МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
РІВНЕНСЬКИЙ ДЕРЖАВНИЙ ГУМАНІТАРНИЙ УНІВЕРСИТЕТ
КАФЕДРА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА МОДЕЛЮВАННЯ

Кіндрат П.В.

З А Х И С Т І Н Ф О Р М А Ц І Ї

конспект лекцій

Рівне 2025

Затверджено на засіданні кафедри інформаційних технологій та моделювання
Протокол від «29» квітня 2025 року № 5

Схвалено навчально-методичною комісією факультету математики та інформатики
Протокол від «30» квітня 2025 року № 4

Рецензент:
Бойчура М.В. к.т.н., доцент.

Кіндрат П.В. Захист інформації: конспект лекцій [Електронний ресурс] Рівненський державний гуманітарний університет. Рівне. 2025. 146 с.

Конспект лекцій покликаний сприяти розширенню знань студентів у сфері інформаційної безпеки та криптографічного захисту інформації. Він розрахований для студентів спеціальностей 121 «Інженерія програмного забезпечення» та 122 «Комп'ютерні науки». Розділи конспекту зосереджують увагу на питаннях визначення ключових напрямків реалізації загроз безпеці інформації, та окреслюють напрямки їх вирішення. Особливоо акцентується увага на принципах реалізації та впровадження систем криптографічного захисту інформації та сучасних проблем цього напрямку.

ЗМІСТ

РОЗДІЛ 1
ОСНОВИ ІНФОРМАЦІЙНОЇ БЕЗПЕКИ ТА ЗАХИСТУ ІНФОРМАЦІЇ

1 Зміст захисту інформації в комп'ютерних системах

1.1 Поняття захисту інформації

Питання інформаційної безпеки займають особливе місце в зв'язку із зростаючою роллю в житті суспільства і вимагають до себе все більше уваги. Як відомо, всі виробничі процеси мають в своєму складі матеріальну і нематеріальну складові. Перша – це необхідне для виробництва устаткування, матеріали, енергія і т.д. Друга – технологія виробництва. В останні роки з'явилось багато галузей виробництва, які майже на 100% складаються із однієї інформації. Наприклад, дизайн, створення програмного забезпечення, реклама та інше. Із підвищенням важливості та цінності інформації відповідно росте і роль її захисту.

Під ***інформаційною безпекою*** розуміють захищеність інформації і підтримуючої інфраструктури від випадкових або навмисних впливів природного або штучного характеру, що можуть завдати неприйнятного збитку суб'єктам інформаційних відносин, в тому числі власникам і користувачам інформації і підтримуючої інфраструктури.

Захист інформації – комплекс заходів, спрямованих на забезпечення інформаційної безпеки. Аналізуючи підходи до проблем інформаційної безпеки, необхідно починати з виявлення суб'єктів інформаційних відносин та інтересів цих суб'єктів, пов'язаних з використанням інформаційних систем (ІС). Загрози інформаційної безпеки пов'язані з використанням інформаційних технологій.

Інформаційна безпека залежить від усього комплексу заходів та сучасних технологій, керування якими відбувається із застосуванням різноманітних інформаційних систем.

Інформаційна безпека – багатогранна, багатовимірна діяльність, в якій успіх може принести тільки системний, комплексний підхід. Безпека використання інформаційних систем полягає у забезпеченні доступності, цілісності, конфіденційності та підтримці інформаційних ресурсів її інфраструктури.

До основних складових інформаційної безпеки належить конфіденційність, тобто захист від несанкціонованого доступу до інформації.

При аналізі проблематики, пов'язаної з інформаційною безпекою, необхідно враховувати специфіку даного аспекту безпеки, яка полягає в тому, що інформаційна безпека є складова частина інформаційних технологій – області, що розвивається безпрецедентно високими темпами. Тут важливі не стільки окремі рішення (закони, навчальні курси, програмно-технічні вироби), що знаходяться на сучасному рівні, скільки механізми генерації нових рішень, що дозволяють жити в темпі технічного прогресу.

1.2 Предмет та об'єкт захисту

Складність механізмів прийняття сучасних управлінських рішень щодо захисту інформації в новому інформаційному середовищі пов'язана із застосуванням стрімко розвиваючих інформаційних систем, призначених для великого обсягу обробки, обміну та використання їх в сучасному житті кожної особистості, підприємства, держави, світі, а також швидкими темпами розвитку технічних засобів.

Тому **предметом захисту** є інформація, яка зберігається, обробляється, передається в комп'ютерних системах (КС).

Актуальність і важливість проблеми забезпечення безпеки інформаційних технологій обумовлені наступними причинами:

– різке збільшення потужності сучасних комп'ютерів при одночасному спрощенні їх експлуатації;

– різке збільшення обсягів інформації, що накопичується, зберігається, обробляється за допомогою КС;

– високі темпи росту парку персональних комп'ютерів (ПК), що знаходяться в експлуатації в самих різних сферах діяльності;

– різке розширення кола користувачів, що мають безпосередній доступ до обчислювальних ресурсів і масивів даних;

– бурхливий розвиток програмних засобів, що не задовольняють навіть мінімальним вимогам безпеки;

– повсюдне поширення мережних технологій і розвиток глобальної мережі Інтернет.

Інформація не матеріальна, зберігається і передається за допомогою матеріальних носіїв, будь-який матеріальний об'єкт містить інформацію про себе чи інший об'єкт. Вона має такі властивості:

1. **Цінність** інформації, яка визначається степенню її корисності для власника.

2. **Конфіденційність** інформації – це статус, що вказує на необхідні введення обмеження на коло суб'єктів, що мають доступ до даної інформації. Якщо доступ до інформації обмежений, то вона конфіденційна. Конфіденційна інформація може містити державну або комерційну таємницю.

3. **Цілісність** інформації – це властивість бути незмінним в сематичному змісті при функціонуванні системи в умовах випадкових чи навмисних перекручувань або впливів, що руйнують.

4. **Доступність** – це властивість бути доступним для авторизованих законних суб'єктів.

5. **Достовірність** визначається потрібною для власника точністю відображати об'єкти і процеси зовнішнього світу в певних часових і просторових рамках. Якщо інформація спеціально викривлена, то вона називається дезінформацією.

6. **Своєчасність** – відповідність цінності і достовірності певному часовому періоду. Дана властивість виражається відношенням:

$$C(t) = C_0 e^{\frac{-2.3t}{\tau}}$$

де C_0 – цінність інформації в момент її виникнення, t – час від моменту виникнення інформації до часу її оцінки, τ – час від моменту виникненняінформації до її старіння.

Об'єктом захисту інформації є комп'ютерна система або автоматизована система обробки інформації (АСОІ).

Під безпекою АСОІ розуміють її захищеність від випадкового або навмисного втручання в нормальний процес її функціонування, а також від спроб розкрадання, зміни або руйнування її компонентів.

Природа впливів на АСОІ може бути найрізноманітнішою (стихійні лиха, вихід з ладу елементів, помилки персоналу, проникнення зловмисника).

Інформаційна безпека досягається проведенням відповідного рівня політики безпеки. Під **політикою інформаційної безпеки** розуміють сукупність норм, правил, практичних рекомендацій, що регламентують роботу засобів захисту АСОІ від заданої множини загроз безпеки. **Система захисту інформації** – сукупність правових, організаційних мір, технічних, програмних ікриптографічних засобів і методів, що забезпечують захищеність системи.

Необхідно навести два важливі висновки:

1. Трактовка проблем, зв'язаних із інформаційною безпекою, для різних категорій суб'єктів відрізняється (в одному випадку – „нехай краще зламається, ніж ворог дізнається хоча б один секретний біт", а в іншому – „нема у нас секретів, лише все б працювало").

2. Інформаційна безпека не зводиться винятково до захисту від несанкціонованого доступу (НСД), це принципово ширше поняття. Суб'єкт інформаційних відносин може понести збитки не лише від НСД, а й від поломки системи, що викликала перерву в роботі. Тому в більшості випадків захист від НСД стоїть не на першому місці.

1.3 Класифікація загроз безпеки інформації в комп'ютерних системах

Загроза інформації – це потенційно можлива подія, що приводить до наслідків порушення цілісності, доступності та конфіденційності інформації.

Порушення безпеки – реалізація даної загрози безпеки.

Атака (вторгнення, напад) – це дія, що починається зловмисником, і полягає у

пошуку і використанні вразливостей КС для реалізації загрози безпеки.

Вразливість КС – це деяка невдала властивість системи, що уможливлює виникнення і реалізацію загрози.

Протидія загрозам безпеки є метою захисту АСОІ. Безпечна або захищена система – це система із засобами захисту, що успішно і ефективно протистоїть загрозам безпеки.

Загроза безпеки поділяється на об'єктивну і суб'єктивну.

Об'єктивна загроза – незалежна від людини.

Суб'єктивна загроза – навмисна і ненавмисна, залежна від людини.

Ненавмисні суб'єктивні загрози.

1. Ненавмисні дії людей, персоналу, що приводять до часткового або повного виходу комп'ютерних систем з ладу.
2. Неправомірне використання устаткування або зміни режимів робіт цього устаткування.
3. Ненавмисне псування носіїв інформації.
4. Нелегальне використання неліцензійних програм і устаткування.
5. Порушення атрибутів, правил, розмежувань і т.д.
6. Неправильне проектування архітектури системи, у яку заздалегідь закладені „люки», „лази" і „дірки" для стороннього порушника. Ігнорування сторонніх порушень.
7. Вхід у комп'ютерну систему в обхід установлених правил.
8. Введення помилкових даних та інші.

Основні навмисні загрози.

1. Фізичне руйнування системи.
3. Дезорганізація комп'ютерної або обчислювальної системи.
4. Впровадження в число персоналу «зацікавленої» людини.
5. Навмисна загроза: шантаж, загроза персоналу комп'ютерної системи.
6. Застосування засобів технічних розвідок: фотографування,підслуховування і т.д.
7. Перехоплення побічних електромагнітних випромінювань і наведень.
8. Перехоплення переданих даних по каналу зв'язку і т.д.
9. Розкрадання носіїв інформації.
10. Незаконне одержання правил і атрибутів розмежування доступу.
11. Незаконне використання терміналів законних користувачів з метою проникнення в систему під іменем санкціонованого користувача.
12. Розкриття шифрів криптозахисту санкціонованих користувачів.
13. Упровадження закладок типу „троянських коней", „жучків", „пасток"і т.д. з метою впровадження недекларованого програмного забезпечення санкціонованих користувачів.
14. Незаконне підключення до ліній передачі даних з метою роботи між рядків, видаючи себе за законного користувача та інші.

Класифікація загроз інформаційної безпеки

Класифікація всіх можливих загроз інформаційної безпеки комп'ютерної системи може бути проведена за рядом базових ознак.

За природою виникнення.

Природні загрози – загрози, викликані впливами на КС і її компоненти об'єктивних фізичних процесів або стихійних природних явищ, що не залежать від людини.

Штучні загрози – загрози інформаційної безпеки КС, викликані діяльністю людини.

За ступенем навмисності прояву.

Загрози випадкової дії і/або загрози, викликані помилками або недбалістю персоналу. Загрози, не зв'язані з навмисними діями зловмисників і реалізовані у випадкові моменти часу, називають випадковими або ненавмисними. Реалізація загроз цього класу приводить до найбільших втрат інформації (до 80 % збитку). При цьому може відбуватися знищення, порушення цілісності, доступності і конфіденційності інформації, наприклад:

− прояв помилок програмно-апаратних засобів КС;

− некомпетентне використання, настроювання або неправомірне відключення засобів захисту персоналом служби безпеки;

− ненавмисні дії, що приводять до часткової або повної відмови системи або руйнуванню апаратних, програмних, інформаційних ресурсів системи (ненавмисне псування устаткування, видалення, перекручування файлів з важливою інформацією або програм, у тому числі системних і т.д.);

− неправомірне включення устаткування або зміна режимів роботи пристроїв і програм;

− ненавмисне псування носіїв інформації;

− пересилання даних по помилковій адресі абонента (пристрою);

− введення помилкових даних;

− ненавмисне ушкодження каналів зв'язку.

Загрози навмисної дії, наприклад:

− традиційне або універсальне шпигунство і диверсії (підслуховування, візуальне спостереження; розкрадання документів і машинних носіїв,розкрадання програм і атрибутів системи захисту, підкуп і шантаж співробітників, збір і аналіз відходів машинних носіїв, підпали, вибухи);

− несанкціонований доступ до інформації (реалізується за допомогою відсутності системи розмежування доступу, збоями або відмовою технічних засобів),

− побічні електромагнітні випромінювання і наведення;

− несанкціонована модифікація структур (алгоритмічної, програмної, технічної);

− інформаційні інфекції (шкідливі програми).

За безпосереднім джерелом загроз.

Загрози, безпосереднім джерелом яких є природнє середовище

(стихійні лиха, магнітні бурі, радіоактивне випромінювання і т.д.).

Загрози, джерелом яких є людина, наприклад:

− впровадження агентів у число персоналу системи (у тому числі, можливо, і в адміністративну групу, що відповідає за безпеку);

− вербування (шляхом підкупу, шантажу і т.д.) персоналу або окремихкористувачів, що мають визначені повноваження;

− загроза несанкціонованого копіювання секретних даних користувачем КС;

− розголошення, передача або втрата атрибутів розмежування доступу

(паролів, ключів шифрування, ідентифікаційних карток, пропусків і т. д.).

Загрози, безпосереднім джерелом яких є санкціоновані програмно- апаратні засоби, наприклад:

− запуск технологічних програм, здатних при некомпетентномукористуванні викликати втрату працездатності системи (зависання або зациклення) або необоротні зміни в системі (форматування або реструктуризацію носіїв інформації, видалення даних і т. д.);

− виникнення відмови в роботі операційної системи.

Загрози, безпосереднім джерелом яких є несанкціоновані програмно- апаратні засоби, наприклад:

− нелегальне впровадження і використання неліцензійних програм (ігрових, навчальних, технологічних і інших, що не є необхідними для виконання зловмисником своїх службових обов'язків) з наступною необґрунтованою витратою ресурсів (завантаження процесора, захоплення оперативної пам'яті і пам'яті на зовнішніх носіях);

− зараження комп'ютера вірусами з деструктивними функціями.

За положенням джерела загроз.

Загрози, джерело яких розташоване поза контрольованою зоною території (приміщення), на якій знаходиться КС, наприклад:

− перехоплення побічних електромагнітних, акустичних і інших випромінювань приладів і ліній зв'язку, а також наведень активних випромінювань на допоміжні технічні засоби, що безпосередньо не беруть участь в обробці інформації (телефонні лінії, мережі харчування, опалення);

− перехоплення даних, переданих по каналах зв'язку, і їхній аналіз з метою визначення протоколів обміну, правил входження в зв'язок і авторизації користувача і подальших спроб їхньої імітації для проникнення в систему;

− дистанційна фото- і відеозйомка.

Загрози, джерело яких розташоване у межах контрольованої зони території (приміщення), на якій знаходиться КС, наприклад:

− розкрадання виробничих відходів (роздруківок, записів, списаних носіїв інформації і т.д.);

− відключення або вивід з ладу підсистем забезпечення функціонування обчислювальних систем (електроживлення, охолодження і вентиляції, ліній зв'язку і т.д.);

− застосування підслухуючих пристроїв.

Загрози, джерело яких має доступ до периферійних пристроїв КС. Загрози, джерело яких розташоване в КС, наприклад:

− проектування архітектури системи і технології обробки даних, розробка прикладних програм, що становлять небезпеку для працездатності системи і безпеки інформації;

− некоректне використання ресурсів КС.

За ступенем залежності від активності КС.

Загрози, що можуть виявлятися незалежно від активності КС, наприклад:

− розкриття шифрів криптоаналізу інформації;

− розкрадання носіїв інформації (магнітних дисків, стрічок, мікросхем пам'яті, запам'ятовуючих пристроїв і комп'ютерних систем).

Загрози, що можуть виявлятися тільки в процесі автоматизованої обробки даних (наприклад, загрози виконання і поширення програмних вірусів).

За ступенем впливу на КС.

Пасивні загрози, що при реалізації нічого не змінюють у структурі і змісті КС, наприклад: загроза копіювання секретних даних.

Активні загрози, що при впливі вносять зміни в структуру і зміст КС, наприклад:

− впровадження апаратних спецвкладок, програмних „закладок" і „вірусів", тобто таких ділянок програм, що не потрібні для виконання заявлених функцій, але дозволяють перебороти систему захисту, таємно і незаконно здійснити доступ до системних ресурсів з метою реєстрації і передачі критичної інформації або дезорганізації функціонування системи;

− дії по дезорганізації функціонування системи (зміна режимів роботи пристроїв або програм, страйк, саботаж персоналу, постановка могутніх активних радіоперешкод на частотах роботи пристроїв системи і т.д.);

− загроза навмисної модифікації інформації.

За етапами доступу користувачів або програм до ресурсів КС.

Загрози, що можуть виявлятися на етапі доступу до ресурсів КС (наприклад, загрози несанкціонованого доступу в КС).

Загрози, що можуть виявлятися після дозволу доступу до ресурсів КС (наприклад, загрози несанкціонованого або некоректного використання ресурсів КС).

Несанкціонований доступ є найбільш розповсюдженим і різноманітним видом комп'ютерних порушень. Суть НСД складається в одержанні користувачем (порушником) доступу до об'єкта з порушенням правил розмежування доступу, установлених відповідно до прийнятої в організації політики безпеки. НСД використовує будь-яку помилку в системі захисту і можливий при нераціональному виборі засобів захисту, їх некоректній установці і настроюванню. НСД може бути здійснений як штатними засобами АСОІ, так і спеціально створеними апаратними і програмними засобами.

За способом доступу до ресурсів КС.

Загрози, спрямовані на використання прямого стандартного шляху доступу до ресурсів. Наприклад:

− Незаконне одержання паролів і інших реквізитів розмежування доступу (агентурним шляхом, використовуючи недбалість користувачів, підбором, імітацією інтерфейсу системи і т.д.) з наступним маскуванням підзареєстрованого користувача – «маскарад».

Перехоплення паролів здійснюється спеціально розробленими програмами. При спробі законного користувача ввійти в систему програма- перехоплювач імітує на екрані дисплея введення імені і пароля користувача, що відразу пересилаються власникові програми-

перехоплювача, після чого на екран виводиться повідомлення про помилку і керування повертається операційній системі. Користувач припускає, що припустився помилки при введенні пароля. Він повторює введення й одержує доступ у систему. Власник програми-перехоплювача, що одержав ім'я і пароль законного користувача, може тепер використовувати їх у своїх цілях. Існують і інші способи перехоплення паролів.

«Маскарад» – це виконання яких-небудь дій одним користувачем від імені іншого користувача, що володіє відповідними повноваженнями. Метою «маскараду» є приписування яких-небудь дій іншому користувачеві абоприсвоєння повноважень і привілеїв іншого користувача. Прикладами реалізації „маскараду" є: вхід у систему під ім'ям і паролем іншого користувача (цьому „маскарадові" передує перехоплення пароля); передача повідомлень у мережі від імені іншого користувача; «маскарад» особливо небезпечний у банківських системах електронних платежів, де неправильна ідентифікація клієнта через „маскарад" зловмисника може привести до великих збитків законного клієнта банку; несанкціоноване використання терміналів користувачів, що мають унікальні фізичні характеристики, такі як номер робочої станції в мережі, фізична адреса, адреса в системі зв'язку, апаратний блок кодування і т.д.

Загрози, спрямовані на використання схованого нестандартного шляху доступу до ресурсів КС, наприклад:

− вхід у систему в обхід засобів захисту (завантаження сторонньої операційної системи зі змінних магнітних носіїв і т.д.);

− загроза несанкціонованого доступу до ресурсів КС шляхом використання недокументованих можливостей ОС.

За поточним місцем розташування інформації, збереженої й оброблюваної в КС.

Загрози доступу до інформації на зовнішніх запам'ятовуючих пристроях (наприклад, загроза несанкціонованого копіювання секретної інформації з твердого диска).

Загрози доступу до інформації в оперативній пам'яті, наприклад:

− читання залишкової інформації з оперативної пам'яті;

− читання інформації з областей оперативної пам'яті, використовуваних операційною системою (у тому числі підсистемою захисту) або іншими користувачами, в асинхронному режимі, використовуючи недоліки мультизадачних КС і систем програмування;

− загроза доступу до системної області оперативної пам'яті зі сторін прикладних програм.

Загрози доступу до інформації, що циркулює в лініях зв'язку, наприклад:

− незаконне підключення до ліній зв'язку з метою роботи „між рядками" з використанням пауз у діях законного користувача від його імені з наступним уведенням помилкових повідомлень або модифікацією переданих повідомлень;

− незаконне підключення до ліній зв'язку з метою прямої підміни законного користувача шляхом його фізичного відключення після входу в систему й успішної аутентифікації з наступним уведенням дезінформації і нав'язуванням помилкових повідомлень.

Загрози доступу до інформації, відображуваної на терміналі або друку на принтері, наприклад, загроза запису відображуваної інформації на приховану відеокамеру.

2. Загрози безпеці інформації та методи протидії їм

2.1 Класифікація вразливостей систем безпеки

Загрози інформаційної безпеки проявляються не самостійно, а через можливу взаємодію з найбільш слабкими ланками системи захисту, тобто через фактори уразливості. Загроза призводить до порушення діяльності систем на конкретному об'єкті-носії.

Основні вразливості виникають внаслідок дії наступних факторів:

- ✓ недосконалість програмного забезпечення, апаратної платформи;
- ✓ різні характеристики будови автоматизованих систем в інформаційному потоці;
- ✓ частина процесів функціонування систем є неповноцінною;
- ✓ неточність протоколів обміну інформацією та інтерфейсу;
- ✓ складні умови експлуатації і розташування інформації.

Найчастіше джерела загрози запускаються з метою отримання незаконної вигоди внаслідок заподіяння шкоди інформації. Але можливі і випадкові загрози через недостатні міри захисту і дії масового загрозливого фактору.

Існує поділ вразливостей за класами:

- об'єктивні;
- випадкові;
- суб'єктивні.

Якщо усунути або, як мінімум, послабити вплив вразливостей, можна уникнути повноцінної загрози, спрямованої на систему зберігання інформації.

Таким чином, класифікація загроз ІБ розподіляється за характером загрози, видом впливу, джерелом та об'єктом загрози.

Дослідники виділяють три основних види загроз безпеки:

- ✓ *загрози розкриття;*
- ✓ *загрози цілісності;*
- ✓ *загрози відмови в обслуговуванні.*

Загроза розкриття полягає в тому, що інформація стає відомою тому, кому не варто було б її знати. У термінах комп'ютерної безпеки загроза розкриття має місце щоразу, коли отриманий доступ до певної конфіденційної інформації, що зберігається в ІС, або передається від однієї системи до іншої.

Загроза цілісності включає будь-яку навмисну зміну (модифікацію або навіть видалення) даних, що зберігаються в ІС, або передаються з однієї системи до іншої. Зазвичай вважається, що загрозі розкриття піддаються більшою мірою державні структури, а загрозі цілісності – ділові або комерційні структури.

Загроза відмови в обслуговуванні виникає щоразу, коли в результаті деяких дій блокується доступ до певного ресурсу ІС. Реальне блокування може бути постійним, так щоб запитуваний ресурс ніколи не був отриманий, або блокування може викликати тільки затримку запитуваного ресурсу, досить тривалу для того, щоб він став непридатним. У таких випадках кажуть, що ресурс вичерпаний.

Крім того, пропонується наступна класифікація загроз ІБ. Хоча єдиної й загальноприйнятої класифікації загроз ІБ не існує й, швидше за усе, не буде взагалі, тому що згодом з'являються нові загрози, які все складніше ідентифікувати.

Найпоширеніші dos атаки (відмова в обслуговуванні).

1. **Ping-of-Death.**

Посилає ICMP-пакет, розміром більше 64 Кб, що може призвести до переповнення буфера операційної системи і виведення системи, що атакується, з ладу.

2. **SYN Flood .**

Дуже швидко посилає велику кількість TCP SYN-пакетів (які ініціюють з'єднання), залишаючи жертву чекати величезну кількість з'єднань і викликаючи таким чином посилене завантаження ресурсів і відмову від санкціонованих з'єднань.

3. **Land/Latierra.**

Посилає підроблений SYN-пакет з ідентичними вихідною адресою та кінцевим портом

так, що система рухається по нескінченній петлі, намагаючись виконати TCP-з'єднання.

4. WinNuke.

Посилає OOB/URG-дані для TCP-з'єднання із портом 139 (NetBIOS Session/SMB), що призводить до зависання ОС Windows. Найбільшому впливу піддається ОС Windows 95, пізніші версії ОС Windows мають проти цієї атаки відповідний захист.

Найпоширеніші процеси сканування

1. Ping sweeps.

Протягом цього простого процесу сканування діапазон IP-адрес аналізується утилітою **ping** (так зване пінгування) з метою визначення активних комп'ютерів. Слід зауважити, що більшість складних сканерів буде використовувати інші протоколи (такі, як SNMP sweep), щоб виконувати ту ж саму дію.

2. TCP-сканування.

Зондування відкритих TCP-портів у пошуках сервісів, які може використовувати порушник. Сеанси сканування можуть використовувати звичайні TCP-з'єднання або приховані (stealth) сеанси сканування, що

використовують наполовину відкриті з'єднання (для того, щоби захистити їх від реєстрації в журналах) або FIN-сеанси сканування (ніколи не відкривають порт, але тестують, якщо щось прослуховується). Сеанси сканування можуть бути послідовними або випадковими, або зконфігуровані за переліком портів.

3. UDP-сканування.

Ці сеанси сканування є складнішими, тому що (User Datagram Protocol) UDP- працює без встановлення віртуального з'єднання. Метод полягає в тому, щоби послати «сміттєвий» UDP-пакет до наміченого порту. Більшість машин будуть реагувати за допомогою ICMP-повідомлення «destination port unreachable», яке вказує, що на даному порту немає сервісу, який прослуховується. Однак багато комп'ютерів «поглинають» ICMP- повідомлення, тому ви не зможете здійснювати швидке UDP-сканування.

4. Ідентифікація ОС.

Шляхом посилання неприпустимих (або дивних) ICMP або TCP-пакетів порушник може ідентифікувати ОС. Стандарти зазвичай встановлюють, яким чином комп'ютери повинні реагувати на легальні пакети, тому машини мають тенденцію бути однаковими у своїй реакції на припустимі вхідні дані. Однак стандарти упускають (як правило навмисно) реакцію на неприпустимі вхідні дані. Таким чином, унікальні реакції кожної ОС на неприпустимі вхідні дані формують сигнатуру, яку хакери можуть використовувати для того, щоби зрозуміти, під чиїм управлінням функціонує обраний комп'ютер. Цей тип діяльності має місце на нижньому рівні (начебто прихованих сеансів TCP- сканування), на якому аналізовані системи не реєструють події.

2.2 Сучасні загрози та методи протидії їм

Носіями загроз інформаційній безпеці є джерела загроз, якими виступають як суб'єктивні (особистості), так і об'єктивні обставини (конкуренти, злочинці, корупціонери, адміністративно-управлінські органи, інше). Джерела загроз переслідують при цьому наступні цілі: ознайомлення з відомостями які охороняють, їх модифікація в корисливих цілях і знищення для нанесення прямих матеріальних збитків.

Всі джерела загроз інформаційній безпеці можна розділити на три основні групи.

Обумовлені діями суб'єкта (антропогенні джерела) – суб'єкти, дії яких призводять до порушення безпеки інформації, кваліфікуються як умисні або випадкові злочини.

Джерела, дії яких можуть призвести до порушення безпеки інформації можуть бути як зовнішніми так і внутрішніми. Дані джерела можна зпрогнозувати, і вжити адекватних заходів.

Обумовлені технічними засобами (техногенні джерела) – ці джерела загроз менш прогнозовані, безпосередньо залежать від властивостей техніки і тому вимагають особливої уваги. Дані джерела загроз інформаційній безпеці, також можуть бути як внутрішніми, так і зовнішніми.

Стихійні джерела – дана група об'єднує обставини, що становлять непереборну силу (стихійні лиха або інші обставини, що неможливо передбачити або запобігти, або можливо передбачити, але неможливо запобігти). Ці обставини, що носять об'єктивний і абсолютний характер, поширюється на всіх. Такі джерела загроз абсолютно не піддаються прогнозуванню і, тому заходи проти них повинні застосовуватися завжди. Стихійні джерела, як правило, є зовнішніми по відношенню до інформаційних джерел що захищаються і під ними, як правило, розуміються природні катаклізми.

Загроза безпеці комп'ютерної системи – це потенційно можлива подія, незалежно від того, навмисна чи ні, що може здійснити небажаний вплив на саму систему, а також на інформацію, що зберігається в ній.

Вразливість комп'ютерної системи – це якась її невдала характеристика, що уможливлює виникнення загрози. Інакше кажучи, саме через наявність вразливостей у системі відбуваються небажані події.

Атака на комп'ютерну систему – це дія, що вчиняється зловмисником і полягає в пошуку й використанні загрози або іншої вразливості. Таким чином, атака – це реалізація загрози. Слід зауважити, що таке тлумачення атаки (за участю людини, яка має злий намір), виключає присутній у визначенні загрози елемент випадковості. Але, як показує досвід, часто буває неможливо розрізнити навмисні й випадкові дії, і надійна система захисту повинна адекватно реагувати на кожне з них.

Більш детально розглянемо особливості зовнішніх загроз.

1. Хакерські атаки.

Отримавши доступ до системи, вони направлені на крадіжку конфіденційних даних або встановлюють шкідливі програми та використовують "взламані" комп'ютери для розсилки спаму. В програми закрадаються помилки, що робить їх уразливими для атаки. Хакерам ці лазівки дозволяють проникнути в систему, а ті, хто пише віруси, використовують помилки в коді додатків, щоб забезпечити автоматичний запуск на комп'ютері шкідливих програм.

Хакери – це електронні "взламники", які проникають в комп'ютерну систему, використовуючи особливі уразливі лазівки у програмному забезпеченні. Захиститися від них можна за допомогою особливого додатку – мережевого екрану з пакетною фільтрацією, що входить до складу антивірусних програм і робить комп'ютер невидимим для хакерів.

Для захисту від шкідливого коду і хакерських атак:

- ✓ встановлюється антивірусна програма;
- ✓ встановлюється оновлення ОС Windows (Update), що відповідає за безпеку;
- ✓ увага при роботі зі спамом в електронній пошті і системах миттєвих повідомлень;
- ✓ збереження резервної копії (BackUp) даних.

2. Технологія інфраструктури відкритих ключів.

Технологія інфраструктури відкритих ключів дозволяє перевіряти і засвідчувати справжність користувача. Інфраструктура відкритих ключів або PKI забезпечує єдину ідентифікацію, аутентифікацію і авторизацію користувачів системи, додатків і процесів і разом з цим гарантує доступність, цілісність і конфіденційність інформації. Інфраструктура PKI являє собою систему цифрових сертифікатів, носіями яких є USB-ключі або смарт-карти.

При використанні індивідуального секретного пароля і засобів криптографічного захисту, цифрові сертифікати отримують роль електронних паспортів. Використання в корпоративній мережі технології інфраструктури відкритих ключів значно підвищує безпеку всієї мережі в цілому, так як дозволяє відмовитися від використання парольної аутентифікації користувачів всередині, а також забезпечує безпечний доступ віддалених користувачів в систему. Основними носіями інформації є USB-ключі та смарт-карти. Користувачам не треба запам'ятовувати складні паролі і періодично їх міняти – досить підключити електронний ключ або смарт-карту і ввести PIN-код.

3. Системи одноразових паролів.

Системи багатофакторної аутентифікації засновані на технології одноразових паролів (one time password) OTP призначені для аутентифікації мобільних користувачів, які

відрізняється простотою у використанні, установці і адмініструванні.

Дана технологія заснована на тому, що пароль користувача не постійний і змінюється з плином часу спеціальним пристроєм (апаратним або програмним) – токеном. Дане рішення широко використовується в системах віддаленого доступу, в тому числі системах клієнт-банк, для аутентифікації користувачів при доступі з недовірених середовища (Інтернет-кафе, бізнес-центри, і т.д.).

OTP-токен – мобільний персональний пристрій, що належить певному користувачеві і генерує одноразові паролі, які використовуються для аутентифікації даного користувача. OTP-токени мають невеликий розмір і випускаються у вигляді: кишенькового калькулятора; брелока; смарт-карти; пристрою, комбінованого з USB-ключем; спеціального програмного забезпечення для кишенькових комп'ютерів. Як приклад рішень OTP, можна привести лінійку RSA SecurID, ActivCard Token, комбінований USB-ключ Aladdin eToken NG-OTP.

Структура системи:

✓ сервер аутентифікації (баз даних акаунтів, прив'язаних до пристроїв, синхронізація за часом);

✓ канал передачі;

✓ форма для введення аутентифікаційних даних (зазвичай три поля (Login, OTP, PIN));

✓ клієнтська частина (OTP брелок і ін.).

4. Біометричні системи.

Біометричні системи – це вимірні фізіологічні або поведінкові дані людини. Біометричні дані унікальні для кожної людини і їх можна використовувати для встановлення особи або перевірки декларованих особистих даних:

✓ для ідентифікації користувача (замість введення імені користувача);

✓ для однофакторної аутентифікації користувача;

✓ спільно з паролем або аутентифікаційї токеном (таким, як смарт- карта) для забезпечення двофакторної аутентифікації.

Біометричні дані діляться на групи:

1. *Фізіологічні біометричні* характеристики – засновані на даних, отриманих шляхом вимірювання анатомічних характеристик людини, таких, як відбиток пальця, форма обличчя або кисті, сітківка ока.

2. *Поведінкові біометричні* характеристики (динамічні) – засновані на даних, отриманих шляхом вимірювання дій людини. Характерною рисою для поведінкових характеристик є їх протяжність в часі – вимірюється дією, що має початок, середину і кінець. Наприклад, голос, підпис.

5. Криптографічний захист даних.

Криптографія – область знань, що вивчає тайнопис (криптографія) і методи його розкриття (криптоаналіз). Криптографія вважається розділом математики.

Мета криптографічної системи полягає в тому, щоби зашифрувати вихідний текст (шифртекст, криптограма).

Одержувач, якому він призначений, повинен бути здатний розшифрувати («дешифрувати») цей шифртекст, відновивши, таким чином, відповідний йому відкритий текст. При цьому зловмисник повинен бути нездатний розкрити вихідний текст.

Існує відмінність між розшифрування (дешифрованием) і розкриттям шифртексту. Широко відомим історичним прикладом криптосистеми є так званий шифр Цезаря, що представляє з себе просту заміну кожної букви відкритого тексту третьої наступної за нею буквою алфавіту (з циклічним перенесенням, коли це необхідно) Наприклад, «A» замінювалося на «D», «B» на «E», «Z» на «C».

Всі методи шифрування поділяються на дві групи:

✓ шифри з секретним ключем (симетрична схема);

✓ шифри з відкритим ключем (асиметрична схема).

Перший тип шифрів має на увазі наявність інформації (ключа), володіння якою

дозволяє як зашифрувати, так і розшифрувати повідомлення. Шифри з відкритим ключем – відкритого і закритого типу; один використовується для шифрування, інший для розшифровки повідомлень.

Основні напрямки шифрування:

✓ шифрування даних на локальних дискових системах (клієнтське шифрування);

✓ криптографічний алгоритм (шифр) – математичний спосіб обробки інформації для приховування її змісту.

Основні тенденції застосування шифрування:

✓ шифрування окремих файлів;

✓ шифрування окремих розділів на жорсткому диску, віртуальних дисків;

✓ шифрування жорстких дисків цілком.

6. Електронний підпис (ЕП).

Електронний підпис – послідовність символів, отримана в результаті криптографічного перетворення вихідної інформації з використанням закритого ключа ЕЦП, яка дозволяє підтверджувати цілісність і незмінність цієї інформації, а також її авторство за умови використання відкритого ключа ЕП і його сертифіката.

Цифровий підпис забезпечує:

✓ вихідні джерела документа. Залежно від деталей визначення «документа» можуть бути підписані такі поля як автор, внесені зміни, мітка часу т.і.;

✓ захист від змін документа. При будь-якому випадковому або навмисному зміні документа (або підпису) зміниться хеш (хеш-функція або геш-функція – функція, що перетворює вхідні дані будь-якого розміру в дані фіксованого розміру), отже підпис стане недійсним;

✓ неможливість відмови від авторства. Так як створити коректний підпис можна лише знаючи закритий ключ, а він відомий тільки власнику, то власник не може відмовитися від свого підпису під документом.

Для підпису документа з початку обчислюється значення хеш-функції для документа, а потім це значення за спеціальним криптоалгоритмом підписується секретним ключем автора документа.

Для перевірки справжності документа необхідно за допомогою відкритого ключа перевірити підпис, потім обчислити його хеш-значення і порівняти з підписаним контрольним підписом. Якщо обидва значення збігаються, то підпис вірний, інакше документ недійсний.

2.3 Інформаційні ризики

Сучасні системи спеціального призначення характеризуються наявністю в своєму складі інформаційних систем (ІС). Реалії часу диктують необхідність в умовах обмеженої можливості фінансування розробки ІС отримати оцінку розумної достатності захищеності таких ІС від ризиків. Класичне визначення ризику визначає його як комбінацію ймовірності події та її наслідків [ISO Guide 73: 2002]. Серед різновидів ризиків українські та зарубіжні автори виділяють операційний ризик відповідно до Базель III (Базель III - документ Базельського комітету з банківського нагляду, що містить методичні рекомендації в області банківського регулювання), що визначається ризиком прямих або непрямих втрат, джерелами яких можуть бути невідповідні або неправильно організовані внутрішні процеси, людські ресурси і системи або зовнішні події. Наслідком виникнення таких подій є зниження вірогідності, повноти та актуальності створюваної і інформації, що обробляється.

Отже, інформаційний ризик може бути визначений як різновид операційного ризику, що відбувається в результаті неадекватних і помилкових внутрішніх процесів, дій працівників або зовнішніх подій. Стандарт ISO27000: 2018 вказує: інформаційний ризик – це потенційна можливість того, що загроза буде використовувати уразливість активу або групи активів, завдаючи шкоди організації.

Ризик викликає інцидент інформаційної безпеки – одне або серія небажаних або несподіваних подій інформаційної безпеки, що мають значну ймовірність порушення бізнес-

операцій або становлять загрозу для інформаційної безпеки. Таким чином, приступаючи до аналізу інформаційних ризиків, необхідно визначити перелік об'єктів і їх вразливостей, на які спрямовані атаки, реалізуючи загрози. Інформаційні ризики спрямовані на наступні види активів: інформація, мережеве, системне і прикладне програмне забезпечення, персональні комп'ютери, накопичувальні і друкувальні пристрої, мережеві сервера, шлюзи, інтерфейси, сервіси.

2.4 Витік інформації

Сьогодні більшість підприємств використовують багаторівневі системи обробки інформації – комп'ютери, хмарні сховища, корпоративні мережі і т. п. Всі ці системи не тільки передають дані, але і є середовищем їх можливого витоку. Витік секретної інформації – процес неконтрольованого розголошення ключових даних.

Комерційна таємниця – інформація про організацію діяльності підприємства, технології розробки продукції, дані про грошові потоки, інтелектуальна власність та інші відомості, володіючи якими отримуються фінансові вигоди.

Причина 1 – Персонал

Кожен співробітник підприємства є потенційною загрозою для безпеки інформації. Часто люди забирають роботу додому – переміщують робочі файли на свої флеш-носії, передають їх по незахищеним каналам з'єднання, обговорюють інформацію зі співробітниками конкуруючих компаній.

Дії персоналу бувають навмисними і ненавмисними. Ненавмисні дії – це наслідок незнання регламенту роботи з комерційною інформацією.

Ризик витоку інформації від персоналу є завжди, і його не можна виключити повністю. Служба безпеки може вжити заходів, щодо обмежень взаємодії працівників з конфіденційною інформацією та впровадити правила розмежування доступу.

Правила – перелік чітких прав і обмежень, що повинні дотримуватися кожним співробітником. Їх основний принцип – кожен працівник взаємодіє тільки з тими даними, які потрібні для його роботи. Таким чином, простий менеджер не зможе дізнатися технологію розробки продукції та інші важливі дані, що бажає знати зловмисник.

Питаннями контролю роботи персоналу з секретними матеріалами повинен займатися уповноважений співробітник або відділ безпеки. Їхнє завдання це стежити за діяльністю працівників протягом усього робочого дня і оперативно виявляти всі випадки витоку інформації.

На практиці виявити людину, що зливає комерційну таємницю, можна за такими ознаками:

– *Співробітник без попередження затримується після роботи на своєму робочому місці.*

В такому випадку є ймовірність того, що він намагається отримати доступ до секретної інформації в момент, коли поруч нікого немає.

На такого працівника потрібно звернути увагу і простежити, чи не є його метою дізнатися таємні відомості. Контролювати час перебування персоналу на робочому місці допомагають спеціальні системи обліку доступу. Починати розслідування потрібно лише в тому випадку, якщо стали відомі конкретні факти витоку інформації, що захищається.

– *Співробітник зберігає на свій персональний комп'ютер або смартфон занадто багато електронних документів компанії.*

Такий варіант витоку можна відстежити в компаніях, що використовують системи захисту файлової системи. Суть їх роботи полягає в створенні загального сервера, що діє в рамках однієї корпоративної або Wi-Fi-мережі. Під час кожного відкриття, копіювання та переміщення даних на службовому комп'ютері вся інформація про процеси надходить на сервер. Таким чином, адміністратор безпеки може виявити, з якого комп'ютера і в якій кількості була переміщена секретна інформація.

– *Співробітник без необхідності копіює паперовий документообіг електронним (сканує документи або фотографує).*

Згідно з нормами документування, все фізичні папки і файли з комерційною таємницею

повинні зберігатися в захищеній частині архіву. Доступ до документів можливий тільки для уповноважених працівників. Всі дані про отримання документа з таємницею для користування повинні документуватися (із зазначенням імені працівника і точного часу видачі документа).

Якщо ж секретний документ потрапив в руки недобросовісного співробітника, відстежити його несанкціоноване копіювання можна на сканері або ксероксі, що зберігає звіт щодо діяльності за останній час. Також існують факсимільні апарати, доступ до яких можливий тільки після правильного введення пари «ідентифікатор користувача-пароль».

– *Працівник регулярно порушує загальні вимоги безпеки при роботі з комерційною таємницею.*

Якщо персонал регулярно намагається обійти систему заборони, переглядаючи заборонені ресурси, або використовує особисту техніку для обробки секретних даних, необхідно впровадити додаткові системи контролю користувачів. Наприклад, DLP-системи. Їх завдання полягає в моніторингу всіх листувань користувачів з комерційної пошти та інших електронних скриньок, які зареєстровані в системі. Також модуль захисту забороняє установку стороннього ПЗ, а всі дії співробітника за комп'ютером видно адміністратору безпеки.

– *Співробітник був викритий в контактах із службовцями конкуруючих компаній.*

У великих компаніях працівники часто спілкуються поза робочим часом. Таким чином, вони отримують більше інформації один про одного і можуть дізнатися про зв'язки колеги і працівника конкуруючої організації. Ймовірність звичайних дружніх відносин між людьми теж можлива, але краще оповістити керівництво компанії про це, щоб уникнути непотрібних підозр.

Причина 2 – Проблеми підбору кадрів

Часта зміна персоналу, масштабні зміни в організації роботи компанії, зниження заробітних плат, скорочення співробітників – все це є частиною

«плинності» кадрів. Таке явище часто стає причиною витоку секретної інформації.

Криза, нестача коштів для видачі зарплат змушують керівництво погіршувати умови роботи персоналу. В результаті підвищується невдоволення працівників, які можуть звільнитися або ж просто почати поширювати секретні дані конкурентам. Проблема зміни персоналу особливо важлива для керівних посад, адже всі керуючі повинні мати доступ до секретної документації.

Загрозу поширення таємниці можуть нести не тільки ті співробітники, які звільнилися, але і поточні працівники, рівень мотивації яких знижений.

Для запобігання проблеми слід створити для працівників максимально комфортні умови роботи. У разі серйозної кризи рекомендується зібрати персонал для обговорення можливих шляхів виходу зі складної ситуації. Важливо повідомляти співробітників про всі зміни в нарахуванні заробітних плат заздалегідь, а не за фактом виплати окладу.

Часом несприятливу атмосферу в колективі створює один співробітник. Встановлення систем автоматизованого профайлингу які аналізують листування працівників в електронній пошті і месенджерах та створюють їх психологічні портрети. Система визначає позитивні і негативні сторони характеру людини, що дозволяє приймати вірні управлінські рішення.

Для усунення «плинності» працівників важливо виконувати наступні рекомендації:

– *Налагодити систему найму кадрів.*

Всі передові організації мають спеціальний відділ, що займається питаннями найму, звільнення і підтримки співробітників. Не слід шукати працівника на вакансію, що звільнилася або з'явилась якомога швидше. Хороший HR (фахівець з підбору кадрів Human resources) зобов'язаний прослухати кілька претендентів на посаду, поширити інформацію про вільну вакансію на популярних Інтернет-майданчиках, провести підсумковий конкурс, результати якого визначать кандидатуру, яка найбільше підходить.

– *Впровадження системи винагород.*

За успіхи в роботі, перевиконання планів і укладення вигідних контрактів співробітників потрібно заохочувати. Прикладами заохочення можуть бути підвищення

заробітної плати, покращення умов роботи, просування по кар'єрним сходах.

– *Надання всім співробітникам можливості професійного зростання, підвищення кваліфікації.*

Хороші компанії завжди відправляють своїх співробітників на курси підвищення кваліфікація або ж закуповують онлайн-тренінги для більш зручного проходження навчання. Також рекомендується організовувати тренінги від провідних професіоналів галузі.

Причина 3 – Відрядження

Робочий процес фірми включає ділові зустрічі, поїздки в інші філії компанії, країни. Співробітники, які часто виїжджають у відрядження, можуть ненавмисно стати основною причиною витоку секретної інформації підприємства.

У поїздці такий працівник завжди має при собі особистий або корпоративний ноутбук/смартфон, що обробляє захищені документи. Техніка може бути залишена в громадському місці, зламана або викрадена. Якщо за співробітником ведеться стеження або ж він зустрічається з керівниками конкуруючої компанії, загублений ноутбук може стати головним джерелом розголошення службової інформації.

Для запобігання подібних випадків важливо використовувати системи шифрування жорсткого диска тих ПК, що видаються співробітникам на час ділових зустрічей. Навіть в результаті крадіжки і несанкціонованого доступу інформація буде надійно захищена, і зламати її, без знання ключа, буде неможливо.

Причина 4 – Співпраця з іншими компаніями

Більшість автоматизованих систем захисту здатні обмежити доступ до службової інформації тільки в рамках однієї будівлі або одного підприємства (якщо кілька філій використовують загальний сервер зберігання даних).

У процесі спільного виконання проєкту декількома фірмами, служби безпеки не можуть в повній мірі простежити за тим, як реалізується доступ до службової таємниці кожного з підприємств.

Як і в попередньому випадку, використання криптоконтейнера (систем шифрування жорсткого диска) дозволить захистити таємну інформацію від злому.

Причина 5 – Використання складних IT-інфраструктур

Великі корпорації використовують комплексні системи захисту службових відомостей. Автоматизовані системи мають на увазі наявність декількох відділів безпеки і роботу понад п'яти системних адміністраторів, завдання яких полягає тільки в підтримці збереження комерційної таємниці.

Складність системи теж є ризиком витоку, адже одночасна робота кількох людей може бути незлагодженою. Наприклад, один адміністратор може впровадити або видалити правила розмежування доступу, а інший – забути внести дані прав доступу до серверів.

При використанні складних систем захисту інформації важливо грамотно розподіляти всі обов'язки і контролювати їх своєчасне виконання. В іншому випадку – створена система може нашкодити компанії.

Наприклад, можна розмежувати доступ співробітників служби безпеки до певних звітів і операцій в системі. Максимальне число повноважень надійніше довірити керівнику ІБ-служби.

Причина 6 – Поломки техніки

Помилки в роботі ПЗ

Всілякі збої в роботі програмного забезпечення виникають постійно. В момент появи уразливості захищені файли ризикують стати перехопленими хакером. Важливо вчасно виявляти всі неполадки в роботі встановлених програмних і апаратних компонентів. За працездатність і взаємодію всіх модулів захисту відповідальний адміністратор безпеки.

В результаті збою в базі даних втрачається значна кількість важливої документації. Відновлення жорстких дисків – це складне завдання, що не дає гарантії повернення втрачених відомостей.

Збої в роботі серверного обладнання

Безпечніше зберігати всю інформацію з використанням хмарних сховищ. Cloud-платформи підвищують швидкість обробки інформації. З їх допомогою кожен співробітник зможе отримати доступ до потрібних файлів з будь-якого пристрою. Система шифрування використовується віддаленим сервером, тому немає необхідності захищати канали передачі.

Збої на серверах постачальника послуг можуть траплятися через природні катаклізми або через масивні хакерські атаки. Як правило, власники хмарних платформ завжди зберігають архівовані резервні копії вмісту акантів користувачів, тому збої швидко вилучаються без втрати важливих документів.

Поломка технічних засобів захисту

Для збереження комерційної таємниці рекомендується захищати не тільки операційні системи і гаджети, але і весь периметр офісного приміщення, а також зону контролю вуличних комунікацій. Для цих цілей використовуються заглушки на вікна, ущільнювачі архітектурних конструкцій (для запобігання прослуховувань), пристрої для екранування і зашумлення (для неможливості перехоплення радіохвиль), інші гаджети.

Через поломку одного з таких пристроїв виникає канал витоку інформації, що стає доступним зловмисникові для перехоплення секретних даних.

У разі поломки комп'ютерів і інших засобів обробки даних, їх необхідно відремонтувати в сервісному центрі. Винос гаджета за межі приміщення і передача його сторонній людині (навіть якщо він не зацікавлений в отриманні службової таємниці) є можливою причиною витоку. Департамент безпеки компанії не може контролювати гаджети, поки вони знаходяться за межами фірми.

Причина 7 – Витік з технічних каналів передачі даних

Канал витоку даних – це фізичне середовище, всередині якого не контролюється поширення таємної інформації. На будь якому підприємстві, що використовує комп'ютери, серверні стійки, мережі, є канали витоку. З їх допомогою зловмисник може отримати доступ до комерційної таємниці.

Існують наступні канали витоку:

Мовний. Конкуренти часто використовують прослуховувачі та інші закладки, за допомогою яких відбувається крадіжка таємниці.

Віброакустичний. Цей канал витоку виникає в процесі зіткнення звуку з архітектурними конструкціями (стінами, підлогою, вікнами). Вібраційні хвилі можна зчитати і перевести в мовної текст. За допомогою спрямованих мікрофонів на відстані до 200 метрів від приміщення зловмисник може зчитати розмову, в якій фігурує службова інформація.

Електромагнітний. В результаті роботи всіх технічних засобів виникає магнітне поле. Між апаратними елементами передаються сигнали, що можна вважати спеціальним обладнанням на великих відстанях і отримати секретні дані.

Візуальний. Приклад появи візуального каналу крадіжки – це проведення нарад і конференцій з неприкритими вікнами. З сусіднього будинку зловмисник може легко переглянути всю інформацію. Також можливі варіанти використання відеозакладок, які передають картинку того, що відбувається конкурентам.

Для захисту технічних каналів витоку рекомендується використовувати:

– *Тепловізор. З*а допомогою такого девайса можна просканувати всі стіни і частини інтер'єру на наявність закладних пристроїв (жучків, відеокамер).

– *Пристрої, що заглушають подачу сигналу радіочастот.*

– *Засоби захисту архітектурних конструкцій* – ущільнювачі для вікон, дверей, підлоги і стелі. Вони ізолюють звук і унеможливлюють зчитування вібраційних хвиль з поверхні будівлі.

– *Пристрої для екранування і зашумлення.* Вони використовуються для захисту електромагнітного каналу витоку.

Також слід заземлити всі комунікації, що виходять за межі приміщення і

контрольованої зони (труби, кабелі, лінії зв'язку).

Існує кілька дієвих способів, що допоможуть знизити ризик витоку і розголошення інформації. Підприємство може використовувати всі методи захисту або тільки кілька з них, адже система безпеки повинна бути економічно вигідною. Збитки від втрати секретної інформації не можуть бути менше вартості впровадження та підтримки системи безпеки.

Шифрування. *Шифрування* – це простий і дієвий метод захисту комерційної таємниці. Сучасні алгоритми шифрування використовують світові стандарти в області криптографії (шифри AES, ГОСТ), двосторонній обмін ключами (з його допомогою хакер не зможе зламати шифр навіть після отримання доступу до каналу передачі), еліптичні криві для генерації захисту. Такий підхід робить злом шифрованого повідомлення неможливим для стандартних комп'ютерів.

Переваги використання шифрування з метою запобігання витоку комерційної інформації:

Простота застосування. Реалізація шифрування проводиться спеціальним ПЗ. Програма повинна бути встановлена на всі комп'ютери і мобільні пристрої, в яких циркулює секретна інформація. Роботу додатка налаштовує системний адміністратор або адміністратор безпеки. Таким чином, звичайному користувачеві автоматизованих систем не потрібно вчитися використовувати систему захисту. Всі файли шифруються і дешифруються автоматично в рамках корпоративної мережі.

У разі необхідності передачі важливих електронних документів за межі комерційної мережі вони будуть зберігатися на флеш-носії, хмарному носії або в клієнтській пошті виключно в зашифрованому вигляді. Недолік – без спеціального ПЗ працівник не зможе переглянути вміст файлу.

Високий ступінь надійності. З використанням потужних обчислювальних алгоритмів криптографії зловмисникові складно перехопити секретні повідомлення або трафік фірми, а розшифровка, без знання відкритого і закритого ключа, неможлива.

Відзначимо, що шифрування є не єдиним варіантом захисту таємниці від всіх можливих атак. Працівники здатні без проблем прочитати вміст електронних документів в рамках комерційної мережі, тому ризик несанкціонованого розголошення третім особам залишається. Використання криптографії є невід'ємною частиною функціоналу кожної комплексної системи безпеки.

Контроль персоналу

Якщо технічні засоби легко контролювати, то персонал є одним з найнебезпечніших джерел витоку. Людський фактор присутній завжди, і навіть співробітники відділу безпеки не завжди можуть встановити, від якого працівника може виходити загроза.

Як правило, пошук зловмисника серед персоналу виконується вже тоді, коли стали відомі перші випадки передачі даних конкурентам. Адміністратори безпеки перевіряють можливість перехоплення інформації технічними каналами витоку, і, якщо всі канали надійно захищені, підозра падає на працівників.

Діяльність співробітників організації контролюється за допомогою систем обліку робочого часу. Це комплексне апаратне і програмне забезпечення, що документує точний час прибуття на роботу, час догляду, діяльність персоналу за комп'ютером, записує листування корпоративної пошти, проводить відеоспостереження і передає всі ці дані керівництву фірми або керівнику відділу безпеки. Далі вся отримана інформації аналізується і виявляється число працівників, які могли поширювати комерційну таємницю.

Норми документування та передачі комерційної таємниці

Захищати треба не тільки електронні документи, а й всю друковану документацію, що містить секретні відомості. Відповідно до Закону "Про зберігання і обробку відомостей, що містять комерційну таємницю", слід виконувати такі вимоги:

– Зберігати всі документи з комерційною таємницею виключно в окремих закритих приміщеннях, що охороняються цілодобово системами відеоспостереження або охоронцями.

– Доступ до службової таємниці можуть мати тільки співробітники, яким вона потрібна в процесі роботи.

– Запис про вилучення документа з архіву вноситься до реєстраційного журналу.

Вказується точна дата, гриф документа та ініціали особи, яка здобула копію файлу. Аналогічні дії проводяться при поверненні об'єкта.

– Документ, що містить комерційну таємницю, не можна виносити за межі офісу без повідомлення про це керівника департаменту безпеки.

– Для передачі таємних документів між філіями підприємства використовується фельд'єгерська пошта – захищена кур'єрська передача документів особливої важливості.

3. Апаратні засоби захисту інформації

3.1. Основи апаратного захисту

Проблема захисту даних стала актуальною з самого початку використання обчислювальної техніки. Втрата недокументованих електронних даних спричиняла необхідність повторного виконання необхідної обробки інформації. У деяких випадках втрата вихідних даних робила неможливою повторну обробку інформації, а отже, і втрату важливих результатів.

Саме проблема захисту даних під час передачі їх між комп'ютерами(поряд із завданням збільшення швидкості їх обробки) стала поштовхомдо створення комп'ютерних мереж.

Можливо, закономірним є той факт, що саме розвиток комп'ютерних мереж став причиною надзвичайного загострення проблеми захисту даних.

Захист даних, а, отже, і захист інформації – комплексна проблема, яка є частиною національної безпеки. Необхідність вирішення проблеми захисту інформації на державному рівні викликала включення цієї проблеми до стратегії національної безпеки та прийняття Закону «Про електронний підпис». Це не останній законодавчий акт в цьому напрямі, оскільки комплексне вирішення проблеми передбачає створення єдиної правової, організаційної та матеріально-технічної бази.

На державному рівні мова йде про захист інформації державної ваги. Але і на рівні органів державного управління та місцевого самоврядування ця проблема на сьогодні досить гостра.

Коли говорять про захист даних, то мають на увазі дві основних небезпеки втрати даних: пошкодження даних і несанкціонованого доступу до них. Боротьба з обома небезпеками ведеться апаратними і програмними засобами та організаційними заходами. Нижче ми розглянемо основні засоби захисту даних, які можна використовувати у практичній діяльності.

До апаратних засобів захисту від пошкодження даних слід зарахувати використання джерел неперервного живлення, а також резервування та архівування даних.

Раптове зникнення напруги може спричинити втрату тих файлів, з якими в момент зникнення напруги працював комп'ютер. Наприклад, як операційна система Windows, так і програми з пакету MS Office під час роботи створюють багато тимчасових файлів, які постійно використовуються програмами. Раптове зникнення напруги може спричинити втрату цих файлів і, як наслідок, «розвал» системи і втрату файлів з даними. Найгіршим випадком є вихід з ладу жорсткого диска. Хоча відновлення інформації з цього диска ще здебільшого можливе, але це може бути не в повному обсязі і коштуватиме досить дорого.

Використання джерел неперервного живлення (UPS) дає можливість у разі зникнення напруги в мережі підтримувати роботу системи ще протягом 20 хв. За цей час завершиться виконання всіх програм, і комп'ютер можна буде вимкнути без втрати всіх даних. У відповідальніших випадках використовують резервне живлення або з мережі, або від автономного джерела.

Резервування інформації дає можливість у разі втрати даних повернути стан системи на момент її резервування. За такої умови резервні копії можна зберігати у захищених місцях (приміщеннях чи сейфах). Для резервування використовують або засоби постійної пам'яті (компакт-диски, стримери, переносні жорсткі диски), або жорсткі диски на іншому комп'ютері в мережі.

Для інформації невеликого обсягу цілком придатний спосіб зберігання даних у вигляді архіву на тому ж комп'ютері, на автономному носії або на іншому комп'ютері. Принципово

– це те ж резервування даних, але в трансформованому (стисненому) вигляді. Для архівування використовують програми WinZIP або WinRAR. До речі, в архівованому вигляді дані легше передавати засобами електронного зв'язку в мережі. За такої умови бажано користуватись архіватором ZIP.

До апаратних засобів захисту від несанкціонованого доступу найкраще використовувати засоби постійної пам'яті – компакт-диски і переносні жорсткі диски. У цьому разі дані будуть захищені навіть при пошкодженні чи втраті комп'ютера. У жодному разі для тривалого зберігання даних не слід використовувати дискети. Для цього їх надійність занадто низька.

3.2. Класифікація технічних засобів зняття інформації

До апаратних засобів захисту даних від несанкціонованого доступу належать також :

криптографічні плати, за допомогою яких дані можна зашифрувати, створити електронний підпис та аутентифікувати користувача;

магнітна картка SmartCard для зберігання секретного ключа та шифрування паролів;

пристрої ActiveCard для введення паролів, в яких паролі обчислюються на підставі уведених даних (динамічні паролі), та **SmartReader** для зчитування цих паролів. Чотиризначний пароль, що вводиться користувачем, перераховується в цих пристроях у спеціальний код.

До технічних засобів можна зарахувати і біометричні засоби, тобто: зчитування візерунка сітчатки ока, відбитків пальців, геометрії рук, динаміки підпису. Крім того, використовують також комплекси технічних засобів захисту від несанкціонованого доступу до даних, які можна згрупувати так:

захист від електромагнітного випромінювання, куди належать: використання оптоволоконних кабелів, захисної плівки на вікнах, захищених дисплеїв;

захист від поновлення знищених даних;

захист від підслуховування шляхом: встановлення фільтрів на лініях зв'язку, попередження встановлення підслуховувальних пристроїв, використання звукопоглинальних покриттів, протипідслуховувального зашумлення.

Бурхливий розвиток техніки, технології, інформатики в останні десятиліття викликав ще більш бурхливий розвиток технічних пристроїв і систем розвідки. Справді, занадто часто виявлялося вигідніше витратити N-у суму на добування, наприклад, технології, що вже існує, ніж у кілька разів більшу на створення власної. А в політиці чи у військовій справі виграш іноді виявляється просто безцінним.

У створення пристроїв і систем ведення розвідки вкладалися і вкладаються величезні кошти у всіх розвинених країнах. Сотні фірм багатьох країн активно працюють у цій галузі. Серійно виробляються десятки тисяч моделей «шпигунської» техніки. Ця галузь бізнесу давно і стійко зайняла своє місце в загальній системі економіки Заходу і має міцну законодавчу базу.

У західній пресі можна знайти дуже захопливі документи про існування і роботу міжнародної організації промислового шпигунства «Спейс Інкорпорейтед», а заодно і познайомитися зі спектром послуг, пропонованих цією компанією. Зокрема, англійська газета «Піпл» повідомляє, що серед клієнтів компанії є не тільки промисловці, але й організовані злочинні угруповання. Як і будь-який бізнес, коли він вигідний, торгівля секретами розширює поле діяльності, знаходячи для свого процвітання вигідний ґрунт.

Тематики розробок на ринку промислового шпигунства охоплюють практично всі сторони життя суспільства, безумовно, орієнтуючись на найбільш фінансово вигідні.

