МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Київський національний університет будівництва і архітектури

В.М. Вишняков

ЗАХИСТ ІНФОРМАЦІЇ В КОМП'ЮТЕРНИХ СИСТЕМАХ

Рекомендовано вченою радою Київського національного університету будівництва і архітектури як навчальний посібник для студентів галузі знань 12 «Інформаційні технології»

Київ 2022

УДК 004.056.5
В55

Рецензенти: *Д.В. Ланде*, д-р техн. наук, професор,
Київський політехнічний інститут імені Ігоря Сікорського;
Р.С. Одарченко, д-р техн. наук, професор,
Національний авіаційний університет

Затверджено на засіданні вченої ради Київського національного університету будівництва і архітектури, протокол № 4 від 24 січня 2022 року.

Вишняков В.М.

В55 Захист інформації в комп'ютерних системах**:** навч. посіб. / В.М. Вишняков. – Київ: КНУБА, 2022. – 120 с.

ISBN 978-966-627-194-8

Розглянуто основні поняття щодо побудови систем захисту даних на базі державних нормативних документів. Наведено відомості про функції та механізми криптографічного захисту від несанкціонованого доступу та інших загроз інформації у комп'ютерних системах.

Призначений для студентів факультету автоматизації та інформаційних технологій.

УДК 004.056.5

ISBN 978-966-627-194-8

ЗМІСТ

Вступ 5
Розділ 1. Концепції та термінологія систем захисту даних 6
1.1. Основні поняття та скорочення 6
1.2. Дії із даними, їх властивості та загрози у комп'ютерних системах 9
1.3. Основні поняття щодо розробки та експлуатації систем захисту 11
1.4. Принципи побудови механізмів захисту даних 13
Висновки 18
Запитання та завдання для самоперевірки 20
Розділ 2. Характеристики функціональних послуг захисту та властивості механізмів для їх реалізації 22
2.1. Структура автоматизованої системи з комплексом засобів захисту даних 22
2.2. Критерії оцінки захищеності даних у інформаційних системах 25
2.2.1. Критерії конфіденційності 27
2.2.2. Критерії цілісності 30
2.2.3. Критерії доступності 33
2.2.4. Критерії спостереженості 35
2.2.5. Критерії гарантій 41
2.3. Дискреційні та мандатні механізми керування доступом 44
Висновки 46
Запитання та завдання для самоперевірки 49
Розділ 3. Криптографічні методи захисту даних 51
3.1. Основні поняття криптографії 51
3.2. Стандартизовані симетричні алгоритми шифрування 56
3.3. Алгоритм шифрування з відкритим ключем RSA 73
3.4. Поняття геш-функції та цифровий підпис 76
3.5. Алгоритм Диффі-Хелмана 82
Висновки 84
Запитання та завдання для самоперевірки 87
Розділ 4. Захист даних в IP-мережах 88

4.1. Особливості мережевих технологій захисту даних 88
4.2. Принципи побудови комплексу засобів IPSec 89
4.3. Протоколи, що забезпечують безпеку передавання даних 93
4.4. Особливості політики IP-безпеки із засобами IPSec 98
4.5. Поглиблений аналіз потоків даних у IP-мережах 100
Висновки 101
Запитання та завдання для самоперевірки 105
Список літератури 106
Додаток 1 108
Додаток 2 110
Додаток 3 118

ВСТУП

Цей посібник допоможе ознайомитись з методами і засобами захисту даних від несанкціонованого доступу та інших загроз інформації в комп'ютерних системах.

Метою захисту даних є забезпечення їх цілісності, конфіденційності та доступності для санкціонованих користувачів за умов наявності випадкових факторів або зловмисних дій, що можуть призвести до порушення перелічених властивостей.

Вимоги до систем захисту даних регулюються державними та міжнародними нормативними документами. У цих документах надано терміни і визначення понять, які є обов'язковими для застосування в усіх видах документації і літератури, що описують системи технічного захисту інформації. Для кожного поняття встановлено один термін. Застосування синонімів терміна не допускається.

Державними нормативними документами визначено вимоги до кожного з етапів створення та функціонування систем захисту, включаючи визначення та аналіз загроз, розробку системи захисту, реалізацію плану захисту, контроль функціонування та керування системою. Особливу увагу приділено оцінюванню захищеності даних в комп'ютерних системах від несанкціонованого доступу [1].

Метою створення цього посібника є надання допомоги студентам при набутті базових знань у галузі захисту інформації в комп'ютерних системах та ознайомлення з напрямами використання цих знань. У посібнику вміщено відомості про функціональні послуги та механізми захисту даних, які обрано у Корпоративній інформаційній мережі будівельного комплексу України. Також розглянуто власні розробки Кафедри кібербезпеки та комп'ютерної інженерії Київського національного університету будівництва і архітектури у співавторстві з Державним науково-дослідним інститутом автоматизованих систем в будівництві (ДНДІАСБ) Міністерства розвитку громад та територій України. Набуття цих знань має полегшити майбутнім фахівцям перехід від навчання до професійної діяльності, допомогти впроваджувати, створювати та використовувати новітні технології захисту даних в інформаційних системах.

Щиру вдячність висловлюю своїм колегам по роботі у ДНДІАСБ за надання інформації про свої розробки, підтримку та цінні зауваження під час підготовки рукопису, а саме: професору Чуприну Володимиру Михайловичу, завідувачу лабораторії технічного та програмного забезпечення інформаційних мереж Тарасюку Дмитру Мефодійовичу, провідному інженеру Бєлкіну Євгену Володимировичу та інженеру Горбунову Олександру Олеговичу.

РОЗДІЛ 1. КОНЦЕПЦІЇ ТА ТЕРМІНОЛОГІЯ СИСТЕМ ЗАХИСТУ ДАНИХ

1.1. Основні поняття та скорочення

Захист даних (*Data protection*) полягає в створенні та підтримці в робочому стані технічних та організаційних заходів, що забезпечують цілісність, конфіденційність і доступність інформації в комп'ютерних системах за умов впливу загроз природного або штучного характеру.

У сучасному суспільстві комп'ютерна інформація (дані) являє собою певну цінність, що має відповідне матеріальне вираження і в залежності від вимог власників інформації (державних установ, підприємств, організацій або фізичних осіб), які визначають розмір цієї цінності, необхідно створювати системи з різними рівнями захисту. Це пов'язано з тим, що вартість систем захисту повинна бути відповідною до розміру збитків у разі можливої реалізації загроз.

Діяльність у галузі захисту даних в Україні регулюється документами Державної служби спеціального зв'язку та захисту інформації України, яка була створена у 2006 році. До того часу функції цієї служби виконував Департамент спеціальних телекомунікаційних систем та захисту інформації Служби безпеки України [2, 3].

У документах цієї Служби регламентуються наступні питання:

- визначення вимог щодо захисту комп'ютерних систем від несанкціонованого доступу;
- створення захищених комп'ютерних систем і засобів їх захисту від несанкціонованого доступу;
- оцінки захищеності комп'ютерних систем і їх придатності для вирішення задач споживача.

Термінологія у даній галузі регламентується нормативним документом [4].

Далі наведемо терміни, що є обов'язковими для застосування в усіх видах документації і літератури, що входить до систем технічного захисту інформації.

Комп'ютерна система; КС (*computer system*) — сукупність програмно-апаратних засобів, для обробки інформації.

Автоматизована система; АС (*automated system*) — організаційно-технічна система, що реалізує інформаційну технологію і об'єднує КС, фізичне середовище, персонал і інформацію, яка обробляється.

Політика безпеки інформації (*information security policy*) — сукупність законів, правил, обмежень, рекомендацій, інструкцій тощо, які регламентують порядок обробки інформації.

Загроза (*threat*) — будь-які обставини або події, що можуть бути причиною порушення політики безпеки інформації і/або нанесення збитків АС.

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

Захист інформації (*information protection, information security, computer system security*) — діяльність, яка спрямована на забезпечення безпеки оброблюваної в АС інформації та АС в цілому, і дозволяє запобігти або ускладнити можливість реалізації загроз, а також знизити величину потенційних збитків внаслідок реалізації загроз.

Комплексна система захисту інформації; КСЗІ — сукупність організаційних і інженерних заходів та програмно-апаратних засобів, які забезпечують захист інформації в АС.

Комплекс засобів захисту; КЗЗ (*trusted computing base*; ТСВ) — сукупність програмно-апаратних засобів, які забезпечують реалізацію політики безпеки інформації.

Захищена комп'ютерна система; (*trusted computer system, trusted computer product*) — комп'ютерна система, що здатна забезпечити захист інформації.

Об'єкт комп'ютерної системи; (*product object, system object*) — елемент ресурсу КС, що характеризується певними атрибутами і поводженням.

Пасивний об'єкт (*passive object*) — об'єкт КС, який в конкретному акті доступу виступає як пасивний компонент системи, над яким виконується дія і/або який служить джерелом чи приймальником інформації.

Об'єкт-процес (*process object*) — виконувана в даний момент програма, яка повністю характеризується своїм контекстом (поточним станом регістрів обчислювальної системи, адресним простором, повноваженнями і т.ін.).

Користувач (*user*) — фізична особа, яка може взаємодіяти з КС через наданий їй інтерфейс.

Об'єкт-користувач (*user object*) — об'єкт, що створюється в процесі входження користувача в систему і повністю характеризується своїм контекстом (псевдонімом, ідентифікаційним кодом, повноваженнями і т. ін.).

Ідентифікатор об'єкта (*object identifier*) — унікальний атрибут об'єкта КС, що дозволяє однозначно виділити даний об'єкт серед подібних.

Потік інформації (*information flow*) — інформація, що передається від одного до іншого об'єкта КС.

Доступ до інформації (*access to information*) — вид взаємодії двох об'єктів КС, внаслідок якого створюється потік інформації від одного об'єкта до іншого і/або відбувається зміна стану системи.

Правила розмежування доступу; ПРД (*access mediation rules*) — частина політики безпеки, що регламентує правила доступу користувачів і процесів до пасивних об'єктів.

Тип доступу (*access type*) — суттєвість доступу до об'єкта, що характеризує зміст здійснюваної взаємодії (наприклад, читання, запис, запуск на виконання, видаляння).

Запит на доступ (*access request*) — звернення одного об'єкта КС до іншого з метою отримання певного типу доступу.

Санкціонований доступ до інформації (*authorized access to information*) — доступ до інформації, що не порушує ПРД.

Несанкціонований доступ до інформації; НСД до інформації (*unauthorized access to information*) — доступ до інформації, здійснюваний з порушенням ПРД.

Захист від несанкціонованого доступу; захист від НСД (*protection from unauthorized access*) — запобігання або істотне утруднення несанкціонованого доступу до інформації.

Право доступу (*access right*) — дозвіл на здійснення певного типу доступу.

Повноваження (*privilege*) — права користувача або процесу на виконання певних дій, зокрема на одержання певного типу доступу до об'єктів.

Керування доступом (*access control*) — сукупність заходів з визначення повноважень і прав доступу, контролю за додержанням ПРД.

Розмежування доступу (*access mediation*) — сукупність процедур, що реалізують перевірку запитів на доступ і оцінку на підставі ПРД можливості надання доступу.

Авторизація (*authorization*) — надання повноважень; встановлення відповідності між повідомленням (пасивним об'єктом) і його джерелом (створившим його користувачем або процесом).

Авторизований користувач (*authorized user*) — користувач, що володіє певними повноваженнями на доступ.

Роль користувача (*user role*) — сукупність функцій щодо керування КС, КЗЗ і обробки інформації, доступних користувачеві.

Адміністратор (*administrator, administrative user*) — користувач, роль якого включає функції керування КС і/або КЗЗ.

