | Б | Е |
| Б | Б | в | Н | І | К |
| С | Ь | м | N | О | Р |
| Б | | Я | Б | Т | и |
| Е | V | | Х | У | Ъ |

Шифруєма буква замінювалася на координати квадрата, в якому вона записана. Так, В замінювалося на АВ, Б на ВА, Я на БВ і т.д. При розшифруванні кожна така пара визначала відповідну букву повідомлення. Ключем та ¬ кого шифру було розташування букв в таблиці 5х5. Цікаво відзначити, що в дещо зміненому вигляді шифр Полібія дійшов до наших днів і отримав своєрідну назву "тюремний шифр". Для його використання потрібно тільки

знати природний порядок розташування літер алфавіту (як в зазначеному вище прикладі для англійської мови). Сторони квадрата позначаються не літерами (АВСБЕ), а числами (12345). Число 3, наприклад, передається шляхом потрійного стуку. При передачі літери спочатку "відстукується" число, відповідне рядку, в якому знаходиться буква, а після номер відповідного стовпця. Наприклад, літера "Б" передається подвійним стуком (другий рядок) і потім одинарним (перший стовпець).

Із застосуванням цього шифру пов'язані деякі історичні казуси. Так, декабристи, посаджені у в'язницю після невдалого повстання, не змогли встановити зв'язок з перебуваючим в "одиночці" князем Одоєвським. Виявилося, що цей князь (добре освічений на ті часи) не пам'ятав природний порядок розташування літер в російському і французькому алфавітах (іншими мовами він не володів). Декабристи для російського алфавіту використовували прямокутник розміру 5х6 (5 рядків і 6 стовпців) і скороченого до 30 літер алфавіту.

"Тюремний шифр", строго кажучи, не шифр, а спосіб перекодування спілкування з метою його приведення до вигляду, зручного для передачі по каналу зв’язку (через стінку). Справа в тому, що в таблиці використовувався природний порядок розташування букв алфавіту. Зазначимо, що при довільному рзташуванні букв в квадраті виникає одне утруднення: або потрібно пам'ятати відправнику і одержувачу повідомлення заданий довільний порядок проходження букв в таблиці (ключ шифру), що взагалі кажучи важко, або мати при собі запис цих букв. У другому випадку з'являється небезпека ознайомлення з ключом сторонніх осіб. Тому в ряді випадків ключ складається таким чином. Береться якесь "ключове слово", яке легко запам'ятати, наприклад "СЯУРТОЬООУ", потім з нього видаляють повтори букв (отримують "СЯУРТОЬОв")





і записують його в початкових клітинах квадрата. В остальні клітини записуються інші літери алфавіту в природному порядку.

А В С Б Е А С Я У Р Т В О Ь в А В С Б Е Е Н І Е и V Х Ъ

У такому шифрі ключем є вказане "ключове слово" ("пароль"). Зауважимо, до речі, що таким же чином можна легко запам'ятати порядок слідування букв і в "лінійці Енея".

Крах Священної Римської імперії породив Середньовіччя. Цей період у житті людства характеризується і занепадом інтелектуальної діяльності. У часи, коли сама грамотність була доступна дуже вузькому колу людей, необхідність у криптографічному захисті інформації стояла не так гостро. Так, король франків і Священної Римської імперії Карл Великий навчився читати і писати у віці 50 років, а "завойовник Всесвіту" Чингісхан залишився неписьменним на все життя. Проте Карл Великий вже знав і використовував деякі шифри заміни. Освіта і грамотність в ці часи зосередилися в церкві, і тайнопис став її монополією. Церкова ухвалила, що простим парафіянам не можна приховувати таємниці від "господа"; їх тайнопис - це "єресь". За використання тайнопису передбачалися жорсткі заходи покарання, аж до смертної кари.

Проте криптографія не померла. Серйозний внесок у її розвиток внесли араби. Деякі історики вважають, що криптографія як наука зародилася саме в арабському світі. Саме в арабських книгах вперше були описані методи криптоаналізу (дешифрування). Про криптографію згадується і в "Іліаді" Гомера.

## 5. ЕЛЕМЕНТАРНІ МЕТОДИ ЦИФРОВОГО ШИФРУВАННЯ

Серед усього спектру методів захисту даних від несанкціонованого доступу особливе місце займають

криптографічні алгоритми. На відміну від інших методів, вони спираються лише на властивості самої інформації і не використовують властивості її матеріальних носіїв, особливості вузлів її обробки, передачі і зберігання. Образно кажучи, криптографічні методи будують бар'єр між інформацією, що захищається і реальним або потенційним зловмисноком із самої інформації. Криптографічні методи захисту інформації в автоматизованих системах можуть застосовуватися як для захисту інформації, що обробляється в ЕОМ або що зберігається в різного типу ЗУ, так і для закриття інформації, переданої між різними елементами системи по лініях зв'язку. Криптографічне перетворення як метод попередження несанкціонованого доступу до інформації має багатовікову історію. В даний час розроблена велика кількість різних методів шифрування, створені теоретичні та практичні основи їх застосування. Переважна кількість цих методів може бути успішно використана і для закриття інформації.

Основні напрямки використання криптографічних алгоритмів - передача конфіденційної інформації з каналів зв'язку (наприклад, електронної пошти), встановлення автентичності переданих повідомлень, зберігання інформації (документів, баз даних) на носіях у зашифрованому вигляді.

Проблема використання криптографічних методів у інформаційних системах (ІС) стала зараз особливо актуальна. З одного боку, розширилося використання комп'ютерних мереж, зокрема глобальної мережі Інтернет, за якими передаються великі обсяги інформації державного , військового, комерційного і приватного характеру, що не допускає можливості доступу до неї сторонніх осіб. З іншого боку, поява нових потужних комп'ютерів, технологій мережевих і нейронних обчислень зробила





можливим дискредитацію криптографічних систем, що ще недавно вважалися практично нерозкриваємими.

### 5.1. Застосування підстановки

При підстановці окремі букви відкритого тексту замінюються іншими буквами або числами або якимись іншими символами. Якщо відкритий текст розглядається як послідовність бітів, то підстановка зводиться до заміни заданих послідовностей бітів відкритого тексту заданими послідовностями бітів шифрованого тексту.

#### 5.1.1. Шифр Цезаря

Найдавнішим і найпростішим з відомих символів шифрів є шифр, який використовував Юлій Цезар. У шифрі Цезаря кожна буква алфавіту замінюється буквою, яка знаходиться на три позиції далі в цьому ж алфавіті. Найпростіше побачити це на прикладі. Відкритий текст: meet me after the toga party. Зашифрований текст: PHHW PH DIWHU WKH WRJD SDUMB. Зверніть увагу на те, що алфавіт вважається "циклічним", тому після Z йде A. Визначити перетворення можна, перерахувавши всі варіанти, як показано нижче.

Відкритий текст: abcdefghijklmnopqrstuvwxyz

Зашифрований текст:

DEFGHIJKLMNOPQRSTUVWXYZABC Якщо кожній букві призначити числовий еквівалент (a = 1, b = 2 і т.д.), то алгоритм можна виразити наступними формулами. Кожна буква відкритого тексту p замінюється буквою шифрованого тексту C:

**C = E (p) = (p + 3) mod (26).** У загальному випадку зсув може бути будь-яким, тому узагальнений алгоритм Цезаря записується формулою

**C = E (p) = (p + k) mod (26),**

де k приймає значення в діапазоні від 1 до 25. Алгоритм дешифрування також простий:

**p = D (C) = (C - k) mod (26).**

Якщо відомо, що певний текст був шифрований за допомогою шифру Цезаря, то за допомогою простого перебору всіх варіантів розкрити шифр дуже просто - для цього достатньо перевірити 25 можливих варіантів ключів. На рис. 5.1 показані результати застосування цієї стратегії до зазначеного вище повідомлення. В даному випадку відкритий текст розпізнається в третьому рядку.

PHHW PH DIWHU WKH WRJD SDUWB
1 oggv og chvgt vjg vqic rctva
2 nffu nf bgufs uif uphb qbsuz
3 meet me after the toga party
4 ldds Id zesdq sgd snfz ozqsx
5 kccr kc ydrcp rfc rmey nyprw
6 jbbq jb xcqbo qeb qldx mxoqv
7 iaap ia wbpan pda pkcw lwnpu
8 hzzo hz vaozm ocz ojbv kvmot
9 gyyn gy uznyl nby niau julns
10 fxxm fx tymxk max mhzt itkmr
11 ewwl ew sxlwj lzw lgys hsjlq
12 dwk dv rwkvi kyv kfxr grikp
13 cuuj cu qvjuh jxu jewq fqhjo
14 btti bt puitg iwt idvp epgin
15 assh as othsf hvs hcuo dofhm
16 zrrg zr nsgre gur gbtn cnegl
17 yqqf yq mrfqd ftq fasm bmdfk
18 xppe xp lqepc esp ezrl alcej
19 wood wo kpdob dro dyqk zkbdi
20 vnnc vn jocna cqn cxpj yjach
21 ummb um inbmz bpm bwoi xizbg
22 tlla tl hmaly aol avnh whyaf
23 skkz sk glzkx znk zumg vgxze
24 rjjy rj fkyjw ymj ytlf ufwyd
25 qiix qi ejxiv xli xske tevxc





Рис. 5.1. Криптоанализ шифру Цезаря методом перебору всіх варіантів ключів

Застосування методу послідовного перебору всіх можливих варіантів виправдано наступними трьома важливими характеристиками даного шифру.

