МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДОНЕЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ВАСИЛЯ СТУСА

Л. В. Загоруйко
А. В. Дудатьєв

КОНСПЕКТ ЛЕКЦІЙ
з дисципліни
«ТЕХНОЛОГІЇ ПРОГРАМНОГО ЗАХИСТУ ІНФОРМАЦІЇ»

Вінниця
2024

УДК 004.056.5(075.8)
К 65

Рекомендовано до друку вченою радою факультету інформаційних та прикладних технологій ДонНУ імені Василя Стуса (протокол № 9 від 24.01.2024 р.)

Укладачі: *Загоруйко Л. В.,* канд. техн. наук, доцент, доцент кафедри прикладної математики та кібербезпеки ДонНУ імені Василя Стуса; *Дудатьєв А. В.,* канд. техн. наук, доцент, доцент кафедри захисту інформації Вінницького національного технічного університету.

Рецензенти: *Крижановський В. Г.,* д-р техн. наук, професор кафедри прикладної математики та кібербезпеки Донецького національного університету імені Василя Стуса;
Войтович О. П., канд. техн. наук, доцент кафедри захисту інформації Вінницького національного технічного університету.

Загоруйко Л. В., Дудатьєв А. В.

К 65 Конспект лекцій з дисципліни «Технології програмного захисту інформації» / укл. Л. В. Загоруйко, А. В. Дудатьєв. Вінниця: ДонНУ імені Василя Стуса, 2024. 80 с.

У конспекті лекцій з дисципліни «Технології програмного захисту інформації» подано теоретичний і методичний матеріал для вивчення і самостійного опрацювання матеріалу із сучасних проблем захисту інформації, який передбачає знайомство здобувачів вищої освіти з загальними проблемами теорії програмного захисту інформації, основними характеристиками загроз інформаційній безпеці в інформаційно-комунікаційних системах, шляхи забезпечення інформаційної безпеки, моделі політики безпеки, програмні методи захисту інформації в операційних системах, програмні моделі безпеки систем та технологія блокчейну для захисту інформації в інформаційно-комунікаційних системах.

Конспект лекцій рекомендовано для здобувачів вищої освіти спеціальності 125 Кібербезпека, а також для фахівців із систем інформаційної безпеки, які спеціалізуються в області використання й упровадження інформаційних технологій у різних сферах діяльності.

УДК 004.056.5(075.8)

ЗМІСТ

Тема 1. Вступ. Проблеми теорії захисту інформації 5
1.1. Вступ 5
1.2. Захист інформації в ІТС 6
1.3. Теорія захисту інформації 7

Тема 2. Характеристика загроз безпеці інформації 10
2.1. Загрози безпеці комп'ютерної системи 10
2.2. Загроза розкриття 12
2.3. Загроза порушення цілісності 12
2.4. Загроза відмови в обслуговуванні 13

Тема 3. Несанкціонований доступ. Порушники безпеки 15
3.1. Способи несанкціонованого доступу 15
3.2. Модель порушника 17

Тема 4. Шляхи забезпечення безпеки інформації. Концепція захисту інформації 19
4.1. Концепція захисту інформації 19
4.2. Стратегія та архітектура захисту інформації 19
4.3. Політика захисту 21
4.4. Види забезпечення безпеки інформації 21

Тема 5. Політика безпеки інформації 23
5.1. Політика безпеки 23
5.2. Етапи розробки політики безпеки 24

Тема 6. Моделі політики безпеки 28
6.1. Дискреційна політика безпеки 28
6.2. Мандатна політика безпеки 29
6.3. Рольова політика безпеки 31
6.4. Монітор безпеки 32

Тема 7. Криптографічні методи захисту інформації 34
7.1. Основні положення та визначення 34
7.2. Характеристика алгоритмів шифрування 36

Тема 8. Методи захисту інформації в операційних системах 41
8.1. Вступ 41
8.2. Алгоритм симетричного шифрування DES (Data Encryption Standard) 42
8.3. Шифрування. Початкова перестановка 43
8.4. Операція розгортання ключа 45
8.5. Операція розшифрування 46

Тема 9. Аналіз безпеки програмного забезпечення та руйнівне програмне забезпечення 48
9.1. Визначення понять 48
9.2. Підкласи РПЗ 49

Тема 10. Методи аналізу безпеки програмного забезпечення 51
10.1. Вступ 51
10.2. Методи, що використовуються для аналізу безпеки ПЗ 51
10.3. Формальна задача для аналізу безпеки ПЗ 53

Тема 11. Поняття про хешувальні алгоритми, їх призначення, вимоги до них 56
11.1. Визначення основних понять 56
11.2. Класифікація хеш-функцій 58

Тема 12. Поняття про цифровий підпис, вимоги до нього 60
12.1. Визначення основних понять 60
12.2. Процес побудови схеми ЕЦП 62

Тема 13. Основні положення керування ключами. Життєвий цикл криптографічного ключа 67
13.1. Визначення основних понять 67
13.2. Життєвий цикл керування ключами 70

Тема 14. Технологія блокчейну. 72
14.1. Визначення основних понять 72
14.2. Підтвердження транзакцій 74

Перелік питань для підсумкового контролю 75

Рекомендована література 78

ТЕМА 1
ВСТУП. ПРОБЛЕМИ ТЕОРІЇ ЗАХИСТУ ІНФОРМАЦІЇ

У лекції розглядаються питання створення систем захисту інформації в інформаційно-комунікаційних системах (ІТС), про об'єкти зазіхань, властивості інформації та захист інформації в ІТС. Розглядаються три періоди розвитку та напрями розвитку теорії захисту інформації.

Ключові слова: доступність, цілісність, конфіденційність, захист інформації, захищена ІТС, безпека інформації, політика безпеки.

План

1.1. Вступ.
1.2. Захист інформації в ІТС.
1.3. Теорія захисту інформації.

1.1. Вступ

Широке використання інформаційних технологій у всіх сферах життя суспільства робить актуальною проблему захисту інформації, її користувачів, інформаційних ресурсів, каналів передачі даних від злочинних зазіхань зловмисників. Концентрація інформації в комп'ютерах (аналогічно концентрації готівки в банках) змушує одних посилювати пошуки шляхів доступу до інформації, а інших відповідно посилювати контроль над нею з метою захисту.

Складність створення системи захисту інформації визначається тим, що дані можуть бути викрадені з комп'ютера (скопійовані), одночасно залишаючись на місці. Цінність деяких даних полягає у володінні ними, а не в їх знищенні або зміні. Для забезпечення безпеки інформації складно кваліфіковано визначити межі розумної безпеки і відповідної підтримки системи в працездатному стані.

Об'єктами зазіхань можуть бути як самі матеріальні технічні засоби (комп'ютери і периферія), так і програмне забезпечення та бази даних.

У процесі розвитку технологій електронних платежів, «безпаперового» документообігу серйозний збій локальних мереж може паралізувати роботу цілих підприємств, що призведе до відчутних збитків. Невипадково захист даних у комп'ютерних мережах стає однією із найгостріших проблем. Забезпечення безпеки інформації у комп'ютерних мережах передбачає створення перешкод для будь-яких несанкціонованих спроб розкрадання або модифікації даних, що передані у мережі. Водночас дуже важливо зберегти такі властивості інформації:

- доступність,
- цілісність,
- конфіденційність.

Доступність інформації – здатність забезпечувати своєчасний і безперешкодний доступ користувачів до інформації, яка їх цікавить.

Цілісність інформації полягає в її існуванні в неспотвореному вигляді (незмінному відносно деякого фіксованого її стану).

Конфіденційність – це властивість, що вказує на необхідність введення обмежень доступу до цієї інформації для визначеного кола користувачів. Для того, щоб правильно оцінити можливий реальний збиток від втрати інформації, що зберігається на комп'ютері, необхідно знати, які загрози під час цього можуть виникнути і яких адекватних заходів для її захисту необхідно вживати.

Захищена ІТС – це система, яка у певних умовах експлуатації забезпечує безпеку інформації, яку вона обробляє, і водночас підтримує свою працездатність в умовах дії на неї заданої множини загроз.

Безпека інформації (information security) – це стан інформації, в якому забезпечується збереження визначених політикою безпеки властивостей інформації.

1.2. Захист інформації в ІТС

Захист інформації в ІТС (information protection, information security, computer system security) – це діяльність, що спрямована на забезпечення безпеки оброблюваної в ІТС інформації та ІТС загалом, і дає змогу запобігти або ускладнити можливість реалізації загроз, а також знизити величину потенційних збитків внаслідок реалізації загроз.

Як показує аналіз проблеми ЗІ, а також численних джерел з цієї проблеми, під час організації ЗІ в ІС можна виділити такі ключові питання:

- доступ до інформації;
- безпека інформації;
- комплексний контроль;
- інтеграція систем захисту інформації з іншими системами безпеки.

Роботи з організації захисту інформації, що обробляється на об'єктах ІТС, зазвичай проводяться за трьома основними напрямами, що не виключають, а доповнюють один одного:

- протидія несанкціонованому отриманню інформації за допомогою технічних засобів розвідки (протидія технічній розвідці) (системи просторового зашумлення, екранування технічних засобів ІТС та ін.);
- вдосконалення організаційних і організаційно-технічних заходів обробки важливої інформації (охорона об'єктів, організація зберігання носіїв інформації та ін.);
- блокування НСД до інформації (розмежування доступу, системи ідентифікації і автентифікації та ін.).

Ці напрями реалізуються з урахуванням таких основних груп чинників, що впливають на захищеність інформації: людський; технічний; алгоритмічний.

Незважаючи на те, що технології захисту інформаційних систем почали розвиватися відносно недавно, сьогодні вже існує багато теоретичних моделей, які дають змогу описувати практично всі аспекти безпеки і забезпечувати засоби захисту формально підтвердженою алгоритмічною базою.

1.3. Теорія захисту інформації

Теорія захисту інформації (ТЗІ) – це наука про загальні принципи та методи побудови захищених ІС. Це природнича наука, яка має відповідні аксіоматику, понятійний та формальний апарат і використовує методи системного аналізу для вивчення систем і теорії прийняття рішень для розв'язання задач синтезу систем захисту інформації.

Теорія захисту інформації до цього часу залишається відносно замкнутою науковою дисципліною у частині розробки та впровадження формальних методів. Розвиток цих методів не завжди є синхронізованим із досягненнями як класичних, так і сучасних наук.

З позицій розвитку методології можна виділити **три періоди розвитку теорії захисту інформації** у комп'ютерних системах та мережах: емпіричний, концептуально-емпіричний та теоретико-концептуальний.

Перший – емпіричний період розвитку теорії захисту інформації розрізняє використання неформальних (описових) методів для вирішення задач аналізу СЗІ. Синтез систем захисту інформації водночас здійснюється методом спроб і помилок з використанням функціонально-орієнтованих механізмів захисту. Цей період розпочався з 70-х років минулого сторіччя.

Другий відрізняється від емпіричного певним узагальненням неформальних підходів до аналізу систем захисту інформації. Синтез систем захисту інформації вже здійснюється з використанням уніфікованих та стандартних рішень із захисту. Початком цього періоду можна визначити 80–90-ті рр. минулого сторіччя.

Третій – теоретико-концептуальний період розвитку теорії захисту інформації – характеризується використанням методів формальної теорії захисту інформації для розв'язання задач аналізу. Задачі синтезу систем захисту інформації починають розв'язуватися з використанням математичної теорії оптимізації, методів системного аналізу та прийняття рішень. Початком теоретико-концептуального періоду розвитку теорії захисту інформації можна визначити 90-ті роки минулого сторіччя.

Найбільш характерні особливості теорії захисту інформації сьогодні полягають у такому:

1) чітка практична спрямованість – в основному більшість положень спочатку реалізуються у вигляді конкретних схем і рекомендацій, і тільки потім узагальнюються та фіксуються у вигляді теоретичних положень чи методичних рекомендацій;

2) сильна залежність теоретичних розробок від конкретних способів реалізації ІТС, що визначаються проєктними, програмними чи апаратними рішеннями, –

конкретна реалізація тієї чи іншої ІТС визначає можливі види атак, а отже, і ті чи інші захисні заходи;

3) багатоаспектність, тобто дослідження із широкого кола напрямів (організаційні заходи, технічний захист, захист від несанкціонованого доступу та ін.);

4) відсутність системонезалежних теоретичних положень, на основі яких можлива реалізація різних проєктів ІТС.

Через розвиток інформаційних технологій виникають нові задачі із забезпечення безпеки інформації, підходи до вирішення яких на початковому етапі майже завжди мають описовий характер.

Наразі виділяються два основні підходи до розгляду питань теорії забезпечення безпеки інформації: **неформальний** (або описовий) і **суто формальний.**

Найбільший розвиток одержали два формальні напрями, кожен із яких заснований на своєму баченні проблеми безпеки і націлений на вирішення певних задач – це **формальне моделювання безпеки** і **криптографія.**

Водночас ці різні за походженням і розв'язуваними задачами напрями доповнюють один одного: криптографія може запропонувати конкретні методи захисту інформації у вигляді алгоритмів ідентифікації, автентифікаці, шифрування і контролю цілісності даних, а формальні моделі безпеки надають розробникам захищених систем основні принципи, що лежать в основі архітектури захищеної ІТС і визначають концепцію її побудови.

Основне призначення моделі – забезпечити необхідний рівень розуміння проблеми захисту для успішної реалізації вимог до безпеки системи.

Одним з основних понять, на основі яких будуються моделі, є політика захисту або безпеки. Під **політикою безпеки** розуміється сукупність норм і правил, що регламентують процес обробки інформації, виконання яких забезпечує захист від певного набору загроз і є необхідною (а іноді і достатньою) умову безпеки системи.

Формальне представлення політики безпеки називають моделлю політики безпеки. Воно відіграє ключову роль у визначенні змісту моделі безпеки. Отже, для успішної розробки хорошої моделі безпеки необхідна наявність чітко визначеної політики безпеки.

У випадку розробки строгої формальної моделі безпеки створення політики повинно спиратися на найбільш придатні математичні методи для опису і аналізу її змісту. Основна мета створення політики безпеки інформаційної системи й опису її у вигляді формальної моделі – це визначення умов, яким повинно підпорядковуватися поводження системи, вироблення критерію безпеки і проведення формального доведення відповідності системи цьому критерію з дотриманням встановлених правил і обмежень.

Зв'язок наведених напрямів теорії захисту інформації можна представити у вигляді схеми (рис. 1.1).

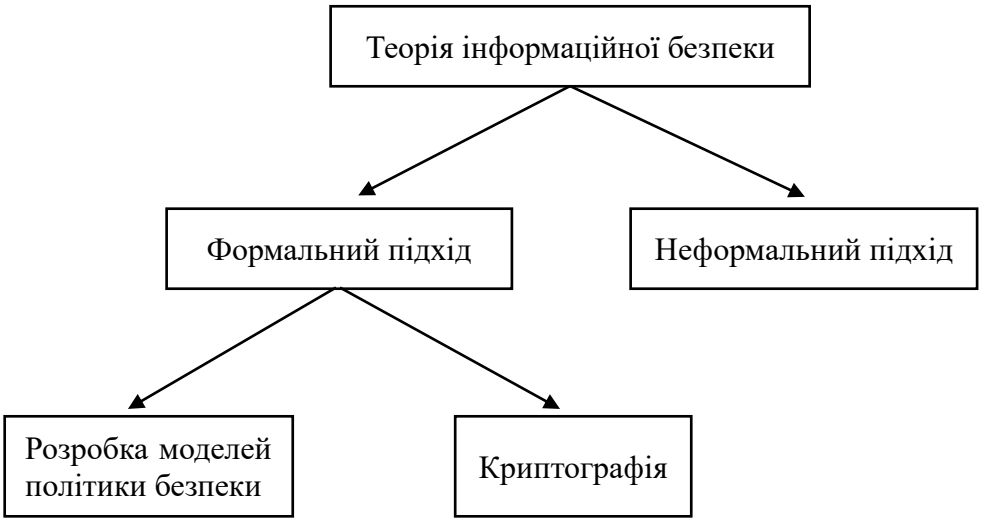

Рис. 1.1. Напрями розвитку теорії захисту інформації

Значно більш ефективним і поширеним поки виявилося застосування **неформальних** описових і класифікаційних підходів.

Замість формальних викладок тут використовуються різноманітні прийоми категоріювання: порушників (за цілями, кваліфікацією та доступними обчислювальними ресурсами); інформації (за рівнями критичності та конфіденційності); загроз (за способами реалізації, місцями реалізації тощо), засобів захисту (за функціональністю і гарантованістю реалізованих можливостей тощо) та ін. Такий підхід не дає точних числових значень показників захищеності, але дає змогу класифікувати ІТС за рівнем захищеності і порівнювати їх між собою.

Прикладами таких класифікаційних методик можуть слугувати різні критерії оцінки безпеки інформаційних технологій і продуктів, які прийняті в багатьох країнах як національні стандарти, що встановлюють класи і рівні захищеності. Зокрема, результатом розвитку національних стандартів у цій області є міжнародний стандарт ISO 15408, який узагальнює світовий досвід.

В Україні також є низка офіційних нормативних документів, що регламентують усі основні аспекти, пов’язані з безпекою КС і захистом інформації в них від НСД. Однак документів, що регламентують процеси побудови моделей безпеки, немає.

Питання для самоконтролю:

1. Які властивості має інформація?
2. Що таке захищена ІТС?
3. Що таке безпека інформації?
4. Що таке захист інформації в ІТС?
5. Які є напрями розвитку теорії захисту інформації?

ТЕМА 2
ХАРАКТЕРИСТИКА ЗАГРОЗ БЕЗПЕЦІ ІНФОРМАЦІЇ

У темі наведена інформація про основні загрози безпеці інформації та розглядаються три основні їх види.

Ключові слова: загроза, загрози безпеці, загрози розкриття, випадкові загрози, навмисні загрози, загрози порушення цілісності, загроза відмови в обслуговуванні.

План

2.1. Загрози безпеці комп'ютерної системи.
2.2. Загроза розкриття.
2.3. Загроза порушення цілісності.
2.4. Загроза відмови в обслуговуванні.

2.1. Загрози безпеці комп'ютерної системи

Загрозою безпеці комп'ютерної системи вважається подія (вплив), що у випадку реалізації стане причиною порушення цілісності інформації, її втрати або заміни.

Означення. Загроза – це потенційно можлива будь-яка несприятлива дія на інформацію, що може призвести до порушень хоча б одної з фундаментальних властивостей захищеної інформації.

Загрози можуть бути випадковими та навмисними. До випадкових загроз належать:

1) помилки обслуговуючого персоналу і користувачів;
2) втрата інформації, обумовлена неправильним збереженням архівних даних;
3) випадкове знищення або зміна даних;
4) збої устаткування і електроживлення;
5) збої кабельної системи;
6) перебої електроживлення;
7) збої дискових систем;
8) збої систем архівування даних;
9) збої роботи серверів, робочих станцій, мережевих карт і под.;
10) некоректна робота програмного забезпечення;
11) зміна даних у разі помилок у програмному забезпеченні;
12) зараження системи комп'ютерними вірусами;
13) несанкціонований доступ;
14) випадкове ознайомлення з конфіденційною інформацією сторонніх осіб.

Найчастіше збиток спричиняється не через чийсь злий намір, а просто через елементарні помилки користувачів. Через це, крім контролю доступу, необхідним

елементом захисту комп'ютерної інформації є розмежування повноважень користувачів. Однак найбільш небезпечним джерелом загроз інформації є навмисні дії зловмисників. Стандартність архітектурних принципів побудови устаткування і програм забезпечує порівняно легкий доступ професіонала до інформації, що знаходиться в персональному комп'ютері. До навмисних загроз належать:

1) несанкціонований доступ до інформації і мережевих ресурсів;

2) розкриття і модифікація даних і програм, їх копіювання;

3) розкриття, модифікація або підміна трафіка обчислювальної мережі;

4) розробка і поширення комп'ютерних вірусів, введення в програмне забезпечення логічних бомб;

5) крадіжка магнітних носіїв і розрахункових документів;

6) руйнування архівної інформації або навмисне її знищення;

7) фальсифікація повідомлень, відмова від факту одержання інформації або зміна часу його прийому;

8) перехоплення та ознайомлення з інформацією, яка передана каналами зв'язку, тощо.

Виділяють три основні види загроз безпеці: загрози розкриття, цілісності і відмови в обслуговуванні (рис. 2.1).

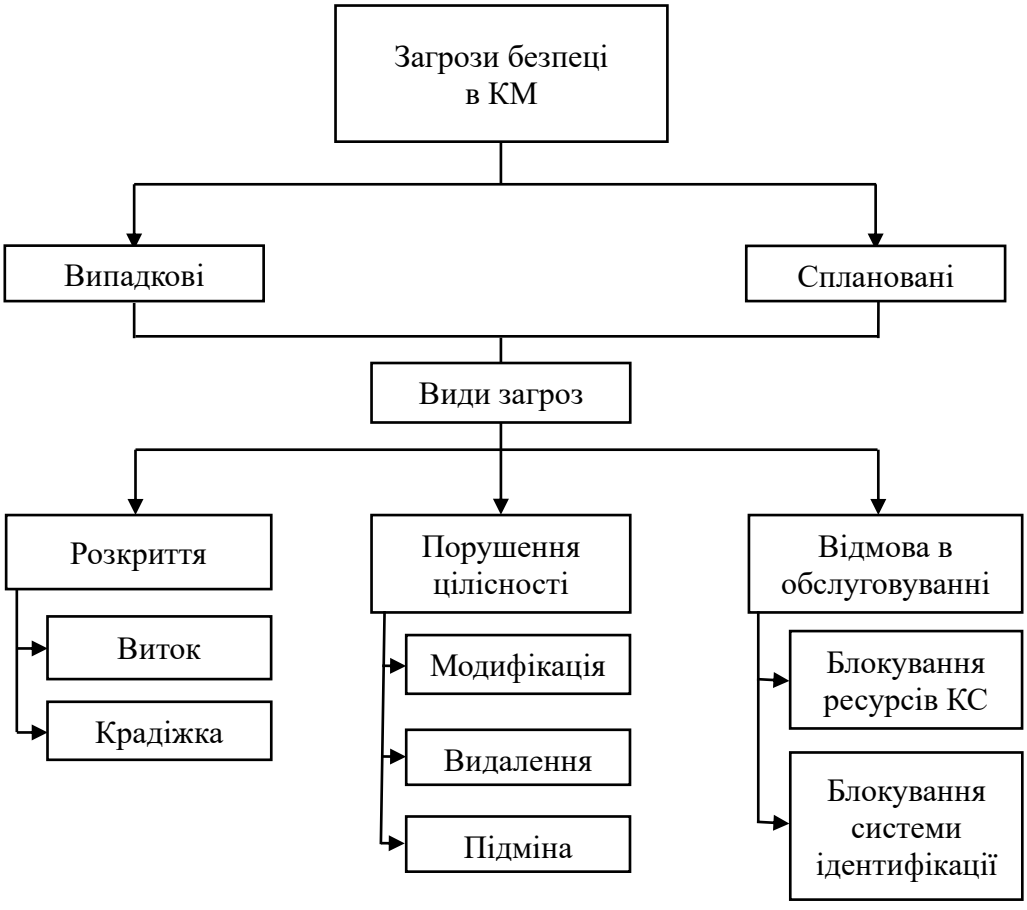

Рис. 2.1. Види загроз безпеці інформації в комп'ютерних мережах

2.2. Загроза розкриття

Загроза розкриття полягає в тому, що інформація стає відомою тому, кому не потрібно її знати. Іноді замість слова «розкриття» використовуються терміни «крадіжка» або «виток».