У СРСР після 1917 р. ведення комерційної розвідки знаходилося під суворим контролем держави. У Радянському Союзі в цій галузі були зосереджені чудові, якщо не сказати кращі, фахівці. Видатним досягненням було і залишиться на багато років чудо технічної розвідки – будинок посольства США в Москві, перетворене у величезне «вухо», у якому кожен подих, кожен шерех був доступний для запису й аналізу. Датчики знаходили навіть у зварених сталевих конструкціях будинку, причому по щільності матеріалу вони відповідали навколишньому металу і були недоступні для рентгенівського аналізу. Ці

системи були здатні функціонувати автономно десятки років. Американці змушені були відмовитися від використання цього будинку, навіть незважаючи на те, що колишній голова Комітету Державної Безпеки Вадим Бакатін передав їм схему побудови цієї системи.

Розпад СРСР і розвиток вільної ринкової економіки відродило попит на техніку подібного роду. Фахівці військово-промислового комплексу, що залишилися без роботи, не затрималися запропонувати свої послуги й у цій сфері. Спектр послуг широкий: від примітивних радіопередавачів до сучасних апаратно-програмних комплексів ведення розвідки. Звичайно, у нас немає ще великих фірм, що виробляють подібну техніку, немає і такого достатку моделей, як на Заході, але техніка наших виробників цілком за своїми даними порівняно з аналогічною західною іноді краще і дешевше. Зрозуміло, мова йде про порівняння техніки, яка є у відкритому продажу.

Природно, апаратура, використовувана спецслужбами (її кращі зразки) набагато перевершує по своїх можливостях техніку, використовувану комерційними організаціями. Як приклад – найменший і найдорожчий у світі радіомікрофон, габарити якого не перевищують чверті олівцевої гумки для стирання. Цей мініатюрний передавач живиться від ізотопного елемента і здатний протягом року сприймати і передавати на прийомний пристрій, розташований за півтора кілометра, розмову, що ведеться в приміщенні пошепки. Крім того, уже зараз виробляються радіозакладки, що можуть записувати перехоплену інформацію, зберігати її протягом доби чи тижня, передавати в режимі швидкодії за мілісекунду, стирати запис і починати процес знову.

Сучасні злочинні угруповання мають на озброєнні новітні технічні засоби отримання, обробки та захисту інформації. Частину таких засобів як вітчизняного (наприклад, радіокапсула Р1-Р9, приймачі RH-03, RA-05, RA-07, конвертори СО-01 та інші), так і імпортного виробництва (наприклад, лазерний мікрофон HP-150 фірми “Hewlett-Packard”) дотепер можна було придбати у відкритій торгівлі. Крім того, є багато літератури, яка дозволяє практично будь-якому радіоаматору кустарно виготовити радіомікрофони та інші технічні засоби ведення розвідувальної та контррозвідувальної діяльності.

3.3. Основні групи технічних засобів ведення розвідки

1. Радіопередавачі з мікрофоном (радіомікрофони):
- з автономним живленням;
- з живленням від телефонної лінії;
- з живленням від електромережі;
- керовані дистанційно;
- що використовують функцію вмикання по голосу;
- напівактивні;
- з накопиченням інформації і передачею в режимі швидкодії.

2. Електронні пристрої знімання акустичної інформації:
- мікрофони з проводами;
- електронні стетоскопи (що прослуховують через стіни);
- мікрофони з гострою діаграмою спрямованості;
- лазерні мікрофони;
- мікрофони з передачею через мережу 220 В;
- прослуховування через мікрофон телефонної трубки;
- гідроакустичні мікрофони.
3. Пристрої перехоплення телефонних повідомлень:
- безпосереднього підключення до телефонної лінії;
- підключення з використанням індукційних датчиків (датчикиХолу й інші);
- з використанням датчиків, розташованих усередині телефонного

апарату;

- телефонний радіотранслятор;
- перехоплення повідомлень сотового телефонного зв'язку;
- перехоплення пейджерних повідомлень;
- перехоплення факс-повідомлень;
- спеціальні багатоканальні пристрої перехоплення телефоннихповідомлень.

4. Пристрої прийому, запису, управління:
   - приймач для радіомікрофонів;
   - пристрої запису;
   - ретранслятори;
   - пристрої запису і передачі в прискореному режимі;
   - пристрої дистанційного управління.
5. Відеосистеми запису і спостереження.
6. Системи визначення місця розташування контрольованогооб'єкта.
7. Системи контролю комп'ютерів і комп'ютерних мереж.

3.4. Радіомікрофони

Радіомікрофон, як випливає з назви, це мікрофон, об'єднаний з радіо, тобто з радіоканалом передачі звукової інформації. На сьогодні немає визначеної остаточно назви цих пристроїв. Їх називають радіозакладками, радіобагами, радіокапсулами, іноді – «жуками», але все-таки найбільш точною назвою варто визнати «радіомікрофон».

Радіомікрофони є найпоширенішими технічними засобами отримання акустичної інформації. Їхня популярність пояснюється, насамперед, зручністю їхнього оперативного використання, простотою застосування (не потрібно тривалого навчання персоналу), дешевизною, дуже невеликими розмірами. У найпростішому варіанті радіомікрофон складається з власне мікрофона, тобто пристрою для перетворення звукових коливань в електричні, а також радіопередавача – пристрою, що випромінює в простір електромагнітні коливання радіодіапазону (несучу частоту), промодульовані електричними сигналами з мікрофона. Мікрофон визначає зону акустичної чутливості (звичайно вона коливається від декількох до 20–30 м), радіопередавач – дальність дії радіолінії. Визначальними параметрами з погляду дальності дії для передавача є потужність, стабільність несучої частоти, діапазон частот, вид модуляції. Істотний вплив на довжину радіоканалу має, звичайно, і тип радіоприймального пристрою.

Пристрій управління не є обов'язковим елементом радіомікрофона. Він призначений для розширення його можливостей: дистанційного вмикання-вимикання передавача, мікрофона, пристрою, що записує. Можуть бути передбачені режими: ввімкнення по голосу, режим запису в реальному часі, режим прискореного відтворення тощо.

Пристрій запису, як випливає з вищесказаного, також не є обов'язковим елементом.

Розроблено і випускаються серійно сотні моделей радіомікрофонів.

Технічні дані радіомікрофонів знаходяться в таких межах:вага від 5 до 350 г

габарити від 1 $см^3$ до 8 $дм^3$

частотний діапазон від 27 до 900 МГц

потужність від 0,2 до 500 МВт

дальність без ретранслятора від 10 до 1 500 м

час безупинної роботи від декількох годин до декількохроків.

Дальність дії, габарити і час безупинної роботи дуже залежать один від іншого. Справді, для збільшення дальності необхідно, насамперед, підвищити потужність, одночасно зростає струм, споживаний від джерела живлення, що швидше витрачає свій ресурс, а виходить, скорочується час безупинної роботи. Щоб збільшити цей час, збільшують ємність батарей живлення, але це збільшує габарити радіомікрофона. Можна збільшити тривалість роботи передавача введенням у його склад пристрою дистанційного управління (вмикання-вимикання), однак це також збільшує габарити. Крім того, потрібно мати на увазі, що

збільшення потужності передавача полегшує можливість його виявлення.

Одним з перспективних напрямків збільшення скритності і часу ефективного використання є застосування дистанційного ввімкнення. Прикладами є вироби TRM-1530 і TRM-1532. Це радіомікрофони з живленням від батарей, габаритами 87х54х70 мм, вагою близько 100 г, ізЧМ передавачем діапазону 380–400 МГц або 100–150 МГц і дальністю до 300 м. Дистанційне вмикання-вимикання дозволяє довести час ефективної роботи виробу до 1 року за умови безупинної роботи 280-300 год. Подібна апаратура, але більш великих габаритів, починає надходити в продаж і від вітчизняних виробників.

Дуже перспективним є застосування радіомікрофонів з активацією від звуку – музики, мови і т.д., наприклад, модель STG-4001: включення увімкнення пристрою відбувається від звуку, вимикання – автоматично через 5 с після зникнення звуку. Застосування функції ввімкненняпо голосу дозволило довести час ефективної роботи до 300 год. Прилад має дуже прийнятні розміри – 20х38х12 мм, вага з батареями – 18 г забезпечує дальність до 500 м, частоти – 130–150 МГц. Варто підкреслити, що такі радіомікрофони досить важко знайти.

У складних випадках можлива побудова системи передавачів. Наприклад, під час руху об'єкта по шляху проходження заздалегідь розміщуються радіомікрофони, що працюють на різних частотах. Спостереження ведеться за допомогою багатоканального приймача. Можлива побудова схеми з використанням передавача-ретранслятора. Потужність радіомікрофона робиться дуже невеликою (для збільшення часу роботи і підвищення скритності), а на невеликій відстані, наприклад, у сусідньому приміщенні, установлюється передавач-ретранслятор, габарити і потужність якого піддаються набагато меншим обмеженням.

Як уже зазначалося вище, дальність дії радіопередавачів визначається якостями радіоприймальних пристроїв, насамперед, чутливістю. Як приймачі часто використовують побутові радіоприймальні пристрої. У цьому разі кращим є застосування магнітол, тому що з'являється можливість одночасного ведення запису. До недоліків таких пристроїв належать низька чутливість і можливість настроювання сторонніх осіб на частоту передавача. Частково ці недоліки можна усунути перебудовою частотного діапазону, у тому числі за допомогою конверторів, а також переналагодженням підсилювачів для підвищення чутливості. Достоїнством таких систем є низька вартість, а також те, що вони не викликають підозр. Але все-таки кращим варто вважати застосування спеціальних прийомних пристроїв.

3.5. Основні методи прослуховування телефонних ліній

Цінність інформації, переданої по телефонних лініях, а також існуюче переконання про масовий характер такого прослуховування викликає найбільше занепокоєння в організацій і приватних осіб за збереження конфіденційності своїх переговорів саме по телефонних каналах. Длязахисту своїх секретів необхідно знати методи, за допомогою яких можуть бути здійснені операції по перехопленню. Але при цьому потрібно врахувати, що організація масового прослуховування (в існуванні якої переконані дуже багато людей) неможлива з причин технічного і фінансового характеру. Справді, для аналізу записаних повідомлень потрібно тримати чималу кількість людей і техніки. Крім того, для організації прослуховування в даний час потрібна санкція суду. Більш ймовірна організація прослуховування без санкції, у комерційних чи інших цілях. Ймовірність витоку інформації по телефонних каналах становить від 5 до 20 %.

В цей час на ринку СНД є сотні типів пристроїв перехоплення телефонних повідомлень як вітчизняного, так і імпортного виробництва.

Можна виділити шість основних зон прослуховування:

- телефонний апарат;
- лінія від телефонного апарата, включаючи розподільну коробку;
- кабельна зона;
- зона АТС;
- зона багатоканального кабелю;
- зона радіоканалу.

Найбільш ймовірна організація прослуховування перших трьох зон,бо саме в цих зонах

найлегше підключитися до телефонної лінії. Фахівці, що займаються захистом інформації, підтверджують, що найчастіше використовується прослуховування за допомогою рівнобіжного апарата. Здебільшого для цього навіть не потрібно прокладати додаткові проводи телефонна мережа настільки заплутана, що завжди є лінії, які не використовуються. Крім того, неважко підключитися в парадній до розподільної коробки. Підключення в третій зоні менш поширено, бо потрібно проникати в систему телефонних комунікацій, що складається з труб із прокладеними усередині них кабелями, а також розібратися в цій системі і визначити потрібну пару серед сотень інших. Однак не слід вважати, що це нездійсненне завдання, оскільки існує вже необхідна для цього апаратура. Як приклад можна навести американську систему «Кріт». За допомогою спеціального індуктивного датчика, що охоплює кабель, знімається передана по ньому інформація. Для установки датчика на кабель використовуються колодязі, через які проходить кабель. Датчик у колодязі закріплюється на кабелі і для утруднення виявлення проштовхується в трубу, що підводить. Сигнал записується на диск спеціального магнітофона. Після заповнення диска видається сигнал, і агент за сприятливих умов заміняє диск. Апарат може записувати інформацію, передану одночасно по 60 каналах. Тривалість безупинного запису становить 115 год. Такі пристрої знаходили в Москві. Для різних типів підземних кабелів розроблені різні датчики: для симетричних високочастотних – індуктивні для відводу енергії з коаксіальних кабелів, для кабелів з надлишковим тиском – пристрої, що виключають його зниження. Деякі прилади забезпечуються радіопередавачем для передачі записаних повідомлень чи перехоплення їх у реальному масштабі часу.

У технічному плані найпростішим способом є контактне підключення.

Можливо тимчасове підключення до абонентської проводки за допомогою стандартної «монтерської трубки». Однак підключення такого типу легко виявляється за допомогою найпростіших засобів контролю напруги телефонної мережі. Зменшити ефект спадання напруги можна підключенням слухавки через резистор 0,6–1кОм. Підключення здійснюється за допомогою дуже тонких голочок і тонких проводів, що прокладаються в якій-небудь існуючій чи виготовленій щілині. Щілина може бути зашпакльована і пофарбована так, що візуально визначити підключення дуже важко.

Відомий спосіб підключення до ліній зв’язку апаратури з компенсацією спадання напруги. Істотними недоліками контактного способу підключення є порушення цілісності проводів і вплив підключеного пристрою на характеристики ліній зв’язку. З метою усунення цього недоліку застосовується індуктивний датчик, виконаний у вигляді трансформатора. Існують також датчики, принцип роботи яких заснований наефекті Холу.

Вартість подібних пристроїв коливається від 20 до 250 $. Як записувальні пристрої застосовуються стандартні диктофони, спеціальні мініатюрні типу OLIMPUS L-400, а також стаціонарні багатоканальні диктофони, наприклад АД-25-1. Як правило, схема прослуховування організована так, що магнітофон включається з появою сигналу в лінії.

Як приклад мініатюрного магнітофона можна навести модель N2502, рекламовану як магнітофон, що неможливо знайти за допомогою сучасних детекторів записувальної техніки. У цьому магнітофоні є гнізда для підключення зовнішнього мікрофона, пульта дистанційного управління і головних телефонів.

Як правило, спеціальні багатоканальні магнітофони для запису телефонних переговорів використовуються в складі спеціальної апаратури для контролю особливо режимних робіт. У цьому разі використовуються спеціальні прийоми, що дозволяють по ключових словах переривати чи записувати телефонну розмову. Такий акустичний контроль може бути організований за допомогою наявних на ринках США, Німеччини і Японії спеціальних багатоканальних магнітофонів, призначених для стаціонарного запису телефонних переговорів і розрахованих на значну (від 10 до 100) кількість каналів.

3.6. Системи прослуховування повідомлень, переданих по стільникових каналах

Стільниковою називається система зв’язку, що складається з деякої кількості осередків, що, з’єднуючись між собою, утворять мережу, соту. Кожен осередок може працювати з

визначеною кількістю абонентів одночасно. Стільникові мережі мають можливість нарощування, а також можуть стикуватися одна з одною. Радіус дії базової стільникової станції становить 5–15 км, а перехоплення повідомлень у цьому разі може проводитися на відстані до 50 км. Як приклад реалізації подібної системи можна навести стільникові системи спостереження Cellmate-lOB і Cellscan.

Cellmate-lOB контролює одночасно до 10 телефонних номерів, тобто один осередок стільникового зв'язку. Є можливість програмувального перебору осередків. Потрібна розмова може визначатися по голосу абонента чи по змісту розмови. Перехоплені один раз номера за бажанням переводяться програмою в особливий режим спостереження. Вбудований запам'ятовуючий пристрій запам'ятовує останні параметри настроювання. Запис починається автоматично, коли об'єкт спостереження починає користуватися телефоном. Інформація про номери телефонів, параметри настроювання, ідентифікації по голосу зчитується з кольорового рідкокристалічного дисплея, так само визначаються коди доступу.

Система Cellscan аналогічна за функціями Cellmate-lOB і також розміщується в аташе-кейсі. Вважається, що кількість програмувальних номерів не обмежена. У режимі сканування на дисплей виводиться інформація про 895 каналів. Спостерігається вся телефонна система, і вибираються канали, по яких відбуваються дзвінки. За допомогою комплекту стільникових карт визначається район, у якому відбувається підозріла розмова, ідентифікований сканером по змісту чи голосу. Можна відключити канали, на яких ви не хочете здійснювати перехоплення. Використовується дороблений стільниковий телефон OKI, що може застосовуватися і як звичайний стільниковий телефон, вага системи – 9 кг.

Сучасні системи стільникового зв'язку можуть використовувати різні системи кодування і(чи) перебудову частоти по випадковому принципу. Існують і спеціальні комплекти радіоперехоплення з можливістю аналізу зашифрованих повідомлень, наприклад, Sigint/Comint Spektra фірми Hollandes Signal, але подібна апаратура дуже дорога. У Росії розроблені і пропонуються програмно-апаратні системи перехоплення пейджингових повідомлень. До складу подібної системи входять дороблений сканер (AR-3000A, IC-7100 і ін.), пристрій перетворення, комп'ютер і спеціальне програмне забезпечення. Система дозволяє здійснюватиприйом і декодування текстових і цифрових повідомлень, переданих у системі радіопейджингового зв'язку і зберігати всі прийняті повідомлення (з датою і часом передачі) на твердому диску персонального комп'ютера. При цьому може здійснюватися фільтрація потоку повідомлень, виділення даних, адресованих конкретним абонентам.

Перехоплення факсів-повідомлень принципово не відрізняється від перехоплення телефонних повідомлень.

Наостанок наведемо приклад організації прослуховування Агентством національної безпеки США, що має в 6 разів більше службовців, ніж ЦРУ. 4 120 могутніх центрів прослуховування на базах у Німеччині, Туреччині, Японії і т.д., а також на кораблях, підводних човнах, літаках і супутниках збирають і аналізують майже всю інформацію, передану електронним способом, включаючи випромінювання систем сигналізації автомобілів, квартир і так далі.

3.7. Використання телефонної лінії для прослуховування приміщень

Телефонна лінія використовується не тільки для передачі телефонних повідомлень, але і для прослуховування приміщення. Щоб ввімкнути подібний пристрій, потрібно набрати номер абонента. Перші два гудки «ігноруються» пристроєм, тобто телефон не дзвонить. Після цього необхідно покласти слухавку і через визначений час (30–60 с) подзвонити знову. Тільки після цього система включається в режим прослуховування.

Аналогічно працюють, наприклад, пристрої ST-01 ELSY, UM103. Ціна таких пристроїв – від 15 (вітчизняні) до 250 $ (закордонні). Як приклад одного з таких пристроїв є пристрій БОКС-Т. Цей пристрій дозволяє контролювати приміщення з будь-якої точки земної кулі по телефону. Для цього досить набрати номер телефону, де вже встановлений прилад «Бокс-Т», і включити мікрофон. Для вимикання досить покласти слухавку.

Необхідно мати на увазі, що існують такі системи передачі акустичної інформації з

телефонних ліній, що дозволяють прослуховувати приміщення без установки якого-небудь додаткового устаткування. Також використовуються недоліки конструкції телефонного апарата: акустичні коливання впливають на якір дзвоника, що, коливаючись, викликає появу в котушці мікрострумів модульованих дросельною системою, що наводиться в котушці та у цьому разі може досягати декількох мілівольтів. Дальність цієї системи не перевищує (через загасання) декількох десятків метрів. Прийом здійснюється на якісний підсилювач низької частоти.

Другий варіант системи пов'язаний з реалізацією ефекту «нав'язування». Коливання частотою від 150 кГц і вище подаються на один провід телефонної лінії, до другого проводу приєднується приймач. Земля передавача і приймача з'єднані між собою або із загальною землею, наприклад, водогінною мережею.

Через елементи схеми телефонного апарата високочастотні коливання надходять на мікрофон, навіть якщо він відключений від мережі, і модулюються мовою. Детектор приймача виділяє мовну інформацію. Через істотне загасання ВЧ сигналу у двопроводовій лінії дальність також не перевищує декількох десятків метрів (без ретранслятора).

3.8. Спеціальні пристрої прослуховування

Спрямовані мікрофони

Звичайні мікрофони здатні реєструвати людську мову на відстані, що не перевищує декількох десятків метрів. Для збільшення дистанції, на якій можна робити прослуховування, практикують застосування спрямованого мікрофона.

Іншими словами, цей пристрій збирає звуки тільки з одного напрямку, тобто має вузьку діаграму спрямованості. Такі пристрої широко застосовуються не тільки в розвідці, але і журналістами, мисливцями, рятувальниками і т.д.

Можна виділити два основних типи спрямованих мікрофонів:

- з параболічним відбивачем;
- резонансний мікрофон.

Мікрофон з параболічним відбивачем

У мікрофоні з параболічним відбивачем власне мікрофон розташований у фокусі параболічного відбивача звуку.

Спрямований параболічний мікрофон з підсилювачем AD-9 концентрує звуки, що йдуть, і підсилює їх. Простий у використанні і настроюванні. У комплект входить мікрофон, підсилювач, кабель і головні телефони. Електроживлення – від батареї 9 В.

Випускаються кілька моделей. Загальним у конструкції всіх цих мікрофонів є наявність рукоятки пістолетного типу, параболічного відбивача діаметром близько 40 мм і підсилювача. Діапазон сприйманих частот становить від 100–250 Гц до 15–18 кГц. Усі мікрофони мають автономне живлення і мають рознімання для підключення до магнітофона. Гостра «голчаста» діаграма спрямованості дозволяє за відсутності перешкод контролювати людську мову на відстані до 1200 м. У реальних умовах (в умовах міста) можна розраховувати на дальність до 100 м.

Резонансний мікрофон

Резонансний мікрофон заснований на використанні явища резонансу в металевих трубках різної довжини. Наприклад, в одній з модифікацій такого мікрофона використовується набір з 37 трубок довжиною від 1 до 92 см.

Звукові хвилі, що приходять до приймача по осьовому напрямку, приходять до мікрофона в однаковій фазі, а з бічних напрямків (через відмінну швидкість поширення звукових хвиль у металі, а також різної довжини трубок) – виявляються зрушеними по фазі.

З погляду схованого контролю звуку застосування спрямованих мікрофонів утруднено через найчастіше неприйнятні їхні габарити і джерела акустичних перешкод. Крім того, для того щоб не бути прослуханим в автомобілі, досить просто підняти скло.

Лазерні мікрофони

У разі, якщо ви підняли скло чи закрили кватирку в автомобілі, може бути використаний лазерний мікрофон. Перші їхні зразки були прийняті на озброєння американськими спецслужбами ще в 60-ті роки.

Як приклад розглянемо лазерний мікрофон HP-150 фірми Hewlett-

Packard з дальністю дії до 1000 м. Він сконструйований на основі гелійнеонового чи напівпровідникового лазера з довжиною хвилі 0,63 мкм (тобто у видимому діапазоні; сучасні пристрої використовують невидимий ІЧ діапазон).

Промінь лазера, відбитий від скла приміщення, у якому ведуться переговори, виявляється промодульованим звуковою частотою. Прийнятий фотоприймачем відбитий промінь детектується, звук підсилюється і записується. Приймач і передавач виконані роздільно, є блок компенсації перешкод. Вся апаратура розміщена в кейсі і має автономне живлення. Подібні системи мають дуже високу вартість (більше ніж 10 тис.

$) і, крім того, вимагають спеціального навчання персоналу і використання комп'ютерної обробки мови для збільшення дальності.

Існує досвідчена вітчизняна система ЛСТ-ЛА2 з дальністю знімання менше 100 м і досить скромною вартістю. Слід зазначити, що ефективність застосування такої системи зростає зі зменшенням освітленостіоперативного простору.

Гідроакустичні датчики

Звукові хвилі поширюються у воді з дуже невеликим загасанням.

Гідроакустики військово-морських сил навчилися прослуховувати шепіт у підводних човнах, що знаходяться на глибині десятків метрів. Цей же принцип можна застосовувати, використовуючи рідину, що знаходиться в системах водопостачання і каналізації. Таку інформацію можна перехоплювати в межах будинку, але радіус прослуховування буде дуже сильно залежати від рівня шумів, особливо у водопроводі. Переважно використовувати датчик, встановлений у батареї опалення. Ще більш ефективним буде використання гідроакустичного передавача, встановленого в батареї приміщення, що прослуховується.

Надвисокочастотні та інфрачервоні передавачі

Для підвищення скритності в останні роки стали використовувати інфрачервоний канал. Як передавач звуку від мікрофона використовується напівпровідниковий лазер. Наприклад, розглянемо пристрій TRM1830. Дальність дії вдень становить 150 м, уночі – 400 м, час безупинної роботи – 20 год. Габарити не перевищують 26 х 22 х 20 мм. До недоліків інфрачервоного (ІЧ) каналу подібної системи можна віднести необхідність прямої видимості між передавачем і приймачем і вплив перешкод.

Підвищити скритність одержання інформації можна також за допомогою використання каналу надвисокочастотного (НВЧ) -діапазону - більше 10 гГц. Передавач, виконаний на діоді Ганна, може мати дуже невеликі габарити. Забезпечується дальність більше 100 м. До переваг такої системи можна віднести відсутність перешкод, простоту і відсутність у цей час ефективних засобів контролю. До недоліків варто віднести необхідність прямої видимості, хоча й у меншому ступені, тому що НВЧ-сигнал може все-таки обгинати невеликі перешкоди і проходить (з ослабленням) крізь тонкі діелектрики, наприклад, штори на вікнах.

Стетоскопи

Стетоскоп являє собою вібродатчик, підсилювач і вібродатчик спеціальною мастикою прикріплюється до стіни, стелі і тому подібне. Розміри датчика, на прикладі пристрою DTI, становлять 2,2 х 0,8 см, діапазон частот – 300–3000 Гц, вага – 126 г, коефіцієнт підсилення – 20 000.

За допомогою подібних пристроїв можна здійснювати прослуховування розмови через стіни товщиною до 1 м. Стетоскоп може оснащуватися проводовим, радіо чи іншим каналом передачі інформації. Основною перевагою стетоскопа можна вважати труднощі виявлення, томущо він може встановлюватися в сусідніх приміщеннях.

Існують стетоскопи, у яких чуттєвий елемент, підсилювач і радіопередавач об'єднані в одному корпусі. Вони мають дуже невеликі габарити і дозволяють не тільки прослуховувати

розмови через стіни, віконні рами, двері, але і передавати інформацію по радіоканалу. Мають високу чутливість і забезпечують гарну розбірливість мовного сигналу. Як правило, робоча частота становить 470 МГц. Дальність передачі – до 100 м. Час безупинної роботи – 24 год, розміри – 40 х 93 мм.

Більшість фахівців прогнозують постійне збільшення випадків застосування стетоскопів, що насамперед пояснюється зручністю застосування подібної техніки, а також тим, що їх надзвичайно важко знайти.

3.9. Системи і пристрої відеоконтролю

Системи і пристрої відеоконтролю одержали могутній імпульс свого розвитку в зв'язку зі створенням мініатюрних відеокамер і відеомагнітофонів. Якщо історія застосування фотокамер у розвідці нараховує 90–100 років, то застосування відеотехніки стримувалося неприйнятними її ваговими та габаритними характеристиками. На сьогодні габарити відеокамер (без відеомагнітофонів) часто можуть бути менше, ніж габарити наймініатюрніших фотокамер. Тим часом застосування відеотехніки в розвідці часто має переваги, недосяжні за допомогою фотоі кінотехніки. Насамперед, це те, що за допомогою відеотехніки легко здійснити запис, передачу на великі відстані й оперативний аналіз зорової і звукової інформації в реальному масштабі часу.

У спрощеному вигляді система відеоспостереження складається з відеокамери, відеомагнітофона і(чи) передавача.

Мікровідеокамери

Відеокамери, що застосовуються для оперативної діяльності, мають, в основному, імпортне походження. Вітчизняні камери по габаритних характеристиках придатні поки тільки для систем відеоконтролю (службові приміщення, відеодомофони, магазини і так далі). Зупинимося більш докладно на характеристиках деяких конкретних моделей камер закордонного виробництва. Найбільший інтерес у цьому разі має опис безкорпусних відеокамер.

Об'єктив і електронна схема управління розміщаються на однійплаті розміром 4,2 х 4,2 см. Стандартний об'єктив має фокусну відстань 3,6 мм. З цим об'єктивом камера має габарити 4,2 х 4,2 х 2,1 см і кут огляду 92°. З точечним об'єктивом габарити становлять 4,2 х 4,2 х 1,2 см, кут огляду – 88°.

Незалежно від типу об'єктива, камера має такі характеристики: мінімальна освітленість – 0,4 лм, розподільна здатність – 380 ліній, живлення – 12 В, вага – 12 г.

Ці і подібні камери можуть монтуватися як разом з об'єктивом, так і з винесеним об'єктивом. Маскування може бути всіляким: у розетках електроживлення, радіоприймачах, настінному і настільному годинниках, одязі, окулярах, датчиках пожежної сигналізації, приладах освітлення і так далі.

Відеокамери можуть забезпечуватися різними змінними об'єктивами. Потрібно мати на увазі, що деякі матеріали, застосовувані для маскування об'єктів (типу «чорне скло»), пропускають тільки невелику частину спектра і можуть успішно працювати при сонячному освітленні, при освітленні ІЧ прожектором або звичайними лампами накалювання, але їхнє застосування неможливе при освітленні об'єкта люмінесцентними чи галогенними лампами.

Становить інтерес опис відеокамери з передавачем. Наприклад, це може бути OVS-25-5.

Розподільна здатність цієї камери – 380 ліній. Чутливість – 0,5 лм, об'єктив з фокусною відстанню 1,6 мм і автоматичним регулюванням діафрагми. Вбудований передавач працює в діапазоні 400–500 МГц і має потужність 40 МВт. Живлення – 12 В, споживаний струм – 120 мА. Габарити – 3,8 х 4,5 х 5,9 см, вага – 120 г.

3.10. Пристрої дистанційного управління, детектор руху

Пристрій управління служить для наведення камери на заданий об'єкт, умикання-вимикання передавача, відеомагнітофона, інфрачервоного освітлювача. У найпростішому

випадку цей пристрій задає швидкість і кут сканування. Розглянемо поворотний пристрій OVS-32, управління яким здійснюється за допомогою пульта.

Спеціальний поворотний пристрій для відеокамер має такі можливості:

– Кут повороту автоматичного сканування становить 180° чи задається за допомогою пульта управління.

– Плавна і безшумна робота поворотного механізму.

– Можливість кріплення на стіні за допомогою спеціальногокронштейна.

Максимальне навантаження – 7 кг, габаритні розміри – 146 х 124мм, вага – 1,4 кг.

Відеодетектор руху служить для активізації апаратури при зміні положення на об'єкті, що спостерігається.

Інфрачервоні освітлювачі (ІЧ)

Застосування ІЧ-освітлювачів буває необхідно під час роботи в умовах недостатньої видимості, а також, якщо для маскування об'єктива відеокамери застосовані непрозорі у видимому діапазоні матеріали. Інфрачервоні освітлювачі можуть випускатися або окремо, або вбудованими в відеокамери. Як приклад ІЧ-освітлювача з камерою можна навести виріб фірми SANYO – VDC-9212. Ця чорно-біла відеокамера може працювати в повній темряві.

Розподільна здатність – 400 ліній, габарити – 10 х 5 см. Окремо виконаний ІЧ-освітлювач використовує випромінюючий елемент на основі галієво-алюмінієвого арсеніду (GaALAs) зі спектром випромінювання в межах 880 нм, поміщений в алюмінієвий корпус. Споживаний струм – 600 мА. Габарити – 10 х 5 х 4,5 см.

Мініатюрні відеомагнітофони

Найпоширенішим режимом відеоспостереження є режим з одночасним записом на відеомагнітофон. Найбільш широко для цих цілей застосовуються відеомагнітофони, розраховані на роботу з 8 мм відеокасетою. Відеомагнітофони мають, як правило, функцію дистанційного управління, звук записується в режимі стерео. Деякі моделі оснащені відеомонітором. Наведемо характеристики деяких моделей таких відеомагнітофонів. OVS-9 має дві швидкості запису, час запису – до 5 год. Живлення здійснюється від убудованого акумулятора чи зовнішнього джерела живлення напругою 7,5 В. Споживана потужність – 4 Вт, габарити – 148 х 130 х 62 мм, вага – 670 г.

Модель OVS-9-1 являє собою відеомагнітофон OVS-9 з убудованим кольоровим плоским монітором. Монітор зручний під час монтажу і настроюванні схованих відеокамер, а також для контролю діючих відеосистем і перегляду знятого відеоматеріалу. Габарити – 150 х 135 х 70 мм, вага – 700 г.

Бездротові лінії передачі і прийому відеоінформації. Узагальнена схема бездротової лінії передачі і прийому відеоінформації складається з блоків приймача і передавача, лінії передачі-прийому відеозображенняі звуку.

Робоча частота комплекту моделі WVL-90 становить від 904 до 928 МГц. Лінія в змозі передавати кольорове чи чорно-біле зображення на відстань від 300 до 900 м, залежно від типу використовуваної антени (вбудована плоска чи зовнішня антена високого посилення типу WVLA902), відношення сигнал/шум не менше ніж 45 дб. Живлення від зовнішнього джерела живлення – 10–25 В, споживаний струм передавача – не менше ніж 50 мА, приймача – не менше ніж 20 мА. Габарити передавача – 23 х 6,3 х 9,5 см, приймача – 23 х 70 х 12 см.

3.11. Системи та засоби виявлення, пошуку та знешкоджування технічних засобів зняття інформації

Одним зі шляхів боротьби з несанкціонованим доступом до каналів передачі інформації є заміна аналогових каналів на цифрові. Цифрові канали передачі даних більш надійні, ніж традиційні, аналогові. Цифрові канали легше захистити за допомогою різноманітних радіоелектронних засобів, наприклад за допомогою комп'ютерних засобів. Такими захищеними цифровими каналами можуть бути і телефон, і радіо.

Найбільший рівень скритності і перешкодозахищеності мають волоконо-оптичні лінії зв'язку. Навмисне перекручування і потаємне перехоплення інформації, переданої по цих лініях зв'язку, в цей час практично неможливі. Крім того, ці канали зв'язку мають дозволяють передавати дуже великі об'єми даних.

Звичайно, для охорони інформації варто використовувати і традиційні засоби – сейфи, коди і так далі. Приміщення доцільно періодично перевіряти на наявність «жучків» й іншої шпигунсько-розвідувальної

«живності». У кімнатах не повинно бути жодних сторонніх електронних приладів, а в переговорних – узагалі жодних. На вікнах – важкі і щільні штори, на склі – спеціальні плівки. Переговорні кімнати – без вікон. Похмура картина? Але така мінімальна плата за обмеження чужого доступу до своїх секретів. Інформація часто того варта!

Доцільно, і навіть необхідно, використовувати відповідні випромінювачі перешкод як в акустичних, так і в радіодіапазонах.

Випромінювачі акустичних перешкод – це, наприклад, різні вібратори, встановлювані в місцях можливого розміщення електронних «жучків».

Перешкоди в радіодіапазонах – це спеціальні передавачі, що передають сигнали перешкод у широкій смузі частот. При цьому використовуються як АМ-, так і ЧМ-сигнали.

Подібні засоби оборони, наприклад, випромінювачі перешкод, не менш різноманітні, ніж засоби нападу.

І, звичайно, для виявлення схованих радіопередавальних засобів гарні різні детектори і сканери, що дозволяють виявити активні пристрої.

Дуже ефективні нелінійні локатори. Ці складні і дорогі апарати дозволяють виявляти сховані електронні пристрої. Принцип роботи нелінійних локаторів заснований на ефекті генерації напівпровідниковими p-n-переходами власних ВЧ-коливань при їхньому опроміненні зовнішніми ВЧ-коливаннями, вироблюваними цими пошуковими засобами. Тобто подібно традиційному локатору нелінійний локатор спочатку випромінює згенеровану послідовність високочастотних (ВЧ) -коливань у напрямку можливого розташування схованої електронної схеми, потім

«слухає». Опромінені p-n-переходи випромінюють ВЧ-коливання і тим самим видають наявність схеми, навіть неактивної! Залишається нагадати, що будь-який електронний пристрій містить мікросхеми, транзистори, діоди, а отже, і p-n-переходи. Сучасні нелінійні локатори дозволяють шукати напівпровідникові елементи навіть у залізобетонній стіні на глибині в кілька десятків сантиметрів!

З огляду на роль комп'ютерів як сучасних інформаційних центрів для збереження, обробки і передачі/прийому інформації необхідно відповідним чином захищати їх від несанкціонованого доступу до конфіденційної інформації.

Для захисту інформації в комп'ютерах, як мінімум, необхідно використовувати паролі для доступу до системи. Звичайно, це не дає 100 % гарантії, але ускладнює доступ. Існують різні апаратні і програмні засоби для обмеження доступу до комп'ютера і кодування інформації, що там знаходиться. Правда, і це не забезпечує повної гарантії, але значно підвищує рівень захисту. Часто такого рівня буває досить. При «розкритті» пароля зловмиснику потрібен час, іноді значний. Можливо, йому будуть потрібні програмні й апаратні засоби. Усе це утрудняє його діяльність. Очевидно, що тривалий час «розкриття» і необхідність використання апаратно-програмних засобів не тільки охороняють секрети, але і заважають збереженню інкогніто викрадача.

Але при обмеженні доступу до конфіденційної комп'ютерної інформації за допомогою стандартних, вбудованих засобів захисту необхідно виявляти визначену обережність. Так, наприклад, нерідко доступ до комп'ютерної інформації обмежують за допомогою пароля, що встановлюють у системному біосі. Введення цього пароля необхідний відразу після ввімкнення комп'ютера. Однак забрати цей пароль можна скиданням системного біоса за допомогою короткого замикання на кілька секунд батарейки на материнській платі. Правда, доступ до материнської плати можливий тільки після розкриття системного блоку, для чого, як правило, досить однієї викрутки. Звичайно, замикання батарейки призведе до очищення всіх системних установок, але повернути їх зовсім неважко. Однак можливий і інший шлях

вирішення проблеми пароля, невідомого для порушника. Дійсно, нерідко існують так звані технологічні паролі. Введення одного з таких паролів замість невідомого дозволяє одержати безперешкодний доступ у систему комп'ютера. Так, наприклад, для комп'ютерів з BIOS фірми Award, зокрема Award Modular BIOS v4.50, один з таких паролів – послідовність J256. І, до речі, це не єдиний пароль! Таких «паролів-всюдиходів» існує цілий список, доступний через комп'ютерні мережі широкому колу комп'ютерних ентузіастів і хакерів. Імовірно, маються аналогічні паролі і для BIOS інших фірм. Отже, для надійного захисту комп'ютера краще скористатися спеціалізованими апаратно-програмними засобами обмеження доступу до конфіденційної інформації.

Як уже зазначалося, комп'ютер у процесі роботи випромінює радіохвилі в широкому частотному діапазоні. І інформацію, що несуть ці радіохвилі, можна перехопити і розшифрувати. Однак рівень радіовипромінювання комп'ютера в радіодіапазонах: СХ-, КХ-, УКХ-, телевізійних і т.д. можна значно зменшити. Це досягається виконанням низки досить простих організаційних заходів.

Для ослаблення радіовипромінювання комп'ютерів доцільно, як мінімум, використовувати комп'ютери, вузли й елементи, сертифіковані на рівень радіовипромінювання. Особливо це стосується комп'ютерних корпусів, що забезпечують значне ослаблення радіовипромінювання. Як правило, комп'ютери відомих брендових фірм і, звичайно, їхні корпуси й елементи відповідають досить сталим нормам на супутнє роботі комп'ютера радіовипромінювання. Ці норми встановлюють міжнародні стандарти.

З огляду на те, що частина радіовипромінювання відбувається через електропроводи мережі живлення, підключення системних блоків і моніторів комп'ютерів доцільно робити через спеціальні пристрої безперебійного живлення (ПБЖ) чи, як мінімум, через подовжувачів типу Pilot. ПБЖ не тільки здійснює захист даних за рахунок короткочасної підтримки енергоживлення у разі його порушення в електромережі, але і завдяки особливостям своєї конструкції забезпечує значне ослаблення радіовипромінювання комп'ютера через електромережу. І ПБЖ, і Pilot мають у своєму складі спеціальні фільтри високочастотних перешкод – ВЧ-фільтри. Ці фільтри не тільки знижують рівень радіовипромінювання комп'ютера через проводи силової електромережі, але і підвищують стійкість роботи комп'ютера за рахунок ослаблення ВЧ-перешкод і коротких імпульсів струму, які надходять з електромережі в комп'ютер.

ВЧ-фільтри для радіоапаратури прості, але досить ефективні, можуть бути виготовлені самостійно.

Слід зазначити, що досить часто кабелі вже мають убудовані фільтри. Однак додаткові фільтри зменшують рівень радіовипромінювань через мережу електроживлення і збільшують ступінь захисту. При користуванні комп'ютерами доцільно застосовувати захисні екрани із заземленням. Дійсно, скло таких екранів має дуже тонкий електропровідний шар, який через спеціальний електропровід підключається до «землі», звичайно через корпус комп'ютера. Цей електропровідний шар є гарним екраном для електростатичного і перемінного електромагнітного полів. Отже, використання подібних захисних екранів істотно підвищує і рівень захисту інформації від несанкціонованого доступу, здійснюваного дистанційно за допомогою різноманітних радіозасобів.

Ефективність використаних засобів можна оцінити за рівнем випромінювання комп'ютера. Як такі засоби доцільно застосовувати спеціалізовані засоби. Однак у суто пізнавальних, навчальних, цілях можуть бути використані і згадані телевізор з антеною й антенним підсилювачем.

Отже, *послабити радіовипромінювання комп'ютера можна суттєво, але неможливо його виключити цілком.*

Подальше підвищення ступеня захисту досягається через застосування спеціальних апаратно-програмних засобів, наприклад тих, що змінюють по спеціальних алгоритмах порядок виведених рядків на екран монітора під час формування зображення. Для користувача комп'ютера, що знаходиться за таким монітором, функціонування цих засобів непомітно і, звичайно, не заважає роботі. Для зловмисника, що перехопив радіовипромінювання, але не знає відповідного коду, інформація залишається недоступною.

Тут під кодом розуміється послідовність виведення комп'ютерних даних на екран монітора. Підбір коду можливий, але вимагає часу, сил, устаткування і, звичайно, засобів.

Погіршити розшифровку інформації з радіовипромінювання системних блоків комп'ютерів і підключених до них моніторів можна і за допомогою їхнього раціонального розміщення. Очевидно, що компактне розміщення декількох комп'ютерів утрудняє їхню дистанційну локалізацію і селекцію комп'ютерної інформації. Спільне випромінювання групи комп'ютерів створює взаємний ефект, що маскує. До того ж, чим більш ідентичні спектри випромінювань комп'ютерів і чим компактніше вони розташовані, тим вище цей ефект. Отже, для захисту інформації доцільно, якщо можна, використовувати однакові компактно розміщені комп'ютери – системні блоки і монітори. Слід зазначити, що різні монітори і системні блоки комп'ютерів не забезпечують такого взаємного ефекту, що маскує, як однакові пристрої. Дійсно, з хору значно легше виділити голос, що відрізняється від інших. Тут – аналогічно. Різні системні блоки і монітори відрізняються один від іншого різними спектрами радіовипромінювання, а це полегшує процес селекції інформації.

Звичайно, цей організаційний метод не знімає проблему радіовипромінювання комп'ютерів. Зменшувати радіовипромінювання необхідно, але зменшувати його вплив у всіх комп'ютерах, що входять до складу групи. І в першу чергу в тих, чия інформація важливіше, і, звичайно, таємніша. Інші комп'ютери будуть забезпечувати своєрідне «прикриття» випромінюванням, що маскує.

На додаток до викладених рекомендацій слід зазначити, що значною мірою проблему випромінювання моніторів комп'ютерів можна вирішити заміною традиційних моніторів, заснованих на принципі високочастотного розгорнення зображення на екрані електроннопроменевої трубки на більш сучасні монітори з рідкокристалічними (РК) екранами. РК-монітори характеризуються значно меншим рівнем випромінювань: радіо й інших. На жаль, ці пристрої набагато дорожче. Але ціни на рідкокристалічні монітори швидко знижуються, що поступово робить їх більш доступними. Тому можна припускати, що в найближчі кілька років монітори цього типу будуть більш поширеними, ніж у цей час, і проблема випромінювання моніторів трохи зменшиться.

3.12. Основні стаціонарні засоби захисту інформації

Вібросистема WNG-006

Комплект призначений для постановки перешкод системам перехоплення інформації, що працює по віброакустичному каналі витоку.

Комплект складається з блока формування перешкоди і датчиків. Блок

формує електричний сигнал, промодульований випадковим чином. Датчик перетворить електричний сигнал, переданий по кабелю від блока формування перешкоди, у вібросигнал. Датчик жорстко кріпиться на поверхні, що захищається, і перекриває площі від 1 до 1,5 кв. м.

Габарити датчика – циліндр діаметром 50 і висотою 10 мм. Живлення блока формування перешкоди – мережеве 220 В / 50 Гц.

Фільтр «Граніт-8»

Фільтр призначений для блокування витоку акустичної інформації по телефонній лінії при покладеній трубці телефонного апарата.

Основні технічні й експлуатаційні характеристики:

– Виріб призначений для роботи на навантаження 600 Ом +/-10 % у безупинному режимі.

– Загасання в смузі частот 0,15–10 кГц при рівні вхідного сигналу 10 В не більше ніж 3 дБ.

– Загасання при вхідній напрузі 10 В на частоті 50 Гц не менше ніж 6 дБ, на частоті 100 кГц не менше ніж 10 дБ.

– Габаритні розміри виробу не більше ніж 95х60х25 мм.

– Маса фільтра не більше ніж 0,2 кг.

Пристрій захисту від диктофонів і радіомікрофонів «Буран-2»

Виріб «Буран-2» призначено для запобігання несанкціонованого запису акустичної інформації в приміщенні на диктофон та ретрансляції інформації за допомогою радіомікрофона. Придушення здійснюється шляхом постановки нечутної для вуха людини перешкоди.

Виріб виконаний у вигляді трьох функціонально і конструктивно закінчених модулів: формувача перешкодного сигналу, антенного вузла і вузла живлення. Усі вузли розміщені на шасі, вбудованому в аташе-кейсі.

Основні технічні й експлуатаційні характеристики:

– Дальність придушення – 1,5 м.

– Ширина головного пелюстка на рівні 3 дБ – 45–60°.

– Напруга живлення від автономного джерела – 30–32 В.

– Споживаний струм від автономного джерела – < 900 ма.

– Живлення – від мережі 220 В / 50 Гц (і від акумуляторів).

Стаціонарний цифровий виявник диктофонів PTKD-018

У новій системі PTRD-018, побудованій на основі мікропроцесора 80C251 SB, застосовані найсучасніші принципи виявлення диктофонів, що дозволяють охопити до 16 посадкових місць. Дальність виявлення за сприятливих умов – 1,5 м для кожного датчика.

Системою можна керувати за допомогою комп'ютера, який підключається через порт RS-232, що дає змогу інтегрувати PTRD-018 у глобальну систему захисту.

Генератор «білого шуму» IVNG-022

Цей пристрій дозволить ефективно забезпечити захист переговорів від систем промислового шпигунства, що прослуховують.

Прилад випромінює так званий «білий шум» в основному спектрі звукових частот, що забезпечує маскування розмови і робить практично неможливим її розуміння після передачі будь-якими типами систем, що прослуховують. Прилад впливає безпосередньо на вхідні низькочастотні системи, що підслуховують (мікрофони), незалежно від особливостей їх схемотехніки і принципів передачі інформації цих пристроїв.

Генератор шуму «Гном-3»

Генератор шуму «Гном-3» «маскує» у перешкоді корисну інформацію, що міститься в побічних електромагнітних випромінюваннях. *Основні технічні й експлуатаційні характеристики.*

Рівень шумового сигналу на вихідних розніманнях генератора в діапазоні частот:

– від 10 до 150 кГц (f прийм.= 200 Гц) – не менше ніж 70

дБ;

– від 150 кГц до 30 МГц (f прийм.= 9 кГц) – не менше ніж 70 дБ;

– від 30 до 400 МГц (f прийм.=120 кГц) – не менш 75 дБ;

– від 400 до 1 гГц (f прийм.= 120 кГц) – не менше ніж 45 дБ.Ослаблення рівня сигналу в піддіапазонах частот:

– від 10 до 150 кГц – не менше ніж 30 дБ;

– від 150кГц до 30 МГц – не менше ніж 30 дБ;

– від 30 300 МГц – не менше ніж 20 дБ. Ентропійний коефіцієнт шуму – не менше ніж 0,8.

СТО-24 «Хуртовина»

Прилад призначений для контролю параметрів телефонних ліній для виявлення несанкціонованого гальванічного підключення електронних засобів знімання інформації (ЗЗІ), а також для/або виключення утруднення нормальної роботи цих засобів. Прилад призначений для роботи на телефонних лініях міських АТС, а також може бути модифікований для роботи на міні АТС.Прилад дозволяє:

1) придушувати послідовно підключені електронні ЗЗІ, що мають «м'який» радіоканал передачі (не мають кварцової стабілізації частоти):

– значно знизити потужність передавача (відношення «сигнал/шум») паралельно підключених ЗЗІ, що мають «твердий» радіоканал (кварцова стабілізація частоти) передачі;

2) знизити ефективність застосування паралельно підключених ЗЗІ, що мають «м'який» канал, за рахунок зсуву і зміни спектра сигналу:

– вмикати пристрої, що прослуховують, якими управляють по сигналу звукового діапазону, на передачу і запис шуму замість інформації за рахунок створення активної шумоподібної перешкоди в різних режимах роботи телефонного апарата: якщо слухавку знято; якщо слухавку покладено;

– установлювати пасивне загородження прийому сигналів мовного діапазону з телефонної лінії за допомогою індукційних датчиків шляхом зменшення відношення сигнал/шум не менше ніж у 3 рази;

– установлювати пасивне загородження поширенню в лінії високочастотних сигналів несучих частот радіопередавачів;

– забезпечувати загасання в смузі частот 0,15...15 кГц при рівні вхідного сигналу 10 В не більше ніж 3 дБ;

– забезпечувати загасання на – 50 кГц при рівні сигналу 10 В – не менше ніж 60 дБ, на – 100 кГц не менше ніж 100 дБ.

Основні технічні й експлуатаційні характеристики:

– живлення генератора перешкоди від телефонної лінії – 60 В;

– струм споживання в режимі захисту – 500 мкА;

– живлення від акумулятора – 9 В;

– погрішність виміру U ± 0,2;

– розмах шумового сигналу прямокутної форми 4,5...5 У;

– габарити 12,5 х 68 х 40 мм.

3.13. Пошукове устаткування

Професійний багатополосний радіоприймач AR-3000A

Має рідкокристалічний індикатор, на якому відображається рівень сигналу (у дБ), частота прийнятого сигналу, вид модуляції, номер каналу

пам’яті, режим роботи й інше. Може працювати разом з IBM/PC у разі наявності програмного забезпечення. Можливе підключення магнітофона, акустичних систем, навушників.

Основні технічні й експлуатаційні характеристики:

– частотний діапазон: 100 кГц – 2 036 МГц;

– види модуляції: USB, LSB, CW, AM, NFM, WFM;

– кількість каналів пам’яті: 4 банки (по 100 каналів);

– чутливість:

Діапазон	Вид модуляції			
	SSB/CW	AM	NFM	WFM
1 Мгц 2.5 Мгц	1.0мВ	3.2 мкВ	-	-
2.5 мГц-1.8ГГц	0.25 мкВ	1.0 мкВ	0.35 мкВ	1.0 мкВ
1.8 ГГц-2 ГГц	0.75 мкВ	3.0 мкВ	1.25 мкВ	3.0 мкВ

– аудіовихід: 4 – 8 Ом;

– напруга: 13,8 В (через блок живлення від мережі 110/220 В);

– розміри: 138 х 80 х 200 мм.

Портативний нелінійний радіолокатор «АТ-6»

Призначений для виявлення пристроїв, що містять напівпровідникові елементи: транзистори, діоди, мікросхеми.

Основні технічні й експлуатаційні характеристики:

– живлення виробу здійснюється як від мережі перемінного струму частотою 50 Гц, напругою 220 В, так і від джерела постійного струму напругою 12 В;

– потужність, споживана виробом від джерела живлення, не перевищує: 4 Вт в імпульсному режимі;

– час безупинної роботи виробу не більше ніж 8 год;

– вага блока портативного нелінійного радіолокатору не перевищує 2 кг;

– передавач працює в імпульсному режимі на частоті 905+1 МГц;

– тривалість імпульсу не більше ніж 1,5 мкс;

– частота проходження імпульсів;

– в імпульсному режимі 400+50 Гц;

– у режимі, що обгинає 20+1 кГц;

– частота настроювання дорівнює подвоєній частоті передавача;

– реальна чутливість при співвідношенні сигнал\шум не менше ніж 6 дБ і при вихідній напрузі 0,1 В (амплітудне значення) – не гірше10 – 11 дБ;

– регулювання посилення приймача здійснюється плавно в діапазоні 0–30 дБ;

– узагальнена ширина діаграми спрямованості за рівнем 0,5. Розмах не більше ніж 90°.

4 Шкідливе програмне забезпечення

4.1 Класифікація шкідливого програмного забезпечення

Під терміном *шкідливе програмне забезпечення* (англ. – malware) розуміють програмні засоби, що несанкціоновано впроваджують у комп'ютерну систему і які здатні викликати порушення політики безпеки, завдавати шкоди інформаційним ресурсам, а в окремих випадках – і апаратним ресурсам комп'ютерної системи. Деякі програми навіть можуть виконувати руйнівну функцію, тому їх називають *руйнівними програмними засобами* (англ. – destructive software). Хоча шкідливі програмні засоби, які не мають вбудованої руйнівної функції, теж не можна вважати безпечними для комп'ютерної системи. По-перше, вони витрачають ресурси системи, а по-друге, порушують її політику безпеки. Спільною ознакою шкідливого програмного забезпечення є те, що воно виготовлене з метою порушення політики безпеки.

Шкідливе програмне забезпечення класифікують за різними ознаками. Наприклад, у монографії його поділяють на дві категорії – таке, що виконує деструктивні функції, і таке, що їх не виконує. В інших джерелах виділяють як окремий клас шкідливого програмного забезпечення так звані програмні закладки. Іноді термін *програмні закладки* (крім комп'ютерних вірусів). Протиставлення вірусів і програмних закладок є помилковим. Будь-який шкідливий програмний засіб може або встановлювати програмну закладку, або не робити цього. Програмна закладка працюватиме на комп'ютері деякий час, допоки її не буде виявлено або згідно із закладеним у неї алгоритмом. Натомість, деякі програмні засоби, скажімо, «троянські коні», можуть здійснювати руйнівні дії з катастрофічними наслідками (наприклад, форматування диска) безпосередньо під час своєї активації, не намагаючись залишити в системі свої компоненти.

Чи виконуватимуть шкідливі програмні засоби руйнівні функції, визначити складно. По-перше, як уже зазначалося, вони у будь-якому разі порушують політику безпеки. По-друге, навіть якщо розробник шкідливого засобу не передбачив у ньому руйнівних функцій, такий засіб може призвести до значних втрат – як через необхідність спрямування зусиль висококваліфікованих (і високооплачуваних) фахівців на виявлення, ідентифікацію, видалення шкідливого програмного засобу, так і через недоступність систем. Яскравий приклад – хробак Морріса, який не мав жодних руйнівних функцій, проте за дуже короткий час свого функціонування в Інтернеті (протягом однієї доби) завдав збитків, які було оцінено впонад 98 млн доларів.

Тому доцільно класифікувати шкідливе програмне забезпечення за двома головними ознаками:

- способом розповсюдження засобу – яким чином засіб потрапляє на комп'ютер і домагається своєї активізації;
- метою функціонування засобу – які саме шкідливі дії він здійснює.

Шкідливе програмне забезпечення завдає шкоди комп'ютеру під час запуску його коду на виконання. Тому шкідливі програмні засоби застосовують такі механізми розповсюдження, що дають їм змогу виконуватися на комп'ютері або взагалі без втручання користувача, або непомітно для нього. За механізмами розповсюдження виділяють такі шкідливі програмні засоби.

✓ Класичні комп'ютерні віруси:

1. файлові віруси;
2. завантажувальні віруси;
3. макровіруси;
4. скриптові віруси.

✓ Мережні хробаки:

1. поштові хробаки;
2. хробаки, що використовують інтернет-пейджери;
3. хробаки в IRC-каналах;
4. хробаки для файлообмінних мереж (Peer-to-Peer);
5. інші мережні хробаки.

✓ «Троянські коні».

✓ Спеціальні хакерські утиліти.

Класичні комп'ютерні віруси – це програмні засоби, які здатні самостійно відтворюватися, тобто розмножуватися, і використовують як носій інший програмний код, який вони модифікують у такий спосіб, щоб впровадити в нього свою копію. У результаті замість програмного коду, запущеного користувачем на виконання, виконується код вірусу.

Класичні мережні хробаки здатні самотужки, без будь-якого втручання користувача, розповсюджуватися у комп'ютерній мережі, виконуючи щонайменше дві функції: передавання свого програмного коду на інший комп'ютер і запуск свого програмного коду на віддаленому комп'ютері. Для цього хробаки використовують уразливості комп'ютерних систем. На відміну від класичних вірусів хробаки, як правило, не використовують як носії коди інших програм, оскільки не мають на меті примусити користувача у такий спосіб запустити їх. Іноді хробаками називають ще такі програми, які не здатні самотужки запустити себе на виконання на віддаленій комп'ютерній системі, й відтак застосовують принцип «троянського коня».

Категорію «троянські коні» не поділяють на підкатегорії за способами їх розповсюдження. їх класифікують за тими діями, які вони здійснюють на зараженому комп'ютері (така класифікація значною мірою повторює класифікації програмних закладок). Щодо способу зараження комп'ютера, то про нього каже сама назва «троянський кінь». Ці програми, використовуючи різні методи соціальної інженерії, приваблюють довірливого користувача, який запускає їх на виконання, отримуючи зовсім не ті результати, на які розраховував (у деяких «троянських коней» шкідливі функції добре приховано, тому користувач може навіть не підозрювати, що його комп'ютер уже скомпрометовано).

До спеціальних засобів належать дуже небезпечні засоби, які не мають своїх механізмів розповсюдження і які користувачі свідомо запускають на виконання як звичайні програми. Такі засоби зловмисники застосовують, якщо мають певні повноваження в системі (можливо, отримані несанкціоновано). Деякі з цих засобів називають *експлойтами* (англ. – exploit), що підкреслює факт використання (експлуатації) ними деякої вразливості системи. Часто такі засоби призначені для атаки не того комп'ютера, на якому вони запущені, а інших комп'ютерів у мережі. Інколи зловмисники використовують ці засоби як інструментарій, а не безпосередньо для атак. Класифікація спеціальних засобів буде наведена відповідно до функцій, які вони виконують.

Цей перелік був би неповним без ще одного класу програмного забезпечення, використання якого може призвести не лише до порушення політики безпеки, але й до пошкодження програмного, а інколи навіть апаратного забезпечення комп'ютерної системи. Це технологічні програми, які використовують адміністратори системи або технічний обслуговуючий персонал, наприклад: засоби резервного копіювання і відновлення з резервних копій, форматування дисків, дефрагментації файлових систем, перепрограмування BIOS, перевірки та редагування реєстру Windows. Такі програми не належать до шкідливих чи руйнівних, тому в цьому розділі ми їх не розглядатимемо. Проте, оскільки їх використання зловмисниками чи просто некомпетентними користувачами може мати дуже серйозні наслідки, розробляючи політику безпеки системи, слід враховувати наявність таких програм і впроваджувати заходи, що запобігають їх використанню невповноваженими користувачами.

4.2 Програмні закладки

Програмні закладки – це програми або окремі функції програм, що тривалий час працюють у комп'ютерній системі, здійснюючи заходи, спрямовані на приховування свого існування від користувача. Програмні закладки можуть впроваджувати віруси, «троянські коні», мережні хробаки чи безпосередньо користувачі- зловмисники. Іноді програмні закладки впроваджують адміністратори з метою виявлення злочинної діяльності користувачів або для керування комп'ютерами користувачів. У таких випадках програмну закладку не слід вважати шкідливою програмою, хоча функціонально вона, напевно, буде абсолютно ідентичною шкідливій програмі. Це ще одне підтвердження того, що програмні засоби в інформаційному середовищі часто відіграють роль зброї, шкода чи користь від

застосування якої залежить лише від того, в чиїх вона руках (і від того, хто оцінюватиме наслідки її застосування).

4.2.1 Функції програмних закладок

Є різні, більш або менш деталізовані класифікації функцій програмних закладок.

Нижче наведено класифікацію, яка враховує «новинки» розробників «троянських коней».

- Перехоплення і передавання інформації:
  - крадіжка паролів;
  - шпигунські програми.
- Порушення функціонування систем («логічні бомби»):
  - знищення інформації;
  - зловмисна модифікація інформації;
  - блокування системи.
- Модифікація програмного забезпечення:
  - утиліти віддаленого адміністрування (люки);
  - інтернет-клікери;
  - проксі-сервери;
  - дзвінки на платні ресурси;
  - організація DoS-і DDoS-атак.
- Психологічний тиск на користувача:
  - реклама;
  - лихі жарти і містифікації.

Як видно із класифікації, програмні закладки першої групи порушують конфіденційність, другої – цілісність і (або) доступність інформації. Функції, які виконують програмні закладки із третьої групи, характерні для «троянських коней», вірусів і хробаків; такі закладки порушують спостережність і керованість комп'ютерної системи. Нарешті, дії програмних закладок четвертої групи безпосередньо спрямовані на користувача системи. Більш детально типові програмні закладки буде розглянуто далі.

4.2.2 Шпигунські програми

Програмні закладки, що здійснюють пошук інформації на зараженому комп'ютері та передають її зловмиснику, розрізняють за типом інформації, яку вони збирають, за режимом ітехнологіями її передавання.

Окрему категорію складають програми, що збирають і надсилають паролі доступу до локальної системи і до мережних ресурсів, зокрема платних, а також до банківських систем і систем електронних платежів.

Шпигунські програми (Spyware) – ширша категорія, до якої належать програми різних типів. Деякі з них стежать за діями користувача зараженого комп'ютера, перехоплюючи інформацію, що вводиться з клавіатури, копії екрана, відомості про активні програми і про те, що користувач із ними робить. Інші здійснюють пошук інформації у файлах користувача за певними ознаками (наприклад, за ключовими словами). Є окремий різновид шпигунських програм, які цікавляться конфігурацією апаратних і програмних засобів на комп'ютері користувача, зокрема серійними номерами ліцензійних програм.