1. Відомі алгоритми шифрування і дешифрування.
2.Необхідно перебрати всього 25 варіантів.
3. Мова відкритого тексту відома і легко впізнавана.

У більшості випадків, коли мова йде про захист мереж, можна передбачати, що алгоритм відомий. Єдине, що робить криптоаналіз на основі методу послідовного перебору практично марним - це застосування алгоритму, для якого потрібно перебрати дуже багато ключів. Наприклад, алгоритм DES, використовує 56-бітові ключі, вимагає при послідовному переборі розглянути простір із 256, або більше ніж 7x1016 ключів.

Третя характеристика також важлива. Якщо мова, якою написаний відкритий текст, невідома, то розшифрований текст можна не розпізнати, тому що більше того, вихідний текст може складатися з скорочень або бути яким-небудь чином стиснутий - це також ускладнює розпізнавання.

**5.2. Моноалфавитні шифри**

При наявності всього 25 можливих варіантів ключів шифр Цезаря далекий від того, щоб вважатися надійно захищеним. Суттєвого розширення простору ключів можна домогтися, дозволивши використання довільних підстановки. Давайте ще раз пригадаємо шифр Цезаря.

Відкритий текст: abcdefghijklmnopqrstuvwxyz
Зашифрований текст
: DEFGHI JKLMNOPQRSTUVWXYZABC

Якщо в рядку "Зашифрований текст" допустити використання будь-якої з перестановок 26 символів алфавіту, то ми отримаємо 26!, Або більш ніж 4x1026

можливих ключів. Це на 10 порядків більше, ніж розмір простору ключів DES, і це здається достатнім для того, щоб зробити неможливим успішне застосування криптоаналізу на основі методу послідовного перебору.

Однак для криптоаналітика існує й інша лінія атаки. Якщо криптоаналітик має уявлення про природу відкритого тексту (наприклад, про те, що це незжатий текст англійською мовою), можна використовувати відому інформацію про характерні ознаки, властивих текстам на відповідній мові. Щоб показати, як цей підхід застосовується на практиці, розглянемо невеликий приклад. Припустимо, нам потрібно розшифрувати наступний шифрований текст:

**UZQSOVUOHXMOPVGPOZPEVSGZWSZOPFPESXUDBMETSXAIZ**
**VUEPHZHMDZSHZOWSFPAPPDTSVPQUZWYMXUZUHSX**
**EPYEPOPDZSZUFPOMBZWPFUPZHMDJUDTMOHMQ**

На першому етапі можна визначити відносну частоту появи в тексті різних букв і порівняти їх із середньостатистичними даними для букв англійської мови (рис. 5.2).

Якщо повідомлення досить довге, цієї методики вже може бути достатньо для розпізнавання тексту, але в нашому випадку, коли повідомлення невелике, точної відповідності очікувати не доводиться. У нашому випадку видно, що відносна частота входження букв в зашифрованому тексті (у відсотках) виявляється наступною:

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **р** | 13,33 | **Н** | **5,83** | **¥** | **3,33** | **В** | **1,67** | **С** | **0,00** |
| **%** | **11,67** | **Б** | **5,00** | | **3,33** | **Є** | **1,67** | **К** | **0,00** |
| **в** | **8,33** | **Е** | **5,00** | **<3** | **2,50** | **У** | **1,67** | **Ь** | **0,00** |





| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **и** | **8,33** | **V** | **4,17** | **Т** | **2,50** | **I** | **0,83** | **N** | **0,00** |
| **О** | **7,50** | **X** | **4,17** | **А** | **1,67** | **J** | **0,83** | **„В** | **0,00** |
| **м** | **6,67** | | | | | | | | |

Порівнюючи ці результати з даними, показаними на рис. 5.2, можна зазначити, що, швидше за все, букви Р і Ъ шифрованого тексту є еквівалентами букв е та 1 відкритого тексту, хоча важко сказати, який саме букві - Р або Ъ - відповідає е, а який - 1. Букви Б, и, О, М і Н, що володіють відносно високою частотою появи в тексті, швидше за все, відповідають буквам з множини {г, п, 1, о, а, 8}. Букви з низькою частотою появи (а саме А, В, в, У, I, І), мабуть, відповідають буквам множини V, Ь, к, х, ^], г}.

Далі можна піти декількома шляхами. Можна, наприклад, прийняти деякі припущення про відповідності і на їх основі спробувати відновити відкритий текст, щоб побачити, чи виглядає такий текст схожим на що-небудь осмислене. Більш систематизований підхід полягає в продовженні пошуку в тексті нових характерних закономірностей. Наприклад, може бути відомо, що в розглянутому тексті обов'язково повинні бути присутніми деякі слова. Або ж можна шукати повторювані послідовності букв шифрованого тексту і намагатися визначити їх еквіваленти у відкритому тексті.

Один з дуже ефективних методів полягає в підрахунку частоти використання комбінацій, що складаються з двох букв. Такі комбінації називають біграмами. Для значень відносної частоти появи в тексті біграм теж можна побудувати гістограму, подібну показаної на рис. 5.2. Відомо, що в англійській мові найпоширенішою є біграма 1Ь. У нашому шифрованому тексті найчастіше (три рази) зустрічається комбінація видання ^ ^ . Тому можна

припустити, що Ъ відповідає 1, а W - Ь. Тоді з раніше сформульованої гіпотези випливає, що Р відповідає е. Зауважимо, що в зашифрованому тексті буквосполучення ЪWР є, і тепер ми можемо представити його як the. В англійській мові the є найпоширенішою триграмою (тобто комбінацією з трьох літер), тому можна сподіватися, що ми рухаємося в правильному напрямку.

1175;

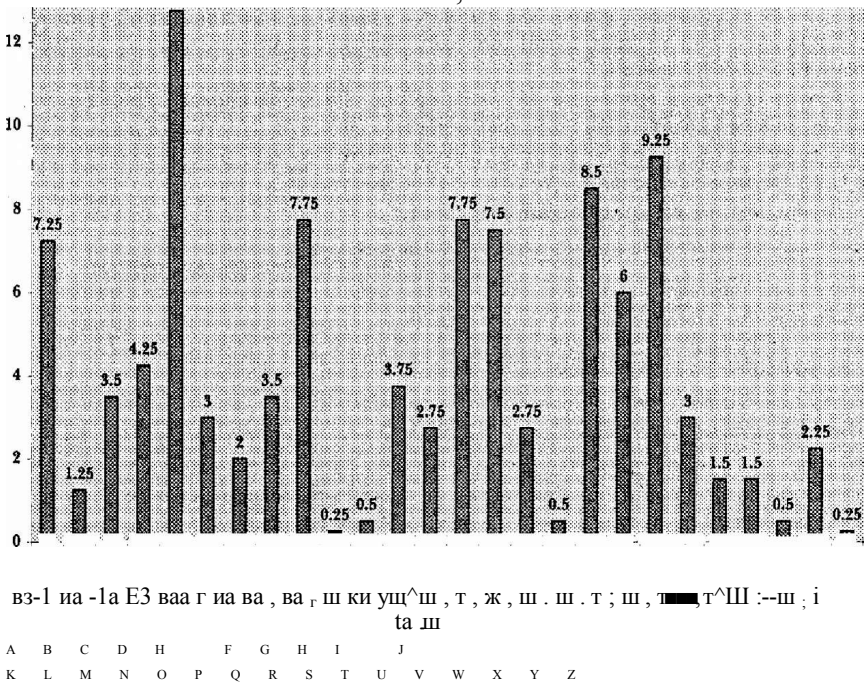



Рис. 5.2. Відносна частота появи букв в англійському тексті

Тепер зверніть увагу на комбінацію ZWSZ в першій строчці. Звичайно, ми не можемо сказати з повною упевненістю, що ці літери належать одному і тому ж слову, але, якщо припустити, що це так, вони відповідають слову





thSt. Звідси висновок, що букві S відповідає а. Тепер ми маємо наступний результат.

**UZQSOVUOHXMOPVGPOZPEVSGZWSZOPFPESXUD**
**BMETSXAIZ t a- e e te athat e e a -at**

**VUEPHZHMDZSHZOWSFPAPPDTSVPQUZWYMXUZ**
**UHSX**
**e ttatha e її a e th ta**

**EPYEPOPDZSZUFPOMBZWPFUPZHMDJUDTMOHM**
**Q**
**e e e tat з th з t**

З'ясувавши значення всього лише чотирьох букв, ми розшифрували вже значну частину повідомлення. Продовжуючи аналіз частоти появи літер, а також застосовуючи метод проб і помилок, залишається виконати зовсім небагато роботи, щоб отримати остаточну відповідь. Розшифрований вихідний текст (з доданими в нього пробілами) має такий вигляд.