Означення. Загроза порушення конфіденційності інформації – це можливість реалізації певної множини доступів для ознайомлення з інформацією користувачам і/або процесам, які не мають на це відповідних повноважень.

Якщо звернутися до питань захисту проти каналів витоку такого типу, то варто сказати, що протидію загрозам порушення конфіденційності можна забезпечити за допомогою таких послуг:

- довірча конфіденційність – таке управління доступом, за якого засоби захисту дають змогу звичайним користувачам управляти (передають управління) потоками інформації між іншими користувачами і об’єктами свого домена (наприклад, на підставі права власності об’єкта), тобто призначення і передача повноважень не вимагають адміністративного втручання;
- адміністративна конфіденційність – це управління, за якого засоби захисту дають змогу управляти потоками інформації між користувачами і об’єктами тільки спеціально авторизованим користувачам;
- повторне використання об’єктів – якщо перед наданням об’єкта користувачеві або процесу в ньому не залишається інформації, яку він містив, і відміняються попередні права доступу до цього об’єкта;
- аналіз прихованих каналів – проводиться з метою виявлення і перекриття наявних потоків інформації, які не контролюються іншими послугами;
- конфіденційність під час обміну – дає змогу забезпечити безпеку обміну інформацією між захищеними об’єктами в незахищеному середовищі.

2.3. Загроза порушення цілісності

Загроза порушення цілісності – будь-яка навмисна зміна (модифікація чи видалення) даних, що зберігаються в обчислювальній системі або передаються з однієї системи в іншу. Зазвичай вважається, що загрозі розкриття найбільше піддаються державні структури, а загрозі порушення цілісності – ділові або комерційні.

Серед заходів захисту від порушення цілісності виділяють такі:

1) своєчасне резервне копіювання цінної інформації;

2) введення надмірності в саму інформацію, тобто використання вадостійкого кодування інформації, що дає змогу контролювати її цілісність;

3) введення надмірності в процес обробки інформації, тобто використання автентифікації, що дає змогу контролювати цілісність об’єктів;

4) введення системної надмірності, тобто підвищення «живучості» системи.

Послуги, за допомогою яких забезпечується цілісність, такі:

- довірча цілісність – аналогічна довірчій конфіденційності;
- адміністративна цілісність – аналогічна адміністративній конфіденційності;
- відкат – дає можливість відновлюватися після помилок користувача, збоїв програмного забезпечення й апаратури та підтримувати цілісність баз даних, додатків та ін.; забезпечує можливість відміни операції або послідовності операцій і повернути захищений об'єкт у попередній стан;
- цілісність під час обміну – дає змогу забезпечити захист об'єктів від несанкціонованої модифікації інформації, що міститься в них, під час їх експорту / імпорту в незахищеному середовищі.

2.4. Загроза відмови в обслуговуванні

Загроза відмови в обслуговуванні виникає кожного разу, коли внаслідок певних дій блокується доступ до деякого ресурсу обчислювальної системи.

Означення. Загроза порушення доступності до інформації – це можливість реалізації певної множини заходів, які не дають змогу її використовувати за вимогами користувачів і/або процесів, що мають на це відповідні повноваження.

Можна виділити такі напрями повсякденної діяльності в ІТС для підтримки її працездатності:

- підтримка користувачів, тобто консультації і різноманітні подання їм допомоги;
- підтримка ПЗ, тобто контроль за ПЗ, яке використовується в ІТС;
- конфігураційне керування, яке дає змогу контролювати зміни в програмній конфігурації;
- резервне копіювання;
- керування носіями, що забезпечує фізичний захист носіїв;
- документування;
- регламентні роботи.

Доступність в ІТС забезпечується правильним використанням таких послуг:

- використання ресурсів – дає змогу користувачам керувати процесами використання послуг і ресурсів;
- стійкість до відмов – покликана гарантувати доступність КС (можливість використання інформації, окремих функцій або КС загалом) після відмови її компонента;
- гаряча заміна – дає змогу гарантувати доступність КС у процесі заміни окремих компонентів;
- відновлення після збоїв – забезпечує повернення КС до відомого захищеного стану після відмови в обмірковуванні.

Питання для самоконтролю:

1. Що таке загроза?
2. Які можуть бути загрози?
3. Які загрози належать до навмисних?
4. Які загрози належать до випадкових?
5. Що таке загроза порушення цілісності?
6. Що таке загроза відмови в обслуговуванні?

ТЕМА 3
НЕСАНКЦІОНОВАНИЙ ДОСТУП. ПОРУШНИКИ БЕЗПЕКИ

У темі наведена інформація про способи несанкціонованого доступу до інформації, яку мету має зловмисник та що таке модель порушника, які є категорії порушників.

Ключові слова: НСД, порушник.

План

3.1. Способи несанкціонованого доступу.

3.2. Модель порушника.

3.1. Способи несанкціонованого доступу

Спосіб несанкціонованого доступу (НСД) – це сукупність прийомів і порядок дій з метою одержання (добування) інформації, що охороняється, незаконним протиправним шляхом і забезпечення можливості впливати на цю інформацію (наприклад, підмінити, знищити та ін.).

Під час здійснення несанкціонованого доступу зловмисник має три мети:

- одержати необхідну інформацію для конкурентної боротьби;
- мати можливість вносити зміни в інформаційні потоки конкурента відповідно до своїх інтересів;
- завдати шкоди конкурентові шляхом знищення матеріалу інформаційних цінностей.

Спроба одержати несанкціонований доступ до комп'ютерної мережі з метою ознайомитися з нею, залишити інформацію, виконати, знищити, змінити або викрасти програму або іншу інформацію кваліфікується як **«комп'ютерне піратство».**

Для запобігання можливим загрозам, фірми повинні не тільки забезпечити захист операційних систем, програмного забезпечення та контроль доступу, але і спробувати виявити категорії порушників і ті методи, які вони використовують.

Залежно від мотивів, мети та методів дії порушників безпеки інформації можна поділити на чотири категорії:

- шукачі пригод;
- ідейні «хакери»;
- «хакери»-професіонали;
- ненадійні (неблагополучні) співробітники.

Шукач пригод рідко має продуманий план атаки. Він вибирає мету у випадковий спосіб і зазвичай відступає, зіштовхнувшись із ускладненнями. Знайшовши діру в системі безпеки, він намагається зібрати закриту інформацію, але практично ніколи не намагається її таємно змінити. Своїми перемогами такий шукач пригод ділиться тільки зі своїми близькими друзями-колегами.

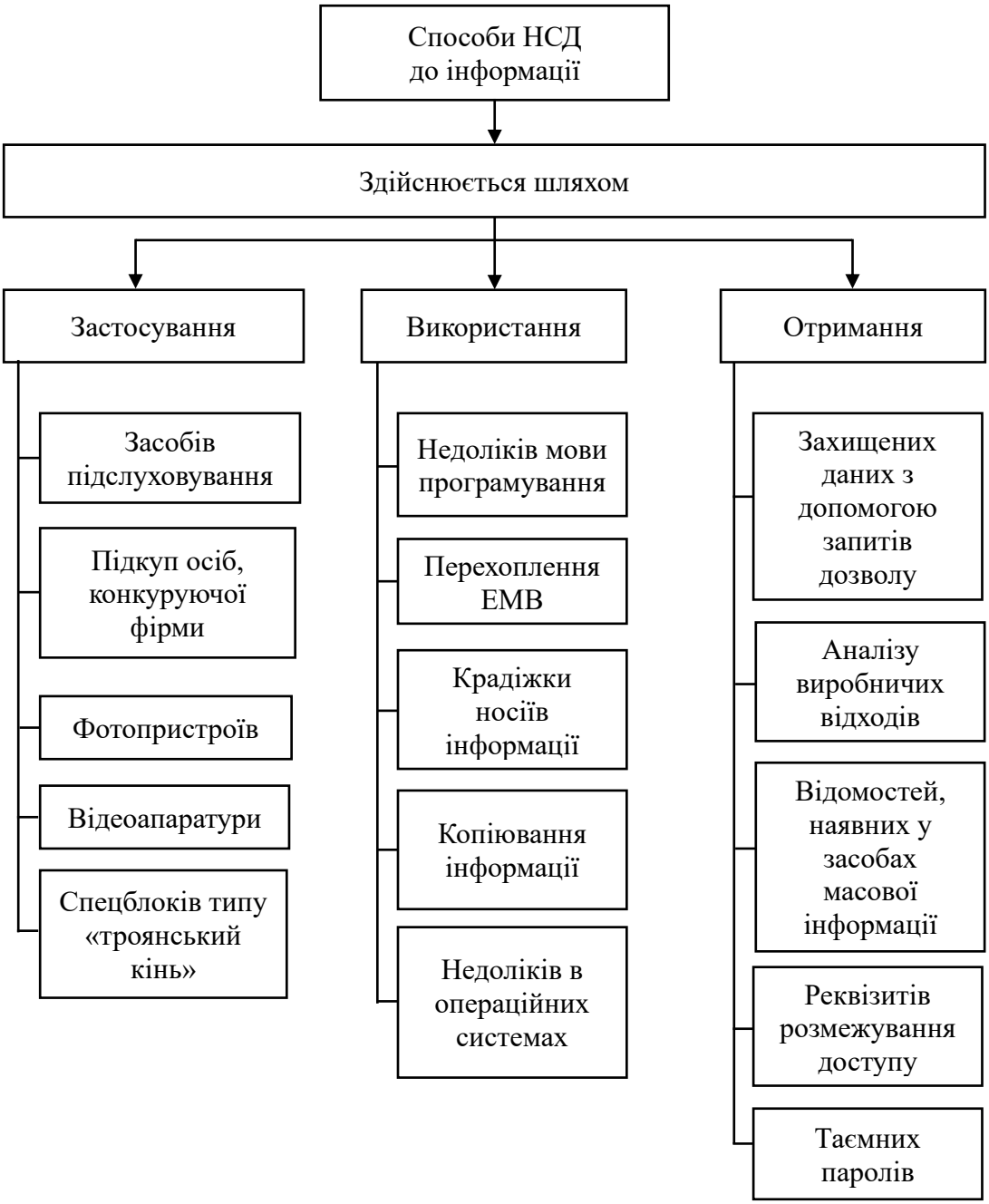

Рис. 3.1. Способи НСД до конфіденційної інформації

Ідейний «хакер» – це той самий шукач пригод, але більш майстерний. Він уже вибирає собі конкретні цілі (хости і ресурси) на підставі своїх переконань. Його улюбленим видом атаки є зміна інформаційного наповнення вебсервера або, рідше, блокування роботи ресурсу, що атакується. Порівняно з шукачем пригод, ідейний «хакер» розповідає про успішні атаки набагато більшій аудиторії, зазвичай розміщаючи інформацію на хакерському вебвузлі.

«Хакер»-професіонал має чіткий план дій і спрямовує його на визначені ресурси. Його атаки добре продумані і зазвичай здійснюються у кілька етапів. Спочатку він збирає попередню інформацію (тип ОС, надані сервіси і засоби захисту). Потім він складає план атаки з урахуванням зібраних даних і добирає (або навіть

розробляє) відповідні інструменти. Далі, провівши атаку, він одержує закриту інформацію, і нарешті знищує всі сліди своїх дій. Такий професіонал зазвичай добре фінансується і може працювати один або у складі команди професіоналів.

Ненадійний (неблагополучний) співробітник своїми діями може спричинити стільки ж проблем (буває і більше), скільки і промисловий шпигун, до того ж його присутність зазвичай складніше знайти. До того ж йому доводиться долати не зовнішній захист мережі, а тільки менш жорсткий, внутрішній. Він не такий витончений у способах атаки, як промисловий шпигун, і тому частіше допускає помилки, чим може видати свою присутність.

3.2. Модель порушника

Модель порушника визначає:

- категорії осіб, серед яких може виявитися порушник;
- можливі цілі порушника і їх градації за ступенем важливості та небезпеки;
- припущення про його кваліфікації;
- оцінка його технічної оснащеності;
- обмеження і припущення про характер його дій.

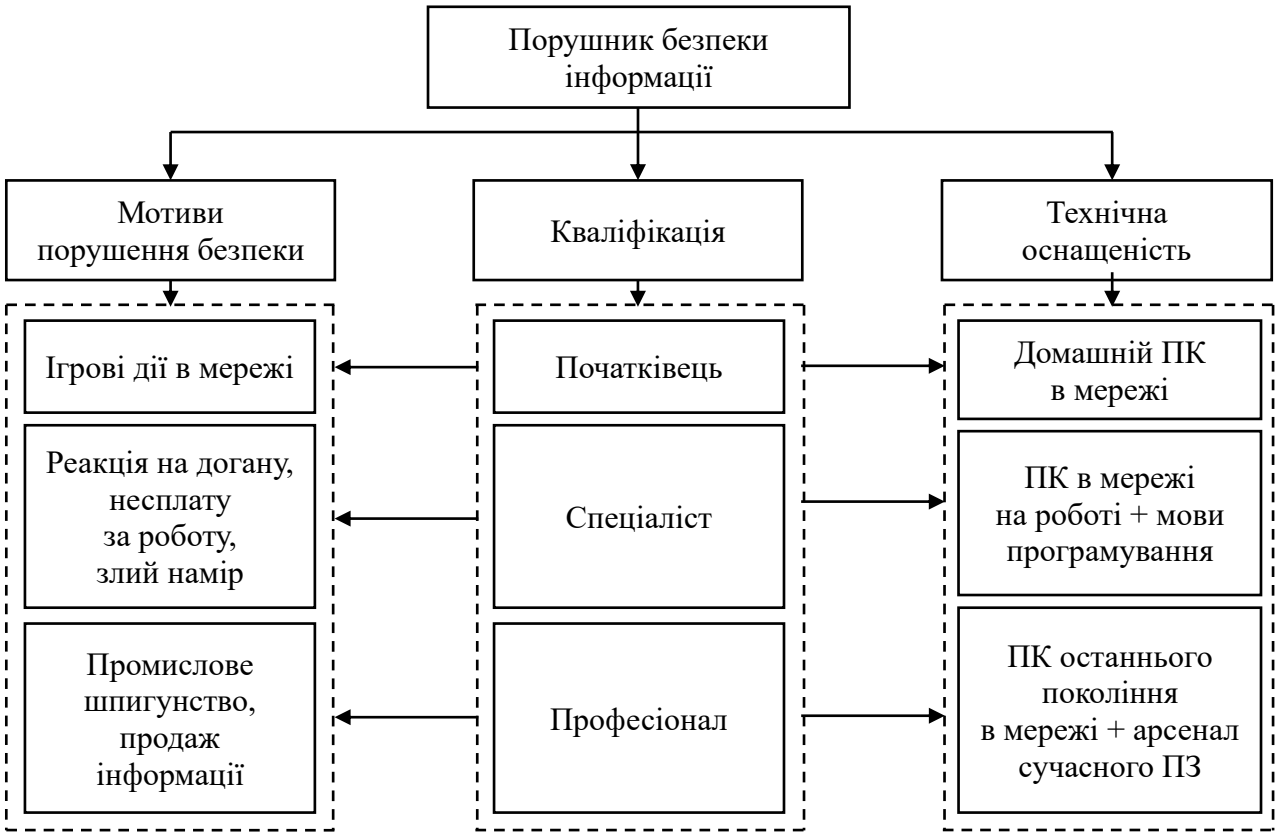

Рис. 3.2. Порушник безпеки інформації

Сьогодні, зі стрімким розвитком інтернету, «хакери» стають справжньою загрозою для державних і корпоративних комп'ютерних мереж. Так, за оцінками експертів США, напади «хакерів» на комп'ютери і мережі федеральних державних систем відбуваються в цій країні не рідше 50-ти раз на день. Багато великих компаній і організацій піддається атакам кілька разів на тиждень, а деякі навіть щодня. Не завжди такі атаки виходять ззовні, 70 % спроб зловмисного проникнення в комп'ютерні системи мають джерело всередині самої організації.

Питання для самоконтролю:

1. Що таке спосіб несанкціонованого доступу?
2. Які три мети має зловмисник?
3. На які категорії поділяють дії порушників?
4. Що визначає модель порушника?

ТЕМА 4
ШЛЯХИ ЗАБЕЗПЕЧЕННЯ БЕЗПЕКИ ІНФОРМАЦІЇ. КОНЦЕПЦІЯ ЗАХИСТУ ІНФОРМАЦІЇ

У темі наведена інформація про шляхи забезпечення інформації, про концепцію захисту інформації, ієрархічний підхід до забезпечення безпеки інформації, етапи розробки концепції захисту інформації, політику захисту, а також види забезпечення безпеки інформації.

Ключові слова: захист інформації, концепція захисту, стратегія захисту інформації, політика захисту, заходи протидії комп'ютерним злочинам.

План

4.1. Концепція захисту інформації.
4.2. Стратегія та архітектура захисту інформації.
4.3. Політика захисту.
4.4. Види забезпечення безпеки інформації.

4.1. Концепція захисту інформації

Вразливість інформації в автоматизованих комплексах обумовлена великою концентрацією обчислювальних ресурсів, їх територіальною розподіленістю, довгостроковим збереженням великого об'єму даних на магнітних та оптичних носіях, одночасним доступом до ресурсів багатьох користувачів.

Вживання заходів захисту мають певні труднощі:

1) немає єдиної теорії захисту систем;

2) виробники засобів захисту в основному пропонують окремі компоненти для вирішення приватних задач, залишаючи питання формування системи захисту і сумісності цих засобів на розсуд споживачів;

3) для забезпечення надійного захисту необхідно розв'язати цілий комплекс технічних і організаційних проблем та розробити відповідну документацію.

Концепція захисту інформації – офіційно прийнята система поглядів на проблему інформаційної безпеки і шляхи її вирішення з урахуванням сучасних тенденцій. Вона є методологічною основою політики розробки практичних заходів для її реалізації. На базі сформульованих у концепції цілей, задач і можливих шляхів їх вирішення формуються конкретні плани забезпечення інформаційної безпеки.

4.2. Стратегія та архітектура захисту інформації

В основі комплексу заходів щодо інформаційної безпеки повинна бути **стратегія захисту інформації.** У ній визначаються мета, критерії, принцип і процедури, необхідні для побудови надійної системи захисту. Найважливішою особливістю

загальної стратегії інформаційного захисту є дослідження системи безпеки. Можна виділити два основних напрями:

- аналіз засобів захисту;
- визначення факту вторгнення.

На основі концепції безпеки інформації розробляються стратегія безпеки інформації та архітектура системи захисту інформації, а далі – політика безпеки інформації (рис. 4.1).

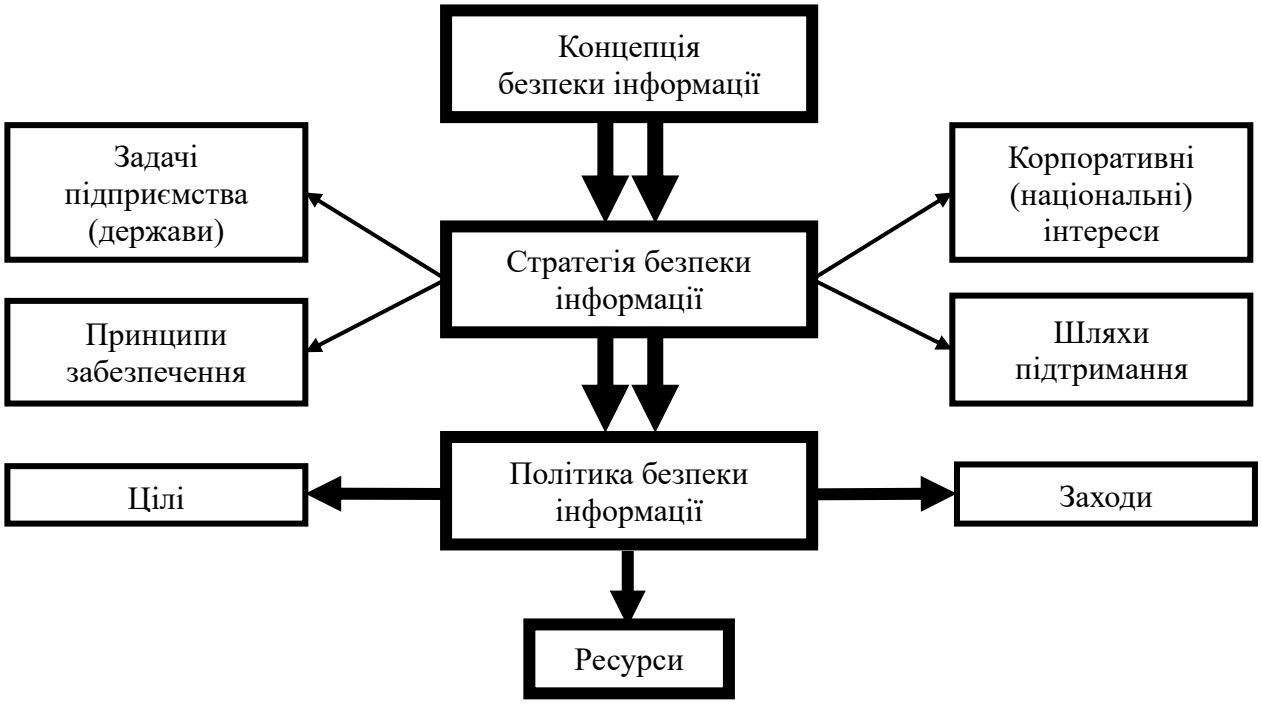

Рис. 4.1. Ієрархічний підхід до забезпечення безпеки інформації

Розробку концепції захисту рекомендується проводити в три етапи (рис. 4.2).

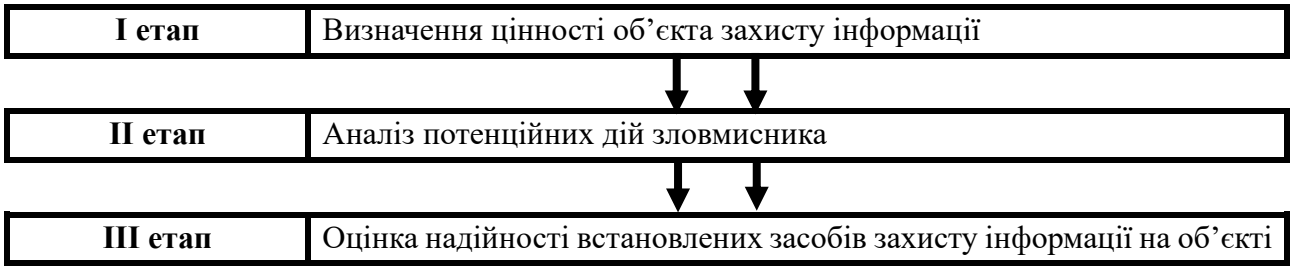

Рис. 4.2. Етапи розробки концепції захисту інформації

На першому етапі має бути чітко визначена цільова установка захисту, тобто які реальні цінності, виробничі процеси, програми, масиви даних необхідно захищати. На цьому етапі доцільно диференціювати за значимістю окремі об'єкти, що вимагають захисту.

На другому етапі має бути проведений аналіз злочинних дій, що можуть бути зроблені відносно об'єкта, що захищається. Важливо визначити ступінь реальної небезпеки найбільш розповсюджених злочинів, як-от економічне шпигунство, саботаж, крадіжки зі зломом. Потім потрібно проаналізувати найбільш ймовірні дії зловмисників стосовно основних об'єктів, що потребують захисту.

