МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ДВНЗ «УЖГОРОДСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ»

Факультет інформаційних технологій

Кафедра програмного забезпечення систем

КОНСПЕКТ ЛЕКЦІЙ З ДИСЦИПЛІНИ:

«ПРОГРАМНІ ТЕХНОЛОГІЇ ЗАХИСТУ ІНФОРМАЦІЇ»

Ужгород 2018

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

У конспекті лекцій з курсу «Програмні технології захисту інформації» розглянуто теоретичні основи, що входять до складу робочої програми; наведено теми лекційних та лабораторних занять, перелік запитань на підсумковий контроль та список рекомендованої літератури.

Розробник: к.т.н. Поліщук В.В., доцент кафедри програмного забезпечення систем факультету інформаційних технологій ДВНЗ «УжНУ».

Рецензенти:

- к.т.н., доц., декан факультету інформаційних технологій ДВНЗ «УжНУ» Повхан І.Ф.
- к.т.н., доцент кафедри інформатики та фізико-математичних дисциплін факультету інформаційних технологій ДВНЗ «УжНУ» Лях І. М.

Рекомендовано кафедрою програмного забезпечення систем від «24» січня 2018 р., протокол №5.

Рекомендовано Вченою радою факультету інформаційних технологій (протокол №8 від «23» лютого 2018 року).

ЗМІСТ

Мета та завдання навчальної дисципліни

Метою дисципліни «Програмні технології захисту інформації»: закласти термінологічний фундамент, навчити студентів правильно проводити аналіз погроз інформаційній безпеці, основними методам, принципам, алгоритмам захисту інформації в комп’ютерних системах з урахуванням сучасного стану та прогнозу розвитку методів, систем та засобів здійснення погрозу зі сторони потенційних порушників.

Завдання навчальної дисципліни «Програмні технології захисту інформації». У цьому курсі передбачається формування у студентів певних знань та вмінь з теорії та практики захисту інформації та створення програмного забезпечення.

За результатами проведення лекцій з курсу «Програмні технології захисту інформації» студенти повинні:

Знати – сучасні погрози безпеці інформаційним системам; технічні методи і засоби захисту інформації; криптографічні методи захисту інформації; програмні методи і засоби захисту; методи захисту інформації в розподілених інформаційних системах; організаційно-правове забезпечення захисту інформації.

Вміти – аналізувати можливості несанкціонованого здобуття інформації потенційними порушниками; аналізувати вплив комп'ютерних вірусів і шкідливих програм на безпеку комп'ютерних систем; досліджувати стійкість секретних криптографічних систем; досліджувати асиметричні криптосистемі; виявляти дії вірусу в ОС Windows за допомогою аналізу процесів, що протікають, за допомогою аналізу кодів підозрілих програм, за допомогою антивірусних програм; організовувати та виконувати практичні дій посадових осіб відділу захисту інформації відповідно до інструкцій і обов'язків.

Програма навчальної дисципліни

ЗМІСТОВИЙ МОДУЛЬ 1. ПРЕДМЕТ ДИСЦИПЛІНИ. ПОЛІТИКИ БЕЗПЕКИ ІНФОРМАЦІЇ. КРИПТОГРАФІЧНІ МЕТОДИ ЗАХИСТУ

Тема 1. Вступ. Проблеми теорії захисту інформації

Тема 2. Характеристика загроз безпеки інформації

Тема 3. Несанкціонований доступ. Порушники безпеки

Тема 4. Шляхи забезпечення безпеки інформації: Концепція захисту інформації; Стратегія та архітектура захисту інформації; Види забезпечення безпеки інформації.

Тема 5. Політика безпеки інформації: Етапи реалізації систем захисту.

Тема 6. Моделі політики безпеки: Дискреційна політика безпеки; Мандатна політика безпеки; Рольова політика безпеки; Монітор безпеки.

Тема 7. Криптографічні методи захисту інформації: Основні положення та визначення; Характеристика алгоритмів шифрування .

ЗМІСТОВИЙ МОДУЛЬ 2. МЕТОДИ І ЗАСОБИ АНАЛІЗУ БЕЗПЕКИ ПЗ ТА ІНФОРМАЦІЙНИХ СИСТЕМ

Тема 8. Методи захисту інформації в операційних системах.

Тема 9. Об`єктно-концептуальна модель обчислювальної системи та руйнуюче програмне забезпечення.

Тема 10. Модель взаємодії об'єктів обчислювальної системи з погляду безпеки.

Тема 11. Методи аналізу безпеки ПЗ.

Тема 12. Моделі безпеки систем.

Тема 13. Технологія блокчейну.

Теми лабораторних занять

№ з/п	Назва теми
1	Шифрування і розшифрування повідомлень шифрами заміни і перестановки
2	Шифрування і розшифрування повідомлень шифрами Полібія та Гронсфельда
3	Біграмні шифри та шифр Вернама
4	Алгоритм шифрування DES
5	Алгоритм шифрування RSA
6	Семінарське заняття по організаційних заходах щодо захисту інформації та нормативно-правових документів з питань захисту інформації

Тема 1. Вступ. Проблеми теорії захисту інформації

Широке використання інформаційних технологій у всіх сферах життя суспільства робить досить 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 усе більш підсилювати пошуки шляхів доступу до інформації, а інших, відповідно, підсилювати 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.

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 і бази даних.

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

- 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 стосовно деякого фіксованого її стану).

Конфіденційність - це властивість, що вказує на необхідність введення обмежень доступу до даної 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 при

цьому можуть виникнути і які адекватні заходи для її захисту необхідно приймати.

Захищена ІТС – це система, яка у певних умовах експлуатації забезпечує безпеку інформації, яку вона обробляє, і при цьому 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.

Захист інформації в ІТС (information protection, information security, computer system security) – діяльність, що спрямована на забезпечення безпеки оброблюваної в ІТС інформації та ІТС в цілому, і дозволяє 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, при організації ЗІ в ІС можна виділити наступні 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 і т.д.);
- вдосконалення організаційних і організаційно-технічних заходів обробки важливої інформації (охорона об'єктів, організація зберігання носіїв інформації і ін.);
- блокування НСД до інформації (розмежування доступу, системи ідентифікації і автентифікації та ін.).

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

Незважаючи на те, що технології захисту інформаційних систем почала розвиватися відносно недавно, сьогодні вже існує досить багато теоретичних моделей, що дозволяють описувати практично всі аспекти безпеки і

забезпечувати засоби захисту формально підтвердженою алгоритмічною базою.

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 розрізняти **три періоди розвитку теорії захисту інформації** у комп'ютерних системах та мережах: емпіричний, концептуально-емпіричний, теоретико-концептуальний.

Перший – емпіричний період розвитку теорії захисту інформації відрізняє використання неформальних (описових) методів для вирішення задач аналізу СЗІ. Синтез систем захисту інформації при цьому здійснюється методом спроб та помилок з використанням функционально-орієнтованих механізмів захисту. Цей період розпочався з 60-70-х років минулого сторіччя.

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

Третій – теоретико-концептуальний період розвитку теорії захисту інформації характеризується використанням методів формальної теорії захисту інформації для розв'язку 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-і роки минулого сторіччя.

Найбільш характерні особливості теорії захисту інформації на сьогодні полягають у наступному:

1. чітка практична спрямованість – в основному більшість положень, спочатку реалізуються у вигляді конкретних схем і рекомендацій і тільки потім узагальнюються і фіксуються у вигляді теоретичних положень чи методичних рекомендацій;
2. сильна залежність теоретичних розробок від конкретних способів реалізації ІТС, що визначаються проектними 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 напрямків (організаційні заходи, технічний захист, захист від несанкціонованого доступу і т.д.);
4. відсутність системонезалежних теоретичних положень, на основі яких можлива реалізація різних проектів ІТС.

У зв'язку з розвитком інформаційних технологій виникають нові задачі із забезпечення безпеки інформації, підходи до рішення яких на початковому етапі майже завжди носять описовий характер.

В даний час виділяються два основних підходи до розгляду питань теорії захисту інформації безпеки: **неформальний** (або описовий) і **чисто формальний.**

Найбільший розвиток одержали два формальних напрямки, кожний з яких заснований на своєму баченні проблеми безпеки і націлений на рішення певних задач – це **формальне моделювання безпеки** і **криптографія**.

Причому ці різні за походженням і розв'язуваними задачами напрямки доповнюють один одного: криптографія може запропонувати конкретні методи захисту інформації у виді алгоритмів ідентифікації, автентифікації, шифрування і контролю цілісності, а формальні моделі безпеки надають розроблювачам 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 певної множини загроз і складає необхідну (а іноді і достатню) умову безпеки системи.

Формальне вираження політики безпеки називають моделлю політики безпеки. Воно відіграє найважливішу роль у визначенні змісту моделі безпеки. Таким чином, 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 підкорятися поводження системи,

вироблення критерію безпеки і проведення формального доведення відповідності системи цьому критерію при дотриманні встановлених правил і обмежень.

Зв’язок наведених напрямків теорії захисту інформаці можна представити у вигляді наступної схеми (Рис. 1.1).

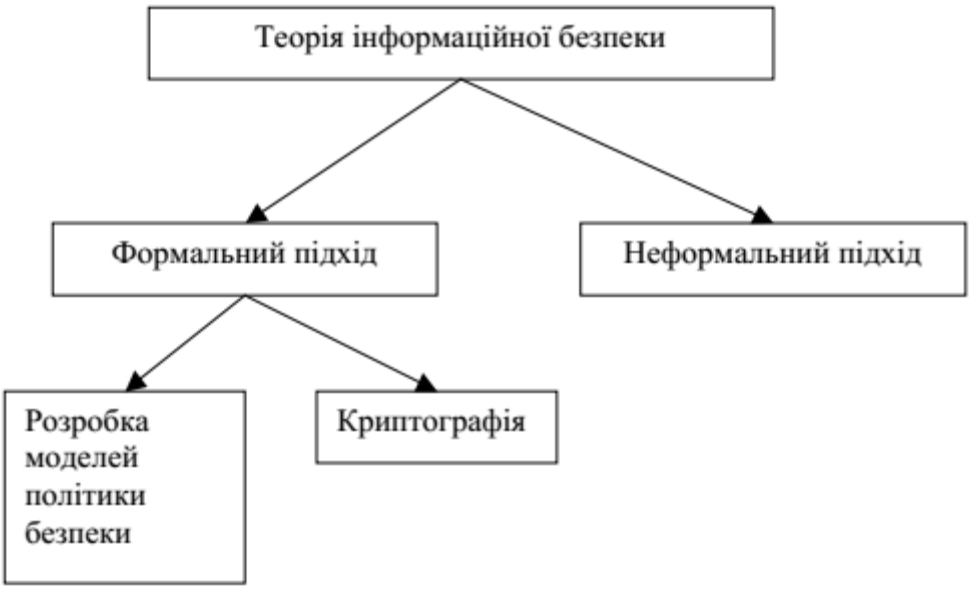


Рис. 1.1. Напрямки розвитку теорії захисту інформації

Значно більш діючим і розповсюдженим поки виявилося використання **неформальних** описових і класифікаційних підходів.

Замість формальних викладок тут використовується різноманітні прийоми категоріювання: порушників (за цілями, кваліфікацією і доступними обчислювальними ресурсами); інформації (за рівнями критичності і конфіденційності); загроз (за способами реалізації, місцями реалізації і т.д.), засобів захисту (за функціональністю і гарантованістю реалізованих можливостей і т.д.) і ін.