**it was disclosed yesterday that several informal but direct contacts have been made with political representatives of the viet cong in moscow**

Моноалфавітні шифри легко розкриваються, так як вони успадковують частотність вживання літер оригінального алфавіту. Контрзаходом в даному випадку є застосування для однієї літери не одного, а декількох замінників (так званих омофонів).Наприклад, букві е вихідного тексту може відповідати кілька різних символів шифру (скажімо, 16, 74, 35 і 21), причому кожен такий символ може використовуватися або по черзі, або за випадковим законом. Якщо число символів-замінників, призначених букві, вибрати пропорційним частоті появи цієї букви, то підрахунок частотності вживання літер в зашифрованому тексті стає безглуздим. Великий математик Карл Фрідріх

Гаусс (Carl Friedrich Gauss) був впевнений, що з використанням омофонів він винайшов шифр, який неможливо зламати. Але навіть при вживанні омофонів кожному елементу відкритого тексту відповідає тільки один елемент шифрованого тексту, тому в останньому як і раніше повинні спостерігатися характерні показники частоти повторення комбінацій декількох букв (наприклад, біграм),і в результаті завдання криптоаналізу як і раніше залишається досить елементарним.

Щоб у тексті, зашифрованим з допомогою методів підстановки, структура вихідного тексту виявлялася менш помітно, можна використовувати два принципово різних підходи. Один з них полягає в заміщенні не окремих символів відкритого тексту, а комбінацій декількох символів, інший підхід передбачає використання для шифрування кількох алфавітів.

**5.2.1. Шифрування інверсними символами (щодо доповнення до 255)**

Даний метод шифрування є окремим випадком одноалфавітної заміни в алфавіті потужності 256 (двійково-вісімкового вектора). Суть методу полягає в заміні символу ASCII з номером і на символ з номером 255-і. Аналогічно проводиться і операція розшифрування.

**5.3. Багатоалфавітні методи**

Слабка криптостійкість моноалфавітної підстановки долається із застосуванням підстановки багатоалфавитної. Багатоалфавітне шифрування (Багатоалфавітна заміна) полягає в тому, що для послідовних символів шифруємого тексту використовуються одноалфавітні методи з різними ключами.

Наприклад, перший символ замінюється за методом Цезаря зі зміщенням 18, другий - зі зміщенням 12 і т.д. до кінця заданого ключа. Потім процедура триває періодично. Більш загальною є ситуація, коли виконується не шифр





Цезаря, а послідовність довільних підстановок, відповідних одноалфавітним методам.

У шифрах багатоалфавітної заміни для шифрування кожного символу перехідного повідомлення застосовується свій шифр простої заміни (свій алфавіт).

Таблиця 5.1

| | |
|---|---|
| | АБВГДЕЁЖЗИКЛМНОПРСТУФХЧШЩЪЫЬЭЮЯ_ |
| А | АБВГДЕЁЖЗИКЛМНОПРСТУФХЧШЩЪЫЬЭЮЯ_ |
| Б | _АБВГДЕЁЖЗИКЛМНОПРСТУФХЧШЩЪЫЬЭЮЗ |
| В | Я_АБВГДЕЁЖЗИКЛМНОПРСТУФХЧШЩЪЫЬЭЮ |
| Г | ЮЗ_АБВГДЕЁЖЗИКЛМНОПРСТУФХЧШЩЪЫЬЭ |
| . | |
| Я | ВГДЕЁЖЗИКЛМНОПРСТУФХЧШЩЪЫЬЭЮЯ_АБ |
| | БВГДЕЁЖЗИКЛМНОПРСТУФХЧШЩЪЫЬЭЮЯ_А |

Кожен рядок в табл. 5.1 відповідає одному шифру заміни аналогічно шифру Цезаря для алфавіту, доповненого пропуском. При шифруванні повідомлення його виписують в рядок, а під ним ключ. Якщо ключ виявився коротшим повідомлення, то його циклічно повторюють. Шифртекст отримують, знаходячи символ у колонці таблиці за буквою тексту і рядку, що відповідає букві ключа. Наприклад, використовуючи ключ АГАВА з повідомлення ПРИЇЖДЖАЮТЬ ШОСТОГО, отримуємо наступну шифровку:

Повідомлення ПРИЕЗЖАЮ_ШЕСТОГО
Ключ АГАВААГАВААГАВАА
Шифровка ПНИГЗЖЮЮЮАЕОТМГО

У комп'ютері така операція відповідає додаванню кодів ASCII символів повідомлення та ключа по модулю 256.

### 5.3.1. Шифр Плейфейєра

Одним з найбільш відомих шифрів, що базуються на методі багатобуквенного шифрування, є шифр Плейфейєра, в якому біграми відкритого тексту розглядаються як самостійні одиниці, перетворені в задані біграми шифрованого тексту.

Алгоритм Плейфейєра заснований на використанні матриці букв розмірності 5x5, створеної на основі деякого ключового слова. Давайте розглянемо приклад, рішення якого знаходить лорд Пітер Уімсі в романі Дороті Сейерс (Dorothy Sayers) "Have His Carcase":

| | | | | |
|---|---|---|---|---|
| M | O | N | A | R |
| C | H | Y | B | D |
| E | F | G | I/J | K |
| L | P | Q | S | Y |
| U | V | W | X | Z |

В даному випадку ключовим словом є monarchy (монархія). Матриця створюється шляхом розміщення букв, використаних у ключовому слові, зліва направо і зверху вниз (повторювані букви відкидаються). Потім літери алфавіту , що залишилися розміщуються в природному порядку в останніх рядках і стовпцях матриці. Літери I і J вважаються однією і тією ж буквою. Відкритий текст шифрується порціями по дві букви у відповідності зі слідуючими правилами.

1.Якщо виявляється, що повторюються букви відкритого тексту і утворюють одну пару для шифрування, то між цими літерами вставляється спеціальна літера-заповнювач,





наприклад x. Зокрема, таке слово як balloon буде переутворено до виду ba lx lo on.

2.Якщо букви відкритого тексту потрапляють в одну й ту ж строчку матриці, кожна з них замінюється буквою, наступної за нею в тому ж рядку справа - з тією умовою, що для заміни останнього елемента рядка матриці служить перший елемент того ж рядка. Наприклад, аг шифрується як RM.

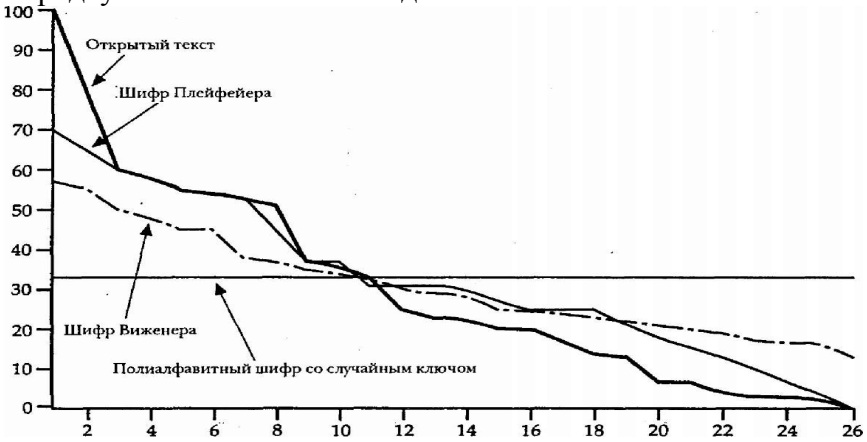
3.Якщо букви відкритого тексту потрапляють в один і той же стовпець матриці, кожна з них замінюється буквою, що стоїть в тому самому стовпці відразу під нею, з тією умовою, що для заміни самого нижнього елемента стовпця матриці береться самий верхній елемент того ж стовпця. Наприклад, mu шифрується як CM.

Якщо не виконується жодна з наведених вище умов, кожна буква з пари букв відкритого тексту замінюється буквою, що знаходиться на перетині рядка матриці і стовпця, в якому знаходиться друга буква відкритого тексту. Наприклад, hs шифрується як BP, а ea - як IM (або JM, за бажанням шифрувальника).

Шифр Плейфейєра значно надійніше простих моноалфавитних шифрів. З одного боку, букв всього 26, а біграмм - 26 x 26 = 676, і вже тому ідентифікувати біграми складніше, ніж окремі літери. З іншого боку, відносна частота появи окремих букв коливається набагато в більш широкому діапазоні, ніж частота появи біграм, тому аналіз частотності вживання біграм теж складніше піддається аналізу частотності вживання літер. З цих причин дуже довго вважалося, що шифр Плейфейєра зламати неможливо. Він служив стандартом шифрування в Британській армії під час Першої світової війни і нерідко застосовувався в армії США і союзних військах навіть в період Другої світової війни.