Головною метою **третього етапу** є аналіз обставин, зокрема місцевих специфічних умов, виробничих процесів, уже встановлених технічних засобів захисту.

Концепція захисту повинна містити перелік організаційних, технічних та інших заходів, що забезпечують максимальну безпеку за заданого залишкового ризику і мінімальні витрати на їх реалізацію.

4.3. Політика захисту

Політика захисту – це загальний документ, де вміщені правила доступу, визначаються шляхи реалізації політики та описується базова архітектура середовища захисту.

Власне документ складається із декількох сторінок тексту. Він формує основу фізичної архітектури мережі, а інформація, що знаходиться в ньому, визначає вибір продуктів захисту. Водночас документ може і не включати список необхідних закупок, але вибір конкретних компонентів після його складання повинен бути очевидним.

Політика захисту повинна обов'язково включати таке:

1) контроль доступу (заборона на доступ користувача до матеріалів, якими йому не дозволено користуватися);

2) ідентифікацію та автентифікацію (використання паролів або інших механізмів для перевірки статусу користувача);

3) облік (запис усіх дій користувача в мережі);

4) контрольний журнал (журнал дає змогу визначити, коли і де відбулося порушення захисту);

5) акуратність (захист від будь-яких випадкових порушень);

6) надійність (запобігання монополізації ресурсів системи одним користувачем);

7) обмін даними (захист усіх комунікацій).

4.4. Види забезпечення безпеки інформації

Сьогодні комп'ютерні злочини надзвичайно різноманітні. Це несанкціонований доступ до інформації, що зберігається в комп'ютері, введення в програмне забезпечення логічних бомб, розробка і поширення комп'ютерних вірусів, розкрадання комп'ютерної інформації, недбалість у розробці, виготовленні та експлуатації програмно-обчислювальних комплексів, підробка комп'ютерної інформації.

Всі заходи протидії комп’ютерним злочинам, що безпосередньо забезпечують безпеку інформації, можна згрупувати у:

- правові;
- організаційно-адміністративні;
- інженерно-технічні.

Питання для самоконтролю:

1. Чим обумовлена вразливість інформації в автоматизованих комплексах?
2. Що таке концепція захисту інформації?
3. Що таке політика захисту?
4. Що повинна включати політика захисту?
5. Які є види забезпечення безпеки інформації?

ТЕМА 5
ПОЛІТИКА БЕЗПЕКИ ІНФОРМАЦІЇ

У темі наведена інформація про основні задачі під час розробки політики безпеки, про комплекс задач під час розробки політики безпеки, основні правила забезпечення політики безпеки інформації, а також про етапи розробки політики безпеки.

Ключові слова: політика безпеки, розробка політики безпеки, реалізація політики безпеки, підтримка політики безпеки.

План

5.1. Політика безпеки.
5.2. Етапи розробки політики безпеки.

5.1. Політика безпеки

Розробка політики безпеки інформації повинна проводитися з урахуванням задач, вирішення яких забезпечить реальний захист об’єкта (рис. 5.1). Автоматизований комплекс можна вважати захищеним, якщо всі операції виконуються відповідно до чітко визначених правил (рис. 5.2), що забезпечують безпосередній захист об’єктів, ресурсів і операцій. Основу для формування вимог до захисту складає список загроз. Захист інформації в комп’ютерній мережі ефективніший в тому випадку, коли проєктування і реалізація системи захисту відбувається в три етапи:

- аналіз ризику;
- реалізація політики безпеки;
- підтримка політики безпеки.

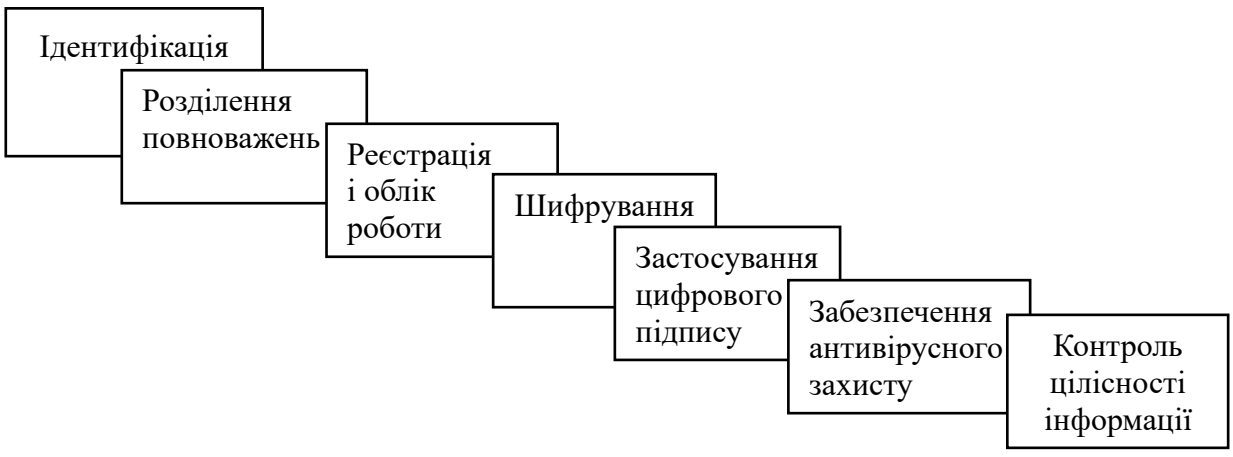

Рис. 5.1. Основні правила забезпечення політики безпеки інформації

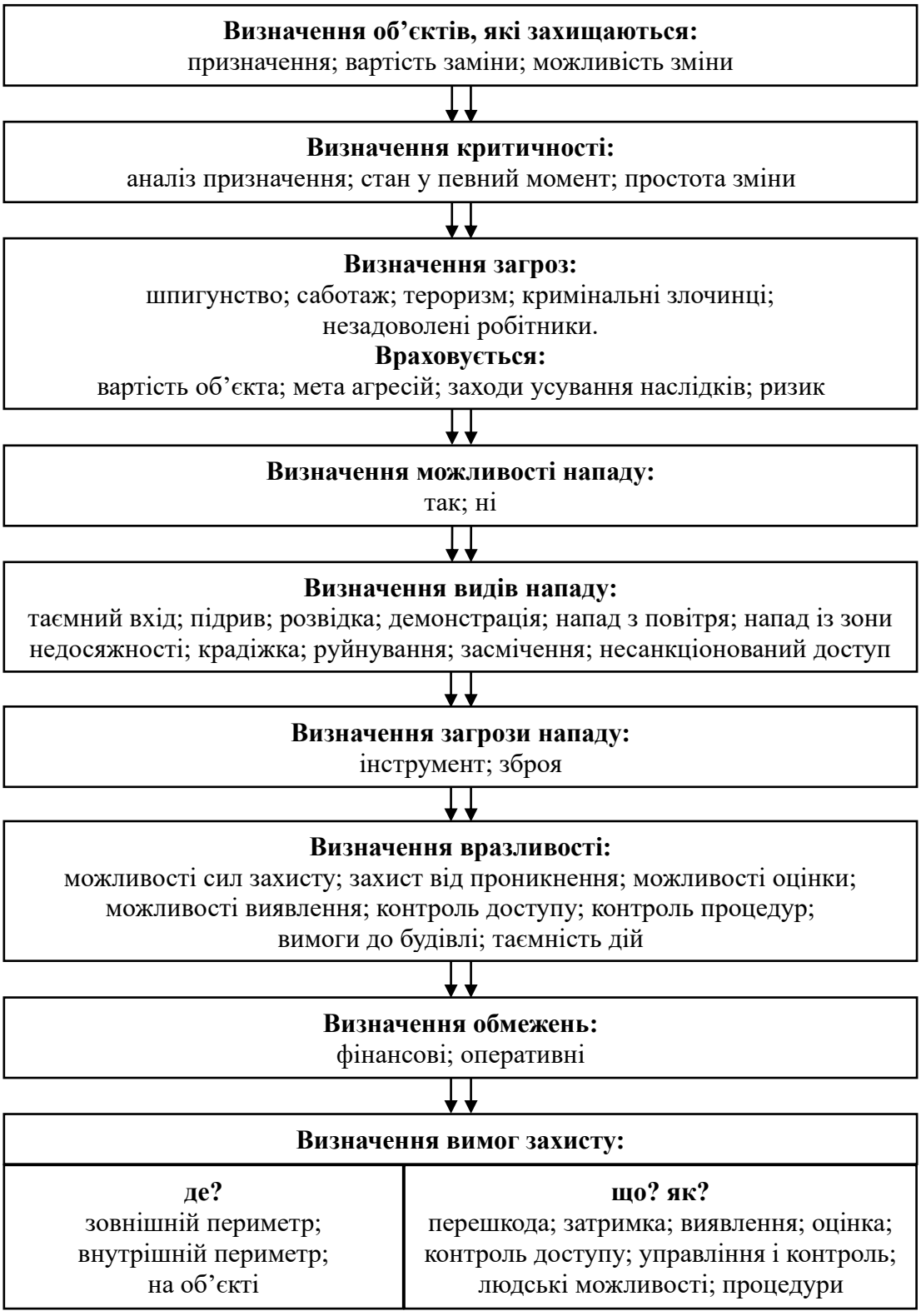

Рис. 5.2. Комплекс задач під час розробки політики безпеки

5.2. Етапи розробки політики безпеки

На першому етапі аналізуються вразливі елементи комп'ютерної мережі, визначаються й оцінюються загрози і підбираються оптимальні засоби захисту. Аналіз ризику закінчується прийняттям політики безпеки.

Політикою безпеки (Security Policy) називається комплекс взаємозалежних засобів, спрямованих на забезпечення високого рівня безпеки. У теорії захисту інформації вважається, що ці засоби повинні бути спрямовані на досягнення таких цілей:

- конфіденційність (засекречена інформація повинна бути доступна тільки тому, кому вона призначена);
- цілісність (інформація, на основі якої приймаються рішення, повинна бути достовірною і повною, а також захищеною від можливих ненавмисного і злочинного перекручувань);
- готовність (інформація і відповідні автоматизовані служби повинні бути доступні і в разі потреби готові до обслуговування).

Вразливість означає невиконання хоча б однієї з цих властивостей. Для **комп'ютерних мереж** можна виділити такі **ймовірні загрози,** які необхідно враховувати під час визначення політики безпеки:

1) несанкціонований доступ сторонніх осіб, що не належать до числа службовців і ознайомлення зі збереженою конфіденційною інформацією;

2) ознайомлення своїх службовців з інформацією, до якої вони не повинні мати доступу;

3) несанкціоноване копіювання програм і даних;

4) перехоплення та ознайомлення з конфіденційною інформацією, переданої каналами зв'язку;

5) крадіжка магнітних носіїв, що містять конфіденційну інформацію;

6) крадіжка роздрукованих документів;

7) випадкове або навмисне знищення інформації;

8) несанкціонована модифікація службовцями документів і баз даних;

9) фальсифікація повідомлень, переданих каналами зв'язку;

10) відмова від авторства повідомлення, переданого каналами зв'язку;

11) відмовлення від факту одержання інформації;

12) нав'язування раніше переданого повідомлення;

13) помилки в роботі обслуговуючого персоналу;

14) руйнування файлової структури через некоректну роботу програм або апаратних засобів;

15) руйнування інформації, викликане вірусними впливами;

16) руйнування архівної інформації, що зберігається на магнітних носіях;

17) крадіжка устаткування;

18) помилки в програмному забезпеченні;

19) відключення електроживлення;

20) збої устаткування.

Політика безпеки повинна визначатися такими заходами:

1) ідентифікація, перевірка дійсності і контроль доступу користувачів на об'єкт, у приміщення, до ресурсів автоматизованого комплексу;

2) поділ повноважень користувачів, що мають доступ до обчислювальних ресурсів;

3) реєстрація та облік роботи користувачів;

4) реєстрація спроб порушення повноважень;

5) шифрування конфіденційної інформації на основі криптографічних алгоритмів високої стійкості;

6) застосування цифрового підпису для передачі інформації каналами зв'язку;

7) забезпечення антивірусного захисту (зокрема і для боротьби з невідомими вірусами) і відновлення інформації, зруйнованої вірусними впливами;

8) контроль цілісності програмних засобів і оброблюваної інформації;

9) відновлення зруйнованої архівної інформації, навіть за значних втрат;

10) наявність адміністратора (служби) захисту інформації в системі;

11) вироблення і дотримання необхідних організаційних заходів;

12) застосування технічних засобів, що забезпечують безперебійну роботу устаткування.

Другий етап – реалізація політики безпеки – починається з проведення розрахунку фінансових витрат і вибору відповідних засобів для виконання цих задач. Водночас необхідно врахувати такі фактори: безконфліктність роботи обраних засобів, репутація постачальників засобів захисту, можливість одержання повної інформації про механізми захисту і надані гарантії. До того ж варто враховувати принципи, в яких відображені основні положення з безпеки інформації:

1) економічна ефективність (вартість засобів захисту повинна бути меншою, ніж розміри можливого збитку);

2) мінімум привілеїв (кожен користувач повинен мати мінімальний набір привілеїв, необхідних для роботи);

3) простота (захист буде тим ефективніший, чим легше користувачеві з ним працювати);

4) відключення захисту (за нормального функціонування захист не повинен відключатися, за винятком особливих випадків, коли співробітник зі спеціальними повноваженнями може мати можливість відключити систему захисту);

5) відкритість проєктування і функціонування механізмів захисту (таємність проєктування і функціонування засобів безпеки – кращий підхід до захисту інформації тому, що фахівці, які мають відношення до системи захисту, повинні повністю уявляти собі принципи її функціонування, і у випадку виникнення скрутних ситуацій адекватно на них реагувати);

6) незалежність системи захисту від суб'єктів захисту (особи, що займалися розробкою системи захисту, не повинні бути серед тих, кого ця система буде контролювати);

7) загальний контроль (будь-які винятки з безлічі контрольованих суб'єктів і об'єктів захисту знижують захищеність автоматизованого комплексу);

8) звітність і підконтрольність (система захисту повинна надавати достатньо доказів, що показують коректність її роботи);

9) відповідальність (особиста відповідальність осіб, що займаються забезпеченням безпеки інформації);

10) ізоляція і поділ (об'єкти захисту доцільно поділяти на групи так, щоб порушення захисту в одній з груп не впливало на безпеку інших груп);

11) відмова за замовчуванням (якщо відбувся збій засобів захисту, і розробники не передбачили такої ситуації, то доступ до обчислювальних ресурсів повинен бути заборонений);

12) повнота і погодженість (система захисту повинна бути цілком специфікована, протестована і погоджена);

13) параметризація (захист стає більш ефективним і гнучким, якщо він допускає зміну своїх параметрів з боку адміністратора);

14) принцип ворожого оточення (система захисту повинна проєктуватися в розрахунку на вороже оточення і припускати, що користувачі мають найгірші наміри, що вони будуть робити серйозні помилки і шукати шляхи обходу механізмів захисту);

15) залучення людини (найбільш важливі і критичні рішення повинні прийматися людиною, тому що комп'ютерна система не може передбачити всі можливі ситуації);

16) відсутність зайвої інформації про існування механізмів захисту (існування механізмів захисту повинно бути за можливості приховане від користувачів, робота яких контролюється).

Підтримка політики безпеки – третій, найбільш важливий, етап. Заходи, проведені на цьому етапі, вимагають постійного спостереження за вторгненнями у мережу зловмисників, виявлення «дір» у системі захисту об'єкта інформації, обліку випадків несанкціонованого доступу до конфіденційних даних. Водночас основна відповідальність за підтримку політики безпеки мережі лежить на системному адміністраторі, що повинен оперативно реагувати на всі випадки злому конкретної системи захисту, аналізувати їх, і використовувати необхідні апаратні і програмні засоби захисту з урахуванням максимальної економії фінансових засобів.

Питання для самоконтролю:

1. Що таке політика безпеки?
2. Які є правила забезпечення політики безпеки інформації?
3. Які є етапи розробки політики безпеки?
4. Якими заходами повинна визначатися політика безпеки?
5. У чому полягає підтримка політики безпеки?

ТЕМА 6
МОДЕЛІ ПОЛІТИКИ БЕЗПЕКИ

У темі наведена інформація про моделі політики безпеки, проблеми, які виникають під час розробки моделей політики безпеки, а також переваги і недоліки кожної моделі політики безпеки.

Ключові слова: дискреційна політика безпеки, переваги, недоліки, мандатна політика безпеки, рольова політика безпеки, монітор безпеки.

План

6.1. Дискреційна політика безпеки.
6.2. Мандатна політика безпеки.
6.3. Рольова політика безпеки.
6.4. Монітор безпеки.

6.1. Дискреційна політика безпеки

Основою дискреційної політики безпеки (ДПБ) є дискреційне управління доступом (Discretionary Access Control – DAC), яке визначається двома властивостями:

- всі суб'єкти і об'єкти системи повинні бути однозначно ідентифіковані;
- права доступу суб'єкта до об'єкта системи визначаються на основі деякого зовнішнього відносно системи правила і реалізуються шляхом безпосереднього звертання суб'єктів до об'єктів на основі певних атрибутів доступу.

Нехай O – множина об'єктів, S – множина суб'єктів, $S \subseteq O$. Якщо $U = \{U_1, \ldots, U_m\}$ – множина користувачів, то можна визначити відображення $\{\text{own}: O \to U\}$, R – множина можливих видів доступів у цій системі. Відповідно до цього відображення кожен об'єкт об'являється власністю відповідного користувача. Користувач, що є власником об'єкта, має певні права доступу до нього, а іноді і право передавати частину або навіть усі права іншим користувачам. До того ж власник об'єкта визначає права доступу інших суб'єктів до цього об'єкта, тобто фактично визначає політику безпеки стосовно цього об'єкта. Вказані права доступу записуються у вигляді матриці доступу *M*, елементи якої є підмножинами множини *R*, що визначають доступи суб'єктів $S_{i,} = 1,2,\ldots,n$ і до об'єктів $O_{j,} = 1,2,\ldots,m$.

M =

	O_1	…	O_m	S_1	…	S_n
S_1	*own, r, w*					
…						
S_n						

До переваг цього класу політик належать:

1) відносно проста реалізація та підтримка відповідних механізмів захисту. Саме цим обумовлений той факт, що більшість розповсюджених у наразі захищених ІТС забезпечують виконання положень ДПБ;

2) під час її реалізації досягається велика економія пам'яті, оскільки матриця доступів зазвичай буває дуже розрядженою, що дає змогу застосовувати техніку роботи з розрядженими матрицями.

Проте є багато проблем захисту, яких ця політика вирішити не в змозі.

Найбільш важливою вадою цього класу політик є те, що вони не витримують атак за допомогою **«троянського коня»,** оскільки вони контролюють лише операції доступу суб'єктів до об'єктів, а не інформаційні потоки. Тому коли «троянський кінь» переносить інформацію з доступного користувачу об'єкта в об'єкт, доступний зловмиснику, формально правила не порушуються, проте витік інформації здійснюється. Це, зокрема, означає, що СЗІ, яка її реалізує, погано захищає від проникнення вірусів у систему та інших засобів прихованої руйнівної дії.

Наступна проблема ДПБ – **автоматичне визначення прав.** Оскільки об'єктів багато і їх кількість безперервно змінюється, то задати заздалегідь вручну перелік прав кожного суб'єкта на доступ до об'єктів неможливо.

Ще одна з найважливіших проблем під час використання ДПБ – це **контроль розповсюдження прав доступу.** Найчастіше буває, що власник файла передає вміст файла іншому користувачу, і той фактично набуває права власника на цю інформацію.

6.2. Мандатна політика безпеки

Основу мандатної (повноважної) політики безпеки складає мандатне управління доступом (Mandatory Access Control – MAC), яке передбачає, що:

1) усі суб'єкти і об'єкти повинні бути однозначно ідентифіковані;

2) задано лінійно упорядкований набір міток таємності;

3) кожному об'єкту системи привласнена мітка таємності, яка визначає цінність інформації, що міститься в ньому – його рівень таємності в ІТС;

4) кожному суб'єкту системи привласнена мітка таємності, яка визначає рівень довіри до нього в ІТС – максимальне значення мітки таємності об'єктів, до яких суб'єкт має доступ;

5) мітка таємності суб'єкта називається його рівнем доступу;

6) доступ суб'єкта до об'єкта здійснюється шляхом порівняння їх міток таємності.

Визначається деяка однозначна функція $c(X)$ (тобто відображення $\{c: O \rightarrow L\}$), яка дає змогу для будь-яких об'єктів X і Y сказати, що коли Y більш цінний об'єкт, ніж X, то $c(Y) > c(X)$.

Означення. Політика МПБ вважає інформаційний потік $X \rightarrow Y$ дозволеним тоді і тільки тоді, коли $c(Y) > c(X)$ в решітці L.

Означення. Політика МПБ вважає інформаційний потік $X \rightarrow Y$ дозволеним тоді і тільки тоді, коли $c(Y) > c(X)$ у решітці L.

$$X \overset{r}{\rightarrow} Y \Leftrightarrow c(X) \geq c(Y),$$
$$X \overset{w}{\rightarrow} Y \Leftrightarrow c(X) \leq c(Y).$$

МПБ в сучасних системах захисту на практиці реалізується мандатним контролем. Він реалізується на найнижчому апаратно-програмному рівні, що дає змогу доволі ефективно будувати захищене середовище для механізму мандатного контроля. Пристрій мандатного контролю називають монітором звернень.

Мандатний контроль ще називають обов'язковим, оскільки його має проходити кожне звернення суб'єкта до об'єкта, якщо вони знаходяться під захистом СЗІ. Організовується він так: кожен об'єкт O має мітку з інформацією про свій рівень секретності c(O); кожний суб'єкт S також має мітку з інформацією про те, до яких об'єктів він має право доступу c(S). Мандатний контроль порівнює мітки і задовольняє запит суб'єкта S до об'єкта O на читання, якщо c(S)>c(O), і задовольняє запит на запис, якщо c(S)≤c(O). Отже, мандатний контроль реалізує МПБ.

Наведемо низку переваг МПБ, порівняно з ДПБ.

1. Для систем, де реалізовано МПБ, є характерним більш високий ступінь надійності. Це пов'язано з тим, що за правилами МПБ відстежуються не тільки правила доступу суб'єктів системи до об'єктів, але і стан самої АС. Отже, канали витоку в системах такого типу не закладені первісно (що є в положеннях ДПБ), а можуть виникнути тільки за практичної реалізації систем внаслідок помилок розробника.

2. Правила МПБ більш ясні і прості для розуміння розробниками та користувачами ІТС, що також є фактором, який позитивно впливає на рівень безпеки системи.

3. МПБ стійка до атак типу «троянський кінь».

4. МПБ допускає можливість точного математичного доказу, що ця система в заданих умовах підтримує ПБ.

Однак МПБ має дуже серйозні вади – вона є винятково складною для практичної реалізації і вимагає значних ресурсів обчислювальної системи. Це пов'язано з тим, що інформаційних потоків в системі величезна кількість, і їх не завжди можна ідентифікувати.

6.3. Рольова політика безпеки

Рольову політику безпеки (Role Base Access Control – RBAC) не можна віднести ані до дискреційної, ані до мандатної, тому що керування доступом у ній здійснюється як на основі матриці прав доступу для ролей, так і за допомогою правил, які регламентують призначення ролей користувачам та їх активацію під час сеансів. РПБ базується на таких властивостях:

1) усі суб'єкти і об'єкти повинні бути однозначно ідентифіковані;

2) визначено набір ролей в системі;

3) кожній ролі встановлено певний обсяг повноважень;

4) доступ суб'єктів до об'єктів здійснюється за допомогою певних правил у межах певної ролі.