Адміністратор безпеки (*security administrator*) — адміністратор, відповідальний за дотримання політики безпеки.

Порушник (*user violator*) — користувач, який здійснює несанкціонований доступ до інформації.

1.2. Дії із даними, їх властивості та загрози у комп'ютерних системах

Користувач через об'єкт-процес може виконувати з даними такі дії:

— ***ознайомлення*** (*disclosure*);

— ***модифікація*** (*modification*).

В залежності від властивостей пасивних та активних об'єктів в комп'ютерних системах ці дії можуть бути легітимними, а можуть являти собою загрозу. Якщо пасивний об'єкт являє собою критичну інформацію, а активний об'єкт не має достатніх повноважень, то маємо загрозу.

Критична інформація (*sensitive information*) — інформація, що вимагає захисту; будь-яка інформація, втрата або неправильне використання якої (модифікація, ознайомлення) може нанести шкоду.

Майже всі комп'ютерні дані у тій чи іншій мірі підпадають під поняття критичної інформації. Цю інформацію характеризують такі вимоги:

— ***конфіденційність*** (*confidentiality*) полягає в забороні на ознайомлення без авторизації;

— ***цілісність*** (*integrity*) полягає в забороні на модифікацію без авторизації;

— ***доступність*** (*availability*) полягає в тому, що користувач або процес, який володіє відповідними повноваженнями, може використовувати ресурс відповідно до правил, встановлених політикою безпеки без зайвих обмежень.

Комп'ютерні системи, що вміщують критичну інформацію, характеризують такі властивості:

— ***цілісність системи*** (*system integrity*) — властивість, яка полягає в тому, що жоден компонент системи не може бути усунений, модифікований або доданий з порушенням політики безпеки;

— ***спостереженість*** (*accountability*) — властивість системи, що дозволяє фіксувати діяльність користувачів і процесів, використання пасивних об'єктів, а також однозначно установлювати ідентифікатори причетних до певних подій користувачів і процесів з метою запобігання порушень політики безпеки і забезпечення відповідальності користувачів.

Щодо розгляду загроз у КС використовують наступні поняття.

Атака (*attack*) — спроба реалізації загрози.

Проникнення (*penetration*) — успішне подолання механізмів захисту системи.

Вразливість системи (*system vulnerability*) — нездатність системи протистояти реалізації певної загрози або сукупності загроз.

Компрометація (*compromise*) — порушення політики безпеки; несанкціоноване ознайомлення.

Втрата інформації (*information leakage*) — неконтрольоване розповсюдження інформації, що веде до її несанкціонованого одержання.

Прихований канал (*covert channel*) — спосіб одержання інформації за рахунок використання шляхів передачі інформації, існуючих у КС, але не керованих КЗЗ, або спостереження за існуючими потоками інформації.

Прихований канал з пам'яттю (*storage covert channel*) — прихований канал, що реалізується шляхом прямого або непрямого запису інформації в певну область пам'яті одним процесом і прямим чи непрямим читанням даної області пам'яті іншим процесом.

Часовий прихований канал (*timing covert channel*) — прихований канал, що дозволяє передавати інформацію від одного процесу до іншого шляхом модулювання першим процесом часових характеристик системи (наприклад, часу зайнятості центрального процесора), що спостерігаються іншим процесом.

Пропускна здатність прихованого каналу (*covert channel bandwidth*) — кількість інформації, що одержується використанням прихованого каналу за одиницю часу.

Відмова (*fault, failure*) — втрата здатності КС або її компонента виконувати певну функцію.

Відмова в обслуговуванні (*denial of service*) — будь-яка дія або послідовність дій, що призводять будь-яку частину (компонент) системи до виходу із ладу; нездатність системи виконувати свої функції (надавати декларовані послуги) внаслідок виходу із ладу якого-небудь компонента або інших причин.

Комп'ютерний вірус (*computer virus*) — програма, що володіє здатністю до самовідтворення і, як правило, здатна здійснювати дії, які можуть порушити функціонування КС і/або зумовити порушення політики безпеки.

Програмна закладка (*program bug*) — потайно впроваджена програма або недокументовані властивості програмного забезпечення, використання яких може призвести до обходу КЗЗ і/або порушення політики безпеки.

Люк (*trap door*) — залишені розробником недокументовані функції, використання яких дозволяє обминути механізми захисту.

Троянський кінь (*Trojan horse*) — програма, яка, будучи авторизованим процесом, окрім виконання документованих функцій, здатна здійснювати приховані дії.

1.3. Основні поняття щодо розробки та експлуатації систем захисту

Створення системи захисту починається з аналізу об’єкта захисту і моделювання можливих загроз. Передусім мають бути визначені ресурси, що підлягають захисту. Загрози мають бути визначені в термінах ймовірності їх реалізації і величини можливих збитків. На підставі аналізу загроз, існуючих в системі, мають бути оцінені ризики. Ризик являє собою функцію ймовірності реалізації певної загрози та розміру можливих збитків. Величина ризику може бути виражена в грошовому вимірі або у вигляді формальної оцінки (високий, низький і т.ін.). На підставі виконаної роботи мають бути вироблені заходи захисту, перетворення яких в життя дозволило б знизити рівень остаточного ризику до прийнятного рівня. Підсумком даного етапу робіт повинна стати сформульована або скоригована політика безпеки.

На підставі проведеного аналізу ризиків розробляється план захисту, який включає в себе опис послідовності і змісту всіх стадій і етапів життєвого циклу системи захисту, що має відповідати стадіям і етапам життєвого циклу АС. Вартість заходів щодо захисту інформації має бути адекватною розміру можливих збитків.

Для реалізації політики безпеки КЗЗ повинен забезпечити ізоляцію об'єктів всередині сфери управління і гарантувати розмежування запитів доступу і керування потоками інформації між об'єктами. Для цього з об'єктами КС має бути пов'язана інформація, що дозволяла б КЗЗ ідентифікувати об'єкти і перевіряти легальність запитів доступу.

Кожний об'єкт КС повинен мати певний набір атрибутів доступу, який включає унікальний ідентифікатор та іншу інформацію, що визначає його права доступу і/або права доступу до нього. Атрибут доступу — термін для опису будь-якої інформації, яка використовується при керуванні доступом і зв'язана з користувачами, процесами або пасивними об'єктами.

Коли користувачі або процеси намагаються одержати доступ до пасивних об'єктів, механізми, що реалізують керування доступом, на підставі політики безпеки і перевірки атрибутів доступу можуть "прийняти рішення" про

легальність запиту. Використовуючи набір атрибутів доступу відповідно до прийнятої політики безпеки, можна реалізувати довірче керування доступом, адміністративне, контроль за цілісністю та інші види керування доступом.

Щодо розробки та експлуатації систем захисту використовують наступні поняття.

Модель загроз (*model of threats*) — абстрактний формалізований або неформалізований опис методів і засобів здійснення загроз.

Модель порушника (*user violator model*) — абстрактний формалізований або неформалізований опис порушника.

Ризик (*risk*) — функція ймовірності реалізації певної загрози, виду і величини завданих збитків.

Аналіз ризику (*risk analysis*) — процес визначення загроз безпеці інформації та їх характеристик, слабких сторін КСЗІ (відомих і припустимих), оцінки потенційних збитків від реалізації загроз та ступеню їх прийнятності для експлуатації АС.

Керування ризиком (*risk management*) — сукупність заходів, що проводяться протягом всього життєвого циклу АС щодо оцінки ризику, вибору, реалізації і впровадження заходів забезпечення безпеки, спрямована на досягнення прийнятного рівня залишкового ризику.

Залишковий ризик (*residual risk*) — ризик, що залишається після впровадження заходів забезпечення безпеки.

Заходи забезпечення безпеки (*safeguards*) — послуги, функції, механізми, правила і процедури, призначені для забезпечення захисту інформації.

Послуга безпеки (*security service*) — сукупність функцій, що забезпечують захист від певної загрози або від множини загроз.

Механізми захисту (*security mechanism*) — конкретні процедури і алгоритми, що використовуються для реалізації певних функцій і послуг безпеки.

Засоби захисту (*protection facility*) — програмні, програмно-апаратні та апаратні засоби, що реалізують механізми захисту.

Політика безпеки послуги (*service security policy*) — правила, згідно з якими функціонують механізми, що реалізують послугу.

Рівень послуги (*level of service*) — міра ефективності і/або стійкості механізмів, що реалізують послугу, відносно до введеної для даної послуги шкали оцінки.

Гарантії (*assurance*) — сукупність вимог (шкала оцінки) для визначення міри упевненості, що КС коректно реалізує політику безпеки.

Рівень гарантій (*assurance level*) — міра упевненості в тому, що КС коректно реалізує політику безпеки.

Модель політики безпеки (*security policy model*) — абстрактний формалізований або неформалізований опис політики безпеки інформації.

Домен комп'ютерної системи (*domain*) — ізольована логічна область КС, що характеризується унікальним контекстом, всередині якої об'єкти володіють певними властивостями, повноваженнями і зберігають певні відносини між собою.

Тестування на проникання (*penetration testing*) — випробування, метою яких є здійснення спроби обминути або відключити механізми захисту.

Оцінка вразливості (*vulnerability assessment*) — дослідження об'єкта оцінки з метою визначення можливості реалізації загроз.

Оцінка безпеки інформації (*information security evaluation*) — процес, метою якого є визначення відповідності стану безпеки інформації в КС встановленим вимогам.

Критерії оцінки захищеності (*security evaluation criteria*) — сукупність вимог (шкала оцінки), що використовується для оцінки ефективності функціональних послуг безпеки і коректності їх реалізації.

Рейтинг (*rating*) — упорядкований перелік рівнів послуг і рівня гарантій, виявлених в процесі оцінки КС.

Функціональний профіль (*functionality profile*) — упорядкований перелік рівнів функціональних послуг, який може використовуватись як формальна специфікація функціональності КС.

1.4. Принципи побудови механізмів захисту даних

Реалізація функцій захисту базується на концепції матриці доступу. Ця матриця являє собою таблицю, уздовж кожного виміру якої відкладені ідентифікатори об'єктів КС, а в якості елементів матриці виступають дозволені або заборонені режими доступу. Матриця доступу може бути двомірною (наприклад, користувачі/пасивні об'єкти або процеси/пасивні об'єкти) або тримірною (користувачі/процеси/пасивні об'єкти). Матриця доступу може бути повною, тобто містити вздовж кожної з осей ідентифікатори всіх існуючих на даний час об'єктів КС даного типу, або частковою. Повна тримірна матриця доступу дозволяє точно описати, хто (ідентифікатор користувача), через що

(ідентифікатор процесу), до чого (ідентифікатор пасивного об'єкта), який вид доступу може одержати.

Основними завданнями засобів захисту є ізоляція об'єктів КС всередині сфери керування, перевірка всіх запитів доступу до об'єктів і реєстрація запитів і результатів їх перевірки і/або виконання. З одного боку, будь-яка елементарна функція будь-якої з послуг, що реалізуються засобами захисту, може бути віднесена до функцій ізоляції, перевірки або реєстрації. З іншого боку, будь-яка з функцій, що реалізуються засобами захисту, може бути віднесена до функцій забезпечення конфіденційності, цілісності і доступності інформації або керованості КС і спостереженості дій користувачів.

Кожна функція може бути реалізована одним або більше внутрішніми механізмами, що залежать від конкретної КС. Водночас одні й ті ж самі механізми можуть використовуватись для реалізації кількох послуг. Наприклад, для розробника слушно реалізувати і адміністративне і довірче керування доступом єдиним набором механізмів.