Незважаючи на таку високу репутацію в минулому, шифр Плейфейєра насправді розкрити відносно легко, так як зашифрований з його допомогою текст все одно зберігає багато статистичних характеристик відкритого тексту. Для злому цього шифру, як правило, досить мати шифрований текст, що складається з декількох сотень літер.

Один із способів оцінки ефективності шифру Плейфейєра та інших шифрів зображений на рис. 5.3. Лінія, позначена на малюнку як відкритий текст, відображає розподіл значень відносної частоти входження символів алфавіту в статті Encyclopaedia Brittanica, присвяченій криптології і містить більше 70 000 символів. Подібний графік характерний для розподілу відносної частоти появи символів і для будь-якого моноалфавітного шифру. Сам графік виходить таким чином. Число входжень букви в тексті ділиться на число появ в тексті символу «е» (самий часто використовуваний символ в англійській мові).У результаті «е» має відносну частоту 1, t - близько 0,76 і т.д. Поділу на горизонтальній осі відповідають буквам в порядку зниження значень відносної частоти їх появи.




Впорядковані за частотою появи літери





Рис. 5.3.Графік розподілу значень частоти для текстів, шифрованих за допомогою шифру Плейфейєра

На рис. 5.3 також показаний графік розподілу значень частоти для текстів, шифрованих за допомогою шифру Плейфейєра. З метою нормалізації числа появи в зашифрованому тексті тієї чи іншої літери поділялося на число входжень літери «е» в відкритому тексті. Отримані в результаті нормалізації графіки показують, наскільки частота розподілу букви (при використанні якої розкриття, наприклад, символів шифрів виявляється зовсім простою справою) маскується шифруванням. Якщо в процесі шифрування інформація про розподіл повністю ховається, графік для шифрованого за допомогою такого шифру тексту має бути горизонтальною прямою лінією, а криптоаналіз такого тексту з використанням лише шифрованого тексту, мабуть,повинен виявитися практично неможливим. Як видно з рис. 5.3, шифр Плейфейєра має більш пологий графік розподілу значень частоти в порівнянні з відкритим текстом, але все ж представляє для кріптоаналітика досить широкі можливості для статистичного аналізу збережених структур.

### 5.3.2. Шифр Хілла

Ще одним цікавим багатобуквенним шифром є шифр, розроблений математиком Лестером Хіллом (Lester Hill) в 1929 р. В його основі лежить алгоритм, який замінює кожні **m** послідовних літер відкритого тексту **m** літерами шифрованого тексту. Підстановка визначається **m** лінійними рівняннями, в яких кожному символу присвоюється числове значення (a = 0, b = 1, ..., z = 25). Наприклад, при **m** = 3, отримуємо наступну систему рівнянь:

**Ci = (AuP! + Kv* + krsPs) mod 26; $C_2$ = (^Pi + Д^Pa + ^Pз) mod 26; $C_3$ = (ДnPi + K2P2 + &13P3) mod 26.**

Цю систему рівнянь можна записати у вигляді добутку вектора і матриці у слідуючрму вигляді

$$\begin{pmatrix} C_1 \\ C_2 \\ C_3 \end{pmatrix} = \begin{pmatrix} k_{11} & k_{12} & k_{13} \\ k_{21} & k_{22} & k_{23} \\ k_{31} & k_{32} & k_{33} \end{pmatrix} \begin{pmatrix} p_1 \\ p_2 \\ p_3 \end{pmatrix}$$

Або **C=kp,** де **C** і **p –** вектори довжини 3, які представляють відповідно шифрований і відкритий текст, а **k-**це матриця розмірності 3x3, яка представляє ключ шифрування. Операції виконуються по модулю 26.

Розглянемо, наприклад, як буде шифрувати текст "раушогешопеу" при використанні ключа

$$\mathbf{K} = \begin{pmatrix} 17 & 17 & 5 \\ 21 & 18 & 21 \\ 2 & 2 & 19 \end{pmatrix}.$$

Перші три букви відкритого тексту представлені вектором (15 0 24). Таким чином, К (15 0 24) = (275 819 486) mod 26 = (11 13 18) = LNS. Продовжуючи обчислення, отримаємо для даного прикладу шифрований текст виду LNSHDLEWMTRW.

Для розшифровки потрібно скористатися матрицею, зворотньою **К**. Зворотньою по відношенню до матриці **К** називається така матриця $\mathbf{K^{-1}}$, для якої виконується рівність $\mathbf{KK^{-1}} = \mathbf{K^{-1}K} = \mathbf{I}$, де **I** - це одинична матриця (матриця, яка складається з нулів всюди, за винятком головної діагоналі, проходить з лівого верхнього кута в правий нижній,на якій передбачаються одиниці).Зворотна матриця існує не для всякої матриці, однак, коли зворотна матриця є, для неї





обов'язково виконується наведена вище рівеність. У нашому прикладі зворотною матрицею є матриця

$$\mathbf{K^{-1}} = \begin{pmatrix} 4 & 9 & 15 \\ 15 & 17 & 6 \\ 24 & 0 & 17 \end{pmatrix}.$$

Це перевіряється наступними обчисленнями:

$$\begin{pmatrix} 17 & 17 & 5 \\ 21 & 18 & 21 \\ 2 & 2 & 19 \end{pmatrix} \begin{pmatrix} 4 & 9 & 15 \\ 15 & 17 & 6 \\ 24 & 0 & 17 \end{pmatrix} = \begin{pmatrix} 443 & 442 & 442 \\ 858 & 495 & 780 \\ 494 & 52 & 365 \end{pmatrix} \bmod 26 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

Легко побачити, що в результаті застосування матриці К "до шифрованого тексту виходить відкритий текст. Щоб пояснити, як отримана зворотна матриця, нам доведеться зробити невеликий екскурс в лінійну алгебру - необхідні подробиці допитливий читач може знайти в будь-якому відповідному підручнику. Визначником квадратної матриці (**m**x**m**) називають суму таких всіляких добутків елементів матриці, що в добутку кожна колонка і кожен рядок представлені рівно одним елементом, причому деякі з цих добутків множаться на -1. Зокрема, для матриці 2x2 виду $\begin{pmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \end{pmatrix}$ визначник обчислюється за формулою $k_{11}k_{22} - k_{12}k_{21}$. Для матриці 3x3 значення визначника підраховується за формулою

$k_{11}k_{22}k_{33} + k_{21}k_{32}k_{13} + k_{31}k_{12}k_{23} - k_{31}k_{22}k_{13} - k_{21}k_{12}k_{33} - k_{11}k_{32}k_{23}$.

Якщо квадратна матриця А має відмінний від нуля визначник, то зворотна матриця обчислюється як $[\mathbf{A^{-1}}]_{ij}= (-1)^{\mathbf{i+j}}(D_{ij})/\det(A)$, де $(D_{ij})$ - визначник матриці, отриманої шляхом видалення **i**-ї строчки і **j**-го стовпця з матриці **A,** а **det(A)** - визначник самої матриці **A**. В нашому випадку всі ці обчислення проводяться за модулем 26.

У загальному вигляді систему Хілла можна записати в такій формі:

$$C = E_k(P) = KP,$$
$$P = D_k(C) = K^{-1}C = K^{-1}KP = P.$$

Як і у випадку шифру Плейфейєра, перевага шифру Хілла полягає в тому, що він повністю маскує частоту входження окремих букв. А для шифру Хілла чим більше розмір матриці в шифрі, тим більше в зашифрованому тексті ховається інформації про відмінності в значеннях частоти появи інших комбінацій символів. Так, шифр Хілла з матрицею 3х3 приховує частоту появи не тільки окремих букв, а й двобуквених комбінацій.

Хоча шифр Хілла стійкий до спроб криптоаналізу в тих випадках, коли відомий тільки шифрований текст, цей шифр легко розкрити при наявності відомого відкритого тексту. Розглянемо шифр Хілла з матрицею (**m**х**m**). Припустимо, що нам відомі **m** пар уривків відкритого і відповідно шифрованого текстів, кожен довжиною **m**. Позначимо такі пари $\mathbf{Pj} = (\mathbf{p_{1j}}, \mathbf{p_{2j}},\ldots,\mathbf{p_{mj}})$, і $\mathbf{C_j} = (\mathbf{C_{1j}}, \mathbf{C_{2j}}, \ldots, \mathbf{C_{mj}})$, щоб виконувалася умова **Cj = КPj**, для всіх $1 \le j \le m$ і деякої невідомої ключово] матриці **К**. Тепер визначимо дві такі матриці $\mathbf{X} = (\mathbf{p_{ij}})$ і $\mathbf{У} = (\mathbf{C_{ij}})$ розміру (**m**х**m**), що **У = ХК**. Тоді, за умови що для матриці **Х** існує зворотна матриця, **К** можна визначити за формулою $\mathbf{К} = \mathbf{X^{-1}У}$. Якщо ж отримати матрицю, зворотну матриці Х, неможливо, необхідно сформувати іншу матрицю **Х** з додатковими





парами відповідності відкритого і шифрованого текстів, до тих пір, поки не буде знайдена обернена матриця.