У РПБ класичне поняття **суб'єкт** заміщується поняттями **користувач** і **роль.** Користувач – це людина, яка працює з системою і виконує певні службові обов'язки. Роль – це активно діюче в системі абстрактне поняття, з яким пов'язаний обмежений, логічно зв'язаний набір повноважень, необхідних для здійснення певної діяльності.

У моделі РПБ визначаються такі множини:

- U – множина користувачів;
- R – множина ролей;
- P – множина повноважень на доступ до об'єктів (представляється, наприклад, у вигляді матриці прав доступу);
- S – множина сеансів роботи користувачів з системою.

Для перелічених множин визначаються такі відношення:

$PA \subseteq P \times R$ – відображає множину повноважень на множину ролей, встановлюючи для кожної ролі набір наданих їй повноважень;

$UA \subseteq U \times R$ – відображає множину користувачів на множину ролей, визначаючи для кожного користувача набір доступних йому ролей.

Правила керування доступом рольової політики безпеки визначаються такими функціями:

user: $S \rightarrow U$ – для кожного сеансу s ця функція визначає користувача u, який здійснює цей сеанс роботи з системою: user(s)=u;

roles: $S \rightarrow R$ – для кожного сеансу s ця функція визначає набір ролей з множини R, що можуть бути одночасно доступні користувачу u в цьому сеансі: roles(s)=$\{r|(\text{user}(s),r) \in \text{UA}\}$;

permissions: $S \rightarrow P$ – для кожного сеансу s ця функція задає набір доступних у ньому повноважень, який визначається як сукупність повноважень усіх ролей, що беруть участь в цьому сеансі: permissions(s)=$\{p|(p,r) \in PA\}$.

В якості критерію безпеки рольової моделі використовується таке правило: **система вважається безпечною, якщо будь-який користувач системи *u*, що**

працює в сеансі *s*, може здійснити дії, які вимагають повноважень *p* тільки в тому випадку, коли *p*∈*permissions*(*s*).

6.4. Монітор безпеки

Для здійснення операцій з об'єктами в захищеній ІТС необхідна додаткова інформація (і наявність відповідного об'єкта, що її містить) про дозволені та заборонені операції. Такою компонентною є **монітор безпеки** – компонента КС, яка активізується під час виникнення будь-якого потоку від одного об'єкта до іншого і дає змогу реалізуватися потокам, що належать тільки множині легального доступу L.

МБ повністю бере участь у потоці від об'єкта до об'єкта, і основна його цільова функція – фільтрація інформаційних потоків для забезпечення безпеки КС, тобто фактично – це механізм реалізації ПБ в КС. Множина об'єктів, що входять до складу МБ як компоненти КС, повинна містити підмножину процесів, з якими повинні бути асоційовані всі інші об'єкти КС, і звичайно, хоча б одного користувача. Всі об'єкти КС повинні бути асоційованими з цим користувачем (якого зазвичай називають адміністратором безпеки).

Вибір методів і механізмів залишається за розробником, і єдиною вимогою є реалізація функції захисту, причому для МБ мають виконуватися такі загальні вимоги:

- МБ повинен забезпечувати неперервний і повний захист;
- бути достовірним (захищеним від модифікацій);
- мати невеликі (відносно) розміри.

Точно сформулювати і формально описати необхідні умови реалізації МБ дуже важко. Проте можна описати деякі важливі властивості МБ, якими він свідомо повинен володіти, незалежно від конкретної ПБ. Серед цих властивостей зазначимо такі:

- під час реалізації будь-якої ПБ найважливішим кроком є ідентифікація всіх об'єктів КС. Водночас повинна бути унікальність імен об'єктів, що дає змогу реалізувати механізм ідентифікації і автентифікації (ІА);
- неможливість доступу до об'єктів без участі МБ: якщо в $\forall t \in N_0$ для $\forall p \subseteq A$ об'єкт $O_i \in O_i$ отримав у момент *t* доступ $O_i \xrightarrow{P} O_j, O_j \in O_t$, то $\exists k > 0,\ k \in N_0$ таке, що в момент *t-k*, $t - k \in N_0$, відбувся запит на доступ, який позначатимемо $O_i \xrightarrow{P?} O_j$ (відсутність обхідних шляхів). Запит на доступ можна також вважати одним з видів доступу від об'єкта O_i, до інших об'єктів. Очевидно, що в якості об'єктів-отримувачів доступу до O_i; повинні виступати лише активні об'єкти $U_i \in U, i = 1, \dots, N_U$ і $P_j \in P, j = 1, \dots, N_P$;

- обов’язкова наявність механізму ІА: якщо для $\forall t \in N_0, \forall p \subseteq A, \forall O_i, O_j \in O_t$ виконується $O_i \overset{P?}{\rightarrow} O_j$, то МБ однозначно визначає належність O_i і O_j до відповідних множин *U*, *P* або *O*;
- обов’язкова наявність у МБ дозвільного механізму, тобто під час запиту $O_i \overset{P?}{\rightarrow} O_j$, O_i, $O_j \in O_i$ залежно від належності потоку до підмножин *L* або *F* приймається відповідне рішення, і доступ від O_i до O_j, здійснюється або ні. Належність визначається на основі правил, що декларуються конкретною ПБ (наприклад, для дискреційної ПБ – це матриця доступу, для мандатної ПБ – митковий контроль).

Питання для самоконтролю:

1. Що є основою ДПБ?
2. Які є проблеми у ДПБ?
3. Що є основою мандатної політики безпеки?
4. На яких властивостях базується РПБ?
5. Що таке монітор безпеки?

ТЕМА 7
КРИПТОГРАФІЧНІ МЕТОДИ ЗАХИСТУ ІНФОРМАЦІЇ

У темі наведена інформація про криптографічні методи захисту інформації, для чого використовується криптографія. Біометричні системи ідентифікації, а також несиметричні та симетричні алгоритми шифрування.

Ключова слова: криптологія, криптографія, криптоаналіз, автентифікація, криптографічний захист, цілісність інформації, симетричне та несиметричне шифрування, потокові та блокові шифри, операції шифрування.

План

7.1. Основні положення та визначення.

7.2. Характеристика алгоритмів шифрування.

7.1. Основні положення та визначення

Проблемою захисту інформації шляхом її перетворення займається **криптологія** (*kryptos* – таємний, *logos* – повідомлення). Вона має два напрями: **криптографію** і **криптоаналіз.** Цілі цих двох напрямів прямо протилежні.

Криптографія займається пошуком, дослідженням і розробкою математичних методів перетворення інформації, основою яких є шифрування, а **криптоаналіз** – дослідженням можливості розшифровки інформації.

Основні напрями використання криптографічних методів – це передача конфіденційної інформації через канали зв'язку (наприклад, електронна пошта), встановлення дійсності переданих повідомлень, збереження інформації (документів, баз даних) на носіях у зашифрованому вигляді.

Сучасна криптографія вивчає і розвиває такі напрями:

- симетричні криптосистеми (із секретним ключем);
- несиметричні криптосистеми (з відкритим ключем);
- системи електронного підпису;
- системи керування ключами.

Допомагаючи зберегти зміст повідомлення в таємниці, **криптографію можна використовувати для забезпечення:**

- автентифікації;
- цілісності;
- незаперечності.

Під час **автентифікації** одержувачеві повідомлення потрібно переконатися, що воно виходить від конкретного відправника. Зловмисник не може надіслати фальшиве повідомлення від будь-якого імені. Під час визначення **цілісності** одержувач повідомлення в змозі перевірити, чи були внесені які-небудь зміни в отримане

повідомлення під час його передачі. Зловмисникові не дозволено замінювати дійсне повідомлення на фальшиве.

Незаперечність необхідна для того, щоб відправник повідомлення не зміг згодом заперечувати, що він не є автором цього повідомлення. Сьогодні **автентифікація,** що здійснюється користувачем, **забезпечується** за допомогою:

- смарт-карт;
- засобів біометрії;
- клавіатури комп'ютера;
- криптографії з унікальними ключами для кожного користувача.

Основною областю застосування смарт-карт є ідентифікація користувачів мобільними телефонами.

Біометрія заснована на анатомічній унікальності кожної людини. Біометричні системи ідентифікації приведені на рис. 7.1.

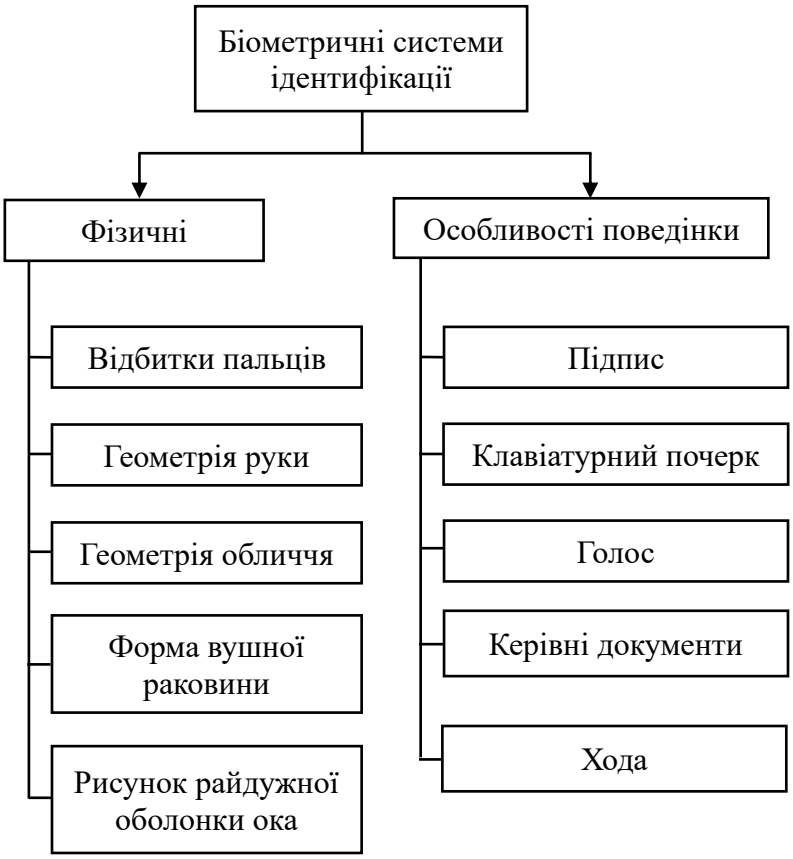

Рис. 7.1. Біометричні системи ідентифікації

Цілісність інформації забезпечується за допомогою криптографічних контрольних сум і механізмів керування доступом і привілеями. У якості криптографічної контрольної суми для виявлення навмисної або випадкової модифікації даних використовується код автентифікації повідомлення – MAC (Message Autentification Code).

Для виявлення **несанкціонованих змін** у переданих повідомленнях можна застосувати:

- електронно-цифровий підпис (ЕЦП), заснований на криптографії з відкритим і закритим ключами;
- програми виявлення вірусів;
- призначення відповідних прав користувачам для керування доступом;
- точне виконання прийнятого механізму привілеїв.

Незаперечність повідомлення підтверджується електронно-цифровим підписом.

7.2. Характеристика алгоритмів шифрування

Криптографічний захист у більшості випадків є більш ефективним і дешевим. Конфіденційність інформації водночас забезпечується шифруванням переданих документів або всього трафіка. Процес криптографічного захисту даних може здійснюватися як програмно, так і апаратно. Апаратна реалізація відрізняється істотно більшою вартістю, однак їй властиві і переваги: висока продуктивність, простота, захищеність тощо. Програмна реалізація більш практична, допускає значну гнучкість у використанні. **Перед сучасними криптографічними системами захисту інформації ставлять такі вимоги:**

1) зашифроване повідомлення має піддаватися читанню тільки за наявності ключа;

2) кількість операцій, необхідних для визначення використаного ключа шифрування за фрагментом шифрованого повідомлення і відповідного йому відкритого тексту, має бути не меншого від загальної кількості можливих ключів;

3) кількість операцій, необхідних для розшифровування інформації шляхом перебору ключів, повинна мати чітку нижню оцінку і виходити за межі можливостей сучасних комп'ютерів (з урахуванням можливості використання мережевих обчислень);

4) знання алгоритму шифрування не має впливати на надійність захисту;

5) незначна зміна ключа має приводити до істотної зміни виду зашифрованого повідомлення навіть під час використання того самого ключа;

6) структурні елементи алгоритму шифрування повинні бути незмінними;

7) додаткові біти, що вводяться в повідомлення в процесі шифрування, повинні бути повністю і надійно сховані в шифрованому тексті;

8) довжина шифрованого тексту повинна бути рівна довжині вихідного тексту;

9) не має бути простих (які легко встановлюються) залежностей між ключами, що послідовно використовуються в процесі шифрування;

10) будь-який ключ із безлічі можливих має забезпечувати надійний захист інформації;

11) алгоритм має допускати як програмну, так і апаратну реалізацію, водночас зміна довжини ключа не повинна призводити до якісного погіршення алгоритму шифрування.

Криптографічний алгоритм, названий алгоритмом шифрування, представлений деякою **математичною функцією,** яка використовується для шифрування і розшифровки. Точніше, таких функцій дві: одна застосовується для шифрування, а інша – для розшифрування.

Розрізняється шифрування двох типів:

- **симетричне** (із секретним ключем);
- **несиметричне** (з відкритим ключем).

У разі **симетричного шифрування** (рис. 7.2) створюється ключ, файл разом із цим ключем пропускається через програму шифрування, отриманий результат пересилається адресатові, а сам ключ передається адресатові окремо, використовуючи інший (захищений або дуже надійний) канал зв'язку. Адресат, запустивши ту ж саму шифрувальну програму з отриманим ключем, зможе прочитати повідомлення. Симетричне шифрування не таке надійне, як несиметричне, оскільки ключ може бути перехоплений, але через високу швидкість обміну інформацією воно широко використовується, наприклад, в операціях електронної торгівлі.

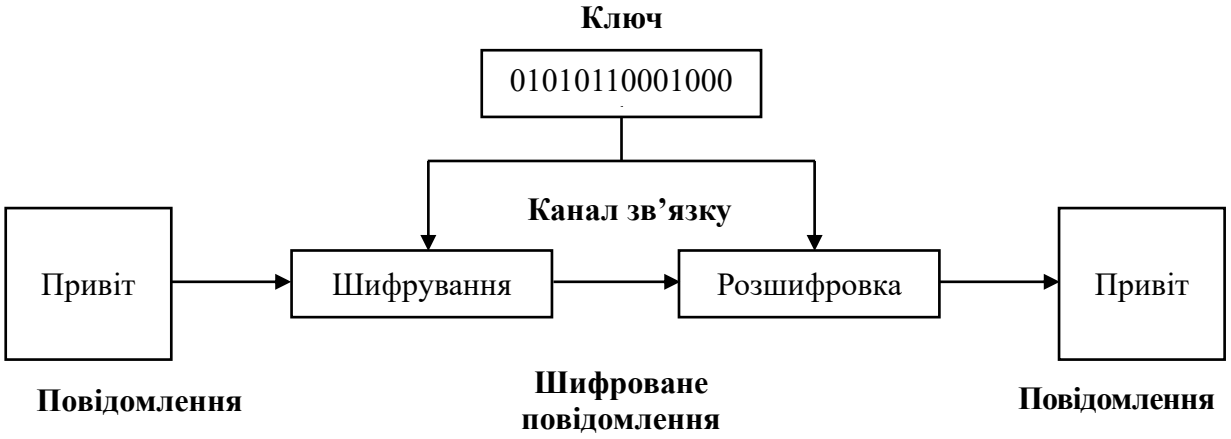

Рис. 7.2. Симетричне шифрування

Несиметричне шифрування складніше, але і надійніше. Для його реалізації (рис. 7.3) потрібні два взаємозалежні ключі: відкритий і закритий. Одержувач повідомляє всім охочим свій **відкритий** ключ, що дає змогу шифрувати для нього повідомлення. **Закритий** ключ відомий тільки одержувачеві повідомлення. Коли комусь потрібно послати зашифроване повідомлення, він виконує шифрування, використовуючи відкритий ключ одержувача. Одержавши повідомлення, останній розшифровує його за допомогою свого закритого ключа. За підвищену надійність несиметричного шифрування приходиться платити: **оскільки обчислення в цьому випадку складніше, то процедура розшифровки займає більше часу.**

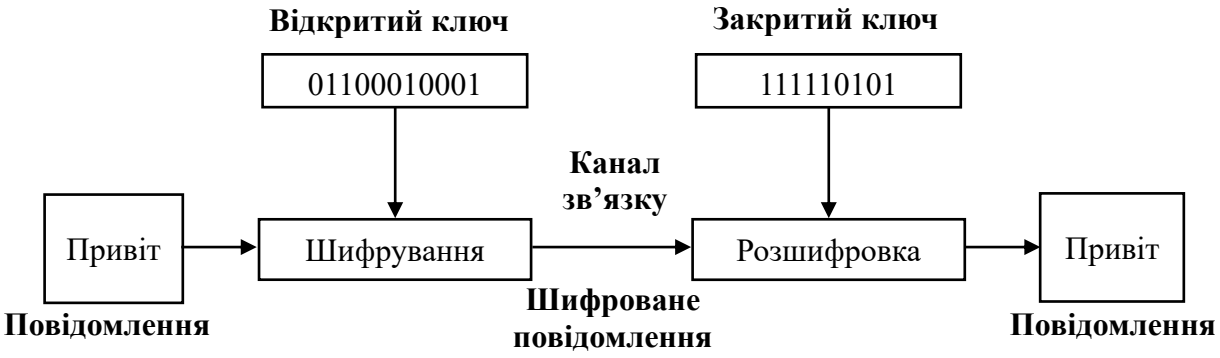

Рис. 7.3. Несиметричне шифрування

Коли надійність криптографічного алгоритму забезпечується завдяки збереженню в таємниці суті самого алгоритму, такий алгоритм шифрування називається **обмеженим.** Обмежені алгоритми становлять значний інтерес з погляду історії криптографії, однак зовсім непридатні за сучасних вимог, які висуваються до шифрування. Адже в цьому випадку кожна група користувачів, що бажають обмінюватися секретними повідомленнями, повинна мати свої оригінальні алгоритми шифрування.

У сучасній криптографії зазначені вище проблеми вирішуються за допомогою використання ключа, який потрібно вибирати серед значень, що належать безлічі (ключовий простір). Функції шифрування і розшифровки залежать від цього ключа. Деякі алгоритми шифрування використовують різні ключі для шифрування і розшифровування. Це означає, що ключ шифрування відрізняється від ключа розшифровування.

Надійність алгоритму шифрування з використанням ключів досягається завдяки їх належному вибору і наступному збереженню в найсуворішому секреті. Це означає, що такий алгоритм не потрібно тримати в таємниці. Можна організувати масове виробництво криптографічних засобів, в основу функціонування яких покладений цей алгоритм. Навіть знаючи криптографічний алгоритм, зловмисник не зможе прочитати зашифровані повідомлення, оскільки він не знає секретного ключа, використаного для його зашифровування.

Симетричні алгоритми шифрування поділяються на:

- потокові;
- блокові.

Алгоритми, у яких відкритий текст обробляється побітно, називаються **потоковими** алгоритмами, або потоковими шифрами. В інших алгоритмах відкритий текст розбивається на блоки, що складаються з декількох бітів. Такі алгоритми називаються **блоковими** або блоковими шифрами. У сучасних комп'ютерних алгоритмах блокового шифрування довжина блоку зазвичай становить 64 біти.

Симетричні алгоритми у разі виявлення в них яких-небудь слабкостей можуть бути дороблені шляхом внесення невеликих змін, а для несиметричних така можливість відсутня.

Симетричні алгоритми працюють значно швидше, ніж алгоритми з відкритим ключем. На практиці несиметричні алгоритми шифрування часто застосовуються в сукупності з симетричними алгоритмами: відкритий текст зашифровується симетричним алгоритмом, а секретний ключ цього симетричного алгоритму зашифровується на відкритому ключі несиметричного алгоритму. Такий механізм називають **цифровим конвертом** (digital envelope).

Найширше сьогодні застосовуються такі алгоритми шифрування:

- DES (Data Encryption Standard) – стандарт шифрування, прийнятий урядом США із 1976 до кінця 1990-х, з часом набув міжнародного застосування;
- Blowfish – Розроблений Брюсом Шнайєром у 1993 р., являє собою шифр на основі мережі Фейстеля. Виконано на простих і швидких операціях: XOR, підстановка, додавання. Не запатентований і вільно поширюваний;
- PGP – комп'ютерна програма, а також бібліотека функцій, що дає змогу виконувати операції шифрування і цифрового підпису повідомлень, файлів та іншої інформації, представленої в електронному вигляді, зокрема прозоре шифрування даних на запам'ятовуючих пристроях, наприклад, на жорсткому диску;
- IDEA (International Decryption-Encryption Algorithm) – симетричний блоковий алгоритм шифрування даних, запатентований швейцарською фірмою Ascom. Відомий тим, що застосовувався в пакеті програм шифрування PGP;
- ГОСТ 28147-89 – радянський і російський стандарт симетричного шифрування, введений у 1990 р., також є стандартом СНД. У 2009 р. ГОСТ 28147-89 перевиданий в Україні під назвою ДСТУ ГОСТ 28147:2009;
- RSA (автори: Rivest, Shamir і Alderman) – це система з відкритим ключем (public-key), призначена як для шифрування, так і для автентифікації, була розроблена в 1977 р. Вона заснована на труднощах розкладання дуже великих цілих чисел на прості множники. RSA – дуже повільний алгоритм. Для порівняння, на програмному рівні DES приблизно в 100 разів швидший від RSA, на апаратному – аж у 1,000 – 10,000 разів, залежно від виконання.

У симетричних криптоалгоритмах (DES, ДСТ, Blowfish, RC5, IDEA) для шифрування і розшифровки інформації використовується той самий секретний ключ. **Перевагами** таких алгоритмів є:

- простота програмної та апаратної реалізації;
- висока швидкість роботи в прямому і зворотному напрямках;
- забезпечення необхідного рівня захисту інформації під час використання коротких ключів.

До основних недоліків цих криптоалгоритмів варто віднести збільшення витрат із забезпечення додаткових заходів таємності під час поширення ключів, а також те, що алгоритм із секретним ключем виконує свою задачу тільки в умовах повної довіри кореспондентів один одному.

У **несиметричних криптоалгоритмах** (RSA, PGP, ECC) пряме і зворотне перетворення виконуються з використанням відкритого і секретного ключів, які не мають взаємозв'язку, що дає змогу за одним ключем обчислити інший. За допомогою відкритого ключа практично будь-який користувач може зашифрувати своє повідомлення або перевірити електронно-цифровий підпис. Розшифрувати таке повідомлення або поставити підпис може тільки власник секретного ключа. Такі алгоритми **дають змогу** реалізувати протоколи типу цифрового підпису, забезпечують відкрите поширення ключів і надійну автентифікацію в мережі, стійку навіть до повного перехоплення трафіка.

Питання для самоконтролю:

1. Основні напрями використання криптографічних методів.
2. Що можна використовувати для виявлення несанкціонованих змін у переданих повідомленнях?
3. Які є типи шифрування?
4. Що називається симетричним шифруванням?
5. Що називається несиметричним шифруванням?

ТЕМА 8
МЕТОДИ ЗАХИСТУ ІНФОРМАЦІЇ В ОПЕРАЦІЙНИХ СИСТЕМАХ

У темі наведена інформація про методи захисту інформації в операційних системах, які використовуються алгоритми шифрування та розшифрування.