Такий підхід не дає точних числових значень показників захищеності, але дозволяє класифікувати ІТС за рівнем захищеності і порівнювати їх між собою. Прикладами таких класифікаційних методик можуть служити 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 узагальнюючий світовий досвід міжнародний стандарт ISO 15408. В Україні також є ряд офіційних нормативних документів, що регламентують всі основні аспекти, зв'язані з безпекою КС і захистом інформації в них від НСД. Однак документів, що регламентують процеси побудови моделей безпеки, немає.

Тема 2. Характеристика загроз безпеки інформації

Під **загрозою безпеки комп'ютерної системи** розуміється подія (вплив), що у випадку своєї 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 як випадковими, так і навмисними. До випадкових загроз відносять:

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 т.д.;
10. Некоректна робота програмного забезпечення;
11. Зміна даних при помилках 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. У зв'язку з цим, 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 программ забезпечує порівняно легкий доступ професіонала до інформації, що знаходиться в персональному комп'ютері.

До навмисних загроз відносять:

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 по каналах зв'язку тощо.

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

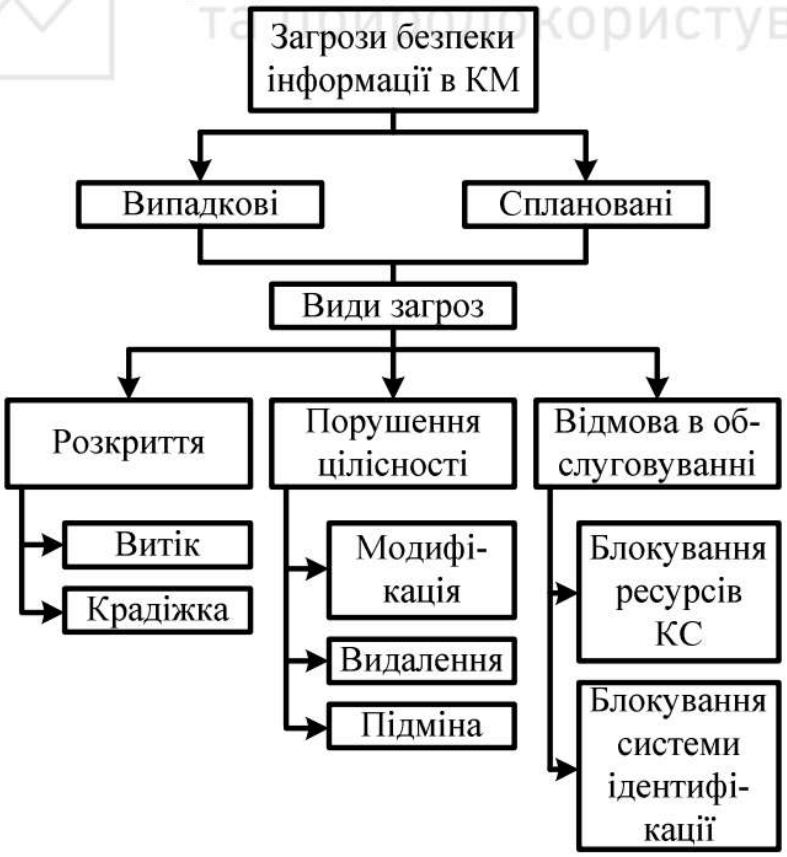


Рис. 2.1. Види загроз безпеки інформації в комп’ютерних мережах

Загроза розкриття полягає в тім, що інформація стає відомою тому, кому не потрібно її знати. Іноді замість слова «розкриття» використовуються терміни «крадіжка» або «витік». (**Означення.** Загроза порушення конфіденційності інформації – цеможливість 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 слід сказати, що протидію загрозам порушення конфіденційності можна забезпечити за допомогою таких послуг:

• довірча конфіденційність – таке управління доступом, при якому засоби захисту дозволяють звичайним користувачам управляти (передають управління) потоками інформації між іншими користувачами і об'єктами свого домена (наприклад, на підставі права власності об'єкту), тобто призначення і передача повноважень не вимагають адміністративного втручання;

• адміністративна конфіденційність – таке управління, при якому засоби захисту дозволяють управляти потоками інформації між користувачами і об'єктами тільки спеціально авторизованим користувачам;

• повторне використання об'єктів – якщо перед наданням об'єкту користувачу або процесу в ньому не залишається інформації, яку він містив, і відміняються попередні права доступу до цього об'єкту;

• аналіз прихованих каналів – проводиться з метою виявлення і перекриття існуючих потоків інформації, які не контролюються іншими послугами;

• конфіденційність при обміні – дозволяє забезпечити безпеку обміну інформацією між захищеними об'єктами в незахищеному середовищі.

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

Серед заходів захисту від порушення цілісності виділяють наступні:

1) своєчасне резервне копіювання цінної інформації;

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

3) введення надмірності в процес обробки інформації, тобто використання аутентифікації, що дозволяє контролювати цілісність об'єктів;

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

Послуги, за допомогою яких забезпечується цілісність, наступні:

• довірча цілісність – аналогічна довірчій конфіденційності;

• адміністративна цілісність – аналогічна адміністративній конфіденційності;

• відкат – дозволяє відновлюватися після помилок користувача, збоїв програмного забезпечення і апаратури і підтримувати цілісність баз даних, додатків і т.д.; забезпечує можливість відміни операції або послідовності операцій і повернути захищений об'єкт в попередній стан;

• цілісність при обміні – дозволяє забезпечити захист об'єктів від несанкціонованої модифікації інформації, що міститься в них, під час їх експорту/імпорту в незахищеному середовищі.

Загроза відмови в обслуговуванні виникає всякий раз, коли у результаті певних дій блокується доступ до деякого ресурсу обчислювальної системи.

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

Можна виділити наступні напрямки повсякденної діяльності в ІТС для підтримки її працездатності:

• підтримка користувачів, тобто консультації і різного роду подання їм допомоги;

• підтримка ПЗ, тобто контроль за ПЗ, яке використовується в ІТС;

• конфігураційне керування, яке дозволяє контролювати зміни в програмній конфігурації;

• резервне копіювання;

• керування носіями, що забезпечує фізичний захист носіїв;

• документування;

• регламентні роботи.

Доступність в ІТС забезпечуєься правильним використанням наступних послуг:

• використання ресурсів – дозволяє користувачам керувати процесами використання послуг і ресурсів;

• стійкість до відмов – покликана гарантувати доступність КС (можливість використання інформації, окремих функцій або КС в цілому) після відмови її компоненту;

• гаряча заміна – дозволяє гарантувати доступність КС в процесі заміни окремих компонентів;

• відновлення після збоїв – забезпечує повернення КС до відомого захищеного стану після відмови в обмірковуванні.

Тема 3. Несанкціонований доступ. Порушники безпеки

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 і т.п.).

При здійсненні несанкціонованого доступу, зловмисник переслідує три мети:

• одержати необхідну інформацію для конкурентної боротьби;

• мати можливість вносити зміни в інформаційні потоки конкурента у відповідності зі своїми інтересами;

• 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**».

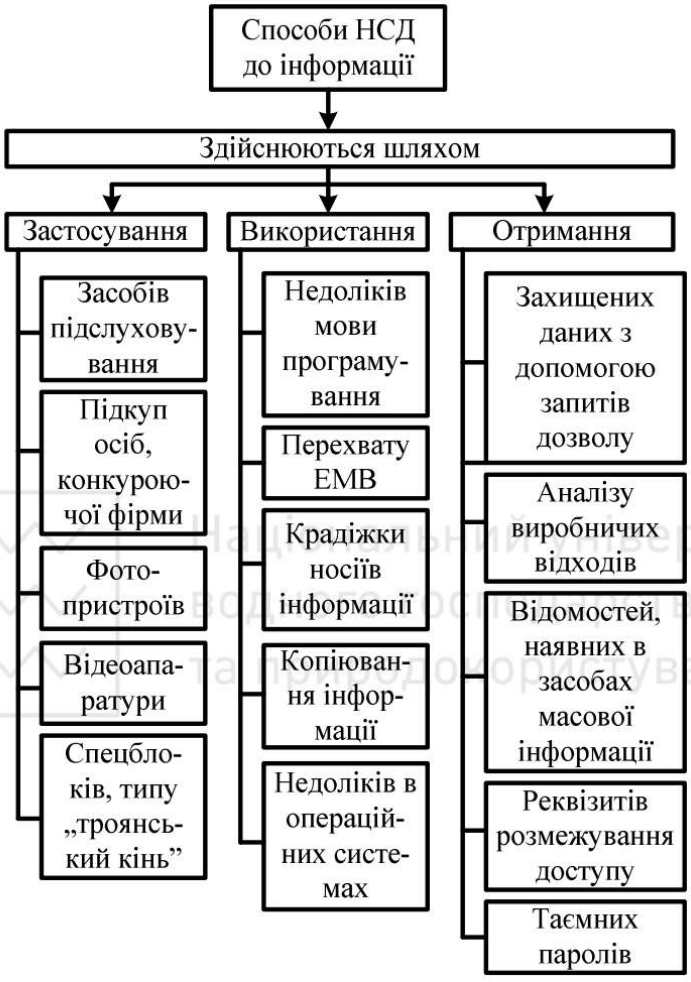


Рис. 3.1. Способи НСД до конфіденційної інформації

Для запобігання можливих загроз, фірми повинні не тільки забезпечити захист операційних систем, програмного забезпечення і 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 є продуманий план атаки. Він вибирає мету випадковим чином і звичайно 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.

Ідейний «хакер» - це той же 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 Webсервера або, рідше, блокування роботи ресурсу, що атакується. У порівнянні із шукачем пригод, ідейний «хакер» розповідає про успішні атаки набагато більшій аудиторії, звичайно розміщаючи інформацію на хакерському Webвузлі.

«Хакер»-професіонал має чіткий план дій і спрямовує його на визначені ресурси. Його атаки добре продумані і, звичайно, здійснюються у кілька етапів. Спочатку він збирає попередню інформацію (тип ОС, надані сервіси і міри захисту). Потім він складає план атаки з урахуванням зібраних даних і підбирає (або навіть розробляє) відповідні інструменти. Далі, провівши атаку, він одержує закриту інформацію і, нарешті, знищує всі сліди своїх дій. Такий професіонал звичайно 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 промисловий шпигун, до того ж, його присутність звичайно складніше знайти. Крім того, йому доводиться долати не зовнішній захист мережі, а тільки, як правило, менш жорсткіший, внутрішній. Він не такий витончений у способах атаки, як промисловий шпигун, і тому частіше допускає помилки, і тим самим може видати свою присутність.