Програмні закладки цього типу здебільшого впроваджують віруси і «троянські коні», проте часто такі програмні модулі встановлюють «нормальні» програми, переважно з числа умовно-безкоштовних (Shareware) або безкоштовних (Freeware).

Головне, що їх цікавить, – це склад апаратних і програмних засобів комп'ютера та його територіальне розміщення (країна, локальна мова, часовий пояс). У такий спосіб розробники збирають статистичну інформацію про використання своєї програми, що в подальшому можестати у пригоді для розроблення її нових версій.

4.2.3 «Логічні бомби»

До категорії «логічних бомб» належать програмні закладки, які за певних умов здійснюють деякі, як правило, руйнівні дії. Іноді виокремлюють категорію «часові міни» – фактично, це окремий випадок «логічних бомб», де умовою запуску є настання

певного моменту часу. Наприклад, вірус, відомий як CIH, впроваджував часову міну, яка спрацьовувала під час завантаження комп'ютера 26 квітня, тобто у річницю Чорнобильської катастрофи (за що дістав назву «Чорнобиль», під якою він більш відомий у нашій країні). Результат діяльності цієї програми був катастрофічним для комп'ютера: вона модифікувала Flash-BIOS, після чого комп'ютер повністю втрачав роботоздатність. Як правило, для відновлення такого комп'ютера потрібно було замінювати системну плату.

4.2.4 Люки – утиліти віддаленого адміністрування

Програмні закладки цієї категорії є утилітами віддаленого адміністрування комп'ютерів у мережі. Функціонально вони подібні до систем адміністрування, що розробляють і розповсюджують відомі виробники програмних продуктів.

Єдине, що вирізняє з-поміж таких програм шкідливі програмні закладки, – це відсутність попереджень про їх інсталяцію і запуск. Користувач не отримує жодних повідомлень про дії програмної закладки в системі. Тим паче, що програмна закладка може бути прихованою і не відображатися у списку активних програм і процесів. І в той час, коли користувач навіть гадки не має про присутність у системі такої програми, його комп'ютер стаєвідкритим для віддаленого керування.

Можна констатувати, що програмні закладки цього типу є одними з найнебезпечніших, оскільки вони потенційно уможливлюють будь-які зловмисні дії. Програми, які впроваджують ці програмні закладки, також належать до найшкідливіших. Так само слід розуміти, що потенційну загрозу можуть нести абсолютно «легальні» утиліти віддаленого адміністрування, позаяк вони роблять комп'ютер уразливим для будь-яких дій ззовні. Єдине, що може захистити у разі їхнього використання, – це надійна автентифікація вузла і користувача, від якого надходять команди керування. Але ніхто не може гарантувати відсутність помилок, що відкривають доступ для сторонніх осіб (зловмисників), які через діючу систему віддаленого керування зможуть упровадити власну приховану систему.

Хоча шкідливих програм цієї категорії дуже багато, насамперед слід згадати «троянського коня» BackOrifice, який з'явився у 1998 році та набув надзвичайного на той час поширення. «Троянець» встановлював програмну закладку віддаленого адміністрування, якою міг скористатися будь-хто. Антивірусні засоби дуже швидко почали його виявляти, а зловмисники використовувати цей люк для проникнення в систему і встановлення свого, непоширеного і тому невідомого антивірусним засобам люка. Саме таким чином було

«зламано» сайт Relcom- Україна: зловмисники, які сканували інтернет у пошуках комп'ютерів, уражених BackOrifice, виявили серед них машину в офісі Relcom, впровадили свою утиліту (суттєво доопрацьований BackOrifice), яка здійснювала перехоплення клавіатурного вводу, що дало їм змогу дізнатися про IP-адреси та паролі доступу до інших машин, зокрема до захищеного сервера.

Крім «агента» – люка, який впроваджувався на віддалених машинах, – і дотепної назви (яка, між іншим, ще й пародіювала розрекламований Microsoft Back Office), BackOrifice мав утиліту-менеджера з великими можливостями і зручним графічним інтерфейсом.

Слідом за BackOrifice з'явилися інші такі програми (NetBus, Phase), що зробили свій внесок у перетворення Інтернету на небезпечне та агресивне середовище.

У наш час використання програмних закладок віддаленого керування набуло характеру глобальної епідемії. Упровадження таких закладок здійснюється спеціалізованими мережними хробаками. У результаті їхньої діяльності формується величезна армія комп'ютерів, на яких впроваджено певну утиліту віддаленого керування (так званий бот). До таких мереж, якими централізовано керує зловмисник, застосовують термін ботнет (мережа ботів). Керування переважно здійснюють через певний IRC-канал, інколи використовують веб-сайт, на якому зловмисник розміщує команди для вражених машин, а в окремих випадках навіть організовують спеціальну P2P-мережу. Перші ботнети було сформовано у 2002 році, а потім відбувся їх стрімкий розвиток. Значною мірою цьому сприяли критичні вразливості в системах Windows: у 2003 році виявили вразливість у

службі RPC DCOM Windows 2000/XP (яку використав хробак Lovesan), а у 2004 році - у службі LSASS (її використав хробак Sasser).

Окрім того, велику роль відіграв поштовий хробак Mydoom, виявлений у 2004 році. Згадані хробаки впроваджували програмні закладки віддаленого керування або інші програмні закладки роботи з мережею, причому деякі з них мали вразливості, що ними скористалися наступні покоління хробаків. Усі ці люки було використано зловмисниками для інсталяції на машинах своїх утиліт-ботів. Між різними групами зловмисників буквально спалахнула війна за машини-зомбі; скомпрометовані комп'ютери могли переходити «із рук у руки» по кілька разів на день, рано чи пізно стаючи одним із учасників деякого ботнета.

За оцінками, у 2005 році кількість уражених комп'ютерів, що входили до складу ботнетів, становила вже кілька мільйонів; і ця кількість щомісяця зростала на 300-350 тис. Ботнети активно використовують зловмисники: через них розсилають спам, організовують DDoS-атаки, розповсюджують нові версії шкідливих програм. Цілком імовірною є організаціярозподілених обчислень, наприклад, для зламу криптосистем. Як приклад DDoS-атаки, організованої з використанням ботнета, можна навести результат діяльності вже згаданого хробака Mydoom.A, який 1 лютого 2004 року надовго вивів із ладу сайт компанії SCO, виробника дистрибутивів UNIX.

4.2.4 Несанкціонована робота з мережею

Програмні закладки, які несанкціоновано працюють із мережею (надсилають або отримують повідомлення чи спеціальні пакети даних), становлять доволі численну групу. Ми вже згадували раніше ті з них, що надсилають шпигунську інформацію задля здобуття даних про користувача і його комп'ютер, а також ті, що отримують команди з мережі та виконують їх. Але є ще багато шкідливих програм, призначених для роботи з мережею, які здатні завдатизначної шкоди користувачу. Розглянемо деякі з них.

Інтернет-клікери

До цієї категорії належать програми (здебільшого «троянські коні»), основна функція яких – організація несанкціонованих звернень до ресурсів Інтернету (переважно до веб-сторінок), для чого або надсилають відповідні команди браузеру, або замінюють системні файли, де вказано «стандартні» адреси ресурсів Інтернету. Такі дії зловмисники можуть здійснювати з метою:

- підвищення кількості відвідувань деяких сайтів;
- організації DoS-атаки на деякий ресурс (хоча значно ефективніше було б здійснитискоординовану атаку, організовану системою віддаленого керування);
- привернення потенційних жертв задля впровадження на їх комп'ютери вірусів або «троянських коней».

Проксі-сервери

Прихований від користувача проксі-сервер можна використовувати для будь-якої злочинної діяльності в мережі, надаючи зловмиснику можливість анонімного (точніше, від імені користувача, який нічого не підозрює) доступу до будь-яких ресурсів Інтернету. Проксі-сервери застосовують для сканування мереж, здійснення атак на інші комп'ютери, алездебільшого їх використовують для розсилання спаму.

Доступ до платних ресурсів

Є програми, що здійснюють доступ до платних ресурсів. В Інтернеті це, як правило, не становить реальної загрози (оскільки там діє принцип – «гроші наперед»), якщо така програма не передає автоматично платіжні реквізити користувача (наприклад, номер кредитної картки).

Інсталяція з мережі

Програмні модулі-інсталятори є в багатьох «троянських конях», хоча їх можна зустріти майже в усіх сучасних програмах. У «троянцях» ці модулі звичайно спрацьовують безпосередньо під час запуску програми і часто не використовують програмних закладок. Натомість, у легальних програмах такі модулі дуже часто оформлені у вигляді типових

програмних закладок. І хоча програмні модулі-інсталятори не можна класифікувати як шкідливі програми, вони безперечно є потенційно небезпечними для комп'ютерної системи.

4.3 Комп'ютерні віруси

Свого часу серед руйнівних програмних засобів саме комп'ютерні віруси набули найбільшого розповсюдження, тому вірусами називають будь-яке шкідливе програмне забезпечення, несанкціоновано впроваджене в систему. Так само і від антивірусних засобів часто очікують захисту не лише від вірусів, а й від руйнівних програмних засобів інших видів; інколи такі сподівання виправдовуються, а інколи – ні. Насправді, часто мова йде не про вірус, а про «троянського коня», хробака чи навіть експлойт, який хтось впровадив у систему. Є різні підходи до визначення комп'ютерних вірусів. Ми зазначимо дві головні властивості, характерні саме для вірусів. Перша – це їх здатність самостійно відтворюватися, тобто розмножуватися. У програмний код вірусу закладено функції копіювання самого себе. Друга – це спосіб, у який вірус потрапляє до системи і примушує користувача запустити його. Вірус використовує як носій інший *програмний код, який він модифікує таким* чином, щоб впровадити в нього свою копію (подібно звичайним вірусам, які нездатні існувати самостійно тривалий час – їм потрібні клітини іншого живого організму). У результаті замість необхідного користувачу програмного коду виконується код вірусу.

Комп'ютерні віруси розрізняють за такими ознаками.

За середовищем існування (системні області комп'ютера, ОС, прикладні програми, до певних компонентів яких впроваджують код вірусу):

- файлові віруси;
- завантажувальні віруси;
- макровіруси;
- скриптові віруси.

За способом зараження (різні методи впровадження вірусного коду в об'єкти, які він заражає; залежно від середовища існування віруси використовують різні способи зараження, тому універсальної класифікації за цією ознакою немає).

Віруси також класифікують (або надають їм додаткових ознак) за тими технологіями, які вони використовують для ускладнення їх виявлення і ліквідації.

4.3.1 Файлові віруси

Файлові віруси для свого розповсюдження використовують файлову систему. Найчастіше віруси цього типу як носій застосовують виконувані файли. За способом зараження їх поділяють на *віруси, що перезаписують* (Overwriting), і *паразитичні віруси* (Parasitic). Оскільки перші замінюють собою оригінальний файл, знищуючи його, то в результаті їхньої діяльності прикладні програми й операційна система дуже швидко перестають функціонувати. Другі модифікують оригінальний файл, зберігаючи його функціональність. Файлові віруси цього типу найпоширеніші, тому їх згодом буде розглянуто детальніше. Є також *компаньйон-віруси*, які створюють файли-двійники, а також *link-віруси,* що використовують особливості організації файлової системи.

Методи зараження файлів паразитичними вірусами

Тепер детальніше розглянемо, як діють паразитичні віруси. їхньою типовою ознакою є те, що вони обов'язково змінюють уміст файлів, залишаючи останні повністю або частково роботоздатними.

Свого часу їх поділяли на *exe-* та *com-віруси*, відповідно до двох форматів виконуваних файлів, що були в MS-DOS. Тепер такий розподіл неактуальний. Зараз характерною ознакою паразитичних вірусів є те, під керуванням якої операційної системи вони можуть функціонувати.

Сучасні ОС підтримують кілька альтернативних форматів виконуваних файлів. Ці доволі складні формати надають широкі можливості для вірусів, які впроваджують свій код без порушення функціональності програми. Є віруси, що записують себе на початку файлів (Prepending), у їх кінець (Appending) і в середину (Inserting). Впровадження вірусів у середину файлів також відбувається у різні способи – перенесенням частини файлу в його

кінець або копіюванням свого коду в ті частини файлу, що не використовуються (Cavity-віруси). Основніспособи зараження файлів вірусами показано на рис.1.

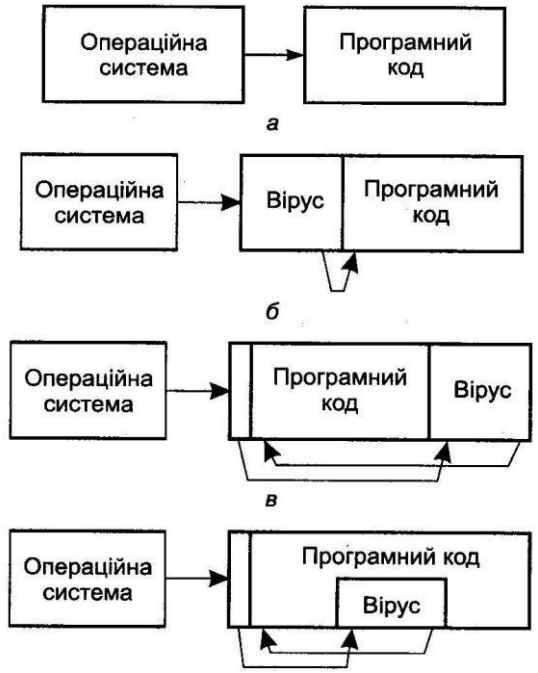


Рис.1. Основні способи зараження файлу вірусом: а – нормальне виконання незараженого файлу; *б* – програмний код вірусу розташовано на початку файлу, сам файл дописано в кінецьвірусу (операційна система передає керування безпосередньо вірусу, а той після виконання своїх функцій передає керування файлу); в – програмний код вірусу дописано в кінець файлу (вірус коригує точку входження програми в заголовку оригінального файлу, щоб першим отримати керування); г – програмний код вірусу розташовано всередині оригінального файлу(вірус коригує точку входження програми, щоб першим отримати керування, або (на рисунку не показано) розміщує перехід на свій код будь-де у програмі, щоб отримати керування в певний момент)

Після того як вірус отримує керування, він виконує певні дії та передає керування програмі-носію. На цьому етапі вірус звичайно не здійснює руйнівних дій, а лише вживає заходів для свого розповсюдження (наприклад, «заражає» інші файли, тобто вбудовує в них свій код) і для отримання керування в подальшому (для чого впроваджує програмну закладку, за старою термінологією MS-DOS – резидентну частину). Часто вірус діє так швидко, що натлі звичайної процедури запуску програми жоден користувач не зможе його помітити, навіть якщо контролюватиме час запуску із секундоміром.

4.3.2 Завантажувальні віруси

Завантажувальні, або бутові (англ. boot – початкове завантаження, специфічний комп'ютерний термін, скорочення від Bootstrap Loader – програма початкового завантаження), віруси активуються у момент завантаження системи. Для цього їм потрібно розташувати частину свого коду в службових структурах носія, з якого відбувається завантаження – жорсткого диска або дискети. Зараження дискети здійснюється записуванням коду вірусу замість оригінального коду boot-сектора дискети. Зараження жорсткого диска відбувається в один із трьох способів: вірус записує себе замість коду MBR (Master Boot Record – головний завантажувальний запис, таблиця у першому секторі завантажувального диска) чи замість коду boot-сектора завантажувального диска (у Windows – це, як правило, диск C) або модифікує адресу активного boot-сектора в таблиці розділів диска (Disk Partition Table), що знаходиться в MBR.

На жорсткому диску бутові віруси можуть вельми спокійно існувати і завдавати величезної шкоди інформаційним ресурсам. Вони також можуть дуже ефективно протидіяти антивірусним засобам, оскільки саме віруси стартують першими, ще до запуску операційної системи, і тому вони здатні залишити за собою керування критичними для їх існування ресурсами комп'ютера, зокрема, файловою системою. З іншого боку, для цього

потрібно фактично утворити для ОС віртуальну машину, що для сучасних систем хоча і є можливим (адже існують спеціальні програмні засоби, на кшталт vmware), але потребує великого обсягупрограмного коду, що не дуже прийнятно для вірусу.

Великим і складним вірусам, розміщеним у завантажувальних секторах дисків (яким, наприклад, був OneHalf), для розповсюдження потрібен був інший носій, позаяк жорсткі диски переносять нечасто, а на дискетах не вистачало місця і для операційної системи, і для вірусу. Тому їх розповсюджували як звичайні файлові віруси, що після запуску намагалися заразити MBR. Поєднання характеристик бутових і файлових вірусів було типовим. Зараження бутовим вірусом особливо небезпечне. За деякими повідомленнями, той самий OneHalf, впроваджений у MBR, досить вільно існував у захищеному середовищі Windows NTWorkstation 4.0.

У сучасних операційних системах модифікація MBR ретельно контролюється. її так само контролюють апаратні засоби деяких системних плат, тому такі віруси мають небагато шансів на успіх. Однак це твердження справедливе лише за умови правильного адміністрування і дотримання політики безпеки.

Сучасні знімні носії (наприклад, flash-накопичувачі, оптичні диски), що мають достатній об'єм, інколи використовують для завантаження операційної системи. Flash-накопичувачі для цього використовують рідко. Проте за всіма ознаками вони могли б стати підґрунтям для відродження бутових вірусів, якби завантаження з них було більш поширеною процедурою. Хоча, безумовно, «підхопити» вірус із піратського диска можна. Також відомі прецеденти, коли віруси містилися на дисках із презентаційними матеріалами, демоверсіями програмного забезпечення, комп'ютерними виданнями.

4.3.3 Макровіруси

Макровіруси – це віруси, які використовують прихований у файлах документів програмний код (так звані макроси). Ідея макросів виникла, коли програмістам під час роботи з документами доводилося багаторазово виконувати одні й ті самі рутинні операції редагування або певні, завжди однакові, послідовності дій. Природно було для виконання таких дій визначити процедуру, яка б виконувала певну послідовність операцій автоматично. Роль елементарних операцій у макросі виконують окремі команди з множини передбачених у програмі оброблення документа команд, або спеціально розроблені макрокоманди. Свої макромови мають графічні редактори, системи автоматизованого проектування, текстові та табличні процесори.

Макровіруси також написано макромовами. Передумови для макровірусів виникли, коли з'явилися вдосконалені системи, нові можливості яких використовують макроси: по-перше, можливість зберігання макросів у файлі документа, а по- друге, суттєве розширення доступних для макрокоманд функцій.

Наприклад, у 1995 році вийшла чергова версія Microsoft Office, де в текстовому процесорі Microsoft Word як мову макросів було використано досить розвинену мову Word Basic, а в табличному процесорі Excel – повноцінну об'єктно- орієнтовану мову програмування Visual Basic for Applications. Архітектура програм, з яких складається Microsoft Office, передбачає для кожної команди, яку можна викликати через меню або за допомогою кнопок на панелях інструментів, виконання вбудованих макросів. Наприклад, Microsoft Word для збереження файлу за командою File ► Save виконує макрос FileSave, для збереження файлу за командою File ► SaveAs – макрос FileSaveAs, для друкування документа – макрос FilePrint.

Макроси мають доступ не лише до документа, в якому вони містяться, а й до інших об'єктів, зокрема до файлової системи. Зберігатися вони можуть не лише в документах, а й у шаблонах документів, зокрема у шаблонах Normal, які використовуються за умовчанням у всіх документах програм пакета Microsoft Office. І найголовніше – передбачено автоматичне виконання визначених макросів під час відкривання документа чи застосування шаблону, причому шаблон Normal активізується автоматично під час старту відповідної програми (Word, Excel та інших програм пакета Microsoft Office).

4.3.4 Скриптові віруси

Програмний код можна впроваджувати і в документи інших видів, наприклад в

HTML-сторінки, що завантажуються з мережі чи локально та відображаються на екрані за допомогою браузера. Наразі автоматично виконується програмний код сценаріїв, які ще називають скриптами (англ. script – сценарій) та інших елементів (ActiveX, Java). Програмний код сценаріїв може існувати й окремо, у спеціальних файлах. Деякі дуже розвинені мови сценаріїв можна вважати повноцінними мовами програмування. Наприклад, в операційних системах UNIX і Linux сценарії використовують як системні команди на рівних правах з бінарними виконуваними файлами. Далеко не всі користувачі знають, що вони запускають – скомпільовану програму чи сценарій. Сценарії можуть також мати встановлений атрибут SUID.

Скриптові віруси розглядають як підгрупу файлових вірусів. Такі віруси пишуть різними мовами сценаріїв (VBS,JS, BAT, PHP тощо). Вони можуть заражати інші програми-сценарії (командні та службові файли Windows або UNIX), бути компонентами багатокомпонентних вірусів, заражати файли інших форматів (наприклад, згаданий вище HTML), якщо вони підтримують виконання сценаріїв.

4.3.5 Захист від комп'ютерних вірусів

Найбільш поширені методи виявлення вірусів (та інших шкідливих програмних засобів) – це:

- пошук сигнатур;
- евристичний аналіз;
- контроль незмінності об'єктів.

Одним із основних методів виявлення вірусів був і залишається пошук характерних ознак відомих вірусів (сигнатур) у файлах і оперативній пам'яті комп'ютера. Деякий час великі надії покладали на евристичний аналіз, коли впровадження шкідливого коду виявлялося за наявністю підозрілих операцій (наприклад, відкриття для модифікації виконуваних файлів, перехоплення переривань тощо). Деякі засоби контролювали незмінність файлів, здебільшого виконуваних, що унеможливлювало впровадження в них коду вірусу. Сучасні антивірусні засоби поєднують у собі всі ці можливості, причому для виявлення відомих вірусів (а також хробаків, «троянських коней» та інших небезпечних програмних засобів) найефективнішим залишається саме пошук їхніх сигнатур. За режимом дії антивірусні засоби поділяють на:

- антивірусні сканери;
- антивірусні монітори;
- антивірусні фільтри.

Антивірусні сканери час від часу або за запитом здійснюють повне сканування файлової системи комп'ютера або вибіркове сканування заданих файлів чи каталогів.

Антивірусні монітори працюють безперервно, але вони здійснюють лише вибіркову перевірку і тому, як правило, не дуже уповільнюють роботу комп'ютера. Обов'язково перевіряються ті файли, над якими здійснюються будь-які операції (відкривання, читання, записування, переміщення файлу, запуск на виконання), а якщо таких операцій небагато, відбувається повільне вибіркове сканування файлової системи.

Антивірусні фільтри призначені для роботи здебільшого з тими потоками даних, що надходять із мережі. Вони ефективні для перевірки повідомлень, які надходять електронною поштою чи з каналів IRC, P2P-мереж та інших.

Віруси (а також хробаки і «троянські коні»), зі свого боку, протидіють антивірусним засобам у різні, часто досить вибагливі, способи.

Головним засобом проти пошуку сигнатур став так званий *поліморфізм* – модифікація коду вірусу від екземпляра до екземпляра. Для реалізації поліморфізму здійснюється переважно зашифровування коду з використанням різних ключів, а першим компонентом вірусу, який отримує керування, є розшифрувальник.

Деякі віруси досить ефективно приховують себе від програм, які контролюють розмір файлів (наприклад, на системний запит видають невірну інформацію про довжину файлу, датуйого модифікації тощо). Подібні технології дістали назву *стелс* (Stealth).

4.4 Мережні хробаки

Основною ознакою мережного хробака є його здатність самостійно, без втручання користувача, розповсюджуватись у комп'ютерній мережі, забезпечуючи щонайменше дві функції: передавання свого програмного коду на інший комп'ютер і запуск свого програмного коду на віддаленому комп'ютері. Здебільшого мережні хробаки, як і комп'ютерні віруси, здатні розмножуватись, і тому їх часто розглядають як різновид вірусів. Однак на відміну відкласичних комп'ютерних вірусів більшість хробаків не використовують як носій код іншої програми, оскільки не мають на меті примусити користувача у такий спосіб запустити їх.

Класичний мережний хробак використовує вразливості програмного забезпечення, яке реалізує ті чи інші мережні протоколи. Таке програмне забезпечення діє автоматично відповідно до вимог протоколу, а часом, через помилки розробників або завдяки їхньому специфічному погляду на деякі вимоги специфікацій протоколів – і всупереч вимогам стандартних протоколів. Слід зазначити, що в переважній більшості ситуацій такі програми не покладаються на користувача, а часто інтерфейс взаємодії з користувачем локальної системи взагалі відсутній. Особливості програмного забезпечення, що обслуговує мережну взаємодію, буде докладно розглянуто в наступних лекціях, оскільки воно суттєво впливає іна можливості компрометації системи через мережу, і на методи її захисту.

4.4.1. Класифікація мережних хробаків

Основною ознакою, за якою хробаків поділяють на різні типи, є спосіб їх розповсюдження –яким чином хробак передає свою копію на віддалені комп'ютери. Іншими ознаками є способи запуску копії хробака на комп'ютері, методи його впровадження в систему та характеристики,притаманні різним видам шкідливого програмного забезпечення (вірусам і «троянським коням») – поліморфізм, прихованість тощо. Розглянемо такі типи хробаків:

- поштові хробаки (Email-worm);
- хробаки, що використовують інтернет-пейджери (IM-worm);
- хробаки у IRC-каналах (IRC-worm);
- хробаки для файлообмінних мереж (P2P-worm);
- інші мережні хробаки (Net-worm).

Поштові хробаки

До цієї категорії належать хробаки, які для свого розповсюдження використовують електронну пошту. Хробак надсилає свою копію у вигляді вкладення (приєднаного файлу) в електронний лист або розміщує посилання (з URL-адресою) на свій файл на мережному ресурсі (наприклад, на скомпрометованому чи хакерському сайті). Як правило, код хробака активізується після втручання користувача: у першому випадку необхідно відкрити зараженевкладення, у другому – скористатися посиланням на заражений файл.

Поштові хробаки надсилають заражені повідомлення у різні способи:

- прямим підключенням до SMTP-сервера;
- використанням сервісів Microsoft Outlook;
- застосуванням функцій Windows MAPI.

Хробаки, що використовують інтернет-пейджери

Хробаки цього типу розсилають повідомлення, що містять URL-адресу файлу з кодом хробака, на контакти, отримані з контакт-листа інтернет-пейджера. Цей спосіб розсилки подібний до того, що використовують поштові хробаки.

Хробаки в IRC-каналах

Ці хробаки, як і поштові, розсилають URL-посилання на копію хробака або безпосередньо заражений файл, причому розсилання здійснюється по IRC-каналах. У другому варіанті користувач, якого атакують, має підтвердити отримання файлу, зберегти його на диску і відкрити.

Хробаки для файлообмінних мереж

Більшість із розглянутих типів мережних хробаків реалізують лише доставляння коду хробакана комп'ютер жертви. При цьому імітується отримання файлу з достовірних джерел (від відомих користувачу контактів), що й провокує користувача на запуск файлу.

Інші мережні хробаки

Це хробаки, які використовують інші способи зараження віддалених комп'ютерів. Серед них такі:

- копіювання хробака на мережні ресурси;
- проникнення в мережні ресурси публічного використання;
- проникнення на комп'ютер через уразливості в операційних системах і застосуваннях;
- паразитування на інших шкідливих програмах.

4.5 «Троянські коні»

Програми, які дістали назву «троянські коні» (іноді їх називають «троянськими програмами» і «троянами» або «троянцями»), – це програми, що мають привабливий зовнішній вигляд, але виконують шкідливі, дуже часто – руйнівні функції. У деяких «троянських коней» ці функції добре приховані, тож користувач може і не підозрювати, що його комп'ютер уже скомпрометований. Класичний «троянський кінь» не має функцій доставляння програми на комп'ютер-жертву, позаяк єдине його завдання – звернути на себе увагу користувача і змусити його запустити цю програму.

Яким чином «троянці» приваблюють користувачів? Методи введення користувачів в оману, що враховують особливості конкретної категорії осіб, називають *соціальною інженерією*. Для прикладу розглянемо одне з творінь, яке належить до категорії хробаків – позаяк активно розсилає себе, але для свого запуску потребує дій користувача (тобто має ознаки «троянського коня»). Цей хробак – перший хробак, розроблений для стільникових телефонів, що розповсюджується за допомогою MMS-повідомлень. Хробак працює на телефонах під керуванням ОС Symbian Series 60 і розповсюджується через Bluetooth і MMS.

Після запуску хробак ініціює пошук пристроїв, доступних через Bluetooth, і передає на них заражений SIS-архів із довільним ім'ям. Щоб його відкрити (і заразити телефон), потрібно кілька разів підтвердити приймання файлу. Цікавим є спосіб розповсюдження через MMS. Хробак розсилає себе по контактах адресної книги в MMS-повідомленнях, вставляючи тему і текст повідомлення, які мають зацікавити користувача і приспати його пильність (слід врахувати, що MMS надходить від особи, відомої потенційній жертві).

Як бачимо, типовими пропозиціями є безкоштовні програми – різні утиліти, зокрема, антивірусні програми, ігри, а також засоби безкоштовного доступу до платних ресурсів Інтернету.

4.5.1 Класифікація «троянських коней»

«Троянських коней» звичайно класифікують за тими діями, які вони здійснюють на зараженому комп'ютері (така класифікація значною мірою повторює класифікацію програмних закладок, розглянуту нами раніше):

- шпигунські програми (Trojan-Spy);
- крадіжка паролів (Trojan-PSW);
- крадіжка кодів доступу до мережі AOL (America Online); ці «троянці» складаютьокрему групу через свою численність (Trojan-AOL);
- сповіщення про успішну атаку (Trojan-Notifier);
- троянські утиліти віддаленого адміністрування (Backdoor);
- інтернет-клікери (Troj an- Clicker);
- доставляння шкідливих програм (Trojan-Downloader);
- інсталяція шкідливих програм (Trojan-Dropper);
- троянські проксі-сервери (Trojan-Proxy);
- «бомби» в архівах (ArcBomb);
- інші троянські програми (Trojan).

Деяких пояснень потребує остання категорія. До неї належать ті «троянці», що здійснюють руйнування або зловмисну модифікацію даних, порушують роботу комп'ютера тощо. Такі дії можуть бути здійснені шляхом впровадження «логічної бомби» (тобто

програмної закладки) або безпосередньо під час запуску «троянського коня». До цієї категорії також належать багатофункціональні «троянські коні», які, наприклад, надають зловмиснику доступ до зараженого комп'ютера або проксі-сервера і водночас стежать за діями локального користувача.

4.5.2 Шпигунські троянські програми

Численні «троянці» відразу після запуску «викрадають» із комп'ютера цінну інформацію і надсилають п зловмиснику за заданою в їхньому коді електронною адресою. Оскільки найчастіше такі програми «крадуть» паролі доступу до Інтернету (нерідко звідповідними номерами телефонів), за ними закріпилася назва Password-Stealing-Ware (PSW). Деякі «троянці» передають також іншу інформацію про заражений комп'ютер, наприклад, про систему, тип поштового клієнта, IP-адресу, а іноді й реєстраційну інформацію до різного програмного забезпечення, коди доступу до мережних ігор тощо.

Також є категорія «троянців», які діють як складова багатокомпонентних наборів шкідливого програмного забезпечення. Завдання цих програм – сповістити розробника про зараження комп'ютера (інсталяцію програмної закладки). На адресу розробника надсилається, наприклад, IP-адреса комп'ютера, номер відкритого порту, адреса електронної пошти тощо.

4.5.3 Троянські інсталятори

Інсталятори поділяють на дві категорії – Downloader і Dropper. Перші здійснюють завантаження програм із мережі, а другі – містять програми для інсталяції у собі.

Програми класу Downloader часто використовують програмну закладку для здійснення періодичного оновлення версій шкідливих програм. Інколи такі програми здійснюють одноразове завантаження з мережі інших «троянців» або рекламних систем. Завантажені з Інтернету програми запускаються на виконання або реєструються на автозапуск відповідно до можливостей операційної системи. Усі ці дії відбуваються без відома користувача. Інформація про імена та розміщення програм, що завантажуються, закладена в код «троянця» або завантажується ним із «керуючого» ресурсу Інтернету (як правило, з веб-сторінки).

Троянські програми класу Dropper створено для приховування інсталяції інших програм (ясно, що шкідливих).

За допомогою основного коду інші компоненти файлу записуються на диск (у корінь диска C, у тимчасовий каталог, каталоги Windows) і запускаються на виконання. Це відбувається без жодних повідомлень або з хибними повідомленнями про помилку в архіві. При цьому щонайменше один із компонентів є «троянським конем», і щонайменше один компонент є «обманкою» (програмою-жартом, грою, картинкою тощо).

«Обманка» відволікає користувача та імітує корисні дії програми, поки троянський компонентінсталюється в системі.

4.5.4 «Троянські бомби»

На відміну від «логічних бомб», які спрацьовують за певної умови, «бомби», закладені у «троянські коні», спрацьовують одразу після запуску такого «троянця». Напевно, це найкласичніший різновид «троянців», хоча останнім часом і не такий поширений. Не всі дії теперішніх зловмисників можна назвати «чистим» вандалізмом, навіть якщо їх ціллю є блокування деякої (але не будь-якої) системи або знищення інформації.

Функції такої «бомби» реалізують таким чином: використовують некоректний заголовок архіву, дані, що повторюються, чи однакові файли в архіві. Некоректний заголовок чи зіпсовані дані в архіві можуть призвести до збою в роботі архіватора або алгоритму розархівування. Дуже великий файл, який містить дані, що повторюються, можна заархівувати в архів невеликого розміру. Величезну кількість (десятки тисяч) однакових файлів спеціальними методами можна упакувати в невеликий архів (десятки кілобайтів). Розпакування таких архівів призводить до несподівано великих затрат ресурсів процесора, пам'яті та дискового простору.

5 Захист інформації в комп'ютерних мережах

5.1. Захист інформаційї в локальних мережах

5.1.1. Використання міжмережевих екранів

Міжмережевий екран, Мережевий екран, Фаєрвол, Файрвол (англ. *Firewall,* буквально «вогняна стіна») – пристрій або набір пристроїв, сконфігурованих, щоб допускати, відмовляти, шифрувати, пропускати через проксі весь комп'ютерний трафік згідно з набором правил та інших критеріїв.

Фаєрвол може бути у вигляді окремого приладу (так званий маршрутизатор або роутер) або програмного забезпечення, що встановлюється на персональний комп'ютер чи проксі-сервер. Простий та дешевий фаєрвол може не мати такої гнучкої системи налаштувань правил фільтрації пакетів та трансляції адрес вхідного та вихідного трафіку (функція редиректу).

Залежно від активних з'єднань, що відслідковуються, фаєрволи поділяють на:

- stateless (проста фільтрація), які не відслідковують поточні з'єднання (наприклад TCP), а фільтрують потік даних виключно на основі статичних правил;
- stateful (фільтрація з урахуванням контексту) – з відслідковуванням поточних з'єднань та пропуском тільки таких пакетів, що задовольняють логіці й алгоритмам роботи відповідних протоколів та програм. Такі типи фаєрволів дозволяють ефективніше боротися з різноманітними DDoS-атаками та вразливістю деяких протоколів мереж.

Функції екранів:

- *Фільтрація пакетів.* Це одна з трьох загальновідомих функцій мережевого екрана. У цьому разі виконуються зовсім прості функції (фактично як у спеціалізованого маршрутизатора), які полягають у перегляді заголовка кожного пакета та перевірки IP адреси та порта на правильність.
- *Проксі сервер.* Це друга загальновідома функція. Різниця між проксі сервером та фільтрацією пакетів полягає в тому, що проксі сервер вимагає, щоб всі сеанси зв'язку встановлювались через нього, а не напряму.
- *Проксі сервер програм*. Це третя функція. Цей різновид проксі сервера відрізняється «розумінням» протоколів програм, що здійснюють передачу даних. Хороший приклад такого сервера – поштовий сервер.
- *Кешування даних.* Це не є традиційною функцією мережевих екранів, але на цей час є надзвичайно популярною властивістю. Ідея полягає у тому, що, оскільки всі дані проходять через мережевий екран, він може зберігати найбільш популярну інформацію і при наступному звертанні за нею видати її зі свого кешу.
- *Статистика та повідомлення*. Важливою властивістю мережевого екрану є ведення історії всіх мережевих з'єднань, а також вивід повідомлень про атаки на мережу чи комп'ютер. Історія з'єднань допомагає правильно настроїти мережевий екран, щоб комп'ютер був одночасно захищений від нападів і відкритий для доступу авторизованим користувачам.
- *Управління.* Для персональних мережевих екранів основна характеристика – зручність їхнього налаштування. Управління міжмережевими екранами здебільшого відбувається дистанційно (використовуючи HTML інтерфейс чи інший), що потребує впевненості в надійності авторизації та каналу зв'язку.

Принципи роботи брандмауера. Різновиди брандмауерів. Брандмауер, або міжмережевий екран – це «напівпроникна мембрана», яка розташовується між внутрішнім сегментом мережі і зовнішньою мережею або іншими сегментами мережі «Інтернет», і контролює всі інформаційні потоки у внутрішній сегмент та з нього. Контроль трафіку полягає в його фільтрації, тобто у вибірковому пропущенні через екран, а іноді і з виконанням спеціальних перетворень і формуванням сповіщень для відправника, якщо його даним у пропуску відмовлено. Фільтрація здійснюється на підставі набору умов, попередньо завантажених в брандмауер, і відображає концепцію інформаційної безпеки корпорації. Брандмауери можуть бути виконані у вигляді як апаратного, так і програмного комплексу,

записаного в комутуючий пристрій або сервер доступу (сервер-шлюз, просто сервер, хост-комп'ютер і т.д.), вбудованого в операційну систему.

Робота брандмауера полягає в аналізі структури і вмісту інформаційних пакетів, що надходять із зовнішньої мережі, і залежно від результатів аналізу пропускає пакети інформації у внутрішню мережу (сегмент мережі) або їх відфільтровує.

Ефективність роботи міжмережевого екрана, що працює під управлінням Windows, зумовлена тим, що він повністю заміщає реалізований стек протоколів TCP\IP, і тому порушувати його роботу з допомогою спотворення протоколів зовнішньої мережі (що часто роблять хакери) неможливо.

Міжмережеві екрани зазвичай виконують такі функції:

- фізичне відділення робочих станцій і серверів внутрішнього сегмента мережі (внутрішньої підмережі) від зовнішніх каналів зв'язку;
- багатоетапну ідентифікацію запитів, що надходять в мережу (ідентифікація серверів, вузлів зв'язку про інших компонентів зовнішньої мережі);
- перевірку повноважень і прав доступу користувача до внутрішніх ресурсів мережі;
- реєстрацію всіх запитів до компонентів внутрішньої підмережі ззовні;
- контроль цілісності програмного забезпечення і даних;
- економію адресного простору мережі (у внутрішній підмережі може використовуватися локальна система адресації серверів);
- приховування IP адрес внутрішніх серверів з метою захисту від хакерів.

Брандмауери можуть працювати на різних рівнях протоколів моделі OSI.

На мережевому рівні виконується фільтрація вступників пакетів, заснована на IP адресі. На транспортному рівні фільтрація припустима ще й за номерами портів TCP і прапорів. На прикладному рівні може виконуватися аналіз прикладних протоколів (FTP, HTTP, SMTP і т.д.) і контроль за змістом потоків даних.

Розглянемо основні типи брандмауера.

Пакетні фільтри здійснюють аналіз інформації мережного і транспортного рівнів моделі OSI. Це мережні адреси (наприклад, IP) відправника та отримувача пакета номера портів відправника й отримувача, прапори протоколу TCP, опції IP, типи ICMP. Зазвичай пакетні фільтри організовуються засобами маршрутизаторів. Часто використовуються штатні засоби операційних систем.

Шлюзи рівня з'єднання. Цей і наступний тип брандмауера заснований на використанні так званого принципу посередництва, тобто запит приймається брандмауером, аналізується і тільки потім перенаправляється реальному серверу. Перш ніж дозволити встановлення з'єднання TCP між комп'ютерами внутрішньої і зовнішньої мережі, посередники рівня з'єднання спочатку як мінімум реєструють клієнта. При цьому неважливо, з якого боку цей клієнт знаходиться. При позитивному результаті реєстрації між зовнішнім і внутрішнім комп'ютерами організовується віртуальний канал, по якому пакети передаються між мережами.

Найбільш відомим прикладом шлюзу рівня з'єднання можна вважати шлюз з перетворенням IP-адрес (Network Address Translation, NAT).

Шлюзи прикладного рівня. Шлюзи прикладного рівня (application – level proxy), що часто називаються proxy-серверами, контролюють і фільтрують інформацію на прикладному рівні моделі OSI. Вони розрізняються по протоколах прикладного рівня, що підтримуються. Найчастіше підтримуються служби Web (HTTP), ftp, SMTP, POP3 I MAP, NNTP, Gopher, Telnet, DNS, RealAudio/RealVideo. Коли клієнт внутрішньої мережі звертається, наприклад, до сервера Web, то його запит потрапляє до посередника Web (чи перехоплюється ним). Останній встановлює зв'язок з сервером від імені клієнта, а отриману інформацію передає клієнтові. Для зовнішнього сервера посередник виступає клієнтом, а для внутрішнього клієнта – в якості сервера Web. Аналогічно посередник може працювати і у разі зовнішнього клієнта і внутрішнього сервера.

Технології Proxy і Stateful inspection. У розглянутих вище типах брандмауера, що припускають посередництво при встановленні з'єднання (шлюзах рівня з'єднання і

прикладного), реалізована так звана технологія Proxy. Ця технологія значно поширена і застосовується в таких відомих моделях брандмауера, як Microsoft Proxy Server і CyberGuard Firewall .

Проте для вироблення остаточних рішень про дозвіл того або іншого з'єднання для служб TCP/IP (тобто пропустити, заборонити, Автентифікувати, зробити запис про це в журналі), брандмауер повинен вміти отримувати, зберігати, витягати і маніпулювати інформацією з усіх рівнів мережної семирівневої моделі та з інших додатків.

Недостатньо тільки переглядати окремі пакети. Інформація про стан, береться з тих з'єднань, що мали місце раніше та інших додатків, використовується для ухвалення остаточного рішення про поточну спробу встановлення з'єднання. Залежно від типу пакета, що перевіряється, для ухвалення рішення важливими можуть бути, як поточний стан з'єднання, якому він належить (отримане з його історії), так і стан додатка, що його використовує.

Таким чином, для забезпечення найвищого рівня безпеки брандмауер повинен вміти зчитувати, аналізувати і використати наступну інформацію:

- інформацію про з'єднання – інформацію з усіх семи рівнів моделі;
- стан з'єднання – стан, отриманий з попередніх пакетів;
- стан додатка – інформація про стан, отримана з інших додатків. Наприклад, коли-небудь авторизованому користувачеві був дозволений доступ через firewall тільки для дозволених типів мережних протоколів.

Крім того, брандмауер повинен уміти виконувати дії над інформацією, що передається, залежно від усіх вищевикладених факторів.

Stateful Inspection – технологія нового покоління, задовольняє усім вимогам до безпеки, наведеним вище.

Технологія інспекції пакетів з урахуванням стану протоколу на сьогодні є найбільш передовим методом контролю трафіка (вона розроблена і запатентована компанією Check Point Software Technologies).

Ця технологія дозволяє контролювати дані аж до рівня додатків, не вимагаючи при цьому окремого процесу-посередника (proxy) для кожного протоколу, що захищається, або мережної служби. В результаті досягаються високі показники продуктивності, висока гнучкість рішень і можливість швидко і досить просто адаптувати систему під нові потреби.

Ґрунтуючись на технології інспекції пакетів з урахуванням стану протоколу, брандмауер забезпечує найвищий рівень безпеки. Метод stateful inspection забезпечує збір інформації з пакетів даних, як комунікаційного, так і прикладного рівнів, що досягається збереженням і накопиченням її в спеціальних контекстних таблицях, які динамічно оновлюються. Такий підхід забезпечує максимально можливий рівень безпеки, контролюючи з'єднання на рівнях від 3 до 7 мережної моделі OSI, тоді як proxy посередники можуть контролювати з'єднання тільки на 5 – 7 рівнях.

Основні функції брандмауера.

Аналіз змісту пакетів. Механізми перевірки змісту фільтрованих інформаційних пакетів (Content Security) реалізовані у багатьох брандмауерах, розширюють функції інспекції даних до найвищого рівня забезпечення інформаційної безпеки. Ці механізми дозволяють захистити користувачів від різних ризиків, включаючи комп'ютерні віруси і шкідливі аплети Java і ActiveX.

Брандмауер представляє першу лінію оборони, забезпечуючи захист від вірусів шляхом запобігання їх проникненню в точці входу у внутрішню мережу організації. Більшість брандмауерів мають засоби, що дозволяють в реальному масштабі часу здійснювати декодування, декомпресію і розпаковування файлів (за протоколом FTP), що входять і виходять, Web-додатків (за протоколом HTTP), поштових повідомлень (за протоколом SMTP) та ін. Усі відправлені файли скануються і/або піддаються "карантину" відповідно до прийнятої політики безпеки. Деякі брандмауери можуть працювати спільно зі спеціалізованими антивірусними сканерами, передаючи їм дані для антивірусного контролю.

Підтримка поштового протоколу SMTP. SMTP – протокол був спочатку розроблений для забезпечення максимально гнучких можливостей взаємодії користувачів поштової

системи. Тоді передбачалося, що доступ до Інтернет користувачі отримують з різних географічних регіонів. Потім протокол був розширений можливостями підтримки передачі різного роду інформації у вигляді вкладень електронної пошти. В результаті виявилось, що досить складно забезпечити максимальну прозорість поштових з'єднань і при цьому захистити від зломщиків внутрішню мережу організації.

5.1.2. Політика безпеки під час роботи в мережі

Під час роботи в мережевому середовищі необхідно бути упевненим у тому, що секретні дані такими і залишаться, оскільки лише користувачі, що мають відповідні повноваження, зможуть одержати до них доступ. Однак важливо забезпечити захист не тільки конфіденційної інформації, але і функціонування мережі в цілому. Кожна мережа має потребу в захисті від навмисного чи випадкового ушкодження. Однак у користувачів не повинно бути труднощів під час виконання роботи.

Найбільшу *загрозу для безпеки* мережі мають:

- несанкціонований доступ;
- електронне підслуховування;
- навмисне чи ненавмисне ушкодження.

Несанкціонований доступ – це навмисне звертання користувача до даних, доступ до яких йому не дозволений, з метою їхнього читання, відновлення чи руйнування.

Рівень захисту мережі залежить від її призначення. Наприклад, мережа, що зберігає дані великого банку, вимагає більш могутнього захисту, ніж локальна мережа, що з'єднує комп'ютери невеликої громадської організації.

Політика безпеки. Для захисту мережі необхідно проводити певну політику, тобто дотримуватися набору правил і розпоряджень. Вироблення політики безпеки (*security policy*) – перший крок, який повинна зробити будь-яка організація, забезпечуючи захист своїх даних. Політика встановлює «генеральну лінію», спираючись на яку і адміністратор, і користувачі будуть вносити зміни, знаходити вихід з позаштатних ситуацій при розширенні мережі.

Адміністратор повинен навчити користувачів мережі всім особливостям роботи і методам безпеки. Для цього він може скласти посібник, а в разі потреби – організувати навчання, особливо нових користувачів.

Керування доступом у Windows. Права користувача призначаються шляхом додавання його в одну з вбудованих груп, що містять набір уже призначених прав користувача. Однак у разі потреби можна створити нову групу і призначити їй певні права. Призначення прав групам здійснюється за допомогою групової політики. Користувачам, доданим у групу, автоматично надаються усі права, призначені групі. У Windows існують такі *групи*:

- *Адміністратори* – мають усі права та можливості в системі.
- *Оператори архіву* – можуть архівувати та відновлювати файли на комп'ютері незалежно від усіх дозволів, установлених для цих файлів.
- *Досвідчені користувачі* – можуть створювати локальні групи та облікові записи користувачів, а також видаляти користувачів з локальних груп, створених ними, змінювати та видаляти створені ними облікові записи.

Вони можуть керувати додаванням та видаленням користувачів з груп *Досвідчені користувачі, Користувачі, Гості*. Вони не мають прав на архівування та поновлення каталогів, завантаження та вивантаження драйверів, керування журналами безпеки та аудиту.

- *Користувачі* – можуть виконувати найбільш поширені завдання: запуск програм, друк документів, копіювання файлів і так далі. Користувачі мають право створювати локальні групи та змінювати групи, створені ними. Вони не можуть організовувати загальний доступ до ресурсів комп'ютера.
- *Гості* – призначена для запуску комп'ютера разовими користувачами. Члену цієї групи надаються обмежені можливості.
- *Реплікатор* – створена для підтримки функції реплікації (створення копії) каталогу.

Обліковий запис – запис користувача, що містить усі відомості, що визначають

користувача в операційній системі Windows. Це ім'я користувача і пароль, необхідні для входу користувача в систему, імена груп, членом яких користувач є, а також права і дозвіл, що він має під час роботи в системі і доступі до її ресурсів.

У Windows є *два вбудовані облікові записи* користувачів – *Адміністратор* і *Гість*, що створюються автоматично під час встановлення системи. Користувач з ім'ям Адміністратор є членом групи адміністраторів і може виконувати всі необхідні дії в мережі. Обліковий запис Гість призначений для тих, хто не має реального облікового запису. Цей обліковий запис не вимагає пароля. Він входить у вбудовану групу Гостей і має всі права, що привласнені цій групі.

З користувачем пов'язаний профіль користувача. *Профіль користувача* – набір параметрів середовища Windows, що завантажується під час входу користувача в систему. Він містить усі параметри настроювання середовища Windows, доступні для користувача, у тому числі групи програм, колір екрана, мережеві підключення дисків і принтерів.

Немаловажне значення *має контроль подій*, що відбуваються в мережі, оскільки в цих умовах зловмисник не настільки помітний і має досить часу і ресурсів для виконання своїх завдань. Цей процес відслідковує дії користувачів у мережі. Він є частиною захисту мережі, оскільки в *журналі безпеки* відбиті імена всіх користувачів, що працювали з конкретними ресурсами або намагалися одержати до них доступ.

5.2. Захист інформації в глобальних мережах

5.2.1 Короткі відомості про глобальні комп'ютерні мережі

До *локальних мереж Local Area Networks (LAN)* належать мережі комп'ютерів, зосереджені на невеликій території (звичайно в радіусі не більше ніж 1-2 км). У загальному випадку локальна мережа являє собою комунікаційну систему, що належить одній організації. Через короткі відстані в локальних мережах є можливість використання відносно дорогих високоякісних ліній зв'язку, які дозволяють, застосовуючи прості методи передачі даних, досягати високих швидкостей обміну даними приблизно 100 Мбіт/с. Через це послуги, що надаються локальними мережами, відрізняються широкою різноманітністю і звичайно передбачають реалізацію в режимі on-line.

Глобальні мережі Wide Area Networks (WAN) об'єднують комп'ютери, що територіально розосередилися: можуть знаходитися в різних містах і країнах. Оскільки прокладка високоякісних ліній зв'язку на великі відстані коштує дуже дорого, в глобальних мережах часто використовуються вже існуючі лінії зв'язку, спочатку призначені зовсім для інших цілей. Наприклад, багато глобальних мереж будуються на основі телефонних і телеграфних каналів загального призначення. Через низькі швидкості таких ліній зв'язку в глобальних мережах (десятки кілобіт в секунду) набір послуг, що надаються, звичайно обмежується передачею файлів переважно не в оперативному, а в фоновому режимі з використанням електронної пошти. Для стійкої передачі дискретних даних по неякісних лініях зв'язку застосовуються методи і обладнання, істотно відмінні від методів і обладнання, характерних для локальних мереж. Як правило, тут застосовуються складні процедури контролю і відновлення даних, оскільки найбільш типовий режим передачі даних по територіальному каналу зв'язку пов'язаний зі значними спотвореннями сигналів.

Міські мережі (або мережі мегаполісів) Metropolitan Area Networks (MAN) є менш поширеним типом мереж. Ці мережі з'явилися порівняно недавно. Вони призначені для обслуговування території мегаполіса. У той час як локальні мережі найкраще підходять для розділення ресурсів на коротких відстанях і широкомовних передач, а глобальні мережі забезпечують роботу на великих відстанях, але з обмеженою швидкістю і небагатим набором послуг, мережі мегаполісів займають деяке проміжне положення. Вони використовують цифрові магістральні лінії зв'язку, часто оптичноволоконні, з швидкостями від 45 Мбіт/с і призначені для зв'язку локальних мереж в масштабах міста і з'єднання локальних мереж з глобальними. Ці мережі спочатку були розроблені для передачі даних, але зараз вони підтримують і такі послуги, як відеоконференції й інтегральну передачу голосу і тексту. Розвиток технології мереж мегаполісів здійснювався місцевими телефонними компаніями. Історично склалося так, що місцеві телефонні компанії завжди мали слабкі технічні

можливості і через це не могли залучити великих клієнтів. Щоб подолати свою відсталість і зайняти гідне місце у світі локальних і глобальних мереж, місцеві підприємства зв'язку зайнялися розробкою мереж на основі найсучасніших технологій, наприклад технології комутації осередків SMDS або ATM. Мережі мегаполісів є суспільними мережами, і тому їх послуги коштують дешевше, ніж побудова власної (приватної) мережі в межах міста.

5.2.2 Характер проведення атак у глобальних мережах

Кіберзлочини поділяють на види залежно від об'єкта, від предмета посягання, залежно від способів скоєння і тому подібне.

За об'єктом посягання виділяють такі групи кіберзлочинів: злочини проти конфіденційності, цілісності та доступності комп'ютерних даних і комп'ютерних мереж, економічні комп'ютерні злочини, комп'ютерні злочини проти особистих прав і недоторканності приватної сфери, комп'ютерні злочини проти суспільних і державних інтересів. Проте варто зазначити, що багато кіберзлочинів зазіхають відразу на декілька об'єктів: наприклад, незаконне перехоплення приватних електронних комунікацій зазіхає на недоторканність приватної сфери і на конфіденційність комп'ютерних даних, комп'ютерне шахрайство – на власність і на цілісність комп'ютерних даних тощо.

Найбільш поширена класифікація кіберзлочинів в цей час ґрунтується на структурі Конвенції Ради Європи про кіберзлочинність. Спочатку кіберзлочини поділяли на чотири групи (потім був прийнятий додатковий протокол, і тепер груп – п'ять). На сьогодні ця класифікація є «еталоном», оскільки наявні міжнародні та регіональні документи, а також наукова практика.

У першу групу виділено злочини проти конфіденційності, цілісності та доступності комп'ютерних даних і систем, такі як незаконний доступ, незаконне перехоплення, втручання в дані, втручання в систему.

У другу групу входять злочини, пов'язані з використанням комп'ютера, як засобу скоєння злочинів, а саме як засобу маніпуляцій з інформацією. У цю групу входять комп'ютерне шахрайство та комп'ютерне підроблення.

Третя група – злочини, пов'язані з контентом, тобто з вмістом даних, розміщених в комп'ютерних мережах. Найбільш поширений і найбільш караний практично у всіх державах вид цих кіберзлочинів – злочини, пов'язані з дитячою порнографією.

У четверту групу увійшли злочини, пов'язані з порушенням авторського права і суміжних прав, при цьому встановлення таких правопорушень затверджено документом і належить до компетенції національних законодавств держав.

П'ята група злочинів зафіксована в окремому протоколі – це акти расизму та ксенофобії, вчинені за допомогою комп'ютерних мереж.

Реагування на інциденти в глобальній мережі. Боротьба з кіберзлочинністю неможлива без глибокого розуміння і правових питань регулювання інформаційних мереж. Саме аналіз взаємозв'язку між технічними характеристиками мережі і зумовленими цими характеристиками правовими і соціальними труднощами, з якими стикаються законодавці та правоохоронні органи, є першим кроком до можливого вироблення механізмів адекватного реагування на розвиток і зростання кіберзлочинності.

Відсутність механізмів контролю. Основна проблема боротьби зі злочинністю в мережі «Інтернет» полягає в транснаціональності самої мережі та відсутності механізмів контролю, необхідних для правозастосування. Коли мережа «Інтернет» створювалася технологічно як структура без ієрархії і без якогось «ядра», зруйнувавши які можна було б паралізувати її роботу, навряд чи хтось міг уявити масштаби розвитку проєкту, спочатку не призначеного для широкої аудиторії. Основною метою створення цієї мережі була стійкість до атак ззовні, і навряд чи хтось міг передбачити подальший масштаб її розвитку та її соціальну та економічну роль у майбутньому. Саме відсутність розроблених механізмів контролю мережі зсередини укупі з її доступністю і легкістю використання стало однією з глобальних проблем інформаційного співтовариства: децентралізована структура мережі і відсутність національних кордонів у кіберпросторі зумовили можливості для зростання злочинності та на роки відклали розробку механізмів соціального та правового контролю у сфері

використання інформаційних мереж для вчинення злочинів.

В останні роки інформаційні мережі розвиваються занадто швидко, щоб існуючі механізми контролю встигали реагувати на нові проблеми. Хмарна обробка даних, автоматизація атак, вразливість персональної інформації в соціальних мережах: поширення так званої «інформаційної зброї», прикладом якого є вірус Stuxnet, розроблений, на думку фахівців, для атак на ядерну промисловість Ірану, але при цьому заподіяв чималої шкоди інфраструктурі багатьох інших країн – на всі ці проблеми правове регулювання поки не може знайти адекватної відповіді.

Кількість користувачів. Як вже було зазначено вище, із збільшенням кількості користувачів зростають такі фактори ризику: залежність суспільства від інформаційних технологій, що, у свою чергу, зумовлює його вразливість до різних інформаційних зазіхань; збільшується можливість використання мережі для вчинення злочинів, а також зростає потенційна можливість стати жертвою використання інформаційних технологій в злочинних цілях. Водночас вчинення злочину не вимагає великих зусиль і витрат – достатньо мати комп'ютер, програмне забезпечення та підключення до інформаційної мережі. Не потрібно навіть глибоких технічних знань: існують спеціальні форуми, на яких можна придбати програмне забезпечення для вчинення злочинів, вкрадені номери кредитних карток й ідентифікаційні дані користувачів, а також скористатися послугами з допомоги в здійсненні електронних розкрадань і атак на комп'ютерні системи як в цілому, так і на окремих стадіях вчинення злочинів.

Автоматизація та швидкість використання. Комп'ютерні дані можуть бути передані з однієї точки світу в іншу за кілька секунд. До того ж, практично будь-яка передача даних у мережі зазвичай включає декілька країн, оскільки інформація розбивається на частини і йде по найбільш зручних та доступних каналах. Контролювати передачу даних з урахуванням їх обсягу та кількості користувачів дуже важко, якщо не неможливо. Злочинець, потерпілий, сервер з необхідною інформацією можуть перебувати в різних країнах і на різних континентах, що вимагає співпраці правоохоронних органів декількох країн під час розслідування злочину.

Автоматизація збільшує ризик здійснення численних злочинів без особливих фінансових і тимчасових витрат. До того ж, вона дозволяє злочинцям акумулювати більший фінансовий прибуток шляхом розкрадання невеликих сум у тисячі користувачів, що створює проблеми виявлення злочинів (власник банківського рахунку може просто не помітити зникнення фінансових коштів) і порушення кримінальних справ. Наприклад, якщо той же власник банківського рахунку звернеться із заявою про зникнення невеликої суми, правоохоронним органам досить важко оцінити масштаб діяльності тих, хто вчинив розкрадання, оскільки шкода, завдана одному потерпілому, дуже мала, в той час як правопорушники шляхом акумуляції цих невеликих сум можуть отримати неабиякий прибуток.

Анонімність мережі «Інтернет», вразливість бездротового доступу і використання проксі серверів істотно ускладнюють виявлення злочинців: для вчинення злочину може використовуватися «ланцюжок» серверів, злочини можуть бути вчинені шляхом виходу в Інтернет через точки загального доступу, такі як інтернет-кафе, технології дозволяють також «зламати» доступ в чужу бездротову мережу Wi-Fi. Отже, існує достатньо способів ускладнити розслідування злочинів.

Проблема територіальної юрисдикції в кіберпросторі та правового співробітництва. Розслідування злочинів в інформаційних мережах зазвичай вимагає швидкого аналізу та збереження комп'ютерних даних, які дуже вразливі за своєю природою і можуть бути швидко знищені. У цій ситуації традиційні механізми правової взаємодопомоги і принцип суверенітету, одним з проявів якого є те, що тільки правоохоронні органи держави можуть проводити слідчі дії, вимагають безліч формальних погоджень, роблять розслідування транснаціональних кіберзлочинів проблематичним. Окрім співробітництва правоохоронних органів, яке вимагає тимчасових витрат і дотримання безлічі формальностей, виникає також питання дотримання фундаментального принципу nullum crimen, nulla poena sine lege, коли необхідна подвійна криміналізація діяння: як у країні, з території якої діяв правопорушник, так і в державі, де знаходиться потерпілий. Різниця у криміналізації діянь, відмінності у

визначенні тяжкості вчиненого діяння, особливо у сфері релігійних злочинів і злочинів проти громадського порядку, у сфері нелегального контенту, в екстремістських злочинах значно ускладнюють процес співробітництва правоохоронних органів, іноді унеможливлюють його.

Отже, ефективний контроль негативних явищ у кібепросторі, таких як злочинність, вимагає більш інтенсивного міжнародного співробітництва, ніж існуючі заходи по боротьбі з будь-якими іншими формами транснаціональної злочинності. Саме тому, крім гармонізації кримінально-правових норм, потрібна гармонізація процесуальних інструментів і вироблення нових механізмів міжнародного співробітництва. Важливу роль у боротьбі з кіберзлочинністю відіграють міжнародні угоди у відповідній сфері, такі як Конвенція Ради Європи про кіберзлочинність, рішення Ради Європейського Союзу, Модельний Закон Співдружності Націй про комп'ютерні злочини 2002 р., Модельний Закон країн Карибського Басейну про кіберзлочинність (проєкт HIPCAR), спільний проєкт Європейського Союзу і Міжнародного Союзу Електрозв'язку для держав Тихоокеанського регіону (проєкт ICB4PAC), проєкт ООН з розробки законодавства в галузі кіберзлочинності для країн Африки (проєкт ESCWA) та інші.

Всі зазначені інструменти не є за своєю суттю універсальними міжнародними інструментами.

Віддалені атаки на обчислювальні мережі. Атака на відмову в обслуговуванні, розподілена атака на відмову в обслуговуванні (англ. *DoS attack, DDoS attack, (Distributed) Denial-of-service attack*) – напад на комп'ютерну систему з наміром зробити комп'ютерні ресурси недоступними користувачам, для яких комп'ютерна система була призначена.