Ключові слова: захист інформації, алгоритм симетричного шифрування, шифрування, розшифрування.

План

8.1. Вступ.
8.2. Алгоритм симетричного шифрування DES (Data Encryption Standard).
8.3. Шифрування. Початкова перестановка.
8.4. Операція розгортання ключа.
8.5. Операція розшифрування.

8.1. Вступ

У більшості операційних систем є механізми ідентифікації користувача, які забезпечують той чи інший рівень захисту інформації. Основні методи захисту інформації в операційних системах такі:

- захист інформації за допомогою матриці управління доступом та списків управління доступом;
- захист інформації за допомогою «паролів»;
- захист інформації за допомогою шифрування-дешифрування (криптографія).

Недоліки двох перших методів полягають у тому, що «ключі» доступу зберігаються в самій системі. Це може призвести до того, що підготовлений недобросовісний користувач може їх розкрити і скористатись секретною інформацією.

Під час шифрування інформації ключ кодування не повинен зберігатись у системі. Користувач вводить його тільки тоді, коли зашифровує або розшифровує інформацію.

Питання шифрування-дешифрування інформації є предметом дисципліни під назвою «криптографія». Розроблено низку стандартів, які забезпечують надійний захист інформації. Найбільш поширеними є дві схеми шифрування – DES (Data Encryption Standard) і RSA (отримав назву за першими буквами прізвищ авторів – Rivest, Shamir, Adleman). DES – схема симетрична, в ній для шифрування і дешифрування використовується один і той самий ключ. Схема RSA асиметрична, ключі шифрування і дешифрування в ній різні.

8.2. Алгоритм симетричного шифрування DES (Data Encryption Standard)

Найпоширенішим і найбільш відомим алгоритмом симетричного шифрування є DES (Data Encryption Standard – Стандарт шифрування даних). Алгоритм був розроблений у 1977 р., у 1980 р. був прийнятий NIST (National Institute of Standards and Technolody США) у якості стандарту. DES є класичною мережею Фейстеля з двома множинами (рис. 8.1).

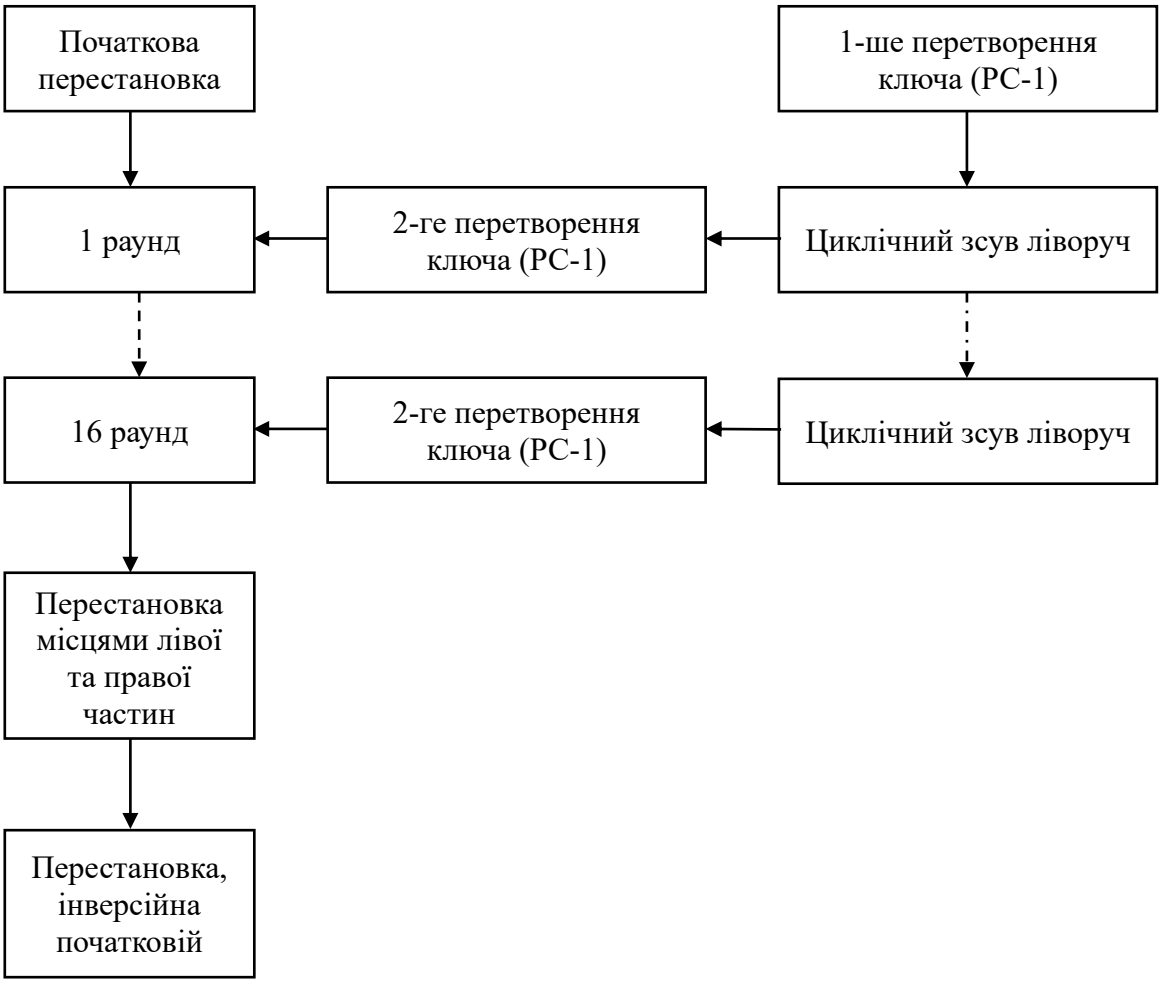

Рис. 8.1. Загальна схема DES

Дані шифруються 64-бітними блоками з використанням 56-бітного ключа. До секретних 56 бітів додається 8 бітів парності, тобто загальна довжина ключа дорівнює 64 біти.

Процес шифрування складається із чотирьох етапів. На першому з них виконується початкова перестановка (IP) 64-бітного вихідного тексту (забілювання), під час якої біти перемішуються відповідно до стандартної таблиці. Наступний етап складається з 16 раундів однієї й тієї ж функції, яка використовує операції зсуву і підстановки. На третьому етапі ліва і права половини виходу останньої (16-ї) ітерації міняються місцями. Нарешті на четвертому етапі виконується пере-

становка IP-1 результату, отриманого на третьому етапі. Перестановка IP-1 обернена до початкової перестановки IP.

8.3. Шифрування. Початкова перестановка

Початкова перестановка та її інверсія визначаються стандартною таблицею. Якщо M – це довільні 64 біти, то X = IP (M) – переставлені 64 біти. Якщо застосувати обернену функцію перестановки:

$$Y = IP^{-1}(X) = IP^{-1}\big(IP(M)\big),$$

то вийде початкова послідовність бітів. Стандартні таблиці IP та IP^{-1}:

Початкова IP-перестановка

58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	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

Кінцева перестановка IP^{-1}

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	25	25

Послідовність перетворень окремого раунду

Розглянемо послідовність перетворень, яка використовується в кожному раунді. 64-бітний вхідний блок проходить через 16 раундів обробки, водночас на кожній ітерації виходить проміжне 64-бітне значення. Ліва і права частини кожного проміжного значення трактуються як окремі 32-бітні значення, позначені L і R. Кожну ітерацію можна описати в такий спосіб:

$$L_i = R_{i-1} \qquad (4.1.)$$
$$Ri = L_{i-1} \oplus \mathrm{F}(R_{i-1}, K_i), s,$$

де $\oplus$ позначає операцію XOR (додавання за модулем 2). Отже, вихід лівої половини L_i дорівнює входу правої половини R_{i-1}. Вихід правої половини R_i є результатом застосування операції XOR до L_{i-1} і функції F, що залежить від R_{i-1} і K_i.

Блок R_i, який подається на вхід функції F, має довжину 32 біти. Спочатку R_i розширюється до 48 бітів з використанням таблиці, яка визначає перестановку і розширення на 16 бітів. Розширення відбувається в такий спосіб: 32 біти розбиваються на групи по 4 біти і потім розширюються до 6 бітів, приєднуючи крайні біти із двох сусідніх груп.

Перестановка з розширенням

31	0	1	2	3	4
3	4	5	6	7	8
7	8	4	10	11	12
11	12	13	14	15	16
15	16	17	18	19	20
19	20	21	22	23	24
23	24	25	26	27	28
27	28	29	30	31	0

У тексті розширення виглядає так. Якщо частина вхідного повідомлення:

... efgh ijkl mnop ...,

внаслідок розширення виходить повідомлення:

... defghi hijklm lmnopq ...

До отриманого в такий спосіб масиву бітів додається за правилами XOR 48-бітний раундовий ключ K_i. Результат подається на вхід блоку заміни, який складається з восьми *S*-боксів, тобто таблиць 4×16, в яких у певний спосіб розміщено десяткові числа від нуля до п’ятнадцяти.

Підстановка виконується у такий спосіб. Масив у 48 бітів розбивається на вісім частин по шість бітів кожна. Кожну частину подають на «свій» *S*-бокс, номер якого визначається її номером. Перший і останній біт 6-бітової частини визначає номер рядка *S*-бокса у двійковому представленні, а чотири середні біти – номер стовпчика. На перетині рядка та стовпчика читаємо 4-бітове число. Воно і буде результатом заміни.

Розглянемо приклад. Припустимо, що перша 6-бітова частина 48-бітового вхідного блоку має значення 110110. Оскільки вона перша, то заміна буде виконуватися на першому *S*-боксі. Він має вигляд:

Перший S-бокс DES

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
2	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

Перша «1» та останній «0» вхідної частини разом (10 дають двійку в десятковому представленні) вказують, що для заміни буде використовуватися рядок № 2. Середні чотири біти (1011) дають у десятковому представленні число 11. Отже, для заміни буде використано стовпчик № 11. На перетині рядка № 2 та стовпчика № 11 знаходиться комірка з числом 7. Вона, точніше, її двійкове представлення 0111, і буде результатом застосування *S*-боксу. Отже, замість 6- бітного числа 110110 отримаємо 0111. Аналогічні виконуються й заміни інших 6-бітних частин вхідного 48-бітного числа.

Далі отримане 32-бітне значення обробляється за допомогою перестановки *P*, метою якої є максимальне перемішування бітів, щоб у наступному раунді шифрування з великою ймовірністю кожен біт оброблявся іншим *S*-боксом:

P-перестановка алгоритму DES

16	7	20	21	29	12	28	17	1	15	23	26	5	18	31	10
2	8	24	14	31	27	3	9	19	13	30	6	22	11	4	25

8.4. Операція розгортання ключа

Раундовий ключ створюється за таким алгоритмом.

Крок 1. Із загального ключа шифрування вилучається кожен восьмий біт (під номерами: 8, 16, 24, 32, 40, 48, 56, 64 – біти парності). Довжина ключа в такий спосіб зменшується до 56 бітів.

Крок 2. Біти ключа розділяються на два блоки C_0 і D_0 відповідно до стандартної таблиці PC-1 (Permuted Choice-1):

Таблиця перемішування бітів ключа PC-1

Блок C_0							Блок D_0						
57	49	41	33	25	17	9	63	55	47	39	31	23	15
1	58	50	42	34	26	18	7	62	54	45	38	30	22
10	2	59	51	43	35	27	14	6	61	53	45	37	29
19	11	3	60	52	44	36	21	13	5	28	20	12	4

Крок 3. На кожному *i*-му раунді C_i та D_i циклічно зсуваються вліво на 1 або 2 позиції, залежно від номера раунду:

Параметри раундового зсуву регістрів C і D

Номер циклу	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Зсув вліво (шифрування)	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1
Зсув вправо (розшифрування)	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

Крок 4. Після зсуву підблоки C_i і D_i об'єднуються та з них за допомогою функції PC-2 (Permuted Choice-2) вибирається 48 бітів раундового підключа K_i. З таблиці PC-2.

Таблиця PC-2 для отримання раундового ключа DES

14	17	11	24	1	5	3	28
15	6	21	10	23	19	12	4
26	8	16	7	27	20	13	2
41	52	31	37	47	55	30	40
51	45	33	48	44	49	39	56
34	53	46	42	50	36	29	32

Вибір бітів виконується так. Підблоки розглядаються як послідовність рядків табл. 4.6, записаних один за одним, починаючи з першого. Біти отриманого в такий спосіб блоку даних перенумеровуються зліва направо, починаючи з одиниці. Кожен елемент S таблиці розглядається як номер біта bS в отриманому блоці даних. Перетворенням є заміна усіх $S \rightarrow bS$.

8.5. Операція розшифрування

Алгоритм DES

Процес розшифрування аналогічний процесу шифрування. На вхід алгоритму подається зашифрований текст, але ключі K_i використовуються в оберненій послідовності: K_{16} використовується на першому раунді, K_1 – на останньому раунді.

Перевагами цієї криптосистеми вважаються:

1) висока швидкодія як в апаратній, так і в програмній реалізації;

2) можливість використання одних і тих самих апаратних або програмних блоків як для шифрування, так і для розшифрування інформації.

Недоліками DES вважають:

1) невелику довжину ключа, усього 56 бітів. За сучасного рівня розвитку комп'ютерних засобів така довжина ключа не може забезпечувати потрібного рівня захисту для деяких типів інформації;

2) наявність «слабких» ключів, викликана тим, що для генерування ключової послідовності виконується два незалежні регістри зсуву.

3) надмірність ключа, що має біти контролю парності.

Алгоритм RSA

Щоб використовувати алгоритм RSA, необхідно спочатку згенерувати відкритий і секретний ключі, виконавши такі кроки:

1. Виберемо два дуже великі прості числа p і q.

2. Визначимо n=p*q.

3. Виберемо велике випадкове число d, яке є взаємно простим з результатом множення (p-1)*(q-1).

4. Визначимо таке число e, для якого істинним є співвідношення (e*d)mod((p-1)*(q-1))=1.

5. Назвемо відкритим ключем числа {e, n}, а секретним ключем числа {d, n}.

Тепер, щоб зашифрувати дані за відкритим ключем {e, n}, необхідно:

1. Розбити текст, що шифрується, на блоки довжиною по n символів і представити кожний символ блоку числом M(i) = 0,1, ..., n – 1.

2. Зашифрувати текст як послідовність чисел $M(i)$ за формулою $C(\mathrm{i}) = (M(\mathrm{i})^e) mod\ n$.

Щоб розшифрувати ці дані з використанням секретного ключа $\{d, n\}$, необхідно виконати такі обчислення:

$$M(i) = (C(i)^d) mod\ n.$$

Тепер необхідно, використовуючи табличні перетворення, за значенням M(i) визначити початковий код символу.

Розроблено також вітчизняний стандарт шифрування даних – **ГОСТ 28147-89.** Однак його програмна реалізація дуже складна і практично не має жодного сенсу через низьку швидкодію.

Питання для самоконтролю:

1. Які є основні методи захисту інформації в операційних системах?

2. Який найпоширеніший і найбільш відомий алгоритм симетричного шифрування?

3. Які переваги та недоліки DES?

4. У чому полягає алгоритм RSA?

ТЕМА 9
АНАЛІЗ БЕЗПЕКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ТА РУЙНІВНЕ ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ

У темі наведена інформація про аналіз безпеки програмного забезпечення (ПЗ) та руйнівне ПЗ, а саме тип руйнівного ПЗ.

Ключові слова: «віруси», «троянський кінь», програми-зломники, безпека, цілісність.

План

9.1. Визначення понять.

9.2. Підкласи РПЗ.

9.1. Визначення понять

Об’єктно-орієнтований аналіз (ООА) спрямований на створення моделей, близьких до реальності. Це методологія, за якою модель формується на основі понять класів і об’єктів, що складають словник предметної області. В ООА клас визначається як множина об’єктів, що пов’язані між собою спільністю структури і поводження.

Означення. Спадкування – це таке відношення між класами, коли один клас (похідний) повторює структуру і поводження іншого (базового). У цьому випадку говорять, що похідний клас успадковує від базового його структуру і поводження.

Означення. Включення – це таке відношення, коли всі члени одного класу є водночас членами іншого.

Базовими для класу **програм** повинні бути класи алгоритмів і даних, тому що програма являє собою сукупність алгоритму, який вона реалізує, і форми представлення цього алгоритму. Відповідно клас програм **успадковує** властивості даних (програми зберігаються у файлі на диску або в деякій області оперативної пам’яті і можуть розглядатися й оброблятися як дані) і алгоритмів (у процесі свого виконання програма реалізує послідовність дій, обумовлену її алгоритмом).

Під системами захисту будемо розуміти програми і фрагменти обчислювальної системи, що забезпечують **безпеку** і **цілісність** обчислювальної системи.

Означення. Нелегітимними будемо називати дії програми або користувача, що призводять до порушень безпеки і/або цілісності системи.

Відмінність поняття легітимності відношень від політики безпеки полягає в тому, що політика безпеки слугує спрощеною моделлю реального розподілу ролей користувачів і функцій програм системи, а легітимність відношень ґрунтується на перевірці порушення основних характеристик обчислювальної системи – цілісності і безпеки.

З урахуванням уведених понять визначимо відношення, які характеризують РПЗ як особливий клас, що базується на класі програм:

1. Нелегітимне використання ресурсів. РПЗ, на відміну від корисних програм, здійснюють нелегітимне споживання ресурсів – захоплення оперативної пам'яті, дискового простору; будь-який РПЗ витрачає процесорний час.

2. Нелегітимний доступ до даних. РПЗ здійснює нелегітимний доступ (читання або запис) до даних. Це одна з основних властивостей РПЗ, якій вони зобов'язані своєю назвою.

3. Нелегітимний запуск програм. Ця властивість належить в основному РПЗ, що функціонують у розвинутих мережних операційних системах (так звані «хробаки»). У цьому випадку РПЗ існують і поширюються не як програми на диску, а як процеси в обчислювальній системі.

4. Нелегітимне виконання програм. РПЗ здійснює конкретні негативні дії. Це властивість, яка є характерною для вірусів.

5. Нелегітимна відмова в обслуговування (порушення доступності). Ця властивість реалізується у випадках, коли виникає можливість ігнорувати запити легітимних користувачів.

9.2. Підкласи РПЗ

Найчастіше виділяють три основні підкласи, пов'язані з класом РПЗ відношенням включення, – **віруси, «троянські коні»,** або «закладки», і **програми-зломники** систем захисту і засобів розмежування доступу. Усі ці підкласи РПЗ характеризуються специфічними відношеннями з об'єктами обчислювальної системи. Зазначимо, що різноманіття видів і типів РПЗ не вичерпується цими трьома класами, просто вони є найбільш розповсюдженими.

Віруси. Підклас РПЗ, що, крім відношень, властивих РПЗ, характеризується ще одним відношенням – **зараженням програм.** Під зараженням програми розуміється така модифікація алгоритму програми, внаслідок якої програма перетворюється в РПЗ. *Існує безліч систематизацій вірусів, заснованих на різних підходах. У розглянутій моделі ці класифікації можуть бути задані за допомогою деталізації відношень з іншими елементами обчислювальної системи.*

«Троянські коні». Цей клас РПЗ характеризується ще одним специфічним типом відношень із класами даних, ресурсів і програм. Це відношення можна назвати відношенням дослідження. «Троянські коні» – це РПЗ, що наносять шкоду після виконання деякої умови спрацьовування. Однак для того, щоб перевірити цю умову, вони повинні досліджувати своє оточення. Звичайно, умовою спрацьовування слугує настання деякого моменту часу або системної події.

Програми-зломники. Цей підклас РПЗ характеризується специфічним відношенням із класом систем захисту, що полягає в подоланні обмежень, що накладаються цими системами.

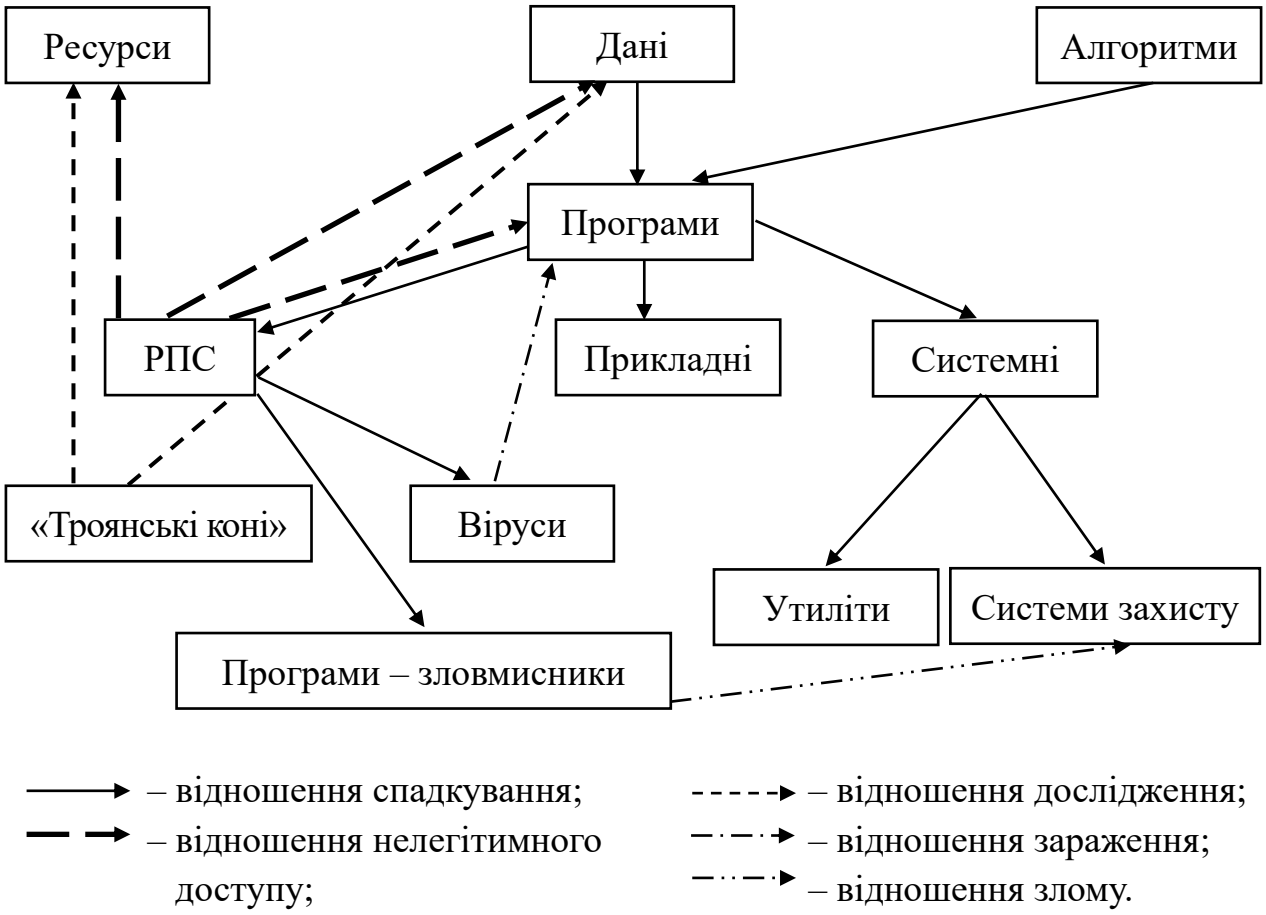

Рис. 9.1. Класи об'єктно-концептуальної моделі предметної області і відношення між ними

Сформулюємо наступний загальний критерій безпеки систем: система, що складається з множини об'єктів *O* і суб'єктів *S*, є безпечною і цілісною на усіх кроках її функціонування, якщо всі наявні в ній до цього кроку відношення між суб'єктами й об'єктами були легітимні, тобто $Ri \in L$.