Модель порушника визначає:

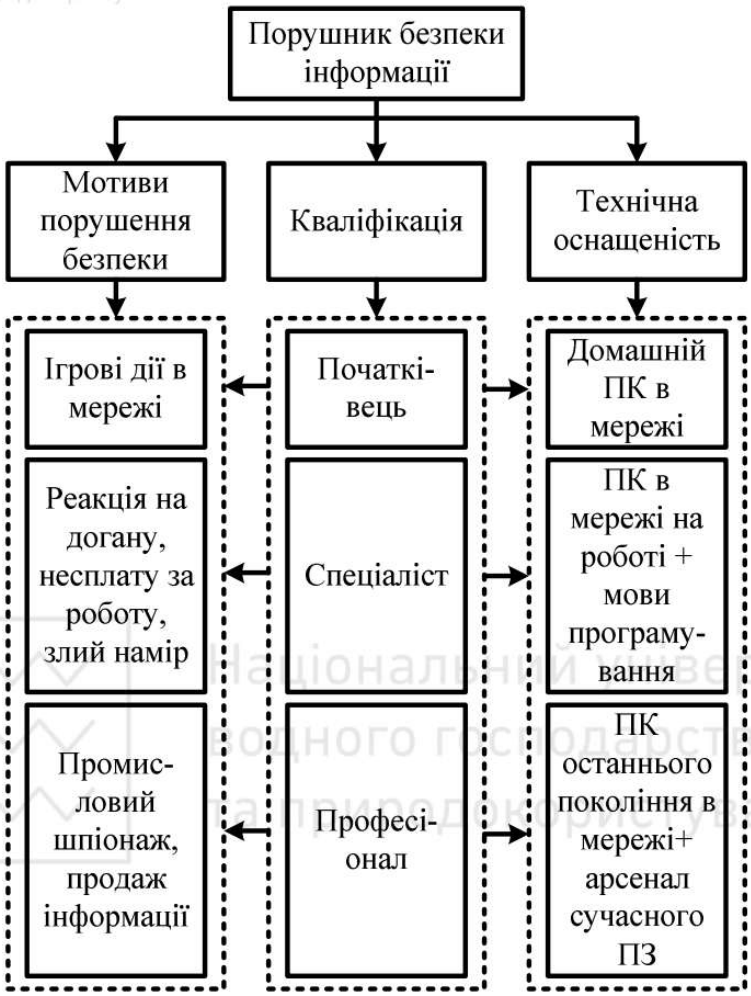
• категорії осіб, у числі яких може виявитися порушник;

• можливі цілі порушника і їх градації за ступенем важливості та небезпеки;

• припущення про його кваліфікації;

• оцінка його технічної озброєності;

• обмеження і припущення про характер його дій.



Сьогодні, зі стрімким розвитком Internet, «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 організації піддаються атакам кілька разів у тиждень, а деякі навіть щодня. Виходять такі атаки не завжди ззовні, 70% спроб зловмисного проникнення в комп'ютерні системи мають джерело всередині самої організації.

Тема 4. Шляхи забезпечення безпеки інформації

Концепція захисту інформації

Вразливість інформації в автоматизованих комплексах обумовлена великою концентрацією обчислювальних ресурсів, їх територіальною розподіленністю, 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 рішення 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 і розробити відповідну документацію.

Концепція захисту інформації - офіційно прийнята система поглядів на проблему інформаційної безпеки і шляхи її рішення 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 рішення формуються конкретні плани забезпечення інформаційної безпеки.

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 напрямки:

- 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).

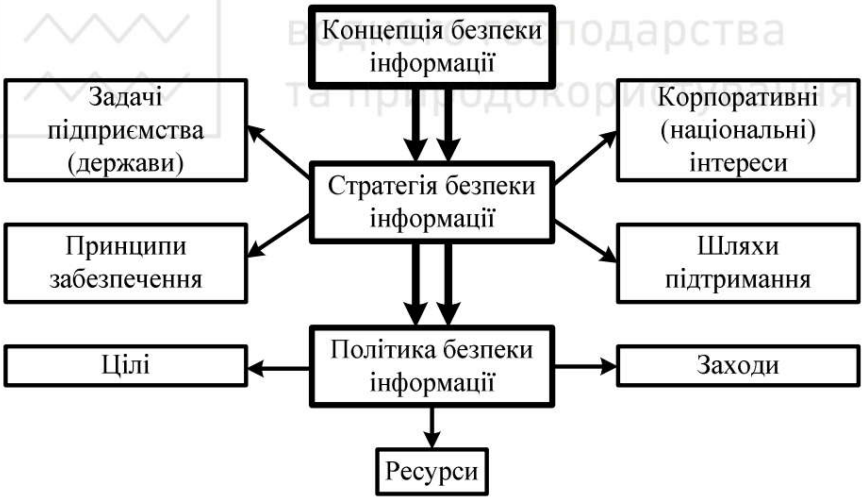


Рис. 4.1 Ієрархічний підхід до забезпечення безпеки інформації

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

Рис. 4.2 Етапи розробки концепції захисту інформації

На першому етапі повинна 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 повинен бути проведений аналіз злочинних дій, що потенційно можуть бути зроблені стосовно об'єкта, що захищається. Важливо визначити ступінь реальної небезпеки таких найбільш широко розповсюджених злочинів, як 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, у тому числі місцевих специфічних умов, виробничих процесів, уже встановлених технічних засобів захисту.

Концепція захисту повинна містити перелік організаційних, технічних і інших заходів, що забезпечують максимальну безпеку при заданому залишковому ризику і мінімальні витрати на їх реалізацію.

Політика захисту - це загальний документ, де перераховуються 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. При цьому, документ може і не включати список необхідних закупок, але вибір конкретних компонентів після його складання повинен бути очевидним.

Політика захисту повинна обов'язково включати наступне:

1. контроль доступу (заборона на доступ користувача до матеріалів, якими йому не дозволено користуватися);
2. ідентифікацію та аутентифікацію (використання паролів або інших механізмів для перевірки статусу користувача);
3. облік (запис усіх дій користувача в мережі);
4. контрольний журнал (журнал дозволяє 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).

Види забезпечення безпеки інформації.

В даний час 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 підрозділити на:

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

До правових заходів варто віднести розробку норм, що встановлюють відповідальність за комп'ютерні злочини, захист авторських

прав програмістів, удосконалювання карного і цивільного законодавства, а також судочинства.

До організаційно-адміністративних заходів відносяться: охорона комп'ютерних систем, підбір персоналу, виключення випадків ведення особливо важливих робіт тільки однією людиною, наявність плану відновлення працездатності центру після виходу його з ладу, обслуговування обчислювального центру сторонньою організацією або особами, незацікавленими в приховуванні фактів порушення роботи центру, універсальність засобів захисту від усіх користувачів (включаючи вище керівництво), покладання відповідальності на осіб, що повинні забезпечити безпеку центру, вибір місця розташування центру і т.п.

До інженерно-технічних заходів можна віднести:

1. захист від несанкціонованого доступу до комп'ютерної системи,
2. резервування важливих комп'ютерних систем,
3. забезпечення захисту від розкрадань і диверсій,
4. резервне електроживлення, розробку і реалізацію спеціальних програмних і апаратних комплексів безпеки тощо.

Фізичні засоби містять у собі різні інженерні засоби, що перешкоджають фізичному проникненню зловмисників на об'єкти захисту, що захищають персонал (особисті засоби безпеки), матеріальні засоби і фінанси, інформацію від протиправних дій.

До **апаратних** засобів відносяться прилади, пристрої, пристосування та інші технічні рішення, які використовуються в інтересах забезпечення безпеки. У практиці діяльності будь-якої організації знаходить широке застосування різна апаратура: від телефонного апарату до розроблених автоматизованих інформаційних систем, що забезпечують її виробничу діяльність. Основна задача апаратних засобів - стійка безпека комерційної діяльності.

Програмні засоби - це спеціальні програми, програмні комплекси і системи захисту інформації в інформаційних системах різного призначення і засобах обробки даних.

Криптографічні засоби - ця спеціальні математичні та алгоритмічні засоби захисту інформації, переданої по мережах зв'язку, збереженої та обробленої на комп'ютерах з використанням методів шифрування.

Тема 5. Політика безпеки інформації

Розробка політики безпеки інформації повинна проводитися з урахуванням задач, рішення яких забезпечить реальний захист даного об'єкта (рис. 5.1).

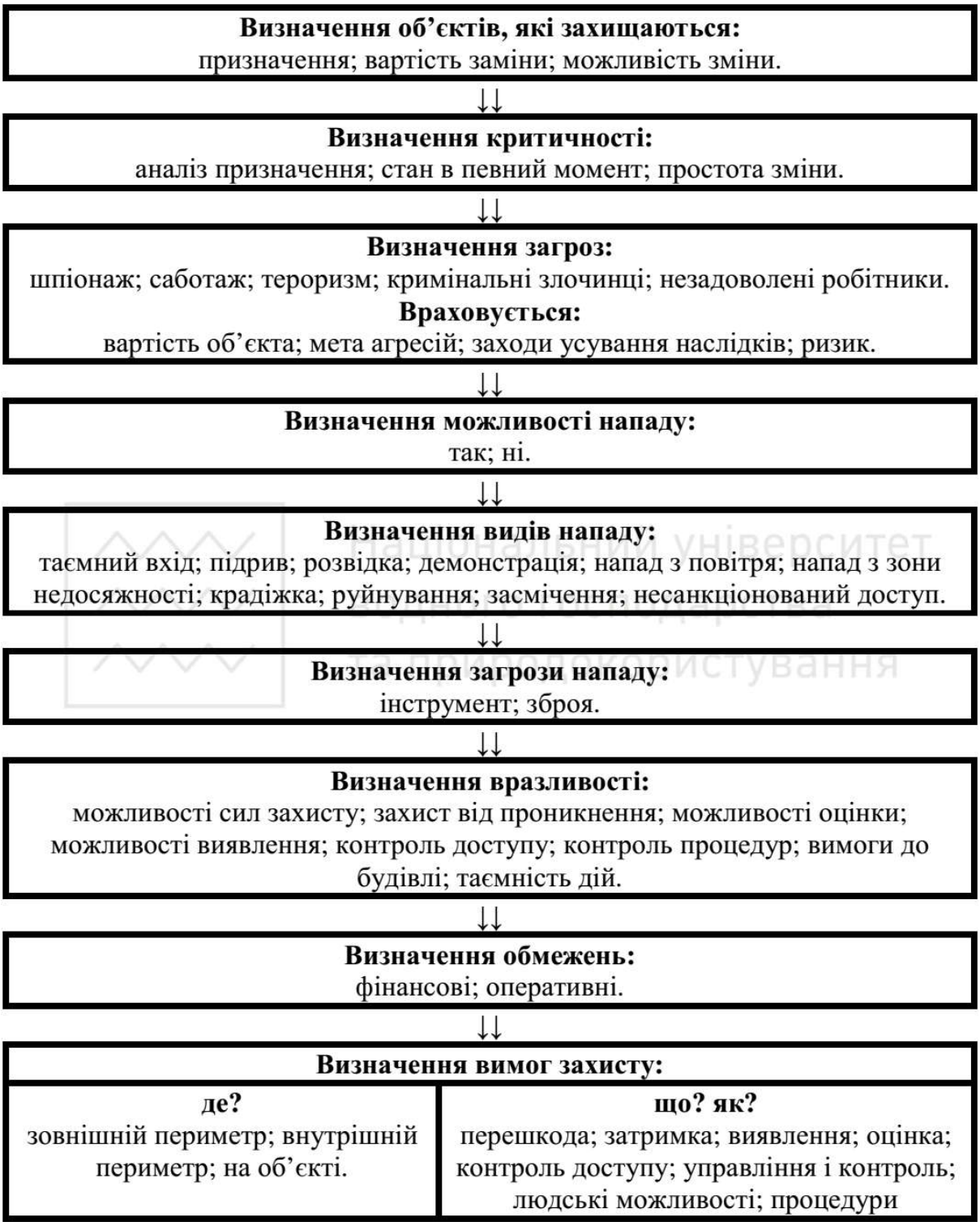


Рис. 5.1. Комплекс задач при розробці політики безпеки

Автоматизований комплекс можна вважати захищеним, якщо всі операції виконуються у відповідності з чітко визначеними правилами (рис. 5.2), що забезпечують безпосередній захист об'єктів, ресурсів і операцій. Основу для формування вимог до захисту складає список загроз.