Припустимо, що відкритий текст "**friday**" зашифрований за допомогою шифру Хілла з використанням матриці 2х2, в результаті чого отримано шифрований текст **PQCFKU**. Таким чином, ми знаємо, що К (5 17) = (15 16), К (8 3) = (2 5) і К (0 24) = (10 20). Використовуючи перші дві пари символів відкритого і шифрованого тексту, отримуємо

$$\begin{pmatrix}15 & 16\\ 2 & 5\end{pmatrix} = \begin{pmatrix}5 & 17\\ 8 & 3\end{pmatrix}K.$$

Обчислимо матрицю, зворотну матриці Х:

$$\begin{pmatrix}5 & 17\\ 8 & 3\end{pmatrix}^{-1} = \begin{pmatrix}9 & 1\\ 2 & 15\end{pmatrix}.$$

Таким чином, тепер можна отримати значення ключа:

$$\mathbf{K}=\begin{pmatrix}9 & 1\\ 2 & 15\end{pmatrix}\cdot\begin{pmatrix}15 & 16\\ 2 & 5\end{pmatrix} = \begin{pmatrix}7 & 19\\ 8 & 3\end{pmatrix}.$$

Отриманий результат можна перевірити за допомогою решти пар відкритого і шифрованого текстів.

### 5.4. Поліалфавітні шифри

Інша можливість удосконалення простого моноалфавитного шифру полягає у використанні кількох моноалфавитних підстановок, що застосовуються в ході шифрування відкритого тексту в залежності від визначених умов. Сімейство шифрів, заснованих на застосуванні таких методів шифрування, називається поліалфавітними

шифрами. Подібні методи шифрування мають такі загальні властивості:

1. Використовується набір пов'язаних моноалфавітних підстановок.

2.Є певний ключ, за яким визначається, яке конкретне перетворення повинно застосовуватися для шифрування на даному етапі.

Самим широко відомим і водночас найпростішим алгоритмом такого роду є шифр Віженера (Vigenere). Цей шифр базується на наборі правил моноалфавитної підстановки, представлених 26 шифрами Цезаря із зсувом від 0 до 25. Кожен з таких шифрів можна позначити ключовою буквою, що є буквою шифрованого тексту, відповідної букві «а» відкритого тексту. Наприклад, шифр Цезаря, для якого зміщення дорівнює 3, позначається ключовою буквою «d». Для полегшення розуміння і застосування цієї схеми була запропонована матриця, названа "**табло Віженера"** (рис. 5.4 та 5.5). Всі 26 шифрів розташовуються по горизонталі, і кожному з шифрів відповідає своя ключова буква, представлена в крайньому стовпці зліва. Алфавіт, відповідний буквам відкритого тексту, знаходиться в першій зверху строчці таблиці. Процес шифрування простий - необхідно за ключовою буквою «х» і буквою відкритого тексту «у» знайти букву шифрованого тексту, яка знаходиться на перетині строчки «х» та стовпця «у». У даному випадку такою буквою є буква V.

Щоб зашифрувати повідомлення, потрібен ключ, який має ту ж довжину, як і саме повідомлення. Зазвичай ключ являє собою число, що повторюється потрібне число раз (ключове слово), щоб отримати строчку відповідної довжини. Наприклад, якщо ключовим словом є deceptive, повідомлення "we are discovered save yourself 'шифрується наступним чином.





Ключ: deceptive deceptive deceptive

відкритий текст: wearediscoveredsaveyourself

шифрований текст:

ZICVTWQNGRZGVTWAVZHCQYGLMGJ

Розшифрувати текст також просто - буква ключа визначає строчку, буква шифрованого тексту, що знаходиться в цій строчці, визначає стовпець, і в ньому у першій строчці таблиці буде перебувати відповідна літера відкритого тексту.

| | a | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t | u | v | w | x | y | z |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| a | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
| b | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A |
| c | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B |
| d | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C |
| e | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D |
| f | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E |
| g | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F |
| h | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G |
| i | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H |
| j | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I |
| k | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J |
| l | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K |
| m | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L |
| n | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M |
| o | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N |
| p | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O |
| q | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P |
| r | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q |
| s | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R |
| t | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S |
| u | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T |
| v | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U |
| w | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V |
| x | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W |
| y | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X |
| z | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y |

Рис. 5.4. Сучасне табло Вінженера для латинського алфавіту

Розглянемо ще один приклад одержання шифрованого тексту за допомогою таблиці Віженера. Нехай вибрано ключове слово АМБРОЗИЯ. Необхідно зашифрувати повідомлення ПРИЛЕТАЮ СЕДЬМОГО.

| Ключ | а | б | в | г | д | е | ж | з | и | й | к | л | м | н | о | п | р | с | т | у | ф | х | ц | ч | ш | щ | ь | ъ | э | ю | я |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | а | б | в | г | д | е | ж | з | и | й | к | л | м | н | о | п | р | с | т | у | ф | х | ц | ч | ш | щ | ь | ы | ъ | э | ю | я |
| 1 | б | в | г | д | е | ж | з | и | й | к | л | м | н | о | п | р | с | т | у | ф | х | ц | ч | ш | щ | ь | ы | ъ | э | ю | я | а |
| 2 | в | г | д | е | ж | з | и | й | к | л | м | н | о | п | р | с | т | у | ф | х | ц | ч | ш | щ | ь | ы | ъ | э | ю | я | а | б |
| 3 | г | д | е | ж | з | и | й | к | л | м | н | о | п | р | с | т | у | ф | х | ц | ч | ш | щ | ь | ы | ъ | э | ю | я | а | б | в |
| 4 | д | е | ж | з | и | й | к | л | м | н | о | п | р | с | т | у | ф | х | ц | ч | ш | щ | ь | ы | ъ | э | ю | я | а | б | в | г |
| 5 | е | ж | з | и | й | к | л | м | н | о | п | р | с | т | у | ф | х | ц | ч | ш | щ | ь | ы | ъ | э | ю | я | а | б | в | г | д |
| 6 | ж | з | и | й | к | л | м | н | о | п | р | с | т | у | ф | х | ц | ч | ш | щ | ь | ы | ъ | э | ю | я | а | б | в | г | д | е |
| 7 | з | и | й | к | л | м | н | о | п | р | с | т | у | ф | х | ц | ч | ш | щ | ь | ы | ъ | э | ю | я | а | б | в | г | д | е | ж |
| 8 | и | й | к | л | м | н | о | п | р | с | т | у | ф | х | ц | ч | ш | щ | ь | ы | ъ | э | ю | я | а | б | в | г | д | е | ж | з |
| 9 | й | к | л | м | н | о | п | р | с | т | у | ф | х | ц | ч | ш | щ | ь | ы | ъ | э | ю | я | а | б | в | г | д | е | ж | з | и |
| 10 | к | л | м | н | о | п | р | с | т | у | ф | х | ц | ч | ш | щ | ь | ы | ъ | э | ю | я | а | б | в | г | д | е | ж | з | и | й |
| 11 | л | м | н | о | п | р | с | т | у | ф | х | ц | ч | ш | щ | ь | ы | ъ | э | ю | я | а | б | в | г | д | е | ж | з | и | й | к |
| 12 | м | н | о | п | р | с | т | у | ф | х | ц | ч | ш | щ | ь | ы | ъ | э | ю | я | а | б | в | г | д | е | ж | з | и | й | к | л |
| 13 | н | о | п | р | с | т | у | ф | х | ц | ч | ш | щ | ь | ы | ъ | э | ю | я | а | б | в | г | д | е | ж | з | и | й | к | л | м |
| 14 | о | п | р | с | т | у | ф | х | ц | ч | ш | щ | ь | ы | ъ | э | ю | я | а | б | в | г | д | е | ж | з | и | й | к | л | м | н |
| 15 | п | р | с | т | у | ф | х | ц | ч | ш | щ | ь | ы | ъ | э | ю | я | а | б | в | г | д | е | ж | з | и | й | к | л | м | н | о |
| 16 | р | с | т | у | ф | х | ц | ч | ш | щ | ь | ы | ъ | э | ю | я | а | б | в | г | д | е | ж | з | и | й | к | л | м | н | о | п |
| 17 | с | т | у | ф | х | ц | ч | ш | щ | ь | ы | ъ | э | ю | я | а | б | в | г | д | е | ж | з | и | й | к | л | м | н | о | п | р |
| 18 | т | у | ф | х | ц | ч | ш | щ | ь | ы | ъ | э | ю | я | а | б | в | г | д | е | ж | з | и | й | к | л | м | н | о | п | р | с |
| 19 | у | ф | х | ц | ч | ш | щ | ь | ы | ъ | э | ю | я | а | б | в | г | д | е | ж | з | и | й | к | л | м | н | о | п | р | с | т |
| 20 | ф | х | ц | ч | ш | щ | ь | ы | ъ | э | ю | я | а | б | в | г | д | е | ж | з | и | й | к | л | м | н | о | п | р | с | т | у |
| 21 | х | ц | ч | ш | щ | ь | ы | ъ | э | ю | я | а | б | в | г | д | е | ж | з | и | й | к | л | м | н | о | п | р | с | т | у | ф |