Одним із найпоширеніших методів нападу є насичення атакованого комп'ютера або мережевого устаткування великою кількістю зовнішніх запитів (часто безглуздих або неправильно сформульованих). Отож атаковане устаткування не може відповісти користувачам або відповідає настільки повільно, що стає фактично недоступним. Взагалі відмова сервісу здійснюється:

- примусом атакованого устаткування до зупинки роботи програмного забезпечення/устаткування або до витрат наявних ресурсів, через що устаткування не може продовжувати роботу;
- Зайняттям комунікаційних каналів між користувачами і атакованим устаткуванням, і як наслідок – якість сполучення не відповідає вимогам.

Якщо атака відбувається одночасно з великої кількості IP-адрес, то її називають *розподіленою.*

Віддалені DoS-атаки поділяють на два види:

1. Віддалена експлуатація помилок в ПЗ з метою довести його до неробочого стану.
2. Flood – посилка на адресу жертви величезної кількості безглуздих (рідше – осмислених) пакетів. Метою флуду може бути канал зв'язку або ресурси машини. У першому випадку потік пакетів займає весь пропускний канал і не дає машині, що атакується, можливості обробляти легальні запити. У другому – ресурси машини захоплюються за допомогою багаторазового і дуже частого звернення до якого-небудь сервісу, що виконує складну, ресурсоємку операцію. Це може бути, наприклад, тривале звернення до одного з активних компонентів (скрипту) вебсервера. Сервер витрачає всі ресурси машини на обробку запитів, що атакують, а користувачам доводиться чекати.

5.2.3 Захист під час використання WWW (World Wide Web)

Розглянемо інформаційні небезпеки, які з'являються під час використання як соціальних мереж, так і мережі «Інтернет» в цілому. Основним джерелом цих небезпек є діяльність хакерів. Одні зловмисники прагнуть одержати персональну інформацію для отримання вигоди, інші – обирають об'єктом атак комп'ютерну систему та намагаються вивести її з ладу або використати для приховування своїх шкідливих дій.

Для зламу акаунта користувача хакеру необхідно докласти зусиль, щоб дізнатися пароль, іншими словами – зламати його. Найбільш поширеними способами реалізації зламу паролю є такі технології.

Брутфорс – метод пошуку та зламу пароля, який дозволяє перебрати всі теоретично

можливі варіанти, складені з певного набору символів. До нього також належить атака перебору пароля за словником або ручного підбору часто використовуваних простих паролів.

Для захисту від цього способу зламу у паролі не слід вказувати дату народження, номери телефонів, ім'я, кличку домашньої тварини, прості відомі паролі і будь-які інші дані, потенційно відомі деякому колу осіб. У паролі слід указувати певний набір символів, який практично неможливо вгадати. Критерієм складності пароля є наявність символів з кожного пункту такого переліку:

- символи a.. .z (лише малі літери);
- символи Aa.. .Zz (великі та малі літери);
- цифри;
- недруковані ASCII-символи, літери інших алфавітів;
- спеціальні символи :;% "№ *? тощо.

Соціальна інженерія або соціотехніка. Цей метод ґрунтується на довірі користувача. Для цього використовуються сфальсифіковані сайти та фіктивні електронні повідомлення від імені реальних компаній з проханням надати особисту інформацію.

Головним захистом у цьому разі є пильність користувача. Нікому не можна повідомляти або надсилати паролі.

Кейлогери – це програмний продукт або апаратний пристрій, що реєструє кожне натискання кнопки миші або клавіші на клавіатурі комп'ютера та записує у файл разом з датою та часом натискання. Отже, у зловмисника буде пароль у головному вигляді.

Гарантованим захистом від цього методу може слугувати лише вихід в мережу «Інтернет» і введення пароля з власного або надійного, перевіреного комп'ютера.

Програмний метод зламу. Цей метод доступний хакерам і полягає в пошуку помилок у коді сайтів, що дозволяють отримати доступ до бази даних з паролями.

- такому разі дані можуть відновити лише адміністратори.

Фішинг – технологія інтернет-шахрайства з метою отримання ідентифікаційних даних користувачів. Реалізується за допомогою заманювання їх на підставні сайти, які є точною або майже точною копією оригіналу.

Для захисту виробники основних інтернет-браузерів домовилися про застосування однакових способів інформування користувачів про те, що вони відкрили підозрілий сайт, який може належати шахраям. Нові версії браузерів уже мають таку можливість, яка відповідно іменується «антифішинг». Від користувачів вимагається лише вчасно оновлювати версії браузерів.

Спамом називають небажану електронну пошту, тобто пошту, що надходить без згоди користувача. Він може прикріплюватись до всіх повідомлень у вигляді посилання на сторонній сайт.

У цьому разі необхідно змінити браузер на Mozilla Firefox або Opera, які блокують заданий користувачем спам. А також варто дотримуватися запобіжних заходів, які не дозволять спамерам дізнатися адресу електронної пошти користувача:

- не варто без потреби публікувати адресу електронної пошти на вебсайтах чи в групах соціальних мереж;
- не потрібно реєструватися на підозрілих сайтах, натомість краще вказати спеціально для цього створену адресу;
- ніколи не відповідати на спам і не переходити за посиланням, які містяться в ньому, оскільки це буде підтвердженням використання цієї електронної адреси і збільшить надходження спаму;
- обираючи ім'я електронної пошти, варто створювати його довгим і незручним для вгадування.

Віруси – малі за розміром програми, які поширюються, копіюючи самих себе. Вони потрапляють до комп'ютерної системи і деякий час можуть себе не проявляти, і лише після настання певної дати чи події активізуються та завдають їй шкоди.

Рекомендується використовувати декілька антивірусних пакетів одночасно, щоденно оновлювати антивірусні бази та встановлювати найновіші версії ліцензійного програмного

забезпечення.

Про користь та ризики соціальних мереж можна довго розмірковувати, але в будь-якому разі це явище існує. Тому необхідно допомогти користувачеві зробити мінімум помилок під час роботи у відповідній мережі.

Варто розуміти, що профіль у будь-якій соціальній мережі вразливий, особливо при використанні стандартних параметрів. Тому слід дотримуватись таких рекомендацій:

- подбайте про надійний пароль для профілю;
- будьте обережні у разі встановлення додатків від сторонніх розробників, у жодному разі не встановлюйте додатки з джерел, яким не довіряєте;
- приймайте пропозиції про дружбу тільки від тих людей, яких знаєте особисто і безпосередньо;
- ретельно прочитавши політику конфіденційності, обмежте особисту інформацію, яку збираєтесь зробити загальнодоступною;
- перевіряйте інформацію, яку надсилаєте на сайт;
- завжди використовуйте для кожного форуму, сайту та поштової скриньки різні паролі, інакше шанс втрати всіх акаунтів збільшується при крадіжці одного пароля.

Потенційні проблеми під час використання електронної пошти. Електронна пошта є повсюдною послугою, однак не можна не визнати, що рівень захисту даних в системі електронної пошти впливає на загальний рівень інформаційної безпеки організації, а отже, і на ефективність її діяльності. Це зумовлює важливість створення надійного захисту для цього виду комунікацій.

Більшість проблем, які з'являються у користувачів електронної пошти (спам, віруси, різноманітні атаки на конфіденційність листів і т.д.), пов'язані з недостатнім захистом сучасних поштових систем.

З цими проблемами доводиться мати справу і користувачам загальнодоступних публічних систем, і організаціям. Практика показує, що одномоментне вирішення проблеми захисту електронної пошти неможливо. Рівень захисту електронної пошти, цілком задовільний вчора, сьогодні може виявитися недостатнім. Для того щоб захист електронної пошти був на максимально можливому рівні, а досягнення цього рівня не вимагало надмірних зусиль і витрат, необхідний систематичний, комплексний, з урахуванням усіх загроз підхід до вирішення цієї проблеми.Боротьба зі спамом та вірусами. Сьогодні є безліч програмних продуктів, у тому і числі і безкоштовних, призначених для боротьби з цією загрозою. Щодо боротьби зі спамом, тут можливі кілька варіантів захисту.

Можна реалізувати систему фільтрів, що дозволяють відсікати вхідну кореспонденцію за адресою, темою чи змістом листа. Фільтри зазвичай розміщуються на клієнтській стороні, і користувач сам може задавати необхідні параметри. Наприклад, системи Spam Buster виробництва компанії Contact Plus, MailWasher, Active Email Monitor (VicMan Software), eMailTrackerPro (Visualware), Spamkiller (Novasoft) та інші. Крім фільтрації спаму, такі програми можуть виконувати функції очищення поштової скриньки, перевірки пошти, читання заголовків листів тощо.

Система фільтрів встановлюється на поштовому сервері; в такому разі листи, що нагадують спам, відсікаються ще до потрапляння в скриньку користувача. Також може бути реалізований захист на основі <Спам-листів>, що містять список інтернет-провайдерів, з адрес яких здійснюється несанкціонована розсилка рекламного характеру. Прикладами можуть бути служба Mail-Filtering Service проєкту Mail Abuse Prevention Project і Realtime Blackhole List, база даних по відкритих поштових серверах (під відкритістю в цьому разі розуміється відсутність адекватного адміністрування, що призводить до неконтрольованих розсилок спаму через такі поштові сервери).

Огляди подібних продуктів регулярно публікуються. Створити систему антивірусного і антиспамового захисту загалом нескладно як для користувача загальнодоступного комунікаційного середовища, так і для IT-підрозділу організації, що розгорнула на своїй обчислювальній інфраструктурі корпоративну поштову систему. Після вибору і установки засобів антивірусного і антиспамового захисту найголовніше – їх акуратне і своєчасне оновлення. Головне пам'ятати, що вірусописьменники не зупиняються на досягнутому, а

спамери з кожним днем стають все активніше.

5.2.4 Захист електронних листів та поштових систем

Захист від фальшивих адрес. Від цього можна захиститися за допомогою використання шифрування для приєднання до листів електронних підписів. Одним популярним методом є використання шифрування з відкритими ключами. Односпрямована хеш-функція листи шифрує, використовуючи секретний ключ відправника. Одержувач використовує відкритий ключ відправника для розшифровки хеш-функції і порівнює його з хеш-функцією, розрахованою за отриманим повідомленням. Це гарантує, що повідомлення насправді написано відправником, і не було змінено в дорозі. Уряд США вимагає використання алгоритму Secure Hash Algorithm (SHA) і Digital Signature Standard там, де це можливо. А найпопулярніші комерційні програми використовують алгоритми RC2, RC4, або RC5 фірми RSA.

*Захист від перехоплення.*_Від нього можна захиститися за допомогою шифрування вмісту повідомлення або каналу, по якому він передається. Якщо канал зв'язку зашифрований, то системні адміністратори на обох його кінцях таки можуть читати або змінювати повідомлення. Було запропоновано багато різних схем шифрування електронної пошти, але жодна з них не стала масовою. Одним з найпопулярніших додатків є PGP. У минулому використання PGP було проблематичним, оскільки в ньому використовувалося шифрування, яке підпадало під заборону на експорт із США. Комерційна версія PGP включає в себе плагіни для декількох популярних поштових програм, що робить її особливо зручною для включення до листа електронного підпису та шифрування листа клієнтом. Останні версії PGP використовують ліцензовану версію алгоритму шифрування з відкритими ключами RSA.

Коректне використання електронної пошти. Всі службовці повинні використовувати електронну пошту так само, як і будь-яке інший офіційний засіб організації. З цього випливає, що коли лист надсилається, то як відправник, так і одержувач повинен гарантувати, що взаємодія між ними здійснюється згідно з прийнятими правилами взаємодії. Взаємодія за допомогою пошти не повинна бути неетичною, не повинна сприйматися як конфліктна ситуація або містити конфіденційну інформацію.

Захист листів, поштових серверів і програм повинен відповідати важливості інформації, переданої по мережах. Як правило, повинно здійснюватися централізоване управління сервісами електронної пошти. Повинна бути розроблена політика, в якій вказувався б потрібний рівень захисту.

РОЗДІЛ 2
КРИПТОГРАФІЧНИЙ ЗАХИСТ ІНФОРМАЦІЇ
6 Основи крипторгафії
6.1. Історія виникнення криптографії

Проблеми захисту інформації хвилювали людство з незапам'ятних часів. Необхідність захисту інформації виникла з потреб таємної передачі, як військових, так і дипломатичних повідомлень. Наприклад, античні спартанці шифрували свої військові повідомлення. У китайців простий запис повідомлення за допомогою ієрогліфів робив його таємним для чужоземців.

Перші канали зв'язку були дуже простими. Їх організовували, використовуючи надійних кур'єрів. Безпека таких систем зв'язку залежала як від надійності кур'єра, так і від його здатності не попадати в ситуації, при яких могло мати місце розкриття повідомлення.

За твердженням спеціалістів - криптографія за віком ровесниця єгипетських пірамід. У документах Стародавніх цивілізацій: Індії, Єгипту, Месопотамії - є відомості про системи і способи створення шифрованих листів. У стародавніх релігійних книгах Індії вказується, що сам Будда знав декілька способів письма, серед яких були присутні шифри перестановок (в сучасній класифікації). Один із стародавніх шифрованих текстів в Месопотамії (20 ст. до н.е.) являє собою глиняну табличку, яка містить рецепт виготовлення глазурі в гончарному виробництві, в якому ігноруються деякі голосні і приголосні і використовуються числа замість імен.

Таблиця 6.1 Етапи розвитку криптосистем

Етап	Час	Особливості
I	V – IV ст. до н.е.	Виникнення перших шифрувальних пристроїв (Скитала, табличка Енея і т.п.)
II	I ст. до н.е.	Початок застосування шифрованого зв'язку в системі органів влади (шифр Цезаря і т.п.)
III	XV-XVI ст. (Відродження)	Розвиток наукових методів криптографії і крипто аналізу (метод частотного аналізу, винайдення поліалфавітних шифрів, поворотної решітки)
IV	XVII-XVIII ст. (ера «чорних кабінетів»)	Використання номенклаторів як основного засобу шифрування
V	XIX ст.	Винайдення гаміювання
VI	XX ст.	Винайдення колісних шифраторів (Енігма)
VII	кінець XX ст.	Народження «нової криптографії»

Так ще в 5-му столітть до нашої ери греки застосовували спеціальний шифрувальний засіб. За описанням, він складався з двох паличок однакової довжини і товщини. Одну паличку залишали собі, а другу віддавали від'їжджаючому. Ці палички називали **скиталами**. Коли товаришам потрібно було повідомити якусь таємницю, вони вирізали довгу і вузьку смужку папіруса, намотували її на скіталу, не залишаючи ніяких проміжків, так щоб вся поверхня скітали була зайнята цією смужкою. Потім, залишаючи папірус так як він є, писали на ньому повідомлення, далі знімали смужку і без палички передавали адресатові. Так як букви на ній були розкидані, то прочитати написане можна було лише за допомогою такої ж скітали.

Аристотелю належить спосіб дешифрування цього шифру. Необхідно виготовити довгий конус, і починаючи з основи, обмотувати його смужкою із шифрованим повідомленням, поступово зсуваючи її до вершини. В якийсь момент почнуть читатись кусочки повідомлення. Так можна визначити розмір скітали.

Іншим шифрувальним засобом часів Спарти була **табличка Енея**. На невеличкій горизонтальній табличці розташовувався алфавіт, а по її боковим сторонам розташовувались виїмки для намотування нитки. При шифруванні нитка закріплювалась з однієї із сторін

таблички і намотувалась на неї. На нитці робились відмітки (наприклад, вузлики) в місцях, що знаходились напроти букв даного тексту. Після шифрування нитка змотувалась і доставлялась адресату. Цей шифр був досить надійним: історії не відомі факти його дешифрування.

У Стародавній Греції (2 ст. до н.е.) був відомий шифр, що називався квадрат Полібія. Цей засіб представляв собою квадрат 5 на 5, стовпці і рядки якого були пронумеровані цифрами від 1 до 5. В кожну клітинку цього квадрату записувалась одна буква (в грецькому алфавіті одна клітина залишалась пустою, а в латинському - в одну клітинку записувались дві букви і, j). У результаті кожній букві відповідала пара чисел і шифроване повідомлення перетворювалось в послідовність пар чисел.

Приклад

13 34 22 24 44 34 15 42 22 34 43 45 32.

Це повідомлення записане за допомогою квадрата Полібія, в якому букви розташовані в алфавітному порядку.

Таблиця 6.2 Заміна букв

„Cogito, ergo sum” - „ Я думаю, значить існую”.

В 1-му ст.н.е. Ю. Цезарь під час війни з Галами, переписуючись зі своїми друзями в Римі, заміняв в повідомленні першу букву латинського алфавіту на четверту, другу - на п’яту і т. д., а останню на третю.

ABCDEFGHIJKLMNOPQRSTUVWXYZ

DEFGHIJKLMNOPQRSTUVWXYZABC

Обидва ці методи належать до класу шифру, що називається підстановка чи проста заміна.

З часів Цезаря до XIV-XV століть криптографія розвивалась, але практика шифрування зберігалась у глибокій таємниці. Так, в часи хрестових походів шифрувальники Папи Римського після року роботи підлягали фізичному знищенню. Тому про цей період майже нічого невідомо.

Розвитком шифру простої заміни став шифр Б. Віженера (XVI ст., Франція). Криптографією займалася багато відомих математиків, таких як Ф. Вієт, Д. Кардано, В. Лейбніц і, нарешті, Ф. Бекон, який запропонував двійкове кодування латинського алфавіту.

Також в епоху Відродження почали розвиватись наукові методи криптографії і крипто аналізу. Зокрема для дешифрування шифрів став використовуватись **метод частотного аналізу** зустрічає мості букв, для шифрування почали використовуватись **шифри багатозначної заміни** (омофони), отримали поширення **багатоалфавітні шифри** (шифр Тритемія, шифр Віженера), в якості засобу шифрування почала використовуватись **поворотна решітка**.

Поворотна решітка представляла собою лист з твердого матеріалу з прорізями. Накладаючи таку решітку на лист паперу, можна записувати у вирізи таємне повідомлення. Після цього, знявши решітку, треба було заповнити вільні місця, що залишились, маскуючим текстом.

Подібним стеганографічним методом маскування повідомлень користувались багато історичних осіб, зокрема, кардинал Рішіл’є у Франції. Він використовував в якості решітки прямокутник розміру 7x10:

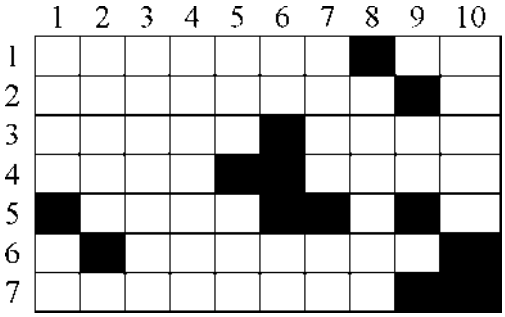


Рис. 6.1 Решітка Рішильє

В історії криптографії XVII-XVIII ст. називають ерою «чорних кабінетів». В цей період в багатьох державах Європи з'явились дешифрувальні підрозділи, які і називались «чорними кабінетами». Перший з них був створений у Франції за ініціативою кардинала Рішіл'є.

Характерним для цього періоду є широке розповсюдження шифрів, які називались **номенклаторами**. Номерклатори поєднували в собі просту заміну і код. В якості прикладу номенклатура наведемо «малий шифр», що використовувався в наполеонівській армії (Таблиця 6.3).

Таблиця 6.3. «Малий шифр», що використовувався в наполеонівській армії

A-15 AR-25 AL-39	J-87 Ш-123	Я-169 ЯА-146 ЯЕ-126 ЯІ-148
B-37 Ви-3 B0-35 BI-	K-?	8-167 8A-171 SE-177 8I-134
C-6 CA-32 CE-20	L-96 LU-103 LE-117	SO-168 8и-174
D-23 DE-52	LA-106	T-176 TI-145 T0-157
E-53 E8-82 ET-50	M-114 MA-107	и-138
F-55 FA-69 FE-58	N-115 ИЕ-94 N1-116	^164 ^УЕ-132 ^-161 VO-175
G-81 GA-51	0-90 OT-153	W, X, Y-?
H-85 HI-77	P-137 P0-152	Z-166
I-119 I8-122	Q-173 QUE-136	

Приклад застосування цього шифру наведений в Таблиці 6.4.

Таблиця 6.4. Приклад шифрування «малим шифром» слова NAPOLEON

N	A	P	O	L	E	O	N
115	15	137	90	96	53	90	115
або							
N	A	PO	LE	O	N		
115	15	152	117	90	115		

В простих нуменклаторах код складався з кількох десятків слів або фраз з двобуквеними кодовими позначеннями. З часом списки слів в номерклаторах збільшились до двох-трьох тисяч.

В цілому XVII-XVШ ст. не дали для криптографії нових ідей. Ера «чорних кабінетів» закінчилась в 40их роках XIX ст.

Промислова революція в розвинених країнах привела до створення шифрувальних машин. У кінці XVIII століття Т. Джефферсоном (майбутнім третім президентом США) були винайдені шифруючі колеса. Першу практично працюючу шифрувальну машину запропонував у 1917 році Г. Вернам. У тому ж році була винайдена роторна шифрувальна машина, що згодом випускалася фірмою Сименс під назвою «Енігма» (загадка) - основний супротивник криптографів Союзних держав у роки Другої світової війни.

Створення сучасних комп'ютерних систем і поява глобальних комп'ютерних мереж радикально змінила характер і діапазон проблем захисту інформації. У широко комп'ютеризованому і інформатизованому сучасному суспільстві володіння реальними

цінностями, керування ними, передача цінностей або доступ до них часто засновані на неопредметненій інформації, тобто на інформації, існування якої не обов'язково зв'язується з яким-небудь записом на фізичному носії. Тому досить важливо створювати і застосовувати ефективні засоби для реалізації всіх необхідних функцій, зв'язаних із забезпеченням конфіденційності і цілісності інформації.

Оскільки інформація може бути дуже коштовною або особливо важливою, можливі різноманітні зловмисні дії стосовно комп'ютерних систем, що зберігають, обробляють або передають таку інформацію. Наприклад, порушник може спробувати видати себе за іншого користувача системи, підслухати канал зв'язку або перехопити і змінити інформацію, якою обмінюються користувачі системи. Порушником може бути і користувач системи, що відмовляється від повідомлення, у дійсності сформованого ним, або який намагається стверджувати, що ним отримане повідомлення, яке в дійсності не передавалося. Зловмисник може спробувати розширити свої повноваження, щоб одержати доступ до інформації, до якої йому наданий тільки частковий доступ, або спробувати зруйнувати систему, несанкціоновано змінюючи права інших користувачів.

Для вирішення зазначених і інших подібних проблем не існує якогось одного технічного прийому або засобу. Однак, загальним у вирішенні багатьох з них є використання криптографії і криптоподібних перетворень інформації.

Протягом більш ніж тисячолітньої історії криптографії вона являла собою постійно оновлюючий й вдосконалюючий набір технічних прийомів шифрування і розшифрування, що зберігалися в строгому секреті.

Період розвитку криптології з старих часів до 1949 року прийнято називати ерою донаукової криптології, оскільки досягнення тих часів були засновані на інтуїції і не підкріплювалися доказами. Криптологією займалися тоді майже винятково як мистецтвом, а не як наукою. Звичайно, це не означає, що історія криптології тих часів не представляє для нас ніякого інтересу. Лише з початком другої світової війни криптологічні служби держав, що воювали усвідомили, що математики можуть внести вагомий вклад у розвиток криптології. Зокрема, в Англії в цей час був закликаний на службу як фахівець з криптології Алан Тьюринг. нових наукових ідей в цей час не з'явилося.

Майже половина XX ст. була пов'язана з використанням **колісних шифраторів**. Їх різні конструкції були запатентовані майже одночасно в різних країнах (Голандії, Німеччині, Швеції). Найбільш відомою шифромашиною цих часів була «Енігма», яка у довоєнний період і під час другої світової війни використовувалась в германській армії. Принцип її роботи такий. На екрані оператору показувалася буква, якою шифрувалася відповідна літера на клавіатурі. Те, якою буде зашифрована літера, залежало від початкової конфігурації коліс. Суть в тому, що існувало понад сто трильйонів можливих комбінацій коліс, і з часом набору тексту колеса зсувалися самі, так що шифр змінювався протягом усього повідомлення. Все Енігми були ідентичними, так що при однаковому початковому положенні коліс на двох різних машинах і текст виходив однаковий. У німецького командування були Енігми і список положень коліс на кожен день, так що вони могли з легкістю розшифровувати повідомлення один одного, але вороги не повідомляючи положень послання прочитати не могли.

Публікація в 1949 році статті К. Шеннона «Теорія зв'язку в секретних системах» стала початком нової ери наукової криптології із секретними ключами. У цій блискучій роботі К. Шеннон зв'язав криптографію з теорією інформації.

У другій половині XX ст. в зв'язку з розвитком елементної бази обчислювальної техніки з'явились **електронні шифратори**. Сьогодні для шифрування даних найбільш широко застосовують три види шифраторів: апаратні, програмно-апаратні та програмні. Їх основна відмінність полягає не тільки в способі реалізації шифрування і ступеня надійності захисту даних, але і в ціні. Найдешевші пристрої шифрування - програмні, потім йдуть програмно-апаратні засоби і, нарешті, найдорожчі - апаратні.

В 70-их роках народилась «нова криптографія» - **криптографія з відкритим ключем**. Криптографія почала широко використовуватись не тільки у військовій, дипломатичній, державній сферах, а також в комерційній, банківській та інших сферах. Вона не тільки

перестала бути секретним зводом прийомів шифрування-розшифрування, але і почала оформлятися в нову математичну теорію. За останні двадцять років відбулося значне підвищення активності в області розвитку криптографії та її застосування для вирішення проблем захисту інформації. Це викликано широким визнанням крайньої необхідності в засобах забезпечення захисту інформації у всіх областях діяльності широко інформатизованого людського співтовариства й обумовлено появою таких нових фундаментальних ідей, як асиметрична криптографія, стійкі протоколи, надійність яких заснована на гарантованій складності розв'язання математичних задач, і т.д.

6.2. Модель криптографічної системи

Для позначення всієї області секретного зв'язку використовується термін «**криптологія**», що походить від грецьких коренів «cryptos» - таємний і «logos» - повідомлення. Криптологія досить чітко може бути розділена на два напрямки: криптографію і криптоаналіз.

Задача **криптограф**а - забезпечити конфіденційність і аутентичність переданих повідомлень. Термін криптографія ввів Д. Валліс.

Задача **криптоаналітик**а - «зламати» систему захисту, розроблену криптографами. Він намагається розкрити зашифрований текст або видати підроблене повідомлення за справжнє.

Метою **криптографії** є приховання вмісту повідомлень за рахунок їх шифрування, на відміну від цього, при стеганографії, приховується сам факт існування таємного повідомлення.

Стеганографія - із грецького «тайнопис». Історично цей напрям з'явився першим, але потім у багатьох випадках був витісненим криптографією. Розвиток засобів обчислювальної техніки за останнє десятиліття дав новий поштовх для розвитку комп'ютерної стеганографії, з'явилось багато нових областей застосування. Повідомлення приховуються у цифрові дані, які мають аналогову природу, це - текст, звук, зображення, відео. Методи стеганографії дозволяють не тільки приховано передавати дані, але й успішно вирішувати завдання завадостійкості аутентифікації, захисту інформації від несанкціонованого копіювання, відстеження поширення інформації мережами зв'язку, пошуку інформації в мультимедійних базах даних.

Криптографічна система (криптосистема) - система секретного зв'язку, в якій зміст інформації, що передається, утаємничується за допомогою криптографічних перетворень; при цьому сам факт передачі інформації не приховується.

Криптографічні перетворення визначаються певним параметром, який називається ключ. Зазвичай ключ є буквеною або числовою послідовністю. Кожне криптографічне перетворення однозначно визначається ключем і описується певним криптографічним алгоритмом.

Криптографічними перетвореннями є:

Шифрування- процес перетворення вихідного тексту (Р) в зашифрований текст (С) за допомогою шифруючої функції (Е) з секретним ключем шифрування (Ке) у відповідності з обраним алгоритмом шифрування: С=Е Ке(Р).

Розшифрування - обернений шифруванню процес перетворення зашифрованого тексту (С) в вихідний текст (Р) за допомогою функції розшифрування (О) з секретним ключем розшифрування (К) у відповідності з обраним алгоритмом шифрування: $P=D_{Kd}(C)$.

Сімейство обернених перетворень защифрування і розшифрування називають **шифром**.

Алгоритми шифрування і розшифрування можуть відрізнятись, відповідно можуть розрізнятись і ключі шифрування і розшифрування.

В загальному випадку криптосистема має наступну структуру (Рис. 6.2):

Рис. 6.2. Структура криптосистеми

Робота криптосистеми може бути описана наступним чином:

1. З джерела ключів вибирається ключі (шифрування K_e і розшифрування K^ і відправляються по надійним каналам передаючій і приймаючій стороні.
2. До вихідного (або відкритого) повідомлення p, що призначене для передачі, застосовується алгоритм шифрування E_{Ke}, внаслідок чого отримується зашифроване повідомлення C=Еке(p).
3. Зашифроване повідомлення пересилається по каналу для обміну повідомленнями, який не вважається надійним (тобто зашифроване повідомлення може бути перехоплене порушником), приймаючій стороні.
4. На приймаючій стороні до зашифрованого повідомлення C застосовується обернене перетворення для отримання вихідного повідомлення p=DKd(C).

В класичній криптографії для шифрування і розшифрування використовувався один і той самий ключ: Ke=Kd=K. Такі криптосистеми отримали назву криптосистем з секретним ключом (secret key cryptosystems); сьогодні їх також називають симетричним криптосистемами (symmetric cryptosystems).

Зауважимо, що алгоритми шифрування і розшифрування E і D відкриті, і секретність вихідного тексту p в даному шифротекста C залежить від таємності ключа K.

Для сучасної криптографії революційним стало усвідомлення того факту, що ключі шифрування і розшифрування можуть не співпадати. Такі криптосистеми отримали назву асиметричних криптосистем (asymmetric cryptosystems).

Оскільки зашифроване повідомлення передається через канал, доступний противнику, можливе його перехоплення і «прочитання» особою, яка не має ключа. В цьому випадку говорять про **дешифрування** повідомлення.

Таким чином, терміши «розшифрування» і «дешифрування» слід розрізняти: при розшифровуванні ключ вважається відомим, тоді як при дешифруванні ключ невідомий.

Розшифрування здійснюється так само просто, як і шифрування. Дешифрування є значно складнішою задачею. Рівень складності цієї задачі і визначає здатність протистояти спробам противника заволодіти інформацією, яка захищається. В зв’язку з цим говорять про криптографічну стійкість шифру, розрізняючи більш і менш стійкі.

6.3. Принцип Керкхоффса

Основне правило криптографії - використовувати відкриті й опубліковані алгоритми та протоколи.

Вперше цей головний принцип був сформульований у 1883 році Агустом Керкхгоффсом (A.Kerckhoffs): в криптографічній системі єдиним секретом має залишатися ключ, сам же алгоритм не повинен бути засекречений.

Сучасні криптологи повністю прийняли цей принцип, називаючи все, що йому не відповідає, "безпекою через неясність". Будь-яка система, що тримає в цілях безпеки свої алгоритми в секреті, просто ігнорується співтовариством і обзивається "ханааньским бальзамом".

Висновок з принципу Кірхгофа полягає в тому, що чим менше секретів містить система, тим вище її безпека. Якщо втрата будь-якого із секретів призводить до руйнування

системи, то система з меншим числом секретів безумовно буде надійнішою. Чим більше секретів містить система, тим вона більш крихка. Менше секретів - вище міцність.

Обґрунтування принципу Кірхгофа полягає у такому: якщо для забезпечення безпеки системи криптоалгоритм повинен залишатися в таємниці, то система буде більш крихкою просто тому, що в ній виявиться більше секретів, які для забезпечення її безпеки потрібно зберігати.

6.4. Підходи та класифікація криптографічних систем

У сучасний час відомо велику кількість криптографічних методів захисту інформації. Класифікація методів шифрування може бути здійснена за наступними ознаками:

- за типом ключів (симетричні і асиметричні);
- за розміром блоку інформації (потокові і блокові);
- за характером дії над даними (метод перестановки, метод заміни, аналітичні методи, методи гамування, комбіновані методи).

У криптографічній системі перетворення шифрування може бути симетричним або асиметричним щодо перетворення розшифрування. Відповідно розрізняють два класи криптосистем:

- симетричні одноключеві криптосистеми (із секретним ключем);
- асиметричні двоключеві криптосистеми (з відкритим ключем).

Сучасні симетричні одноключеві криптоалгоритми базуються на принципах, викладених у згаданій роботі К. Шеннона. До них відносяться закордонні криптоалгоритми DES, IDEA і вітчизняний криптоалгоритм, описаний у стандарті ГОСТ 28147-89 та інші. Схеми реалізації цих криптоалгоритмів відкрито опубліковані і ретельно проаналізовані багатьма дослідниками. У цих криптосистемах секретним є тільки ключ, за допомогою якого здійснюється шифрування і дешифрування інформації. Дані криптосистеми можуть використовуватися не тільки для шифрування, але і для перевірки дійсності (аутентифікації) повідомлень.

Появі нового напрямку в криптології - асиметричної криптографії з відкритим ключем - сприяли дві проблеми, що не вдавалося вирішити в рамках класичної симетричної одноключевої криптографії. Перша з цих проблем зв’язана з поширенням секретних ключів. Наявність секретного ключа, відомого тільки одержувачеві повідомлення і його відправникові, сторіччями вважалося неодмінною умовою безпечної передачі інформації. Але при використанні симетричних криптосистем із секретними ключами вимагають рішення наступні питання: 1) як передати учасникам обміну інформацією змінювані секретні ключі, що потрібні їм для виконання цього обміну? 2) як учасники обміну зможуть переконатися в цілісності того, що вони одержали? Друга з цих проблем зв’язана з формуванням електронного цифрового підпису. Наприкінці листа або іншого авторизованого документа відправник звичайно ставить свій підпис. Подібна дія переслідує дві мети: по-перше, одержувач може переконатися в дійсності листа, звіривши підпис з наявним у нього зразком; по-друге, особистий підпис є юридичним гарантом авторства документа. Цей аспект особливо важливий при заключені різного роду торговельних угод, складанні зобов'язань, доручень і т.д.

Підробити підпис людини на папері зовсім не просто, а скопіювати ланцюжок цифр на ЕОМ - нескладна операція. Як у такому випадку гарантувати дійсність і авторство електронних повідомлень? У той же час існує багато додатків, що вимагають достовірного цифрового підпису для цифрової інформації, яка б виконувала всі ті задачі, що виконує підпис, поставлений на документі рукою.

Обидві ці проблеми здавалися нерозв’язними. Однак вони були успішно вирішені за допомогою криптографії з відкритими ключами. В опублікованій у 1976 р. статті «Нові напрямки в криптографії» У. Діффі і М. Хеллман уперше показали, що секретний зв’язок можливий без передачі секретного ключа між відправником і одержувачем. В основі цього криптографічного методу лежать так звані однобічні функції: при заданому значенні x відносно просто обчислити значення $/(x)$, однак, знаючи $y=/(x)$, визначити по в значення x

надзвичайно важко.

В асиметричних криптосистемах з відкритим ключем використовуються два ключі, принаймні, один із яких неможливо обчислити з іншого. Один ключ використовується відправником для шифрування інформації; інший - одержувачем для дешифрування одержуваних шифротекстів. Звичайно в додатках один ключ повинний бути несекретним, а інший-секретним.

А) Якщо ключ дешифрування неможливо одержати з ключа шифрування за допомогою обчислень, то таємність інформації, зашифрованої на несекретному (відкритому) ключі, буде забезпечена. Однак цей ключ повинний бути захищений від підміни або модифікації, інакше відправник може бути обдуреним і буде виконувати зашифрування на підробленому ключі, відповідно ключ дешифрування якого відомий супротивникові. Для того, щоб забезпечити секретність інформації, ключ розшифрування одержувача повинний бути секретним і фізично захищеним від підміни. Так працює канал забезпечення конфіденційності (таємності) інформації.

Б) Якщо ж, навпаки, неможливо одержати ключ шифрування з ключа дешифрування, то ключ дешифрування може бути несекретним, а секретний ключ шифрування можна використовувати для формування електронного цифрового підпису під повідомленням. У цьому випадку, якщо результат розшифрування цифрового підпису містить аутентифікаційну інформацію (заздалегідь погоджену законним відправником інформації з потенційним одержувачем), цей підпис засвідчує цілісність повідомлення, отриманого від відправника. Так працює канал аутентифікації повідомлення.

Поява нових інформаційних технологій і інтенсивний розвиток комп'ютерних мереж залучають усе більшу увагу користувачів до глобальної мережі Internet. Багато компаній і організації підключають сьогодні свої локальні мережі до мережі Internet, щоб скористатися її ресурсами і перевагами. Бізнесмени і державні організації використовують Internet у різних цілях, включаючи обмін електронною поштою, поширення інформації серед зацікавлених осіб і т.д. Підключення до Internet дає великі переваги, однак при цьому виникають серйозні проблеми з забезпеченням інформаційної безпеки підключеної локальної або корпоративної мережі. У силу відкритості своєї ідеології Internet надає зловмисникам багато можливостей для вторгнення у внутрішні мережі підприємств і організацій з метою розкрадання, перекручування або руйнування важливої і конфіденційної інформації. Розв’язання задач по захисту внутрішніх мереж від найбільш ймовірних атак через Internet може бути покладене на міжмережні екрани, іноді називані брандмауерами або firewall. Застосовуються і програмні методи захисту, до яких відносяться захищені криптопротоколи SSL і SKIP.

Важливим додатком, що потребує ефективні засоби захисту інформації, є електронні платіжні системи. У цих системах у якості універсального платіжного засобу використовуються банківські пластикові карти. Для забезпечення надійної роботи електронна платіжна система повинна бути надійно захищена. З погляду інформаційної безпеки в системах електронних платежів існує ряд потенційно уразливих місць, зокрема, пересилання платіжних і інших повідомлень між банками, між банком і банкоматом, між банком і клієнтами. Для забезпечення захисту інформації на окремих вузлах системи електронних платежів повинні бути реалізовані наступні механізми захисту: керування доступом на віконних системах, забезпечення цілісності і конфіденційності повідомлень, взаємна аутентифікація абонентів, гарантії доставки повідомлення і т. д. Якість вирішення зазначених проблем істотно залежить від раціональності вибору криптографічних засобів при реалізації механізмів захисту.

6.5. Принципи криптографічного захисту інформації

Криптографія являє собою сукупність методів перетворення даних, спрямованих на те, щоб зробити ці дані марними для супротивника. Такі перетворення дозволяють вирішити дві головні проблеми захисту даних: проблему конфіденційності (шляхом позбавлення супротивника можливості витягти інформацію з каналу зв’язку) і проблему цілісності (шляхом позбавлення супротивника можливості змінити повідомлення так, щоб змінився його зміст, або ввести помилкову інформацію в канал зв’язку).

Крім забезпечення конфіденційності інформації, криптографію часто використовують для вирішення інших завдань:

- перевірки справжності інформації - отримувач повідомлення повинен мати можливість встановити джерело інформації, а супротивник - неспроможний замаскуватися під когось іншого;
- цілісності інформації - отримувач повідомлення повинен мати можливість перевірити, чи не було це повідомлення змінено у процесі кореспонденції, а супротивник - неспроможний видати фальшиве повідомлення за справжнє;
- невідмови від авторства - відправник повідомлення у подальшому не повинен мати можливості неправдиво відмовитися від надісланого повідомлення.

Криптосистема - набір інструкцій, апаратні засоби, комплекс програм комп'ютера, що дозволяють зашифрувати відкритий текст і розшифрувати шифротекст різними способами, один із яких вибирається за допомогою конкретного ключа.

До складу криптосистеми чи шифру входять нижченаведені об'єкти:

1) Абетка А, в якій записані відкриті тексти - упорядкований набір слів із літер абетки. Будь-яке повідомлення є словом в абетці A, а множина слів A^* - простором відкритих текстів.

2) Абетка B, в якій записані криптограми. Будь-яка криптограма є словом в абетці B. Множина B^* у цьому разі називається простором криптограм. Для деяких криптосистем $A = B$.

3) Криптоалгоритм - алгоритм криптографічного перетворення, тобто математична функція, за допомогою якої шифрується та дешифрується інформація.

4) Ключ - змінний елемент шифру, застосованого для шифрування окремого повідомлення або дешифрування криптограми. Множина всіх можливих ключів називається простором ключів даного шифру. Аби задати ключ (вибрати ключ із простору ключів), слід задати конкретні таємні значення (стан) параметрів криптоалгоритму, що забезпечує вибір серед багатьох можливих варіантів одного варіанта перетворення відкритого тексту на шифротекст чи навпаки. Усі сучасні шифри базуються на принципі Керкхоффса, згідно з яким секретність шифру забезпечується секретністю ключа, а не секретністю алгоритму шифрування.

Відзначимо деякі основні характеристики шифрів:

- стійкість шифру - здатність шифру протистояти всіляким несанкціонованим атакам на нього. Це поняття - центральне для криптографії. Якщо наявність криптограми у супротивника не зменшує невизначеності можливого вибору відкритого тексту, що відповідає криптограмі, то такий шифр називають теоретично стійким;
- ефективність шифру, яка визначається швидкістю шифрувальних і дешифрувальних відображень;
- довжина ключа, точніше, об'єм ключового простору;
- простота виконання операцій шифрування та дешифрування;
- збільшення кількості помилок (для деяких шифрів помилка в одній літері при шифруванні викликає значну кількість помилок у дешифрованому тексті). При виборі шифру для зв'язку намагаються мінімізувати цю властивість шифру;
- завадостійкість шифру - здатність шифру при дешифруванні криптограми протидіяти утворенню помилок, що виникли під час шифрування через перешкоди на лініях зв'язку;
- імітостійкість шифру - здатність шифру протидіяти спробам супротивника нав'язати абонентові неправдиву інформацію шляхом спотворення криптограми на каналах зв'язку.

Про подання інформації у цифровій формі. Традиційно при використанні сучасних засобів зв'язку для передачі повідомлення або комп'ютерного шифрування інформація подається у цифровій формі, коли кожен символ тексту замінюється його номером у абетці. Наприклад, слово «аналіз» згідно з табл. 6.5 для української абетки (нумерація починається з нуля) буде подано як 00 17 00 15 11 09.

Таблиця 6.5 Заміна букв на цифри

00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16
а	б	в	г	ґ	д	е	є	ж	з	и	і	ї	й	к	л	м
17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	
н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ю	я	

Часто номери букв записують не в десятковій системі числення, а в двійковій. Слову «аналіз» у цьому разі відповідатиме двійкове слово

00000 010001 000000 001111 001011 001001,

де кожен блок з шести цифр є номером букви у двійковому записі. Отже, будь-який текст можна подати у двійковій формі за допомогою лише двох символів - 0 і 1, які називають бітами. Також використовують таблицю ASCII кодів для перетворення символів тексту в біти.

У зарубіжній криптографічній літературі літерою *M* (від слова message - повідомлення) прийнято позначати повідомлення відкритого тексту, що підлягає шифруванню, *C* - криптограму (від слова ciphertext), *E* (encrypt) і *D* (decrypt) - математичні функції, за допомогою яких виконуються процеси шифрування та дешифрування відповідно.

Відправник генерує відкритий текст вихідного повідомлення М, що повинне бути передано законному одержувачеві по незахищеному каналі. За каналом стежить перехоплювач з метою перехопити і розкрити передане повідомлення. Для того щоб перехоплювач не зміг довідатися зміст повідомлення *M*, відправник шифрує його за допомогою оборотного перетворення *Ек* і одержує шифротекст (або криптограму) *C=Ек(M)*, що відправляє одержувачеві.

Законний одержувач, прийнявши шифротекст *C*, розшифровує його за допомогою оберненого перетворення *D = E*і одержує вихідне повідомлення у виді відкритого тексту *M*: $D_K\ (C) = E''^{1}(\ E'\ (M)) = M$. Узагальнена схема криптографічної системи, що забезпечує шифрування переданої інформації, показана на рис. 6.3.

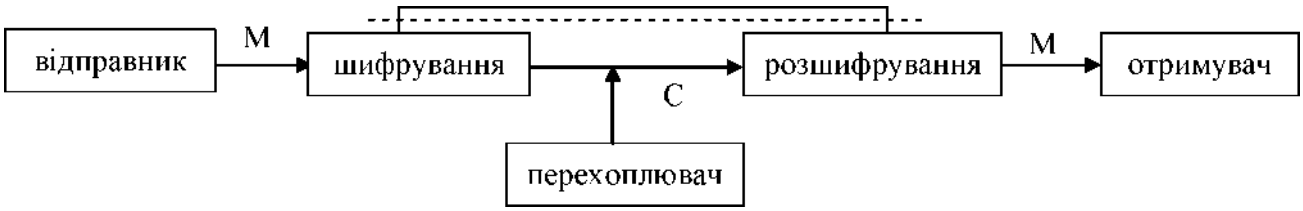


Рис. 6.3 Узагальнена схема криптографічної системи

Перетворення $E_к$ вибирається із сімейства криптографічних перетворень, що називають криптоалгоритмами. Параметр, за допомогою якого обирається використовуване перетворення, називається криптографічним ключем *К*.

Криптографічна систем це однопараметричне сімейство (*Ек*), де *к* є *К* зворотних перетворень $E_к : M \wedge C$ з простору *м* повідомлень відкритого тексту в простір *С* шифрованих текстів. Параметр *К* вибирається з скінченої множини *К,* що називається простором ключів.

Схема симетричної криптосистеми з одним секретним ключем була показана на рис. 7.1. У ній використовуються однакові секретні ключі в блоці шифрування і блоці розшифрування.

Узагальнена схема асиметричної криптосистеми з двома різними ключами *К1* і *К2* показана на рис. 6.4. У цієї криптосистеми один із ключів є відкритим, а інший - секретним.

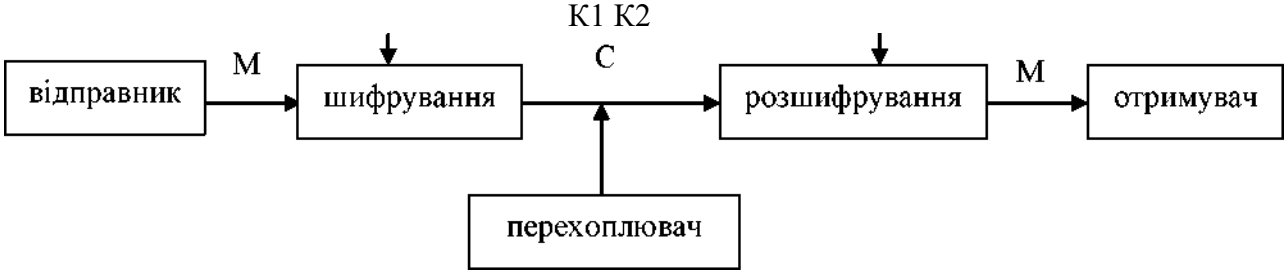


Рис. 6.4 - Узагальнена схема асиметричної криптосистеми

У симетричної криптосистеми секретний ключ треба передавати відправникові й одержувачеві по захищеному каналі поширення ключів, наприклад такому, як кур'єрська служба. На рис.7.1 цей канал показаний «екранованою» лінією. Існують і інші способи розподілу секретних ключів, вони будуть розглянуті пізніше. В асиметричної криптосистемі передають по незахищеному каналі тільки відкритий ключ, а секретний ключ зберігають на місці його генерації.

На рисунку 6.5 показаний потік інформації в криптосистемі у випадку активних дій перехоплювача. Активний перехоплювач не тільки зчитує всі шифртексти, передані по каналі, але може також намагатися змінювати них за своїм розсудом.

Будь-яка спроба з боку перехоплювача розшифрувати шифротекст *С* для одержання відкритого тексту *М* або зашифрувати свій власний текст *М'* для одержання правдоподібного шифротекста *С*, не маючи справжнього ключа, називається криптоаналітичною атакою.

К

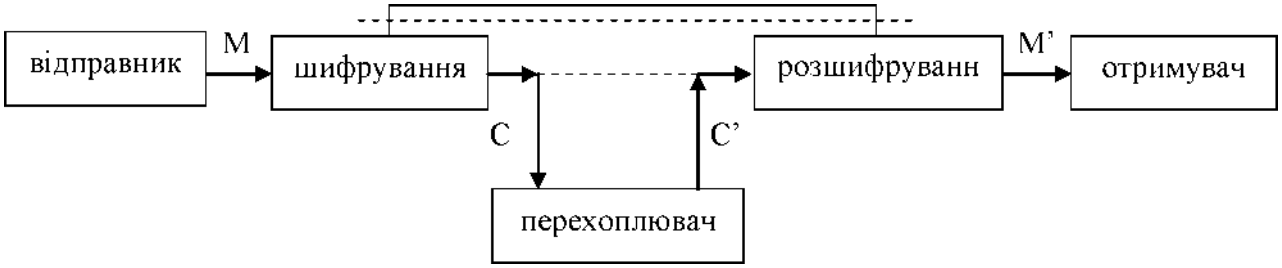


Рис 6.5 - Потік інформації в криптосистемі у випадку активних дій перехоплювача

6.6. Деякі відомості з теорії складності задач

Теорія складності - це методика аналізу обчислювальної складності різних математичних (зокрема криптографічних) методів та алгоритмів. У криптографії цю теорію використовують як основу для визначення надійності криптографічних алгоритмів, що дає змогу проаналізувати, чи можна зламати той чи інший шифр за прийнятний час. Очевидно, складність обчислювального алгоритму залежить від обчислювальних потужностей, необхідних на його виконання. Найчастіше складність алгоритму характеризують двома параметрами: *Т* - часова складність та *5* - просторова складність або вимоги до пам'яті комп'ютера.

З'ясуємо поняття часової складності. Нехай *А* - алгоритм розв'язку деякого класу задач, *п* - розмірність окремої задачі цього класу (масив або довжина вхідної послідовності, іншими словами, величина, що характеризує розмір вхідних даних). Функцію $/_A(n)$ назвемо робочою, якщо вона визначає верхню межу максимальної кількості операцій (додавання, множення, порівняння і т. д.), які має виконати алгоритм *А* під час розв'язку задачі розмірності *п*. Тоді часова складність залежить від того члена розкладання робочої функції, який із зростанням *п* зростатиме швидше інших (члени розкладання, порядок яких нижче, ігноруються). У багатьох криптографічних та математичних задачах порядок величини обчислювальної складності алгоритмів визначається стандартно за допомогою символу «*о*». Отже, для визначення часової складності алгоритму не обов'язково знати точний термін виконання різних операцій, кількість бітів, потрібних для зображення змінних, навіть швидкість процесора. З цієї оцінки наочно видно, як розмірність вхідних даних впливає на вимоги до часу виконання та об'єму пам'яті. Так, якщо часова складність одного алгоритму $T = o(n)$, то подвоєння розмірності вхідних даних відповідно подвоює і час виконання алгоритму, а якщо часова складність деякого іншого алгоритму дорівнює $T = o(2^n)$, то подвоєння часу виконання викличе навіть додавання лише одного біта до вхідних даних.

Наведемо відому класифікацію алгоритмів за їх часовою складністю:

- алгоритм називається сталим, якщо його складність не залежить від *п*;
- алгоритм називається поліноміальним за часом, якщо його часова складність дорівнює $o(n^m)$, де *т* - константа, тобто робоча функція $/_A(n)$ поліноміального алгоритму є поліномом *Pm(n)* степеня *т.* Зокрема, алгоритм лінійний, якщо його складність *T=o(n),*квадратичний - у випадку *T=o(n2* і т.д.;

- алгоритми, складність яких дорівнює $o(t^{\partial(n)})$, де $\Gamma>1$ -константа, $\partial(n)$ - деяка поліноміальна функція від n, називаються експоненціальними;
- експоненціальні алгоритми, складність яких є $o(C^{((iП)})$, де із зростанням n функція $\partial(n)$ зростає швидше константи c, називаються супер- поліноміальними.

Якщо сучасні комп'ютери здатні сприймати поліноміальні алгоритми для задач великої розмірності, то експоненціальні алгоритми часто стають каменем спотикання. Тому в ідеальному випадку кожен криптограф хотів би стверджувати, що його шифр можна зламати тільки за допомогою алгоритму, який характеризується експоненціальною часовою складністю. Із збільшенням n часова складність алгоритмів може стати настільки великою, що почне заважати практичній реалізації алгоритму. Для ілюстрації цього у табл. 6.6 наведено час виконання алгоритмів різних класів при розмірі вхідних даних $n=10^6$.

Таблиця 6.6 Час виконання алгоритмів

Клас алгоритмів	Складність	Кількість операцій комп'ютера	Час зі швидкістю 106 операцій за секунду
Сталі	$o(1)$	1	1 мкс
Лінійні	$o(n)$	106	1с
Квадратич	$o(n^2)$	10^{12}	11,6 доби
Кубічні	$o(n3)$	10^{18}	32000 років
Експоненціальні	$o(2^n)$	10301030	У ю^{30}!^{006} разів довше за термін існування

Зауваження 1. Часова складність силової атаки пропорційна кількості можливих ключів, яка експоненціально залежить від довжини ключа. Якщо n - довжина ключа, то складність такого зламування дорівнює o (2^n).

Зауваження 2. Усі криптографічні системи мають бути ефективними, тобто всі обчислення законним користувачем за криптографічним алгоритмом повинні виконуватися за поліноміальний час. Проблема обґрунтування стійкості криптографічної системи зводиться до доведення відсутності поліноміального алгоритму розв'язання задачі, яка виникає у супротивника при намаганні зламати алгоритм. На жаль, сучасний стан теорії складності не дозволяє доводити над поліноміальні нижні оцінки складності алгоритмів.

6.7. Основи теорії К. Шеннона

Клод Шеннон, а точніше його книга «теорія зв'язку в секретних системах», зробила визначальний внесок у сучасну криптографічну науку. Вважають, що зазначена праця визначила онови та сформулювала сутність сучасної криптографії. Автор розглядав модель, в якій джерело повідомлень утворює відкритий текст *M*. Джерело ключів генерує *K*. Шифратор перетворює *M* за допомогою *K* у текст С: $c = Ек (M)$. Дешифратор, отримавши *C*, використовує оператор: $M = E' к(C)$. Задачею криптоаналітика є отримання тексту і ключа на основі аналізу шифротексту (рис.6.6).

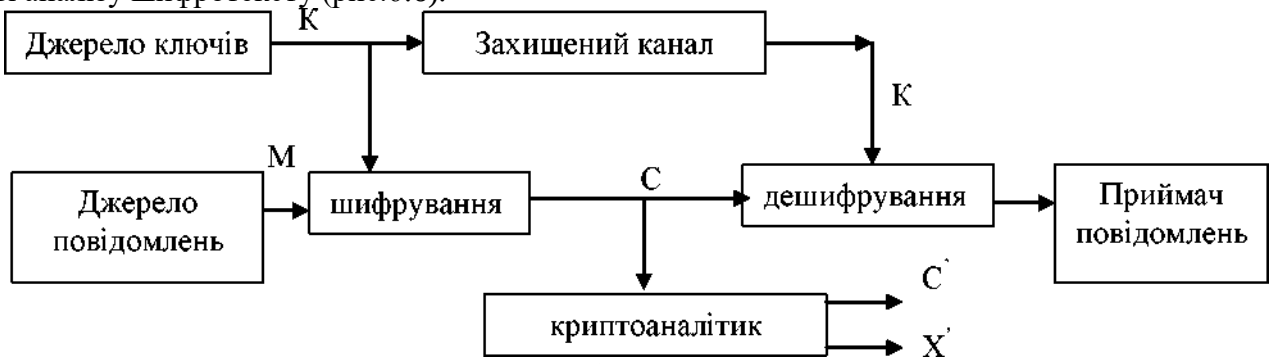


Рис 6.6 - Модель обміну повідомленнями

К. Шеннон розглянув питання теоретичної і практичної секретності.

Для визначення теоретичної секретності він сформулював такі питання:

- наскільки стійка система, якщо криптоаналітик противника необмежений часом і володіє всіма необхідними засобами для аналізу криптограм;
- чи має криптограма єдиний розв'язок; який об'єм шифротексту необхідно перехопити крипто аналітику, щоб розв'язок був єдиним.

Для відповіді на ці питання криптолог ввів поняття **цілковитої секретності** за допомогою наступної умови: для будь-яких *C* апостеріорні ймовірності рівні апріорним ймовірностям, тобто перехоплення зашифрованого повідомлення не дає крипто аналітику противника ніякої інформації:

Pc(M) = P(M)Pм(C)/P(C), де *P(M)* - апріорна ймовірність повідомлення *M*; $P_M(C)$ - умовна ймовірність криптограми *C* при умові, вибрано повідомлення *M*, тобто сума всіх тих ключів, які переводять повідомлення *M* в криптограму *C*; *P(C)* - ймовірність отримання криптограми *C*; $P_C(M)$ - апостеріорна ймовірність події *M* при умові, що перехоплена криптограма *C*. Для цілковитої секретності значення $P_C(M)$ та *P(M)* повинні бути рівні для всіх *M* і *C*.

Існують криптосистеми, для яких будь-який об'єм перехопленої інформації недостатній для знаходження шифрувального відображення, причому це не залежить від обчислювальної потужності обладнання криптоаналітика. Шифри такого типу називаються безумовно стійкими (за К. Шенноном - ідеально секретними). Криптосистеми, які використовують рівномірний випадковий ключ, який має туж саму довжину , що і відкритий текст називають шифрами із одноразовим блокнотом (на практиці такі шифри не зручні). Шифри, стійкість яких грунтується на значній обчислювальній складності розв'язку криптографічної задачі називають умовно стійкими (практично стійкими за К. Шенноном). Шифри, які грунтуються на обчислювально нездійснених задачах, називають обчислювально стійкими і отримали найбільшу практичну розповсюдженість.

6.8 Види історичних шифрів

Історичними (докомпютерними) називають шифри, що

використовувались до 1960 року, тобто в доком'ютерну епоху. Ці шифри є типовими прикладами симетричних алгоритмів шифрування і залишаються основою багатьох сучасних криптосистем.

В класичній криптографії доведено, що в основі криптографічних алгоритмів знаходяться тільки два основних типи перетворень - заміни і перестановки; всі інші є лише їх комбінацією.

В **перестановочних** шрифтах символи відкритого тексту змінюють свої розташування. Наприклад, відкритий текст виписується у вигляді матриці з пронумерованими стовпцями, після чого порядок стовпців змінюється:

Простий шифр маршрутної перестановки

Саме цей шифр реалізовувала Сцитала.

У даному вигляді шифру текст пишеться на горизонтально разграфленому аркуші паперу фіксованої ширини, а шифротекст зчитується по вертикалі. Розшифрування полягає в записі шіфротекста вертикально на аркуші розграфлений паперу фіксованої ширини і потім зчитуванні відкритого тексту горизонтально.

Для приклада в якості відкритого тексту взято КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ.

К	И	Ї	В	С
	Й	И	К	Ь
П	О	Л	І	Т
Ч	І	Н	Х	Е
Н	И	Й		І
Т	И	Т	С	Н
У	Т			

Рис. Приклад використання маршрутної перестановки

Тоді зашифрований текст буде таким: к пчнтуийоіиит тйнлиївкіх с ніетьс.

Стовпчикова транспозиція

Стовпчикова транспозиція - перестановочний шифр, в якому використовується

блоковий алгоритм шифрування:

1. Відкрите повідомлення вписується в прямокутник [пхт] заздалегідь обумовленим способом.
2. Стовпчики прямокутника нумеруються в звичайному порядку.
3. Стовпчики переміщуються в порядок, що задається вказаною ключовою послідовністю (або в порядку букв ключа - літерного ключового слова).

6	2	7	3	1	4	5
К	Л	А	С	И	Ч	Н
А		К	Р	И	П	Т
О	Г	Р	А	Ф	І	Я

Ключ 6273145 →

1	2	3	4	5	6	7
Ч	Л	Н	А	К	С	И
П		Т	К	А	Р	И
І	Г	Я	Р	О	А	Ф

Рис. Приклад використання стовпчикової перестановки

Простий перестановочний шифр

Фіксується матриця перестановки п-елементів. Для п=5 вона може бути такою:

$$\sigma = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 2 & 5 & 1 & 3 & 4 \end{pmatrix}$$

Вхідне повідомлення розбивається на блоки по п символів, в кожному з яких символи переставляються у відповідності з матрицею перестановки.Саме перестановка є секретним ключем.

Зашифруємо в якості прикладу відкритий текст ПЕРЕСТАНОВКА.

Розіб'ємо текст на частини по 5 букв: ПЕРЕС ТАНОВ КА. Потім переставимо букви в них відповідно до нашої матриці: еспре автно ка. Прибравши проміжки між групами, отримаємо шіфротекст: еспреавтнока.

Шифр зсуву

Шифр зсуву - один з перших відомих шифрів, що використовувався ще в давнину. Використовується поточний алгоритм шифрування, який полягає у такому:

1.Перенумеровуються всі літери вихідного алфавіту, починаючи з 0, наприклад, букві «a» присвоюється номер 0, «b» - 1, і т.д. до літери «z» під номером 25.

2.Переписується вихідний текст з заміною кожної букви відповідним номером.

3.До кожного числа в утвореній послідовності додається значення3 ключа К по модулю 26; К- ціле число між 0 і 26.

Криптостійкість шифру зсуву вкрай низька. Наївний шлях атаки на шифр зсуву полягає в простому переборі можливих значень ключа до тих пір, поки не вийде осмислений текст. Оскільки існує рівно 25 варіантів (ключ О не змінює тексту), то для їх перебору потрібно не дуже багато часу, особливо у випадку короткої шифрограми.

Приклад 1. Шифр зсуву з кроком 1 використовував ще імператор Август (1 в. н. е.). У своєму листуванні він замінював першу букву латинського алфавіту на другу, другу - на третю і т. д., нарешті, останню - на першу:

a b c d e f g h i j k l m n o p q r s t u v w x y z

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

Улюблений вислів імператора Августа виглядало так: GFTUJOB MFOUF ("Festina lente" - лат. "Поспішай повільно").

Приклад 2. Шифр зсуву з ключем 3 використовував ще Юлій Цезар; тому його також називають **шифром Цезаря**. У 1 в. н.е. Юлій Цезар під час війни з галлами, листуючись зі своїми друзями в Римі, заміняв у повідомленні першу літеру латинського алфавіту (A) на четверту (D), другу (B) - на п'яту (E), нарешті, останньому - на третю:

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

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

Донесення Ю. Цезаря Сенату про здобутуї їм перемогу над Понтійським царем виглядало так: YHQL YLGL YLFL ("Veni, vidi, vici" - лат. "Прийшов, побачив, переміг").