Рис. 5.2. Основні правила забезпечення політики безпеки інформації

Захист інформації в комп'ютерній мережі ефективніший в тому випадку, коли проектування і реалізація системи захисту відбувається в три етапи:

• аналіз ризику;
• реалізація політики безпеки;
• підтримка 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 наступних 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 захищена від можливих ненавмисного і злочинного перекручувань);

• готовність (інформація і відповідні автоматизовані служби повинні бути доступні та, у разі потреби, готові до обслуговування).

Вразливість означає невиконання хоча б однієї з цих властивостей. Для **комп'ютерних мереж** можна виділити наступні **ймовірні загрози**, які необхідно враховувати при визначенні 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 по каналах 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 по каналах зв'язку;

10. відмова від авторства повідомлення, переданого по каналах 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 наступними 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 по каналах зв'язку;

7. забезпечення антивірусного захисту (у тому числі 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 при значних втратах;

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. При цьому, необхідно врахувати такі фактори як: безконфліктність роботи обраних засобів, репутація постачальників засобів захисту, можливість одержання повної інформації про механізми захисту і надані гарантії. Крім того, варто враховувати принципи, в яких відображені основні положення по безпеці інформації:

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

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

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

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

5. відкритість проектування і функціонування механізмів захисту (таємність проектування і функціонування засобів безпеки - кращий підхід до захисту інформації тому, що фахівці, які мають відношення до системи захисту, повинні цілком уявляти собі принципи її функціонування та, 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 в числі тих, кого ця система буде контролювати);

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

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

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

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

11. відмова за замовчуванням (якщо відбувся збій засобів захисту і розроблювачі 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 гнучкішим, якщо він допускає зміну своїх параметрів з боку адміністратора);

14. принцип ворожого оточення (система захисту повинна проектуватися 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 по можливості приховане від користувачів, робота яких контролюється).

Підтримка політики безпеки - третій, найбільш важливий, етап. Заходи, проведені на даному етапі, вимагають постійного спостереження за вторгненнями у мережу зловмисників, виявлення «дір» у системі захисту об'єкта інформації, обліку випадків несанкціонованого доступу до конфіденційних даних.

При цьому основна відповідальність за підтримку політики безпеки мережі лежить на системному адміністраторі, що повинен оперативно реагувати на усі випадки злому конкретної системи захисту, аналізувати їх і використовувати необхідні апаратні і програмні засоби захисту з урахуванням максимальної економії фінансових засобів.

Тема 6. Моделі політики безпеки

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 $U=\{U_1,...,U_m\}$ – множина користувачів, то можна визначити відображення {own: $O \rightarrow U$}, R – множина можливих видів доступів в даній системі. Відповідно до цього відображення кожний 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. Крім того, власник об'єкта визначає права доступу інших суб'єктів до даного 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, i=1,2,...,n до об'єктів O_j, j=1,2,...,m

$M =$

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

До переваг цього класу політик слід віднести:

1. відносно просту реалізацію та підтримку відповідних механізмів захисту. Саме цим обумовлений той факт, що більшість розповсюджених в теперешній час захищених ІТС забезпечують виконання положень ДПБ;

2. при її реалізації досягається велика економія пам'яті, оскільки матриця доступів звичайно буває дуже розрядженою, що дозволяє застосовувати техніку роботи з розрядженими матрицями.

Проте виявилося 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 вирусів в систему і інших засобів прихованої руйнівної дії.

Наступна проблема ДПБ – **автоматичне визначення прав**. Так як 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 при використанні ДПБ – це **контроль розповсюдження прав доступу**. Найчастіше буває, що власник файла передає вміст файла іншому користувачу і той, таким чином, фактично набуває права власника на цю інформацію.

Мандатна політика безпеки

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

1. всі 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,
5. до яких суб'єкт має доступ; 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 дозволяє 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. В системі з двома доступами *r* і *w* політика МПБ визначається наступними правилами доступу

$$X \xrightarrow{r} Y \Leftrightarrow c(X) \geq c(Y),$$

$$X \xrightarrow{w} Y \Leftrightarrow c(X) \leq c(Y).$$

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

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

Наведемо ряд 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. Таким чином, 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 і користувачами ІТС, що також є фактором, що позитивно впливає на рівень безпеки системи.

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

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

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

Рольова політика безпеки.

Рольову політику безпеки (Role Base Access Control – RBAC) не можна віднести ані до дискреційної, ані до мандатної, тому що керування доступом в 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 наступних властивостях:

1. всі 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 в рамках певної ролі.

В 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 діюча в системі абстрактна суттєвість, з якою пов'язаний обмежений, логично зв'язаний набір повноважень, які необхідні для здійснення певної діяльності.

В моделі РПБ визначаються наступні множини:

• U – множина користувачів;

• R – множина ролей;

• P – множина повноважень на доступ до об'єктів, що представляється, наприклад, у вигляді матриці прав доступу;

• S – множина сеансів роботи користувачів з системою.

Для перелічених множин визначаються наступні 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 наступними функціями:

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

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

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

В якості критерію безпеки рольової моделі використовується наступне 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*).**

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 компонентою є **монітор безпеки** – компонента КС, яка активізується при виникнення будь-якого потоку від одного об'єкта до іншого і дозволяє реалізуватися потокам, що належать тільки множині легального доступу *L*.

МБ повністю приймає 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'єкты КС повинні бути асоційованими з цім користувачем (якого зазвичай називають адміністратором безпеки).

Вибір методів і механізмів залишається за розробником і єдиною вимогою є реалізація функції захисту, причому для МБ повинні виконуватися наступні 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 завідомо повинен володіти незалежно від конкретної ПБ. Серед цих властивостей зазначимо наступні:

• при реалізації будь-якої ПБ найважливішим кроком є ідентифікація всіх об'єктів КС. При цьому повинна мати місце унікальність імен об'єктів, що дозволяє 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 $O_i \in O_t$ отримав в 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 $U_i \in U, i = 1,..., N_U$ і $P_j \in P, j = 1,..., N_P$;

- обов'язкова наявність механізму ІА: якщо для $\forall t \in N_0$, $\forall p \subseteq A$, $\forall O_i, O_j \in O_t$ має місце $O_i \xrightarrow{p?} O_j$, то МБ однозначно визначає належність O_i і O_j до відповідних множин U, P або O;

- обов'язкова присутність в МБ дозвільного механізму, тобто при запиті $O_i \xrightarrow{p?} O_j$, $O_i, O_j \in O_t$ в залежності 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 – міточний контроль)

Тема 7. Криптографічні методи захисту інформації

Основні положення та визначення

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

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

Основні напрямки 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 виді.

Сучасна криптографія вивчає і розвиває такі напрямки:

• 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:**

• аутентифікації;

• цілісності;

• незаперечності.

При **аутентифікації** одержувачеві повідомлення потрібно переконатися, що воно виходить від конкретного відправника. Зловмисник не може надіслати фальшиве повідомлення від будь-якого імені. При визначенні **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. В даний час **аутентифікація**, 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 аутентифікації 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.

Характеристика алгоритмів шифрування

Криптографічний захист у більшості випадків є більш ефективним і дешевим. Конфіденційність інформації при цьому 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 це - висока продуктивність, простота, захищеність і т.д. Програмна реалізація більш практична, допускає значну гнучкість у використанні. **Перед сучасними криптографічними системами захисту інформації ставлять наступні вимоги:**

1. зашифроване повідомлення повинне піддаватися читанню тільки при наявності ключа;
2. число операцій, необхідних для визначення використаного ключа шифрування по фрагменту шифрованого повідомлення і відповідного йому відкритого тексту, повинне бути не менше загального числа можливих ключів;
3. число операцій, необхідних для розшифровування інформації шляхом перебору ключів, повинно мати чітку нижню оцінку і виходити за межі можливостей сучасних комп'ютерів (з урахуванням можливості використання мережевих обчислень);
4. знання алгоритму шифрування не повинне впливати на надійність захисту;
5. незначна зміна ключа повинна приводити до істотної зміни виду зашифрованого повідомлення навіть при використанні того самого ключа;
6. структурні елементи алгоритму шифрування повинні бути незмінними;
7. додаткові біти, що вводяться в повідомлення в процесі шифрування, повинні бути цілком і надійно сховані в шифрованому тексті;
8. довжина шифрованого тексту повинна бути рівна довжині вихідного тексту;
9. не повинно бути простих (які легко встановлюються) залежностей між ключами, що послідовно використовуються в процесі шифрування;
10. будь-який ключ з безлічі можливих повинен забезпечувати надійний захист інформації;
11. алгоритм повинен допускати як програмну, так і апаратну реалізацію, при цьому 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).

При **симетричному шифруванні** (рис. 7.2) створюється ключ, файл разом з цим ключем пропускається через програму шифрування та 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 взаємозалежних ключі: відкритий і закритий. Одержувач повідомляє всім бажаючий свій **відкритий** ключ, що дозволяє 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.**

Коли надійність криптографічного алгоритму забезпечується за рахунок збереження в таємниці суті самого алгоритму, такий алгоритм шифрування називається **обмеженим**. Обмежені алгоритми становлять значний інтерес з погляду історії криптографії, однак зовсім непридатні при сучасних вимогах, які висуваються до шифрування. Адже, в цьому випадку,

кожна группа користувачів, які бажають обмінюватися секретними повідомленнями, повинна мати свої оригінальні алгоритми шифрування.

Рис. 7.3. Несиметричне 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 за рахунок їх належного вибору і наступного збереження в найсуворішому секреті. Це означає, що такий алгоритм не потрібно тримати в таємниці. Можна організувати масове виробництво криптографічних засобів, в основу функціонування яких покладений даний алгоритм. Навіть знаючи криптографічний алгоритм, зловмисник не зможе прочитати зашифровані повідомлення, оскільки він не знає секретний ключ, використаний 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 біт. Такі алгоритми називаються **блоковими** або блоковими шифрами. У сучасних комп'ютерних алгоритмах блокового шифрування довжина блоку звичайно складає 64 біта.

Симетричні алгоритми при виявленні 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 в даний час застосовуються наступні 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 в 1993 році. 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, також бібліотека функцій, що дозволяє виконувати операції шифрування і цифрового підпису повідомлень, файлів та іншої інформації, представленої в електронному вигляді, в тому числі 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 в 1990 році, також є стандартом СНД. У 2009 році ГОСТ 28147-89 перевиданий в Україні під назвою ДСТУ ГОСТ 28147:2009.;

• RSA (автори: Rivest, Shamir і Alderman) – (автори: Rivest, Shamir і Alderman) це система з відкритим ключем (public-key) призначена як для шифрування, так і для аутентифікації була розроблена в 1977 році. Вона заснована на труднощі розкладання дуже великих цілих чисел на прості множники. RSA дуже повільний алгоритм. Для порівняння, на програмному рівні DES по менше мірі в 100 разів швидше RSA, на апаратному аж в 1,000-10,000 разів, в залежності 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 при використанні коротких ключів.

До основних недоліків цих криптоалгоритмів варто віднести збільшення витрат по забезпеченню додаткових заходів таємності при поширенні 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, що не мають взаємозв'язку, що дозволяє по одному ключу 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 **дозволяють** реалізувати протоколи типу цифрового підпису, забезпечують відкрите поширення ключів і надійну аутентифікацію в мережі, стійку навіть до повного перехоплення трафіка.