Реалізація механізмів може бути абсолютно різною. Для реалізації функцій захисту можуть використовуватись програмні або апаратні засоби, криптографічні перетворення, різні методи перевірки повноважень і т. ін. Вибір методів і механізмів практично завжди залишається за розробником. Єдиною вимогою залишається те, щоб функції захисту були реалізовані відповідно до декларованої політики безпеки і вимог гарантій.

Для реалізації певних послуг можуть використовуватись засоби криптографічного захисту. Криптографічні перетворення можуть використовуватись безпосередньо для захисту певної інформації (наприклад, при реалізації послуг конфіденційності) або підтримувати реалізацію послуги (наприклад, при реалізації послуги ідентифікації і автентифікації). Згідно із законодавством створення переліку вимог, сертифікація і атестація систем шифрування покладається на відповідний уповноважений орган виконавчої влади. Ця діяльність регламентується “Положенням про порядок здійснення криптографічного захисту інформації в Україні”.

Для опису механізмів захисту використовують наступні поняття.

Довірче керування доступом (*discretionary access control*) — принцип керування доступом, який полягає в тому, що звичайним користувачам дозволено керувати (довіряють керування) потоками інформації між іншими користувачами і об’єктами свого домена (наприклад, на підставі права володіння об’єктами) без втручання адміністратора.

Адміністративне керування доступом (*mandatory access control*) — принцип керування доступом, який полягає в тому, що керувати потоками інформації між користувачами і об'єктами дозволено тільки спеціально авторизованим користувачам, а звичайні користувачі не мають можливості створити потоки інформації, які могли б призвести до порушення встановлених ПРД.

Експорт інформації (*information export*) — виведення інформації з під керування КЗЗ назовні.

Імпорт інформації (*information import*) — уведення інформації ззовні під керування КЗЗ.

Ідентифікація (*identification*) — процедура присвоєння ідентифікатора об'єкту КС або встановлення відповідності між об'єктом і його ідентифікатором; впізнання.

Автентифікація (*authentication*) — процедура перевірки відповідності пред'явленого ідентифікатора об'єкта КС на предмет належності його цьому об'єкту; встановлення або підтвердження автентичності.

Інформація автентифікації (*authentication information*) — інформація, що використовується для автентифікації.

Пароль (*password*) — секретна інформація автентифікації, що являє собою послідовність символів, яку користувач повинен ввести через обладнання вводу інформації, перш ніж йому буде надано доступ до КС або до інформації.

Персональний ідентифікаційний номер; ПІН (*personal identification number, PIN*) — вид паролю, що звичайно складається тільки із цифр, і який, як правило, має бути пред'явлений нарівні з носимим ідентифікатором.

Достовірний канал (*trusted path*) — захищений шлях передачі інформації між користувачем і КЗЗ, що не може бути імітований, а інформація, що передається ним, не може бути отримана або модифікована стороннім користувачем або процесом.

Реєстрація (*audit, auditing*) — послуга, що забезпечує збирання і аналіз інформації щодо використання користувачами і процесами функцій і об'єктів, контрольованих КЗЗ.

Журнал реєстрації (*audit trail*) — упорядкована сукупність реєстраційних записів, кожен з яких заноситься КЗЗ за фактом здійснення контрольованої події.

Довірча конфіденційність (*discretionary confidentiality*) — послуга, що забезпечує конфіденційність інформації відповідно до принципів довірчого керування доступом.

Адміністративна конфіденційність (*mandatory confidentiality*) — послуга, що забезпечує конфіденційність інформації відповідно до принципів адміністративного керування доступом.

Довірча цілісність (*discretionary integrity*) — послуга, що забезпечує цілісність інформації відповідно до принципів довірчого керування доступом.

Адміністративна цілісність (*mandatory integrity*) — послуга, що забезпечує цілісність інформації відповідно до принципів адміністративного керування доступом.

Очищення пам'яті (*memory clearing*) — знищення даних в пам'яті шляхом встановлення полів цих даних в заданий або випадковий стан.

Розділюваний об'єкт (*shared object*) — об'єкт КС, який одночасно або почергово використовується різними користувачами і/або процесами.

Повторне використання об'єкта (ob*ject reuse*) — послуга, що забезпечує очищення пам'яті і призупинення дії повноважень щодо розділюваного об'єкта, який раніше використовувався одним користувачем або процесом, перед наданням його іншому користувачеві або процесу.

Аналіз прихованих каналів (*covert channels analyse*) — послуга, яка забезпечує гарантію того, що приховані канали в КС відсутні, знаходяться під наглядом або, принаймні, відомі.

Керування потоками (*flow control*) — сукупність функцій і процедур, які забезпечують неможливість передачі інформації прихованими каналами, тобто в обхід КЗЗ. В більш вузькому значенні часто розуміється сукупність процедур, які забезпечують неможливість передачі інформації від об'єкта КС з більш високим рівнем доступу до об'єкта КС з більш низьким рівнем доступу.

Відкат (*rollback*) — послуга, що забезпечує повернення об'єкта КС до відомого попереднього стану після виконання над об'єктом певної операції або серії операцій.

Квота (*quota*) — обмеження можливості використання певного ресурсу КС користувачем або процесом.

Стійкість до відмов (*fault tolerance*) — послуга, що забезпечує здатність КС продовжувати функціонування в умовах виникнення збоїв і відмов окремих компонентів.

Ініціалізація (*initialization*) — встановлення системи або об'єкта у відомий чи визначений стан.

Диспетчер доступу (*reference monitor*) — реалізація концепції абстрактного автомата, яка забезпечує дотримання ПРД і характеризується

такими трьома особливостями: забезпечує безперервний і повний контроль за доступом, захищений від модифікації і має невеликі розміри.

Ядро захисту (*security kernel*) — частина КЗЗ, в якій зосереджено мінімально необхідний набір механізмів, що реалізують ПРД.

Атрибут доступу (*tag, access mediation information*) — будь-яка зв'язана з об'єктом КС інформація, яка використовується для керування доступом.

Матриця доступу (*access matrix*) — n-мірна таблиця, вздовж кожного виміру якої відкладені ідентифікатори об'єктів КС одного типу (об'єктів-користувачів, об'єктів-процесів чи пасивних об'єктів), і містить як елементи права доступу за кожним із типів доступу.

Список доступу (*access control list*) — перелік користувачів і/або процесів з зазначенням їх прав доступу до об'єкта КС, з яким пов'язаний цей перелік.

Список повноважень (*privilege list, profile*) — перелік об'єктів з зазначенням прав доступу до них з боку користувача або процесу, з яким пов'язаний цей перелік.

Мітка (*label*) — атрибут доступу, що відображає категорію доступу об'єкта КС.

Категорія доступу (*security level*) — комбінація ієрархічних і неієрархічних атрибутів доступу, що відображає рівень критичності (наприклад, конфіденційності) інформації або повноважень користувача щодо доступу до такої інформації.

Рівень доступу (*access level*) — ієрархічна частина категорії доступу пасивного об'єкта.

Рівень допуску (*clearance*) — ієрархічна частина категорії доступу користувача або процесу, що визначає максимальний рівень доступу пасивного об'єкта, до якого може одержати доступ користувач чи процес.

Криптографічне перетворення — перетворення даних, яке полягає в їх шифруванні, вироблення імітовставки або цифрового підпису.

Шифрування даних — процес зашифрування або розшифрування.

Зашифрування даних (*data encryption*) — процес перетворення відкритого тексту в шифртекст.

Розшифрування даних (*data decryption*) — процес перетворення шифртексту у відкритий текст.

Відкритий текст (*clear text*) — дані з доступним семантичним змістом.

Шифртекст (*ciphertext*) — дані, отримані у результаті зашифрування відкритого тексту.

Ключ (*key*) — конкретний стан деяких параметрів алгоритму криптографічного перетворення, що забезпечує вибір одного перетворення із сукупності можливих для даного алгоритму.

Імітовставка (*data authentication code*) — блок інформації фіксованої довжини, що одержується із відкритого тексту і ключа, однозначно відповідний даному відкритому тексту.

Цифровий підпис (*digital signature*) — дані, одержані в результаті криптографічного перетворення блоку даних і/або його параметрів (геш-функції, довжини, дати утворення, ідентифікатора відправника і т. ін.), що дозволяють приймальнику даних впевнитись в цілісності блоку і справжності джерела даних і забезпечити захист від підробки і підлогу.

Завірення (*notarization*) — реєстрація даних у довіреної третьої особи з метою забезпечення надалі впевненості в правильності таких характеристик як зміст, джерело даних, час відправлення чи одержання тощо.

Висновки

1. У сучасному суспільстві комп’ютерна інформація (дані) являє собою певну цінність, що має відповідне матеріальне вираження і в залежності від вимог власників інформації (державних установ, підприємств, організацій або фізичних осіб), які визначають розмір цієї цінності, необхідно створювати системи захисту з різними рівнями захищеності в залежності від визначеного розміру цінності.
2. Діяльність у галузі захисту даних в Україні регулюється документами Державної служби спеціального зв'язку та захисту інформації України. Термінологія у цій галузі також регламентується нормативними документами цієї Служби.
3. Захист даних полягає в створенні та підтримці в робочому стані технічних та організаційних заходів, що забезпечують цілісність, конфіденційність і доступність інформації в комп’ютерних системах за умов впливу загроз природного або штучного характеру. Цілісність полягає в забороні на модифікацію без авторизації, конфіденційність полягає в недопущенні несанкціонованого ознайомлення з об’єктом захисту, а доступність полягає в тому, щоб користувач, який наділений відповідними повноваженнями, мав доступ до необхідних об’єктів відповідно до правил, встановлених політикою безпеки, без зайвих обмежень.

4. Дані, що підлягають захисту, прийнято називати критичними пасивними об’єктами, а об’єкти, які потенційно можуть утворювати загрози, прийнято розподіляти на об’єкти-процеси та об’єкти-користувачі, при цьому обидва вони являють собою активну (діючу) комп’ютерну програму.
5. Можливість загроз комп’ютерній інформації утворює наявність прихованих каналів, які існують у КС, але не керуються КЗЗ, комп’ютерних вірусів, програмних закладок та програм типу Троянський кінь.
6. Створення системи захисту починається з аналізу об’єкта захисту і моделювання можливих загроз. Передусім мають бути визначені ресурси, що підлягають захисту. Загрози мають бути визначені в термінах ймовірності їх реалізації і величини можливих збитків. На підставі аналізу загроз, існуючих в системі, мають бути оцінені ризики. Ризик являє собою функцію ймовірності реалізації певної загрози та розміру можливих збитків. Величина ризику може бути виражена в грошовому вимірі або у вигляді формальної оцінки (високий, низький і т.ін.).
7. На підставі аналізу ризиків розробляється план захисту, який включає в себе опис послідовності і змісту всіх стадій і етапів життєвого циклу системи захисту, що має відповідати стадіям і етапам життєвого циклу АС. Вартість заходів щодо захисту інформації має бути адекватною розміру можливих збитків.
8. Для реалізації політики безпеки КЗЗ повинен забезпечити ізоляцію об'єктів всередині сфери управління і гарантувати розмежування запитів доступу і керування потоками інформації між об'єктами. Для цього з об'єктами КС має бути пов'язана інформація, що дозволяла б КЗЗ ідентифікувати об'єкти і перевіряти легальність запитів доступу.
9. Реалізація функцій захисту базується на концепції матриці доступу. Ця матриця являє собою таблицю, уздовж кожного виміру якої відкладені ідентифікатори об'єктів КС, а в якості елементів матриці виступають дозволені або заборонені режими доступу. Матриця доступу може бути двомірною (наприклад, користувачі/пасивні об'єкти або процеси/пасивні об'єкти) або тримірною (користувачі/процеси/пасивні об'єкти).
10. Основними завданнями засобів захисту є ізоляція об'єктів КС всередині сфери керування, перевірка всіх запитів доступу до об'єктів і реєстрація запитів і результатів їх перевірки і/або виконання. З одного боку, будь-яка елементарна функція будь-якої з послуг, що реалізуються засобами

захисту, може бути віднесена до функцій ізоляції, перевірки або реєстрації. З іншого боку, будь-яка з функцій, що реалізуються засобами захисту, може бути віднесена до функцій забезпечення конфіденційності, цілісності і доступності інформації або керованості КС і спостереженості дій користувачів.