Питання для самоконтролю:

1. Що таке спадкування?
2. Що розуміється під зараженням програми?
3. Що таке РПЗ «троянський кінь»?
4. Що таке програма-зломник?

ТЕМА 10
МЕТОДИ АНАЛІЗУ БЕЗПЕКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

У темі наведена інформація про методи аналізу безпеки ПЗ, систематизацію методів аналізу безпеки ПЗ, а також формальну постановку задачі аналізу безпеки ПЗ для вирішення її за допомогою контрольно-іспитових методів.

Ключові слова: методи захисту, феномен, ноумен, безпека програми, ПЗ виявлень елементів РПЗ, контрольно-іспитові і логіко-аналітичні методи.

План

10.1. Вступ.
10.2. Методи, що використовуються для аналізу безпеки ПЗ.
10.3. Формальна задача для аналізу безпеки ПЗ.

10.1. Вступ

Під безпекою ПЗ будемо розуміти відсутність у ньому елементів РПЗ. Для доказу безпеки програми потрібно довести, що програма не встановлює нелегітимних відношень з об'єктами обчислювальної системи. З урахуванням уведених понять наведемо формальну постановку задачі аналізу безпеки програм.

Для того, щоб довести, що програма p, яка досліджується, є безпечною, необхідно і достатньо довести, що $p \notin V$. Це, з урахуванням запропонованого визначення РПЗ, означає, що множина відношень Ap, до якої належать усі відношення з об'єктами обчислювальної системи, що установлюються програмою р у процесі її виконання, не містить нелегітимних відношень, тобто $Ap \cap Lp = \varnothing$.

Проте розв'язання цієї задачі утруднюється двома проблемами. По-перше, у загальному випадку неможливо побудувати процедуру розв'язання, що дає змогу визначити легітимність відношення доступу. По-друге, неможливо одержати всі елементи для визначення їх легітимності.

Широко відомі різні засоби ПЗ для виявлення елементів РПЗ – від найпростіших антивірусних програм-сканерів до складних відладчиків і дизасемблерів-аналізаторів, водночас теоретичні дослідження в області методів аналізу безпеки мають трохи відвернений характер. Все ж можна зробити спробу встановити зв'язок між методами, що лежать в основі конкретних засобів, і теоретичними розробками в області аналізу безпеки ПЗ.

10.2. Методи, що використовуються для аналізу безпеки ПЗ

Методи, що використовуються для аналізу безпеки ПЗ, поділяються на дві категорії: **контрольно-іспитові** і **логіко-аналітичні.** В основу цього поділу покладені принципові розходження поглядів на об'єкт (програму), що досліджується – конт-

рольно-іспитові методи аналізу розглядають РПЗ як феномен, а логіко-аналітичні – як ноумен.

Якщо розглядати РПЗ як **феномен,** то задача вирішується в просторі відношень. У такій постановці для доказу того, що досліджувана програма містить РПЗ, необхідно довести, що робочий простір *Ap* програми містить відношення нелегітимного доступу, тобто представити зафіксований факт здійснення нелегітимного доступу до об'єктів обчислювальної системи.

Якщо ж розглядати РПЗ як **ноумен,** то задача вирішується в просторі програм шляхом апроксимації множини РПЗ деякою розв'язною підмножиною. Процес аналізу зводиться до перевірки значення характеристичної функції цієї підмножини для досліджуваної програми. Прикладами реалізації цих методів слугує більшість сучасних засобів пошуку вірусів, що використовують метод пошуку сигнатур або перевірку деякого набору ознак.

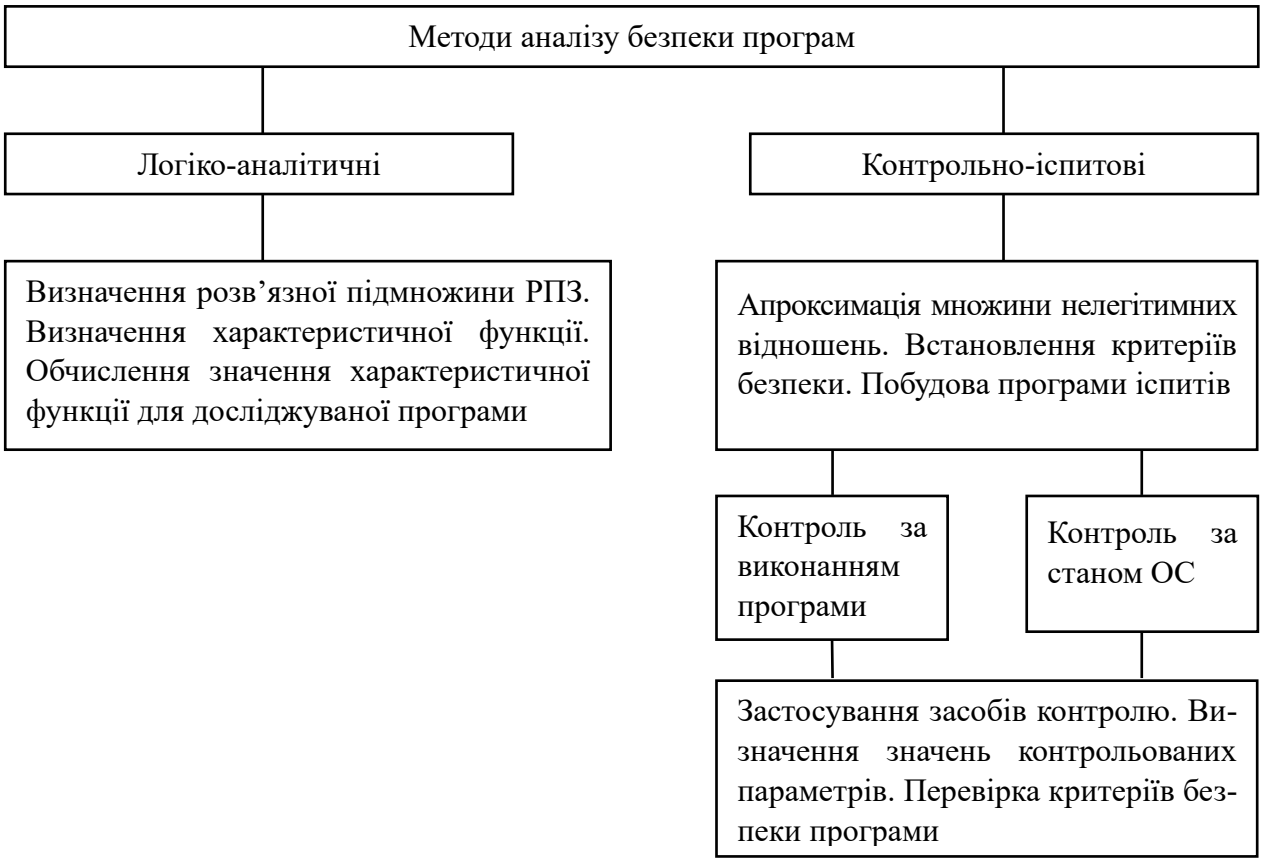

Рис. 10.1. Систематизація методів аналізу безпеки ПЗ

Нехай *Ap* – повна множина відношень доступу до об'єктів обчислювальної системи; *Sp* – несанкціонований доступ; *Lp* – нелегітимний доступ; *Apc* – заборонений доступ.

Водночас критерієм безпеки програми слугує факт реєстрації в під час тестування порушення вимог із безпеки, що пред'являються у системі передбачуваного застосування досліджуваної програми.

10.3. Формальна задача для аналізу безпеки ПЗ

Розглянемо **формальну постановку задачі аналізу безпеки ПЗ для розв'язання її за допомогою контрольно-іспитових методів.**

Нехай задана програма p і обчислювальна система Σ, у якій вона буде функціонувати. Нехай обчислювальна система Σ містить множину об'єктів C_Σ, критичних для її безпеки. Тоді вимоги з безпеки, які має задовольняти програма, можуть бути задані у вигляді множини заборонених відношень p з об'єктами $C\Sigma - A_p^s$. Елементи цієї множини повинні бути задані або в явному вигляді за допомогою перерахування, або у вигляді набору правил, що дає змогу визначити приналежність відношення до цієї множини. Множина C_Σ містить у собі об'єкти всіх типів – ресурси, дані і програми: $C_\Sigma = Rss \cup Dss \cup Pss$.

Схема аналізу безпеки програм контрольно-іспитовими методами представлена на рис. 10.2.

Рис. 10.2. Схема аналізу безпеки програм контрольно-іспитовими методами

Логіко-аналітичні методи вирішують задачу в просторі програм. Це означає, що для доказу того, що програма безпечна, необхідно довести, що вона не належить множині РПЗ V.

Формальна постановка задачі аналізу безпеки **логіко-аналітичними методами** може бути здійснена в такий спосіб.

Обрано деяку систему моделювання програм, у якій кожна програма може бути представлена своєю моделлю, що володіє множиною атрибутів $B = \{bi | i = 1, ..., N\}$. В обраній системі досліджувана програма p представляється своєю моделлю Mp, що характеризується множиною атрибутів $Bp = \{bpi | i = 1, ..., N\}$. У межах цієї системи моделювання повинна бути задана розв'язна підмножина РПЗ $V^* \subset V$, що володіє визначеною на множині атрибутів B характеристичною функцією $\phi(b1, b2, ..., bN)$. Підмножина РПЗ V^* може бути отримана або шляхом побудови моделей усіх відомих РПЗ, або шляхом породження моделей усіх РПЗ, можливих у цій системі моделювання.

Тоді задача аналізу безпеки зводиться до обчислення значення характеристичної функції ϕ на множині атрибутів програми p – якщо $\phi(b_{p1}, b_{p2}, ..., b_{pN})$ істинне, то програма p є РПЗ, що належить підмножині РПЗ V^* ($p \in V^*$); якщо хибне, то програма p не є РПЗ, що належить виділеній розв'язній підмножині РПЗ V^* ($p \notin V^*$).

Структурна схема логіко-аналітичних методів дослідження безпеки програм подана на рис. 10.3.

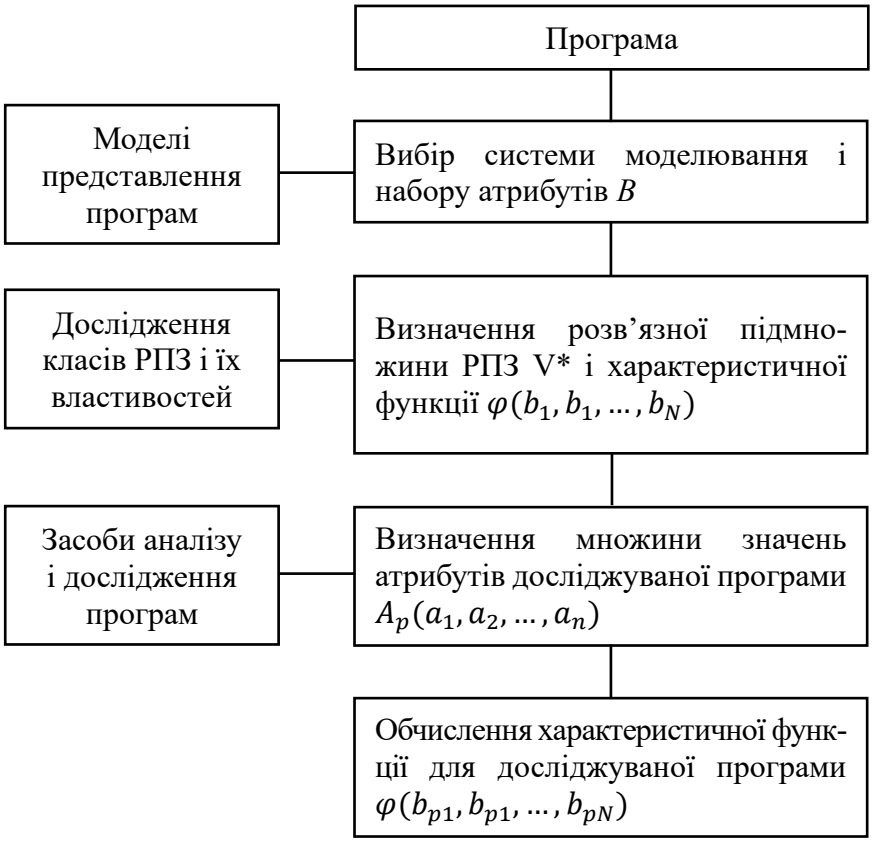

Рис. 10.3. Структурна схема логіко-аналітичних методів дослідження безпеки програм

Застосування методики ООА для побудови концептуальної моделі безпеки ПЗ обчислювальної системи дає змогу:

- представити процес взаємодії компонент обчислювальної системи з погляду безпеки;
- формалізувати властивості РПЗ і створити основу для їх систематизації;
- формалізувати задачу аналізу безпеки ПЗ.

Питання для самоконтролю:

1. На які категорії поділяють методи для аналізу безпеки?
2. Розгляд РПЗ як феномена.
3. Розгляд РПЗ як ноумена.

ТЕМА 11
ПОНЯТТЯ ПРО ХЕШУВАЛЬНІ АЛГОРИТМИ, ЇХ ПРИЗНАЧЕННЯ, ВИМОГИ ДО НИХ

У темі наведена інформація про хешувальні алгоритми, їх призначення, вимоги до них, поняття хеш-функції і які є види хеш-функцій.

Ключові слова: хеш-функція, стійка до колізій хеш-функція, одностороння хеш-функція, безключова хеш-функція, ключова хеш-функція.

План

11.1. Визначення основних понять.

11.2. Класифікація хеш-функцій.

11.1. Визначення основних понять

Особливе місце серед механізмів забезпечення цілісності і автентичності займають функції хешування: безключові та ключові, дають змогу забезпечити широкий спектр послуг із безпеки інформації згідно з ISO 7498. Односторонні хеш-функції визначені в окремому міжнародному стандарті ISO/IEC 10118.

Вибір та реалізація механізмів забезпечення цілісності та справжності інформаційних ресурсів у сучасних автоматизованих системах є одними з важливих етапів проєктування та розробки підсистем захисту інформації. Це пов'язано з постійним зростанням послуг, які надаються різними мережевими службами. Більшість послуг надаються за відсутності фіксованих мережевих адрес клієнтів та їх особливостей, тому ризик порушення цілісності та автентичності інформації збільшується. Для захисту від таких загроз безпеки інформації зазвичай використовують механізми хешування даних – ключові та безключові хеш-функції. Хеш-функції також можуть використовуватись у складі електронного цифрового підпису, який є потужним механізмом забезпечення автентифікації в сучасних автоматизованих системах.

Хеш-функція – це функція h: $D \rightarrow R$, де область визначення $D = \{0,1\}^*$ і область значень $R = \{0,1\}^n$ для деякого $n \geq 1$.

Компресійна функція – це функція $y_I = h(x_I)$, де $D = \{0,1\}^a \times \{0,1\}^b$ і $R = \{0,1\}^n$ для деяких a, b і $n \geq 1$, з $a+b \geq n$.

Ітеративний хеш компресійної функції f: $(\{0,1\}^n \times \{0,1\}^b \rightarrow \{0,1\}^n)$ – це хеш-функція h:($\{0,1\}^b$)$\rightarrow \{0,1\}^n$, визначена як: $h(X_1 \ldots X_t) = H_t = H_i = f(H_i, X_i)$ для 1≤i≤t.

Далі наведені визначення стійкості за прообразом другим прообразом та стійкістю до колізій.

Стійкість за прообразом. Хеш-функція h: $\{0,1\} \rightarrow R$ є стійкою за прообразом ступеня (t,∈), якщо не існує імовірнісного алгоритму I_h, який приймає вхід $Y \in_R Ri$

виводить значення $X \in \{0,1\}^*$ під час виконання не більше *t*, де $h(X) = Y$ з імовірністю щонайменше ε, отриманою випадковими виборами I_h і Y.

Стійкість за другим прообразом. Нехай *S* буде кінцевою підмножиною з $\{0,1\}$. Хеш-функція h: $\{0,1\}^* \rightarrow R$ є стійкою за другим прообразом ступеня (t, ∈, S), якщо не існує імовірнісного алгоритму S_h, який приймає вхід $X \in_R S$ і виводить значення $X' \in \{0,1\}^*$ під час виконання не більше t, де $X' \neq Xh(X') = h(X)$ ймовірністю щонайменше ε, отриманою випадковими виборами S_h і X.

Хеш-функції використовуються як будівельний блок у різних криптографічних додатках. Найбільш важливе їх використання для захисту автентифікації інформації і як інструменту для схем цифрових підписів.

Хеш-функція – це функція, яка відображає вхід довільної довжини в фіксовану кількість вихідних бітів – хеш-значення. Для того, щоб бути корисною в криптографічних додатках, хеш-функція повинна задовольняти деякі вимоги. Хеш-функції можуть поділятися на **односторонні хеш-функції** та **стійкі до колізій хеш-функції.**

Одностороння функція повинна бути стійкою за прообразом і другим прообразом, тобто має бути «важко» знайти повідомлення із заданим хешем (прообразом), або яке хешується в одне і те ж значення, що і задане повідомлення (другий прообраз).

Стійка до колізій хеш-функція – це одностороння хеш-функція, для якої «важко» знайти два різні повідомлення, для яких хеш-значення однакове.

Одностороння хеш-функція – це функція h, яка задовольняє такі умови:

1) аргумент X може бути довільної довжини, а результат h(X) має фіксовану довжину n бітів;

2) хеш-функція повинна бути односторонньою в тому сенсі, що за заданим Y в образі h складно знайти повідомлення *X* таке, що $h(X) = Y$ (стійкі за прообразом) і за заданим повідомленням *X* і значенням $h(X)$ важко знайти повідомлення $X' \neq X$ таке, що $h(X') = h(X)$ (стійкі за другим прообразом).

Стійка до колізій хеш-функція – це функція *h*, яка задовольняє такі умови:

1) аргумент X може бути довільної довжини, а результат $h(X)$ має фіксовану довжину *n* бітів;

2) хеш-функція повинна бути односторонньою, тобто стійкою за прообразом і стійкою за другим прообразом.

Для того, щоб хеш-функція *H* вважалася **криптографічно стійкою,** вона повинна задовольняти три основні вимоги, на яких заснована більшість застосувань хеш-функцій в криптографії:

1) незворотність або стійкість до відновлення прообразу: для заданого значення хеш-функції *m* має бути обчисленням неможливо знайти блок даних *x*, для якого $h(x) = m$;

2) стійкість до колізій першого роду або відновлення другий прообразів: для заданого повідомлення m повинно бути обчисленням неможливо підібрати інше повідомлення n, для якого $h(n) = h(m)$;

3) стійкість до колізій другого роду: має бути обчисленням неможливо підібрати пару повідомлень, що мають однаковий хеш.

Більшість хеш-функцій мають ітеративні конструкції в тому сенсі, що вони базуються на функції компресії з фіксованими входами, вони обробляють кожен блок повідомлення у подібний спосіб. Введення X доповнюється за однозначним правилом доповнення до кратності розміру блоку. Зазвичай це також включає додавання загальної довжини входу в бітах. Доповнений вхід потім ділиться на t блоків, які охоплюють від X_1 до X_t.

11.2. Класифікація хеш-функцій

Методи хешування інформації подані на рис. 11.1.

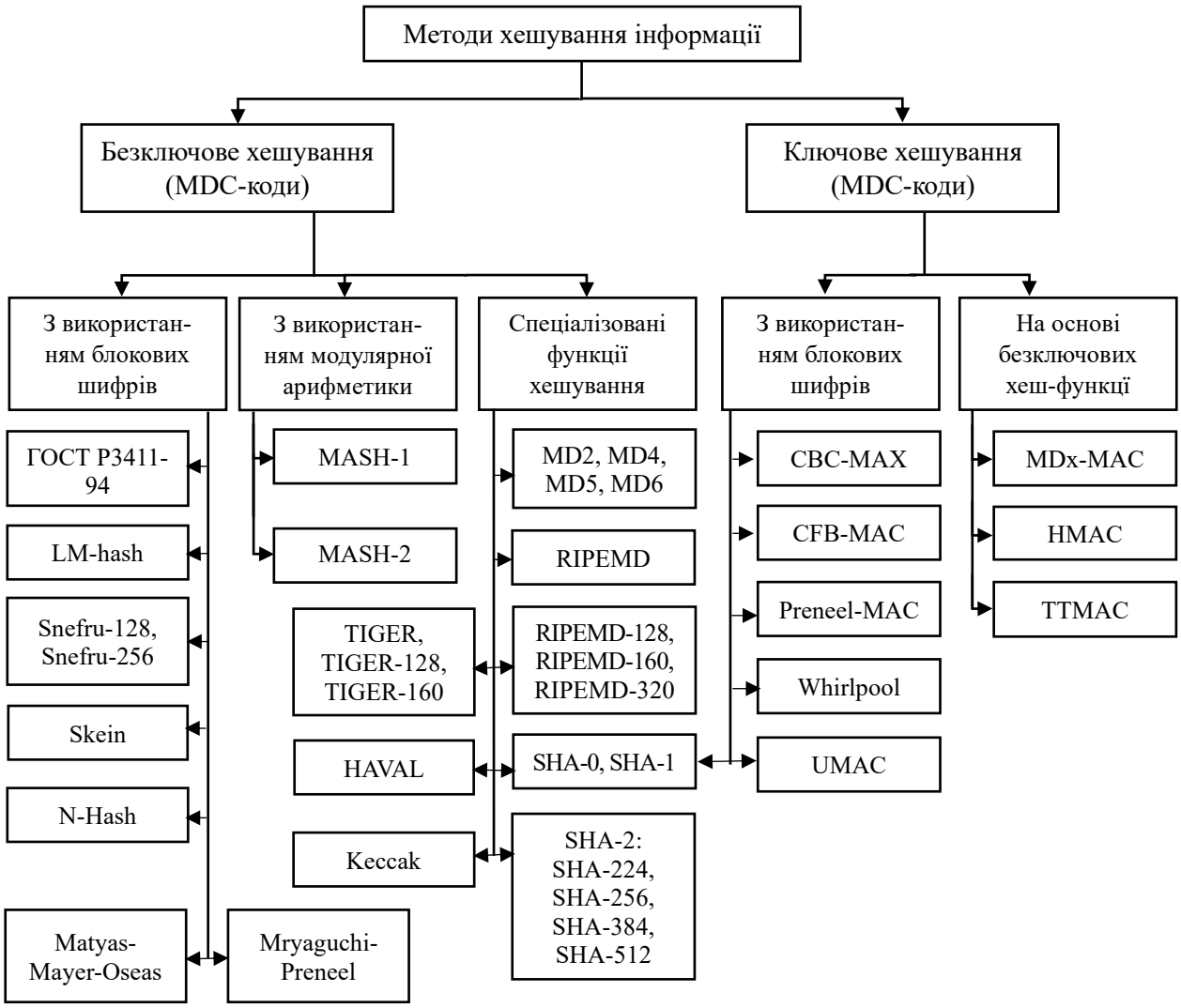

Рис. 11.1. Методи хешування інформації

До безключових хеш-функцій належать коди виявлення змін повідомлення (MDC-код, modification detection code), а також відомі як коди виявлення маніпуляцій над повідомленнями або коди цілісності повідомлень. Суттєвим недоліком безключових хеш-функцій є те, що вони не захищені від можливості підбору такого самого повідомлення з однаковим хешем, і мають відсутність властивості обчислювальної стійкості. Зрештою *MDC*-коди забезпечують, спільно з іншими механізмами, цілісність даних.