Тема 8. Методи захисту інформації в операційних системах

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

• 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.

При шифруванні 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 ряд стандартів, які забезпечують надійний захист інформації. Найбільш поширеними є дві схеми шифрування – DES (Data Encryption Standard) і RSA (отримав назву по перших буквах прізвищ авторів – Rivest, Shamir, Adleman). DES-схема симетрична, в ній для шифрування і дешифрування використовується один і той же ключ. Схема RSA асиметрична, ключі шифрування і дешифрування в ній різні.

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 році, в 1980 році був прийнятий NIST (National Institute of Standards and Technolody США) у якості стандарту. DES є класичною сіткою Фейстеля з двома множинами (рис. 8.1).

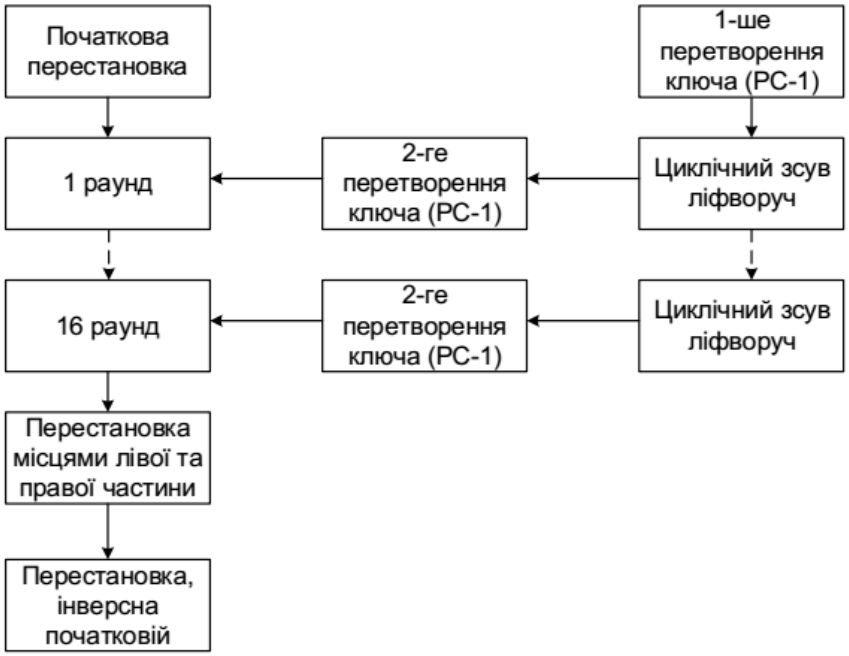


Рис. 8.1. Загальна схема DES

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 перестановка IP^{-1} результату, отриманого на третьому етапі. Перестановка IP^{-1} обернена до початкової перестановки IP.

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

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

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	57	25

Послідовність перетворень окремого раунду

Розглянемо послідовність перетворень, яка використовується в кожному раунді. 64-бітний вхідний блок проходить через 16 раундів обробки, при цьому 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:

$$L_i=R_{i-1}. \qquad (4.1.)$$

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

Блок Ri, який подається на вхід функції F, має довжину 32 біти. Спочатку Ri розширюється до 48 бітів, використовуючи таблицю, 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 таким чином. Якщо частина вхідного повідомлення:

. . . efgh ijkl mnop . . .,

то в результаті 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 4x16, в яких певним чином 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 в десятковому представленні число 11. Отже, для заміни

буде використано стовпчик № 11. На перетині рядка № 2 та стовпчика № 11 знаходиться комірка з числом 7. Воно, точніше його двійкове представлення 0111, і буде результатом застосування S-боксу. Таким чином, замість 6-бітного числа 110110 отримаємо 0111. Аналогічним чином 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 кожний 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	32	27	3	9	19	13	30	6	22	11	4	25

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 таким чином 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	46	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. Таблицю 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 таким чином. Підблоки розглядаються як послідовність рядків табл. 4.6, записаних один за одним, починаючи з першого. Біти отриманого таким чином блоку даних перенумеровуються зліва направо, починаючи з одиниці. Кожен елемент *S* таблиці розглядається як номер біта *b*S в отриманому блоці даних. Перетворенням є заміна усіх $S \rightarrow b$S.

Операція розшифрування

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.

П**ереваги та недоліки DES**.

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 на сьогодні вважають:

1. невелику довжину ключа, усього 56 бітів. При сучасному рівні розвитку комп’ютерних засобів така довжина ключа не може забезпечувати потрібний рівень захисту для деяких типів інформації;

2. наявність "слабких" ключів, викликана тим, що для генерування ключової послідовності виконується два незалежних регістри зсуву.

3. надмірність ключа, що має біти контролю парності.

Інші схеми шифрування

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 по відкритому ключу {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(i)=(M(i)**e)modn.

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

M(i)=(C(i)**d)modn.

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

Розроблено також і вітчизняний стандарт шифрування даних - ГОСТ 28147-89. Однак його програмна реалізація дуже складна і практично немає ніякого сенсу через низьку швидкодію.

Тема 9. Аналіз безпеки ПЗ та руйнуюче ПЗ

«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 відносин від політики безпеки полягає в тому, що політика безпеки служить спрощеною моделлю реального розподілу ролей користувачів і функцій програм системі, легітимність же відносин ґрунтується на перевірці порушення основних характеристик обчислювальної системи – цілісності і безпеки.

З урахуванням уведених понять визначимо відносини, що 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, принаймні, 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 в розвинутих мережних операційних системах (так називані «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.

Найчастіше виділяють три основних підкласи, зв'язаних із класом РПЗ відношенням включення, – **віруси**, **троянські коні** або «закладки» і **програми зломщики** систем захисту і засобів розмежування доступу. Усі ці підкласи РПЗ характеризуються специфічними відносинами з об'єктами обчислювальної системи. Помітимо, що різноманіття видів і типів РПЗ не вичерпується цими трьома класами, просто вони є найбільш розповсюдженими.

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

Троянські коні. Цей клас РПЗ, характеризується ще одним специфічним типом відносин 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, як умова спрацьовування служить настання деякого моменту часу або системної події.

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

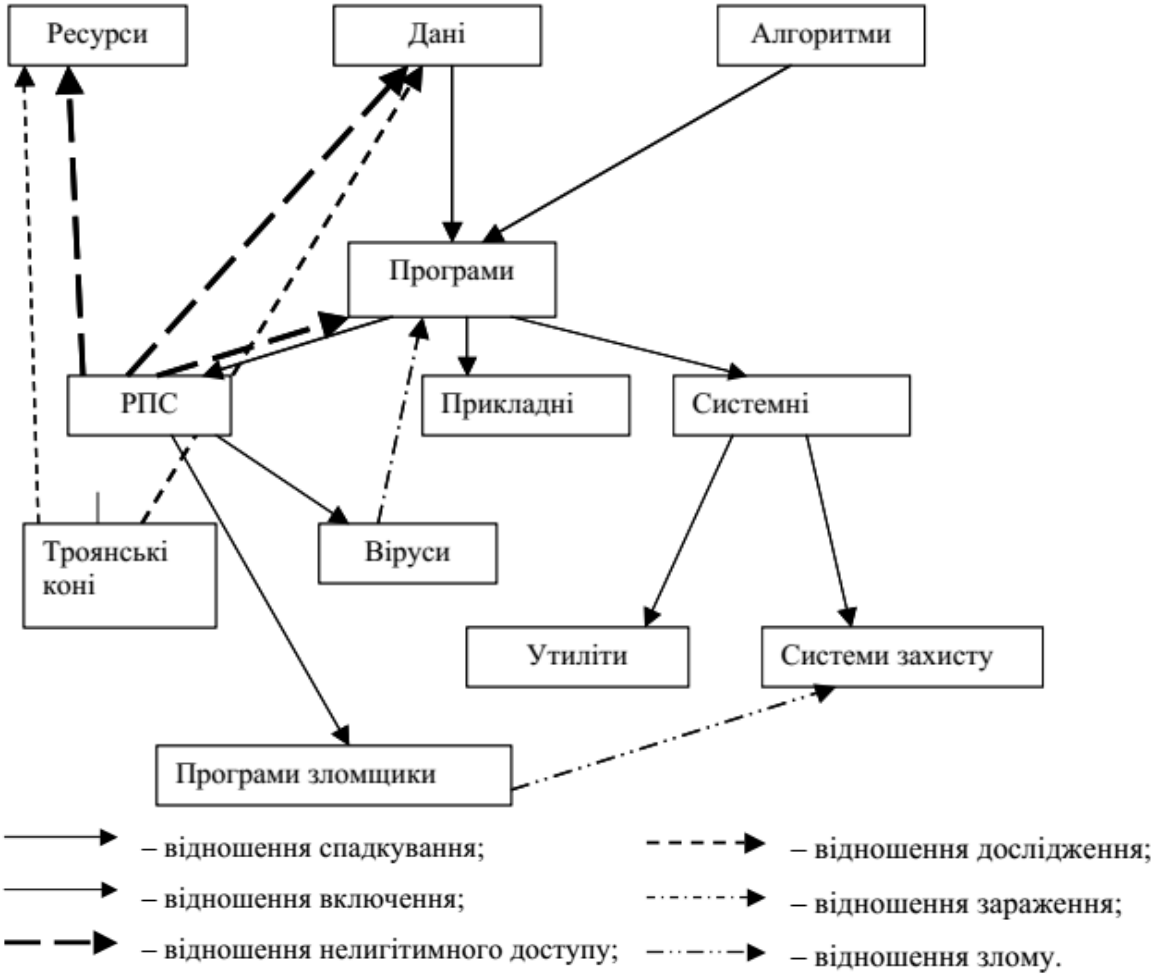


Рис. 9.1. Класи об’єктно-концептуальної моделі предметної області і відносини між ними

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 існуючі в ній до цього кроку відносини між суб’єктами й об’єктами були легітимні, тобто $Ri \in L$.

Тема 10. Методи аналізу безпеки ПЗ

Під безпекою ПЗ будемо розуміти відсутність у ньому елементів РПЗ. Для доказу безпеки програми потрібно довести, що програма не встановлює нелегітимних відносин з об'єктами обчислювальної системи. З урахуванням уведених понять приведемо 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 відносин Ap, до якої належать усі відносини з об'єктами обчислювальної системи, що установлюються програмою p у процесі її виконання, не містить нелегітимних відносин, тобто $Ap \cap Lp = \emptyset$.

Проте, розв'язок цієї задачі утруднюється двома проблемами.

По-перше, у загальному випадку неможливо побудувати розв'язуючу процедуру, що дозволяє визначити легітимність відносини доступу.

По-друге, неможливо одержати всі елементи для визначення їхньої легітимності.

Широко відомі різні засоби ПЗ виявлень елементів РПЗ – від найпростіших антивірусних програм-сканерів до складних відладчиків і дизасемблерів-аналізаторів, у той час як теоретичні дослідження в області методів аналізу безпеки носять 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.

Методи, що використовуються для аналізу безпеки ПЗ, на дві категорії: **контрольно-іспитові** і **логіко-аналітичні**. В основу даного поділу покладені принципові розходження точок зору на об'єкт (програму), що досліджується – контрольно-іспитові методи аналізу розглядають РПЗ як феномен, а логіко-аналітичні – як ноумен.