| 22 | ц | ч | ш | щ | ь | ы | ъ | э | ю | я | а | б | в | г | д | е | ж | з | и | й | к | л | м | н | о | п | р | с | т | у | ф | х |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 23 | ч | ш | щ | ь | ы | ъ | э | ю | я | а | б | в | г | д | е | ж | з | и | й | к | л | м | н | о | п | р | с | т | у | ф | х | ц |
| 24 | ш | щ | ь | ы | ъ | э | ю | я | а | б | в | г | д | е | ж | з | и | й | к | л | м | н | о | п | р | с | т | у | ф | х | ц | ч |
| 25 | щ | ь | ы | ъ | э | ю | я | а | б | в | г | д | е | ж | з | и | й | к | л | м | н | о | п | р | с | т | у | ф | х | ц | ч | ш |
| 26 | ь | ы | ъ | э | ю | я | а | б | в | г | д | е | ж | з | и | й | к | л | м | н | о | п | р | с | т | у | ф | х | ц | ч | ш | щ |
| 27 | ы | ъ | э | ю | я | а | б | в | г | д | е | ж | з | и | й | к | л | м | н | о | п | р | с | т | у | ф | х | ц | ч | ш | щ | ь |
| 28 | ъ | э | ю | я | а | б | в | г | д | е | ж | з | и | й | к | л | м | н | о | п | р | с | т | у | ф | х | ц | ч | ш | щ | ь | ы |
| 29 | э | ю | я | а | б | в | г | д | е | ж | з | и | й | к | л | м | н | о | п | р | с | т | у | ф | х | ц | ч | ш | щ | ь | ы | ъ |
| 30 | ю | я | а | б | в | г | д | е | ж | з | и | й | к | л | м | н | о | п | р | с | т | у | ф | х | ц | ч | ш | щ | ь | ы | ъ | э |
| 31 | я | а | б | в | г | д | е | ж | з | и | й | к | л | м | н | о | п | р | с | т | у | ф | х | ц | ч | ш | щ | ь | ы | ъ | э | ю |

Рис. 5.5. Таблиця Віженер для російського алфавіту

Випишемо вихідне повідомлення в рядок і запишемо під ним ключове слово з повторенням. У третій рядок (строчку) будемо виписувати літери шифртексту, що визначаються з таблиці Віженера.

| Повідомлення | П | Р | и | Л | Е | Т | А | ю | С | Е | д | Ь | М | О | г | О |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Ключ | А | м | Б | Р | О | З | и | З | А | М | Б | Р | О | З | и | З |
| Шифртекст | П | ъ | Й | ы | У | щ | и | э | С | С | Е | К | Ь | Х | Л | Н |

Перевага цього шифру полягає в тому, що для представлення однієї і тієї ж букви відкритого тексту в зашифрованому тексті є багато різних варіантів - по одному на кожну з неповторюваних букв ключового слова. Таким чином, ховається інформація, що характеризує частотність вживання літер. Але і за допомогою цього методу все ж не вдається повністю сховати вплив структури відкритого тексту на структуру шифрованого. Наприклад, на рис. 5.3 показаний графік розподілу значень частоти для шифру Віженера при довжині ключового слова 9 символів. У наявності явна перевага в порівнянні з

шифром Плейфейєра, але очевидно і те, що повністю інформацію про розподіл частоти замаскувати не вдається. Не завадить хоча б коротенько розглянути метод злому цього шифру, так як на прикладі цього методу можна показати деякі з математичних принципів, що лежать в основі більшості сучасних методів криптоаналізу.

Перш за все, припустимо, що противник впевнений в тому, що шифруваний текст був отриманий або за допомогою моноалфавітної підстановки, або за допомогою шифру Віженера. Щоб з'ясувати, який саме з цих двох методів був використаний, можна провести простий тест. Якщо використовувалася моноалфавітная підстановка, статистичні показники шифрованого тексту не будуть відрізнятися від відповідних показників мови, якою написаний відкритий текст. Так, відповідно до рис. 5.2, в цьому випадку в зашифрованому тексті один символ повинен зустрічатися в 12,75% випадків, інший - у 9,25% і т.д. Якщо для аналізу є лише одне повідомлення, точного збігу статистичних показників можна і не отримати. Але якщо статистика досить точно повторює статистику звичайного відкритого тексту, можна припустити, що використовувалася моноалфавітна підстановка.

Якщо ж, навпаки, все вказує на те, що був застосований шифр Віженера, то, як ми побачимо трохи пізніше, успіх подальшого аналізу тексту залежить від того, чи вдасться визначити довжину ключового слова. Поки давайте зконцентруємося на тому, як визначити довжину ключового слова. Вирішення цієї задачі засновано на наступній особливості даного шифру: якщо початкові символи двох однакових послідовностей відкритого тексту знаходяться один від одного на відстані, кратній довжині ключа, ці послідовності будуть представлені однаковими послідовностями і в зашифрованому тексті. У розглянутому прикладі є дві послідовності "red" і початок





другої з них виявляється на дев'ять символів далі щодо початку першої. Отже, в обох випадках «r» буде шифруватся з використанням ключової букви «e», «e» - за допомогою ключової букви «p», а «d» - за допомогою ключової літери. Таким чином, в обох випадках для шифрованого тексту буде отримана послідовність VTW. Аналітик, що має в своєму розпорядженні тільки шифрований текст, виявить повторювану послідовність VTW зі зміщенням в дев'ять символів, і тому може припустити, що довжина ключового слова дорівнює трьом, або дев'яти. Звичайно, для повторів всього два рази послідовності VTW збіг може виявитися і випадковим, а тому й невідповідність шифрованих з однаковими ключовими літерами однакових фрагментів відкритого тексту, але якщо повідомлення буде досить довгим, то таких повторюваних послідовностей в ньому буде чимало. Визначивши загальний множник для зміщення початку таких послідовностей, аналітик отримає достатньо надійну основу для припущень про довжину ключового слова.

Подальший аналіз базується на іншій особливості даного шифру. Якщо ключове слово має довжину N, то шифр, по суті, складається з N моноалфавітних символів шифрів. Наприклад, при використанні ключового слова «deceptive» літери, що знаходяться на 1-й, 10-й, 19-й і т.д. позиціях, шифруються одним і тим же моноалфавитним шифром. Це дає можливість використання відомих характеристик частотних розподілів букв відкритого тексту для злому кожного моноалфавитного шифру окремо.

Періодичності в ключовому рядку можна уникнути, використовуючи для ключового рядка неперіодичні послідовність тієї ж довжини, що й саме повідомлення. Віженер запропонував підхід, який отримав назву системи з автоматичним вибором ключа, коли послідовність ключового рядка отримується в результаті конкатенації

ключового слова з самим відкритим текстом. Для розглянутого прикладу ми отримаємо наступне.
відкритий текст: **wearediscoveredsaveyourself**
шифрований текст:
**ZICVTWQNGKZEIIGASXSTSLVVWLA**

Однак і ця схема виявляється вразливою. Оскільки і в ключовому рядку, і у відкритому тексті значення частоти розподілу букв будуть однакові, статистичні методи можна застосувати і в даному випадку. Наприклад, літера «е», шифрована за допомогою ключа е, повинна зустрічатися з частотою (+0,1275) 2 = 0,0163, тоді як «d», шифрована за допомогою «t», може зустрітися з частотою, приблизно в два рази меншою. Саме такі закономірності дозволяють добитися успіху при аналізі шифрованого тексту.
Кращим захистом від подібних методів криптоаналізу є вибір ключового слова, по довжині рівного довжині відкритого тексту, але відмінного від відкритого тексту за статистичними показниками. Така система була запропонована інженером компанії AT&T Гілбертом Вернамом (Gilbert Vernam) в 1918 р. Його система оперує не буквами, а двійковими числами. Коротко її можна виразити формулою

**Ci = Pi ® ki,**

де **pi** - і-та двійкова цифра відкритого тексту**; k** - і-а двійкова цифра ключа; Ci - і-а двійкова цифра шифрованого тексту; ® - операція XOR (що виключає "АБО").
Таким чином, шифрований текст генерується шляхом побітового виконання операції XOR для відкритого тексту і ключа. Завдяки властивостям цієї операції для розшифровки досить виконати подібну операцію:

**Pi = Ci ® ki.**

Суттю цієї технології є спосіб вибору ключа. Г.Вернам запропонував використовувати закільцьовану стрічку, що