11. Реалізація механізмів може бути абсолютно різною. Для реалізації функцій захисту можуть використовуватись програмні або апаратні засоби, криптографічні перетворення, різні методи перевірки повноважень і т. ін. Вибір методів і механізмів практично завжди залишається за розробником. Єдиною вимогою залишається те, щоб функції захисту були реалізовані відповідно до декларованої політики безпеки і вимог гарантій.
12. Криптографічні перетворення можуть використовуватись безпосередньо для захисту певної інформації (наприклад, при реалізації послуг конфіденційності) або підтримувати реалізацію послуги (наприклад, при реалізації послуги ідентифікації і автентифікації). Згідно із законодавством створення переліку вимог, сертифікація і атестація систем шифрування покладається на відповідний уповноважений орган виконавчої влади. Ця діяльність регламентується “Положенням про порядок здійснення криптографічного захисту інформації в Україні”.

Запитання та завдання для самоперевірки

1. Які питання у галузі захисту даних в Україні регламентуються документами Департаменту спеціальних телекомунікаційних систем та захисту інформації Служби безпеки України?
2. Поясніть чим відрізняється поняття комп’ютерної системи від поняття автоматизованої системи.
3. Які властивості інформації забезпечують за допомогою засобів захисту даних?
4. У якому співвідношенні знаходяться поняття пасивних об’єктів, критичних об’єктів, об’єктів-користувачів та об’єктів-процесів?
5. Які об’єкти або явища можуть утворювати загрози для комп’ютерної інформації?
6. Яка послідовність дій передбачена нормативними документами на початку створення системи захисту комп’ютерної інформації?
7. Яким чином оцінюється вартість заходів щодо захисту інформації?

8. На яких процедурах засновано принцип розмежування доступу та яку роль відіграє цей принцип у побудові систем захисту даних?
9. Що являє собою матриця доступу та які існують особливості її побудови?
10. Поясніть чим відрізняється довірче керування доступом від адміністративного.
11. Яким вимогам повинні відповідати функціональні послуги захисту та хто несе відповідальність за вибір тих чи інших механізмів захисту?
12. На кого покладається відповідальність за створення переліку вимог, сертифікацію і атестацію систем шифрування? Чим регламентується ця діяльність?

РОЗДІЛ 2. ХАРАКТЕРИСТИКИ ФУНКЦІОНАЛЬНИХ ПОСЛУГ ЗАХИСТУ ТА ВЛАСТИВОСТІ МЕХАНІЗМІВ ДЛЯ ЇХ РЕАЛІЗАЦІЇ

2.1. Структура автоматизованої системи з комплексом засобів захисту даних

Дані, що підлягають захисту являють собою пасивні об'єкти комп'ютерної системи. Найчастіше це є файли на жорстких дисках, але під час роботи з ними копії цих файлів перебувають у оперативній пам'яті. Підмножину з цих пасивних об'єктів, що потребує захисту, називають критичними об'єктами.

Об'єкти, що можуть бути загрозою для безпеки даних, являють собою комп'ютерні програми, що знаходяться в оперативній пам'яті у стадії виконання. Ці об'єкти є активними, їх прийнято називати процесами.

Для кожного користувача автоматизованої системи на період взаємодії з комп'ютерною системою утворюється об'єкт-користувач. Цей об'єкт є активним процесом, що реагує на дії користувача (суб'єкта) та в залежності від цих дій викликає інші активні процеси, які в свою чергу можуть звертатись як до активних, так і до пасивних об'єктів комп'ютерної системи. При цьому існує можливість завдання шкоди власникам інформації через неправомочні або безвідповідальні дії користувачів. Через такі дії користувачі можуть стати порушниками безпеки інформації.

Щоб уникнути загроз від порушників безпеки використовують метод розмежування доступу користувачів до тих чи інших інформаційних ресурсів комп'ютерної системи. Це розмежування робиться на основі прийнятої власниками інформації політики безпеки.

Реалізація політики безпеки інформації покладається на комплекс засобів захисту (КЗЗ), який повинен контролювати та обмежувати потоки інформації, що утворюються між об'єктами комп'ютерної системи (КС).

Метою створення КЗЗ є усунення можливостей завдання шкоди інформаційним ресурсам КС з боку порушників, що передбачені у моделі порушника і ліквідації загроз, що передбачені у моделі загроз, які показані на рис. 2.1.

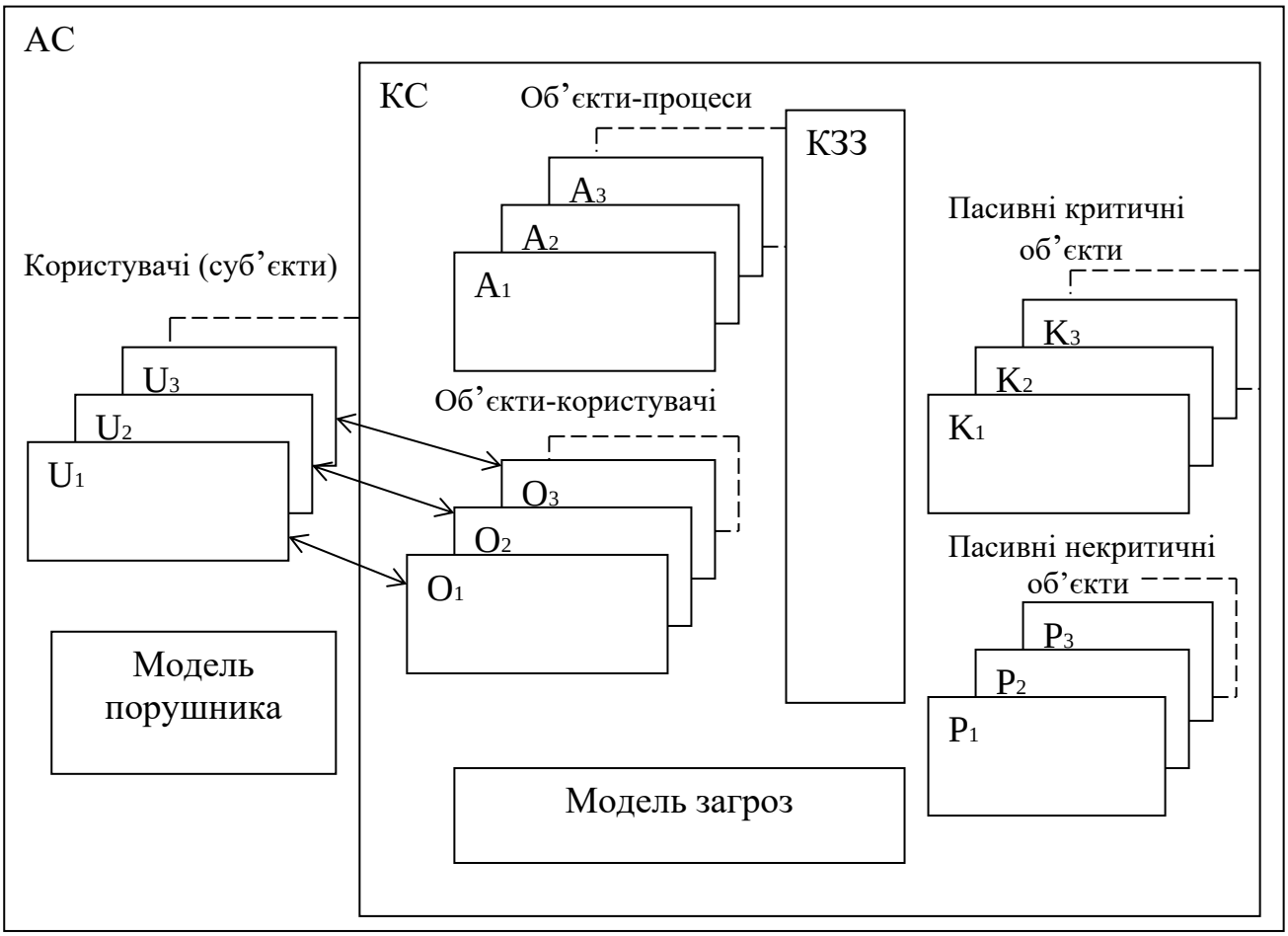

Рис. 2.1. Структурна схема автоматизованої системи АС, що обладнана комплексом засобів захисту КЗЗ

На першому етапі побудови системи захисту необхідно визначити від яких саме загроз слід захищати критичні об’єкти. Також бажано визначити у грошовій оцінці розмір втрат від реалізації кожної із цих загроз. Результатом першого етапу є побудова моделі загроз, що являє собою опис методів та засобів здійснення кожної загрози до кожного з об’єктів захисту. У якості загроз можуть бути описані наступні процеси:

— витік інформації (неконтрольоване поширення конфіденційної інформації, яке призводить до її несанкціонованого одержання);

— блокування інформації (унеможливлення доступу до інформації для санкціонованих користувачів);

— порушення цілісності інформації (спотворення інформації, її руйнування або знищення).

У моделі загроз повинні бути описані можливі шляхи здійснення загрози. Це може бути несанкціонований доступ з робочого місця користувача або можливість побудови неконтрольованого технічного каналу. Також слід передбачити можливість проникнення комп’ютерних вірусів. У разі особливо

цінної інформації може бути розглянуто застосування закладних пристроїв чи програм.

Крім моделювання загроз на першому етапі створюють абстрактний формалізований або неформалізований опис можливого порушника. Цей опис прийнято називати моделлю порушника.

У якості порушника розглядається особа, яка має доступ до роботи з КС. Порушники класифікуються за рівнем можливостей, що надаються їм штатними засобами КС. Прийнято виділяти чотири наступні рівні цих можливостей:

— перший рівень визначає найнижчий рівень можливостей проведення діалогу з КС — можливість запуску фіксованого набору програм, що реалізують заздалегідь передбачені функції обробки інформації;

— другий рівень визначається можливістю створення і запуску власних програм з новими функціями обробки інформації;

— третій рівень визначається можливістю управління КС, тобто впливом на базове програмне забезпечення системи і на конфігурацію її устаткування;

— четвертий рівень визначається всім обсягом можливостей осіб, що здійснюють проектування, реалізацію і ремонт апаратних компонентів КС, аж до включення до складу КС власних засобів з новими функціями обробки інформації.

Припускається, що в своєму рівні порушник — це фахівець вищої кваліфікації, який має повну інформацію про КС і КЗЗ.

Така класифікація порушників є корисною для використання в процесі оцінки ризиків, аналізу вразливості системи, ефективності існуючих і планових заходів захисту.

На другому етапі побудови системи захисту розробляється план, що містить організаційні та технічні заходи захисту, визначається зона безпеки інформації. Для технічного захисту даних застосовується спосіб їх приховування від можливих порушників. Можливо також використовувати спосіб технічної дезінформації. У всіх випадках заходи захисту повинні бути відповідними загрозам. Вартість цих заходів не повинна перевищувати розмір шкоди від реалізації можливих загроз. Порядок розрахунку ефективності систем захисту та порядок їх атестації встановлюється нормативними документами системи ТЗІ [3].