Якщо розглядати РПЗ як **феномен**, то задача вирішується в просторі відносин. 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 служить більшість сучасних засобів пошуку вірусів, що використовують метод пошуку сигнатур або перевірку деякого набору ознак.*

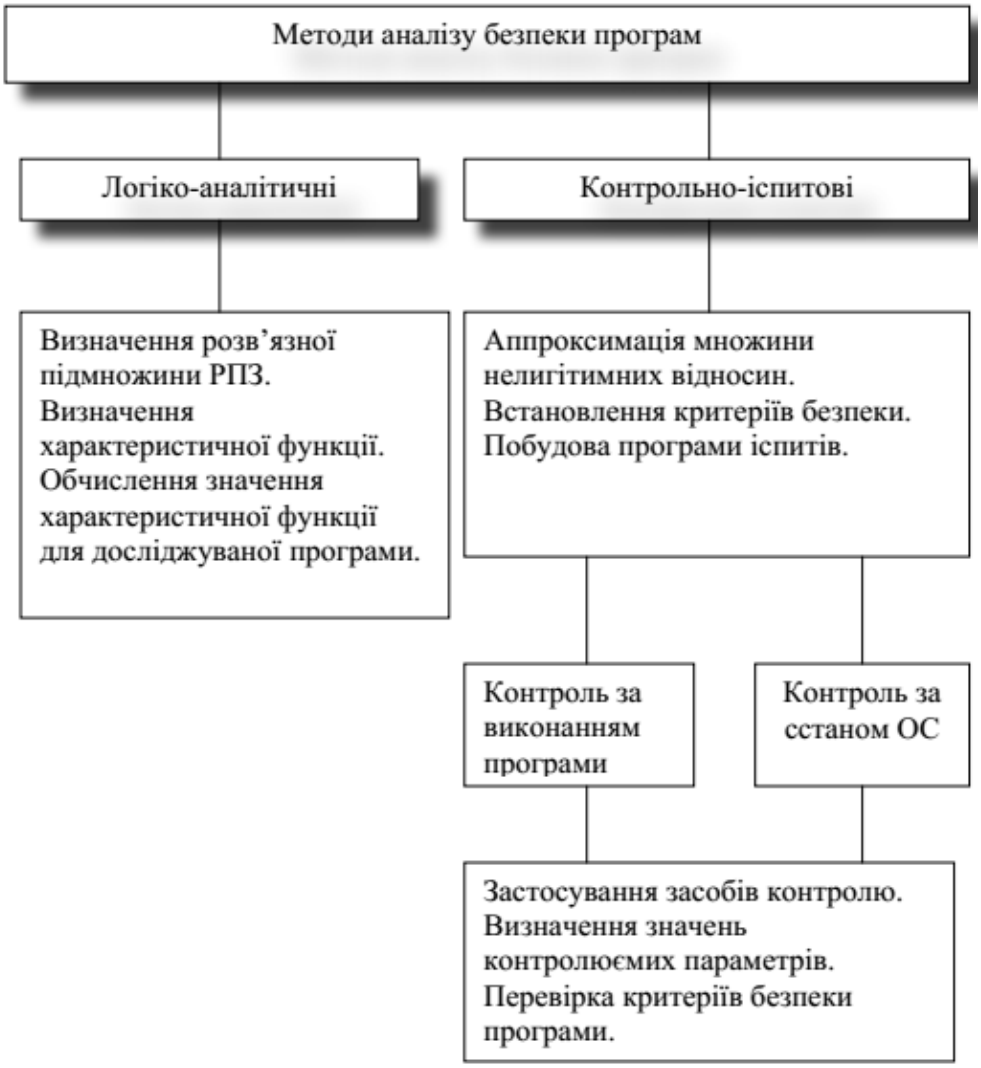


Рис. 10.1. Систематизація методів аналізу безпеки ПЗ

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

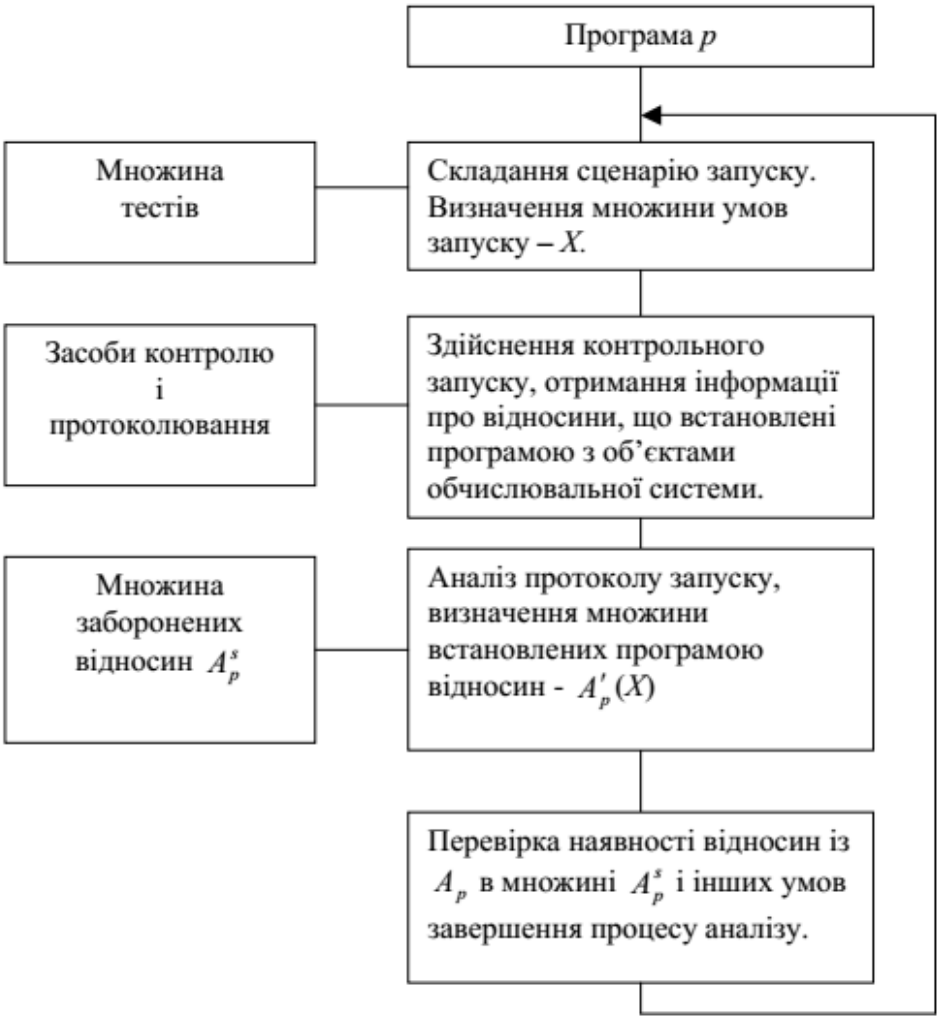
При цьому критерієм безпеки програми служить факт реєстрації в ході тестування порушення вимог по безпеці, що пред'являються у системі передбачуваного застосування досліджуваної програми.

Розглянемо **формальну постановку задачі аналізу безпеки ПЗ для рішення 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 по безпеці, яким повинна задовольняти програма, можуть бути задані у виді множини заборонених

відносин p з об'єктами C_Σ – Aps. Елементи цієї множини повинні бути задані або в явному виді за допомогою перерахування, або у виді набору правил, що дозволяє визначити приналежність відносини 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:



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 $Bp=\{b_{pi}|i=1,...,N\}$. У рамках цієї системи моделювання повинна бути задана розв'язна підмножина РПЗ $V^* \subset V$, що володіє визначеної на множині

атрибутів *B* характеристичною функцією ϕ(*b1*,*b2*,...,*bN*). Підмножина РПЗ *V** може бути отримана або шляхом побудови моделей усіх відомих РПЗ, або шляхом породження моделей усіх РПЗ, можливих у даній системі моделювання.

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

Структурна схема логіко-аналітичних методів дослідження безпеки програм:

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

• представити процес взаємодії компонент обчислювальної системи з точки зору безпеки;

• формалізувати властивості РПЗ і створити основу для їхньої систематизації;

• дати формальне визначення поняття безпеки програм;

• формалізувати задачу аналізу безпеки ПЗ.

Тема 11. Поняття про гешувальні алгоритми, їх призначення, вимоги до них

Особливе місце серед механізмів забезпечення цілісності і автентичності займають функції гешування: безключові та ключові, що дозволяють забезпечити широкий спектр послуг безпеки інформації згідно з ISO 7498. Односторонні геш-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 проектування та розробки підсистем захисту інформації. Це пов'язано з постійним зростанням послуг, які надаються різними мережними службами. Більшість послуг надаються при відсутності фіксованих мережених адрес клієнтів та їх особливостей. В зв'язку з чим, ризик порушення цілісності та автентичності інформації збільшується. Для захисту від таких загроз безпеки інформації, як правило, використовують механізми гешування даних – ключові та безключові геш-функції. Геш-функції також можуть використовуватись у складі електронного цифрового підпису, який є потужним механізмом забезпечення автентифікації в сучасних автоматизованих системах.

Геш-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 $y_1 = h(x_1)$, де $D = \{0,1\}^a \times \{0,1\}^b \times i$ і $R=\{0,1\}^n$ для деяких a,b і $n \geq 1$, з $a+b \geq n$.

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

Далі наведені визначення для стійкості за (другим) прообразом і стійкістю до колізій.

Стійкість за прообразом. Геш-функція $h : \{0,1\} \longrightarrow R$ є стійкою за прообразом ступеня $(t,\in)$, якщо не існує імовірнісного алгоритму I_h, який приймає вхід $Y \in_R R$ і 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 $\{0,1\}$. Геш-функція $h : \{0,1\}^* \rightarrow R$ є стійкою за другим прообразом ступеня $(t,\in,S)$, 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 $X' \neq X\, h(X') = h(X)$ ймовірністю щонайменше ε, отриманою випадковими виборами S_h і X.

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

Геш-функція – це функція, яка відображає вхід довільної довжини в фіксоване число вихідних біт – геш-значення. Для того щоб бути корисною в криптографічних додатках, геш-функція повинна задовольняти деякі вимоги. Геш-функції можуть поділятися на **односторонні** гешфункції та **стійкі до колізій геш-функції**.

Одностороння функція повинна бути стійкою за прообразом і другим прообразом, тобто повинно "важко" знайти повідомлення із заданим гешем (прообразом) або яке гешується в одне і те ж значення, що і задане повідомлення (другий прообраз).

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

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

1. аргумент X може бути довільної довжини, а результат h(X) має фіксовану довжину n біт;
2. геш-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′≠ X таке, що h(X′) = h(X) (стійкі за другим прообразом).