Приклад 3. Прикладом шифру простої заміни може служити програма ROT13, яку зазвичай можна знайти в операційній системі UNIX. З її допомогою буква "A" відкритого тексту англійською мовою замінюється на літеру "N", "B" - на "O" і так далі. Таким чином, ROT13 циклічно зсуває кожну букву англійського алфавіту на 13 позицій вправо. Щоб отримати вихідний відкритий текст треба застосувати функцію шифрування ROT 13 двічі: P

= ROT13 (ROT13 (P)).

В шифрах **заміни** один символ відкритого тексту замінюється символом зашифрованого тексту.

Шифр заміни

Основний недолік шифру зсуву полягає в тому, що існує надто мало можливих ключів - всього 25. З метою усунення цього недоліку був винайдений шифр заміни.

Шифр заміни використовує блочний алгоритм шифрування, який полягає у такому:

1. Описуєтьяс ключ такого шифру, для чого спочатку виписується алфавіт, а беспосередньо під ним - той же алфавіт, але з переставленими літерами тогож або іншого алфавіту.
2. Шифрування полягає в заміні кожної букви у відкритому тексті на відповідну їй нижню літеру.
3. Щоб розшифрувати шифротекст, потрібно кожну його букву знайти в нижньому рядку таблиці і замінити її відповідною верхньою.

Кількість всіх можливих ключів шифру заміни збігається з числом всіх перестановок 26 елементів, тобто рівним 26! $\sim 10^9$. Тому, перебираючи всі можливі ключі за допомогою комп'ютера, ми витратимо стільки часу, що задача про дешифрування конкретного повідомлення перестане бути актуальною. Тим не менш, шифр заміни можна зламати, спираючись на частотному аналізі символів мови.

Шифр заміни є блоковим, блок в якому складається з однієї букви. Блок шифротексту отримується з блоку відкритого тексту в результаті застосування ключа (таблиці відповідності).

Приклад 4. Криптограма слова «hello» буде виглядати як ESVVJ, якщо користуватися наведеною

відповідністю:

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
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

Поліалфавітний шифри заміни

Шифр заміни відноситься до так званих **моноалфавітних** шифрів заміни, в яких використовується тільки один впорядкований набір символів, підміняючий собою стандартний алфавіт.

Основний недолік шифру заміни (і зсуву) полягає в тому, що кожна буква відкритого тексту при шифруванні замінюється раз і назавжди фіксованим символом. Тому при дешифруванні ефективно працює частотний аналіз символів мови. З початку XIX століття розробники шифрів намагалися подолати такий зв'язок між відкритим текстом і його шифрованим варіантом.

Один із шляхів вирішення вказаної проблеми полягає в тому, щоб брати кілька наборів символів замість стандартного алфавіту і шифрувати букви відкритого тексту, вибираючи відповідні знаки з різних наборів у визначеній послідовності. Шифри такого типу носять назву **поліалфавітних** шифрів заміни.

Алгоритм поліалфавітного шифру заміни може полягати у такому:

1. Описується відповідність між алфавітом відкритого тексту і кількома рівнями алфавітів шифротексту:

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

2. При шифруванні букви відкритого тексту, що знаходяться на непарних позиціях, замінюються відповідними буквами другого рядка, а ті, що знаходяться на парних позиціях - буквами третього рядка.
3. При розшифруванні букви криптотексту, що знаходяться на непарних позиціях, шукаються в другому рядку, а ті, що знаходяться на парних позиціях - в третьому рядку.

Приклад 5. Вхідне слово hello в шифротексті буде виглядати як SHLJV, що суттєво

ускладнює застосування для атаки частотного аналізу мови.

На практиці можна використовувативати не два, а аж до п'яти різних алфавітів шифротексту, багаторазово збільшуючи простір ключів. Дійсно, легдо підрахувати, що якщо ми беремо символи з п'яти заміщуючих наборів, то число можливих ключів $(26!)^{5}\sim 2^{441}$. Проте в цьому випадку ключ - послідовність з 26*5 = 130 літер. Для середнього користувача початку XIX століття такий ключ був занадто великим, щоб запам'ятати його.

Незважаючи на зазначений недолік, найвідоміші шифри XIX століття грунтувалися саме на описаному принципі. До них відносяться модифікований шифр Цезаря та шифр Віженера.

Модифікований шифр Цезаря

Модифікований шифр Цезаря - один з варіантів поліалфавітного шифру зсуву.

Використовується потоковий алгоритм шифрування, який полягає у такому:

1. Як і у випадку шифру зсуву, перенумеруємо 26 літери вхідного алфавіту, починаючи з 0:

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
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

2. Створюємо секретний ключ - будь-яке слово, яке наивають **гаслом**.
3. Гасло підписується ють під повідомленням з повторенням.
4. Щоб отримати шифрований текст, складають номер чергової букви з номером відповідної букви ключа. Якщо отримана сума більше 26, то з неї віднімають 26. В результаті отримують послідовновательность чисел від 1 до 31. Знову замінюючи числа цієї последовательности відповідними буквами, отримують шифрований текст.

Приклад 6. Якщо гаслом є слово sesame, то шифрування виглядає так:

t	h	i	s	i	s	a	t	e	s	t	m	e	s	s	a	g	e
19	7	8	18	8	18	0	19	4	18	19	12	4	18	18	0	6	4
s	e	s	a	m	e	s	e	s	a	m	e	s	e	s	a	m	e
18	4	18	0	12	18	18	4	18	0	12	18	18	4	18	0	12	18
L	L	A	S	U	W	S	X	W	S	F	Q	W	W	K	A	S	I
11	11	0	18	20	22	18	23	22	18	5	16	22	22	10	0	18	8

Шифр Віженера

Шифр Віженера є поліалфавітним блоковим шифром. Шифрування і розщифрування грунтується на використанні **таблиці Віженера**. В загальному випадку першим її рядком є алфавіт відкритого тексту, а першим стовпчиком - алфавіт ключа. Тіло таблиці складається з циклічно зсунутих алфавітів, причому його перший рядок може бути довільним змішаним алфавітом.

В найпростішому випадку використання незмішаного алфавіту для самоключа таблиця Віженера матиме такий вигляд (складена з 31 літери російського алфавіту - без літер **е** і **ь**) :

	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	е	ю	я
а	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	е	ю	я
б	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	е	ю	я	а
в	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	е	ю	я	а	б
г	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	е	ю	я	а	б	в
д	д	е	ж	з	и	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ы	е	ю	я	а	б	в	г
е	е	ж	з	и	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ы	е	ю	я	а	б	в	г	д
ж	ж	з	и	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ы	е	ю	я	а	б	в	г	д	е
з	з	и	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ы	е	ю	я	а	б	в	г	д	е	ж
...																															

Щоб зашифрувати повідомлення:

1. Вибирається слово - гасло і підписується з повторенням над буквами відкритого повідомлення.

2. Щоб отримати шифрований текст, знаходять черговий знак лозунгу, починаючи з першого у вертикальному алфавіті, а відповідний йому

знак повідомлення - в горизонтальному.

Приклад 7. Вибираємо гасло - математика. Знаходимо стовпець, що відповідає букві "м" гасла, а потім рядок, відповідну букві "к". На перетені виділених стовпця і рядка знаходимо букву "ц".

м	а	т	е	м	а	т	и	к	а	м	а	т	е	м	а	т	и	к	а	м	а	т	е	м	а
к	р	и	п	т	о	г	р	а	ф	и	я	с	е	р	ь	е	з	н	а	я	н	а	у	к	а

Продовжуючи далі, знаходимо весь зашифрований текст:

Ц Р Ь Ф Я О Х Ш К Ф Ф Я Д К Э Ь Ч П Ч А Л Н Т Ш Ц А.

У 1888 р. француцз маркіз де Віарі довів, що щифрування по Віженеру відтворює алгебраїчна формула

$$C=(p+G) \bmod 26 ,$$

де С - будь-яка буква шифротексту, р- будь-яка буква відкритого тексту, G - будь-яка буква гами, а заміна букв числами здійснюється за таблицею:

abcdefghij k I m п o p q r s t u v w x ***y*** z

а 1 2 3 4 S Б 7 S 9 ID 11 12 13 14 15 1Б 17 IB 13 2D 21 22 Z3 24 25

Процес розшифрування відтворює формула

$$p = (C - G +26) \bmod 26 ,$$

Використання рівняння шифрування дозволило відмовитись від громіздкої таблиці Віженера. Згодом лозунгова гама стала довільною числовою послідовністю, а щифр з вказаним рівнянням шифрування став називатись **шрифтом гаміювання**.

Як бачимо, у простому варіанті з коротким ключовим словом і з таблицею, що складається зі звичайних алфавітів, шифр Віженера є еквівалентним модифікованому шифру Цезаря. В літературі часто саме цей найпростіший варіант і називають шифром Віженера.

7 Симетричні системи шифрування. Блочні шифри.

7.1. Блокові алгоритми і режими шифрування

7.1.1 Загальні властивості блокових шифрів

Як зазначалось раніше, головний недолік абсолютно криптостійкого шифру одноразових блокнотів - велика довжина ключа, яка має бути не меншою від довжини повідомлення. Через це використовувати його на практиці складно.

Для побудови стійкого шифру, зручного для практичного використання,

можна запропонувати такі підходи:

1. **Блоковість**. Вибираємо довжину ключа меншою за довжину повідомлення, розбиваємо повідомлення на окремі блоки і шифруємо кожний з них (крім, можливо, останнього) шляхом сумування з ключем по модулю два.

2. **Режими шифрування**. Для збільшення стійкості блокового

шифрування можна забезпечити використання при шифруванні кожного наступного блоку результатів шифрування попереднього блоку (наприклад, в якості нового ключа шифрування для блоку). Тоді зловмисник не зможе розшифрувати блок криптотексту, доки не розшифрує всі попередні блоки.

3. **Багатораундовість**. Додатково збільшити стійкість блокового шифрування можна з рахунок багаторазового виконання шифрування кожного блоку за умови, що функція шифрування є нелінійним перетворенням.

В блокових алгоритмах вхідна послідовність розбивається на **блоки** - ділянки певної довжини (найчастіше, по 64 біти). Якщо довжина відкритого тексту виявляється некратною довжині блоку, застосовується операція **доповнення** (padding) останнього блоку до необхідної довжини, яка полягає у дописуванні необхідної кількості нулів або випадкового набору символів (Рис.7.1).

Рис.7.1. Представлення даних в блоковому алгоритмі

Криптографічне перетворення в блокових алгоритмах шифрування здійснюється над кожним блоком окремо (Рис.7.2). Його сутність полягає у застосуванні до блока багаторазово математичного перетворення. Внаслідок цього результуюче перетворення виявляється криптографічно більш сильним, ніж перетворення над окремо взятим блоком. Метою таких перетворень є створення залежності кожного біту блоку шифротексту від кожного біту ключа і кожного біту відкритого тексту:

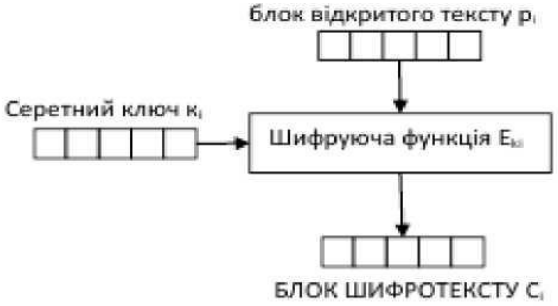


Рис.7.2. Схема шифрування блоку даних в блоковому алгоритмі

Зашифрований блок може використовуватися для шифрування наступного, в результаті чого кожний блок отримає контекст, що властивий всьому повідомленню (Рис.7.3). Такі механізми шифрування використовуються для уникнення від деяких атак, заснованих на стиранні або вставки блоків, і визначаються відповідним *режимом шифрування*.

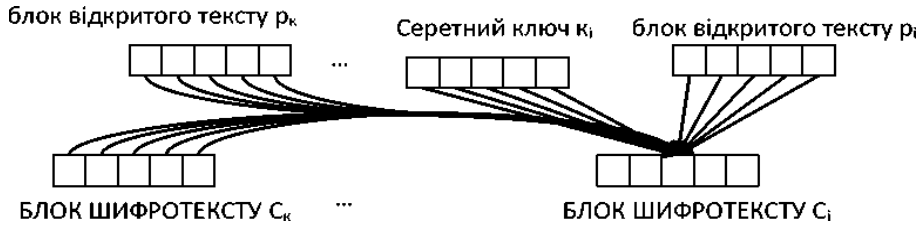


Рис.7.3. Механізм режимів шифрування

Режим шифрування - метод застосування блокового шифру, в якому для забезпечення вищого рівня криптостійкісті шифрування блоків вхідного повідомлення здійснюється з використання блоків криптотексту.

Розрізняють п’ять основних режимів шифрування:

1. **ECB** (Electronic Codebook Mode) - режим електронної кодової книги.
2. **CBC** (Cipher Block Chaining Mode) - режим зціплення блоків по криптотексу.
3. **CFB** (Cipher Feedback Mode) - режим з оберненим зв’язком по криптотексу.
4. **OFB** (Output-Feedback Mode) - режим з оберненим зв’язком по виходу.
5. **CTR** (Coutner) - режим з лічильником.

Блокові алгоритми шифрування сьогодні є основним засобом криптографічного захисту інформації. Основні переваги блокових алгоритмів шифрування:

- Висока швидкість шифрування/розшифрування.
- Висока гарантована стійкість, яка до того ж може бути доведена математично.
- Можливість ефективної програмної реалізації.

7.1.2 Режим електронної кодової книги (ECB)

В цьому режимі блоки відкритого тексту шифрують ся незалежно від інших за допомогою одного й того ж ключа.

Шифрування описується рівнянням: $C_i=E_{кі}(p_i)$ для i=КМ, де C_i та p_i - блоки відповідно зашифрованого і відкритого тексту, $E_{кі}$ - функція шифрування.

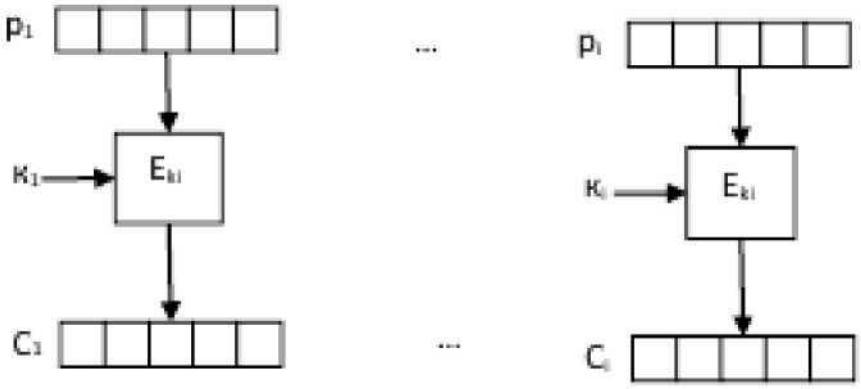


Рис.7.4. Схема шифрування в режимі ECB

Розшифрування здійснюється за допомогою функції розшифрування Pi=Dki(Ci):

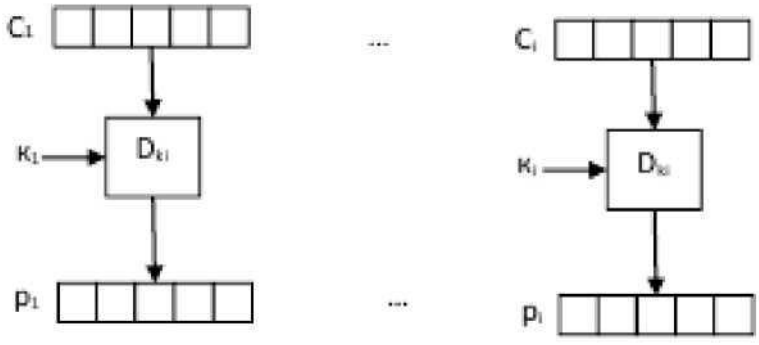


Рис.7.5. Схема розшифрування в режимі ECB

Такий режим є криптографічно слабким. Основні недоліки:

- Режим не є критичним по відношенню до вставки і втрати блоків в процесі передачі.
- Однакові блоки будуть перетворюватись в такі ж самі, що може дати ключ для аналізу вмісту повідомлення.

Перевагою режиму є те, що шифрування (розшифрування) блоків може виконуватись паралельно.

7.1.3 Режим зціплення блоків по криптотексу (CBC)

В цьому режимі шифрування кожний блок відкритого тексту,крім першого, складається по модулю 2 (операція XOR) з попереднім зашифрованим блоком. Шифрування першого блоку здійснюється за допомогою початкового **вектора ініціалізації** (**синхропосилки**), яка передається по відкритому каналу передачі даних.

Шифрування описується рівнянням: Ci= Eki(pi XOR Ci-1) для і 2ⁱN. Co - синхропосилка (Рис.7.6).

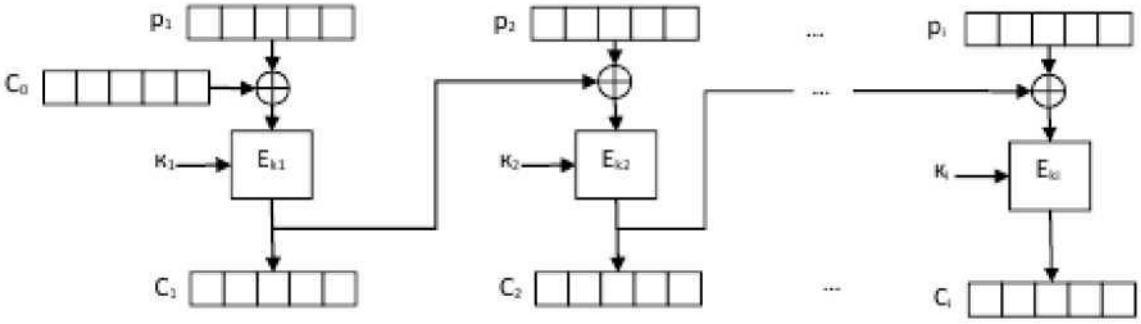


Рис.7.6. Схема шифрування в режимі CBC

Розшифрування описується рівнянням: pi= Q-1XOR Dki(Ci) для і 2ⁱN. Co - синхропосилка (Рис.7.7).

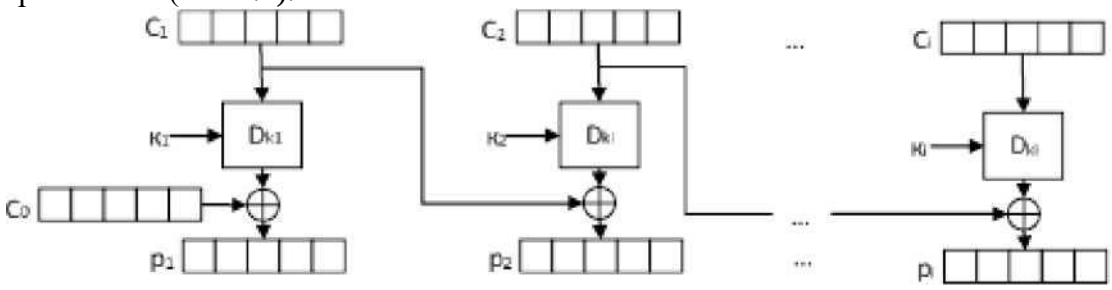


Рис.7.7 Схема розшифрування в режимі CBC

Важливо відмітити, що останній блок шифротексту залежить від ключів, синхропосилки і всіх бітів відкритого тексту. Тому його можна розглядати в якості ідентифікатора повідомлення. Цей ідентифікатор широко використовується для автентифікації повідомлення та відправника і називається **кодом автентифікації повідомлення** або **MAC** (Message Authentication Code).

Коди автентифікації повідомлень не забезпечують секретність, але гарантують автентифікацію і цілісність. Вони дають впевненість, що повідомлення прийшло саме від тієї людини, який позначений як автор, і що повідомлення по дорозі не змінилося. Хто завгодно може прочитати повідомлення. Але той, хто знає ключ MAC, може впевнитися, що воно не було змінено.

Для використання MAC Аліса спочатку домовляється з Бобом про ключ. Потім, коли вона хоче послати Бобу повідомлення, вона обчислює MAC повідомлення і додає його до повідомлення. Коли Боб отримує повідомлення, він обчислює його MAC і порівнює його з тим значенням MAC, яке прислала Аліса. Якщо вони збігаються, то він може бути впевнений у двох речах: повідомлення дійсно прийшло від Аліси і це повідомлення незмінене. Банки використовують таку просту систему автентифікації вже кілька десятиліть.

MAC постійно використовуються в Інтернет, наприклад, у протоколі IPsec, щоб гарантувати, що IPпакети не були змінені в проміжку між відправленням і прибуттям на місце призначення. Їх використовують у всіх можливих протоколах міжбанківських переказів для встановлення автентичності повідомлень.

Переваги режиму CBC:

- Режим CBC є критичним по відношенню до вставки і втрати блоків в процесі передачі.
- Забезпечується автентифікація повідомлення та відправника на основі MAC.

Недоліком режиму є те, що шифрування (розшифрування) не піддається розпаралеленню.

7.1.4 Режим з оберненим зв'язком по криптотексу (CFB)

В цьому режимі алгоритму блокового шифрування черговий блок відкритого тексту шумується по модулю 2 з блоком попереднього шифротексту, але тільки після того, як він буде додатково оброблений за допомогою функції шифрування.

Важливо відмітити, що в цьому режимі як для шифрування, так і розшифрування використовується тільки функція шифрування, а обернена до неї функція розшифрування взагалі не потрібна.

Шифрування описується рівняннями: $C_i = E_{ki}(C_{i-1})$ XOR pi для і 2˙N. C_0 - синхропосилка (Рис.7.8).¶

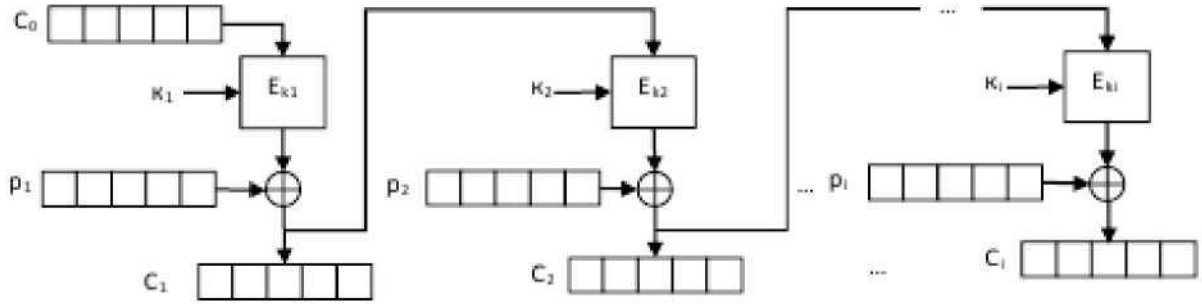


Рис.7.8. Схема шифрування в режимі CFB

Розшифрування описується рівнянням: pi = Екі(Ci-i) XOR Cі для і 2˙M Co - синхропосилка (Рис.7.8).

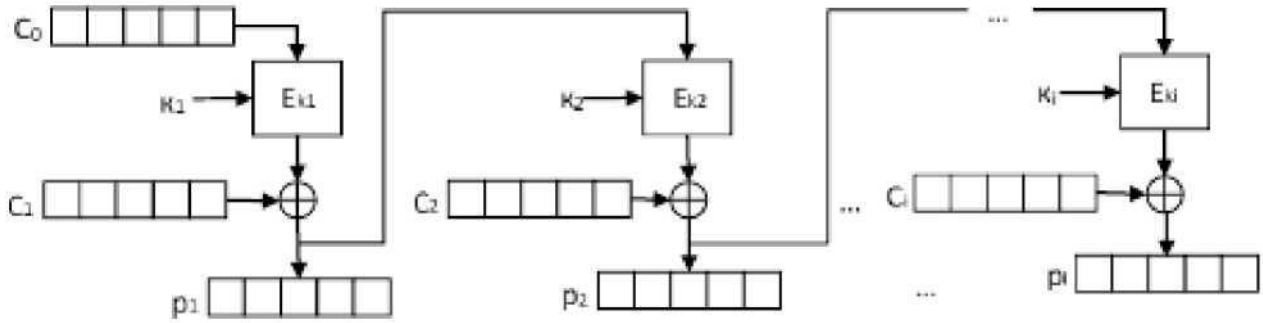


Рис.7.9. Схема розшифрування в режимі CFB

Переваги режиму CPB:

- Критичність по відношенню до вставки і втрати блоків в процесі передачі.
- Використовується тільки функція шифрування.
- Симетрія операцій шифрування/розшифрування

Недоліком режиму є те, що шифрування (розшифрування) не піддається розпаралеленню.

7.1.5 Режим з оберненим зв'язком по виходу (OFB)

В цьому режимі функція шифрування викликається до сумування по модулю 2 з блоком відкритого тексту, але на її вхід подається не шифротекст, а ті ж вхідні дані. Можна сказати, що при використанні даного режиму блочний шифр перетворюється в синхронний шифропотік, тобто генеруються ключові блоки, які є результатом сумування по модулю 2 з блоками відкритого тексту.

Операції шифрування і розшифрування виявляються симетричними і описуються рівняннями $C_i = p_i$ XOR O_i для i=1<M (шифрування, Рис.8) і $p_i = C_i$ XOR O_i (розшифрування, Рис.9) для i=1^H де O1= $E_к(C_0)$, $O_i = E_{кі}(O_{i-1})$ для i=1-К

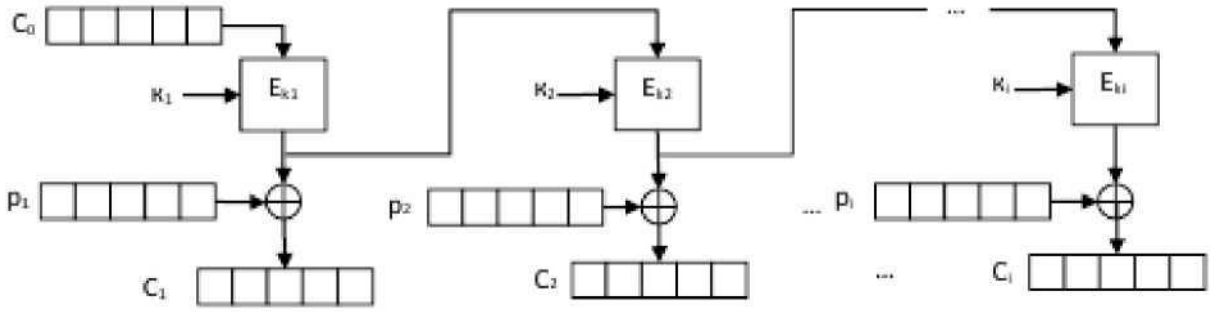


Рис.7.10. Схема шифрування в режимі OFB

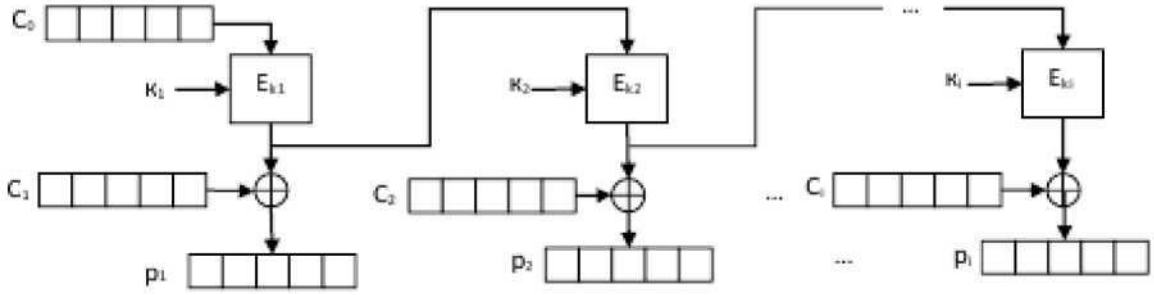


Рис.7.11. Схема розшифрування в режимі OFB

Переваги режиму CFB:

- Критичність по відношенню до вставки і втрати блоків в процесі передачі.
- Використовується тільки одна функція шифрування, а обернена до неї функція розшифрування взагалі не потрібна.
- Операції шифрування і розшифрування виявляються симетричними.

Паралельне виконання операцій шифрування (розшифрування) блоків в такому режимі неможливе.

7.1.6 Режим з лічильником (CTP)

В даному режимі на кожній і-ітерації на вхід шифрування подається деяке випадкове значення T_i. Всі T_i мають бути різними, тому генератор таких значень називається лічильником.

Операції шифрування і розшифрування теж виявляються симетричними і описуються такими рівняннями $C_i = p_i$ XOR O_i для i=1<M (шифрування, Рис.10) і $p_i = C_i$ XOR O_i (розшифрування, Рис.11) для i=НН де $O_i = E_{кі}(T_i)$ для i=1<№

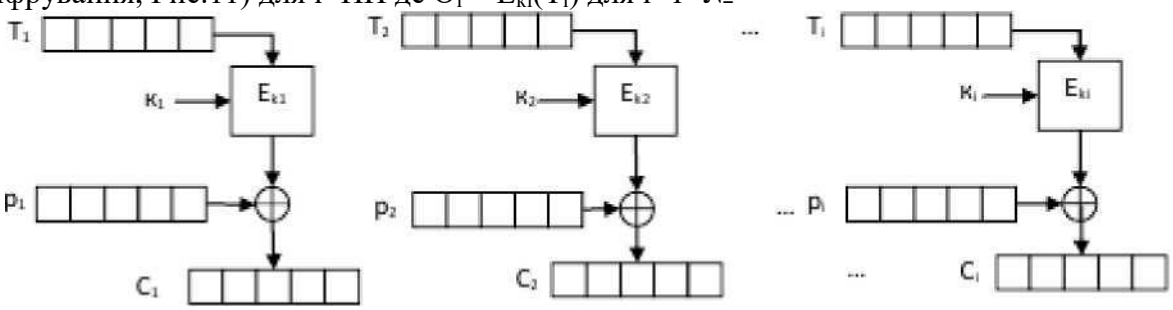


Рис.7.12. Схема шифрування в режимі CTR

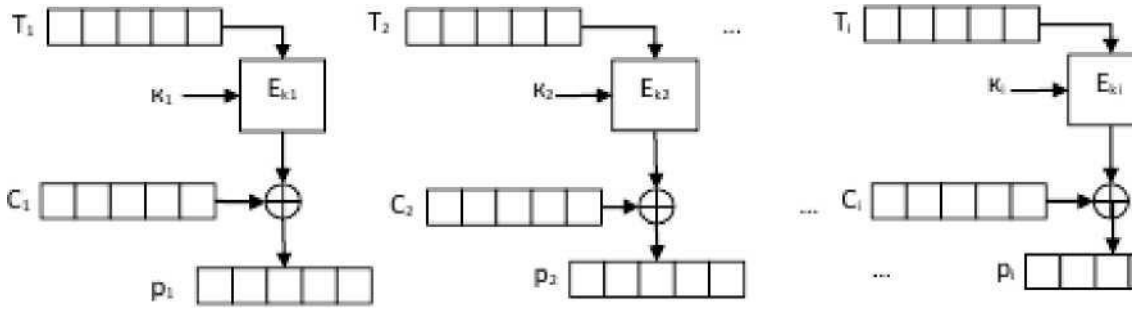


Рис.7.13. Схема розшифрування в режимі CTR

Переваги режиму CTR:

- На відміну від режиму OFB, тепер допускається паралельне виконання операцій шифрування (розшифрування) блоків.
- Для шифрування/розшифрування використовується одна функція. • Операції шифрування/розшифрування є симетричними.
- Можливість розпаралелення шифрування/розшифрування блоків. Недоліки:
- Некритичність по відношенню до ставки і втрати блоків.
- Використання псевдо генератора випадкових чисел.

Узагальнені відомості про стандартні режими шифрування представленні в Таблиці 7.1.

Таблиця 7.1. Узагальнені відомості про стандартні режими шифрування

Режим	**Перетворення**	**Co**	**Переваги**	**Недоліки**
ECB	Ci=Eki(Pi) Pi=Dki(Q	-	Можливість розпаралелення шифрування	Некритичність до вставки і втрати блоків
CBC	CI= Eki(pi XOR Ci—i) Pi= Ci—iXOR	+	Критичність до вставки і втрати блоків Автентифікація з використанням MAC	Неможливість розпаралелення шифрування
CFB	CI= Eki(Ci-i) XOR pi Pi = Eki(Ci-i) XOR CI	+	Критичність до вставки і втрати блоків Використовується одна функція шифрування Симетричність функцій шифрування/розшифрування	Неможливість розпаралелення шифрування
OFB	CI= pi XOR Oi Pi = CI XOR Oi	+	Критичність до вставки і втрати блоків Використовується одна функція шифрування Симетричність функцій шифрування/розшифрування	Неможливість розпаралелення шифрування
CTR	CI= pi XOR Oi Pi = CI XOR Oi	-	Можливість розпаралелення шифрування Використовується одна функція шифрування Симетричність функцій шифрування/розшифрування	Некритичність до вставки і втрати блоків

7.2. SP-мережа

На початку 70-их років XIX ст. Х.Фейстелем (IBM) було запропоновано два підходи до побудови блокових шифрів: SP-мережі і мережі Фейстеля.

Ідея, що лежить в основі SP-мереж, полягає в побудові крипостійкої системи за допомогою багаторазового застосування відносно простих криптографічних перетворень, в якості яких К.Шеннон запропонував використовувати перетворення **підстановки** (substitution) і **перестановки** (permutation).

У найпростішому варіанті SP-мережа (*Substitution-Permutation network,*

підстановочноперестановочна мережа) являє собою «сендвіч» з шарів двох типів, які використовуються багаторазово по черзі (Рис.7.14). Перший тип шару - P-шар, що складається з P-блоку великої розрядності, за ним йде другий тип шару - S-шар, що представляє собою велику кількість S-блоків малої розрядності, потім знову P-шар і т.д.

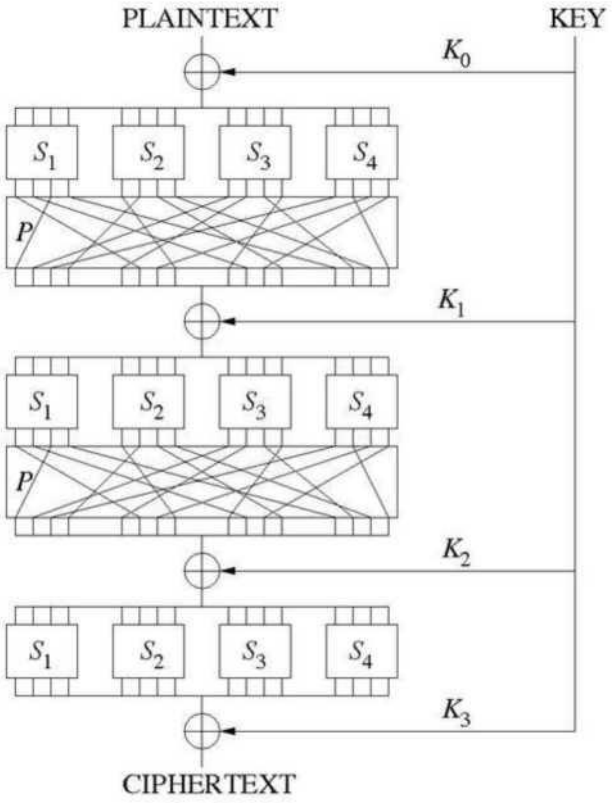


Рис.7.14. Схема SP-мережі

S-блок заміщає невеликий блок вхідних біт на інший блок вихідних біт. Ця заміна повинна бути взаємно однозначною, щоб гарантувати оборотність. Призначення 8-блоку полягає в нелінійному перетворенні, що перешкоджає проведенню лінійного криптоаналізу.

P-блок (англ. Permutation box or P-box) - отримує на вхід виводи S-блоків, змінює місцями все біти і подає результат S-блокам наступного раунду. Важливою якістю P-блоку є можливість розподілити виводи одного S-блоку між входами якомога більшої кількості S-блоків.

Багаторазове використання цих перетворень дозволяє забезпечити дві властивості, які повинні бути притаманні стійким шифрами: дифузія (diffusion) і конфузія (confusion). Лінійна стадія перестановки розподіляє надмірність по всій структурі даних, породжуючи дифузію. Нелінійна стадія підстановки перемішує біти ключа з бітами відкритого тексту, створюючи конфузію.

Дифузія передбачає поширення впливу одного знаку відкритого тексту на значну кількість знаків шифротексту. Наявність у шифра цієї властивості дозволяє:

- приховати статистичну залежність між знаками відкритого тексту, інакше кажучи, перерозподілити надмірність вихідного мови за допомогою розповсюдження її на весь текст;
- не дозволяє відновлювати невідомий ключ по частинах.

Мета **конфузії** - зробити якомога більш складною залежність між ключем і шифротекстом.

Криптоаналітик на основі статистичного аналізу перемішаного тексту не повинен отримати будь-якої значної інформації про використаний ключ.

Застосування дифузії і конфузії порізно не забезпечує необхідну стійкість, надійна криптосистема виходить тільки в результаті їх спільного використання.

Лавинний ефект (avalanche) - це число символів, яке змінилося в зашифрованому тексті при зміні одного біта відкритого тексту або ключа. Чим більше лавинний ефект, тим вище

надійність шифру.

Першим криптографічним алгоритмом на основі SP-мережі був «Люцифер» (1971). В даний час з алгоритмів на основі SP-мереж широко використовується AES.

Альтернативою SP-мереж є мережі Фейстеля.

7.3. Мережі Фейстеля

Мережею Фейстеля називається метод оборотних перетворень тексту, при якому значення, обчислене від однієї з частин тексту, накладається на інші частини.

Часто структура мережі виконується таким чином, що для шифрування і розшифрування використовується один і той самий алгоритм - відмінність полягає тільки в порядку використання матеріалу ключа.

В схемі Фейстеля кожний блок розбивається на ліву (1_0) і праву (r_0) частини, над якими здійснюються S раундів шифрування. Після завершення S раундів шифрування ліва (L_s) і права (R) частини міняються місцями:

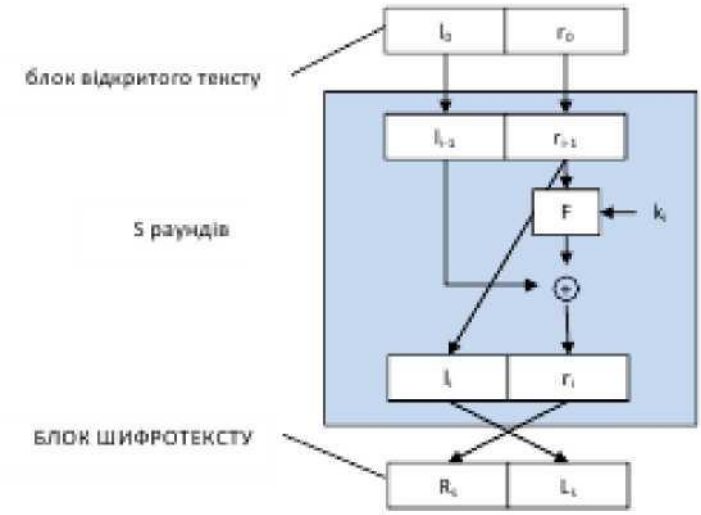


Рис.7.15. Схема мережі Фейстеля

Кожний раунд шифрування здійснюється за правилом:

$$l_i = r_i\text{-}1,\ r_i = l_i\text{-}1 + F(k_i, r_{i-1}).$$

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

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

Цікава особливість шифру Фейстеля полягає в тому, що функція раунду є оберненою незалежно від властивості функції Б.

Для створення криптостійкого шрифту залишилося визначити:

• Спосіб генерування підключів $к_i$.

- Кількість раундів S.
- Визначити функцію F.

Відповіді на ці питання були дані в ході роботи над шифром DES.

7.4. Криптосистема DES

7.4.1 Загальна характеристика

Алгоритм **DES (Data Encryption Standard)** був розроблений у 1977 році і рекомендований

Національним бюро стандартів США в якості основного засобу криптографічного захисту інформації як в державних, так і в комерційних структурах. Він став першим доступним всім бажаючим офіційним алгоритмом і першим світовим стандартом, що проіснував більше 20. Тому його слід відмітити як найважливішу віху на шляху криптографії від чисто військового використання до широкомасштабного застосування.

DES - алгоритм блокового шифрування з довжиною блоку 64 біти і симетричним ключем довжиною 56 біт. На практиці ключ має довжину 64 біти,з яких кожний восьмий використовується для контролю парності байту.

Головні риси шифру DES визначаються тим, що він ґрунтується на схемі Фейстеля з такими параметрами:

- довжина блоку - 64 біти,
- кількість раундів - 16,
- розмір ключа - 56 бітів,
- розмір кожного з підключів k_1, k_2, ..., k_N- 48 бітів.

7.4.2 Алгоритм шифрування

Структура алгоритму DES показана на рис.14.

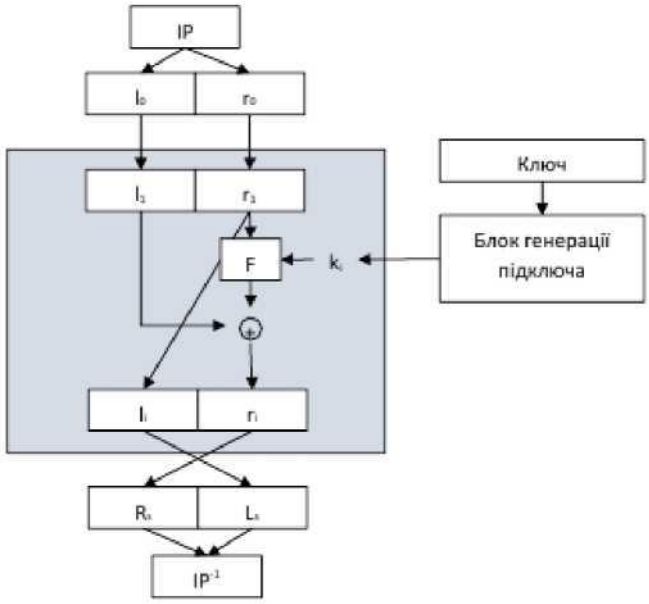


Рис.7.16. Структура алгоритму DES

Алгоритм DES описується за допомогою трьох етапів:

1. До вхідного блоку довжиною 64 біти застосовується фіксована початкова перестановка IP, яка описується виразом $(l_0,r_0) \leftarrow$ IP (Вхідний блок).

Тут l_0 та r_0 називаються лівою і правою половинами блоку, кожна з яких має довжину 32 біти.

Перестановка IP застосовується для того, щоб здійснити початкове розсіювання статистичної структури повідомлення. Вона є фіксованою і відкритою (Рис.7.17).

58	50	42	34	26	18	10	2	60	52	44	36	28	20	12	4
62	54	46	38	36	22	14	6	64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1	59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5	63	55	47	39	31	23	15	7

Рис.7.17 Матриця початкової перестановки IP

2. Виконуються 16 раундів з таких операцій:

- Права половина блоку перетворюється функцією S з використанням поточної ключової послідовності довжиною 48 біт, яка знімається з виходу блоку вироблення ключової послідовності.

• Результат перетворення правої половини складається по модулю 2 з лівою частиною, а сума записується у вихідний регістр, при цьому вихідна права половина за допомогою операції зсуву записується на місце вихідної лівої половини.

Їх можна описати рівняннями:

$$l_i = r_i \ , \ r_i = l_i \ F(r_{i-1}, k_i) \ ,$$

де $к_i$ - ключ раунду, який складається з 48-бітового підрядка 56-бітного вихідного ключа, S - функція шифрування, яка представляє собою прерстановочний шифр. Ці операції перестановки забезпечують значний рівень «дифузії повідомлення».

3. До результату 16-го раунду L16, R16 застосовується завершуюча перестановка, яка є оберненою до початкової перестановки(Рис.7.18)

40	8	48	16	56	24	64	32	39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30	37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28	35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26	33	1	41	9	49	17	57	25

Рис. 7.18. Матриця завершуючої перестановки IP

7.4.3 Структура функції F

Структура функції Fзображена на Рис.5.4.

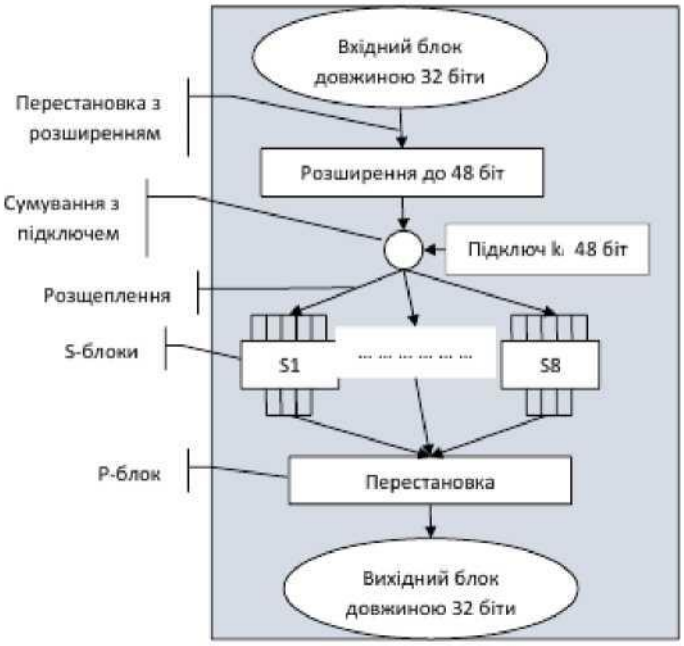


Рис. 7.19. Структура функції F

В кожному раунді перетворення S складається з п’яти кроків:

1. **Перестановка з розширенням**. Права половина з 32 бітів розширюється до 48 бітів і перемішується. Ця операція передбачає дописування у вихідну послідовність окремих бітів у відповідності з підстановкою розширення (Рис.7.20).

32	1	2	3	4	5	4	5	6	7	8	9
8	9	10	11	12	13	12	13	14	15	16	17
16	17	18	19	20	21	20	21	22	23	24	25
24	25	26	27	28	29	28	29	30	31	32	1

Рис.7.20. Матриця підстановки розширення

Підстановка розширення забезпечує, що один вхідний біт впливає на дві заміни через S-блоки, що створює «лавиноподібний» ефект - мала відмінність між двома наборами вхідних даних перетворюється у велику на виході.

2. **Сумування з підключем.** До отриманого після перестановки з розширенням

рядка з 48 бітів і підключа довжиною також 48 бітів застосовується операція виключаю чого АБО, тобто кожна пара відповідних бітів складається по модулю 2.

48-бітний підключ отримується з 56-бітного ключа у такий спосіб. Біти ключа записуються у два 28-бітні циклічних регістрів зсуву, які переміщують вміст в кожному такті на кількість бітів, що залежить від номера раунду (Рис.7.21).

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

Рис.7.21. Значення циклічного зсуву в 16-ти раундах шифрування

Результуюча послідовність отримується шляхом вибірки 48 бітів з вмісту регістрів (Рис.7.22).

57	49	41	33	25	17	9	1	58	50	42	34	26	18
10	2	59	51	43	35	27	19	11	3	60	52	44	36
63	55	47	39	31	23	15	7	62	54	46	38	30	22
14	6	61	53	45	37	29	21	13	5	28	20	12	4

Рис.7.22. Матриця вибірки з вмісту регістрів

3. **Розщеплення.** Результат сумування з підключем розщеплюється на 6 частин по 8 бітів в кожному, кожна з яких передається в один з восьми S-блоків. 4) **S-блок** . Перетворює набір з 6 бітів в набір з 4 бітів.

S-блоки є нелінійними компонентами алгоритму DES, які і забезпечують криптостійкість шрифту. Кожний S-блок представляє собою пошукову таблицю з чотирьох рядків і шістнадцяти стовпців (Таблиця 7.2). Шість бітів, що входять до блоку, визначають який рядок і стовпець необхідно використовувати для заміни. Перший і шостий біт задають номер рядка, а інші - номер стовпця. Вихід S-блоку - бітове представлення числа відповідної комірки таблиці.

Таблиця 7.2. Підстановки в S-блоках

S1															
14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	9
15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13
S2															
15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
3	13	4	7	15	2	8	14	12	0	1	19	6	9	11	5
0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9
S3															
10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
1	10	13	0	6	5	8	7	4	15	14	3	11	5	2	12
S4															
7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
13	8	11	5	6	15	9	3	4	7	2	12	1	10	14	9
19	6	9	0	12	11	7	13	15	1	3	14	5	2	3	4
3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14
S5															
2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
14	11	2	12	4	7	13	1	5	0	15	19	3	6	3	6
4	2	1	11	10	13	7	8	15	9	12	5	б	3	0	14
11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3

S6															
12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
19	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
9	14	15	5	2	8	12	3	7	0	4	19	1	13	11	6
4	3	2	12	9	5	15	10	11	14	1	7	б	0	3	13
S7															
4	11	2	14	15	9	8	13	3	12	9	7	5	10	6	1
13	0	11	7	4	9	1	10	14	3	5	12	2	15	3	6
1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	12
6	11	13	3	1	4	10	7	9	5	0	15	14	2	3	12
S8															
13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
1	15	13	3	10	3	7	4	12	5	6	11	0	14	9	2
7	11	4	1	9	12	14	2	0	6	19	13	15	3	5	8
2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11

4. **Р-блок**. Вихід S-блоків з восьми 4-бітових елементів надходить до Р-блоку, в якому відбувається перестановка (Рис.7.23).

16	7	20	21	29	12	28	17	1	15	23	26	5	18	31	16
2	8	24	14	32	27	3	9	19	13	30	6	22	11	4	25

Рис.7.23. Матриця Р-перестановки

Розшифровування в алгоритмі DES відбувається аналогічно шифруванню з тією різницею, що вибірка ключової послідовності в раундах розшифровування буде оберненою, тобто k_{16}, k_{15} ... k_1.

7.4.4 Стійкість DES

Основними недоліками DES, що суттєво зменшують рівень його безпеки, є такі:

1. **Наявність слабких ключів.** Їх виникнення пов'язане з тим, що в кожному раунді при генерації підключа використовуються два регістри зсуву. При цьому може виникати ситуація, коли для кожного підключа буде генеруватись одна, або лише дві, чотири ключових послідовності. Такі ключі називаються слабкими. Приклд слабкого ключа: 1F1F1F1F 0E0E0E0E (з урахуванням бітів парності); однакові підключі будуть генеруватись у всіх 16 раундах.
2. **Невелика довжина ключа**. На сучасному рівні розвитку

мікропроцесорних засобів довжини ключа 56 біт (або 64 біти з котролем парності) не забезпечує достатнього рівня захисту.

3. **Надлишковість ключа**. Наявність контролю парності кожного байту ключа, що і приводить до надлишковості, дозволяє відновлювати ключ при виникненні помилок в памяті. 0 **Використання статичних підстановок в S-блоках**.

Слабкість алгоритму DES, що пов'язана з невеликою довжиною ключа, стала очевидною в 1990-их роках. Основні відомі атаки на DES:

Диференційний криптоаналіз. Метод запропонований в 1991 р. ізраїльськими вченими Елі Бахамом та Еді Шаміром. Ключ 67 шифрування обчислюється за умови наявності у атакуючого можливості генерації 247 спеціально вибраних пар «відкритий-зашифрований» тексти.

4. **Лінійний криптоаналіз.** Метод запропонований в 1993 р. японцем Міцуру Мацуі. Доведена можливість обчислення ключа DES за умови наявності у атакуючого 247 пар «відкритийзашифрований» тексти.
5. **Метод Девіса.** Метод в 1994 р. запропоновали Елі Біхам і Алекс Бірюков. Він дозволяє обчислити 6 бітів ключа при наявності 250 пар або 24 біти при наявності 252 пар «відкритийзашифрований» тексти.

Таким чином всі відомі атаки на DES вимагають великої кількості пар «відкритий-зашифрований» тексти, отримання яких у свою чергу є працезатратним процесом. Тому найбільш простим способом взлому DES вважається перебір всіх варіантів ключа шифрування. В 1998 році була створена пошукова машина під назвою DES Cracker вартістю 250 000 дол., яка знаходила ключі за 56 годин.

7.4.5 Похідні від DES шифри 3DES

З метою ефективного збільшення довжини ключа, в 1978 році був запропонований алгоритм **3DES** або **потрійний DES** (**TripleDES**).

Існує 3 типи алгоритму 3DES:

- DES-EEE3 (**шифрування-шифрування-шифрування**): шифрування виконується три рази з трьома різними ключами.
- DES-EDE3 (**шифрування-розшифрування-шифрування**):

виконується операції шифровкарозшифровка-шифровка з трьома різними ключами.

- DES-EEE2 (**шифрування-шифрування**): як DES-EEE3, в якому на першому і третьому кроці використовується однаковий ключ
- DES-EDE2 (**шифрування-розшифрування**): як DES-EDE3, в якому на першому і третьому кроці використовується однаковий ключ.

Найбільш популярний різновид 3DES - це DES-EDE3 (Рис.21), для нього алгоритм виглядає так: C = *Екз(Ек-21(Екі(p')')'), p = Ек-1[1] (Ек2 (Ек~з[1](C)')').*

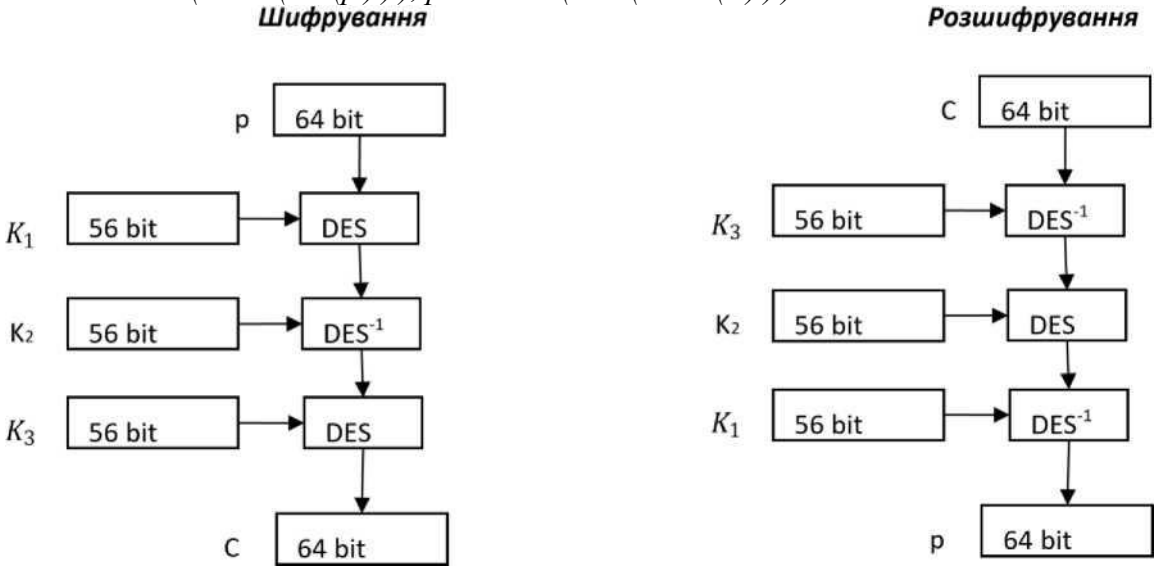


Рис.7.24. Схема шифрування/розшифрування для 3DES

При цьому, як неважко порахувати, довжина ключа стає рівною 168 бітам.

Логічним виглядає запитання при використання «подвійного» 2DES, в якому криптотекст C отримується з вхідного тексту P в результаті «двокрокового» перетворення C= EK2(EK1(P)), де K1 і K2 - ключі шифрування. Такий алгоритм можливий, але він вразливий до атаки «зустріч посередені». За допомогою цієї атаки криптоаналітик може отримати обидва ключі K1 і K2 для 2DES за наявності лише двох пар «відкритий-зашифрований» тексти P1- C1, P2-C2 наступним способом:

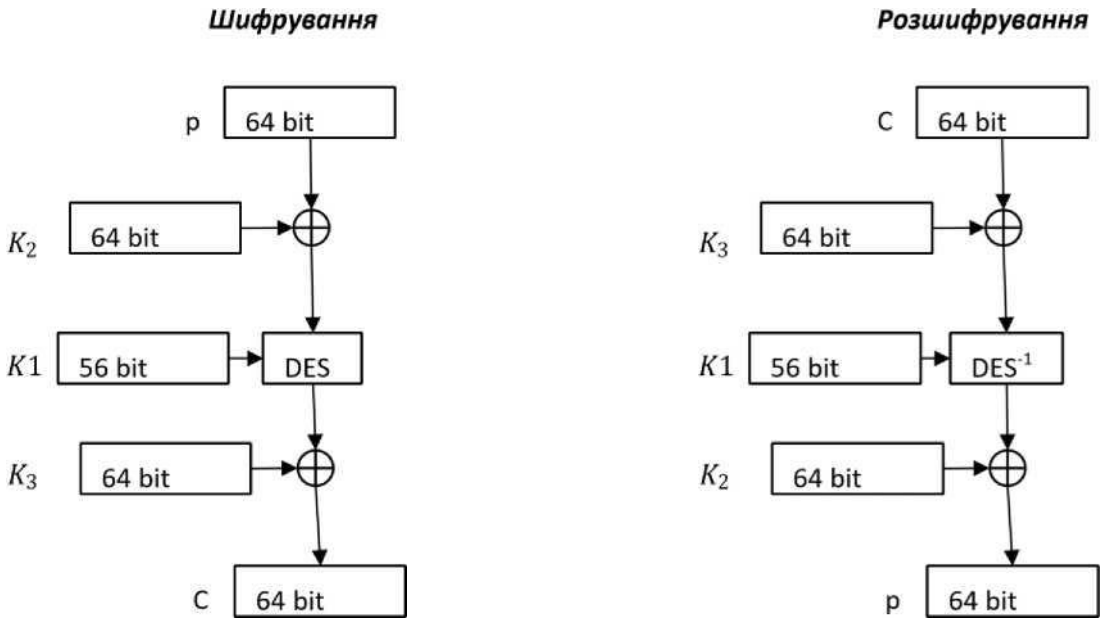
1. Виконуємо шифрування EKX(P1) для всіх X ключів.
2. Виконуємо розшифрування DKY(C1) для всіх Y ключів.
3. Порівнюємо результати виконання п.1 і п.2. Якщо вони співпали, то KX=K1 і KY=K2.

Але число таких співпадінь може бути більшим і оцінюється як ~248. Тому для виключення «хибних» ключів необхідно повторити п.1-п.3 з парою P2-C2.¶

Ймовірність наявності більш ніж одного співпадіння ~2^{-16}. Тобто така атака вимагає приблизно в 2 рази більше обчислень, ніж при переборі ключів DES.

В 1996 р. запропонований алгоритм **DESX**. Суть алгоритму полягає в тому, що перед

виконанням одноразового DES і після нього на дані операцією XOR накладаються різні 64-



бітні фрагменти ключа: C = $K3 \oplus DESK1(K2 \oplus p)$, $p = K2 \oplus DESK -1\ 1(K3 \oplus c)$

Операція накладання фрагмента ключа на вхід і / або вихід алгоритму шифрування називається **відбілюванням**, сенс якого полягає в тому, щоб перешкодити криптоаналітику отримати для дослідження алгоритму пари «відкритий текст - шифротекст».

DESX повністю сумісний з алгоритмом DES в разі, якщо К2 = К3 = 0.

Оскільки К2 та К3 мають розмір по 64 біта, а К1 - це звичайний 56-бітний ключ DES, то повний розмір ключа DESX становить 184 біта.

Запропоновано ряд різновидів алгоритму DESX:

- З використанням змінного розміру ключа шифрування, попередньо застосувавши до нього хешування за алгоритмом SHA-1.
- З використанням 120-бітного ключа шифрування (при цьому К3 встановлюється рівним $К_2$). о Замість обох операцій XOR виконується складання по модулю 264.

Загалом DESX вважається кращим, ніж DES. Проте в алгоритмі DESX було знайдено декілька уразливостей, що перешкодило широкому поширенню цього алгоритму.

7.4.6 DES і шифрована файлова система EFS

Шифрована файлова система (Encrypting File System, EFS) - це базова технологія, що реалізує шифрування на рівні файлів в операційних системах Microsoft Windows NT (починаючи з Windows 2000 і вище, за винятком «домашніх» версій - Windows XP Home Edition, Windows Vista Basic і Windows Vista Home Premium і т.п.).

Система надає можливість «прозорого шифрування» даних, що зберігаються на розділах з файловою системою NTFS, для захисту конфіденційних даних від несанкціонованого доступу при наявності фізичного доступу до комп'ютера.

В Windows 2000 EFS використовувала один алгоритм шифрування - це DESX, який є спеціальною модифікацією широко поширеного стандарту DES. В інших версіях Windows EFS використовує різні симетричні алгоритми шифрування (Таблиця 3).

EFS використовує симетричне шифрування для захисту файлів, а також шифрування, засноване на парі відкритий / закритий ключ для захисту випадково згенерованого ключа шифрування для кожного файлу. За замовчуванням закритий ключ користувача захищений за допомогою шифрування користувальницьким паролем, і захищеність даних залежить від стійкості пароля користувача.¶

Таблиця 3. Алгоритми симетричного шифрування в ОС Windows

Операційна система	Алгоритм шифрування за замовчуванням	Альтернативні доступні алгоритми
Windows 2000	DESX	(none)
Windows XP RTM	DESX	3DES
Windows XP SP1	AES	3DES, DESX
Windows Server 2003	AES	3DES, DESX
Windows Vista	AES	3DES, DESX
Windows Server 2008	AES	3DES, DESX (?)

Процедура шифрування здійснюється драйвером EFS і полягає у такому (Рис.7.25):

1. Для кожного файлу (каталогу), що шифрується, випадковим чином генерується ключ симетричного шифрування - так званий FEK (File Encryption Key).
2. Файл (каталог) шифрується за допомогою алгоритму симетричного шифрування з використанням в якості ключа FEK.
3. FEK захищається шляхом асиметричного шифрування за алгоритм RSA відкритим ключем користувача.
4. Зашифрований FEK зберігається в заголовку зашифрованого файлу (каталогу).