означає циклічне повторення ключового слова, так щоб його система насправді передбачала роботу хоч і з дуже довгим, але все ж повторюємим ключом. Незважаючи на те що така схема в силу дуже великої довжини ключа значно ускладнює завдання криптоаналізу, схему, тим не менш, можна зламати, маючи в розпорядженні достатньо довгий фрагмент шифрованого тексту, відомі або ймовірно відомі фрагменти відкритого тексту або і те,і інше відразу.
Офіцер армійського корпусу зв'язку Джозеф Моборн (Joseph Mauborgne) запропонував такі поліпшення схеми шифрування Вернама, які зробили цю схему виключно надійною. Моборн запропонував відмовитися від повторень, а випадковим чином генерувати ключ, за довжиною рівній довжині повідомлення. Така схема, що отримала назву стрічки одноразового використання (або схеми з одноразовим блокнотом), злому не піддається. У результаті її застосування на виході виходить випадкова послідовність, яка не має статистичного взаємозв'язку з відкритиом текстом. Оскільки в цьому випадку шифрований текст не дає ніякої інформації про відкритий текст і немає способу і зламати код.
Складність практичного застосування цього методу полягає в тому, що відправник та одержувач повинні розташовувати одним і тим же випадковим ключом і мати можливість захистити його від сторонніх. Тому, незважаючи на всі переваги шифру Вернама перед іншими шифрами, на практиці до нього вдаються рідко.

## 5.5. Шифр "подвійний квадрат" Уїтстона

У 1854 р. англієць Чарльз Уїтстон розробив новий метод шіфрування біграмами, який називають "подвійним квадратом". Свою назву цей шифр отримав за аналогією з полібіанським квадратом. Шифр Уитстона відкрив новий

етап в історії розвитку криптографії. На відміну від полібіанського шифр "подвійний квадрат" використовує відразу дві таблиці, розміщені по одній горизонталі, а шифрування йде біграмами, як в шифрі Плейфейєра. Ці не настільки складні модифікації привели до появи на світ якісно нової криптографічної системи ручного шифрування. Шифр "подвійний квадрат" виявився дуже надійним і зручним і застосовувався Німеччиною навіть в роки Другої світової війни.

Пояснимо процедуру шифрування цим шифром на прикладі. Нехай маємо дві таблиці з випадково розташованими в них російськими алфавітами (рис. 5.6). Перед шифруванням вихідне повідомлення розбивають на біграми. Кожна біграма шифрується окремо. Першу букву біграм знаходять в лівій таблиці, а другу літеру - у правій таблиці. Потім подумки будують прямокутник так, щоб букви біграми лежали в його протилежних вершинах. Інші дві вершини цього прямокутника дають літери біграми шифртексту.

| | | | | |
|---|---|---|---|---|
| | Щ | Н | Ю | Р |
| И | Т | ь | Ц | Б |
| З | м | Е | . | С |
| В | ы | П | ч | |
| | Д | У | О | К |
| З | Э | Ф | Г | Ш |
| Х | А | , | Л | ъ |

| | | | | |
|---|---|---|---|---|
| И | ч | Г | З | Т |
| , | Ж | ь | м | О |
| З | Ю | Р | В | Щ |
| Ц | | П | Е | Л |
| ъ | А | Н | . | Х |
| Э | К | С | Ш | Д |
| Б | Ф | У | ы | |

Рис. 5.6.Дві таблиці з випадково розташованими символами російського алфавіту для шифру "подвійний квадрат"





Припустимо, що шифрується біграма вихідного тексту ИЛ. Буква И знаходиться в стовпці 1 і рядку 2 лівої таблиці. Буква Л знаходиться в стовпці 5 і рядку 4 правої таблиці. Це означає, що прямокутник утворений рядками 2 і 4, а також стовпчиками 1 лівої таблиці та 5 правої таблиці. Отже, в біг раму шифртексту входить буква О, розташована у стовпці 5 і рядку 2 правої таблиці, і буква В, розташована у стовпці 1 і рядку 4 лівої таблиці, тобто отримуємо біграму шифртексту ОВ.

Якщо обидві літери біграми повідомлення лежать в одному рядку, то і букви шифртексту беруть з цього ж рядка. Першу букву біграми шифртексту беруть з лівої таблиці в стовпці, відповідно другої літери біграми повідомлення. Друга ж буква біграми шифртексту береться з правої таблиці в стовпці, відповідному першій букві біграми повідомлення. Тому біграма повідомлення ТО перетворюється на біграму шифртекста ЖБ. Аналогічним чином шифруються всі біграми повідомлення:
Повідомлення ПР ИЛ ЕТ АЮ _Ш ЕС ТО ГО Шифртекст ПЕ ОВ ЩН ФМ ЕШ РФ БЖ ДЦ

Шифрування методом "подвійного квадрату" дає дуже стійкий до розкриття і простий у застосуванні шифр. Злом шифртексту "подвійного квадрату" вимагає великих зусиль,при цьому довжина повідомлення повинна бути не менше тридцяти рядків.

### 5.6. Застосування перестановок

Всі розглянуті вище методи грунтувалися на заміщенні символів відкритого тексту різними символами шифрованого тексту. Принципово інший клас перетворень будується на використанні перестановок букв відкритого тексту. Шифри, створені за допомогою перестановок, називають **перестановочними шифрами**.

Простійший з таких шифрів використовує перетворення "драбинки", заключається в тому, що відкритий текст записується уздовж похилих рядків певної довжини ("сходинок"), а потім зчитується порядково по горизонталі. Наприклад, щоб шифрувати повідомлення "meet me after the toga party" за методом драбинки зі сходами довжиною 2, запишемо це повідомлення як

Шифроване повідомлення буде мати такий вигляд:

MEMATRHTGPRYETEFETEOAAT

Такий "шифр" особливої складності для криптоаналізу не представляє. Більш складна схема передбачає запис тексту повідомлення в горизонтальні рядки однакової довжини і подальше зчитування тексту стовпець за стовпцем, але не по порядку, а відповідно до деякої перестановки стовпців. Порядок зчитування стовпців при цьому стає ключом алгоритму. Розглянемо наступний приклад.

Зашифрований текст:

TTNAAPTMTSOOAODWCOIXKNLYPETZ

Простий перестановний шрифт дуже легко розпізнати, оскільки літери в ньому зустрічаються з тією ж частотою, що і у відкритому тексті. Наприклад, для щойно розглянутого способу шифрування з перестановкою стовпців, аналіз шифру виконати досить просто - необхідно записати шифруваний текст у вигляді матриці і перебрати можливі варіанти перестановок для стовпців. Можна використовувати також таблиці значень частоти біграм і триграм.

Перестановний шифр можна зробити істотно більш захищеним, виконавши шифрування з використанням перестановок кілька разів. В цьому випадку застосовану для шифрування перестановку створити вже не так просто. Наприклад, якщо попереднє повідомлення шифрувати ще раз за допомогою того ж самого алгоритму, то результат буде наступним.





4 3 1 2 5 6 7 t

Ключ: t п а а о р t т і

Відкритий текст: з и о а ociVI

с о і h k п 1 у

р е t ъ

Зашифрований текст:

ИЗСУАиОРТТІ^ТМОНАОІЕРАХТТОКг

Щоб наочніше уявити те, що ми отримаємо в результаті повторного застосування перестановки, можна порівняти кожну букву вихідного відкритого тексту з номером відповідної їй позиції. Наше повідомлення складається з 28 літер, і вихідною послідовністю буде послідовність

01 02 03 04 05 06 07 08 09 10 11 12 13 14

15 16 17 18 19 20 21 22 23 24 25 26 27 28

Після першої перестановки отримаємо послідовність, яка все ще зберігає деяку регулярність структури.

03 10 17 24 04 11 18 25 02 09 16 23 01 08

15 22 05 12 19 26 06 13 20 27 07 14 21 28

Після другої перестановки виходить наступна послідовність.

17 09 05 27 24 16 12 07 10 02 22 20 03 25

15 13 04 23 19 14 11 01 26 21 18 08 06 28

Регулярність цієї послідовності вже зовсім не проглядається, тому її криптоаналіз вимагатиме значно більше зусиль.