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

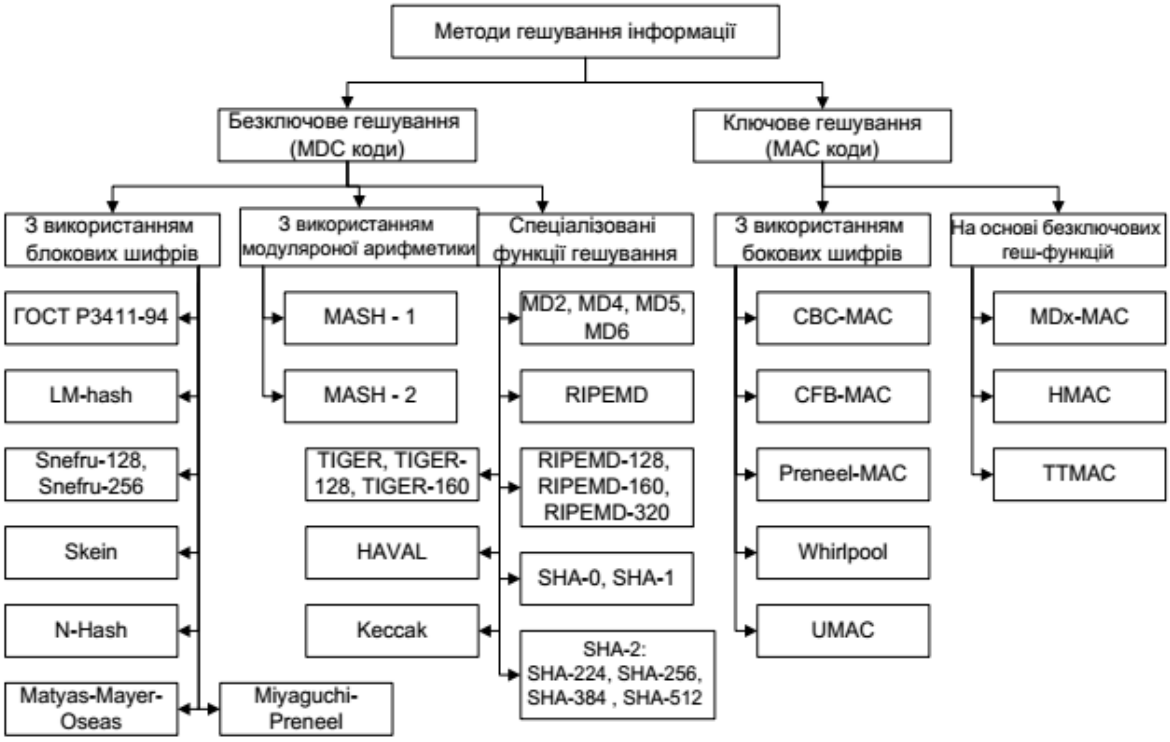
1. аргумент X може бути довільної довжини, а результат h(X)має фіксовану довжину n біт;
2. геш-функція повинна бути односторонньою, тобто стійкою за прообразом і стійкою за другим прообразом.

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

1. незворотність або стійкість до відновлення прообразу: для заданого значення геш-функції m має бути обчислювально 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 обчислювально неможливо підібрати інше повідомлення n, для якого h(n)=h(m);
3. стійкість до колізій другого роду: має бути обчислювально неможливо підібрати пару повідомлень, що мають однаковий геш.

Більшість геш-функцій мають ітеративні конструкції в тому сенсі, що вони базуються на функції компресії з фіксованими входами, вони обробляють кожен блок повідомлення подібним чином. 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 .

Класифікація геш-функцій.



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

До ключових геш-функцій відносяться *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 біт, де вторинний вхід K позначає секретний ключ.

2. При наявності даних h і X (але з невідомим K), повинне бути складно визначити h(K;X) з імовірністю успіху значно більшою $1/2^n$. Навіть при великій кількості відомих пар $\{X_i;h(K;X_i)\}$ 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 кожний блок повідомлень аналогічниим способом. Вхід X є однозначним заповненням, кратним розміру блоку. Звичайно це також включає збільшення загальної довжини на 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 й g – вихідне перетворення. Секретний ключ K може бути застосований в Z, у функції стиску, і/або у вихідному перетворенні.

Тема 12. Поняття про цифровий підпис, вимоги до нього

Електронний цифровий підпис (ЕЦП) – реквізит електронного документа, призначений для захисту даного електронного документа від підробки, отриманий у результаті криптографічного перетворення інформації з використанням закритого ключа електронного цифрового підпису, що дозволяє 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:

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

Повідомлення M та його електронний підпис S відправляють за призначенням.

Отримувач, маючи *(M, S)* та публічний ключ відправника повідомлення *e,* може перевірити виконання співвідношення:

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

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

Така схема призводить до наступного:

- 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 стосунки 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 цифровогопідпису:

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

2. Підпис повинен використовувати деяку унікальну інформацію про відправника для запобігання підробки або відмови.

3. Створювати цифровий підпис повинно бути відносно легко.

4. Повинно бути розрахунково неможливо підробити цифровий підпис як створенням нового повідомлення для існуючого *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 діляться на два класи:

схема ЕЦП із відновленням повідомлення;

схема ЕЦП із додаванням.

За **кількістю** учасників ЕЦП підрозділяється на:

одиночну схему ЕЦП;

групову схему ЕЦП.

За **способом перевірки** ЕЦП поділяються на:

інтерактивні схеми ЕЦП, що вимагають протокольної взаємодії;

не інтерактивні схеми ЕЦП, які не потребують протокольної взаємодії.

Існуючі алгоритми ЕЦП можна розділити 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 при 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.

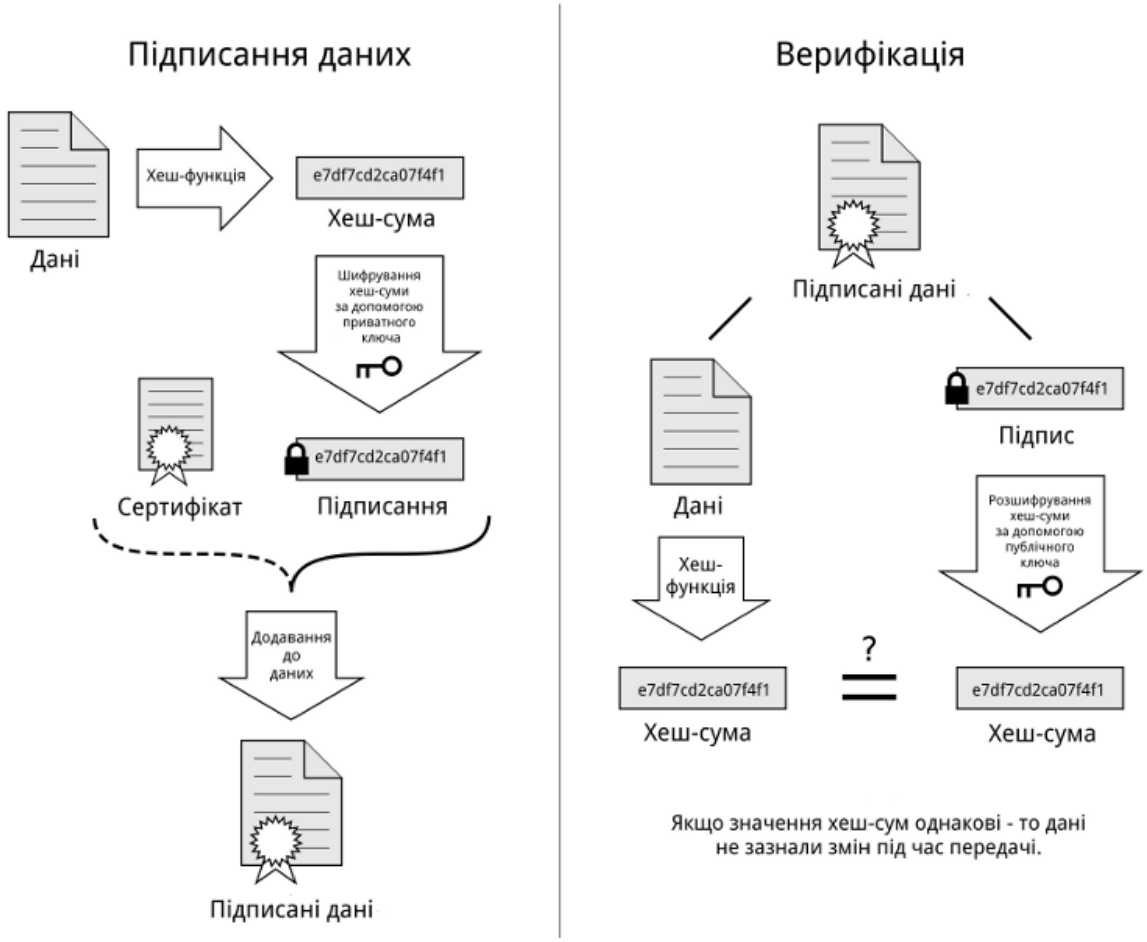
5. *Процес* (*процедура*) *зняття ЕЦП* – сукупність алгоритму верифікації ЕЦП і методів відновлення даних.

Для побудови схеми ЕЦП необхідно визначити два алгоритми: алгоритм **генерації** ЕЦП і алгоритм **верифікації** ЕЦП. Алгоритм верифікації доступний для всіх потенційних одержувачів підписаних повідомлень, у той час, як 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 (автентична) і "ні" – у протилежному випадку.

Існуючі на сьогоднішній день схеми ЕЦП діляться на два класи: схеми ЕЦП із відновленням повідомлення; схеми ЕЦП із додаванням.

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

У *схемах ЕЦП із* ***додаванням*** 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 при генерації підпису використовується деякий випадковий параметр (число), що призводить до формування різних підписів для однакових вхідних рядків (при використанні тих самих ключів). У рандомізованих схемах необхідно забезпечити непередбачуваність випадкових чисел.



ЕЦП може бути сформована за допомогою двох схем:

симетрична схема – дана схема передбачає наявність в системі третьої особи, яка користується довірою обох сторін. Авторизацією документа в даній схемі є сам факт зашифрування електронного документа секретним ключем і передача його третій особі;

асиметрична схема – дана схема відноситься до криптосистем з відкритим ключем.

На сьогоднішній день 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 даному випадку потрібно і передавати ключ шифрування, саме це може створити проблему, адже якщо канал передачі не захищений, то ключ може бути перехоплений зловмисником. В асиметричних системах цей недолік відсутній, оскільки кожний учасник має пару ключів: відкритий та секретний, які зв'язані між собою. При цьому формування ЕЦП відбувається за допомогою секретного ключа відправника, а перевірка підпису – за допомогою відкритого ключа, тому необхідність передачі секретного ключа відсутня. У зв'язку з цим,

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 ключа;

формування підпису;

перевірка підпису.

На сьогоднішній день 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 p і q приблизно 512 біт завдовжки кожне
2. Обчислюється їх добуток $n = pq$
3. Обчислюється функція Ейлера $\varphi(n) = (p-1)(q-1)$
4. Вибирається ціле число e таке, що $1 < e < \varphi(n)$ та e взаємно просте з $\varphi(n)$
5. За допомогою розширеного алгоритму Евкліда знаходиться число d таке, що $ed \equiv 1 \pmod{\varphi(n)}$

Число n називається модулем, а числа e і d — відкритою й секретною експонентами (англ. *encryption and decryption exponents*), відповідно. Пари чисел (n, e) є відкритою частиною ключа, а (n, d) — секретною. Числа p і 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 в ціле число m так, щоб $0 \leq m < n$ за допомогою узгодженого оборотного протоколу, відомого як схеми доповнення. Потім він обчислює зашифрований текст c, використовуючи відкритий ключ Аліси e, за допомогою рівняння:

$$c = m^e \bmod n\,.$$

Це може бути зроблено досить 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:

$$m = c^d \bmod n\,.$$

Неважко переконатися, що при розшифруванні відновиться вихідне повідомлення

$$c^d \equiv (m^e)^d \equiv m^{ed} \pmod{n}$$

З умови

$$ed \equiv 1 \pmod{\varphi(n)}$$

випливає, що

$ed = k\varphi(n) + 1$ для деякого цілого k, отже
$m^{ed} \equiv m^{k\varphi(n)+1} \pmod{n}$

Згідно з теоремою Ейлера:

$$m^{\varphi(n)} \equiv 1 \pmod{n},$$

тому

$$m^{k\varphi(n)+1} \equiv m \pmod{n}$$
$$c^d \equiv m \pmod{n}$$

Етап	Опис операції	Результат операції
Генерація ключів	Обрати два простих різних 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:

$$m = s^e \mod n$$

Тема 13. Основні положення керування ключами. Життєвий цикл криптографічного ключа

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

- ініціалізацію системних користувачів усередині домену безпеки;
- генерацію, розподіл й 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 таких основних погроз:

- 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.