До ключових хеш-функцій належать *MAC*-коди.

Визначення автентифікуючих кодів повідомлення, згідно з Пренилем *(Preneel)*: *MAC* – функція h, що задовольняє такі умови:

1. Аргумент X може бути довільної довжини й результат $h(K; X)$, має фіксовану довжину n бітів, де вторинний вхід K позначає секретний ключ.

2. За наявності даних h і X (але з невідомим K) має бути складно визначити $h(K;X)$ з імовірністю успіху значно більшою $1/2^n$. Навіть за великої кількості відомих пар $\{X_i;\mathrm{h}(K; X_i)\}$ складно визначити ключ K або обчислити $h(K; X')$ для будь-якого $X' \neq X_i$.

Більшість *MAC* є повторюваними конструкціями, у тому розумінні, що вони засновані на функції стиску з фіксованим розміром вхідних значень; вони обробляють кожен блок повідомлень аналогічним способом. Вхід X є однозначним заповненням, кратним розміру блоку. Зазвичай це також включає збільшення загальної довжини бітах вхідних значень. Заповнений вхід потім розділяється на t блоків, що позначають X_1 через X_t. *MAC* включає функцію стиску f і єднальну змінну H_i між етапом i–1 і етапом i:

$$H_0 = Z_K;$$
$$H_i = f_k(H_{i-1}, X_i), 1 \leq i \leq t;$$
$$h(K; X) = g_k(H_t).$$

Тут Z позначає початкове значення, а g – вихідне перетворення. Секретний ключ K може бути застосований в Z, у функції стиску, і/або у вихідному перетворенні.

Питання для самоконтролю:

1. Що таке хеш-функція?
2. Що таке компресійна функція?
3. Що таке одностороння хеш-функція?
4. Як класифікують хеш-функції?
5. Що таке стійка до колізій хеш-функція?

ТЕМА 12
ПОНЯТТЯ ПРО ЦИФРОВИЙ ПІДПИС, ВИМОГИ ДО НЬОГО

У темі наведена інформація про цифровий підпис, вимоги до нього, критерії поділу ЕЦП, а також описаний процес побудови ЕЦП.

Ключові слова: цифровий підпис, криптографічний механізм, схеми ЕЦП, відновлення повідомлення, симетрична та асиметрична схеми.

План

12.1. Визначення основних понять.

12.2. Процес побудови схеми ЕЦП.

12.1. Визначення основних понять

Електронний цифровий підпис (ЕЦП) – реквізит електронного документа, призначений для захисту цього електронного документа від підробки, отриманий внаслідок криптографічного перетворення інформації з використанням закритого ключа електронного цифрового підпису, що дає змогу ідентифікувати власника сертифіката ключа підпису, а також установити відсутність перекручування інформації в електронному документі.

Було розроблено зовсім новий криптографічний механізм, який став можливим лише після винайдення асиметричної криптографії. Цей механізм В. Діффі та М. Хеллман назвали цифровим підписом. Його суть пояснимо на прикладі системи RSA.

До повідомлення M застосуємо перетворення за допомогою приватного ключа d і назвемо його *цифровим підписом,* тобто:

$$S = M^d mod\ n.$$

Повідомлення *M* та його електронний підпис *S* відправляють за призначенням. Отримувач, маючи *(M*, *S)* та публічний ключ відправника повідомлення *e,* може перевірити виконання співвідношення:

$$S^e mod\ n = M.$$

Якщо обчислене *M* співпадає з отриманим повідомленням, то підпис справжній.

Ця схема призводить до такого:

- отримувач, перевіривши справжність підпису, впевнений у тому, що це повідомлення *M* сформував саме власник приватного ключа *d* (оскільки більше ніхто не має до нього доступу);
- відправник не зможе відмовитися від цього листа з тієї ж самої причини.

Отже, створюється можливість утворення юридично чинних документів на основі такого механізму електронного підписування. Ця схема має суттєвий недолік: цифровий підпис має ту ж довжину, як і документ, що ним підписаний. Отже, каналом зв'язку пересилається вдвічі більше інформації, ніж це потрібно для самого документа.

Було запропоновано підписувати не саме повідомлення, а його хеш-образ, що значно зменшить навантаження на канали зв'язку.

В Україні всі відношення електронних документів та підписів визначаються Законами України «Про електронні документи та електронний документообіг» та «Про електронний цифровий підпис».

Цифровий підпис повинен мати такі властивості:

1. Повинна бути можливість перевірити автора, дату й час створення підпису.

2. Повинна бути можливість автентифікувати повідомлення під час створення підпису.

3. Необхідно передбачити можливість перевірки підпису третьою стороною для вирішення суперечок.

Сформулюємо такі вимоги до цифрового підпису:

1. Підпис повинен бути бітовим відбитком повідомлення, що підписується.

2. Підпис повинен використовувати деяку унікальну інформацію про відправника для запобігання підробці або відмові.

3. Створювати цифровий підпис повинно бути відносно легко.

4. Повинно бути розрахунково неможливо підробити цифровий підпис як створенням нового повідомлення для наявного цифрового підпису, так і створенням підробленого цифрового підпису для деякого повідомлення.

5. Цифровий підпис повинен бути компактним, аби не перевантажувати канали зв'язку.

За **способом** побудови схеми ЕЦП поділяють на два класи:

- схема ЕЦП із відновленням повідомлення;
- схема ЕЦП із додаванням.

За **кількістю** учасників ЕЦП поділяють на:

- одиночну схему ЕЦП;
- групову схему ЕЦП.

За **способом перевірки** ЕЦП поділяють на:

- інтерактивні схеми ЕЦП, що вимагають протокольної взаємодії;
- неінтерактивні схеми ЕЦП, які не потребують протокольної взаємодії.

Наявні алгоритми ЕЦП можна поділити також **за типами використовуваних односпрямованих функцій** із секретом:

- схеми ЕЦП, засновані на стійкості факторизації великого числа;
- схеми ЕЦП, засновані на стійкості дискретного логарифма;
- схеми ЕЦП, засновані на стійкості дискретного логарифма в групі точок ЕК.

Для опису процесів обробки інформації з використанням механізмів ЕЦП скористаємося такою термінологією.

1. *Алгоритм генерації ЕЦП* – це метод формування ЕЦП.

2. *Алгоритм перевірки (верифікації) ЕЦП* – метод перевірки того, що підпис є автентичним, тобто дійсно створений конкретним об'єктом і не модифікований під час передачі.

3. Схема *ЕЦП (або механізм ЕЦП)* – сукупність взаємозалежних алгоритмів генерації і верифікації цифрового підпису.

4. *Процес (процедура) накладання ЕЦП* – це сукупність математичного алгоритму генерації ЕЦП і методів представлення (форматування) даних, що підписуються.

5. *Процес (процедура) зняття ЕЦП* – сукупність алгоритму верифікації ЕЦП і методів відновлення даних.

12.2. Процес побудови схеми ЕЦП

Для побудови схеми ЕЦП необхідно визначити два алгоритми: алгоритм **генерації** ЕЦП і алгоритм **верифікації** ЕЦП. Алгоритм верифікації доступний для всіх потенційних одержувачів підписаних повідомлень, водночас алгоритм генерації ЕЦП відомий тільки особі, яка підписує, що для деякого повідомлення $m \in M$ визначає відповідний підпис $s \in S$. Верифікатор, одержавши пари (m, s) і деяку відкриту інформацію про особу, що підписує, застосовує відповідний алгоритм верифікації ЕЦП. Цей алгоритм видає двійковий результат: «так», якщо підпис правильний (автентичний) і «ні» – у протилежному випадку.

Наявні сьогодні схеми ЕЦП поділяють на два класи: схеми ЕЦП із відновленням повідомлення та схеми ЕЦП із додаванням повідомлення.

У схемах *ЕЦП* із *відновленням повідомлення* все або частина підписаного повідомлення може бути відновлена безпосередньо з цифрового підпису. Отже, на вхід алгоритму верифікації надходить лише цифровий підпис s.

У схемах ЕЦП із додаванням повідомлення цифровий підпис приєднується до повідомлення й у такому вигляді відправляється адресату. Для верифікації такого ЕЦП необхідно мати і підпис s, і відповідне повідомлення m. Кожна з цих схем може бути **детермінованою** або **рандомізованою.** Застосування детермінованих схем характеризується тим, що цифровий підпис одного і того ж вхідного рядка даних призводить до формування однакових цифрових підписів. У рандомізованій схемі під час генерації підпису використовується деякий випадковий параметр (число), що призводить до формування різних підписів для однакових вхідних рядків (під час використання тих самих ключів). У рандомізованих схемах необхідно забезпечити непередбачуваність випадкових чисел.

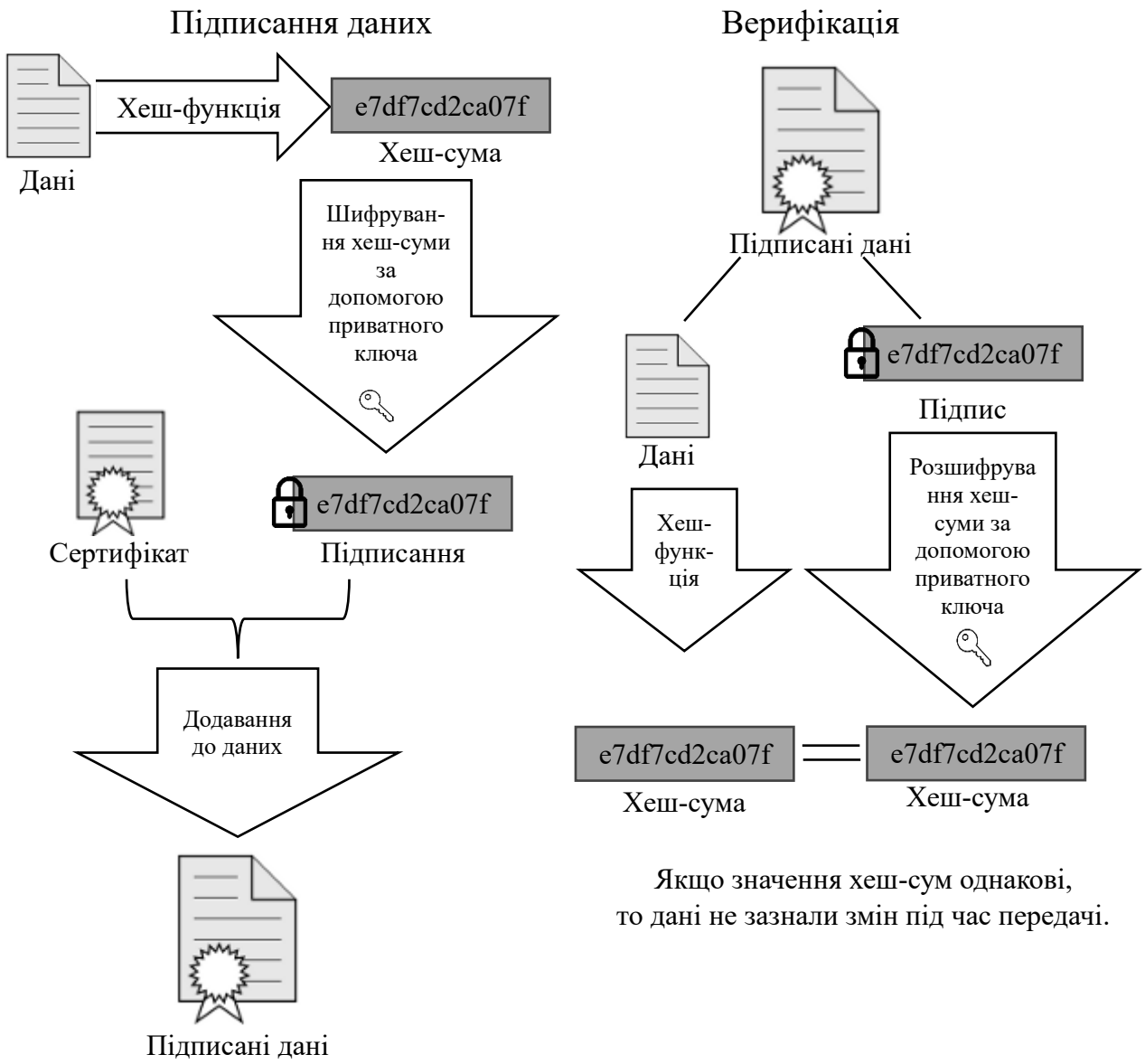

Рис. 12.1. Схеми формування ЕЦП

ЕЦП може бути сформований за допомогою двох схем:

- **симетрична** схема передбачає наявність у системі третьої особи, яка користується довірою обох сторін. Авторизацією документа в цій схемі є сам факт зашифрування електронного документа секретним ключем і передача його третій особі;
- **асиметрична** схема належить до криптосистем з відкритим ключем.

Сьогодні асиметрична схема формування ЕЦП є найбільш поширена і використовується частіше, ніж симетрична схема. Це обумовлено тим фактом, що симетричні схеми для формування і розшифрування підпису використовують один і той самий ключ. Якщо зашифровану інформацію потрібно передавати, то в цьому випадку потрібно передавати і ключ шифрування, а саме це може створити проблему, адже якщо канал передачі не захищений, то ключ може бути викрадений

зловмисником. В асиметричних системах цей недолік відсутній, оскільки кожен учасник має пару ключів: відкритий та секретний, які зв'язані між собою. Водночас формування ЕЦП відбувається за допомогою секретного ключа відправника, а перевірка підпису – за допомогою відкритого ключа, тому необхідність передачі секретного ключа відсутня. Через це асиметрична система має набагато більшу криптостійкість, тому саме їй надають перевагу під час створення ЕЦП.

Загальновизнана схема ЕЦП, заснована на асиметричному алгоритмі охоплює три процеси:

- генерація відкритого та закритого ключів;
- формування підпису;
- перевірка підпису.

Сьогодні існують такі алгоритми створення цифрового підпису: Схема RSA, Эль-Гамаля, DSA, ECDSA, ГОСТ Р 34.10-2001, ДСТУ 4145-2002.

RSA (абревіатура від прізвищ *Rivest, Shamir* та *Adleman*) – криптографічний алгоритм з відкритим ключем, що базується на обчислювальній складності задачі *факторизації* великих цілих чисел.

Алгоритм RSA складається з 4 етапів: генерації ключів, шифрування, розшифрування та розповсюдження ключів.

Безпека алгоритму RSA побудована на принципі складності *факторизації цілих чисел*. Алгоритм використовує два *ключі* – відкритий (public) і секретний (private), разом відкритий і відповідний йому секретний ключі утворюють пари ключів (keypair). Відкритий ключ не потрібно зберігати в таємниці, він використовується для шифрування даних. Якщо повідомлення було зашифровано відкритим ключем, то розшифрувати його можна тільки відповідним секретним ключем.

Генерація ключів:

Для того, щоб згенерувати пари ключів, виконуються такі дії:

1. Вибираються два великі прості числа $\boldsymbol{p}$ і $\boldsymbol{q}$ 512 біт завдовжки кожне.

2. Обчислюється їх добуток $\boldsymbol{n = pq}$.

3. Обчислюється функція Ейлера $\boldsymbol{\varphi(n) = (p-1)(q-1)}$.

4. Вибирається ціле число e таке, що $\boldsymbol{1 < e < \varphi(n)}$ та e взаємно просте з $\varphi(n)$.

5. За допомогою розширеного алгоритму Евкліда знаходиться число $\boldsymbol{d}$ таке, що $\boldsymbol{ed \equiv 1\ (mod\ \varphi(n))}$.

Число $\boldsymbol{n}$ називається модулем, а числа $\boldsymbol{e}$ і $\boldsymbol{d}$ – відкритою й секретною експонентами (англ. *encryption and decryption exponents*) відповідно. Пари чисел $(\boldsymbol{n, e})$ є відкритою частиною ключа, а $(\boldsymbol{n, d})$ – секретною. Числа $\boldsymbol{p}$ і $\boldsymbol{q}$ після генерації пари ключів можуть бути знищені, але в жодному разі не повинні бути розкриті.

Шифрування

Припустимо, що Боб хотів би відправити повідомлення M Алісі. Спочатку він перетворює M у ціле число m так, щоб $0 \leq m < n$ за допомогою узгодженого обо-

ротного протоколу, відомого як схема доповнення. Потім він обчислює зашифрований текст *c,* використовуючи відкритий ключ Аліси *e*, за допомогою рівняння:

$$\mathrm{m} = m^e mod\ n.$$

Це може бути зроблено доволі швидко, навіть для 500-бітних чисел, з використанням модульного зведення в ступінь. Потім Боб передає *c* Алісі.

Розшифрування

Для розшифрування повідомлення Боба *m* Алісі потрібно обчислити таку рівність:

$$\mathrm{m} = c^d mod\ n.$$

Неважко переконатися, що під час розшифрування відновиться вихідне повідомлення:

$$c^d = (m^e)^d = m^{ed}\ (mod\ n).$$

З умови $ed = 1\ (mod\ \varphi(n))$ випливає, що $ed = k\varphi(n) + 1$ для деякого цілого *k*, отже, $m^{ed} = m^{k\varphi(n)+1}\ (mod\ n)$.

Згідно з теоремою Ейлера:

$m^{\varphi(n)} = 1\ (mod\ n)$, тому $m^{k\varphi(n)+1} = m\ (mod\ n); c^d = m\ (mod\ n)$.

Етап	Опис операції	Результат операції
Генерація ключів	Обрати два прості різні числа	p = 3557, q = 2579
	Обчислити добуток	$n = p \cdot q = 3557 \cdot 2579 = 9173503$
	Обчислити функцію Ейлера	$\varphi(n) = (p-1)(q-1) = 9167368$
	Обрати відкриту експоненту	$e = 3$
	Обчислити секретну експоненту	$d = e^{-1}\ (mod\ \varphi(n))$; $d = 6111579$
	Опублікувати *відкритий ключ*	{*e, n*} = {3,9173503}
	Зберегти *секретний ключ*	{*d, n*} = {6111579, 9173503}
Шифрування	Обрати текст для шифрування	m = 111111
	Обчислити шифротекст	$c = E(m) = m^e\ mod\ n =$ $= 111111^3\ \ mod\ 9173503 =$ $= 4051753$
Розшифрування	Обчислити вихідне повідомлення	$m = D(c) = c^d\ mod\ n =$ $= 4051753^{6111579}\ \ mod\ 9173503 =$ $= 111111$

Цифровий підпис

RSA може використовуватися не тільки для шифрування, але й для цифрового підпису. Підпис *s* повідомлення *m* обчислюється з використанням секретного

ключа за формулою: $s = m^d \ mod \ n$. Для перевірки правильності підпису потрібно переконатися, що виконується рівність: $m = s^e \ mod \ n$.

Питання для самоконтролю:

1. Що таке електронний цифровий підпис?
2. Що таке симетрична та асиметрична схеми?
3. Яка термінологія обробки інформації з використанням механізмів ЕЦП?
4. Які властивості повинен мати цифровий підпис?
5. Що таке RSA?

ТЕМА 13
ОСНОВНІ ПОЛОЖЕННЯ КЕРУВАННЯ КЛЮЧАМИ. ЖИТТЄВИЙ ЦИКЛ КРИПТОГРАФІЧНОГО КЛЮЧА

У темі наведена інформація про керування ключами та життєвий цикл криптографічного ключа, описані різні стани ключа, а також процес переходу ключа з одного стану в інший.

Ключові слова: керування ключами, стан очікування, активний стан, постактивний стан, генерація, активізація, деактивизація, реактивізація, знищення, генерація ключа, реєстрація ключа, створення сертифіката ключа, розподіл ключа, інсталяція ключа, зберігання ключа, похідна ключа, архівування ключа, скасування (анулювання) ключа, дереєстрація ключа, знищення ключа, реєстрація користувача, ініціалізація користувача, генерація ключа, реєстрація ключа, нормальне використання, резервування ключа, відновлення ключа, архівування, відновлення ключа, видалення (анулювання) ключа.

План

13.1. Визначення основних понять.
13.2. Життєвий цикл керування ключами.

13.1. Визначення основних понять

Під ***керуванням ключами*** розуміють множину методів і процедур, що здійснюють встановлення і керування ключовими відношеннями між авторизованими об’єктами. Керування ключами включає методи і процедури, які підтримують:

- ініціалізацію системних користувачів усередині домену безпеки;
- генерацію, розподіл та інсталяцію ключового матеріалу;
- керування і контроль використання ключового матеріалу;
- відновлення, анулювання і знищення ключового матеріалу;
- зберігання, резервування / відновлення та архівування ключового матеріалу.

Метою керування ключами є запобігання таким основним погрозам:

- компрометація конфіденційності секретних ключів;
- компрометація автентичності секретних і відкритих ключів;
- неавторизоване використання секретних і відкритих ключів.

Керування ключами здійснюється на основі спеціальної політики безпеки, яка прямо або побічно визначає погрози безпеки. Політика також визначає:

- заходи і процедури, що випливають із застосування технічних і організаційних аспектів керування ключами як автоматичного, так і ручного;
- права, обов’язки і відповідальність кожної зі сторін, що брали участь у керуванні ключами;

- тип і зміст записів, внесених у контрольні журнали (журнали контролю безпеки) щодо настання яких-небудь подій, пов'язаних з безпекою керування ключами.

Криптографічний ключ може перебувати в різних станах, які визначають його життєвий цикл. Стандарт ISO/IEC 11770 розрізняє основні перехідні стани. Основними станами є:

- **стан очікування** (черговий стан) *(pending active)* – стан, у якому ключ не використовується для звичайних операцій;
- **активний стан** *(active)* – стан, у якому ключ використовується для криптографічної обробки інформації;
- **постактивний стан** *(post active)* – стан, у якому ключ може використовуватися тільки для дешифрування або верифікації. Якщо буде потреба використання ключа за призначенням, він переводиться з постактивного стану в активний. Ключ, про який відомо, що він скомпрометований, повинен бути негайно переведений у постактивний стан.

Під час переходу з одного основного стану в інший ключ може перебувати в одному з перехідних станів *(transition).* Такими перехідними станами є:

- ***генерація*** – процес генерації ключа, під час якого відповідно до запропонованих правил генерується ключ;
- ***активізація*** *(activation)* – процес або сукупність процесів, під час яких ключ робиться придатним для використання, тобто переводиться зі стану очікування в активний стан;
- ***деактивизація*** *(deactivation)* – процес або сукупність процесів, що обмежують використання ключа, наприклад, через закінчення терміну дії ключа або його анулювання, що і переводять ключ із активного в постактивний стан;
- ***реактивізація*** *(reactivation)* – процес або сукупність процесів, які дають змогу перевести ключ із постактивного в активний стан для повторного використання;
- ***знищення*** *(destruction)* – завершує життєвий цикл ключа. Схематично поданий взаємозв'язок основних і перехідних станів (рис. 13.1).

Життєвий цикл ключа підтримується одинадцятьма функціями керування ключами (key management services). Коротко охарактеризуємо ці функції.

1. Генерація ключа – забезпечує генерацію криптографічного ключа із заданими властивостями для конкретних криптографічних додатків.

2. Реєстрація ключа – зв'язує ключ із об'єктом (зазвичай тільки відповідні секретні ключі). Об'єкт, що бажає зареєструвати ключ, контактує з адміністратором реєстрації.