Рис.7.25. Схема шифрування при використанні EFS

8 Асиметричні алгоритми шифрування

8.1. Передумови виникнення асиметричних систем

Традиційні криптографічні системи мають два суттєвих недоліки:

– Проблема розподілення ключів в управління ними. Система з n учасниками вимагає використання n/2 ключів і стільки ж безпечних каналів для їх розповсюдження. При зміні ключа одним з учасників доводиться генерувати і розподіляти (n-1) ключ. А додавання нового учасника вимагає генерування і розподілення n нових ключів.

– Проблема автентифікації. Традиційні криптосистеми не забезпечують потребу користувачів у використанні електронного еквіваленту підпису: будь-яке повідомлення, що відправлене одним з них, може бути відправлене і іншим.

Намагання усунути ці недоліки спонукало дослідників до пошуку криптографічних систем нового типу. Ідея такої системи була опублікована в 1976 р. у піонерській роботі У. Діффі і Д.Хеллмана «Нові напрями в криптографії». Криптографічні системи нового типу отримали назву криптосистем з публічними ключами або асиметричних криптосистем.

Наведемо приклади проблем, які вирішуються за допомогою асиметричних систем.

– Проблема зберігання паролів на комп'ютері. Якщо зберігати паролі на магнітному диску, то зловмисник може прочитати їх і використати для несанкціонованого доступу. Тому необхідно організувати зберігання паролів на диску так, щоб унеможливити таке зчитування.

– Проблема радіолокації в ППО. При перетині літаком кордону у нього запитується пароль. Зловмисник може перехопити пароль і використати його для нелегального перетину кордону.

– Проблема віддаленої взаємодії. Так, при взаємодії банка з клієнтом на початку сеансу банк запитує клієнта ім'я і секретний пароль, який при використанні відкритого каналу зв'язку також може бути перехоплений.

8.2. Модель криптосистеми з публічними ключами

Ідея У. Діффі і Д.Хеллмана полягала у такому. Кожний користувач и криптосистеми створює (або отримує з надійного джерела) пару узгоджених алгоритмів P_u і S_u (Рис.8.1). Алгоритм P_u оголошується публічним, а алгоритм S_u утримується в секреті. Ці алгоритми в залежності від застосування мають задовольняти окремим з наступних умов:

У1. Алгоритми P_u і S_u ефективні, тобто не вимагають занадто великого часу обчислень і великих обсягів пам'яті.

У2. Su(Pu(m))=m для будь-якого користувача и і будь-якого повідомлення m.

У3. Неможливо виходячи з алгоритму Pu знайти такий алгоритм Su*, що Su*(Pu(m))=m для всіх m.

Умову **У3** сформовано неточно і їх зміст має уточнюватись в залежності від області застосування

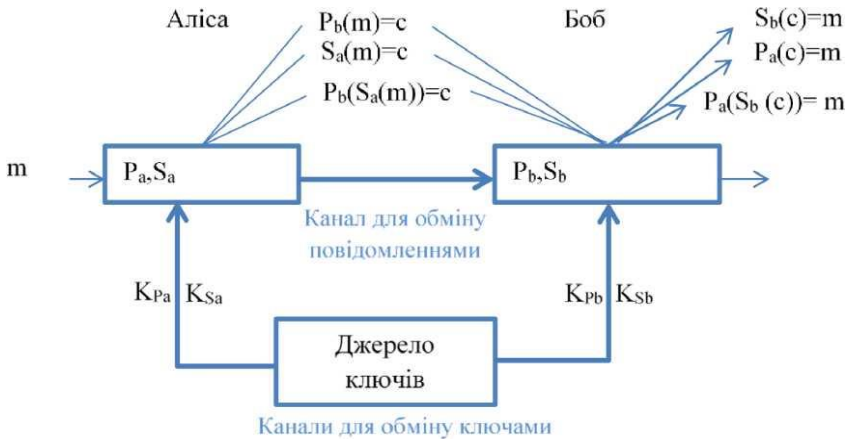


Рис.8.1. Модель асиметричної криптосистеми (Діффі-Хеллмана)

Розглянемо окремі випадки застосування цих умов.¶

Нехай виконуються умови **У1- У3**.

В цьому випадку забезпечується можливість **шифрування** повідомлень (Рис.8.2).

Дійсно, якщо Аліса хоче надіслати Бобу зашифроване повідомлення m, то вона спочатку знаходить публічний ключ Боба P_B і за його допомогою шифрує m: c= P_B(m). Боб може відновити m, скористувавшись своїм секретним алгоритмом S_B, оскільки за умовою **У2** S_B(c)= S_B(P_B(m))= m. Виконання умови **У1** забезпечує практичність системи. А виконання умови **У3** дозволяє публікування алгоритму утворення відкритого ключа.

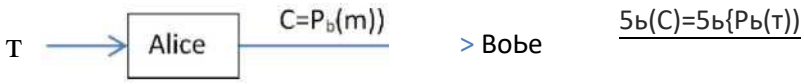


Рис.8.2. Шифрування в асиметричній криптосистемі

Нехай виконуються умови **У1,У4 і У5**.

В цьому випадку забезпечується можливість використання **цифрового підпису** (Рис.8.3). Дійсно, якщо Аліса хоче надіслати Бобу підписане повідомлення m, то вона може спочатку застосувати до нього секретний алгоритм S_A і відправити c= S_A(m). Боб може відновити m, скористувавшись публічним алгоритмом P_A, оскільки за умовою **У4** P_A(c)= P_A (S_A(m))= m.

Виконання умови **У1** забезпечує практичність системи. А виконання умови **У5** дозволяє публікування алгоритму утворення цифрового підпису.

Рис.8.3. Цифровий підпис в асиметричній криптосистемі

Нехай виконуються всі умови **У1-У5**.

В цьому випадку забезпечується можливість **шифрування підписаних повідомлень** (Рис.8.4). Дійсно, якщо Аліса хоче надіслати Бобу повідомлення т в зашифрованому вигляді зі своїм підписом, то вона може спочатку підписати його за допомогою власного секретного алгоритму S_A, потім зашифрувати за допомогою публічного алгоритму Боба P_B і відправити

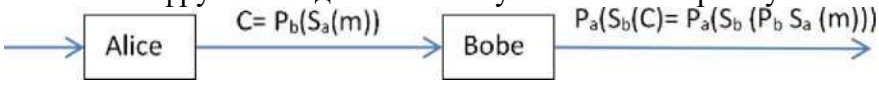


Рис.8.4. Шифрування підписаного повідомлення в асиметричній криптосистемі

c= P_B (S_A(m)). Боб може відновити т, скористувавшись публічним алгоритмом P_A і своїм секретним SB, оскільки за умовою **У2,У4** P_A(S_B (c))= P_A (S_B (P_B(S_A(m))))= P_A(S_A(m))= m.

Виконання умови **У1** забезпечує практичність системи. А виконання умови **У3,У5** дозволяє публікування алгоритми шифрування і утворення цифрового підпису.

8.3. Поняття односторонньої функції-пастки

У. Діффі і Д.Хеллман запропонували використовувати для шифрування одностороннюю функцію-пастку.

Означення. Односторонньою функцією називається функція f: A→B з такими властивостями (Рис.8.5):

1. f(a) легко обчислюється для будь-якого a Є A.
2. Чисельно неможливо знайти f^{-1}(b) майже для всіх b Є B

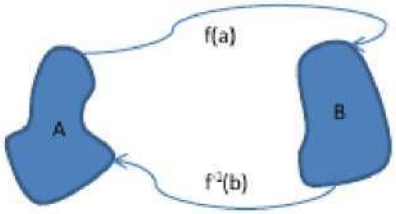


Рис.8.5. Відображення f і оберненого f^{-1}

Означення. Односторонньою функцією-пасткою (trapdoor function) називається така

одностороння функція f(a), для якої $f^{-1}(b)$, b є B легко обчислюється, якщо відома деяка додаткова інформація (існує лазівка).

Означення. Односторонню функцію-пастку називають криптографічною функцією.

Приклад (криптографічної функції). Візьмемо телефонний довідник. Для шифрування кожної букви відкритого повідомлення із довідника випадково вибирається прізвище, що починається з такої букви. Відповідний телефонний номер утворює шифр для даного випадку появи букви; таким чином, система шифрування є поліалфавітною. Результат шифрування слова «шифр» може бути таким:

Бунвз	Прізвище	Шифр
ш	Шарапов	4013435
и	Ивліч	5556753
ф	Фоменко	4517390
р	Рудчечко	6738399

Рис.8.6. Приклад шифрування з використанням телефонного довідника

Криптотекст утворюється написанням всіх номерів з правого стовпчика.

Отримати телефон за прізвищем - проста задача, а ось обернена задача визначення прізвища за номером значно складніша. Але вона теж може бути легко розв'язана, якщо відомий деякий секрет- лазівка. Таким секретом може бути обернений телефонний довідник.

8.4. Задача рюкзака

Першим алгоритмом для узагальненого шифрування з відкритим ключем став алгоритм рюкзака, розроблений Ральфом Меркле і Мартіном Хеллманом.

Проблему рюкзака можна сформулювати так: *Нехай задано множину натуральних чисел A = (a1,a_2 ,..., a_n) і натуральне число S; потрібно встановити, чи існує така підмножина множини A, сума елементів якої б дорівнювала S.*

Еквівалентним є наступне формулювання: *Чи існує такий набір чисел x_i з (0,1), i < n, для якого* $\Sigma a_i x_i = S$ $(1 <= i <= n)$*?*

Свою назву задача отримала в зв'язку з тим, що може бути поставлена також в наступному вигляді. Є набір предметів з відомими вагами і рюкзак, який може витримати вагу, що не перевищує задану. Чи можна вибрати набір предметів для навантаження в рюкзак так, щоб вони в точності мали максимально допустиму вагу?

Приклад. Для набору з 10 цифр A = (43, 129, 215, 473, 903, 302, 561, 1165, 697, 1523) і S=3231 методом перебору можна знайти, що 3231 = 129 + 473 + 903 + 561 + 1165, тобто проблема рюкзака розв'язується.

В принципі рішення завжди може бути знайдено повним перебором підмножин A і перевіркою, яка з їх сум дорівнює S. Але при великих n доведеться перебрати 2^n варіантів. Навіть для n = 300 пошук серед 2^{300} підмножин не піддається обробці. Суть тут у тому, що невідомі алгоритми, які мають суттєво меншу складність у порівнянні з повним перебором.

"Проблема рюкзака" виявляється досить складною, її рішення з поліноміальною складністю в даний час не відомо.

Ідея побудови системи шифрування на основі проблеми рюкзака полягає у виділенні деякого підкласу задач про укладання рюкзака, що розв'язуються порівняно легко, і "маскування" задач цього класу (за допомогою деякого перетворення параметрів) під загальний випадок. Параметри підкласу визначають секретний ключ, а параметри модифікованої задачі - відкритий ключ.

В якості задачі, що легко розв'язується, Р. Меркль і М. Хеллман в 1978 р. запропонували задачу про укладання "суперзростаючого" рюкзака. Її суть полягає у такому.

Означення. Назвемо суперзростаючою послідовність таких натуральних чисел (b_1, b_2 ,..., b_n), що $b_i > \Sigma b_j$, для $1 <= j <= i-1$, $2 < n$.

Неважко переконатися в тому, що задача рюкзака для суперзростаючої послідовності може бути вирішена за допомогою виконання наступного алгоритму.

Алгоритм розв'язання задачі «суперзростаючого» рюкзака:

Введення: натуральні числа n,S, суперзростаюча послідовність натуральних чисел B=(bi, b2 ,..., bn), набір чисел Xi з (0,1).

Виведення: такі bi і xi, для яких Ebi xi = S.

Крок 1. Покласти i = n.

Крок 2. Якщо i> 1, то покласти xi рівним 1 і S рівним (S - bi), якщо S> bi, і покласти xi рівним 0 в іншому випадку.

Крок 3. Покласти і рівним (i-1) і повернутися до кроку 2.

Приклад. Для суперзростаючої послідовності з 5 цифр B = (43, 129, 215, 473, 903) і S=1161 знаходимо: x_5=1 - оскільки 903<1161; x4=0 - оскільки 1161- 903=258<473; x_3=1 - оскільки 215<258; X2=0 - оскільки 258-215=43<129; xi=1 - оскільки 43=43. Таким чином, X=(1,0,1,0,1). Дійсно, 43+215+903=1161.

Набір A з n значень визначає криптографічну функцію f (x) так. Будь-яке число x в інтервалі 0 < x < 2n - 1 може бути задано двійковим представленням з n розрядів, в якому за необхідності добавляються початкові нулі. Таким чином, 1, 2 і 3 представляються в вигляді 0. . . 01, 0. . . 010, 0. . . 011, тоді як 1. . . 111 є представленням для (2n - 1). Тепер визначимо f (x) як число, що отримується з A сумуванням всіх таких ai, у яких відповідний розряд в двійковому поданні x дорівнює 1. Так, f(1) = f (0... 001) = an, f(2)=f (0... 010) = an-1, f (3) = f (0... 011) = an-1 + an, і т. д.

Використовуючи векторне множення, можна записати, що **f** (x) = A**X**, де **X** - вектор-стовпець двійкового представлення x.

"Рюкзачні вектори" A можуть бути використані в якості основи криптосистеми. Для цього пропонується наступний механізм шифрування. Початкове повідомлення спочатку кодується і розбивається на n-розрядні блоки. Якщо це необхідно, останній блок доповнюється наприкінці нулями. Кожен з n- розрядних блоків шифрується за допомогою обчислення значення функції f для цього блоку.

Алгоритм шифрування в задачі рюкзака:

Введення: натуральне число n>1, послідовність натуральних чисел A=(ai, a2 ,..., an), вхідне повідомлення p.

Виведення: шифротекст C.

Крок 1. Представити p у вигляді бінарної послідовності.

Крок 2. Розбити отриману бінарну послідовність на n-розрядні блоки pi=Pi1pi2...pin.

Крок 3. Зашифрувати кожний блок за допомогою перетворення $C_i = \Sigma\, p_{ij} \cdot a_j,\ _j = 1 \ldots n.$

Крок 4. Отримати шифротекст C= (Ci; C2; ...; Ci)

Приклад. Якщо вихідне повідомлення написано по-англійськи, природним способом кодування є заміна однієї літери двійковим представленням її порядкового номера в алфавіті. Для цих цілей буде потрібно п'ять бітів. Співставим кожному символу 5-бітний код: пробіл - 00000, A - 00001, B - 00010, C - 00011 і т.д.

Розглянемо попередній 10-набір і вихідне повідомлення HEALTH. Так як блоки шифрування складаються з 10 розрядів, то поділ на блоки вихідного тексту дасть:HE AL TH.

Відповідні 3 двійкових послідовності: 0100000101, 0000101100, 1010001000.

Для них f(0100000101) = 129 + 1165 + 1523 = 2817, f(0000101100) = 903 + 561 + 1165 = 2629, f(1010001000) = 43 + 215 + 561 = 819.

Отже, криптотекстом буде набір (2817, 2629, 819).

Визначену таким чином на основі функції рюкзака f (x) криптосистему можна використовувати як класичну. Тоді криптоаналітик знаходить n-набір A і після цього вирішує ще й задачу про рюкзак. Але для розшифрування легальний одержувач також повинен розв'язувати задачу про рюкзак. Це означає, що розшифрування однаково важко і для криптоаналітика, і для легального одержувача. Це означає, що криптосистема дуже недосконала. В хороших криптосистемах розшифрування має бути значно важче для криптоаналітика, ніж для легального одержувача.

Удосконалити таку криптосистему і перетворити її на систему з відкритим ключем можна у такий спосіб.

Якщо розглядати суперзростаючий набір B як основу криптосистеми, то

розшифрування буде однаково легким як для крипто аналітика, так і легального одержувача. Щоб уникнути цього, ми "збовтаємо" B таким чином, щоб результуючий набір A не був суперзростаючим і виглядав як довільний вектор рюкзака.

Для цього виберемо ціле m> Ebi. Так як B - суперзростаючий набір, то m велике в порівняно з усіма числами з B.

Виберемо інше ціле t, що не має спільних множників з m. Такий вибір t гарантує, що знайдеться інше ціле t^{-1}, таке, що tt^{-1} = 1(mod m). Ціле t^{-1} можна назвати зворотним до t. Зворотний елемент може бути легко обчислений по t і m.

Тепер утворюємо добутки tbi, i = 1,. . . , n, і зведемо їх за модулем m: нехай ai - найменший додатній залишок tbi по модулю m. Результуючий вектор A = (a1, a_2..., an) розглядається як відкритий ключ для шифрування. Алгоритм шифрування для n-розрядних блоків вихідного повідомлення описаний вище.

Приклад. Нехай B = (1, 3, 5, 11, 21, 44, 87, 175, 349, 701), m=1590, t=43. Відмітимо, що t і m взаємно прості, оскільки 43 • 37 = 1591 = 1 (mod 1590) і тому t^{-1} = 37.

Знаходимо значення A: 43 • 1 =43, 43 • 3 =129, 43 • 5 =215, 43 • 11 =473, 43 • 21 =903, 43 • 44 = 1892=1590 + 302,43 • 87 = 3741=2 • 1590 + 561, 43 • 175= 7525=4 • 1590 + 1165, 43 • 349=15007=9 • 1590 + 697, 43 • 701=30143=18 • 1590 + 1523.

Числа t, t^{-1} і m зберігаються як секретна лазівка. Тоді легкий алгоритм розшифрування для легального одержувача полягатиме у такому. Оскільки легальний одержувач знає t^{-1} і m, він у змозі знайти A з відкритого ключа B. Після отримання блоку C криптотекста, який є цілим числом, легальний одержувач обчислює C'=t^{-1}C і його найменший додатній залишок C' (mod m). При розшифруванні він вирішує легку задачу укладання рюкзака для значень B і C'. Рішенням є єдина послідовність p з n бітів. Вона також є і правильним блоком вихідного повідомлення, тому що будь-яке рішення p' задачі укладання рюкзака, яке визначається B і C', повинна дорівнювати p. Дійсно, C = t-1C' = t-1Bp' = Ap' (mod m).

Алгоритм розшифрування в задачі рюкзака:

Введення: натуральне число n>1, суперзростаюча послідовність натуральних чисел B=(bi, b_2 ,..., bn), натуральні числа m> Ebi, і t=1(mod m), шифротекст C= (C_1; C2; ...; Ci).

Виведення: відкрите повідомлення p.

Крок 1. Знайти таке дійсне t^{-1}, що tt^{-1} = 1(mod m).

Крок 2. Для кожного блоку шифротексту обчислити Ci'= t^{-1}Ci(mod m).

Крок 3. Розв'язати задачу «суперзростаючого» рюкзака для B і кожного Ci', отримавши відповідну бінарну послідовність pi з n бітів.

Крок 4. Шляхом декодування pi отримати текст повідомлення p.

Приклад. Розшифруємо отриманий раніше крипто текст (2817, 2629, 819). Множимо ці цифри на t^{-1} = 37 і знаходимо залишок по модулю 1590:

37 • 2817= 104229= 65 • 1590 + 879 ^ 879 (mod 1590), 37 • 2629= 372629 = 61 • 1590 + 283 ^ 283 (mod 1590), 37 • 819= 30303= 19 • 1590 + 569 ^ 93 (mod 1590), Число 879 і суперзростаючий набір B дають 10-розрядну послідовність 0100000101. Дійсно, так як 879 > 701. Числа з A тепер порівнюються з різницею 879 - 701 = 178. Перше число, справа наліво, менше ніж 178, є 175. Наступне число 3 дорівнює різниці 178 - 175 = 3. Позиції 3, 175, і 701 у A є відповідно 2, 8 і 10. Декодуючи послідовність 0100000101, отримуємо перші дві літери тексту: HE.

Число 283 дає послідовність 00001011000. Перше число, справа наліво, менше ніж 283, є 175.

Наступне число менше різниці 283 - 175 = 108 є 87. Наступне число 21 дорівнює різниці 108 - 87 = 21. Позиції 21, 87, і 175 у A є відповідно 5, 7 і 8. Декодуючи послідовність 0100000101, отримуємо перші дві літери тексту: AL.

Декодуючи всі три послідовності, легальний одержувач обчислює вихідне повідомлення: HEALTH.

8.5. Протоколи асиметричної криптографії

8.5.1 Протокол Діффі-Хеллмана

Проблема розподілу ключів була вирішена в тій же основній роботі Діффі і Хеллмана, в

якій була розроблена схема шифрування з відкритим ключем. Цей протокол розподілу ключів, названий протоколом Діффі-Хеллмана обміну ключами, дозволяє двом сторонам узгодити секретний ключ по відкритому каналу зв'язку. Стійкість протоколу ґрунтується на складності проблеми дискретного логарифмування в кінцевій абелевій групі.

В алгоритмі Діффі-Хеллмана симетричний сеансовий ключ не генерується і не розподіляється між учасниками. Алгоритм забезпечує формування одного й того ж секрету двома сторонами, який можна використовувати для побудови сеансового ключа в симетричному алгоритмі. Ця процедура називається *погодженням ключа*: сторони погоджують ключ, який будуть

використовувати. Вона полягає у такому: кожна із сторін отримує секретне і відкрите значення (пара ключів ДіффіХеллмана); об'єднання секретного значення однієї із сторін з відкритим значенням іншої забезпечує створення одного й того ж самого секретного значення.

Протокол **Діффі-Хеллмана** спрощується, якщо зв'язок здійснюється тільки між двома абонентами. В цьому випадку:

1. Спочатку генеруються два великих простих числа n і q. Ці два числа не обов'язково зберігати в секреті.
2. Один з партнерів A генерує випадкове число *x* і посилає іншому учаснику B значення $L=Q^x(\text{mod } N)$.
3. Після отримання L партнер B генерує випадкове число *y* і посилає A обчислене значення $M=Q^y \text{ (mod } N)$.
4. Партнер A, отримавши M, обчислює $Kx = M^x \text{ (mod } N)$, а партнер B обчислює $Ky = L^y \text{ mod } N$.

Алгоритм гарантує, що Ky=Kx і можуть бути використані в якості секретного ключа для шифрування. Адже навіть перехопивши числа L і M, важко обчислити Kx або Ky.

Основні повідомлення в протоколі Діффі-Хеллмана представляються наступною діаграмою :

A~B: N,Q;

A: x; A^B: $L=Q^x(\text{mod } N)$;

B: y; B^A: $M=Q^y \text{ (mod } N)$;

A: $Kx = M^x \text{ (mod } N)$;

B: $Ky = L^y \text{ mod } N$);

Приклад. Візьмемо N = 2147483 659 і G = 2. Тоді маємо:

A	**B**
x=12	y=34
$L=2^{12}(\text{mod } 2\ 147\ 483\ 659)= 4096$	$M=2^{34}(\text{mod } 2\ 147\ 483\ 659)= 2147483571$
$Kx = 2147483571^{12}(\text{mod } 2\ 147\ 483\ 659)= 1082609919$	$K_y = 40\ 96^{34} \text{ (mod } 2\ 147\ 483\ 659)= 1082609919$

Розглянута система має суттєвий недолік: у Аліси немає впевненості, що вона листується саме з Бобом. Це може привести до наступної атаки, яка умовно називається «людина посередині»:

- Аліса домовляється про ключ з Євою, думаючи, що переписується з Бобом;
- Боб веде переговори про ключ з Євою, вважаючи, що переписується з Алісою;
- Єва може вивчати повідомлення, т.я. вони проходять через неї як через комутатор. Оскільки вона не вносить змін в відкритий текст, її дії не можуть бути виявлені.

Отже, можна зробити висновок про те, що самого по собі протоколу Діффі- Хеллмана

не достатньо для забезпечення секретності розподілення ключів. Але на основі протоколу ДіффіХеллмана можна створити справжню криптосистему, яка називається системою Ель-Гамаля.

8.5.2 Шифр Шаміра

Перший шифр, який дозволив організувати обмін секретними повідомленнями по відкритим лініям для осіб, які не мають ніяких захищених каналів і секретних ключів, був запропонований Шаміром (Adi Shamir). Наведемо його опис.

Нехай абонент А хоче передати повідомлення m абоненту В по відкритій лінії зв'язку. Для цього: абонент А вибирає випадкове велике просте число p і відкрито передає його абоненту В. Потім абонент А вибирає два секретні числа c_A і d_A такі, що c_A d_A mod (p-1) =1. Абонент В також вибирає два секретні числа cB і dB такі, що cB dB mod (p-1) =1.

Після цього абонент А передає своє повідомлення m за допомогою три ланкового протоколу. Якщо m<p, то повідомлення m передається зразу; якщо ж m>=p, то повідомлення представляється у вигляді m1, m2, .. .,mt, де всі mi<p, а потім передаються послідовно m1, m2, ...,mt . При цьому для кожного mi рекомендується вибирати випадково нові пари (СА ДА) і (св ,d в), в іншому випадку надійність системи стає нижчою. Такий протокол може використовуватись для передачі секретних ключів.

Далі розглядається тільки випадок m<p.

Протокол Шаміра складається з таких кроків:

1. Абонент А обчислює число xi = $m^{Cл}$ *mod p* і пересилає його абоненту В.
2. Абонент В, отримавши xi, обчислює $X2 = xi^{Св}$ *mod p* і пересилає його абоненту А.
3. Абонент А, отримавши X2, обчислює xз = $X2^{dA}$ *mod p* і пересилає його абоненту В.
4. Абонент В, отримавши xз, обчислює X4 = $X4^{dB}$ *mod p* і пересилає його абоненту А.

Неважко довести, що: (а)x4=ш; (б) зловмисник не може знати m.

Дійсно, будь-яке ціле число e>=0 можна представити у вигляді e=k(p-1)+r, r=e mod (p-1).

За теоремою Ферма: x^e *mod p* = $x^{k(p-1>+Y}$*mod p = xrmod p.*

Тому x4 = *xздв mod p = fadA^de mod p = (хісвУйАйв mod p = (mcA~)cBdAd*$_B$ *mod p =* $m.cAc^Bd^Ad^B$ *mod p =* $m.cAc^Bd^Ad^B$ *mod (p-p mod p =* $m.cAc^Bd^Ad^B$ *mod (p-p mod p = rn.cACBm.od* $(_P\text{-}i)d^Ad^B$ *mod (p-1) mod p = nod mod p = m.*

Зловмиснику для знаходження m доведеться розв'язувати задачу дискретного логарифмування, що для великих p неможливо.

Для знаходження пар (СА ДА) і (св ,d в) можна запропонувати наступний спосіб. Перше число (cA)вибирається випадково так, щоб воно було взаємно простим з (p-1), а друге число (d_A) знаходиться за допомогою розширеного алгоритму Евкліда.

Приклад. Нехай А хоче передати В m=10. А вибирає p=23, c_A=7 (оскільки gcd(7,22)=1) і обчислює d_A=19. Аналогічно, В вибирає c_B=5 і обчислює d $_в$=9. Переходимо до протоколу Шаміра:

1. x1=107 mod 23=14.
2. x2=145 mod 23=15.
3. x3=1519 mod 23=19.
4. x4=199 mod 23=10.

Таким чином, В отримав m=10.

8.5.3 Шифр Ель-Гамаля

Шифр, запропонований Ель-Гамалем (Taher ElGamal), як і шифр Шаміра, вирішує задачу передачі між абонентами зашифрованих повідомлень по незахищених каналах, але використовує при цьому лише одну пересилку повідомлення. Фактично тут використовується протокол ДіффіХеллмана для формування секретного ключа і далі повідомлення шифрується шляхом множення його на цей ключ.

Для групи абонентів A,B,C,... вибирається велике просте число p і деяке число q, 1<q<p-1, таке, що всі числа з множини {1,2,..., p-1} можуть бути представлені як різні степені

числа q mod p. Числа p і q передаються всім абонентам у відкритому вигляді.

Потім кожний абонент вибирає своє секретне число c_i, $1< c_i <p-1$ і обчислює відповідне йому число

$$dj = q^{Ci} \ mod \ p$$

В результаті отримуємо таблицю ключів:

Абонент	Секретний ключ	Відкритий ключ
A	CA,	dA
B	св	dB
C	сс	de

Покажемо, як А пересилає В повідомлення m:

1. Абонент А вибирає випадкове число k, 1<=k<=p-2, обчислює числа |d_A = qC^A mod p,e=m d^CA mod P P і передає пару чисел (r,e) абоненту В.

2. Абонент В, отримавши (r,e), обчислює ^‘ = e • d_A^{p-1-CB} *mod p*.

Неважко довести, що: (а) m’ = m; (б) зловмисник не може знати m. Дійсно, $m' = e \cdot d_A^{p-1-CB} \ mod \ p$ = m • $d^c{}_B{}^A$ • d_A^{p-1-CB} mod p = m • $(q^{CB})^{CA}$ •

$(q^cAyp^{-1-c}B$ mod p = m • $q^cBCA+k(p^{-1})^-c_Ac_B$ mod p = m • q^te-i) mod p = m • 1^{CA} = m.

Зловмиснику для знаходження m доведеться розв’язувати задачу дискретного логарифмування, що для великих p неможливо.

Приклад. Нехай m=15. Візьмемо p=23, q=5. Нехай абонент В обрав для себе секретне число cB=13 і обрахував $dB=5^{13}$ mod 23=21. Абонент А вибирає випадкове k=7 і обчислює $r=5^7$ mod 23=17, $e=15*21^7$ mod 23=12. Абонент А надсилає абоненту В пару чисел (17,12). Абонент В знаходить $m'=12*17^{23-1-13}$ mod $23=12*17^9$mod 23=12*7 mod 23 =15.

Зауважимо, що об’єм шифру в 2 рази перевищує об’єм повідомлення, але вимагається лише одна передача даних.

8.5.4 Шифр RSA

Система RSA, названа на честь його розробників Ріверса (Ron Rivers), Шаміра (Adi Shamir) і Адлемана (Leonard Adleman).

Система ґрунтується на використанні іншої односторонньої функції. В цій системі використовуються наступні факти з теорії чисел:

1. Задача перевірки числа на простоту є порівняно простою.
2. Задача розкладання числа n=p*q, де p і q - прості числа, на

множники є дуже складною задачею, якщо ми знаємо тільки n, а p і q - великі числа (задача факторизації).

В групі абонентів A,B,C,... кожний абонент:

1. Вибирає випадково два великих простих числа P і Q і обчислює

N= PQ.

2. Обчислює число f=(P-1)(Q-1).
3. Вибирає деяке число d<f, взаємно просте з f, і за розширеним

алгоритмом Евкліда знаходить таке c, що |cd mod f=1.

В результаті отримуємо таблицю ключів:

Абонент	Секретний ключ	Відкритий ключ
A	PA, QA,cA	NA, dA,
B	PB, QB,CB	NB, dB,
C	PC, QC,CC	Ne, de,

Нехай А хоче передати В повідомлення т< №. Протокол передачі складається з таких кроків:

1. Абонент А шифрує повідомлення, використовуючи відкриті параметри абонента В: ^=*n^uuміРГ^*

2. . Абонент В, отримавши зашифроване повідомлення, обчислює $\underline{m' = e^{Cв} \bmod N_B}$.

Неважко довести, що *m = m.* Дійсно, $m = e^{Cв} \bmod N_B = m^{dBCB} \bmod N$ *в.* Але *deCB* + / +1, а *fB = (Рв — 1)(QB —* 1) = *v(NB~), де ф* — функція Ейлера. Оскільки за теоремою Ейлера $m^{\wedge's:!} = 4 \bmod N$ *в),* то $m = mk_V(NB)+i \bmod Nв = m.$

Відмітимо, що для RSA важливо, щоб кожний абонент вибирав власну пару простих чисел Р і Q, тобто всі N мають різнитись,інакше один абонент міг би читати повідомлення, призначені для іншого. Проте інший відкритий параметр d може бути однаковим у всіх абонентів. Часто рекомендується d=3. Тоді шифрування виконується максимально швидко - всього за дві операції множення.

Приклад. Нехай абонент А передає абоненту В повідомлення m=15. Нехай абонент В обрав для себе секретне число P_B=3, Q_B=11, N_B=33 і dB=3.За допомогою розширеного алгоритму Евкліда знаходимо $c_в$=7. Тоді $e=15^3 \bmod 33=9$. Це число передається абоненту В. Тільки абонент В знає $c_в$, тому він розшифровує $m'=9^7 \bmod 33=15$.

8.5.5 Програмна реалізація алгоритму Діффі-Хеллмана засобами .NET

Платформа .NET надає реалізацію CNG алгоритму Діффі-Хеллмана на еліптичних кривих (ECDH)

через використання об’єктів класу **ECDiffieHellmanCng** з простору імен **System.Security.Cryptography**.

Клас **ECDiffieHellmanCng** дозволяє двом сторонам обмінюватися матеріалом закритих ключів, навіть якщо взаємодія здійснюється по відкритим каналам. Обидві сторони можуть обчислити одне і те ж таємне значення, яке називається **секретною угодою** в керованих класах

Діффі-Хеллмана. Секретна угода може надалі використовуватися для різних цілей, в тому числі як симетричний ключ. Проте, замість прямого представлення секретної угоди клас **ECDiffieHellmanCng** робить деяку її обробку перед наданням значення. Ця постобробка називається **функцією формування ключа** (key derivation function, KDF). Можна вибрати, яку функцію формування ключа використовувати, і задати її параметри через набір властивостей в екземплярі об'єкта Діффі-Хеллмана. Результат передачі секретної угоди через функцію створення ключа представляє собою масив байтів, який можна використовувати як матеріал ключа. Кількість байтів створеного матеріалу ключа залежить від функції формування ключа; наприклад SHA-256 створить 256 бітів матеріалу ключа, а SHA-512 - 512 бітів матеріалу ключа.

Обмін ключами ECDH відбувається наступним чином:

1. Аліса і Боб створюють пару ключів для операції обміну ключами Діффі-Хеллмана.
2. Аліса і Боб налаштовують функцію формування ключа з використанням обумовлених параметрів.
3. Аліса відправляє свій відкритий ключ Бобу.
4. Боб відправляє свій відкритий ключ Алісі.
5. Аліса і Боб використовують відкриті ключі один одного для створення секретної угоди і застосовують функцію формування ключа для створення матеріалу ключа.

Порядок узгодження ключа шифрування між Алісою і Бобом за допомогою об’єктів класу **ECDiffieHellmanCng** такий:

1. Аліса створює сховище ключів і експортує з нього свій публічний ключ для передачі Бобу: CngKey aliceCngKey =

CngKey.Create(CngAlgorithm.ECDiffieHellmanP256);

byte[] alicePublicKeyBlob = aliceCngKey.Export(CngKeyBlobFormat.EccPublicBlob);

2. Боб створює сховище ключів і експортує з нього свій публічний ключ для передачі Алісі:

CngKey bobCngKey = CngKey.Create(CngAlgorithm.ECDiffieHellmanP256);
byte[] bobPublicKeyBlob = aliceCngKey.Export(CngKeyBlobFormat.EccPublicBlob);

3. Аліса імпортує публічний ключ Боба в окреме сховище:

CngKey bobPubCngKey = CngKey.Import(bobPublicKeyBlob, CngKeyBlobFormat.EccPublicBlob);

4. Аліса створює екземпляр класу ECDiffieHellmanCng з ключами, що беруться з її сховища ключів:

ECDiffieHellmanCng aliceAlgorithm = new ECDiffieHellmanCng(aliceCngKey)

5. Аліса отримує секретний ключ з ключового матеріалу:

byte[] aliceKey = aliceAlgorithm.DeriveKeyMaterial(bobPubCngKey);

6. Боб імпортує публічний ключ Аліси в окреме сховище і використовує його для отримання секретного ключа з ключового матеріалу:

CngKey alicePubCngKey = CngKey.Import(alicePublicKeyBlob, CngKeyBlobFormat.EccPublicBlob);
ECDiffieHellmanCng bobAlgorithm = new ECDiffieHellmanCng(bobCngKey); byte[] bobKey = bobAlgorithm.DeriveKeyMaterial(alicePubCngKey);

Слід зауважити, що обмінюватись публічними ключами зручніше, якщо вони представлені у форматі XML.

Експортувати з алгоритму публічний ключ у XML-формат дозволяє метод **ToXmlString():**

ECDiffieHellmanCng bobAlgorithm = new ECDiffieHellmanCng(bobCngKey); string xmlBobPublicKey = bobAlgorithm.PublicKey.ToXmlString();

Імпортувати до алгоритму публічний ключ з XML-формату дозволяє метод **FromXmlString(),** після чого він може бути конвертований в байтовий масив за допомогою методу **ToByteArray():**

ECDiffieHellmanCng bobAlgorithm = new ECDiffieHellmanCng(bobCngKey); bobAlgorithm.FromXmlString(xmlBobPublicKey,ECKeyXmlFormat.Rfc4050); byte[] bobPublicKeyBlob = bobAlgorithm.PublicKey.ToByteArray(); Схематично описаний порядок узгодження показано на Рис.8.6.

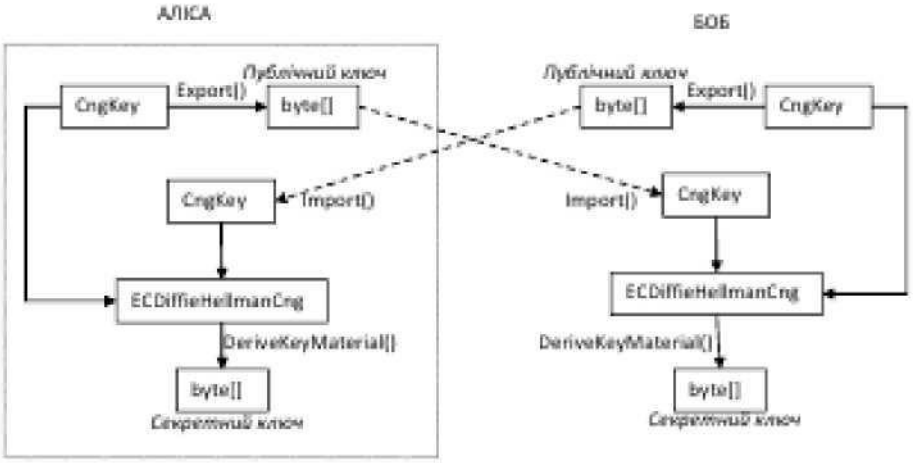


Рис.8.6. Схема реалізації алгоритму Діффі-Хеллмана

8.5.6 Програмна реалізація алгоритму RSA засобами .NET

На платформі .NET алгоритм RSA реалізується за допомогою об'єктів класу **RSACryptoServiceProvider** з простору імен **System.Security.Cryptography**.

Генерація набору, що складається з відкритого та закритого ключів, здійснюється при створенні нового екземпляра класу.

Після створення нового екземпляра класу можна отримати інформацію про ключ одним із двох способів:

1. Метод **ToXMLString** - повертає інформацію про ключ в форматі XML.
2. Метод **ExportParameters** - повертає структуру **RSAParameters**, що містить ключові відомості.

Обидва методи приймають як параметр логічне значення, яке показує:

- **false** - слід повертати відомості тільки про відкритий ключ;
- **true** - слід повертати відомості і про відкритий, і про закритий ключі.

Ініціалізація класу RSACryptoServiceProvider може бути здійснена також двома шляхами:

1. Метод **FromXmlString** - використовує дані ключа з рядка XML.
2. Метод **ImportParameters** - використовує дані структури RSAParameters.

Асиметричні закриті ключі ніколи не повинні зберігатися в роздрукованому вигляді або у вигляді простого тексту на локальному комп'ютері. Якщо необхідно зберігати закритий ключ, слід використовувати для цього *контейнер ключа*.

Контейнер ключа представляє собою екземпляр класу **CspParameters** (з простору імен **System.Security.Cryptography)**. В полі **CspParameters.KeyContainerName** задається ім’я контейнера.

Створити екземпляр класу **CspParameters** і зберегти в ньому згенерований ключ можна так:

```
CspParameters cp = new CspParameters(); cp.KeyContainerName
= “ContainerName”;
RSACryptoServiceProvider rsa = new RSACryptoServiceProvider(cp);
Console.WriteLine("Key added to container: \n {0}", rsa.ToXmlString(true));
```

Передача ключів провайдеру здійснюється через передачу імені контейнера конструктору об’єкта класу **RSACryptoServiceProvider**.

Отримати ключ з контейнеру можна так:

```
CspParameters cp = new CspParameters(); cp.KeyContainerName
= “ContainerName”;
RSACryptoServiceProvider rsa = new RSACryptoServiceProvider(cp);
Console.WriteLine("Key retrieved from container : \n {0}", rsa.ToXmlString(true));
```

Добавлений ключ можна видалити з контейнера. Для цього в екземплярі класу **RSACryptoServiceProvider** необхідно спочатку відмовитись від збереження ключа, присвоївши властивості **PersistKeyInCSP** значення **false**, а потім скористатись методом **Clear**() для вивільнення всіх ресурсів класу і видалення контейнера ключа.

```
CspParameters cp = new CspParameters(); cp.KeyContainerName = ContainerName;
RSACryptoServiceProvider rsa = new RSACryptoServiceProvider(cp);
rsa.PersistKeyInCsp = false; rsa.Clear(); Console.WriteLine("Key deleted.");
```

Порядок розшифрування за допомогою об’єктів класу **RSACryptoServiceProvider** такий:

1. Створюється контейнер для збереження ключів:

```
CspParameters cp = new CspParameters(); cp.KeyContainerName = "Key Name";
```

2. Створюється екземпляр криптопровайдера з розміщенням ключів у контейнері:

```
RSACryptoServiceProvider rsa = new RSACryptoServiceProvider(cp)
```

3. Публічний ключ експортується для передачі іншій стороні:

```
string pubKey = rsa.ToXmlString(false);
Console.WriteLine("Public Key: \n {0}", pubKey);
```

4. Після отримання байтових даних *byte[] EncryptBytes*, зашифрованих за допомогою публічного ключа, здійснюється їх розшифрування за допомогою закритого ключа:

```
byte[] DecryptBytes = rsa.Decrypt(EncryptBytes, false); string decryptStr =
Encoding.Unicode.GetString(DecryptBytes);
//string decryptStr =BitConverter.ToString(DecryptBytes);
Console.WriteLine("Decripted string: \n {0}", decryptStr);
```

Порядок шифрування полягає у такому:

1. Створюється екземпляр криптопровайдера :

```
RSACryptoServiceProvider rsa1 = new RSACryptoServiceProvider()
```

2. Імпортується публічний ключ: rsa1.FromXmlString(pubKey);

3. Текст повідомлення перетворюється у байтову послідовність і зашифровується публічним ключем:

string dataToEncrypt = "Data to encrypt"; byte[] byteToEncript = Encoding.Unicode.GetBytes(dataToEncrypt); byte[] EncryptBytes = rsa1.Encrypt(byteToEncript, false);

4. Зашифрована байтова послідовність відправляється стороні, яка має для розшифрування відповідний закритий ключ.

Описаний процес шифрування за алгоритмом RSA схематично показано на Рис.8.7.

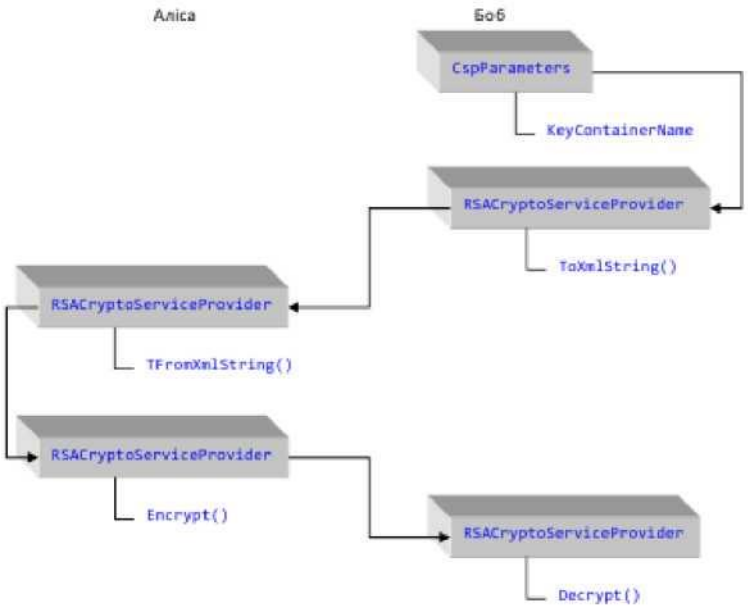


Рис.8.7. Схема реалізації алгоритму RSA

Зауваження. Для перетворення текстових даних в байтові і навпаки можна також скористуватись екземпляром класу UnicodeEncoding з простору імен System.Text. Цей клас надає кодування символів Юнікоду у форматі UTF-16. Наприклад, перетворити текстові дані в байтові можна так:

UnicodeEncoding ByteConverter = new UnicodeEncoding(); byte[] byteToEncript = ByteConverter.GetBytes(dataToEncrypt);

9 Цифровий підпис

9.1. Загальна схема використання

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

ЕЦП накладається за допомогою особистого ключа та перевіряється за допомогою відкритого ключа.

Схема застосування цифрового підпису (Рис.9.1):

1. Беремо вихідне повідомлення і створюємо 160-бітовий геш (дайджест повідомлення) за допомогою алгоритму SHA-l.
2. Потім дайджест повідомлення шифрується за допомогою секретного ключа, відомого тільки його власнику, тобто відправнику повідомлення.
3. Будь-який бажаючий може розшифрувати геш, користуючись загальнодоступним відкритим ключем.
4. Результат цього шифрування і називають цифровим підписом.
5. Підписане повідомлення формується об'єднанням вихідного повідомлення, його цифрового підпису та відкритого ключа.

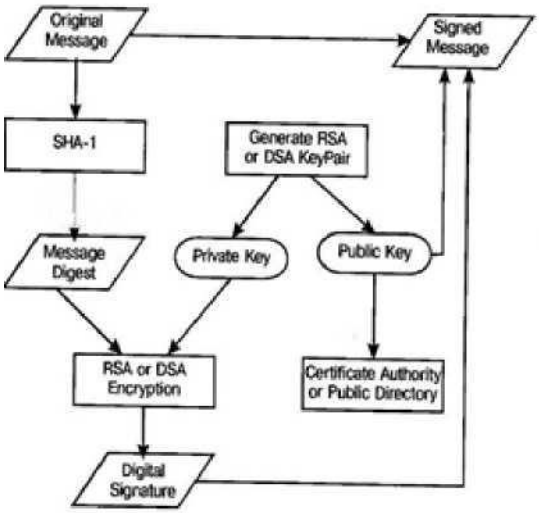


Рис.9.1. Схема створення цифрового підпису

Схема перевірки цифрового підпису (рис.9.2):

6. Отримане повідомлення розбивається на три компоненти: на вихідне повідомлення, відкритий ключ і цифровий підпис.
7. Заново обчислюється геш повідомлення.
8. Якщо обчислений геш збігається з розшифрованим гешем, то перевірка підпису пройдена.

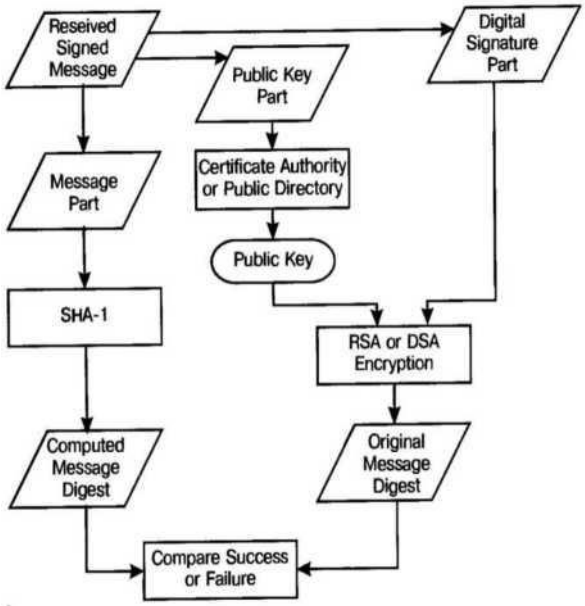


Рис.9.2. Схема перевірки цифрового підпису

9.2. Цифровий підпис на основі шифру RSA

Створити цифровий підпис можна на основі схеми RSA. Так, якщо Аліса має намір використовувати цифровий підпис, вона:

1. Вибирає два великих простих числа P і Q.
2. Обчислює N=PQ і f=(P-1)(Q-1).
3. Вибирає d, взаємно просте з f.
4. Обчислює $c=d^{-1} \bmod f$.
5. Зберігає c в якості секретного ключа (інші числа - P,Q і f більше не потрібні).
6. Публікує N і d в якості відкритого ключа.

Тепер Аліса готова до використання цифрового підпису. Нехай вона хоче підписати повідомлення m= m1, ... , mn. Тоді вона:

1. Спочатку обчислює так називану геш-функцію h =h(m1, ... , mn).

Найбільш важлива особливість цієї функції в тому, що практично неможливо змінити текст m1, . , mn , не змінивши h. Вважається, що

алгоритм її обчислення загальновідомий.

2. Обчислює число $s= hc \bmod N$, яке і є цифровим підписом.
3. Додає s до повідомлення m, формуючи підписане повідомлення <m,s> .

Тепер кожний, хто знає відкритий ключ Аліси, може перевірити її справжність. Для цього йому необхідно:

1. Обрахувати геш-функцію h =h(m1, . , mn).
2. Обрахувати $w=s^d \bmod N$..
3. Перевірити виконання рівності w = h.

Неважко переконатись у тому, що у випадку справжнього підпису w = h.

Дійсно, $w=s^d \bmod N= h^{cd} \bmod N= h$.

Приклад. Нехай P=5 і Q=11. Тоді N=5*11=55, f=4*10=40. Візьмемо d=3, т.я. gcd(40,3)=1. За допомогою розширеного алгоритму Евкліда знаходимо $c=3^{-1} \bmod 40 =27$.

Нехай значення геш-функції повідомлення y=13. Тоді Аліса знаходить $s = 13^{27} \bmod 55=7$.

Для перевірки підпису обраховуємо $w=7^3 \bmod 55=13$. Оскільки це значення співпадає зі значенням геш-функції, підпис справжній.

Цифровий RSA-підпис має два суттєвих недоліки:

1. Він є дуже великим, а іноді необхідно, щоб підпис займав мало місця.
2. Генерація RSA-підпису - надто дорога операція, що унеможливлює його використання в окремих додатках.

9.3. Цифровий підпис на основі шифру Ель-Гамаля

У той час, як цифровий підпис RSA ґрунтується на складності розв'язання задачі факторизації, цей підпис ґрунтується на складності розв'язання задачі дискретного логарифмування.

Нехай Аліса має намір використовувати цифровий підпис. Для цього вона:

1. Вибирає велике просте число p.
2. Вибирає таке число q, що різні степені q є різними по модулю p.
3. Публікує p і q у відкритому виді.
4. Вибирає випадкове число x, 1<x<p-1, і зберігає його в якості секретного ключа.
5. Обчислює число $y=qx \bmod p$.
6. Публікує y в якості відкритого ключа.

Тепер Аліса може користуватись цифровим підписом. Нехай вона хоче підписати повідомлення т= ті, ... , $т_п$. Тоді вона:

1. Обчислює значення геш-функції h=h(m), яке має задовольняти нерівності 1<h<p.
2. Вибирає випадкове число k (1<k<p-1), взаємно просте з (p-1), і обчислює число $r=q^k \bmod p$.
3. Обчислює числа $u=(h-xr) \bmod (p-1)$, $s=k^{-1}u \bmod (p-1)$. Тут під k^{-1} розуміється число, що задовольняє рівняння $k^{-1}k \bmod (p-1)$ і знаходиться за розширеним алгоритмом Евкліда.
4. Формує підписане повідомлення <m, r, s>.

Тепер кожний, хто знає відкритий ключ Аліси, може перевірити її справжність. Для цього йому необхідно:

1. Обрахувати геш-функцію h=h(m).
2. Перевірити виконання рівності $y_r r^s=q_h \bmod p$

Неважко переконатись у тому, що у випадку виконання рівності $y^r r^s=q^h \bmod p$ підпис є справжнім.

ДІЙСНО, *yrTs = (Qx)r(Qk)s = QxrQk(k-1(h-xr)) = QxrQhQ—xr = Qh* W.Orf *fl.*

Приклад. Нехай загальними параметрами для деякої спільноти вибрані p=23 і q=5. Аліса вибирає свій секретний ключ x=7 і обчислює $y=5^7 \bmod 23=17$.

Нехай значення хеш-функції повідомлення h(m)=3. Тоді Аліса генерує випадкове число k, наприклад, k = 5. Тоді $r=5^5 \bmod 23=20$, $u=(3-7*20) \bmod 22=17$.

Далі Аліса знаходить $k^{-1} \bmod 22 = 5^{-1} \bmod 22 = 9$ і $s = 9*17 \bmod 22 = 21$, а потім формує повідомлення<т,20,21>

Для перевірки підпису Боб знаходить значення хеш-функції і обраховуємо $y^r r^s = 17^{20} *20^{21} \bmod 23 =16*15 \bmod 23 =10$ і після цього $q^h \bmod p = 5^3 \bmod 23 =10$. Оскільки ці значення співпадають, підпис справжній.

Розглянутий метод складніше RSA, але на його основі може бути побудований більш ефективний алгоритм, в якому час обрахунків значно скорочується за рахунок використання «коротких» показників степенів. Він розглядається далі.

9.4. Стандарт DSA

В DSA використовуються наступні параметри домена:

1. p - просте число p, де $2L^{-1} < p < 2L$, 512 =< L =< 1024 и L кратно 64;
2. q - простий дільник p-1, при цьому 2159 < q < 2160;
3. g = h(p-1)/q mod p, де h - будь-яке ціле число 1 < h < p - 1 таке, що h(p- 1)/q mod p > 1;
4. x - випадкове ціле число, 0 < x < q; 5. y = gx mod p;
6. k - випадкове ціле число, 0 < k < q.

Генерація підпису виконується наступним способом:

1.r = ($g^k \bmod p$) mod q

2.s = (k^{-1}(SHA(M) + xr)) mod q.

3.Якщо r = 0 або s = 0, має бути згенеровано нове k і обчислений новий підпис.

Тут SHA (M) - 160-бітний бінарний рядок, що отримується в наслідок застосування геширування за алгоритмом SHA-1.

Перевірка підпису:

Числа p, q, g і відкритий ключ знаходяться у відкритому доступі. Нехай M ', r' і s 'отримані версії M, r і s, відповідно, і нехай y - відкритий ключ. При перевірці підпису спочатку потрібно подивитися, чи виконуються наступні нерівності: 0 <r '<q, 0 <s '<q. Якщо хоча б одне нерівність не виконано, підпис повинна бути відкинута. Якщо умови нерівностей виконані, виробляються наступні обчислення:

1. $w = (s')^{-1} \bmod q$
2. $u1 = ((SHA (M ')\ w) \bmod q$
3. $u2 = ((r')\ w) \bmod q$
4. $v = (((g)^{u1}\ (y)^{u2}) \bmod p) \bmod q$.

Якщо v = r ', то справжність підпису підтверджена.

Приклад. Виберем такі параметри домену: q=13, p=4q+1=53 і $g=2^4 \bmod 53=16$. Нехай ключова пара користувача приймає значення x=3, $y=16^3 \bmod 53 =15$.

Будемо вважати, що геш-функція повідомлення SHA(M)=5. В якості ефемерного ключа виберемо k=2.

Генеруємо підпис: $r = (16^2 \bmod 53) \bmod 13 =5$, $s = ((5 + 3*5)/2) \bmod 13=10$.

Перевірка підпису: $w = (10)^{-1} \bmod 13 = 4$, $u1 = (5*4) \bmod 13=7$, $u2 = (5*4) \bmod 13=7$, $v = (16^7*15^7) \bmod 53) \bmod 13=5$.

9.5. Стандарт ДСТУ Р34.10-94

Параметри домена теж схожі на параметри домена в алгоритмі Ель-Гамаля:

1. Вибираються два простих числа - p довжиною 1024 біт і q довжиною 256 біт, між якими виконується співвідношення $p=bq+1$ для деякого цілого числа b.
2. Вибирається таке число a>1, що $|a^q \bmod p=1$
3. Публікуються у відкритому виді p, q і a.
4. Кожний користувач вибирає випадкове число x, 1<x<д , і зберігає його в якості секретного ключа.
5. Обчислює число $y=ax \bmod p$ і публікує його в якості відкритого ключа.

Генерація підпису виконується наступним способом:

1. Обчислюється значення геш-функції h=h(m), яке має задовольняти нерівності 1<h<p.
2. Вибирається випадкове число k (1<k< q).
3. Обчислюється число $|r=(a^k \bmod p) \bmod q;$ якщо r=0, повертаємося до кроку 2.
4. Обчислюється число $s=(kh+xr) \bmod q$; якщо s =0, повертаємося до кроку 2.
5. Формується підписане повідомлення <m, r, s>.

Перевірка підпису виконується наступним способом:

1. Обчислюється геш-функцію h=h(m).
2. Перевіряється виконання нерівності |0<r<q, |0<s<q.
3. Обчислюється $|ui=sh^{-1} \bmod q|$, $U2=-r\ h^{-1} \bmod q$
4. Обчислюється $|y = (a^u 1 y^u 2 \bmod p) \bmod q$

10. Крипто провайдери .Net

Концепція *CryptoAPI* має на увазі приховування від програміста всіх тонкощів процесу шифрування даних.У CryptoAPI можна виділити п'ять основних функціональних областей. Функції, що входять в кожну з областей, як правило, містять в імені відповідні ключові слова:

- базові криптографічні функції (Crypt). У цю групу входять функції для підключення до кріптопровайдеру, генерації і зберігання ключів, обміну ключами, управління параметрами ключа;
- функції кодування / декодування (Crypt) забезпечують доступ до алгоритмів шифрування / дешифрування і хеширования;
- функції роботи з сховищем сертифікатів (Store) призначені для управління наборами цифрових сертифікатів, використовуваних при цифрового підпису та шифрування з відкритим ключем;
- спрощені функції для роботи з повідомленнями (Message) дозволяють шифрувати / дешифрувати повідомлення і блоки даних, підписувати їх і перевіряти цифровий підпис;
- низькорівневі функції для роботи з повідомленнями (Msg) дозволяють більш гнучко виконати ту ж роботу, що і спрощені функції, але вимагають для цього великих зусиль з боку програміста. Для використання функцій CryptoAPI в додатках, написаних на C / C ++, необхідно використовувати заголовки *WinCrypt.h* і підключити до додатка бібліотеку імпорту *Crypt32.lib*.

10.1. Криптопровайдери CryptoAPI

Одним з ключових понять *CryptoAPI* є криптографічний провайдер (*CSP, Cryptographic Service Provider*), який реалізує базовий набір криптографічних функцій, що відповідають тому чи іншому криптографічному алгоритму (або сімейству алгоритмів). Криптопровайдери реалізовані у вигляді окремих *DLL*, так, що сторонні розробники можуть самостійно включати до складу *CryptoAPI* реалізації своїх алгоритмів (хоча для поширення криптопровайдера потрібно підписати його цифровим підписом в *Microsoft*). Одночасно в операційній системі можна встановити кілька CSP.Вибір конкретного алгоритму задається параметрами функції і залежить від використовуваного CSP, а універсальний набір функцій визначено для кожного типу криптографічних операцій.

CryptoAPI надає наступні стандартні криптопровайдери:

- *Microsoft Base Cryptographic Provider*. Визначення: *MS_DEF_PROV*. Бібліотека: *rsabase.dll;*
- *Microsoft Strong Cryptographic Provider* (поставляється починаючи з *Windows 2000*, є розширенням базового). Визначення: *MS_STRONG_PROV*. Библиотека: *rsaenh.dll;*
- *Microsoft Enhanced Cryptographic Provider* (в порівнянні з базовим забезпечує роботу з ключами більшої довжини). Визначення: *MS_ENHANCED_PROV*. Библиотека: *rsaenh.dll;*
- *Microsoft AES Cryptographic Provider* (підтримка *AES*).
- *Microsoft DSS Cryptographic Provider* (реалізує алгоритми *SHA* і *DSS*). Визначення: *MS_DEF_DSS_PROV*. Бібліотека: *dssbase.dll;*
- *Microsoft Base DSS and Diffie-Hellman Cryptographic Provider* (додана підтримка алгоритма обміну ключей Диффі-Хеллмана). Визначення: *MS_DEF_DSS_DH_PROV*. Бібліотека: *dssbase.dll;*
- *Microsoft DSS and Diffie-Hellman/Schannel Cryptographic Provider* (додана аутентифікація клиентів по захищеному протоколу взаємодії). Визначення: *MS_DEF_DH_SCHANNEL_PROV*. Бібліотека: *dssenh.dll;*
- *Microsoft RSA/Schannel Cryptographic Provider* (реалізація алгоритмів *RSA*).

10.2. Типи криптопровайдерів

Всі *CSP* відрізняються один від одного своїми типами, які визначаються набором

параметрів, що включає:

- алгоритм обміну сесійним (симетричним) ключем;
- алгоритм обчислення цифрового підпису;
- формат цифрового підпису;
- схема генерування сесійного ключа по хешу;
- довжина ключа.

У файлі *WinCrypt.h* дано визначення типів криптопровайдерів, визначених у даний час. Найбільш відомі типи *CSP* представлені в таблиці 1.

Таблиця 10.1

Тип криптопровайдера	Підтримувані алгоритми
PROV_RSA_FULL(1)	ключовий обмін – RSA; цифровий підпис – RSA; шифрування – RC2, RC4; хешування – MD5, SHA.
PROV_RSA_SIG (2)	ключовий обмін – не підтримується; цифровий підпис – RSA; шифрування – не підтримується; хешування – MD5. SHA.
PROV_RSA_SCHANNEL(12)	ключовий обмін – RSA; цифровий підпис – RSA; шифрування – RC4. DES, Triple DES (не обов'язково всі); хешування – MD5, SHA.
PROV_DSS (3) PROV_RSA_SIG	ключовий обмін – не підтримується; цифровий підпис – DSS(DSA); шифрування – не підтримується; хешування – MD5, SHA.
PROV_DSS_DH(13)	ключовий обмін – DH; цифровий підпис – DSS; шифрування – CYLINK MEK; хешування – MD5, SHA.
PROV_DH_ SCHANNEL (18)	ключовий обмін – DH (Ephemeral); цифровий підпис – DSS; шифрування – DES, Triple DES; хешування – MD5, SHA.
PROV_FORTEZZA (4)	ключовий обмін – KEA; цифровий підпис – DSS; шифрування– Skipjack; хешування– SHA.
PROV_MS_EXCHANGE (5)	ключовий обмін – RSA; цифровий підпис – RSA; шифрування – CAST; хешування – MD5.
PROV_SSL (6)	ключовий обмін – RSA; цифровий підпис – RSA; шифрування – різні алгоритми; хешування – різні алгоритми. шифрування – DES, Triple DES; хешування – MD5, SHA.