На третьому етапі провадиться реалізація та атестація системи захисту. Надання послуг з технічного захисту інформації, атестації систем захисту та сервісне обслуговування цих систем можуть здійснювати особи, що мають ліцензію на право проведення відповідних робіт.

На четвертому етапі створення системи захисту провадиться аналіз функціонування системи та контроль ефективності її роботи. За фактами виявлення недоліків функціонування або нових загроз слід у найкоротший строк реалізувати додаткові заходи для забезпечення безпеки інформації. Порядок проведення перевірок і контролю ефективності системи захисту встановлюється нормативними документами системи ТЗІ [3].

Нормативні документи ТЗІ забезпечують загальну технічну політику безпеки інформації, створення і розвиток єдиної термінології систем ТЗІ, сертифікацію, ліцензування й атестацію систем захисту, розвиток послуг у галузі ТЗІ та порядок підготовки кадрів для цієї галузі.

2.2. Критерії оцінки захищеності даних у інформаційних системах

Одним з основних нормативних документів ТЗІ є Критерії оцінки захищеності інформації в комп’ютерних системах від несанкціонованого доступу (далі — Критерії) [3]. Цей документ є методологічною базою для визначення вимог з захисту інформації в комп'ютерних системах від несанкціонованого доступу; створення захищених комп'ютерних систем і засобів захисту від несанкціонованого доступу; оцінки захищеності інформації в комп'ютерних системах і їх придатності для обробки критичної інформації.

Критерії можуть застосовуватися до всього спектра комп'ютерних систем, включаючи однорідні системи, багатопроцесорні системи, бази даних, вбудовані системи, розподілені системи, мережі, об'єктно-орієнтовані системи та ін.

Критерії надають порівняльну шкалу для оцінки надійності механізмів захисту інформації від несанкціонованого доступу, реалізованих в комп'ютерних системах, та базу (орієнтири) для розробки комп'ютерних систем, в яких мають бути реалізовані функції захисту інформації.

В контексті Критеріїв система захисту розглядається як набір функціональних послуг. Кожна послуга являє собою набір функцій, що дозволяють протистояти певній множині загроз. Кожна послуга може включати декілька рівнів. Чим вище рівень послуги, тим більш повно забезпечується захист від певного виду загроз. Рівні послуг мають ієрархію за повнотою захисту, проте не обов'язково являють собою точну підмножину один одного. Рівні починаються з першого і зростають до значення n, де n — число, що залежить від конкретного виду послуги.

Функціональні критерії розбиті на чотири групи, кожна з яких описує вимоги до послуг, що забезпечують захист від загроз одного із наступних чотирьох типів.

Конфіденційність. Загрози, що відносяться до несанкціонованого ознайомлення з інформацією, становлять загрози конфіденційності. Якщо існують вимоги щодо обмеження можливості ознайомлення з інформацією, то відповідні послуги треба шукати в розділі "Критерії конфіденційності". В цьому розділі описані послуги довірчої та адміністративної конфіденційності, повторне використання об'єктів, аналіз прихованих каналів та конфіденційність при обміні (експорті/імпорті).

Цілісність. Загрози, що відносяться до несанкціонованої модифікації інформації, становлять загрози цілісності. Якщо існують вимоги щодо обмеження можливості модифікації інформації, то відповідні послуги треба шукати в розділі "Критерії цілісності". В цьому розділі описані послуги довірчої та адміністративної цілісності, відкат і цілісність при обміні.

Доступність. Загрози, що відносяться до порушення можливості використання комп'ютерних систем або оброблюваної інформації, становлять загрози доступності. Якщо існують вимоги щодо захисту від відмови в доступі або захисту від збоїв, то відповідні послуги треба шукати в розділі "Критерії доступності". В цьому розділі описані такі послуги: використання ресурсів, стійкість до відмов, гаряча заміна, відновлення після збоїв.

Спостереженість. Ідентифікація і контроль за діями користувачів, керованість комп'ютерною системою становлять предмет послуг спостереженості і керованості. Якщо існують вимоги щодо контролю за діями користувачів або легальністю доступу і за спроможністю комплексу засобів захисту виконувати свої функції, то відповідні послуги треба шукати у розділі "Критерії спостереженості". В цьому розділі описані такі послуги: реєстрація, ідентифікація і автентифікація, достовірний канал, розподіл обов'язків, цілісність комплексу засобів захисту, самотестування, автентифікація при обміні, автентифікація відправника (невідмова від авторства), автентифікація одержувача (невідмова від одержання).

Крім функціональних критеріїв, що дозволяють оцінити наявність послуг безпеки в комп'ютерній системі, є критерії гарантій, що дозволяють оцінити коректність реалізації послуг. Критерії гарантій включають вимоги до архітектури комплексу засобів захисту, середовища розробки, послідовності розробки, випробування комплексу засобів захисту, середовища функціонування і експлуатаційної документації. Існують сім рівнів гарантій (Г-1, ..., Г-7), які є

ієрархічними. Ієрархія рівнів гарантій відбиває поступово наростаючу міру певності в тому, що реалізовані в комп'ютерній системі послуги дозволяють протистояти певним загрозам, що механізми, які їх реалізують, в свою чергу коректно реалізовані і можуть забезпечити очікуваний споживачем рівень захищеності інформації під час експлуатації комп'ютерної системи.

Структуру Критеріїв показано на рисунку (рис. 2.2).

2.2.1. Критерії конфіденційності

Конфіденційність забезпечується послугами довірчої та адміністративної конфіденційності, забезпеченням безпечного повторного використання об'єктів, аналізом прихованих каналів і конфіденційністю при обміні.

Послуга довірчої конфіденційності надається у системах з довірчим керуванням доступом, у яких користувачам дозволено керувати потоками інформації між іншими користувачами і об’єктами без втручання адміністратора. Рівні даної послуги розподіляються на підставі повноти захисту і вибірковості керування наступним чином.

— КД-1 — Мінімальна довірча конфіденційність при якій розмежування доступу здійснюється на підставі атрибутів процесів і захищених об’єктів.

— КД-2 — Базова довірча конфіденційність при якій розмежування доступу здійснюється на підставі атрибутів користувачів і захищених об’єктів.

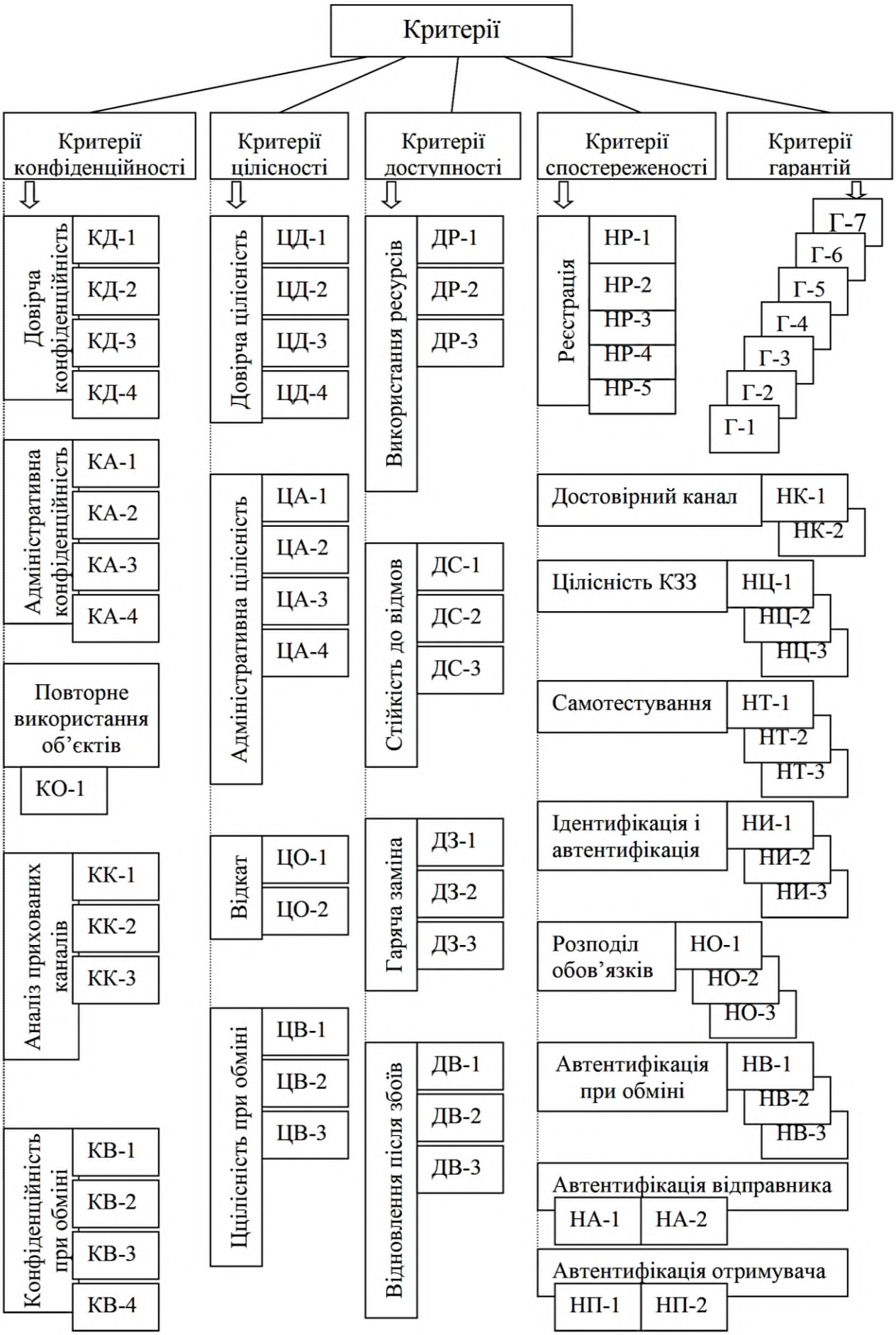

Рис. 2.2. Структура Критеріїв

— КД-3 — Повна довірча конфіденційність при якій розмежування доступу здійснюється на підставі атрибутів користувачів і захищених об’єктів. При цьому визначаються користувачі або групи користувачів, що не мають права одержувати інформацію від захищених об’єктів.

— КД-4 — Абсолютна довірча конфіденційність при якій розмежування доступу здійснюється на підставі атрибутів користувачів, процесів і захищених об’єктів. При цьому визначаються користувачі і процеси або групи користувачів і процесів, що не мають права одержувати інформацію від захищених об’єктів.

Послуга адміністративної конфіденційності надається у системах з адміністративним керуванням доступом, у яких керувати потоками інформації між користувачами і об’єктами покладено на адміністратора. Рівні даної послуги розподіляються на підставі повноти захисту і вибірковості керування наступним чином.

— КА-1 — Мінімальна адміністративна конфіденційність при якій розмежування доступу здійснюється на підставі атрибутів процесів і захищених об’єктів.

— КА-2 — Базова адміністративна конфіденційність при якій розмежування доступу здійснюється на підставі атрибутів користувачів і захищених об’єктів.

— КА-3 — Повна адміністративна конфіденційність при якій розмежування доступу здійснюється на підставі атрибутів користувачів і захищених об’єктів. При цьому визначаються користувачі або групи користувачів, що не мають права одержувати інформацію від захищених об’єктів.

— КА-4 — Абсолютна адміністративна конфіденційність при якій розмежування доступу здійснюється на підставі атрибутів користувачів, процесів і захищених об’єктів. При цьому визначаються користувачі і процеси або групи користувачів і процесів, що не мають права одержувати інформацію від захищених об’єктів.