3. Створення сертифіката ключа – гарантує взаємозв'язок відкритого ключа з об'єктом і забезпечується вповноваженим органом сертифікації (crtification authority), який генерує відповідні сертифікати.

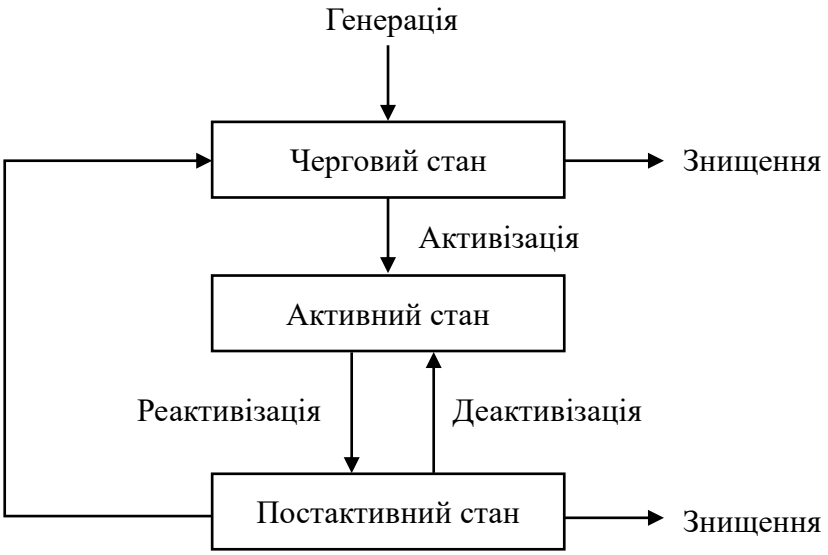

Рис. 13.1. Взаємозв'язок основних і перехідних станів

4. Розподіл ключа (distribute key) – множина процедур безпечного (секретного) забезпечення ключами і пов'язаної з ними інформації вповноважених об'єктів.

5. Інсталяція ключа (install-key) – розміщення ключа в устаткуванні керування ключами у безпечний спосіб і готовим до використання.

6. Зберігання ключа (store-key) – безпечне зберігання ключів для подальшого використання або відновлення ключа для повторного використання.

7. Похідна ключа (derive-key) – формування великої кількості ключів, які називаються похідними ключами, шляхом комбінування секретного вихідного ключового матеріалу, називаного ключем деривації, з несекретними даними на основі використання необоротних процесів.

8. Архівування ключа (archive-key) – забезпечення безпечного зберігання ключів після їх використання. Ця функція використовує функцію зберігання ключа й інші засоби, наприклад, зовнішні сховища.

9. Скасування (анулювання) ключа (revoke-key) (відоме як видалення ключа *(delete key)*) – у випадках компрометації ключа функція забезпечує безпечну деактивізацію ключа.

10. Дереєстрація ключа (deregister-key) – функція реалізується повноважним органом реєстрації, який забирає запис про те, що цей секретний ключ пов'язаний з об'єктом.

11. Знищення ключа (destroy-key) – забезпечує безпечне знищення ключів, у яких минув термін дії. Ця функція включає і знищення всіх архівних копій ключа.

13.2. Життєвий цикл керування ключами

Життєвий цикл керування ключами містить такі основні етапи й процеси.

1. Реєстрація користувача – процес, під час якого об'єкт стає авторизованим членом домену безпеки. Це допускає придбання (створення) і обмін первинним ключовим матеріалом між користувачем і доменом безпеки, наприклад, поділюваним паролем або персональним ідентифікаційним номером (PIN). Усі дії під час реєстрації здійснюються безпечними одноразовими способами, наприклад, через особистий обмін, замовлення поштою, довіреним кур'єром.

2. Ініціалізація користувача – процес, під час якого об'єкт ініціалізує свій криптографічний додаток (наприклад, інсталює та ініціалізує програмне або апаратне забезпечення), включно з використанням або інсталяцією первинного ключового матеріалу, який отриманий під час реєстрації користувача.

3. Генерація ключа. Генерація криптографічних ключів обов'язково повинна включати заходи, спрямовані на забезпечення відповідних властивостей ключа і його випадковості. Ці властивості забезпечуються шляхом використання методів генерації випадкових або псевдовипадкових чисел. Об'єкт може генерувати собі ключі або самостійно, або запитувати їх у довірчої сторони.

4. Інсталяція ключа. Ключовий матеріал інсталюється в програмне або апаратне забезпечення об'єкта за допомогою різних методів, наприклад, ручне введення пароля або PIN, передача даних з використанням диска, постійного запам'ятовувального пристрою, чіп-карти або інших апаратних обладнань (наприклад, завантажника ключа). Первинний ключовий матеріал може слугувати для організації безпечного онлайн-сеансу зв'язку, за допомогою якого вводяться в дію основні робочі криптографічні ключі. Надалі відновлення новим ключовим матеріалом замість використовуваного повинне здійснюватися за допомогою безпечних онлайн-методів відновлення.

5. Реєстрація ключа здійснюється в тісному зв'язку з інсталяцією ключа і полягає в тому, що ключовий матеріал офіційно заноситься під унікальним іменем об'єкта в закриту базу ключів адміністратором реєстрації. Для відкритих ключів адміністратор сертифікації створює сертифікати відкритих ключів, які виступають у ролі гарантів дійсності й цілісності ключа. Сертифікати містяться в довідниках відкритих ключів і є загальнодоступними.

6. Нормальне використання. На цьому етапі ключі перебувають в оперативній доступності для стандартних криптографічних додатків, включно з контролем використання криптографічних ключів. За нормальних умов функціонування системи в період нормального використання ключів збігається із криптоперіодом ключів. У несиметричних криптосистемах ключі з однієї пари можуть перебувати на різних етапах свого існування. Наприклад, у деякий момент часу відкритий ключ шифрування може вважатися недійсним, водночас закритий особистий ключ залишається в активному стані і може використовуватися для розшифрування.

7. Резервування ключа становить резервування ключового матеріалу в незалежному, безпечному сховищі з метою здійснення, якщо буде потреба, відновлення ключа. Резервні копії ключа в основному відправляються на короткострокове зберігання під час нормального використання ключа.

8. Відновлення ключа. Наприкінці криптоперіоду оперативний ключовий матеріал замінюється новим. Це допускає використання комбінації функції генерації ключів, вироблення або установки нових ключів, реалізації двосторонніх протоколів запровадження в дію ключів або організації зв'язку із залученням довірчої третьої сторони. Для відкритих ключів відновлення й реєстрація нових ключів зазвичай допускає реалізацію безпечних комунікаційних протоколів за участю адміністратора сертифікації.

9. Архівування ключового матеріалу, який надалі не буде використовуватися для здійснення криптографічних операцій, здійснюється з метою забезпечення можливості пошуку ключа у разі виникнення особливих умов, наприклад, вирішення спорів, включно з реалізацією функції причетності. Архівування допускає довгострокове автономне зберігання ключа, який виводиться під час цього в постактивний стан.

10. Перед *знищенням ключа* здійснюється його *дереєстрація,* тобто видалення відповідного запису з довідника, внаслідок чого знищується взаємозв'язок значення конкретного ключа з об'єктом. Якщо здійснюється знищення секретного ключа, необхідно забезпечити безпечне і надійне знищення всіх його слідів (залишків).

11. Відновлення ключа здійснюється у випадку, якщо ключовий матеріал був загублений внаслідок непримусової (ненавмисної) компрометації (збої, несправність устаткування, забування (втрата) пароля). У цьому випадку використовуються резервні копії ключа.

12. Видалення (анулювання) ключа передбачає виведення ключа з активного стану в постактивний внаслідок, наприклад, його компрометації. Водночас безпосередньо ключ не знищується. Для відкритих ключів у цьому випадку здійснюється видалення (анулювання) сертифіката. Представлений життєвий цикл керування ключами є найбільш загальним і більш застосовуваним до несиметричних криптосистем. У симетричних криптосистемах керування ключами в загальному випадку менш складне. Так, сеансові ключі можуть не реєструватися, не резервуватися, не віддалятися й не архівуватися.

Питання для самоконтролю:

1. Що розуміється під керуванням ключами?
2. Що є метою керування ключами?
3. Які є стани криптографічного ключа?
4. Описати функції життєвого циклу ключа.
5. Описати етапи та процеси життєвого циклу ключа.

ТЕМА 14
ТЕХНОЛОГІЯ БЛОКЧЕЙНУ

У темі наведена інформація про те, які саме технології використовуються у блокчейні.

Ключові слова: блокчейн, блок транзакцій, хеш, майнер, БД, зберігання даних.

План

14.1. Визначення основних понять.

14.2. Підтвердження транзакцій.

14.1. Визначення основних понять

Блокчейн (англ. *blockchain* або *block chain*) – вибудувана за певними правилами безперервний послідовний ланцюжок блоків, що містять інформацію. Вперше термін з'явився як назва розподіленої бази даних, реалізованої в системі «біткойнів», через що блокчейн часто відносять до транзакцій у різних криптовалютах, проте технологія ланцюжків блоків може бути поширена на будь-які взаємопов'язані інформаційні блоки.

Блок транзакцій – спеціальна структура для запису групи транзакцій у системі біткойнів і аналогічних їй. Транзакція вважається завершеною і достовірною («підтвердженою»), коли перевірені її формат і підписи, і коли сама транзакція об'єднана в групу з декількома іншими і записана у спеціальну структуру – блок. Вміст блоків може бути перевірено, оскільки кожен блок містить інформацію про попередній блок. Всі блоки збудовані в один ланцюжок, яку містить інформацію про всі вчинені коли-небудь операції в базі. Найперший блок у ланцюжку – первинний блок (англ. *genesis block*) – розглядається як окремий випадок, оскільки у нього відсутній батьківський блок.

Блок складається з **заголовка** і **списку транзакцій.** Тема блоку включає в себе свій хеш, хеш попереднього блоку, хеші транзакцій і додаткову службову інформацію. *В системі біткоїнів першою транзакцією в блоці завжди вказується отримання комісії, яка стане нагородою користувачеві за створений блок*. Далі йде список транзакцій, сформований з черги транзакцій, ще не записаних у попередні блоки. Критерій відбору з черги задає майнер самостійно. Це необов'язково повинна бути хронологія за часом. Для транзакцій у блоці використовується **деревоподібне хешування,** аналогічне формування хеш-суми для файла в протоколі BitTorrent. Транзакції, крім нарахування комісії за створення блоку, містять всередині параметра input посилання на транзакцію з попереднім станом даних.

Створений блок буде прийнятий іншими користувачами, якщо числове значення хешу заголовка однакове або нижче певного числа, величина якого періо-

дично коригується. Оскільки результат хешування функції SHA-256 вважається незворотним, зараз немає алгоритму отримання бажаного результату, крім випадкового перебору. Якщо хеш не задовольняє умову, то в заголовку змінюється параметр nonce і хеш перераховується. Зазвичай потрібна велика кількість перерахунків. Коли варіант знайдений, вузол розсилає отриманий блок іншим підключеним вузлів, які перевіряють блок. Якщо помилок немає, то блок вважається доданим у ланцюжок, і наступний блок повинен включити в себе його хеш.

Блоки одночасно формуються множиною «Майнер». Блоки, які задовольняють критерії відправляються в мережу, включаючись в розподілену базу блоків. У кожному з нових блоків можуть зустрічатися як однакові транзакції, так і різні, що увійшли тільки в один з них. Коли ретрансляція блоків відновлюється, Майнер починають вважати головним ланцюжком з урахуванням рівня складності хешу і довжини ланцюжка.

Отже, ланцюжок блоків містить історію володіння, з якою можна ознайомитися, наприклад, на спеціалізованих сайтах.

Розподілена база даних Blockchain формується як ланцюжок блоків із записами про всі транзакції, який безперервно зростає. Копії бази або її частини одночасно зберігаються на безлічі комп'ютерів і синхронізуються відповідно до формальних правил побудови ланцюжка блоків. Інформація в блок не шифрується і доступна у відкритому вигляді, але відсутність змін засвідчується криптографічно через хеш-ланцюжок (елемент цифрового підпису).

База публічно зберігає в незашифрованому вигляді інформацію про всі транзакції, що підписуються за допомогою асиметричного шифрування. Для запобігання багаторазової витрати однієї і тієї ж суми використовуються мітки часу, реалізовані шляхом розбиття БД на ланцюжок спеціальних блоків, кожен з яких, серед іншого, містить у собі хеш попереднього блоку і свій порядковий номер. Кожен новий блок здійснює підтвердження транзакцій, інформацію про які містить, і додаткове підтвердження транзакцій у всіх попередніх блоках ланцюжка. Змінювати інформацію в блоці, який вже знаходиться в ланцюзі, не практично, оскільки в такому випадку довелося б редагувати інформацію в усіх наступних блоках. Завдяки цьому успішна *double-spending*-атака (повторна трата раніше витрачених коштів) на практиці вкрай малоймовірна.

Найчастіше умисна зміна інформації в будь-якій із копій бази або навіть у доволі великій кількості копій не буде визнане істинним, тому що не відповідає правилам. Деякі зміни можуть бути прийняті, якщо будуть внесені в усі копії бази (наприклад, видалення декількох останніх блоків через помилки в їх формуванні).

14.2. Підтвердження транзакцій

Поки транзакція не включена в блок, система вважає, що кількість біткоїнів на якійсь адресі залишається незмінною. У цей час є технічна можливість оформити кілька різних транзакцій із передачі з однієї адреси одних і тих же біткоїнів різним одержувачам. Але як тільки одна з подібних транзакцій буде включена в блок, інші транзакції з цими ж біткоїнами система буде вже ігнорувати. Наприклад, якщо в блок буде включена пізніша транзакція, то більш рання буде вважатися помилковою. Є невелика ймовірність, що під час розгалуження дві подібні транзакції потраплять у блоки різних гілок. Кожна з них буде вважатися правильною, лише у разі відмирання гілки одна з транзакцій стане вважатися помилковою. Водночас не буде мати значення час здійснення операції.

Отже, попадання транзакції в блок є підтвердженням її достовірності незалежно від наявності інших транзакцій з тими ж біткоїнами. Кожен новий блок вважається додатковим «підтвердженням» транзакцій з попередніх блоків. Якщо в ланцюжку 3 блоки, то транзакції з останнього блоку будуть підтверджені 1 раз, а поміщені в перший блок матимуть 3 підтвердження. Достатньо дочекатися декількох підтверджень, щоб звести ймовірність скасування транзакції до мінімуму.

Питання для самоконтролю:

1. Що таке блокчейн?
2. Що таке блок транзакцій?
3. Із чого складається блок?
4. Як формується розподілена база даних Blockchain?
5. Як відбувається підтвердження транзакцій?

ПЕРЕЛІК ПИТАНЬ ДЛЯ ПІДСУМКОВОГО КОНТРОЛЮ

1. Вкажіть, у чому складність створення систем захисту інформації.
2. Опишіть поняття захисту інформації в ІТС та її роботи з організації.
3. Опишіть поняття теорії захисту інформації та її періоди розвитку.
4. Наведіть особливості теорії захисту інформації.
5. Вкажіть, у чому полягають формальні та неформальні підходи до розгляду питань теорії захисту інформації.
6. Вкажіть, які є напрями розвитку теорії захисту інформації.
7. Вкажіть, що собою являє загроза безпеці КС.
8. Вкажіть, які загрози безпеки КС відносять до випадкових.
9. Вкажіть, які загрози безпеки КС відносять до навмисних.
10. Вкажіть, що собою являє загроза розкриття і їх протидія.
11. Вкажіть, що собою являє загроза порушення цілісності і їх протидія.
12. Вкажіть, що собою являє загроза відмови в обслуговуванні.
13. Вкажіть напрями повсякденної діяльності в ІТС для підтримки її працездатності.
14. Вкажіть, якими послугами забезпечується доступність в ІТС.
15. Вкажіть, що собою являє спосіб несанкціонованого доступу та яку мету має зловмисник.
16. Вкажіть, що таке комп'ютерне піратство та категорії порушників безпеки.
17. Вкажіть, що визначає модель порушника безпеки.
18. Опишіть концепцію захисту інформації.
19. Опишіть стратегію захисту інформації та ієрархічний підхід до забезпечення безпеки інформації.
20. Опишіть етапи розробки концепції захисту інформації.
21. Вкажіть поняття політики захисту інформації.
22. Охарактеризуйте правові та організаційно-адміністративні заходи протидії комп'ютерним злочинам.
23. Охарактеризуйте інженерно-технічні заходи протидії комп'ютерним злочинам.
24. Вкажіть комплекс задач під час розробки політики безпеки.
25. Вкажіть правила забезпечення політики безпеки інформації.
26. Опишіть перший етап проєктування та реалізації системи захисту.
27. Вкажіть, які ймовірні загрози виділяють у комп'ютерних мережах.
28. Вкажіть, якими заходами повинна визначатися політика безпеки.
29. Опишіть другий етап проєктування та реалізації системи захисту – реалізацію політики безпеки.

30. Опишіть третій етап проєктування та реалізації системи захисту – підтримку політики безпеки.

31. Опишіть дискреційну політику безпеки.

32. Опишіть переваги та недоліки дискреційної політики безпеки.

33. Опишіть мандатну політику безпеки.

34. Опишіть переваги та недоліки мандатної політики безпеки.

35. Опишіть рольову політику безпеки.

36. Опишіть політику безпеки – монітор безпеки.

37. Вкажіть, що собою являє криптографія.

38. Вкажіть, для забезпечення чого можна використовувати криптографію.

39. Вкажіть, що застосовують для виявлення несанкціонованих змін у переданих повідомленнях.

40. Вкажіть, що собою являє криптографічний захист.

41. Вкажіть, які вимоги ставляться перед криптографічними системами захисту інформації.

42. Опишіть поняття симетричного шифрування.

43. Опишіть поняття несиметричного шифрування.

44. Розкрийте поняття потокових та блокових алгоритмів шифрування.

45. Охарактеризуйте найпопулярніші алгоритми шифрування.

46. Опишіть особливості симетричних криптоалгоритмів.

47. Опишіть особливості несиметричних криптоалгоритмів.

48. Вкажіть, які методи захисту інформації у операційних системах.

49. Зобразіть загальну схему алгоритму шифрування DES.

50. Опишіть алгоритм шифрування DES.

51. Опишіть алгоритм операції розгортання ключа у DES.

52. Вкажіть переваги та недоліки алгоритму шифрування DES.

53. Опишіть алгоритм шифрування RSA.

54. Наведіть означення спадкування та включення у об’єктно-орієнтованому аналізі.

55. Опишіть поняття нелегітимних відношень руйнівного програмного забезпечення.

56. Опишіть основні підкласи, що належать до класу руйнівного програмного забезпечення.

57. Сформулюйте загальний критерій безпеки системи.

58. Опишіть поняття безпеки програмного забезпечення.

59. Опишіть контрольно-іспитовий метод аналізу безпеки ПЗ.

60. Опишіть логіко-аналітичний метод аналізу безпеки ПЗ.

61. Опишіть формальну постановку задачі аналізу безпеки ПЗ за допомогою контрольно-іспитових методів.

62. Наведіть схему аналізу безпеки програм контрольно-іспитовими методами.

63. Опишіть формальну постановку задачі аналізу безпеки ПЗ за допомогою логіко-аналітичних методів.

64. Наведіть схему аналізу безпеки програм логіко-аналітичними методами.

65. Наведіть означення хеш-функції.

66. Дайте означення стійкості за першим та другим прообразом.

67. Дайте означення односторонньої хеш-функції.

68. Вкажіть, коли хеш-функція стійка до колізій.

69. Вкажіть, що собою являють безключові хеш-функції.

70. Вкажіть, що собою являють ключові хеш-функції.

71. Розкрийте поняття електронного цифрового підпису.

72. Наведіть властивості електронного цифрового підпису.

73. Сформулюйте вимоги до електронного цифрового підпису.

74. Опишіть, як класифікуються електронні цифрові підписи.

75. Опишіть алгоритми генерації та верифікації ЕЦП.

76. Опишіть схеми ЕЦП з додаванням та відновленням повідомлення.

77. Розкрийте поняття симетричної та асиметричної схеми ЕЦП.

78. Вкажіть, як використовується RSA для цифрового підпису.

79. Вкажіть, що розуміють під керуванням ключами.

80. Вкажіть, яка мета керування ключами.

81. Вкажіть основні стани криптографічного ключа у життєвому циклі.

82. Вкажіть перехідні стани криптографічного ключа у життєвому циклі.

83. Вкажіть, якими функціями керування ключами підтримується його життєвий цикл.

84. Вкажіть, які етапи та процеси містить життєвий цикл керування ключами.

РЕКОМЕНДОВАНА ЛІТЕРАТУРА

1. Сенів М. М., Яковина В. С. Безпека програм та даних: навч. посіб. Львів: Видавництво Львівської політехніки, 2015. 256 с.

2. Горбенко І. Д., Гриненко Т. О. Захист інформації в інформаційно-телекомунікаційних системах: навч. посіб. Ч. 1. Криптографічний захист інформації. Харків: ХНУРЕ, 2004. 368 с.

3. Лагун А. Е. Криптографічні системи та протоколи: навч. посіб. Львів: Видавництво Львівської політехніки, 2013. 96 с.

4. Остапов С. Е., Євсеєв С. П., Король О. Г. Технології захисту інформації. Харків: Вид. ХНЕУ, 2013. 476 с.

5. Остапов С. Е., Євсеєв С. П., Король О. Г. Кібербезпека: сучасні технології захисту. Львів: «Новий Світ-2000», 2024. 678 с.

6. Програмні технології захисту інформації: конспект лекцій для студентів за напрямом підготовки 6.050103 «Програмна інженерія» факультету інформаційних технологій УжНУ / розр.: В. В. Поліщук. Ужгород: 2018. 80 с.

7. Історія розвитку інформаційних технологій в Україні. URL: http://www.icfcst.kiev.ua/MUSEUM/museum-map_u.html

8. Щотижневик «Мій комп'ютер». URL: http://www.mycomp.com.ua

9. Безкоштовні антивіруси і антивірусні програми для ПК, нетбуків та мобільних телефонів. URL: http://best-free-soft.at.ua

10. Methods and computer tools for identifying diabetes-induced fundus pathology. S. V. Pavlov, T. A. Martianova, L. V. Zagoruiko, Y. R. Saldan, Y. I. Saldan, O. Yu. Pinaieva, Z. Omiotek, K. Dassibekov. *Information Technology in Medical Diagnostics II*, 2019. Taylor & Francis Group, London, UK.

11. Personal Data Protection with Smart Cards Using Eye-Ground Image Recognition Technique. Y. R. Saldan, L. V. Zagoruiko, T. A. Martianova, P. S. Serhiienko, D. V. Chernov, A. K. Zilgaraeva. *Proceedings of the 1st International Workshop on Computational Information Technologies for Risk-Informed Systems (CITRisk 2020)* co-located with XX International scientific and technical conference on Information Technologies in Education and Management (ITEM 2020). Kherson, Ukraine, October 15–16, 2020. P. 144–161.

ДЛЯ НОТАТОК

Навчальне видання

Загоруйко *Любов Василівна*
Дудатьєв *Андрій Веніамінович*

КОНСПЕКТ ЛЕКЦІЙ
з дисципліни
«ТЕХНОЛОГІЇ ПРОГРАМНОГО ЗАХИСТУ ІНФОРМАЦІЇ»

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

Підписано до друку 17.04.2024
Формат 60×84/16. Папір офсетний.
Друк – цифровий. Умовн. друк. арк. 4,65.
Тираж 30. Зам. 5.

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