Функції роботи с криптопровайдерами можна розділити на наступні групи:

- функції ініціалізації контексту і отримання параметрів криптопровайдера;
- функції генерації ключів і роботи з ними;
- функції шифрування / розшифрування даних;
- функції хешування і отримання цифрового підпису даних.

Кожен CSP має свою базу ключів (*key database*), в якій знаходиться один або кілька контейнерів (*key containers*), де зберігаються пари закритих / відкритих ключів. Контейнери бувають двох типів - призначені для користувача (цей тип використовується за умовчанням) і машинні (*CRYPT_MACHINE_KEYSET*). Тип контейнера задається прапором при отриманні контексту провайдера. На комп'ютері зазвичай встановлено кілька провайдерів і відповідних типів криптопровайдерів. Для того, щоб переглянути всі встановлені криптопровайдери і типи треба застосувати відповідні функції *CryptEnumProviders* і *CryptEnumProviderTypes*.

Приклад перегляду встановлених криптопровайдерів на комп'ютері:

```
#include <stdio.h>
```

```
#include <windows.h>
#include <wincrypt.h>
#include <stdlib.h>
#include <wtypes.h> #include <iostream> #include <fstream> #pragma
comment(lib,"crypt32.lib") using namespace std;
void view_installed_providers() { HCRYPTPROV hProv = NULL; LPTSTR pszName =
NULL;
DWORD dwType;
DWORD cbName;
DWORD dwIndex = 0;
// Цикл по типам провайдерів, що перераховуються.
dwIndex = 0;
while(CryptEnumProviderTypes(
dwIndex,
NULL,
0,
&dwType,
NULL,
&cbName
))
{
//Розподіл пам’яти в буфері для відновлення цього імені. pszName = (LPTSTR)
malloc(cbName);
if(!pszName)
puts("ERROR - malloc failed!");

memset(pszName, 0, cbName);
// Отримання імені типу провайдера.
if(CryptEnumProviderTypes(
dwIndex++,
NULL,
0,
&dwType,
pszName,
&cbName))
{
printf (" %4.0d ",dwType);
wcout<<pszName<<endl;
}
else
{ puts("ERROR - CryptEnumProviders");
}
}
// Друк заголовка перерахунка провайдерів.
printf("\n\n Listing Available Providers.\n");
printf("Provider type Provider Name\n");
printf(" \n");
// Цикл по провайдерам, що перераховуються.
dwIndex = 0;
while(CryptEnumProviders(
dwIndex,
NULL,
0,
&dwType,
```

```
NULL,
&cbName
))
{
//Розподіл пам’яти в буфері для відновлення цього імені.
pszName = (LPTSTR)malloc(cbName);
if(!pszName) puts("ERROR - malloc failed!");
memset(pszName, 0, cbName);
// Отримання імені провайдера. if(CryptEnumProviders( dwIndex++, NULL, 0, &dwType,
(LPWSTR)pszName, &cbName // pcbProvName -- длина pszName ))
{ printf (" %4.0d ",dwType);
wcout<<pszName<<endl; } else { puts("ERROR - CryptEnumProviders");
}
} // Кінець цикла while
printf("\nProvider types and provider names have been listed.\n");
}
```

При роботі з криптопровайдерами додатки мають ряд обмежень:

Додаток не може безпосередньо скористатися реалізаціями функцій з криптопровайдера, і все взаємодія проходить через базові криптографічні функції. В ідеальному випадку можлива зміна одного криптопровайдера на іншого без модифікації використовує його застосування, хоча на практиці деякі додатки можуть зажадати функціональності певного криптопровайдера.

- Додаток не має прямого доступу до ключів, які використовуються функціями криптопровайдера.
- Додатки не можуть впливати на виконання криптографічних операцій інакше як завданням алгоритму і вибором ключа.
- Додатки не відповідають за аутентифікацію користувачів, всю цю роботу виконує криптопровайдер.
- Для отримання доступу до кріптопровайдеру використовується функція *CryptAcquireContext,* по завершенні роботи необхідно викликати функцію *CryptReleaseContext*.

CryptAcquireContext виконує підключення до кріптопровайдеру з заданим типом і ім'ям і повертає його дескриптор (контекст). Таким чином, сеанс роботи починається з ініціалізації (отримання контексту) за допомогою функції *CryptAcquireContex*t. Як параметри ця функція приймає ім'я контейнера ключів, ім'я криптопровайдера, тип провайдера і прапори, що визначають тип і дії з контейнером ключів і режим роботи криптопровайдера.

BOOL WINAPI CryptAcquireContext(
HCRYPTPROV phProv*,
LPCTSTR pszContainer, // им’я контейнера ключів або *NULL* - за замовчуванням
LPCTSTR pszProvider, // вказують им’я одного або кілька имен криптопровайдерів
DWORD dwProvType, // тип криптопровайдера
DWORD dwFlags);// 0 або *CRYPT_NEWKEYSET*

Розглянемо приклад підключення до кріптопровайдеру:

```
//вказуємо им’я і тип криптопровайдера і відчиняємо або ствоюємо контейнер для
ключів
HCRYPTPROV hProv;
if(!CryptAcquireContextW(
&hProv,NULL,MS_ENHANCED_PROV,PROV_RSA_FULL,0) &&
(!CryptAcquireContextW(
&hProv,NULL,MS_ENHANCED_PROV,PROV_RSA_FULL,CRYPT_NEWKEYSET)))
{ puts("Не вдається отримати контекст\n");
return -1;
}
```

Можливий інший варіант, коли криптопровайдер вибирається за замовчуванням, а тип *PROV_DH_SCHANNEL*:

```
HCRYPTPROV hProv;
//намагаємося відкрити існуючий keyset або створити новий
if(!CryptAcquireContext(&hProv, NULL,
NULL,PROV_DH_SCHANNEL,0) &&
(!CryptAcquireContext(&hProv,NULL,NULL,PROV_DH_SCHANNEL,CRYPT
_NEWKEYSET))
{
puts("Не вдається створити контекст\n"); return -1;
                                                    }
```

Функції *CryptSetProvParam* и *CryptGetProvParam* дозволяют відповідно задати та дізнатися ряд характеристик криптопровайдера.

10.3. Робота з сертифікатами X509

Алгоритми шифрування з відкритим ключем і цифрові підписи тісно пов'язані з поняттям сертифіката. Цифровий сертифікат видається якоюсь авторизованої організацією (*Certification Authority, CA*) і крім пари публічний / закритий ключ містить інформацію, що дозволяє ідентифікувати його власника. Крім цього, сертифікат має термін дії і підписується за допомогою іншого ключа *CA*, який використовується для забезпечення гарантії достовірності цих атрибутів і, що найбільш важливо, самого відкритого ключа. *CryptoAPI* підтримує сертифікати, відповідні специфікації *X.509* (поточна версія 3), описані в документі *RFC 3280*. Стандарт *X.509* не визначає обов'язкового типу ключа, вбудованого в сертифікат, але в даний час алгоритм *RSA* є найбільш відомим з застосовуваних криптографічних алгоритмів з асиметричними шифрами.Існує можливість створювати власні сертифікати. Метод їх створення зазвичай залежить від способу їх застосування. Для звичайних ситуацій в Інтернеті, коли користувач не знає, з ким він зв'язується, запитується, як правило, сертифікат від комерційного центру сертифікації. Перевагою такого підходу є те, що ці відомі центри сертифікації вже визнані довіреними операційною системою Windows і всіма іншими операційними системами (і оглядачами), що підтримують сертифікати і протокол *SSL*. Це дозволяє обходитися без обміну ключами центру сертифікації.Можна створювати сертифікати за допомогою програми *OpenSSL*. *OpenSSL* підтримує два основні формати, які визначають кодування криптографічних документів: *DER* - бінарні файли і *PEM* - текстові файли.

Криптографічний документ в форматі *PEM* має вигляд:

BEGIN <ТИП ДОКУМЕНТА>
<Документ в форматі DER, закодований в base64>-
END <ТИП ДОКУМЕНТА>

Тобто зміст документа оточується заголовками спеціального виду, що складаються з п'яти знаків «-», слів *begin* і *end* і вказівок типу документа (*CERTIFICATE, CRL* і т.д.) Завдяки наявності таких заголовків в одному файлі може міститися кілька документів в форматі *PEM*: наприклад, ланцюжок сертифікатів або сертифікат і його закритий ключ. За замовчуванням в *OpenSSL* використовується саме формат *PEM*. Файли, що містять ланцюжка сертифікатів, сертифікати та асоційовані з ними ключі, можуть представлятися в декількох форматах, що визначають склад і будова файлів, що містять ключову інформацію; сама ця інформація кодується в вищеописаних форматах *PEM* і *DER*.

Найбільш поширені з форматів, що визначають структуру файлів, що містять криптографічний інформацію - PKCS # 7 і PKCS # 12. PKCS # 7 - формат захищених повідомлень, який, крім самого повідомлення, може містити необхідні для роботи з ним сертифікати і CRL. Багато з центрів, що засвідчують, поширюють свої сертифікати і CRL у вигляді PKCS # 7 документів, в яких немає повідомлення, є тільки службова інформація. PKCS # 7 - документ може бути закодований і в форматі DER, і в форматі PEM. Розширення файлів сертифікатів можуть бути .cer або .crt. PKCS # 12 - формат для перенесення сертифіката і пов'язаного з ним закритого ключа з машини на машину або для

резервного копіювання. У цьому форматі можуть також міститися сертифікати центру, що засвідчує. Файли формату PKCS # 12 завжди кодуються в форматі DER. Ці файли вимагають особливо пильної уваги, оскільки містять закритий ключ; при необережному поводженні з таким файлом закритий ключ легко скомпрометувати. Як правило, ці файли захищені на паролі. Розширення файлів формату PKCS # 12 або .p12, або .pfx. У Windows зручно працювати з сертифікатом даного формату. При виконанні експорту пари ключів Windows пропонує зашифрувати PFX-файл за допомогою пароля; при імпорті пари ключів користувач повинен знову надати цей пароль.

10.4. Сховище сертифікатів Windows

Сертифікати та відповідні їм закриті ключі можна зберігати на різних пристроях, наприклад жорстких дисках, смарт-картах і в ключах для порту *USB*. У *Windows* передбачений рівень абстракції, званий сховищем сертифікатів, який служить для забезпечення єдиного способу доступу до сертифікатів незалежно від місця їх зберігання. До тих пір поки у апаратного пристрою є підтримуваний *Windows* - постачальник служби криптографії (*CSP*) - можна отримувати доступ до даних, що зберігаються на цьому пристрої, використовуючи інтерфейс *API* сховища сертифікатів. Сховище сертифікатів глибоко заховане в профілі користувача. Це дозволяє використовувати списки управління доступом (*ACL*) для ключів конкретного облікового запису. Кожне сховище розділено на контейнери:

Контейнер з назвою *Personal* (особистий), призначений для зберігання призначеного для користувача сертифіката (тобто, у яких є відповідний закритий ключ).

У контейнері *Trusted Root Certification Authorities* (довірені кореневі центри сертифікації) зберігаються сертифікати всіх центрів сертифікації, яким довіряє користувач.

У контейнері *Other People* (інші) містяться сертифікати колег, з якими є безпечний зв'язок. І так далі.

Найпростіший спосіб отримати доступ до свого сховища сертифікатів полягає у виконанні команди *certmgr.msc*. Існує також сховище на рівні комп'ютера, що використовується обліковими записами комп'ютера з ОС *Windows* (мережі, локальної служби і локальної системи) або служить для спільного використання сертифікатів і ключів різними обліковими записами.

Будь-який сертифікат *X.509* одержуваний користувачем поштою або за запитом з центру сертифікації необхідно помістити в сховище сертифікатів в *WINDOWS*. Для цього виконують подвійне клацання миші по файлу сертифіката і потім в діалоговому вікні вибир

10.5. Робота з сертифікатом X.509 за допомогою CryptoAPI

Алгоритм роботи з сертифікатом складається з виклика наступних функцій:

- *CertOpenStore* - відкриваєм сховище сертифікатів.
- *CertOpenSystemStore* відкриваєм сховище системних сертифікатів.
- *CertFindCertificatelnStore* - пошук потрібного сертифікату.
- *CertEnumCertificatesInStore*- перебирає всі сертифікати в сховищі.
- *CertGetNameString*- витягує им'я поточного сертифікату.

Приклад роботи з сертифікатом:

```
// Сертифікат, наведений в форматі PKSC#12
#include <stdio.h>
#include <windows.h>
#include <stdlib.h>
#include <wtypes.h>
#include <wincrypt.h>
#define MY_TYPE (PKCS_7_ASN_ENCODING | X509_ASN_ENCODING)
//Назва персонального сховища
#define CERT_STORE_NAME L"MY"
//Назва сертифікату, встановленого в це сховище
#define SIGNER_NAME L"EVGENY"
```

```
//Відкриваємо сховище сертифікатів HCERTSTORE hStore;
if ( !( hStore = CertOpenStore( CERT_STORE_PROV_SYSTEM, 0,
NULL,
CERT_SYSTEM_STORE_CURRENT_USER, CERT_STORE_NAME)))
{ puts("Неможна відкрити сховище MY ");
}
//Отримуємо покажчик на наш сертифікат PCCERT_CONTEXT pSignerCert=0;
if(pSignerCert = CertFindCertificateInStore( hStoreHandle, MY_TYPE, 0,
CERT_FIND_SUBJECT_STR, SIGNER_NAME, NULL))
{ printf("Серификат знайдений\п");
}
else
{
printf("Сертификат не знайдений.");
//За допомогою даної функції виведемо ім'я CSP та ім'я
контейнера ключів
DWORD dwUserNameLen = 100;
CHAR szUserName[100];
if(CryptGetProvParam(
hProv, // Дескриптор на CSP
PP_NAME, // параметр для отримання імені CSP
(BYTE *)szUserName, // Покажчик на буфер, що містить ім'я CSP
&dwUserNameLen, // длина буферу
0))
{ printf("Name CSP: %s\n",szUserName);
} else
{
puts("noMKHKa CryptGetProvParam.\n");
}
if(CryptGetProvParam(
hProv, // Дескриптор на CSP
PP_CONTAINER, // параметр для отримання імені key
container
(BYTE *)szUserName, // Покажчик на буфер, що містить ім'я key container
&dwUserNameLen, // длина буферу
0))
{ printf("Name key container: %s\n",szUserName);
} else {
puts("ErrorCryptGetProvParam.\n"); }
```

Приклад, що дозволяє проглянути всі сертифікати в сховищі.

```
//Відкриваємо сховище сертифікатів if(!(hStore=CertOpenSystemStore(NULL,"
EVGENY "))) {cout<<"Error opening store\n";cin.get();return 1;} char* pszName=new char[128];
int i=0;int certNum=0;
//перебираємо і виводимо всі імена сертифікатів
while(hContext=CertEnumCertificatesInStore(hStore,hContext)) {
if(!CertGetNameString(hContext,CERT_NAME_SIMPLE_DISPLAY_TYPE,
0,0,pszName,128))
{cout<<"Error getting name\n";cin.get();return 1;} i++;
cout<<i<<" "<<pszName<<'\n';
} cout<<"Enter number of certificate\n";
cin>>certNum;hContext=NULL;
//отримуємо зі сховища вибраний сертифікат for(i=0;i<certNum;i++) {
hContext=CertEnumCertificatesInStore(hStore,hContext);
```

```
}
delete [] pszName;
//перевіряємо як отримали сертифікат
if(hContext==NULL){cout<<"Error getting certificate\n"; cin.get();return 1;}
```

*Всі ключі в CRYPTOAPI управляються і використовуються за допомогою якихось ідентифікаторів, і додаток не дістає відкритого доступу до н*их. При підключенні до криптопровайдеру він може надати доступ до стандартного сховища ключів, або скористатися сховищем, створеним по запиту додатку.

Отримавши покажчик на сертифікат X.509, тепер можна витягувати відкритий ключ за допомогою функції *CryptImportPublicKeyInfo*. Надалі, отримавши ідентифікатор відкритого ключа можна використовувати його для обміну ключами (експорту сесійного ключа) або для перевірки електронно- цифрового підпису.

Приклад імпорту відкритого ключа з сертифікату:

```
// Імпортуємо public key для подальшої верифікації підпису або шифрування сеансового
                                        ключа
HCRYPTKEY hPublicKey;
if(CryptImportPublicKeyInfo(
hProv,
MY_TYPE, &(pSignerCert->pCertInfo->SubjectPublicKeyInfo), &hPublicKey)) {
printf("Import public key.\n");
}
else
{ printf ("Помилка CryptAcquireContext.");
}
```

Закритий ключ, відповідний сертифікату відкритого ключа, зберігається в спеціальному контейнері ключів. Для роботи з даним ключем необхідно взятися до роботи з CSP і контейнером нашого сертифікату через функцію *CryptAcquireCertificatePrivateKey*. А потім за допомогою функції *CryptGetUserKey* набути значень закритого ключа з вказаного контейнера ключів. Даний ключ може використовуватися для обміну ключами (імпорту сесійного ключа) або для створення цифрового підпису.

Розглянемо приклад витягання секретного ключа, відповідного сертифікату відкритого ключа PKSC#12:

```
HCRYPTKEY hPrivateKey=0;
DWORD keySpec=0;
// Витягуємо з сертифікату контекст приватного ключа
if(!CryptAcquireCertificatePrivateKey(
pSignerCert, 0, NULL, &hProv, &keySpec, NULL))
{ cout<<"Error getting private context\n"; getchar(); return -1;
}
// Витягуємо закритий ключ
if(!CryptGetUserKey(hProv,keySpec,&hPrivateKey)) {cout«"Error getting private key\n";
getchar ();
return -1;
}
```

10.6. Робота з ключами шифрування за допомогою CRYPTOAPI

CRYPTOAPI дозволяє шифрувати дані, розміщені в оперативній пам'яті у вигляді послідовності байтів. Для шифрування можуть бути використані як симетричні, так і асиметричні алгоритми. *CRYPTOAPI* надає методи для роботи з сесійними (симетричними) ключами і з відкритими ключами. Асиметричні алгоритми використовуються для обміну сеансовими ключами. Зашифрувавши повідомлення електронної пошти симетричним алгоритмом, ви додаєте до зашифрованого повідомлення сам сеансовий ключ, зашифрований на відкритому ключі одержувача з використанням асиметричного алгоритму. Це дозволяє

вам знищити сеансовий ключ відразу ж після шифрування повідомлення. Навіть якщо ваш комп'ютер піддасться злому, зловмисник все одно не дізнається зміст повідомлення.

У *CRYPTOAPI* генерація ключів може виконуватися випадковим чином за допомогою функції *CryptGenKey* або на основі деяких даних, наприклад, призначеного для користувача пароля за допомогою функції *CryptDeriveKey*.

Після закінчення роботи з ключем він знищується функцією *CryptDestroyKey*.

У *CRYPTOAPI* алгоритм шифрування повідомлення указується при генерації сесійного ключа в *CryptGenKey*:

BOOL

CryptGenKey (

HCRYPTPROV hProv, ALG_ID Algid, // ідентифікатор алгоритму шифрування, для якого генерується ключ (наприклад, *CALG_3DES*). Для ключової пари *RSA* для шифрування і підпису використовуються *AT_KEYEXCHANGE і AT_SIGNATURE*.

DWORD dwFlags, // задає різні опції ключа, які залежать від алгоритму і провайдера. *CRYPT_EXPORTABLE*- для експорту сесійного ключа. *CRYPT_ENCRYPT* - для шифрування.

HCRYPTKEY* //ідентифікатор ключа.

phKey);

Приклад генерації випадкового сесійного ключа:

HCRYPTKEY hKey;

//генеруємо сесійний ключ для DES if(!CryptGenKey(hProv, CALG_DES, CRYPT_EXPORTABLE|CRYPT_ENCRYPT, &hKey)) { puts("Не вдається створити ключ DES\n"); return -1;

}

10.7. Параметри ключа

Після створення ключа його параметрами можна маніпулювати за допомогою функцій *CryptGetKeyParam()* і *CryptSetKeyParam()*. Ці функції надають доступ до таких атрибутів, як алгоритм створення ключа, довжина шифрованого блоку, що ініціює значення, початковий вектор, режим

буферизації і режим шифрування, а також допуски, які задають операції,

допустимі для даного ключа.

Алгоритми симетричного шифрування мають наступні режими шифрування повідомлень:

CRYPT_MODE_ECB	-	режим простій заміни;
CRYPT_MODE_CBC	-	режим зчеплення блоків шифру;
CRYPT_MODE_OFB	-	режим гамміpування;
CRYPT_MODE_CFB	-	режим гамміpування із зворотним зв'язком.

За умовчанням в *CRYPTOAPI* блокові шифри використовуються в режимі зчеплення блоків шифру (*CBC - cipher block chaining*). У цьому режимі використовується вектор, що ініціалізував (*IV - initialization vector*). За умовчанням *IV* нульовий. Вектор, що ініціалізував, повинен генеруватися окремо за допомогою функції *CryptGenRandom* і, як і солт-значення, передаватися разом з ключем у відкритому вигляді. Розмір *IV* рівний довжині блоку шифру. Наприклад, для алгоритму *RC2*, підтримуваного базовим криптопровайдером *Microsoft*, розмір блоку складає 64 бита (8 байтів).

Якщо режим шифрування даних повинен бути не *CBC*, то треба встановити відповідні параметри за допомогою функції *CryptSetKeyParam.*

CryptSetKeyParam(HCRYPTKEY hKey,

DWORD dwParam,

BYTE *pbData,

DWORD *pdwDataLen,

DWORD dwFlags);

Якщо ключ *hKey* сесійний, то як параметр можна встановити початковий вектор *KP_IV*, режим шифрування *KP_MODE*, спосіб доповнення *KP_PADDING*.

```
// Встановлюємо режим шифрування повідомлення OFB
DWORD dwMode = CRYPT_MODE_OFB;
if(!CryptSetKeyParam(hKey, KP_MODE, (BYTE*)&dwMode, 0)) {
puts ("Error CryptSetKeyParam!\n"); return -1;
}
```

10.8. Пересилка ключів в CRYPTOAPI

Як ключі експорту/імпорту можуть використовуватися або ключова пара *RSA* (з типом *AT_KEYEXCHANGE*), або симетричний сеансовий ключ. Обмін ключами реалізується за допомогою функцій *CryptExportKey* і *CryptImportKey***.**

Функція експорту ключа для його передачі по каналах інформації

```
BOOL WINAPI CryptExportKey(
HCRYPTKEY hKey,
HCRYPTKEY hExpKey,
DWORD dwBlobType,// PUBLICKEYBLOB-експорт відкритого ключа
// PRIVATEKEYBLOB -експорт закритого ключа
// SIMPLEBLOB-експорт сеансового ключа
DWORD dwFlags,
BYTE* pbData, //буфер, що містить ключовий блоб (зашифрований ключ)
DWORD* pdwDataLen); // розмір ключового блоба.
```

Виконаємо експорт сесійного ключа за допомогою функції *CryptExportKey* і запишемо результат у файл:

```
//Визначуваний розмір Bloba сесійного ключа
DWORD dwBlobLenght = 0;
if(CryptExportKey(hKey,hPublicKey,SIMPLEBLOB,0,0,&dwBlobLengh t))
{ printf(“size of the Blob”);
}
else
{
printf(“error computing Blob length”);
getchar(); return -1;
}
//Розподіляємо пам'ять для сесійного ключа
BYTE *ppbKeyBlob;
ppbKeyBlob = NULL;
if(ppbKeyBlob = (LPBYTE)malloc(dwBlobLen)) {
printf(“memory has been allocated for the Blob”);
}
else
{
printf ("Error memory for key length!!!"); getchar();
return -1;
}
//Зашифруємо сесійний ключ hKey відкритим ключом hPublicKey
if(CryptExportKey(hKey, hPublicKey, SIMPLEBLOB, 0, ppbKeyBlob, &dwBlobLenght)) {
printf(“contents have been written to the Blob”); }
else {
printf("Could not get exporting key.");
free(ppbKeyBlob);
ppbKeyBlob=NULL;
getchar(); return -1; }
//Записуємо експортований ключ у файл out.
```

```
if(fwrite(ppbKeyBlob,sizeof byte, dwBlobLenght,out))
{printf("the session key has been written to the file\n"); free(ppbKeyBlob);
}else
{
printf("the session key could not be written to the file\n"); getchar(); return -1;
}
```

Функція, призначена для отримання з каналів інформації значення ключа.

```
BOOL
CryptImportKey
(HCRYPTPROV
hProv,
BYTE* pbData,  //ключовий блоб (зашифрований ключ, отриманий в результаті дії функції CryptExportKey)
DWORD dwDataLen,  // довжина ключового блоба (у байтах)
HCRYPTKEY //дескриптор ключа, яким дешифруємо
hImpKey,
DWORD dwFlags,
HCRYPTKEY* // адреса, по якій функція копіює phKey); дескриптор мпортованого ключа
```

Приклад імпорту сесійного ключа:

```
//Розподіляємо пам'ять для сесійного ключа BYTE *ppbKeyBlob;
ppbKeyBlob = NULL;
if(ppbKeyBlob = (LPBYTE)malloc(dwBlobLen)) {
printf("memory has been allocated for the Blob");
}
else
{
printf ("Error memory for key length!!!"); getchar();
return -1;
}
//Прочитуємо сесійний ключ з файлу in.
if(fread(ppbKeyBlob,sizeof byte, dwBlobLenght,in)) {
printf("the session key has been read to the file\n"); free(ppbKeyBlob);
}
else
{
printf("the session key could not be read from the file\n"); getchar();
return -1;
}
//Імпортуємо сесійний ключ за допомогою закритого ключа ассиметричного алгоритму
hkey=0;
if(CryptImportKey(hProv,ppbKeyBlob,dwBlobLenght,hPrivateKey,0
,
&hKey))
{
printf(" the key has been imported.\n");
CryptDestroyKey(hPrivateKey); //очищаємо ресурси free(ppbKeyBlob);
} else {
printf("the session key import failed.\n"); getchar();
return -1;}
```

10.9. Симетричне шифрування

До групи функцій шифрування/розшифровки даних входять:

- *СгуіїЕпсгуії.* Основна базова функція шифрування даних. Як параметри використовує раніше отримані контексти криптопровайдера і сесійного ключа. Дані, що генеруються на виході цієї функції, не є форматованими і не містять ніякий іншій інформації, крім шифрованого контексту.
- *CryptDecrypt.* Основна базова функція розшифровки даних. Як параметри використовуються раніше отримані контекст криптопровайдера і ідентифікатор сесійного ключа.

Базова функція шифрування даних має наступне оголошення:

```
BOOL
CryptEncrypt(
HCRYPTKEY //ідентифікатор сесійного ключа
hKey, HCRYPTHAS // використовується для отримання хеш- кодування даних.
hHash,
BOOL Final, //дозволяє обробляти дані блоками
DWORD  //служить покажчиком на масив
dwFlags,  вхідних/вихідних даних. Зазвичай встановлюється в 0.
BYTE* pbData, //порція даних для шифрування
DWORD* //служить для повернення розміру даних,
pdwDataLen,  повертаних функцією
DWORD.  // служить для вказівки довжини вхідного
dwBufLen);
буфера даних
```

Приклад використання функції CryptEncrypt приведений нижче:

```
# define BLOCK_SIZE 15
BYTE *pCryptBuf = 0;
DWORD buflen;
BOOL bRes;
DWORD datalen;
// Визначуваний розмір буфера необхідного для блоків довжини BLOCK SIZE
buflen=BLOCKSIZE;
if(!CryptEncrypt(hKey, 0,TRUE,0,NULL,&buflen, 0 ))
{
cout<<" Crypt Encrypt (bufSize) failed."«endl;
getchar ();
return -1;
}
//Виділимо пам'ять під буфер
pCryptBuf = (BYTE*)malloc( buflen );
int t=0;
// Шифруємо файл IN
while((t=fread(pCryptBuf,sizeof byte,BLOCK_SIZE,in))) { datalen = t;
bRes=CryptEncrypt( hKey,0, TRUE,0,pCryptBuf,&datalen,buflen);
if( !bRes ) {
cout«"CryptEncrypt (encryption) failed, "«endl; getchar(); return -1;
}
fwrite(pCryptBuf,sizeof byte,datalen,out);
}
cout«"File encryption completed successfully"«endl; fclose(in);
fclose(out);
free(pCryptBuf);
CryptDestroyKey(hKey);
CryptReleaseContext(hProv,0);
```

Приклад розшифровки файлу за допомогою функції CryptDecrypt.

```
// Визначуваний розмір буфера необхідного для блоків довжини BLOCK SIZE
buflen=BLOCK_SIZE;
bRes = CryptEncrypt( hKey,0, TRUE,0, NULL, &buflen,0); pCryptBuf = (BYTE*)malloc(
buflen );
// Розшифровуємо файл
while((t=fread(pCryptBuf,sizeof byte,datalen,in)))
{
buflen = t;
bRes = CryptDecrypt( hKey, 0,TRUE,0,pCryptBuf,&buflen ) ;
if( !bRes )
{
cout<<"CryptEncrypt (buffer size) failed, "«endl;
getchar(); return -1;
}
fwrite(pCryptBuf,sizeof byte,buflen,out);
}
cout«"File decryption completed successfully"«endl;
```

11 Криптографічна стійкість шифрів

11.1 Поняття криптографічної стійкості

Криптографічна стійкість (або крипостійкість) - здатність криптографічного алгоритму протистояти можливим атакам на нього.

Стійким вважається алгоритм, який для успішної атаки вимагає від противника:

недосяжних обчислювальних ресурсів,

недосяжного обсягу перехоплених відкритих і зашифрованих повідомлень чи

такого часу розкриття, що по його закінченню захищена інформація буде вже не актуальна, і т. д.

Стійкість не можна підтвердити, її можна тільки спростувати зломом.

Для оцінки стійкості шифру використовують оцінку обчислювальної складності алгоритму успішної атаки на криптосистему.

Рівень криптостійкості (англ. security level) - показник криптостійкості алгоритму, пов'язаний з обчислювальною складністю виконання успішної атаки на криптосистему.

Зазвичай рівень криптостійкості вимірюється в бітах. N-бітний рівень криптостійкості криптосистеми означає, що для її злому потрібно виконати n обчислювальних операцій. Наприклад, якщо симетрична криптосистема зламуються не швидше, ніж за повний перебір значень n-бітного ключа, то кажуть, що рівень криптостійкості дорівнює n.

Складність алгоритмів зазвичай оцінюють за часом виконання, але не менш важливі й інші показники - вимоги до обсягу пам'яті, вільного місця на диску.

В теорії алгоритмів обчислювальна складність алгоритму - це функція, яка визначає залежність обсягу роботи, виконуваної деяким алгоритмом, від розміру вхідних даних.

У загальному випадку складність алгоритму можна оцінити по порядку величини. Алгоритм має складність O (f (n)), якщо при збільшенні розмірності вхідних даних n, час виконання алгоритму зростає з тією ж швидкістю, що і функція f (n).

Використання великої літери O (або так звана O-нотація) прийшло з математики, де її застосовують для порівняння асимптотичної поведінки функцій. Формально O (f (n)) означає, що час роботи алгоритму (або обсяг займаної пам'яті) росте в залежності від обсягу вхідних даних не швидше, ніж деяка константа, помножена на f (n).

Типові приклади використання O-нотації:

O (n) - лінійна складність. Таку складність має, наприклад, алгоритм пошуку найбільшого елемента в невідсортованому масиві. Нам доведеться пройтися по всіх n елементах масиву, щоб зрозуміти, який з них максимальний.

O (log n) - логарифмічна складність. Найпростіший приклад - бінарний пошук. Якщо масив відсортований, ми можемо перевірити, чи є в ньому якесь конкретне значення, шляхом поділу навпіл. Перевіримо середній елемент, якщо він більше шуканого, то відкинемо другу половину масиву - там його точно немає. Якщо ж менше, то навпаки - відкинемо початкову половину. І так будемо продовжувати ділити навпіл, в результаті перевіримо log n елементів.

O (n2) - квадратична складність. Таку складність має, наприклад, алгоритм сортування вставками. У канонічної реалізації він вдає із себе два вкладених циклу: один, щоб проходити по всьому масиву, а другий, щоб знаходити місце чергового елементу уже відсортованої частини. Таким чином, кількість операцій буде залежати від розміру масиву як n * n.

Для наочності наводимо час виконання алгоритму з певною складністю в залежності від розміру вхідних даних при швидкості виконання 106 операцій в секунду (Таблиця 11.1).

Таблиця 11.1. Час виконання алгоритму з певною складністю в залежності від розміру вхідних даних при швидкості виконання 106 операцій в секунду

розмір складність	10	20	ЗО	40	50	60
N	0,00001 сек.	0,00002 сек.	0,00003 сен.	0,00004 сек.	0,00005 сек.	0,00005 сек.
N^2	0,0001 сен.	0,0004 сек.	0,0009 сек.	0,0016 сек.	0,0025 сен.	0,0036 сек.

N^3	0,001 сек	0,003 сек.	0,027 сек.	0,064 сек.	0,125 сек.	0,216 сек.
N^5	0,1 сек.	3,2 сен.	24,3 сек.	1,7ХВ.	5,2 хв.	13 іХВ.
2^N	0,0001 сек.	1 сек.	17,9 ХВ.	12,7 ДНІВ	35,7 століть	366 століть

Множина задач, для вирішення яких існують алгоритми, схожі за обчислювальною складністю, утворює клас складності.

Розрізняють такі основні класи складності:

Клас P - вміщує всі ті проблеми, вирішення яких вважається «швидким», тобто поліноміально залежать від розміру входу (наприклад, сортування, пошук, з'ясування зв'язності графів і т.п.).

Клас NP - містить задачі, які не в змозі вирішити за поліноміальну кількість часу (наприклад, факторизація, дискретне логарифмування та інші).

Межі застосування «грубої сили» до атак на шифри

Найбільш загальний вид атаки на шифр - метод повного перебору (або метод «грубої сили»).

Повний перебір (англ. brute force) - загальний метод вирішення завдань шляхом перебору всіх можливих потенційних рішень.

У криптографії на обчислювальній складності повного перебору ґрунтується оцінка криптостойкости шифрів. Зокрема, шифр вважається крипостійким, якщо не існує методу «злому» істотно більш швидкого, ніж повний перебір всіх ключів. Криптографічні атаки, засновані на методі повного перебору, є найбільш універсальними, але і найдовшими.

Стійкість до brute-force атаки визначає використовуваний в криптосистемі ключ шифрування. Так, зі збільшенням довжини ключа складність злому цим методом зростає експоненціально. У найпростішому випадку шифр довжиною в n бітів зламуються, в найгіршому випадку, за час O(2П).

Взагалі будь-яка задача з класу NP може бути вирішена повним перебором, але це вимагатиме експоненціального часу роботи.

Існують способи підвищення стійкості шифру до «brute force», наприклад заплутування (обфускація) шифрованих даних, що робить нетривіальним відмінність зашифрованих даних від незашифрованих.

Виходячи з фізичних міркувань можна показати, що існує можливість вибрати таку довжину ключа, що таку атаку методом «грубої сили» неможливо буде виконати в принципі, безвідносно до потужностей наявної обчислювальної техніки.

Доведення 1 (термодинамічний підхід).

За законами термодинаміки поглинання енергії при здійсненні незворотного перетворення має порядок ~kT, де k=1.4*10-23 Дж/К, Т - температура зовнішнього середовища.

Потужність сонячного випромінення Pc~3.86 *1026 Вт. Час існування

Всесвіту tc ~14 млд. років~14*109 років ~ 4*1019 с. Температура Сонця Tc~106 K.

Тоді витрати на одну обчислювальну операцію ~kT=1.4*10-23*106~1.4*10-17 (Дж).

Вся енергія виділена Сонцем за час його існування Всесвіту E= Pc* tc ~ 3.86 *1026*4*1019 ~1.6* 1046 (Дж). Тому можлива кількість виконаних операцій N=E/ kT~1.6* 1046/ 1.4*10-17~ 1063.

А це означає, при довжині ключа шифрування L>63 перебір всіх варіантів не можна здійснити за час існування Всесвіту.

Доведення 2 (гео-протонний підхід).

Маса Землі M~6*1024 кг, маса протону т~1.6*10’27 кг; тобто Земля містить ^M/т~ 6*1024 /

1.6*1027 ~3.8*1051 протонів.

Співставим кожному протону комп’ютер і вважатимемо, що на виконання однієї операції в ньому витрачається час 1, за який світловий промінь (швидкість с~3*1010 м/с)

проходить відстань, рівну діаметру протона ^~10-15 м): 1= d/c~10-15/3*1010 ~ 1/3*10—25 (с).

Це означає, що швидкість роботи одного такого уявного комп'ютера 1-1 ~

3*1025 операцій/с.

Сумарна швидкодія всіх «протонних» комп'ютерів V=N*v ~ 3.8*1051 * 3*1025~1076 операцій/с. За весь час існування Всесвіту 1с ~ 4*1019 с можна виконати не більше ніж М^* 1с~ 1076 * 4*1019 ~1095 операцій.

Знову ж таки, це означає, при довжині ключа шифрування L>95 перебір всіх варіантів не можна здійснити за час існування Всесвіту.

Резюмуючи, робимо наступний висновок: «хорошим» ключем шифрування (тобто таким, який не можна підібрати методом «грубої сили») є ключ довжиною не менше L>100.

11.2 Абсолютна криптостійкість шифрів

Абсолютна крипостійкість означає, що:

результатом розшифрування може бути будь-який текст,

не існує ніякого способу перевірки, що розшифрування виконане правильно.

Доведення існування абсолютно стійких алгоритмів шифрування було виконано Клодом Шенноном та опубліковано в роботі «Теорія зв'язку в секретних системах» (1946 рік). Там же визначені вимоги до такого роду систем:

Ключ генерується для кожного повідомлення (кожен ключ використовується один раз).

Ключ статистично надійний (тобто ймовірності появи кожного з можливих символів однакові, символи в ключовий послідовності незалежні і випадкові).

Довжина ключа дорівнює або більше довжини повідомлення.

Стійкість цих систем не залежить від того, якими обчислювальними можливостями володіє криптоаналітик.

Майже всі використовувані на практиці шифри характеризуються як умовно надійні, оскільки вони можуть бути розкриті в принципі при наявності необмежених обчислювальних можливостей. Абсолютно надійні шифри не можна зруйнувати навіть при наявності необмежених обчислювальних можливостей.

Єдиним відомим шифром, який задовольняє вимогам абсолютної криптостійкості, є шифр Вернама.

Шифр Вернама (інша назва: англ. One-time pad - схема одноразових блокнотів) названий на честь телеграфіста AT & T Гільберта Вернама, який в 1917 році побудував телеграфний апарат, що виконував цю операцію автоматично - треба було тільки подати на нього стрічку з ключем.

Для шифруівння відкритий текст «сумується» з ключем (який називається одноразовим блокнотом або шифроблокнотом) аналогічно розширеному шифру Цезаря. Але при цьому ключ повинен задовольняти трьом критично важливими умовам:

Бути істинно випадковим.

Співпадати за розміром з заданим відкритим текстом.

Застосовуватись тільки один раз.

Будемо вважати, що число символів розширеного алфавіту, тобто сукупності літер, а також знаків пунктуації та пропусків між словами, дорівнює N. Занумерувавши всі символи розширеного алфавіту числами від 0 до N, текст, що передається, можна розглядати як послідовність {an} чисел множини A = {0,1,2, ..., N}.

Маючи випадкову послідовність {кп} з чисел множини А тієї ж довжини що і ключ, складаємо по модулю N число ап переданого тексту з відповідним числом кп ключа ап + кп = сп (mod К), 0 < сп < N одержимо послідовність {сп} знаків шифрованого тексту.

Щоб отримати переданий текст, можна скористатися тим ж ключем: ап = сп- кп (mod К), 0 < ап < N.

У двох абонентів, що знаходяться в секретному листуванні, є два однакових блокнота, складених з відривних сторінок, на кожній з яких надрукована таблиця з випадковими числами або буквами, тобто випадкова послідовність чисел з множини А. Таблиця має тільки дві копії: одна використовується відправником, інша - одержувачем.

Приблизна сторінка шифроблокноту показана на Рис.3.1.

гохмм езкаио ресіре ИБнгір рхркіу обміні jzti.bc уібоуо
НТБУТУ БНУУЕБ ЕХМСЕА БЕдУБР РНІСІБ ЕАТІЗІЗК 01Х504 0Б6ІАК
унотеч тлсдт нгвзії ібюуб бвуорі аінсиі осухтл' укрдмк
СКНУЕХ УХУУСЗ МОЕДАІ ОСТУПИ БСНЕУИ І_МБІ_У0 5В0С0С РСБУОХ
дхдиір рхіди лииун сатшмз июоні оинреи ивзяиз нбаари
С0*7НІ РБНТЧН А21У*БТ АСАКИО БЇКІІБ УідбБК ІЗС'л'НР БТТЗЗЕ
ЕХРІРБ КІСА55 юисзтр 2ББХБН УТУСТІ ВА25ТМ ЗКМРХІ РЕРУ^'Е

Рис.3.1. Приклад сторінки шифроблокноту

Відправник свій текст шифрує зазначеним вище способом за допомогою першої сторінки блокнота. Зашифрувавши повідомлення, він знищує використувану сторінку і відправляє її іншому абоненту. Одержувач шифрованого тексту розшифровує його і також знищує використаний лист блокнота.

Неважко бачити, що одноразовий шифр не можна розкрити в принципі, так як символ у тексті може бути замінений будь-яким іншим символом і цей вибір абсолютно випадковий.

11.3 Основи квантової криптографії

Квантова криптографія! - наука, що вивчає методи захисту систем зв’язку і базується на принциповій непорушності закономірностей квантової фізики, об’єкти якої забезпечують процеси безпечного передавання інформації між легітимними користувачами.

Основними засадами є:

постулат вимірювання (результат принципу невизначеності Гейзенберґа) та

теорема про заборону клонування.

Принцип невизначеності - це фундаментальна нерівність (відношення невизначеностей), згідно з якою усі динамічні параметри поділяють на дві групи: перша - часові та просторові координати; друга - імпульс та енергія. При чому неможливо одночасно виміряти параметри з різних груп (напр., місцерозташування та енергію об’єкта).

Теорема про заборону клонування стверджує неможливість створення точних (на 100 %) копій квантових об’єктів.

Ці властивості не притаманні класичним системам зв’язку і забезпечують абсолютну й безумовну стійкість квантової криптографії. В останні роки значний інтерес викликають системи захисту інформації на базі методів квантової криптографії. Виокремлюють такі напрями, як квантовий розподіл ключів (КРК), квантовий прямий безпечний зв’язок (КПБЗ), квантове розділення секрету (КРС), квантовий цифровий підпис (КЦП) та кванттова стеганографія.

Ідею використання квантових об’єктів для захисту інформації вперше запропонував С. Вайснер (1970 рік). В 1984 році Ч. Беннет (компанія ІВМ) та Ж. Брассар (Монреальський університет) розвинули її і запропонували перший протокол квантової криптографії ВВ84, що став альтернативним вирішенням проблеми розподілу ключів шифрування.

У ньому для передачі даних використовуються фотони, поляризовані в чотирьох різних напрямках, в двох базисах - під кутом 0 і 90 градусів (позначається знаком +) або 45 і 135 градусів (позначається знаком х).

Відправник повідомлення Аліса поляризує кожен фотон в випадково обраному базисі, а потім відправляє його одержувачу Бобу. Боб вимірює кожен фотон, теж в випадково обраному базисі. Після цього Аліса по відкритому каналу повідомляє Бобу послідовність своїх базисів, і Боб відкидає неправильні (не співпали) базиси і повідомляє Алісі, які дані «не пройшли». При цьому самі значення, отримані в результаті вимірів, вони по відкритому каналу не обговорюють. Якщо шпигун Єва захоче перехопити секретний ключ, вона повинна буде вимірювати поляризацію фотонів. Оскільки вона не знає базису, то повинна буде визначати його випадковим чином. Якщо базис буде визначено неправильно, то Єва не отримає вірних даних, а крім того, змінить поляризацію фотона. Помилки, що з’являться, відразу виявлять і Аліса, і Боб.

Описану схему можна проілюструвати наступним чином:

Аліса надсилає фотони, що мають одну з чотирьох можливих поляризацій, яку вона обирає випадково.

Для кожного фотона Боб обирає випадково тип вимірювання: прямолінійна поляризація (+), або діагональна (х).

Боб записує та таємно зберігає отримані результати вимірювань.

Боб відкрито оголошує типи вимірювань, які він обрав для кожного фотона, а Аліса повідомляє йому, які вимірювання були правильними.

Аліса і Боб зберігають всі результати, що були отримані у тих випадках, коли Боб застосував правильний тип вимірювання. Ці результати переводять у біти (0 і 1), послідовність яких і є результатом квантового розподілу ключа.

Протокол BB84 вважають основоположником протоколів квантової криптографії. Нині існують вже кілька десятків декілька подібних систем, які вийшли на світовий ринок і позиціонуються як надійні системи розподілу ключів та криптографічного захисту інформації.

Першим у світі комерційним рішенням була система QPN Security Gateway (QPN-8505), запропонована американською компанією MagiQ. Вона пропонує захист VPN за допомогою КРК (бл. 100 ключів 256 біт за секунду на відстань до 140 км) та інтегроване шифрування. За допомогою цієї системи можна організувати захищену квантову мережу.

Іншим ефективним рішенням є система КРК Clavis швейцарської компанії ID Quantique. З її допомогою можна здійснювати захищений розподіл ключів шифрування між двома абонентами на відстань до 100 км. Крім того, компанія пропонує квант. систему Cerberis. Ця система може розподіляти ключі на відстань до 50 км.

На початку 21 ст. британською компанією Toshiba Research Europe Ltd представлено систему КРК Quantum Key Server, яка відрізняється простотою своєї архітектурири та забезпечує генерацію близько 100 ключів (довжиною 256 біт) за секунду та їхнє одностороннє передавання від передавача до приймача.

Інша британська компанія QinetiQ запропонувала першу у світі комп'ютерну мережу, що використовує квантову криптографію, - Quantum Net (Qnet). Максимальна довжина ліній зв'язку цієї мережі становить 120 км.

12 Основи криптоаналізу

12.1. Криптоаналітика як наука

Криптоаналіз - це наука про розкриття вихідного тексту зашифрованого повідомлення без доступу до ключа. Успішний аналіз може розкрити вихідний текст або ключ. Він дозволяє також знайти слабкі місця в криптосистемі, що, у кінцевому рахунку, веде до тих же результатів.

Фундаментальне правило криптоаналізу, уперше сформульоване голландцем А. Керкхоффом ще в XIX столітті полягає в тому, що стійкість криптосистеми повинна визначатися тільки таємністю ключа. Іншими словами, правило Керкхоффа полягає в тому, що весь алгоритм шифрування, крім значення секретного ключа, відомий для супротивника. Це обумовлено тим, що криптосистема, що реалізує сімейство криптографічних перетворень, звичайно розглядається як відкрита система. Такий підхід відбиває дуже важливий принцип технології захисту інформації: захищеність системи не повинна залежати від таємності чого-небудь такого, що неможливо швидко змінити у випадку витоку секретної інформації. Звичайно криптосистема являє собою сукупність апаратних і програмних засобів, яку можна змінити тільки при значних витратах часу і засобів, тоді як ключ є легко змінюваним об'єктом. Саме тому стійкість криптосистеми визначається тільки таємністю ключа.

Інше майже загальноприйняте допущення в криптоаналізі полягає в тому, що криптоаналітик має у своєму розпорядженні шифротексти повідомлень.

Існує чотири основних типи криптографічних атак. Звичайно, усі вони формулюються в припущенні, що криптоаналітику відомо застосовуваний алгоритм шифрування і шифротексти повідомлень. Перерахуємо ці криптоаналітичні атаки.

1) Криптоаналітична атака при наявності тільки відомого шифртексту. Криптоаналітик має тільки шифртексти *C1, C2,... C_m* декількох повідомлень, причому усі вони зашифровані з використанням того самого алгоритму шифрування *EK*. Робота криптоаналітика полягає в тому, щоб розкрити вихідні тексти *M1, M2, ..., M_m*, або, ще краще, обчислити ключ *K*, використаний для шифрування цих повідомлень, для того, щоб розшифрувати й інші повідомлення, зашифровані цим ключем.

2) Криптоаналітична атака при наявності відомого відкритого тексту. Криптоаналітик має доступ не тільки до шифротекстів *C1, C2,... C_m* декількох повідомлень, але також до відкритих текстів *M1, M2, ..., M_m* цих повідомлень. Його робота полягає в пошуку ключа, використовуваного при шифруванні цих повідомлень, або алгоритму розшифрування D_K будь-яких нових повідомлень, зашифрованих тим же самим ключем.

3) Криптоаналітична атака при можливості вибору відкритого тексту. Криптоаналітик не тільки має доступ до шифрів-текстів *C1, C2,... C_m* і зв'язаним з ними відкритим текстам *M1, M2, ..., M_m* декількох повідомлень, але і може за бажанням обирати відкриті тексти, що потім одержує в зашифрованому виді. Такий криптоаналіз виходить більш могутнім у порівнянні з криптоаналітиком за відомим відкритим текстом, тому що криптоаналітик може вибрати для шифрування такі блоки відкритого тексту, що дадуть більше інформації про ключ. Робота криптоаналітика складається в пошуку ключа, використаного для шифрування повідомлень, або алгоритму расшифрування *OK* нових повідомлень, зашифрованих тим же ключем.

4) Криптоаналітична атака з адаптивним вибором відкритого тексту. Це особливий варіант атаки з вибором відкритого тексту. Криптоаналітик може не тільки обирати відкритий текст, що потім шифрується, але і змінювати свій вибір у залежності від результатів попереднього шифрування. При криптоаналізі з простим вибором відкритого тексту криптоаналітик звичайно може вибирати кілька великих блоків відкритого тексту для їхнього шифрування; при криптоаналізі з адаптивним вибором відкритого тексту він має можливість вибрати спочатку більш дрібний спробний блок відкритого тексту, потім вибрати наступний блок у залежності від результатів першого вибору, і т.д. Ця атака надає йому ще більше можливостей, чим попередні типи атак.

Крім перерахованих основних типів криптоаналітичних атак, можна відзначити, принаймні, ще два типи.

5) Криптоаналітична атака з використанням обраного шифротекста. Криптоаналітик

може обирати для розшифрування різні шифротексти *C1, C2,... C_m* і має доступ до розшифрованих відкритих текстів *M1, M2, ..., M_m*. Наприклад, він одержав доступ до захищеного від несанкціонованого розкриття блоку, що виконує автоматичне розшифрування. Робота аналітика полягає в відшуканні ключа. Цей тип криптоаналізу становить особливий інтерес для розкриття алгоритмів з відкритим ключем.

6) Криптоаналітична атака методом повного перебору всіх можливих ключів. Ця атака припускає використання криптоаналітиком відомого шифротексту і здійснюється за допомогою повного перебору всіх можливих ключів з перевіркою, чи є осмисленим відкритий текст, що виходить. Такий підхід вимагає залучення великих обчислювальних ресурсів і іноді називається силовою атакою. Існують і інші, менш розповсюджені, криптоаналітичні атаки.

7)

12.2. Основні принципи криптоаналізу

1. **Принцип Керкхоффа.** Тільки супротивник може судити про криптостійкість системи.

2. **Принцип Керкхоффа-Шеннона.** Супротивник знає використовувану систему з точністю до ключової інформації.

3. **Принцип Жеверже.** Поверхневі ускладнення системи можуть бути ілюзійними, так як породжують хибні оцінки її стійкості.

4. При оцінці криптостійкості необхідно враховувати можливі криптографічні помилки та інші порушення безпеки.

Криптограф Ларс Кнудсен пропонує наступну класифікацію успішних результатів криптоаналізу блочних шифрів в залежності від об'єму і якості секретної інформації, яку вдалося отримати:

- повний злом - криптоаналітик обчислює ключ;
- глобальна дедукція – криптоаналітик розробляє функціональний еквівалент досліджуваного алгоритму, що дозволяє зашифровувати і розшифровувати інформацію без знання ключа;
- часткова дедукція – криптоаналітику вдається розшифрувати деякі повідомлення;
- інформаційна дедукція - крипто аналітик отримує деяку інформацію про відкритий текст чи ключ.

Як правило, криптоаналіз розпочинається із спроб злому (атаки) спрощеної версії алгоритму, після чого результати розповсюджуються на повну версію.

Властивість системи протистояти будь-яким атакам називається її **стійкістю**. Кількісно стійкість визначається складністю криптоалгоритму. **Універсальний метод прямого перебору** всієї множини ключів дозволяє зробити оцінку зверху для стійкості алгоритмів.

Розрізняють стійкість ключа (складність розкрити найкращим методом); стійкість безключового читання (складність нав'язування неправдивої інформації найкращим алгоритмом).

Аналогічно розглядають стійкість криптоалгоритму, протоколу, алгоритму генерування і розподілу ключів. У залежності від складності злому алгоритми забезпечують різні степені захисту**. Існують безумовно стійкі (теоретично стійкі), доведено стійкі і припустимо стійкі.**

Теоретично стійкий криптоалгоритм - такий алгоритм, коли ніякий метод криптоаналізу не дозволяє не тільки визначити ключ і відкритий текст, але й навіть отримати деяку інформацію про них. Такі системи на практиці незручні (симетричні системи із одноразовим використанням ключа потребують більшої захищеної пам'яті для збереження ключів, системи квантової криптографії потребують волоконно-оптичних каналів зв'язку, які є дорогими). Крім того доведення їх стійкості виходить за межі області математики і фізики. Тому абсолютно стійкі шифри використовуються тільки в межах зв'язку з невеликим об'ємом інформації, що передається. Зазвичай, ці мережі використовуються для передачі особливо важливої державної інформації.

Стійкість **доведено стійких криптоалгоритмів** визначається складністю розв'язання

відомої математичної задачі. Приклад таких криптосистем: Діффі- Хеллмана - складність дискретного логарифмування, RSA - складність розкладу великого числа на множники та інші.

Припустимо стійкі криптоалгоритми засновані на складності розв'язання частинної математичної задачі. Приклади: FEAL, AES та інші. Ці шифри характеризує порівняно мала вивченість математичних задач, на яких базується їх стійкість. Однак, такі шифри мають велику гнучкість, що дозволяє при визначенні слабих місць не відмовлятися від алгоритму, а провести його доробку.

12.3. Універсальні методи криптоаналізу

Метод повного перебору ключа.

Часто криптоаналітики розкривають шифри на ЕОМ методом перебору ключів. У процесі криптоаналізу доводиться перебирати більше міліарду

ключів зі швидкістю тисяча ключів у секунду.

Якщо впорядкувати множину ключів $K=\{k_1, k_2,...,k_n\}$ то повний перебір потребує А' операцій, де перевірка одного ключа одна операція. Ймовірність правильного вибору ключа - $\frac{1}{|K|}$.

Алгоритми повного перебору допускають розпаралелення, що дозволяє значно прискорити процес знаходження ключа. Є різні підходи до реалізації розпаралелення: метод конвеєру, поділ задачі, «китайська лотерея», «криптографічні водорості».

Атака на ключі.

Однією із вразливостей криптосистем є використання слабих ключів, які не забезпечують достатнього рівня захисту. Відповідно до вже згаданого принципу Кіркхгофа, стійкість криптоалгоритма визначається секретністю ключа. Слабкий ключ - це ключ, не забезпечує достатнього рівня захисту, чи використовує в шифруванні закономірності, які можуть бути зламані. Зазвичай вважається, що алгоритм симетричного шифрування повинен по можливості не мати слабких ключів. Якщо це неможливо, то кількість слабких ключів має бути мінімальною. Тим не менш, всі слабкі ключі, якщо їх не можливо уникнути, мають бути відомими (як у алгоритмі DES).

Генератор випадкових чисел - ще одне слабе місце криптосистем. Це означає, що, якщо для генерування ключів використовується криптографічний слабкий алгоритм, незалежно від використовуваного шифру вся система буде нестійкою. Якісний ключ, призначений для використання в рамках симетричної криптосистеми, являє собою випадковий двійковий набір. Якщо потрібно ключ розрядністю n , в процесі його генерування з однаковою ймовірністю повинен виходити з будь-яких 2^n можливих варіантів. Генерування ключів для асиметричних криптосистем - процедура складна, так як ключі, вживані в таких системах, повинні володіти певними математичними властивостями. Наприклад, у випадку системи RSA модуль шифрування являє собою добуток двох великих простих чисел.

За допомогою генераторів псевдовипадкових можна будувати стійкі криптосистеми. Хороші генератори випадкових чисел складні в розробці, так як їх надійність часто залежить від особливостей апаратного та програмного забезпечення.

Частотний аналіз.

Визначення окремих символів і їх сполучень. Цей аналіз використовує статистичні і лінгвістичні методи для отримання додаткової інформації про ключ.

Лінійний аналіз.

Заміна нелінійної функції криптографічного алгоритму деяким лінійним аналогом. Найчастіше застосовується для симетричних алгоритмів.

Різницевий аналіз.

Різницевий або диференціальний аналіз використовує аналіз пари відкритих текстів, що мають визначені відмінності, після їх шифрування. Використовується для блочних систем і загальним методом.

Методи безключового читання.

Метод читання в колонках реалізує спосіб відновлення тексту, зашифрованого

нерівноймовірною гамою.

Метод Полларда. Метод колізій для хеш-функцій. Метод «зустрічі посередині» та інші.

12.4. Первинний аналіз шифровки

Опишемо деякі найбільш прості та ефективні методики розкриття зашифрованого тексту. Ці методики, звичайно, не дають гарантії розкриття будь-якого шифру, бо криптоаналіз це скоріше мистецтво, засноване на інтуїції і досвіді, ніж точна наука, заснована на законах і формулах. Однак ці методики слугують інструментом злому, свого роду набором відмичок, які можуть принести результат лише в руках досвідченого майстра. Деякі методики застосовуються для розкриття ручних шифрів, а деякі для розкриття машинних. В обох випадках злом зазвичай відбувається з використанням обчислювальної техніки, що дозволяє позбавитися від рутинного перебору ключів за ручним методом. При цьому вдалий лозунг «працювати повинна машина, а думати повинна людина». При традиційному криптоаналізі, результат роботи часто визначається кваліфікацією зломщика, а при машинному - швидкістю обчислювальної системи. Оптимальне співвідношення цих двох факторів є необхідною умовою успіху крипто аналізу.

Завжди процес криптоаналізу починається зі збору різноманітних відомостей про зашифрований документ:

- на якій мові написаний оригінал;
- які лінгвістичні особливості даної мови (вживання артиклів і займенників, узгодження відмінків, правила утворення суфіксів і префіксів, порядок слідування слів у реченні, ...);
- яка мінімальна змістовна одиниця інформації (біт, символ, слово,...);
- які імовірно слова і фрази можуть знаходитися в тексті і в якій послідовності;
- яка приблизно довжина вихідного тексту;
- які ймовірно методи шифрування могли бути застосовані;
- яка службова інформація може знаходитися в тексті (контрольна сума пакета, адресна інформація, дата передачі);
- на який апаратурі шифрувався текст.

Такі і подібні дані збираються агентурними або аналітичними шляхами, що часто значно полегшує поставлене перед зломщиком задачу.

Первинний аналіз шифровки - дія, яка проводиться завжди при зломі будь-якого шифру. Так як будь-яка перехоплена шифровка (папірус з ієрогліфами, клаптик паперу з колонками цифр, повідомлення на морзянці, закодований файл або протокол активності комп'ютерної мережі) в кінцевому підсумку набрана за допомогою певного алфавіту (ієрогліфічного, цифрового, літерного, байтового), то можна, по-перше, оцінити інформативність всього повідомлення, по-друге, проаналізувати діаграму розподілу ймовірностей появи окремих символів алфавіту в тексті.

Інформативність повідомлення в тому сенсі, який запропонував Шеннон, визначається за формулою:

$$H = \left| \sum P_i \cdot \log_2(P_i) \right|$$

де P_i- частота появи і-го символу в тексті. Простіше кажучи, чим вище інформативність, тим більше щільність корисної інформації в обсязі повідомлення.

Другим пунктом після визначення інформативності шифровки є побудова **діаграми розподілу ймовірностей прояву окремих символів в тексті шифровки**. Переважно діаграму представляють у вигляді лінійного графіка, де по одній осі відкладені порядкові номери символів в алфавіті, а по інший нормалізовані ймовірності прояву цих символів у тексті шифровки.

Ймовірності появи окремих символів англійської, української, російської мов стандартного літературного тексту є у вільному доступі. Ймовірності наведені в процентному поданні та округлені до цілого.

Твердження про те, що ймовірність прояву чергового символу в середньостатистичному тексті є величина незалежна неправильно, так як деяких поєднань символів в мові взагалі може не бути. Правильніше сказати, що ймовірність появи певного символу в черговий позиції тексту залежить від попереднього символу. Якщо припустити, що алфавіт повідомлення складається з символів української мови і пробілу, то неважко вивести таблицю ймовірностей появи конкретних біграм в довільному українському тексті. Аналогічно з українськими біграмами, можна скласти таблицю ймовірностей прояву англійських біграм в англомовному тексті та інше.

Крім біграм, можна побудувати аналогічні таблиці для триграм, тетраграмма і т.д.

Отже, після всього сказаного, можна зробити висновок, що діаграма розподілу ймовірностей прояву символів і інформативність можуть дати важливі непрямі відомості про мову шифровки або хоча б про можливий тип шифрування.

12.5. Конкретні приклади криптоаналізу

Якщо за результатами первинного аналізу шифровки був успішно визначений метод шифрування, то можна вважати, що ми зробили близько 20% всієї роботи. Якщо методами шифрування виявилися різні види підстановок, перестановок шифрування біграм або шифрування по системі Віжінера, а довжина шифровки досить велика, то можна сказати, що дешифрування вихідного тексту діло техніки. Якщо найбільш імовірним методом шифрування виявилося гамування, а ще гірше взбивка, то доведеться визнати, що робота по розкриттю шрифту тільки починається і радіти ще дуже рано.

Ми ніколи не доб'ємося розкриття шрифту, якщо в результаті первинного аналізу шифровки ми зробили неправильне припущення про метод шифрування.