Послуга повторне використання об’єктів (КО-1) гарантує, що в разі, якщо об'єкт виділяється новому користувачу або процесу, то він не містить інформації, яка залишилась від попереднього користувача або процесу.

Послуга аналізу прихованих каналів виконується з метою виявлення і усунення потоків інформації, які існують, але не контролюються іншими послугами. Рівні даної послуги розподіляються на підставі того, чи виконується тільки виявлення, контроль або перекриття прихованих каналів наступним чином.

— КК-1 — Виявлення прихованих каналів.

— КК-2 — Контроль прихованих каналів.

— КК-3 — Перекриття прихованих каналів.

В усіх цих трьох випадках повинен бути виконаний аналіз прихованих каналів.

Послуга конфіденційності при обміні дозволяє забезпечити захист об'єктів від несанкціонованого ознайомлення з інформацією під час їх експорту/імпорту через незахищене середовище. Рівні даної послуги розподіляються на підставі повноти захисту і вибірковості керування наступним чином.

— КВ-1 — Мінімальна конфіденційність при обміні повинна забезпечити захист від безпосереднього ознайомлення з інформацією для визначеної множини об’єктів та інтерфейсних процесів, крім цього повинен бути визначений рівень захищеності і надана можливість керувати цим рівнем.

— КВ-2 — Базова конфіденційність при обміні крім того, що забезпечено на рівні КВ-1, повинна надавати дозвіл на керування рівнем захищеності на підставі атрибутів доступу інтерфейсного процесу.

— КВ-3 — Повна конфіденційність при обміні повинна забезпечувати захист всіх об’єктів і інтерфейсних процесів даної АС і враховувати атрибути інтерфейсного процесу, захищаємого об’єкта, джерела і приймальника інформації.

— КВ-4 — Абсолютна конфіденційність при обміні крім того, що забезпечено на попередніх рівнях, повинна включати опис інформації, яку можливо отримати шляхом сумісного аналізу ряду одержаних об’єктів та повинна бути забезпечена реєстрація використання затвердженої підмножини знайдених прихованих каналів, їх часткове перекриття або усунення.

2.2.2. Критерії цілісності

Цілісність забезпечується послугами довірчої та адміністративної цілісності, відкатом і цілісністю при обміні.

Послуга довірчої цілісності надається у системах з довірчим керуванням доступом, у яких користувачам дозволено керувати потоками інформації між іншими користувачами і об’єктами без втручання адміністратора. Рівні даної послуги розподіляються на підставі повноти захисту і вибірковості керування наступним чином.

— ЦД-1 — Мінімальна довірча цілісність при якій розмежування доступу здійснюється на підставі атрибутів користувачів і захищених об’єктів. Довіреному користувачу надається можливість для кожного об’єкта, що належать його домену, визначати користувачів, які мають право модифікувати об’єкт. Права доступу до кожного захищеного об'єкта повинні встановлюватися в момент його створення або ініціалізації. Як частина політики довірчої цілісності мають бути представлені правила збереження атрибутів доступу об'єктів під час їх експорту і імпорту.

— ЦД-2 — Базова довірча цілісність при якій розмежування доступу здійснюється на підставі атрибутів процесів і захищених об’єктів. Довіреному користувачу надається можливість для кожного об’єкта, що належать його домену, визначати конкретні процеси і/або групи процесів, які мають право модифікувати об'єкт, а також визначати конкретних користувачів і/або групи користувачів, які мають право ініціювати процес.

— ЦД-3 — Повна довірча цілісність при якій захисту підлягають усі об’єкти КС. Довіреному користувачу надається можливість для кожного об’єкта, що належать його домену, визначати конкретні процеси і групи процесів, які мають, а також ті, що не мають право модифікувати об'єкт, а також визначати конкретних користувачів і групи користувачів, які мають, а також ті, що не мають право ініціювати процес.

— ЦД-4 — Абсолютна довірча цілісність при якій розмежування доступу здійснюється на підставі атрибутів процесу, користувача і захищеного об'єкта. При цьому захисту підлягають усі об’єкти КС. Довіреному користувачу надається можливість для кожного об’єкта, що належать його домену, визначати конкретних користувачів і процеси (і групи користувачів і процесів), які мають, а також тих, що не мають права модифікувати об'єкт та ініціювати процес.

Послуга адміністративної цілісності дозволяє адміністратору або спеціально авторизованому користувачу керувати потоками інформації від користувачів до захищених об'єктів. Рівні даної послуги розподіляються на підставі повноти захисту і вибірковості керування наступним чином.

— ЦА-1 — Мінімальна адміністративна цілісність при якій розмежування доступу здійснюється на підставі атрибутів користувачів і захищених об’єктів. Адміністратору надається можливість для кожного об’єкта визначати користувачів, які мають право модифікувати об’єкт. Права доступу до кожного захищеного об'єкта повинні встановлюватися в момент його створення або ініціалізації. Як частина політики довірчої цілісності мають бути представлені правила збереження атрибутів доступу об'єктів під час їх експорту і імпорту.

— ЦА-2 — Базова адміністративна цілісність при якій розмежування доступу здійснюється на підставі атрибутів процесів і захищених об’єктів. Адміністратору надається можливість для кожного захищеного об’єкта визначати конкретні процеси і/або групи процесів, які мають право модифікувати об'єкт, а також визначати конкретних користувачів і/або групи користувачів, які мають право ініціювати процес.

— ЦА-3 — Повна адміністративна цілісність при якій захисту підлягають усі об’єкти КС. Адміністратору надається можливість для кожного захищеного об’єкта визначати конкретні процеси і групи процесів, які мають, а також ті, що не мають право модифікувати об'єкт, а також визначати конкретних користувачів і групи користувачів, які мають, а також ті, що не мають право ініціювати процес.

— ЦА-4 — Абсолютна адміністративна цілісність при якій розмежування доступу здійснюється на підставі атрибутів процесу, користувача і захищеного об'єкта. При цьому захисту підлягають усі об’єкти КС. Адміністратору надається можливість для кожного об’єкта визначати конкретних користувачів і процеси (і групи користувачів і процесів), які мають, а також тих, що не мають права модифікувати об'єкт та ініціювати процес.

Послуга відкат забезпечує можливість відмінити операцію або послідовність операцій і повернути (відкатити) захищений об'єкт до попереднього стану. Рівні даної послуги розподіляються на підставі множини операцій, для яких забезпечується відкат, наступним чином.

— ЦО-1 — Обмежений відкат при якому надається можливість користувачу або процесу відкатити або відмінити певний набір (множину) операцій, виконаних над захищеним об'єктом за певний проміжок часу.

— ЦО-2 — Повний відкат при якому надається можливість користувачу або процесу відкатити або відмінити всі операції, що виконані над захищеним об'єктом за певний проміжок часу.

Послуга цілісність при обміні дозволяє забезпечити захист об'єктів від несанкціонованої модифікації інформації, що міститься в них, під час їх експорту/імпорту через незахищене середовище. Рівні даної послуги розподіляються на підставі повноти захисту і вибірковості керування наступним чином.

— ЦВ-1 — Мінімальна цілісність при обміні повинна забезпечувати можливість виявлення порушення цілісності інформації, що міститься в об'єкті, який передається а також фактів його видалення або дублювання.

— ЦВ-2 — Базова цілісність при обміні повинна забезпечувати обробку запитів на експорт та імпорт захищених об'єктів на підставі атрибутів доступу інтерфейсного процесу при цьому запити на присвоєння або зміну рівня захищеності повинні обробляються тільки в тому випадку, якщо вони надходять від адміністраторів або користувачів, яким надані відповідні повноваження.

— ЦВ-3 — Повна цілісність при обміні повинна забезпечувати обробку запитів на експорт та імпорт захищених об'єктів на підставі атрибутів доступу інтерфейсних процесів як з боку відправника, так і з боку приймальника об’єкта. Представлення захищеного об'єкта має бути функцією атрибутів доступу інтерфейсного процесу, самого об'єкта, а також його джерела і приймальника.

2.2.3. Критерії доступності

Доступність забезпечується послугами використання ресурсів, стійкістю до відмов, гарячої заміна та відновленням після збоїв.

Послуга використання ресурсів дозволяє користувачам керувати використанням ресурсів. Запити на встановлення та зміну обмежень обробляються тільки в тому випадку, якщо вони надходять від адміністраторів або від користувачів, яким надані відповідні повноваження. Рівні даної послуги розподіляються на підставі повноти захисту і вибірковості керування доступністю наступним чином.

— ДР-1 — Квоти для доступу до визначеної множини об’єктів. При цьому визначаються обмеження, які можна накладати на кількість об’єктів або обсяг ресурсів, що виділяються окремому користувачу.

— ДР-2 — Недопущення захоплення ресурсів відносно усіх об’єктів системи. При цьому повинна існувати можливість встановлювати обмеження таким чином, щоб запобігати діям, які можуть призвести до неможливості доступу інших користувачів до функцій КЗЗ або захищених об'єктів. КЗЗ повинен контролювати такі дії, здійснювані з боку окремого користувача.

— ДР-3 — Пріоритетність використання ресурсів відносно усіх об’єктів системи. При цьому визначаються обмеження, які можна накладати на кількість об’єктів або обсяг ресурсів, що виділяються окремому користувачу і довільним групам користувачів, а також повинна існувати можливість встановлювати обмеження таким чином, щоб запобігати діям, які можуть призвести до неможливості доступу інших користувачів до функцій КЗЗ або захищених об'єктів. КЗЗ повинен контролювати такі дії, здійснювані з боку окремого користувача і довільних груп користувачів.

Послуга стійкість до відмов гарантує доступність об'єктів після відмови компонента КС. Повинні бути чітко вказані рівні відмов, при перевищенні яких відмови призводять до зниження характеристик обслуговування або недоступності послуги. КЗЗ повинен бути спроможний повідомити адміністратора про відмову будь-якого захищеного компонента. Рівні даної послуги розподіляються на підставі спроможності КЗЗ забезпечити можливість функціонування КС в залежності від кількості відмов і послуг, доступних після відмови, наступним чином.

— ДС-1 — Стійкість при обмежених відмовах полягає у визначенні множини компонентів КС, що мають можливість відмов певного типу при яких КС в змозі продовжувати функціонування. Відмова одного захищеного компонента не повинна призводити до недоступності всіх послуг, а має в гіршому випадку проявлятися в зниженні характеристик обслуговування.

— ДС-2 — Стійкість з погіршенням характеристик обслуговування повинна відноситись до всіх компонентів КС, забезпечуючи ті самі властивості, що у випадку ДС-1.

— ДС-3 — Стійкість без погіршення характеристик обслуговування полягає в тому, що відмова одного захищеного компонента не повинна призводити до недоступності послуг або до зниження характеристик обслуговування.

Послуга гаряча заміна дозволяє гарантувати доступність КС (можливість використання інформації, окремих функцій або КС в цілому) в процесі заміни окремих компонентів. Рівні даної послуги розподіляються на підставі повноти реалізації наступним чином.

— ДЗ-1 — Модернізація полягає у тому, щоб адміністратор або користувач, якому надано відповідні повноваження, мав можливість провести модернізацію КС при умові, що це не призводить до необхідності ще раз проводити інсталяцію КС або до переривання виконання КЗЗ функцій захисту.

— ДЗ-2 — Обмежена гаряча заміна передбачає визначену множину компонентів КС, які можуть бути замінені без переривання обслуговування. При цьому адміністратор або користувач, якому надані відповідні повноваження, повинні мати можливість замінити будь-який з цих компонентів.

— ДЗ-3 — Гаряча заміна будь-якого компонента передбачає можливість заміни будь-якого компонента КС без переривання обслуговування.