### 5.6.1. Застосування магічних квадратів

У середні віки для шифрування перестановкою застосовувалися і магічні квадрати.
Магічними квадратами називають квадратні таблиці з вписаними у їх клітини послідовними натуральними числами, починаючи від 1, які дають в сумі по кожному стовпцю, кожному рядку і кожній діагоналі одне і те ж число.
Шифруємий текст вписували в магічні квадрати відповідно до нумерації їх клітин. Якщо потім виписати вміст такої таблиці по рядкам, то вийде шифртекст, сформований завдяки перестановці літер вихідного повідомлення. У ті часи вважалося, що створені за допомогою магічних квадратів шифртексти охороняє не тільки ключ, а й магічна сила.
Приклад магічного квадрату і його заповнення повідомленням: ПРИЛЕТАЮ ВОСЬМОГО зображений на рис. 5.7.

| | | | |
|---|---|---|---|
| **16** | **3** | **2** | **13** |
| **5** | **10** | **11** | **8** |
| **9** | **6** | **7** | **12** |
| **4** | **15** | **14** | **1** |

| | | | |
|---|---|---|---|
| **О** | **И** | **Р** | **М** |
| **Е** | **О** | **С** | **Ю** |
| **В** | **Т** | **А** | **Ь** |
| **Л** | **Г** | **О** | **П** |

Рис.5.7. Приклад магічного квадрату 4х4 і його заповнення
повідомленням ПРИЛЕТАЮ ВОСЬМОГО

Шифртекст, одержуваний при зчитуванні вмісту правої таблиці по рядках, має цілком загадковий вигляд:
ОИРМ ЕОСЮ ВТАЬ ЛГОП





Число магічних квадратів швидко зростає зі збільшенням розміру квадрата. Існує тільки один магічний квадрат розміром 3х3 (якщо не враховувати його повороти). Кількість магічних квадратів 4х4 складає вже 880, а кількість магічних квадратів 5х5 - близько 250 000. Магічні квадрати середніх і великих розмірів могли служити ДОБРОЮ базою для забезпечення потреб шифрування того часу, оскільки практично нереально виконати вручну перебір всіх варіантів для такого шифру.

### 5.7. Метод гамування

Процес зашифрування полягає в генерації гами шифру і накладення цієї гами на вихідний відкритий текст.
Власне процедура накладення може здійснюватися одним з двох способів:
1.Символи закриваємого тексту і гами замінюються цифровими еквівалентами, а потім складаються по модулю К, де К - кількість символів алфавіту, тобто

$$\mathbf{Тш = (Тв + Тг)\ mod К,} \qquad \mathbf{(5.7.1)}$$

де **Тш** - закриваємий; **Тв** - відкритий текст; **Тг** - гама.
2.Символи тексту і гами представляються в двійкових кодах, а потім кожна пара двійкових розрядів складається по модулю 2. Додавання по модулю 2 може бути розширено до перетворення за правилом логічної еквівалентності або логічної нееквівалентності. Неважко бачити, що таке розширення рівносильно введенню ще одного ключа, а саме - правил заміни.
Стійкість закриття способом гамування визначається, головним чином, якістю гами,яка визначається двома характеристиками: довжиною періоду і випадковістю розподілу по періоду.

Довжиною періоду гами називається мінімальна кількість символів, після якого послідовність починає повторюватися. Випадковість разподілу символів по періоду означає відсутність закономірностей між появою різних символів в межах періоду.

По довжині періоду розрізняються гами з кінцевим і нескінченним періодом. Кінцеві гами, в свою чергу, можуть бути розділені на короткі і довгі, хоча цей розподіл є, певною мірою, умовний.

При добрій якості гами за характеристикою випадковості стійкість закриття визначається виключно завдовжки її періоду. При цьому якщо довжина періоду гами перевищує довжину закриває мого тексту, то таке перетворення теоретично є абсолютно стійким, тобто його не можна розкрити на основі статистичної обробки закритого тексту. Проте теоретична неможливість розкриття не означає, що розкриття взагалі неможливе; при наявності деякої додаткової інформації відкритий текст може бути цілком або повністю розгаданий навіть при нескінченній гамі.

Як гами може бути використана будь-яка послідовність випадкових символів: наприклад, послідовність цифр основи натуральних логарифмів числа е, числа Рі і т.і. Якщо ж закриття здійснюється на ЕОМ, то такі послідовності можна генерувати за допомогою датчика псевдовипадкових чисел (ПСЧ). До нашого часу відомо кілька таких датчиків, які забезпечують задовільну якість гами.

Якщо, наприклад, в якості датчика (ПСЧ) вжити послідовність

$$X_{(j+1)} = aX_{(i)} + C(\text{mod } m) \quad , \qquad (5.7.2)$$

де $m = 2^k$ $(k \geq 2 - \textit{ціле})$, то найбільший період буде тоді, коли С- непарна і $a * \text{mod}\, 4 = 1$.

Більш складним є датчик





$$X_{(j+1)} = X_{(i)} + X_{(i-L)} (\text{mod } m) \quad , \qquad (5.7.3)$$

де **m = 2 і b = 35** початкова послідовність $X_1, X_2, \ldots, X_{(L)}$. генерується по датчику (5.7.2).

Для початку роботи задається $X_0$, що містить 6 цифр. За допомогою цього числа визначається **L** як сума числа 16 і числа, що визначається шістьма бітами $X_0$. Визначивши **L**, по датчику (5.7.2) генерується послідовність $X_1, X_2, \ldots, X_{(L)}$. -після чого по датчику (5.7.3) генерується послідовність з N чисел гами, причому N визначається як сума числа 2^k і числа, визначеного **K** молодшими бітами числа $X_L$..

При цьому якщо потрібна висока надійність закриття, то використовується значення К = 18, в інших К = 12.

Після того, як буде отримано число $X_{(L+n)}$, процедура починається спочатку , але замість числа $X_0$ використовується число $X_L$.

Відомі й більш складні датчики ПСЧ.

## Список літератури

1. Анин Б.Ю. Защита компьютерной информации. - СПб., 2000. Ахо А., Хопкрофт Дж., Ульман Дж. Построение и анализ вычислительных алгоритмов. - М.: Мир, 1979.

2. Вербицкий О.В. Вступление к криптологии. - Львов: Изд-во научно-технической литературы, 1998.

3. Домашев А.В., Грунтович М.М., Попов В.О. Программирование алгоритмов защиты информации. - М.: Изд-во "Нолидж", 2002.

4. Донцов Д. Как защитить компьютер от ошибок, вирусов, хакеров. СПБ.:Питер.2007.-144 с.

5. Кнут Д. Искусство программирования для ЭВМ. Т. 2. Получисленные алгоритмы. - М.: Мир, 1977.

6. Крис Вебер, Гэри Бадур. Безопасность в Windows ®XP. Готовые решения сложных задач защиты компьютеров ISBN5-93772-102-0, 464 с

7. Кузьминов В.И. Криптографические методы защиты информации. - Новосибирск: Высшая школа, 1998.

8. Маховенко Е.Б. Математические основы криптографии. - СПб.: Изд-во СПБГТУ, 1999.

9. Молдовян А.А., Молдовян Н.А., Советов Б.З. Криптография. - СПб.: Лань, 2000.

10. Нечаев В.И. Элементы криптографии. Основы защиты информации. - М.: Высшая школа, 1999.

11. Ростовцев А.Г. Алгебраические основы криптографии. - СПБ.: Мир и Семья, Интерлайн, 2000.

12. Соколов А.В., Шаньгин В.Ф. Защита информации в распределенных корпоративных сетях и системах. - М.: ДМК Пресс, 2002.

13. Столлингс Вильям. Криптографическая защита сетей. - М.: Издательский дом "Вильямс", 2001.

14. Теоретические основы компьютерной безопасности: учебное пособие для вузов / П.Н.Деревянин, О.О.Михальский, Д.И. Правиков. - М.: Радио и связь, 2000.

15. Фомичев В.М. Симметричные криптосистемы. Краткий обзор основ криптологии для шифросистем с секретным ключом. - М.: Изд-во МИФИ, 1995.





16. Чмора А.Л. Современная прикладная криптография. - М.: Гелиос АРВ, 2001.

17. Шаханова, М.В.Современные технологии информационной безопасности: учеб. пособие / М.В .Шаханова; Дальневосточный государственный технический университет. - Владивосток: Изд-во ДВГТУ, 2007. - 217 с.

18. Шнайер Брюс. Прикладная криптография. - М.: АВР, 1995.

19. Koblitz N. A Course in Number Theory and Cryptography. 2nd edition. New York: Springer-Vedag, 1994; Коблиц Н. Курс теории чисел и криптографии: Пер. с англ. - М.: ТВП, 2001.

20. Menezes A., van Oorschot P., Vanstone S. Handbook of Applied Cryptography. CRC Press Series on Discrete Mathematics and Its Applications, 1997.

**Навчальне видання**

# СУЧАСНІ ТЕХНОЛОГІЇ ІНФОРМАЦІЙНОЇ БЕЗПЕКИ

Частина 1

Навчальний посібник

Літнарович Руслан Миколайович

*Комп'ютерний набір, верстка – дизайн у редакторі Microsoft® Office 2003® Р.М.Літнарович*

***Міжнародний економіко-гуманітарний університет ім. академіка С. Дем'янчука***

**Кафедра математичного моделювання**
**33027,м.Рівне,Україна**
**Вул.акад. С.Дем'янчука,4, корпус 1**
**Телефон:(+00380) 362 23-73-09**
**Факс:(+00380) 362 23-01-86**
**E-mail:mail@regi.rovno.ua**