12.5.1. Криптоаналіз шифрів за методом перестановок

Перед описом методу злому шифрування за перестановкою перерахуємо кілька властивостей даного методу шифрування, якими треба буде скористатися.

По-перше, якою б не була перестановка, у неї завжди є максимальний радіус. Іншими словами, символ може переміщатися з поточної позиції не більше ніж на відстань R вперед або назад.

По-друге, для розрахунку зсуву конкретного символу відносно його положення в початковому тексті зазвичай використовується проста математична формула, а не повна матриця перестановок, так як в протилежному випадку обсяг ключової інформації для шифрування буде близьким до об'єму вихідного тексту.

І по-третє, математична формула для перестановки символів повинна бути однозначною, тобто кожному символу в початковому тексті повинен відповідати тільки один символ тексту шифровки і не повинно бути накладок.

Розкриття шифру перестановки зазвичай здійснюють перебором можливих формул і аналізом одержуваного відкритого тексту. В якості аргументів у формулах фігурують значення поточної позиції **i** і максимального радіуса **R**. В якості операцій над аргументами у формулах можуть використовуватися додавання, віднімання, побітового операції, множення, зведення в ступінь, взяття залишку за модулем **R,** а також різні комбінації цих операцій.

Під час розв’язання таких задач необхідно відновити початковий порядок слідування букв тексту. Для цього використовують аналіз сумісності символів та таблиці частот біграм відповідної мови для першого рядка таблиці.

Приклад 1.

Розшифрувати текст «ИСВИЇПНКУАПЬОТІРІУВКИТПАЦ»,

зашифрований стовпцевою перестановкою, без ключа (ключем є розмір таблиці та порядок перестановки стовпців).

Текст містить 25 символів, тому, ймовірно, можемо розглядати таблицю розміром 5х5. Таким чином, запишемо текст по СТОВПЦЯХ у таблицю. Можемо припустити, що мова на якій написаний текст українська(буква Ї).

И	П	П	Р	И
С	Н	Ь	І	Т

В	К	О	У	П
И	У	Т	В	А
Ї	А	І	К	Ц

Розглянемо у першому рядку можливі біграми і згідно додатку А запишемо їх ймовірності:

ПР- **0,0071** ПИ-0, 0011 РИ - **0,0057**

ИР - 0,0017 РП - 0,0001 ИП - 0,0007

Із даного аналізу бачимо, що найбільш можливими є сполучення: ПР, РИ. Тому перший рядок, ймовірно, буде - ПРИПИ. Важливо визначити прядок букв, які повторюються, наприклад:

И	П	П	Р	И
5	1	4	2	3

Переставимо стовпці згідно цього припущення:

П	Р	И	П	И
Н	І	Т	Ь	С
К	У	П	О	В
У	В	А	Т	И
А	К	Ц	І	Ї

У результаті цієї перестановки ми отримали розшифрований текст: **ПРИПИНІТЬ СКУПОВУВАТИ АКЦІЇ**.

12.5.2 Криптоаналіз шифрів за методом простої заміни.

Дешифрування проводиться в 7 етапів:

1) В якості одиниці інформації приймається 1 символ для простої підстановки або 2 символи для шифрування біграм.

2) Будується діаграма розподілу примітивних елементів для шифрованого тексту.

3) Побудована діаграма порівнюється з діаграмою розподілу примітивних елементів стандартного літературного тексту.

4) Вручну або автоматично робиться припущення про відповідність примітивних елементів вихідного тексту і тексту шифровки, причому аналіз починається з елементів, які найчастіше зустрічаються (найбільш високі піки на графіку), а закінчується елементами, які найрідше зустрічаються.

5) Після зворотної заміни примітивних елементів шифрованого тексту на відповідні їм примітивні елементи вихідного тексту, виходить досить читабельний текст вихідного повідомлення.

6) Якщо вихідне повідомлення містить помилки, або зовсім не схоже на зв'язний осмислений текст, то це значить, що ми зробили помилкове припущення про відповідності елементів і слід повернутися до пункту 4.

7) Аналіз закінчується тоді, коли отриманий текст не містить помилок та є осмисленим.

Підстановки на основі простої заміни не приховують статистичні властивості тексту. Тому криптоаналіз шифру простої заміни заснований на використанні статистичних закономірностей мови (в інтернет-джерелах такі таблиці є у вільному доступі). Додатково можна використовувати аналіз к-грам (в основному біграм), враховувати властивості мови (суфікси, префікси, подвоєння, сполучники та інше).

Приклад 1.

Розшифрувати текст, зашифрований за допомогою простої заміни:

29 15 10 17 29 22 25 31 15 33 35 41 43 45 35 57 45 25 17 59 15 10 25
41 25 69, 59 78 29 82 25 78 25 17 15 10 88 90 78 25 62 25 22 10 57 73 79 35
67 78 90 88 29 45 35 29, 54 57 90 31 90 73 22 88 15 88 29 15 17 69 41 25 15,
70 17 90 57 43 59 15 78 15 62 22 25 17 57 25 69 88 15 82 17 25 88 29 45 35...

Підрахуємо частоту кожного слова у шифровці:

29	15	1 0	1 7	22	25	31	33	35	41	43	45	57	59	69	78	82	88	90	62	73	79	67	54	70
7	10	4	7	4	12	2	1	5	3	2	4	5	3	3	4	2	6	5	1	2	1	1	1	1

Із таблиці частот російської мови мови (в інтернет-джерелах такі таблиці є у вільному доступі) найчастіше зустрічається буква «о» та «е». Тому припустимо що 25=«о», 15=«е» (може бути і навпаки). Слово у третьому рядку перед другою комою може закінчуватися на «ое» або «ео». Із таблиці біграм бачимо, що найчастіше зустрічається «ое» - частота 15, а у «ео» - 7. Тому наші припущення вірні: **25=«о», 15=«е».**

Тепер оцінимо букву 29 перед першою комою у третьому рядку, швидше за все це може бути голосні буква. Так як вона зустрічається 7 разів у тесті, що є середньою величиною, то це може бути буква «а» або «и». Тоді в останньому рядку 25 88 29, 88 - приголосна буква, згідно частот, ймовірно «т» або «н». Отже, у третьому рядку 22 88 15 88 29 15 маємо: *22 (т, н) е (т, н) (а, и) е* - найочевидніше - «мнение». Тому **22=«м», 88=«н», 29=«и».**

У першому рядку 29 15 10 17 29 22 маємо *и е 10 17 и м о*. Робимо

висновки, що 10 і 17 приголосні, згідно таблиці частот це можуть бути «с» та «л». Звідси слово - «если». Тому **10=«с», 17=«л».**

У другому рядку 59 78 29 82 25 - *58 78 и 82 о.* 58 і78 можливо

приголосні, наприклад - «при». **58=«п», 78=«р».**

У першому рядку 45 25 17 59 15 - *45 о л п е*. 45 може бути «т», «в», «з», «б», «г», «ч». Із усіх випадків підходить «т» - *толпе*. **45=«т».**

12.5.3. Криптоаналіз шифру Цезаря

Криптосистема Цезаря визначається рівнянням $y = (x_г + к)$тод m, де $i = 1, n$, $У_I$ - порядок букви в алфавіті криптограми, X_I - порядок букви в алфавіті відкритого тексту, $к$ - ключ шифру, який може змінюватися від *1* до m, n - довжина криптограми, m - потужність алфавіту (кількість букв).

Ключ можна знайти за допомогою декількох методів:

- Частотного аналізу, знайти букву, яка найчастіше зустрічається у шифрі і порівняти відстань від неї до букви, яка найчастіше зустрічається у алфавіті відповідної мови.
- Здійснити атаку повним перебором на множині всіх можливих ключів.
- Обчислити ключ за допомогою формули:

$k^* = \arg\max_k l(k),\ l(k) = \sum_{j=0}^{m-1} v_{(j-k) mod\ m} \cdot \log_2 p(j)$

v_j- частота j-ої букви у криптограмі, $p(j)$ - ймовірність j-ої букви у алфавіті.

Приклад 3.

Розшифрувати криптограму зашифровану за допомогою шифру Цезаря.

«pelcgbybtlvfpelcgbtenculnaqpelcgbnanylfvf»

Перетворимо її у числову, нумерація букв починається із нуля:

«15 4 11 2 6 1 24 1 19 11 21 5 15 4 11 2 6 1 19 4 13 2 20 11 12 0 16 15 4 11 2 6 1 13 0 13 24 11 5 21 5».

Складемо таблицю ймовірнісних характеристик англійського алфавіту

буква	a	b	c	d	e	f	g	h	i	j	k	l	m	n
j	0	1	2	3	4	5	6	7	8	9	10	11	12	13
$Log(p_j)$	-3.6	-6	-5	-4.7	-3	-5.4	-5.7	-4.2	-3.8	-9.3	-7.2	-4.6	-5.3	-3.8
буква	o	P	q	r	s	t	u	v	w	x	y	z		
j	14	15	16	17	18	19	20	21	22	23	24	25		
$Log(p_j)$	-3.7	-5.6	-9.8	-4	-3.9	-3.4	-5.2	-6.7	-5.7	-9	-5.9	-10.1		

Знайдемо частотні характеристики криптограми

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

vj	2	4	4	0	4	3	3	0	0	0	0	6	0	4
буква	o	P	q	r	s	t	u	v	w	x	y	z		
j	14	15	16	17	18	19	20	21	22	23	24	25		
vj	0	3	1	0	0	2	1	2	0	0	2	0		

Обчисливши та побудувавши графік логарифмічної функції правдоподібності *l(k)*, знайдемо ключ рівний 13.

k	0	1	2	3	4	5	6	7	8
L(k)	-203,5	-230,8	-193,4	-219,8	-234,2	-258,6	-207,3	-211,2	-235,2
k					12	**13**	14	15	
L(k)		....			-260,2	**-185,4**	-253,5	-211,8	

Зсунемо всі букви криптограми на 13 позицій і отримаємо вихідний текст: cryptology is cryptography and cryptoanalysis

13 Сучасні напрямки розвитку криптографії

13.1. Постквантові криптографічні системи

Квантові обчислення здатні вирішувати складні математичні проблеми, на яких базується велика кількість алгоритмів. Тому для забезпечення безпеки від квантових атак, необхідно використовувати криптографічні системи, які є стійкими до таких атак, тобто квантово-стійкі. Ці системи використовують алгоритми, які базуються на математичних проблемах, для яких не існує ефективних алгоритмів криптоаналізу. Існує декілька типів квантово-стійких алгоритмів, які можуть бути використані для розробки криптографічних систем, стійких до атак з використанням квантових обчислень (як і для атак за допомогою звичайних комп'ютерів), а саме :

- Lattice-based cryptography - криптографія на основі ґраток;
- Multivariate Cryptography - мультиваріативна криптографія;
- Hash-based cryptography - криптографія на основі хеш-функцій;
- Code-based cryptography - криптографія на основі кодів коригування помилок;
- Isogeny-based cryptography - криптографія на основі ізогенії;
- Symmetric key cryptography - симетрична криптографія (за умовою використання ключа великого розміру стійка до квантових алгоритмів).

Інформацію про алгоритми було взято за результатом аналізу раундів по відбору, аналізу та стандартизації квантовостійких алгоритмів, який проводиться національною організацією зі стандартизації у США NIST (National Institute of Standards and Technology) . Один із важливих проектів, які веде ця організація є підпроєкт PQC (Post-Quantum Cryptography) . У межах проекту PQC, NIST проводить конкурси з аналізу раундів з відбору, дослідження та стандартизації квантовостійких алгоритмів.

NIST збирає пропозиції щодо постквантових криптосистем (фахівці можуть запропонувати свій алгоритм) і коментарі від громадськості в рамках процесу оцінки, плануючи провести кілька раундів відбору для отримання найкращого результату. Метою цього процесу є вибір ряду прийнятних криптосистем- кандидатів для стандартизації. На теперішній час організація провела три раунди відбору, наразі триває 4-й раунд. Під час оцінки кандидатів, проводиться ретельний аналіз поданих алгоритмів відкритим і прозорим для громадськості способом. Криптографічна спільнота заохочується проводити аналізи та оцінку зі свого боку, що створює неупередженість при оцінюванні та збільшує кількість досліджень як щодо алгоритмів, так і у галузі загалом. NIST заохочує рецензентів продемонструвати свої висновки та можливі практичні атаки як на версії з параметрами, які забезпечують повний рівень безпеки, так і на надані набори параметрів із нижчими рівнями безпеки. Тому для підкреслення значущості алгоритмів після їх розгляду буде наведено відомості про участь/перемогу в номінаціях від NIST.

Існує п'ять рівнів безпеки, якими NIST характеризує алгоритми . Рівень безпеки визначається відповідно до теоретичного часу, необхідного для компрометації криптографічного протоколу за допомогою квантового комп'ютера. Для визначення часу використовуються відповідні атаки: для симетричних блочних шифрів використовується атака, яка порушує відповідне визначення безпеки, повинна вимагати обчислювальних ресурсів, порівнянних або більших, ніж необхідні для пошуку ключів; для хеш-функцій - атака з пошуком колізій. На теперішній час, NIST визначає п’ять рівнів безпеки постквантових криптоалгоритмів наступним чином:

- алгоритм має бути настільки складно скомпрометувати, як і виконати пошук ключів у симетричному блочному шифрі з довжиною ключа у 128 біти (наприклад, AES128);
- має відповідати стійкості 256-бітної хеш-функції (наприклад, SHA256);
- має відповідати стійкості симетричного шифру з ключем довжиною 192 біти (наприклад, AES192);
- має відповідати стійкості 384-бітної хеш-функції (наприклад, SHA384);
- найвищий рівень, має забезпечувати захист еквівалентний до симетричного шифрування з ключем довжиною 256 бітів (наприклад, AES256).

13.1.1 Криптографія на основі ґраток

Криптографія на основі ґраток (англ. lattice-based cryptography) - це тип криптографічної схеми, що базується на математичній теорії ґраток, які є типом дискретної математичної структури, яка виникає в різних галузях математики, включаючи теорію чисел, алгебру та геометрію. Ґратка – це дискретний набір точок у -вимірному просторі, які розташовані за регулярним шаблоном. Ці точки можна розглядати як сітку точок, яка нескінченно тягнеться в усіх напрямках. Може використовуватися в різних програмах, включаючи обмін ключами, цифрові підписи та повністю гомоморфне шифрування.

Ґраткова криптографія заснована на обчислювальній складності ґратчастих задач, основою якої є задача найкоротшого вектора (SVP), яка вимагає знайти найкоротший ненульовий вектор у даній ґратці. Надано вхідну ґратку, представлену довільним базисом, і мета зловмисника полягає в тому, щоб знайти найкоротший вектор від початку координат, коли задано основу ґратки (нульовий вектор не працює як відповідь) . Найвідоміші атаки на криптографію на основі ґраток покладаються на класичні алгоритми, які є неефективними та вимагають експоненціально великих ресурсів, а найвідоміший алгоритм має час роботи O*(2n logn)* для точного розв'язку SVP. Іншою важливою проблемою є проблема найближчого вектора (CVP), яка вимагає знайти точку ґратки, найближчу до даного цільового вектора і вирішується в найкращому алгоритмі за $O(2^{3.5n})$. Криптографія на основі ґраток є відносно ефективною у реалізації та має дуже сильні докази безпеки, засновані на складності найгіршого випадку. Незважаючи на значні дослідницькі зусилля, невідомо ефективних квантових алгоритмів для вирішення базових задач алгоритму, які значно перевершують класичні (єдина перевага квантових комп'ютерів у цьому відношенні - це скромне загальне прискорення).

Криптосистеми, засновані на ґратках, часто є алгоритмічно простими, ефективними та високо паралелізованими. Алгоритми, які використовуються в криптографії на основі ґраток, базуються на простих математичних операціях, таких як множення матриць і модульна арифметика, які піддаються апаратній реалізації та оптимізації. Проаналізуємо потенційні квантово-безпечні алгоритми, засновані на ґратках.

CRYSTALS-KYBER - забезпечує безпечний механізм інкапсуляції ключа (KEM) IND-CCA2, безпека якого забезпечується складністю розв'язання задачі навчання з помилками на модульній ґратці (Module-LWE, або M-LWE). Алгоритм обчислювально швидкий (здатен генерувати від 9 тисяч до 90 тисяч ключів за секунду в залежності від рівня безпеки алгоритму) і має невеликий відкритий ключ (від 800 до 1568 байтів - для публічного ключа, від 1632 до 3168 байтів - для закритого). Він має різні параметри, налаштовані відповідно до рівнів безпеки NIST 1, 3 і 5.

NTRU - розроблений у 1996 році математиками Джеффрі Хоффштайном, Джилл Пайфер і Джозефом Х. Сільверманом. У 2013 році Деміен Штеле та Рон Штайнфельд розробили постквантову безпечну версію NTRU. Він заснований на складності розв'язання проблеми кільцевого навчання з помилками (Ring-LWE, або R-LWE). Алгоритм стійкий до атак з використанням алгоритму Шора. Має ще коротший ключ, та співвідношення довжини відкрито/закритого ключа 1/~1.3.

SABER – безпека алгоритму грунтується на складності вирішення проблеми модульного навчання з округленням (Module-LWR). SABER є простим і ефективним алгоритмом . Пропонує три рівні безпеки:

- LightSABER: постквантовий рівень безпеки подібний до AES-128 - 1й рівень NIST;
- SABER: постквантовий рівень безпеки подібний до AES-192 - 3й рівень NIST;
- FireSABER: постквантовий рівень безпеки подібний до AES-256 - 5й рівень NIST.

FrodoKEM – це сімейство алгоритмів, які надають механізмів

інкапсуляції ключів, яке базується на алгебраїчно неструктурованій ґратці. Алгоритм менше обмежений параметрами, але має великий розмір ключа (відкритий ключ - від 9616 до 21520 байтів, приватний - від 19888 до 43088). Забезпечує три рівня безпеки:

- FrodoKEM-640: 1й рівень NIST;
- FrodoKEM-976: 3й рівень NIST;

- FrodoKEM-1344: 5й рівень NIST.

CRYSTALS-DILITHIUM - базується на техніці «Fiat-Shamir with Aborts» Любашевського . Має найменший відкритий ключ і розмір підпису серед будь- яких схем підпису на основі ґраток, яка використовує лише рівномірну вибірку, та показує гарну швидкодію (кількість цифрових підписів, які алгоритм може створювати за секунду, варіюються від 2 до 15 тисяч, та здатен перевірити від 3 до 45 тисяч підписів за секунду).

FALCON - це схема підпису на основі ґратки. Безпека алгоритму базується на проблемі коротких інтегральних розв'язків (SIS) над ґратками NTRU, для якої в даний час не відомий ефективний алгоритм вирішення в загальному випадку, навіть за допомогою квантових комп'ютерів .

Максимальна кількість алгоритмів, оголошених NIST у 3-му раунді, належить до сімейства криптографії на основі ґраток. Три алгоритми – CRYSTALS-KYBER, NTRU та SABRE, оголошені фіналістами Key Encapsulation Mechanisms; CRYSTALS-DILITHIUM і FALCON оголошені фіналістами цифрових підписів, FrodoKEM був оголошений альтернативним кандидатом на механізм інкапсуляції ключів . Алгоритм CRYSTALS-KYBER також було оголошено переможцем в категорії Key Encapsulation Mechanisms за результатами 2022 року, CRYSTALS-DILITHIUM та FALCON - в категорії цифрових підписів . Більш детальне порівняння за характеристиками в порівнянні з іншими алгоритмами наведено в таблицях 1-2.

13.1.2 Мультиваріативна криптографія

Мультиваріативна криптографія (англ. multivariate cryptography, MC) – це тип криптографії з відкритим ключем, яка базується на складності розв'язування систем багатовимірних поліноміальних рівнянь над кінцевими полями. У деяких випадках ці поліноми можуть бути визначені як над основним, так і над полем розширення. Якщо поліноми мають другий ступінь, то говорять про багатовимірні квадрати. Доведено, що розв'язування систем багатовимірних поліноміальних рівнянь є NP-трудоємною задачею. Однак безпека MC вважається вразливою до алгебраїчних, диференціальних атак та атак на базис Грьобнера . Як наслідок, MC не має широкого використання на практиці, а його застосування обмежене.

У MC відкритий ключ - це система рівнянь багатовимірного полінома, а закритий ключ - це знання коефіцієнтів цих поліномів. Щоб зашифрувати повідомлення, відправник перетворює відкритий текст на систему поліноміальних рівнянь і розв'язує їх за допомогою відкритого ключа. Щоб розшифрувати повідомлення, одержувач використовує свої знання закритого ключа, щоб розв'язати ту саму систему рівнянь. Проаналізуємо потенційні квантово-безпечні алгоритми на основі мультиваріативної криптографії.

Rainbow - розроблений Jintai Ding і Dieter Schmidt у 2004 році. Алгоритм заснований на схемі підпису «UOV» (Unbalanced Oil and Vinegar) , яку винайшов Жак Патарен. Алгоритм базується на тому факті, що розв'язування набору випадкових багатовимірних квадратичних систем є NP-складною задачею. Алгоритм можна використовувати для цифрового підпису, а його параметри можна встановити для досягнення рівнів безпеки NIST 1, 3 і 5.

GeMSS (A Great Multivariate Short Signature) - це схема підпису, що створює маленькі підписи. GeMSS базується на криптосистемі Hidden Field Equations (HFE) з використанням модифікаторів, наприклад HFEv- . Алгоритм має середній/великий відкритий ключ і теоретично дуже швидкий процес перевірки.

Rainbow оголошено фіналістом 3 раунду NIST у категорії цифрових підписів, а GeMSS оголошено альтернативним кандидатом для алгоритму цифрового підпису . Незважаючи на свої обмеження, MC було запропоновано як потенційний постквантовий криптографічний алгоритм через його стійкість до атак квантових комп'ютерів. Однак для підвищення ефективності та безпеки MC для практичного використання необхідні подальші дослідження. Більш детальне порівняння за характеристиками в порівнянні з іншими алгоритмами цифрового підпису наведено в таблиці 2.

13.1.3 Криптографія на основі хешування

Криптографія на основі хешування (англ. hash-based cryptography) – це тип постквантової криптографії, яка покладається на математичні властивості хеш- функцій для забезпечення безпеки. Основна ідея полягає у використанні односторонньої хеш-функції для створення дайджесту повідомлення або хеш- значення з вихідного повідомлення. Потім це хеш-значення використовується як цифровий підпис або ключ для шифрування та дешифрування.

Основна перевага криптографії на основі хешування полягає в тому, що це добре зрозуміла, вивчена і широко використовувана технологія, яка гарантує високу стійкість до квантових атак, що робить її кандидатом на довгострокову безпеку в постквантову еру (за умови використання достатньо довгого ключа та/або дайджеста).

Однак криптографія на основі хешування також має деякі потенційні проблеми. По-перше, безпека хеш-функції може бути порушена, якщо зловмисники знайдуть колізії, тобто два різних повідомлення, які створюють однакове хеш-значення . По-друге, криптографія на основі хешування вимагає відносно великого розміру ключа (який є результатом обчислення дайджесту повідомлення) для забезпечення безпеки, оскільки алгоритм Гровера можна використовувати для зламу хеш-функції за час $O\sqrt{n)}$ (де n - довжина ключа), коли звичайні алгоритмі мають експоненційну складність O($2^{\frac{n}{2}}$), хоча і ефективний лише для коротких ключів. Це призводить до сповільнення генерації ключа або підпису, збільшення вимог до пам’яті та розміру фінального повідомлення для транспортування.

Криптографія на основі хешування широко використовується для цифрових підписів. Підпис на основі хешу (HBS) – це набір багатьох схем одноразового підпису. HBS використовує структуру даних дерева для ефективного об’єднання багатьох одноразових підписів. Щоб підписати повідомлення, HBS вибирає один із одноразових підписів зі своєї колекції та використовує його для підпису повідомлення (алгоритм ніколи не повинен використовувати один і той самий одноразовий підпис з колекції двічі, інакше існує загроза, що безпеку буде порушено) . Проаналізуємо квантово-безпечні алгоритми на основі хешування.

SPHINCS+ - це схема підпису без збереження стану на основі хешу. Це імпровізована версія SPHINCS , спеціально спрямована на зменшення розміру підпису. NIST оголосив SPHINCS+ альтернативним кандидатом для алгоритму цифрового підпису в 3-му раунді . Може використовуватися для цифрових підписів і задовольняє рівням безпеки NIST 1, 3 і 5, пропонуючи три різні схеми підпису:

- SPHINCS+-SHA3 (з використанням хеш-функції SHAKE256);
- SPHINCS+-SHA2 (з використанням SHA2);
- SPHINCS+-Haraka (з використанням Haraka).

Криптографія на основі хешування залишається багатообіцяючим і широко використовуваним методом для забезпечення безпечних цифрових підписів і автентифікації в різних програмах. Складність рішення та потенційні проблеми, пов’язані з криптографією на основі хешування, залежатимуть від конкретної реалізації та варіанту використання. Більш детальне порівняння за характеристиками в порівнянні з іншими алгоритмами цифрового підпису наведено в таблиці 2.

13.1.4 Криптографія на основі кодів коригування помилок

Криптографія на основі кодів коригування помилок (англ. code-based cryptography) – покладається на складність задачі декодування лінійного коду з виправленням помилок, який, можливо, вибрано з певною структурою або в певному сімействі (наприклад, квазіциклічні коди або коди Гоппи). Безпека базується на складності проблеми декодування, яка вимагає пошуку вектора помилки, який був введений при передаванні кодового слова. Вважається, що ця проблема складна з точки зору обчислень (складність рішення декодування кодів з виправленням помилок залишається NP-трудоємною, проте точна складність є активною областю досліджень), що робить криптографію на основі коду перспективним кандидатом на постквантову криптографію.

Класичним представником даної системи вважається система McEliece –

криптосистема з відкритим ключем на основі бінарного кода Гоппи. Classic McEliece , засновану на коді постквантову криптосистему з відкритим ключем, яка претендує на стандартизацію NIST, запропонував Даніель Дж. Бернштейн у 2017 році. Згідно з ідеєю МакЕліса, відкритий ключ є продуктом коду Гоппи та лінійного перетворення. Щоб зашифрувати повідомлення, відправник повинен додати певну кількість випадкового «шуму» до повідомлення, який можна видалити лише за допомогою коду Гоппи . Відновити повідомлення, не знаючи, як врахувати відкритий ключ, для зловмисника є складною обчислювальною проблемою.

Одна з потенційних проблем криптографії на основі коду полягає в тому, що відкритий ключ може бути вразливим до алгебраїчних атак, коли зловмисник може отримати інформацію про секретний ключ, аналізуючи властивості відкритого. Інша проблема полягає в тому, що криптографія на основі коду може бути вразливою до атак по бічному каналу, коли зловмисник може спостерігати за виконанням алгоритму шифрування або дешифрування та вилучати інформацію про секретний ключ . Також слід зазначити, що криптографія на основі коду вимагає більшого розміру ключа. Це може призвести до довшого часу шифрування та дешифрування, а також може потребувати більше місця для зберігання. Крім того, криптографія на основі коду може бути дорогою з точки зору обчислень, оскільки операції кодування та декодування вимагають великої кількості множень матриць, що може бути дорогим з точки зору обчислювальних ресурсів. Проаналізуємо квантово-безпечні алгоритми на основі кодів коригування помилок.

Classic McEliece - KEM, що відповідає всім п’яти рівням безпеки NIST. Час обчислення алгоритму відносно швидкий, але алгоритм потребує використання дуже великого розміру відкритого ключа (від 0.25 до 1.35 мегабайт, що є дуже великим розміром для ключа, який необхідно передати каналом зв'язку).

BIKE - це механізм інкапсуляції ключів на квазіциклічних кодах перевірки парності помірної щільності (QC-MDPC), які можуть бути декодовані за допомогою методів декодування з перевертанням бітів . Має найкоротший відкритий ключ (від 1541 до 5122 байтів) та високу швидкість, може задовольнити 1-му, 3-му і 5-му рівням безпеки NIST.

HQC - KEM на основі коду, яка забезпечує безпеку IND-CCA2. Має найкоротший відкритий ключ та співвідношення відкритого/закритого ключа майже 1 (закритий більше на кілька байтів). Алгоритм розроблений для досягнення 1, 3 і 5-го рівнів безпеки NIST .

Classic McEliece оголошено фіналістом 3 раунду NIST у категорії KEM, тоді як BIKE і HQC були оголошені альтернативними кандидатами . Також на четвертий раунд NIST, де організація шукає альтернативи алгоритмам, заснованим на гратках, наразі надійшло 4 заявки. З цих 4-х заявок, 3 відносяться до алгоритмів, заснованих на кодах, які були оглянуті вище . Як і для будь-якого класу криптосистем, криптографія на основі коду є компромісом між безпекою та ефективністю . Більш детальне порівняння за характеристиками в порівнянні з іншими KEM-алгоритмами наведено в таблиці 1.

13.1.5 Криптографія на основі ізогенії

Ізогенія - це математичне відображення мід еліптичними кривими, яке зберігає групову структуру точок на цих кривих. Криптографія на основі ізогенії (англ. isogeny-based cryptography) – відносно нова система, найвідоміший представник – протокол обміну ключами Supersingular Isogeny Diffie-Hellman (SIDH) . Використовує відображення між еліптичними кривими для побудови криптосистем з відкритим ключем. Безпека базується на одній із небагатьох важких математичних задач, яка наразі протистоїть атакам квантових комп’ютерів – на так званих проблемах суперсингулярних ізогеній, або знаходженні відображення ізогенії між двома суперсингулярними еліптичними кривими з однаковою кількістю точок . Протоколи на основі ізогенії вимагають дуже маленького ключа порівняно з будь-яким іншим кандидатом пост-квантової криптографії (однак, все ще набагато більші, ніж для алгоритмів на звичайних еліптичних кривих), але продуктивність і придатність для більш просунутих криптографічних примітивів обмежені порівняно, наприклад, з системами на основі ґраток .

Перший запропонований алгоритм, заснований на ізогеніях, використовував звичайні

еліптичні криві, які спираються на складність задачі дискретного логарифмування еліптичної кривої. Однак після вивчення можливих атак виявилося, що існують субекспоненціальні атаки проти цього алгоритму з використанням квантових комп'ютерів (атаку виявили А. Чайлдс, Д. Джао та В. Сухарєв).

Однією з потенційних проблем криптографії на основі ізогенії є те, що це відносно нова та неперевірена криптосистема, а це означає, що можуть бути невиявлені вразливості або слабкі місця, якими можуть скористатися зловмисники. Крім того, хоча розміри ключів, необхідні для криптографії на основі ізогенії, невеликі, обчислювальна вартість генерації ключів все ще відносно висока, що може бути недоліком у деяких середовищах з обмеженими ресурсами.

Проаналізуємо алгоритм на основі ізогенії зі списку кандидатів NIST.

SIKE - алгоритм інкапсуляції ключів на основі ізогенії. Реалізація алгоритму базується на псевдовипадкових блуканнях у суперсингулярних графах ізогенії. Має найменший розмір відкритого ключа (від 197 до 564 байтів для відкритого ключа, від 28 до 644 байтів - для закритого), але також найповільніший з оглянутих алгоритмів . Нещодавно був скомпрометований та вже не може бути рекомендованим до використання (алгоритм Magma порушує екземпляр SIKEp434, який націлений на 1-й рівень безпеки NIST, приблизно за одну годину на одному ядрі).

SIKE оголошено альтернативним кандидатом KEM у 3-му раунді NIST . Був запропонований як кандидат на 4-й раунд NIST , проте через знаходження атаки з нотаткою, що алгоритм був скомпрометований, щоб остаточна версія пропозиції SIKE точно відображала поточний стан криптосистеми.

Криптосистеми на основі ізогенії демонструють великі перспективи для забезпечення безпечної та ефективної криптографії з відкритим ключем. А зважаючи на те, що національний стандарт України для цифрових підписів ґрунтується на використанні еліптичних кривих (ДСТУ 4145-2002) , цей напрямок є дуже важливим та має перспективу для подальшого використання в Україні. Більш детальне порівняння за характеристиками в порівнянні з іншими KEM-алгоритмами наведено в таблиці 1.

13.2 Порівняльний аналіз квантовостійких криптографічних алгоритмів

Для того, щоб скомпрометувати поточні криптографічні алгоритми, необхідний квантовий комп'ютер з мільйонами "кубітів". Наразі невідомо, коли такі комп'ютери стануть доступні. Але експоненційний розвиток квантових комп'ютерних технологій свідчить про те, що це може статися дуже швидко, і ми не можемо ігнорувати зростаючу загрозу з їх боку (наприклад, у 2019 році IBM створив квантовий процесор з 27 кубітами, а у 2021 році - досягли вже відмітки у 127 кубітів, що є надзвичайним прогресом за два роки) .

Тому при розробці власного протоколу слід обирати постквантові алгоритми, які захищені від потенційних атак з використанням квантових обчислень, для захисту користувачів. Для цього необхідно обрати конкретні криптографічні алгоритми, які будуть використовуватися під час розробки власного мережевого протоколу. Усього необхідно три криптографічні примітиви, на основі яких буде проведено розподіл та порівняння оглянутих квантовостійких алгоритмів:

- алгоритм для отримання спільного секретного ключа - забезпечує можливість узгодити єдиний секретний ключ, який буде використовуватися для симетричного шифрування та дешифрування даних через відкритий канал зв'язку;
- алгоритм автентифікації сторін - дозволяє встановити, що сторони є тими, за кого себе видають, та взаємодія з ним(и) є передбачуваною;
- симетричний алгоритм захисту даних - буде використовуватися на основному етапі спілкування (після узгодження спільного секретного ключа сеансу та додаткових перевірок за потребою) при передаванні даних для їх захисту через значно кращу продуктивність у порівнянні з асиметричними.

У результаті аналізу існуючих вживаних мережевих протоколів у першому розділі, прийнято рішення у розробці нового протоколу для постквантової криптографії, базуватися на принципах TLS - передбачається, що його можна буде нескладно розширити. Тому, при

аналізі примітивів можна створити пріоритетний список або виділити кандидатів з кожної категорії для подальшого використання або хоча б розуміння, що в майбутньому їх можливо додати і чому це сталося. На теперішній час достатньо обрати по одному алгоритму для кожного примітиву.

Аналіз продуктивності квантово-стійких алгоритмів проведено за допомогою проекту Open Quantum Safe (OQS) , який розробляє та прототипує такі алгоритми. Бібліотека liboqs , розроблена OQS, містить відкритий код для квантово-стійких криптографічних алгоритмів. Основна мета OQS - стандартизація постквантової криптографії NIST для механізмів шифрування ключів (KEM) і схем підпису. Також OQS надає дані порівняльного аналізу для різних квантово-стійких алгоритмів, зокрема, щодо їх продуктивності та споживання пам'яті . Використання офіційних даних швидкодії гарантує оминання похибки, щодо інших критеріїв - вони будуть сформовані відповідно до алгоритмів.

13.2.1 Алгоритм для отримання спільного секретного ключа

Алгоритм обміну ключами дозволяє двом сторонам, які ніколи не зустрічалися, домовитися про спільний ключ. Навіть якщо хтось стежить за каналом зв'язку, він не може визначити узгоджений ключ сеансу. KEM-алгоритми складаються з трьох головних частин, загальний принцип роботи наведено на рисунку 13.1.

Key encapsulation mechanism (KEM)

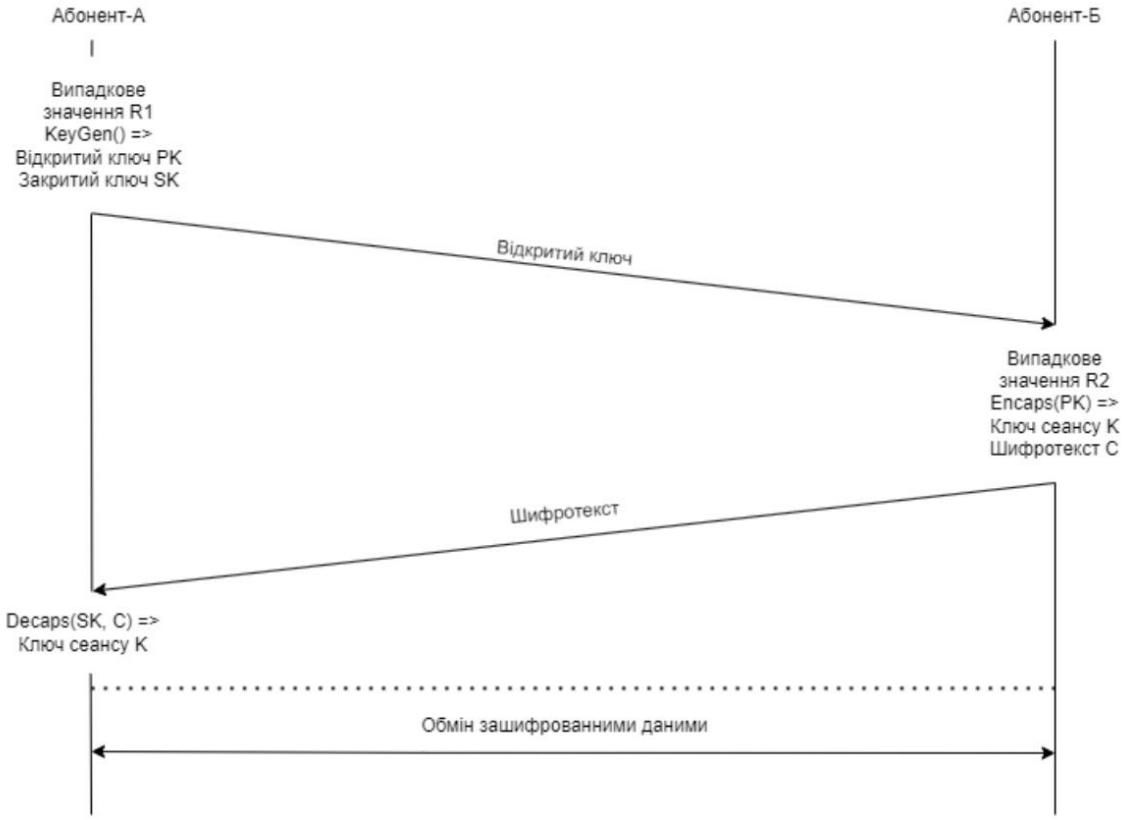


Рисунок 13.1 – Загальний принцип роботи KEM-алгоритмів

Крок 1: операція KeyGen() - дозволяє отримати пару відкритий-закритий ключ («PK/SK») за допомогою випадкового вхідного значення («R»). Закритий ключ залишається у Абонента-А, а відкритий ключ - передається в незахищеному вигляді мережею Абоненту-Б.

Крок 2: операція Encaps(PK) - дозволяє Абоненту-Б отримати ключ сеансу («K») та випадковий шифротекст («C»), використовуючи відкритий ключ («PK») Абонента-А та випадкове вхідне значення («R2»). Для цього використовується операція Encaps(PK), яка гарантує узгодження ключа сеансу для однієї сторони, та створює необхідні вхідні дані для другої - Абонент-Б надсилає Абоненту-А згенерований шифротекст.

Крок 3: операція Decaps(SK, C) - дозволяє Абоненту-А отримати ключ сеансу («K») за допомогою шифротексту («C»), який був згенерований Абонентом-Б, використовуючи відкритий ключ («PK»). Для цього необхідно застосувати операцію Decaps(SK, C), яка

приймає закритий ключ та шифротекст, в результаті генеруючи ключ сеансу, який співпадає з отриманим Абонентом-Б на 2-му кроці.

Крок 4: відтепер, абоненти можуть переходити на симетричне шифрування, використовуючи узгоджений ключ сеансу - встановлено захищене з'єднання.

Порівняння алгоритмів для вибору найбільш підходящого для реалізації у власному протоколі є важливою частиною аналізу. Для цього проаналізуємо розглянуті вище у пунктах 1.1-1.5 алгоритми. Значущим фактором є швидкодія, так як постквантова криптографія у загальному випадку повільніша, ніж звичайна (оскільки ключі мають більші розміри та складність обчислень). Тому для вибору оптимального рішення з оглянутих постквантових алгоритмів слід означити критерії для їх порівняння.

Довжина відкритого/закритого ключів (байт) - необхідна для розуміння складності алгоритму, так як має прямий вплив (при ключі, довжиною n біт, в найгіршому випадку необхідно 2^n операцій для знаходження ключа). Також надає інформацію про те, яку кількість даних необхідно передати мережі (для відкритого ключа) та впливає на час, необхідний алгоритму для генерації пар ключів.

Час генерації пари ключів (од/с) - визначає, скільки алгоритм здатен генерувати пар відкритого/закритого ключів за секунду часу. Критерій відображає частину загальної швидкодії.

Encaps (од/с) - кількість ключів сеансу («К») та випадкового шифротексту («С»), які можуть бути отримані за допомогою відкритого ключа («РК») за секунду часу, використовуючи Encaps(PK) операцію. Критерій відображає частину загальної швидкодії.

Decaps (од/с) - кількість ключів сеансу («К»), які можуть бути отримані використовуючи шифротекст («С») та закритий ключ («SK») за допомогою Decap(SK, C) операції за секунду часу. Критерій відображає частину загальної швидкодії.

Узагальнений коефіцієнт швидкодії (0-10) - оцінка швидкодії алгоритму, яка представляє собою комбінацію попередніх трьох параметрів (генерація пари ключів, операції Encaps та Decaps) і вираховується за наступною формулою:

$$\text{УКШ} = \left(\frac{\text{ГПК}}{10000} + \frac{Encaps}{10000} + \frac{Decaps}{10000}\right)/3 = \frac{\text{ГПК} + Encaps + Decaps}{30000}$$

Нижня межа - 0, верхня - 10, так як кожен окремий критерій алгоритму демонструє максимальну швидкість не перевищуючи 100.000 операцій за секунду, то окрема частина коефіцієнту швидкодії вимірюється у межах 0-10, загальний коефіцієн - як сумма частин, поділенний на кількість цих частин. Результат підрахунку коефіцієнта округлений до сотих.

Таблиця 13.1 – Порівняльні характеристики KEM-алгоритмів

Алгоритм і рівень безпеки NIST	Відкрити й ключ (байт)	Закрити й ключ (байт)	Генераці я пари ключів (од/с)	Encaps (од/с)	Decaps (од/с)	Узагальнени й коефіцієнт швидкодії (0-10)
CRYSTALS -KYBER (5)	1568	3168	71094	61995	96649.33	7.66
NTRU (5)	1230	1590	2437.33	47725.33	44142.33	3.14
SABER (5)	1312	3040	38432.1	30412.7	31734.6	3.35
FrodoKEM (5)	21520	43088	1025.22	773.33	795.07	0.09
Classic McEliece (5)	1047319	13908	2.28	19568	6900	0.88
BIKE (5)	5122	16494	559.48	4996.67	227.11	0.19
HQC (5)	7245	7258	5566	2982.67	1792.67	0.34
SIKE (5)	564	48	104.79	50.17	127.4	0.01

За обраними критеріями оцінки, проведено порівняльний аналіз (результат наведено в таблиці 1). Для порівняння використано алгоритми, які забезпечують вищій наявний рівень безпеки NIST (якщо декілька модифікацій алгоритму забезпечують цей рівень захисту - обрано кращий).

За результатом порівняння можна зробити висновок, що кращим кандидатом є CRYSTAL-KYBER, оскільки він має найкращу швидкодію та прийнятну довжину ключа. Також, якщо необхідно впровадити альтернативні алгоритми, можна рекомендувати використання алгоритмів SABER та NTRU, які демонструють гарні результати: SABER має гарну швидкодію, NTRU - найкращий розмір ключа (не беручи до уваги SIKE через компрометацію) та середню швидкодію. При виборі алгоритмів, які не базуються на ґратках, слід обрати HQC або BIKE. Алгоритм Classic McEliece демонструє середню швидкодію та має надзвичайно великі ключі (довжина публічного ключа більше одного мільйона байт, що створює додаткове навантаження на канал зв'язку та на час генерації пари ключів, проте не надає явної переваги, окрім захисту від атаки з використанням повного перебору ключів).

На базі порівняння з таблиці 1, кращим варіантом для подальшої розробки є CRYSTAL-KYBER, тому його буде першочергово реалізовано у протоколі як KEM-алгоритм.

13.2.2 Алгоритм автентифікації сторін

Алгоритм автентифікації повинен підтвердити ідентичність однієї зі сторін зв'язку. Одним із можливих підходів є використання SSL сертифікатів, однак цей підхід вимагає додаткової сторони - центру сертифікації, до якого клієнт буде зобов'язаний звернутися, використовуючи сертифікат, отриманий від сервера. Це додає додатковий крок у загальну схему встановлення з'єднання та вимагає додаткової конфігурації (як отримання сертифікату сервером, так і взаємодія клієнта з центром сертифікації).

Алгоритм цифрового підпису також може бути використано для автентифікації сторін у процесі обміну повідомленнями чи даними в мережі. А враховуючи той факт, що раніше було оглянуто декілька квантовостійких алгоритмів цифрового підпису, їх використання є доцільним. Для автентифікації необхідно застосувати наведені нижче дії (загальний принцип роботи алгоритму цифрового підпису наведено на рисунку 13.2).

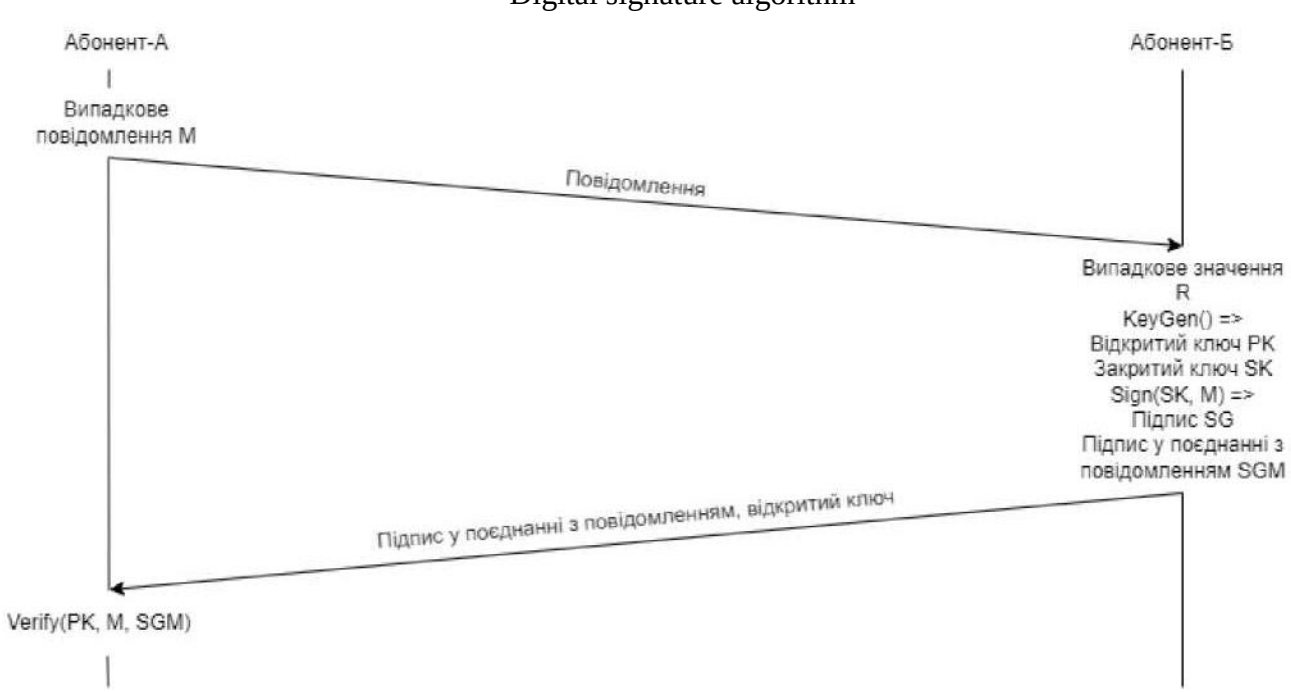


Рисунок 13.2 – Загальний принцип роботи алгоритму цифрового підпису

Крок 1: абонент-А генерує випадкове повідомлення («M») довільної довжини та надсилає його у незахищеному вигляді мережею Абоненту-Б.

Крок 2: абонент-Б генерує пару відкритий-закритий ключ («PK^K») за допомогою випадкового вхідного значення («R»), після чого за допомогою закритого ключа («SK») та отриманого повідомлення («M») створює цифровий підпис («SG»), або виконує операцію підписування. Абонент-Б передає згенерований підпис («SG») у поєднанні з повідомленням («M») - «SGM», та згенерований відкритий ключ («PK»).

Крок 3: абонент-А використовує отриманий відкритий ключ («РК»), цифровий підпис («SG») та повідомлення («М») для перевірки, чи інша сторона дійсно підписала відповідне повідомлення та чи дійсно згенеровано валідний підпис. Для цього використовується операція перевірки підпису, яка забезпечує автентифікацію сторін.

Під час використання у протоколі, у незахищеному вигляді передано буде тільки випадкове повідомлення («М»). Відповідь від Абонента-Б вже буде передано у зашифрованому вигляді, використовуючи ключ сеансу, узгоджений за допомогою обраного KEM-алгоритму, що дозволяє підвищити безпеку та затруднити доступ до публічного ключа, який використовується алгоритмом, та зменшити необхідну кількість кроків для встановлення з'єднання.

Порівняємо розглянуті раніше алгоритми цифрового підпису для вибору конкретного варіанту для реалізації у протоколі. Як і для алгоритмів отримання спільного секретного ключа, необхідно сформулювати критерії для порівняльного аналізу. Загальні критерії, як розмір відкритого/закритого ключів та швидкість їх генерації алгоритмом, залишаються, також введені наступні, специфічні для даного типу алгоритмів, критерії.

Sign (од/с) - кількість операцій підписування, яку алгоритм може опрацювати за одну секунду (використовуючи закритий ключ та випадкове повідомлення). Цей критерій відображає частину загальної швидкості алгоритму.

Verify (од/с) - кількість підписів, яку алгоритм може перевірити за одну секунду (використовуючи відкритий ключ та цифровий підпис, в результаті отримуючи повідомлення). Цей критерій відображає частину загальної швидкості алгоритму.

Узагальнений коефіцієнт швидкодії (0-10) - оцінка швидкодії алгоритму, яка представляє собою комбінацію попередніх трьох спецефічних параметрів (генерація пари ключів, операції Sign та Verify) і вираховується за наступною формулою:

$$\text{УКШ} = \left(\frac{\text{ГПК}}{2000} + \frac{Sign}{2000} + \frac{Verify}{2000}\right)/3 = \frac{\text{ГПК} + Sig + Verify}{6000}$$

Нижня та верхня межі коефіцієнту коефіцієнту залишаються в діапазоні від 0 до 10 як і для KEM-алгоритмів (максимальна сума окремих показників знаходиться в межах 60.000, тому саме 60.000 - прийнято за еталоне значення - 10).

Результат підрахунку коефіцієнта округлений до сотих.

Таблиця 13.2 – Порівняльні характеристики DS-алгоритмів

Алгоритм і рівень безпеки NIST	Відкрит ий ключ (байт)	Закрити й ключ (байт)	Генераці я пари ключів (од/с)	Sign (од/с)	Verify (од/с)	Узагальнени й коефіцієнт швидкодії (0-10)
CRYSTALS-DILITHIUM (5)	2592	4864	21586.33	9008.67	21314.33	8.65
FALCON (5)	1793	2305	42.09	1542.82	8759.33	1.72
Rainbow (5)	536136	1408736	0.18	23.82	24.48	0.01
GeMSS (5)	352180	32				
SPHINCS+ (5)	64	128	955.67	44.22	1815	0.47

У цій категорії кращим є CRYSTALS-DILITHIUM, алгоритм з прийнятною довжиною ключа, який демонструє найкращу продуктивність. У якості альтернативи може виступати FALCON - має коротший ключ, проте помітно гіршу продуктивність, та SPHINCS+ - має найкоротшу пару ключів та велику кількість реалізацій, проте й найповільніший (за операціями підписування/перевірки підпису) з трьох алгоритмів. GeMSS наразі не реалізовано у проекті OQS, тому точних даних щодо його продуктивності не існує, хоча за теоретичними даними повинен мати прийнятну швидкодію на рівні алгоритму FALCON.

На базі порівняння з таблиці 2, кращим варіантом для подальшого використання є CRYSTAL-DILITHIUM, тому саме його буде реалізовано у протоколі для автентифікації сторін.

13.2.3 Симетричний алгоритм захисту даних

Після узгодження спільного закритого ключа та автентифікації сторін з'єдняння, клієнт

та сервер переходять на використання симетричного шифрування для забезпечення шифрування даних під час взаємодії (або раніше, як для забезпечення приватності відкритого ключа та підпису, необхідних для автентифікації). Симетричне шифрування захищене від атак за допомогою як звичайних так і квантових комп'ютерів (за умовою використання достатньо великого ключа). Вирізняють дві підкатегорії - блокові та поточні.

Обидві сторони використовують єдиний спільний ключ як для шифрування, так і дешифрування повідомлень. Головною слабкістю даного підходу є використання єдиного ключа, тому при його компрометації з'язок перестає бути конфеденційним. Загальний принцип роботи симетричного алгоритму наведено на рисунку 13.3.

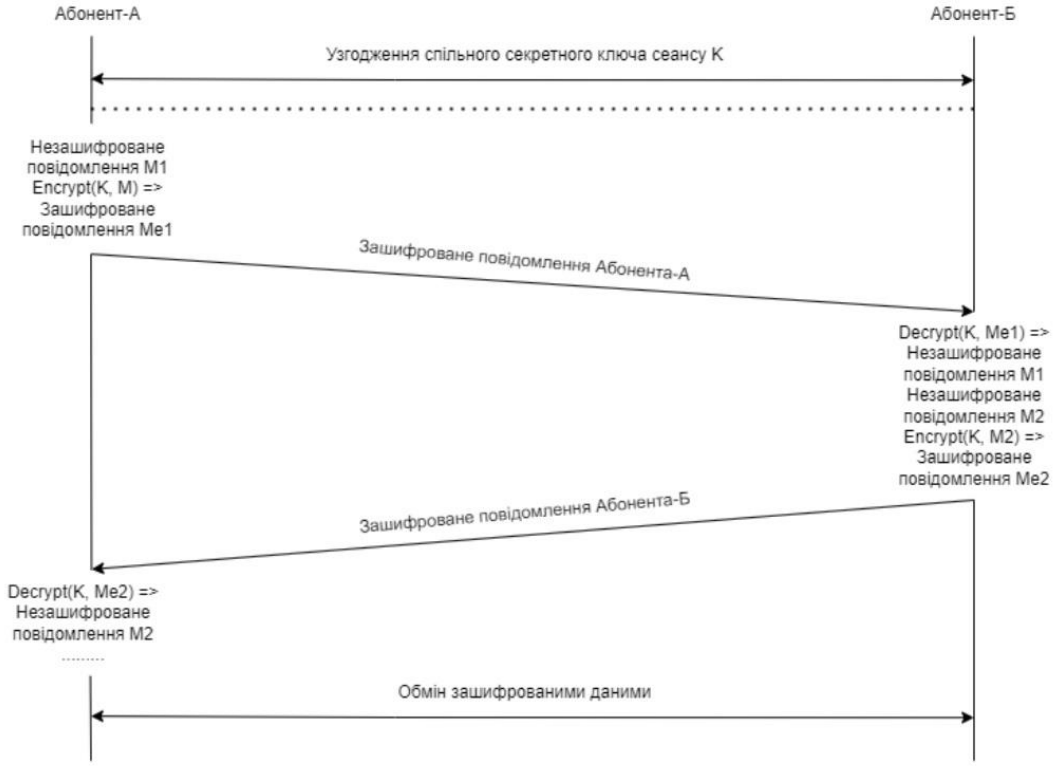


Рисунок 13.3 – Загальний принцип роботи симетричного алгоритму

Вибір симетричного алгоритму залежить від конкретної задачі та потреб користувача. При виборі алгоритму слід керуватися наступними критеріями:

- підтримуваний розмір ключа (біт): чим довший ключ, тим вищий рівень безпеки, але в той же час менша продуктивність та складність управління ключами;
- розмір блоку (біт): чим більший розмір блоку, тим швидше можна шифрувати великі обсяги даних, але при цьому може збільшуватися розмір шифротексту;
- швидкодія (мс): алгоритм має бути ефективним та маловитратним з точки зору обчислювальних ресурсів (наведено результат для випадкових вхідних даних довжиною в 256 та 1024 байти);
- використання пам'яті (Кб): кількість пам'яті яку використовує алгоритм для шифрування/дешифрування даних, чим менше - тим краще (наведено результат для випадкових вхідних даних довжиною в 256 та 1024 байти).

Враховуючи те, що за специфікацією NIST 5й рівень криптостійкості розрахований на використання симетричного алгоритму з використанням 256 бітного ключа, відомості в таблиці 3 наведено для алгоритмів, з використанням ключа у 256 бітів. Задача для заміру швидкодії - створення екземпляру класу, який реалізує відповідний алгоритм, шифрування та дешифрування випадкового вхідного значення довжиною у 256 та 1024 байти та перевірка, що дешифроване значення співпадає зі вхідним.

Примітка: для ДСТУ 7624 дані наведено відносно алгортму з різним розміром блоку, так як конфігурації з розміром блоку і у 128 біт і у 256 біт підтримують 256 бітний ключ

(використання блоку розміром 256 біт надає можливість застосовувати ключ у 512 біт, проте це виходить за встановлені рамки та погіршує швидкодію).

Таблиця 13.3 – Порівняльні характеристики симетричних алгоритмів

Алгоритм	Підтримуваний розмір ключа (біт)	Розмір блоку (біт)	Швидкодія (мс)	Використання пам'яті (Кб)
AES	128, 192 та 256	128	0.447 1.77	847.28 3348.9
ДСТУ 7624 (128)	128 та 256	128	0.456 1.801	847.41 3349.02
ДСТУ 7624 (256)	256 та 512	256	0.477 1.797	848.05 3349.67
Blowfish	32-448	64	0.48 1.847	849.49 3351.11
Twofish	128, 192 та 256	128	0.437 1.815	858.15 3359.77

За результатами порівняльного аналізу, можна зробити висновок, що раціональним буде використання AES (Advanced Encryption Standard) як одного з найбільш популярних та ефективних алгоритмів симетричного шифрування. AES забезпечує надійний рівень безпеки при досить високій швидкодії та малому розмірі ключа (128, 192 або 256 біт). Крім того, розмір блоку у AES становить 128 біт, що дозволяє шифрувати великі обсяги даних та зменшує розмір шифротексту (AES та Twofish є одними з найшвидших алгоритмів симетричного шифрування).

AES є широко використовуваним стандартом, тому існує багато наочних прикладів та реалізацій на різних мовах програмування. Крім того, він є підтримуваним і оновлюваним організацією NIST та має відкритий код, що дозволяє спеціалістам з усього світу оцінювати та покращувати його безпеку. Також варто зазначити, що AES є рекомендованим для застосування у державному секторі США , що свідчить про його високу надійність та безпеку. Також його використовують для вимірювання рівнів безпеки постквантових алгоритмів, тому використовуючи саме його, буде легше пристосуватися до використання алгоритмів одного рівня або тих, що не сильно відрізняються один від одного.

Хоча загалом, алгоритм симетричного шифрування має найменший вплив на зв'язність системи та може бути замінений на будь-який інший (наприклад, на затвердженний національним стандартом України ДСТУ 7624:2014 - алгоритм «Калина») за потребою за невеликий час.

РЕКОМЕНДОВАНА ЛІТЕРАТУРА

1. Nigel Cawthorne. Alan Turing: The Enigma Man. – Acturus, 2019. – 128 p.
2. Вишня В. Б. Основи інформаційної безпеки : навч. посібник / В. Б. Вишня, О. С. Гавриш, Е. В. Рижков. Дніпро : Дніпроп. держ.ун-т внутріш. справ, 2020. 128 с.
3. Гребенюк А.М. Основи управління інформаційною безпекою: навч. посібник / А.М. Гребенюк, Л.В. Рибальченко. Дніпро: Дніпроп. держ. унт внутріш. справ, 2020. – 144 с.
4. Інформаційна безпека. Підручник / В. В. Остроухов, М. М. Присяжнюк, О. І. Фармагей, М. М. Чеховська та ін.; під ред. В. В. Остроухова – К.: Видавництво Ліра-К, 2021. – 412 с.
5. Інформаційна безпека/ За ред. Ю. Я. Бобала та І. В. Горбатого, Львівська політехніка, 2019.-540 с.
6. Кібербезпека : сучасні технології захисту. Навчальний посібник для студентів вищих навчальних закладів. / С. Е. Остапов, С. П. Євсеєв,О.Г. Король. – Львів: «Новий Світ-2000», 2020 . – 678 с.
7. Кібербезпека в сучасному світі : матеріали ІІІ Всеукраїнської науково практичної конференції (м. Одеса, 19 листопада 2021 р.) / за ред. О. В. Дикого ;уклад.: С. А. Горбаченко, Н. І. Логінова. – Одеса, 2020. – 148 с.
8. Криптоаналіз. Криптографічні протоколи / О.М. Гапак // Навчальний посібник з курсу «Комп'ютерна криптографія» для студентів інженерно-технічного факультету спеціальності 123-«Комп’ютерна інженерія». Ужгород: видавництво ПП «АУТДОР-ШАРК», 2021р. – 96с..
9. Лісовська Ю. Кібербезпека. Ризики та заходи. - К.: Кондор, 2019. - 272 с.
10. Логінова Н. І. Правовий захист інформації : навч. посібн. / Н. І. Логінова, Р. Р. Дробожур. - Одеса : Фенікс, 2015. - 264 с.
11. Організаційно-правові основи захисту службової 0-64 інформації: навч. посіб. /1. П. Касперський, С. 0. Князєв, О. І. Матяш та ін. - Київ : Нац. акад. СБУ, 2017.-120 с.
12. Організація захисту інформації з обмеженим доступом: навч. посіб. /А.М.Гуз, І.П.Касперський, С.О.Князєв та ін. – К.: Нац. акад., СБУ, 2018. –252 с
13. Стандарти захисту персональних даних в соціальній сфері / М. В. Бем,, І. М. Городиський. –Львів: б.в., 2018. - 110 с.
14. Тарнавський Ю.А. Технології захисту інформації [Електронний ресурс]: підручник. – К.: КПІ ім. Ігоря Сікорського, 2018. – 162 с. Режим доступу до ресурсу: https://ela.kpi.ua/bitstream/123456789/23896/1/TZI_book.pdf