Послуга відновлення після збоїв забезпечує повернення КС у відомий захищений стан після відмови або переривання обслуговування. При цьому повинна бути визначена множина типів відмов КС і переривань обслуговування,

після яких можливе повернення у відомий захищений стан без порушення політики безпеки. Також повинні бути чітко вказані рівні відмов, у разі перевищення яких необхідна повторна інсталяція КС. Рівні даної послуги розподіляються на підставі міри автоматизації процесу відновлення компонентів наступним чином.

— ДВ-1 — Ручне відновлення при якому після відмови КС або переривання обслуговування КЗЗ повинен перевести КС до стану, із якого повернути її до нормального функціонування може тільки адміністратор або користувач, якому надані відповідні повноваження.

— ДВ-2 — Автоматизоване відновлення при якому після відмови КС або переривання обслуговування КЗЗ повинен визначити чи можуть бути використані автоматизовані процедури для повернення КС до нормального функціонування безпечним чином. Якщо такі процедури можуть бути використані, то КЗЗ має бути здатним виконати їх і повернути КС до нормального функціонування.

— ДВ-3 — Вибіркове відновлення при якому після відмови КС або переривання обслуговування, що не призводить до необхідності нової інсталяції КС, КЗЗ повинен бути здатним виконати необхідні процедури і безпечним чином повернути КС до нормального функціонування або, в гіршому випадку, функціонування в режимі з погіршеними характеристиками обслуговування. Повинні існувати ручні процедури, за допомогою яких можна безпечним чином повернути КС з режиму з погіршеними характеристиками обслуговування в режим нормального функціонування.

2.2.4. Критерії спостереженості

Захист даних неможливий без спостереження за процесами, які відбуваються в комп’ютері. Послуги спостереженості є необхідним доповненням до послуг конфіденційності, цілісності та доступності, що були описані вище, як це показано у таблицях 2.1, 2.2 та 2.3.

Таблиця 2.1.

Необхідні умови доповнення послуг конфіденційності

Код послуг конфіденційності	Код послуг, що необхідні для доповнення
КД-1, КД-2	НЦ-1, НИ-1
КД-3, КД-4	НЦ-1, КО-1, НИ-1
КА-1, КА-2	НЦ-1, НО-1, НИ-1
КА-3, КА-4	НЦ-1, КО-1, НО-1, НИ-1

КК-1, КК-3	НЦ-1, КО-1, Г-3
КК-2	НЦ-1, КО-1, НР-1, Г-3
КВ-2	НЦ-1, НО-1
КВ-3	НЦ-1, НО-1, НВ-1
КВ-4	НЦ-1, НО-1, НВ-1, НР-1, Г-3

Таблиця 2.2.

Необхідні умови доповнення послуг цілісності

Код послуг цілісності	Код послуг, що необхідні для доповнення
ЦД-1, ЦД-2	НЦ-1, НИ-1
ЦД-3, ЦД-4	НЦ-1, КО-1, НИ-1
ЦА-1, ЦА-2	НЦ-1, НО-1, НИ-1
ЦА-3, ЦА-4	НЦ-1, КО-1, НО-1, НИ-1
ЦО-1, ЦО-2	НЦ-1, НИ-1
ЦВ-2	НЦ-1, НО-1
ЦВ-3	НЦ-1, НО-1, НВ-1

Таблиця 2.3.

Необхідні умови доповнення послуг доступності

Код послуг доступності	Код послуг, що необхідні для доповнення
ДР-1, ДР-2, ДР-3	НЦ-1, НО-1
ДС-1, ДС-2, ДС-3	НЦ-1, НО-1
ДЗ-1	НЦ-1, НО-1
ДЗ-2, ДЗ-3	НЦ-1, НО-1, ДС-1
ДВ-1, ДВ-2, ДВ-3	НЦ-1, НО-1

Спостереженість забезпечується в КС такими послугами: реєстрація (аудит), ідентифікація і автентифікація, достовірний канал, розподіл обов'язків, цілісність КЗЗ, самотестування, ідентифікація і автентифікація при обміні, автентифікація відправника, автентифікація одержувача.

Послуга реєстрація дозволяє контролювати небезпечні для КС дії. Рівні даної послуги розподіляються залежно від повноти і вибірковості контролю, складності засобів аналізу даних журналів реєстрації і спроможності вияву потенційних порушень наступним чином.

— НР-1 — Зовнішній аналіз при якому визначають перелік подій, що реєструються. Ці події повинні мати безпосереднє відношення до забезпечення безпеки інформації. Журнал реєстрації повинен містити інформацію про дату, час, місце, тип і успішність чи неуспішність кожної зареєстрованої події, що

необхідно для ідентифікації користувача, процесу і/або об'єкта, які мали відношення до кожної зареєстрованої події. Цей журнал повинен передаватись в інші системи з використанням механізмів захисту.

— НР-2 — Захищений журнал при якому забезпечується захист журналу реєстрації від несанкціонованого доступу, модифікації або руйнування. Адміністратори і користувачі, яким надані відповідні повноваження, повинні мати в своєму розпорядженні засоби перегляду і аналізу журналу реєстрації.

— НР-3 — Сигналізація про небезпеку при якій КЗЗ має бути здатним контролювати одиничні або повторювані реєстраційні події, які можуть свідчити про прямі (істотні) порушення політики безпеки КС. КЗЗ має бути здатним негайно інформувати адміністратора про перевищення порогів безпеки і, якщо реєстраційні небезпечні події повторюються, здійснити неруйнівні дії щодо припинення повторення цих подій.

— НР-4 — Детальна реєстрація при якій КЗЗ має бути здатним здійснювати реєстрацію подій, що мають безпосереднє або непряме відношення до безпеки.

— НР-5 — Аналіз в реальному часі при якому КЗЗ повинен виявляти і аналізувати несанкціоновані дії в реальному часі.

Послуга ідентифікації і автентифікації дозволяє визначити і перевірити особистість користувача, що намагається одержати доступ до КС. Рівні даної послуги розподіляються залежно від числа задіяних механізмів автентифікації наступним чином.

— НИ-1 — Зовнішня ідентифікація і автентифікація при якій кожен користувач повинен однозначно ідентифікуватись. При цьому визначаються атрибути, якими характеризується користувач, і послуги, для виконання яких необхідні ці атрибути. Дозвіл на виконання користувачем контрольованих дій надається після одержання від зовнішнього джерела автентифікованого ідентифікатора цього користувача.

— НИ-2 — Одиночна ідентифікація і автентифікація при якій перш ніж дозволити користувачу виконувати контрольовані дії, КЗЗ повинен автентифікувати цього користувача з використанням захищеного механізму та забезпечувати захист даних автентифікації від несанкціонованого доступу, модифікації або руйнування.

— НИ-3 — Множинна ідентифікація і автентифікація при якій перш ніж дозволити користувачу виконувати контрольовані дії, КЗЗ повинен автентифікувати цього користувача з використанням двох або більше типів захищених механізмів.

Послуга достовірний канал дозволяє гарантувати користувачу можливість безпосередньої взаємодії з КЗЗ. Рівні даної послуги розподіляються залежно від гнучкості надання можливості КЗЗ або користувачу ініціювати захищений обмін наступним чином.

— НК-1 — Однонаправлений достовірний канал при якому існують визначені механізми забезпечення достовірного зв’язку між користувачем і КЗЗ. Цей канал повинен використовуватись для початкової ідентифікації і автентифікації. Зв'язок з використанням даного каналу повинен ініціюватися виключно користувачем.

— НК-2 — Двонаправлений достовірний канал використовується для початкової ідентифікації і автентифікації та у випадках, коли необхідний прямий зв'язок користувач/КЗЗ або КЗЗ/користувач. Зв'язок з використанням даного каналу повинен ініціюватися користувачем або КЗЗ. Обмін з використанням достовірного каналу, що ініціює КЗЗ, повинен бути однозначно ідентифікований як такий і має відбутися тільки після позитивного підтвердження готовності до обміну з боку користувача.

Послуга розподіл обов’язків дозволяє зменшити потенційні збитки від навмисних або помилкових дій користувача і обмежити авторитарність керування. Рівні даної послуги розподіляються на підставі вибірковості керування можливостями користувачів і адміністраторів гарантувати користувачу можливість безпосередньої взаємодії з КЗЗ наступним чином.

— НО-1 — Виділення адміністратора полягає у визначенні ролі адміністратора і звичайного користувача, а також у визначенні притаманних їм функцій.

— НО-2 — Розподіл обов’язків адміністраторів полягає у визначенні як мінімум двох адміністраторів: адміністратора безпеки та іншого адміністратора. Функції, притаманні кожному з адміністраторів, повинні бути мінімізовані так, щоб включати тільки ті функції, які необхідні для виконання своєї ролі.

— НО-3 — Розподіл обов’язків на підставі привілеїв полягає у визначенні множини ролей користувачів.

Послуга цілісність комплексу засобів захисту визначає міру здатності КЗЗ захищати себе і гарантувати свою спроможність керувати захищеними об'єктами. Рівні даної послуги розподіляються наступним чином.

— НЦ-1 — КЗЗ з контролем цілісності при якому визначаються склад КЗЗ і механізми контролю цілісності компонентів, що входять до складу КЗЗ. В разі виявлення порушення цілісності будь-якого із своїх компонентів КЗЗ повинен повідомити адміністратора і/або автоматично відновити відповідність

компонента еталону або перевести КС до стану, з якого повернути її до нормального функціонування може тільки адміністратор або користувачі, яким надані відповідні повноваження. Повинні бути описані обмеження, дотримання яких дозволяє гарантувати, що послуги безпеки доступні тільки через інтерфейс КЗЗ і всі запити на доступ до захищених об'єктів контролюються КЗЗ.

— НЦ-2 — КЗЗ з гарантованою цілісністю при якому повинен бути визначеним домен КЗЗ та інші домени, а також механізми захисту, що використовуються для реалізації розподілення доменів. КЗЗ повинен підтримувати домен для свого власного виконання з метою захисту від зовнішніх впливів і несанкціонованої модифікації і/або втрати керування.

— НЦ-3 — КЗЗ з функціями диспетчера доступу при якому повинні надаватись гарантії, що послуги безпеки доступні тільки через інтерфейс КЗЗ і всі запити на доступ до захищених об'єктів контролюються КЗЗ.

Послуга самотестування дозволяє КЗЗ перевірити і на підставі цього гарантувати правильність функціонування і цілісність певної множини функцій КС. Рівні даної послуги розподіляються на підставі можливості виконання тестів у процесі запуску або штатної роботи наступним чином.

— НТ-1 — Самотестування за запитом при якому повинні бути описані властивості КС і реалізовані процедури, які можуть бути використані для оцінки правильності функціонування КЗЗ. Тести повинні виконуватися за запитом користувача, що має відповідні повноваження.

— НТ-2 — Самотестування при старті при якому тести повинні виконуватись при ініціалізації КЗЗ.

— НТ-3 — Самотестування в реальному часі при якому тести повинні виконуватись при ініціалізації КЗЗ і в процесі штатного функціонування.

Послуга ідентифікації і автентифікації при обміні дозволяє одному КЗЗ ідентифікувати інший КЗЗ (встановити і перевірити його ідентичність) і забезпечити іншому КЗЗ можливість ідентифікувати перший, перш ніж почати взаємодію. Рівні даної послуги розподіляються на підставі повноти реалізації наступним чином.

— НВ-1 — Автентифікація вузла при якій повинна бути визначена множина атрибутів КЗЗ і процедури, що необхідні для взаємної ідентифікації при ініціалізації обміну даними з іншим КЗЗ. Ідентифікація і автентифікація КЗЗ при цьому повинна відбуватись з використанням захищеного механізму на підставі затвердженого протоколу.