При переході з одного основного стану в інший ключ може перебувати в одному з перехідних станів (transition). Такими перехідними станами є:

генерація – процес генерації ключа, у ході якого відповідно до запропонованих правил генерується ключ;

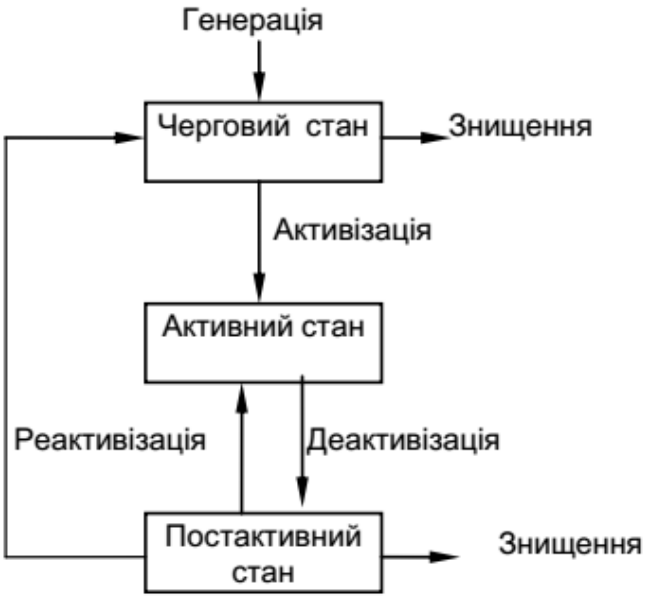
активізація (activation) – процес або сукупність процесів, у ході яких ключ робиться придатним для використання, тобто переводиться зі стану очікування в активний стан;

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 дозволяють перевести ключ з постактивного в активний стан для повторного використання;

знищення (destruction) – завершує життєвий цикл ключа.

Схематично поданий взаємозв'язок основних і перехідних станів.



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 (звичайно 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.

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 безпечним чином 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 даний 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.

Життєвий цикл керування ключами містить такі основні етапи й процеси.

1. Реєстрація користувача – процес, у ході 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). Всі дії в ході реєстрації здійснюються безпечними одноразовими способами, наприклад, через особистий обмін, замовлення поштою, довіреним кур'єром.

2. *Ініціалізація користувача* – процес, у ході якого об’єкт ініціалізує свій криптографічний додаток (наприклад, інсталює та ініціалізує програмне або апаратне забезпечення), включаючи використання або інсталяцію 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 служити для організації безпечного on-line 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 on-line-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 даному етапі ключі перебувають в оперативній доступності для стандартних криптографічних додатків, включаючи контроль 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 у той час, як 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 при виникненні особливих умов, наприклад, вирішення спорів, включаючи реалізацію функції причетності. Архівування допускає довгострокове автономне зберігання ключа, який виводиться при цьому в постактивний стан.

10. Перед *знищенням ключа* здійснюється його *дереєстрація*, тобто видалення відповідного запису з довідника, у результаті 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 у результаті не примусової (ненавмисної) компрометації (збої, несправність устаткування, забування (втрата) пароля). У цьому випадку використовуються резервні копії ключа.

12. Видалення (анулювання) ключа має на увазі виведення ключа з активного стану в постактивний в результаті, наприклад, його компрометації.

ри цьому 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.

Тема 14. Технологія блокчейну.

Блокчейн (англ. Blockchain або block chain) - вибудувана за певними правилами безперервна послідовна 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 в різних криптовалюта, проте технологія ланцюжків блоків може бути поширена на будь-які взаємопов'язані інформаційні блоки.

Блок транзакцій - спеціальна структура для запису групи транзакцій в системі біткойнів і аналогічних їй. Транзакція вважається завершеною і достовірною («підтвердженої»), коли перевірені її формат і підписи, і коли сама транзакція об'єднана в групу з декількома іншими і записана в спеціальну структуру - блок. Вміст блоків може бути перевірено, так як кожен блок містить інформацію про попередній блок. Всі блоки збудовані в один ланцюжок, яку містить інформацію про всі вчинені коли-небудь операціях в базі. Найперший блок в ланцюжку - первинний блок (англ. Genesis block) - розглядається як окремий випадок, так як у нього відсутня 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 біткойнов першої 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 в попередні блоки. Критерій відбору з черги задає майнер самостійно. Це не обов'язково повинна бути хронологія за часом. Для транзакцій в блоці використовується **деревоподібне хешування**, аналогічне формування хеш-суми для файлу 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 одно або нижче певного числа, величина якого періодично коригується. Так як результат хешування функції SHA-256 вважається незворотнім, на даний момент немає алгоритму отримання бажаного результату, крім випадкового перебору. Якщо хеш не задовольняє умові, 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 в ланцюжок і наступний блок повинен включити в себе його хеш.

Блоки одночасно формуються множиною «**Майнер**». Задовольняють критеріям блоки 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 головною ланцюжок з урахуванням рівня складності хешу і довжини ланцюжка.

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

Розподілена база даних Blockchain формується як безперервно зростаюча ланцюжок блоків із записами про всі транзакції. Копії бази або її частини одночасно зберігаються на безлічі комп'ютерів і синхронізуються відповідно до формальних правил побудови ланцюжка блоків. Інформація в блок не шифрувати і доступна у відкритому вигляді, але відсутність змін засвідчується криптографически через хеш-ланцюжка (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, в числі іншого, містить в собі хеш попереднього блоку і свій порядковий номер. Кожен новий блок здійснює підтвердження транзакцій, інформацію про яких містить і додаткове підтвердження транзакцій у всіх попередніх блоках ланцюжка. Змінювати інформацію в блоці, який вже знаходиться в ланцюзі, не практично, так як 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, умисне зміна інформації в будь-який з копій бази або навіть в досить великій кількості копій нічого очікувати визнано істинним, тому що не буде відповідати правилам. Деякі зміни можуть бути прийняті, якщо будуть внесені в усі копії бази (наприклад, видалення декількох останніх блоків через помилки в їх формуванні).

Підтвердження транзакцій

Поки транзакція не включена в блок, система вважає, що кількість біткойнов на якомусь адресу залишається незмінним. У цей час є технічна можливість оформити кілька різних транзакцій з передачі з однієї адреси одних і тих же біткойнів різним одержувачам. Але як тільки одна з подібних

транзакцій буде включена в блок, інші транзакції з цими ж біткойнов система буде вже ігнорувати. Наприклад, якщо в блок буде включена пізніша транзакція, то більш рання буде вважатися помилковою. Є невелика вірогідність, що при розгалуженні дві подібні транзакції потраплять в блоки різних гілок. Кожна з них буде вважатися правильною, лише при відмирання гілки одна з транзакцій стане вважатися помилковою. При цьому не буде мати значення час здійснення операції.

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

Перелік питань на підсумковий контроль

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 напрямки розвитку теорії захисту інформації.
7. Вкажіть, що собою представляє загроза безпеки КС.
8. Вкажіть, які загрози безпеки КС відносять до випадкових.
9. Вкажіть, які загрози безпеки КС відносять до навмисних.
10. Вкажіть, що собою представляє загроза розкриття і їх протидія.
11. Вкажіть, що собою представляє загроза порушення цілісності і їх протидія.
12. Вкажіть, що собою представляє 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 представляє спосіб несанкціонованого доступу та які мети переслідує 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 при розробці політики безпеки.
25. Вкажіть правила забезпечення політики безпеки інформації.
26. Опишіть перший етап проектування та реалізації системи захисту.
27. Вкажіть, які ймовірні загрози виділяють у комп'ютерних мережах.
28. Вкажіть, яким заходам повинна визначатися політика безпеки.
29. Опишіть другий етап проектування та реалізації системи захисту – реалізація політики безпеки.
30. Опишіть третій етап проектування та реалізації системи захисту – підтримка 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 представляє криптографія.
38. Вкажіть для забезпечення чого можна використовувати криптографію.
39. Вкажіть, що застосовують для виявлення несанкціонованих змін у переданих повідомленнях.
40. Вкажіть, що собою представляє 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 на 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 відносин руйнуючого програмного забезпечення.
56. Опишіть основні підкласи, що відносяться до класу руйнуючого 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 геш-функції.
66. Дайте означення стійкості за першим та другим прообразом.
67. Дайте означення односторонньої геш-функції.
68. Вкажіть, коли геш-функція стійка до колізій.
69. Вкажіть, що собою представляють безключові геш-функції.
70. Вкажіть, що собою представляють ключові геш-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.

Література та джерела

1. Сенів М. М. Безпека програм та даних: навч. посібник / М.М. Сенів, В.С. Яковина. – Львів : Видавництво Львівської політехніки, 2015. – 256 с.
2. Горбенко І. Д. Гриненко Т. О. Захист інформації в інформаційно-телекомунікаційних системах: Навч. посібник. Ч.1. Криптографічний захист інформації - Харків: ХНУРЕ, 2004 - 368 с.
3. Лагун А. Е. Криптографічні системи та протоколи: нав. посібник / А. Е. Лагун. – Львів : Видавництво Львівської політехніки, 2013. – 96 с.
4. Белов Е. Б., Лось В. П., Мещеряков Р. В., Шелупанов А. А. Основы информационной безопасности. Учебное пособие для вузов - М.: Горячая линия - Телеком, 2006. - 544 с: ил.
5. Биячуев Т. А. / под ред. Л. Г.Осовецкого Безопасность корпоративных сетей. – СПб: СПб ГУ ИТМО, 2004.- 161 с.
6. Завгородний В. И. Комплексная защита информации в компьютерных системах: Учебное пособие. - М.: Логос, 2001. - 264 с : ил.
7. Зегжда Д. П., Ивашко А. М. Основы безопасности информационных систем. – М.: Горячая линия – Телеком, 2000. 452с., ил.
8. Малюк А. А. Информационная безопасность: концептуальные и методологические основы защиты информации. Учеб. пособие для вузов. - М: Горячая линия-Телеком, 2004. -280 с. ил.
9. Малюк А. А., Пазизин С. В., Погожин Н.С. Введение в защиту информации в автоматизированных Системах. - М.: Горячая линия-Телеком, 2001. - 148 с: ил.
10. Мамаев М., Петренко С. Технологии защиты информации в Интернете. Специальный справочник. – СПб.: Питер, 2002.- 848 с.: ил.
11. Романец Ю. В., Тимофеев П. А., Шаньгин В. Ф. Защита информации в компьютерных системах и сетях / Под ред. В. Ф. Шаньгина.-2-е изд., перераб. и доп.-М.: Радио и связь, 2001.-376 с: ил.