— НВ-2 — Автентифікація джерела даних при якій КЗЗ повинен використовувати захищені механізми для встановлення джерела кожного об'єкта, що експортується та імпортується.

— НВ-3 — Автентифікація з підтвердженням при якій протокол, що використовується для автентифікації, повинен забезпечувати можливість однозначного підтвердження джерела об'єкта незалежною третьою стороною.

Послуга автентифікації відправника дозволяє забезпечити захист від відмови від авторства і однозначно встановити належність певного об'єкта певному користувачу, тобто той факт, що об'єкт був створений або відправлений даним користувачем. Рівні даної послуги розподіляються на підставі можливості підтвердження результатів перевірки незалежною третьою стороною наступним чином.

— НА-1 — Базова автентифікація відправника при якій повинна бути визначена множина властивостей і атрибутів об'єкта, що передається, користувача-відправника і інтерфейсного процесу, а також процедури, які дозволяли б однозначно встановити, що даний об'єкт був відправлений (створений) певним користувачем. Встановлення належності має виконуватися на підставі затвердженого протоколу автентифікації.

— НА-2 — Автентифікація відправника з підтвердженням при якій додатково повинні бути визначені ті властивості, атрибути і процедури, які можуть використовуватися для однозначного підтвердження належності об'єкта незалежною третьою стороною. Використовуваний протокол автентифікації повинен забезпечувати можливість однозначного підтвердження належності об'єкта незалежною третьою стороною.

Послуга автентифікації отримувача дозволяє забезпечити захист від відмови від одержання і дозволяє однозначно встановити факт одержання певного об'єкта певним користувачем. Рівні даної послуги розподіляються на підставі можливості підтвердження результатів перевірки незалежною третьою стороною наступним чином.

— НП-1 — Базова автентифікація отримувача при якій повинна бути визначена множина властивостей і атрибутів об'єкта множина властивостей і атрибутів об'єкта, що передається, користувача-одержувача і інтерфейсного процесу, а також процедури, які дозволяли б однозначно встановити, що даний об'єкт був одержаний певним користувачем. Ідентифікація одержувача повинна відбуватись на підставі затвердженого протоколу.

— НП-2 — Автентифікація отримувача з підтвердженням при якій додатково повинні бути визначені ті властивості, атрибути і процедури, які

можуть використовуватися для однозначного підтвердження факту одержання об'єкта незалежною третьою стороною. Використовуваний протокол автентифікації повинен забезпечувати можливість однозначного підтвердження незалежною третьою стороною факту одержання об'єкта.

Необхідні доповнення послуг спостереженості іншими послугами показані у таблиці 2.4.

Таблиця 2.4.

Необхідні умови доповнення послуг спостереженості

Код послуг спостереженості	Код послуг, що необхідні для доповнення
НР-1	НЦ-1, НИ-1
НР-2, НР-3, НР-4, НР-5	НЦ-1, НИ-1, НО-1
НИ-2, НИ-3	НЦ-1, НК-1
НО-1, НО-2, НО-3	НЦ-1, НИ-1
НЦ-1	НР-1, НО-1
НТ-1, НТ-2, НТ-3	НЦ-1, НО-1
НА-1, НА-2	НЦ-1, НИ-1
НП-1, НП-2	НЦ-1, НИ-1

2.2.5. Критерії гарантій

Критерії гарантій включають вимоги до архітектури КЗЗ, середовища розробки, послідовності розробки (проектування), середовища функціонування, документації і випробувань КЗЗ. В цих критеріях вводиться сім рівнів гарантій, які є ієрархічними від найнижчого рівня Г-1 до найвищого Г-7. Для того, щоб система одержала певний рівень гарантій повинні бути задоволені всі вимоги, визначені для даного рівня.

Вимоги до архітектури забезпечують гарантії того, що КЗЗ може повністю реалізувати політику безпеки. Для всіх рівнів гарантій необхідно щоб компоненти КЗЗ були чітко визначені. Починаючи від рівня Г-3 необхідно, щоб ці компоненти були незалежними і спроектовані відповідно до принципу мінімуму повноважень. Для вищих рівнів, починаючи від Г-4, додається вимога захисту критичних компонентів КЗЗ від некритичних за рахунок використання спеціальних механізмів захисту. Для рівнів Г-5, Г-6 та Г-7 необхідно, щоб критичні для забезпечення безпеки компоненти КЗЗ були відокремлені від некритичних. При цьому КЗЗ має бути спроектований і структурований так, щоб використовувати повний і концептуально простий механізм захисту. Цей

механізм повинен бути спроектований як окремий модуль та відігравати центральну роль в реалізації внутрішньої структури КЗЗ.

Вимоги до середовища розробки повинні забезпечують гарантії того, що процеси розробки і супроводження КС повністю керовані з боку розробника. Розробник повинен визначити всі стадії життєвого циклу КС, розробити, запровадити і підтримувати в робочому стані документально оформлені методики своєї діяльності на кожній стадії. Мають бути документовані всі етапи кожної стадії життєвого циклу КС. Система керування конфігурацією КС повинна забезпечувати керування внесенням змін в апаратне та програмне забезпечення і гарантувати постійну відповідність між документацією і реалізацією поточної версії.

Починаючи від рівня Г-3 розробник повинен описати стандарти кодування, яких необхідно додержуватися в процесі реалізації, і повинен гарантувати, що всі вихідні коди компілюються відповідно до цих стандартів. Будь-яка з використовуваних під час реалізації мов програмування має бути добре визначена. Всі залежні від реалізації параметри мов програмування або компіляторів повинні бути документовані.

Починаючи від рівня Г-4 розробник повинен розробити, запровадити і підтримувати в робочому стані документально оформлені методики забезпечення фізичної, технічної, організаційної і кадрової безпеки. При цьому система керування конфігурацією використовується для генерації КЗЗ з вихідного коду і обліку всіх змін з появою нових версій. Ця система також повинна видавати звіти про поточний стан елементів конфігурації.

Для рівнів Г-6 та Г-7 необхідно створювати спеціальну систему заходів технічної, фізичної, організаційної і кадрової безпеки, спрямованих на захист усіх засобів і матеріалів, використовуваних для генерації КЗЗ, від несанкціонованої модифікації або руйнування.

Вимоги до процесу проектування повинні забезпечувати гарантії того, що на кожній стадії розробки реалізація КС точно відповідає вимогам політики безпеки.

Для всіх рівнів гарантії розробник повинен на стадії розробки технічного завдання надати опис функціональних специфікацій КС та опис політики безпеки, що реалізується КЗЗ. Функціональні специфікації повинні включати модель політики безпеки та в залежності від рівня гарантій можуть бути неформалізовані (для рівня Г-2), частково формалізовані (для рівня Г-3) або повністю формалізовані (для рівнів Г-4 та вищих).

На стадії ескізного проектування КС для всіх рівнів гарантій необхідно розробити проект архітектури КЗЗ в якому повинні бути описані всі послуги безпеки. Зовнішні інтерфейси КЗЗ повинні бути описані в термінах повідомлень про помилки і кодів повернення. При цьому стиль опису в залежності від рівня гарантій може бути неформалізованим (для рівнів Г-1 та Г-2), частково формалізованим (для рівнів Г-3, Г-4 та Г-5) або повністю формалізовані (для рівнів Г-6 та Г-7).

На стадіях розробки технічного та робочого проекту повинен бути представлений детальний перелік всіх компонентів КЗЗ і точний опис функціонування кожного механізму. Повинні бути описані призначення і параметри інтерфейсів для кожного компонента КЗЗ. При цьому стиль опису в залежності від рівня гарантій може бути неформалізованим (для рівнів Г-1, Г-2 та Г-3), частково формалізованим (для рівнів Г-4, Г-5 та Г-6) або повністю формалізовані (для рівня Г-7).

Вимоги до середовища функціонування забезпечують гарантії того, що КС поставляється без несанкціонованих модифікацій. Розробник повинен представити засоби інсталяції, генерації і запуску КС, які гарантують, що експлуатація КС починається з безпечного стану. Розробник повинен представити перелік усіх можливих параметрів конфігурації, які можуть використовуватися в процесі інсталяції, генерації і запуску. Для рівнів гарантій вище за Г-2 повинна існувати система технічних, організаційних і фізичних заходів безпеки, яка гарантує, що програмне і апаратне забезпечення КЗЗ точно відповідає еталонній копії, а для рівнів гарантій вище за Г-5 повинна існувати система керування розповсюдженням захищеної КС.

Вимоги до документації є загальними для всіх рівнів гарантій. Розробник повинен подати опис послуг безпеки, що реалізуються КЗЗ, настанови адміністратору щодо послуг безпеки, настанови користувача щодо послуг безпеки. В описі функцій безпеки повинні бути викладені основні, необхідні для правильного використання послуг безпеки, принципи політики безпеки, що реалізується КЗЗ. Настанови адміністратору щодо послуг безпеки мають містити опис засобів інсталяції, генерації і запуску КС, опис всіх можливих параметрів конфігурації, які можуть використовуватися в процесі інсталяції, генерації і запуску КС, опис властивостей КС, які можуть бути використані для періодичної оцінки правильності функціонування КЗЗ, а також інструкції щодо використання адміністратором послуг безпеки для підтримки політики безпеки, прийнятої в організації, що експлуатує КС.

Для всіх рівнів гарантії розробник повинен надати для перевірки програму і методику випробувань усіх механізмів, що реалізують послуги безпеки. Мають бути представлені аргументи для підтвердження достатності тестового покриття. Необхідно також надати детальний перелік усіх можливих результатів тестування. Для рівнів Г-4 та вищих розробник повинен виконати тести з подолання механізмів захисту і довести, що КЗЗ відносно або абсолютно стійкий до такого роду атак.

2.3. Дискреційні та мандатні механізми керування доступом

Основою захисту інформації в комп'ютерних системах є механізми керування доступом, що розв'язують задачу розмежування доступу до критичних об'єктів для суб'єктів з різними повноваженнями. У системі обов'язково повинен бути хоч один суб'єкт з правами адміністратора, який може керувати комплексом засобів захисту. Згідно нормативним документам для розмежування доступу до конфіденційної інформації слід використовувати дискреційні механізми керування доступом, а для таємної інформації – мандатні механізми керування доступом [9].

Дискреційні механізми керування доступом створюють на основі матриць доступу, які можуть бути двовимірними або тривимірними.

Позначимо множину суб'єктів доступу $U = \{ U_1, \ldots, U_k \}$, а множину критичних об'єктів $K = \{ K_1, \ldots, K_r \}$, при цьому двовимірна матриця доступу має наступний вигляд:

$$\boldsymbol{D} = \begin{array}{c|cccccc} & U_1 & U_2 & \ldots & U_{k-1} & U_k \\ K_1 & D_{11} & D_{12} & \ldots & D_{1(k-1)} & D_{1k} \\ K_2 & D_{21} & D_{22} & \ldots & D_{2(k-1)} & D_{2k} \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ K_{r-1} & D_{(r-1)1} & D_{(r-1)2} & \ldots & D_{(r-1)(k-1)} & D_{2k} \\ K_r & D_{r1} & D_{r2} & \ldots & D_{r(k-1)} & D_{rk} \end{array}, \qquad (2.1)$$

де елемент матриці $D_{ij} = 1$ у разі коли суб'єкт U_j має право доступу до об'єкта K_i або $D_{ij} = 0$ у разі коли суб'єкту U_j доступу до об'єкта K_i заборонено.

Для забезпечення розмежування доступу до критичних об'єктів у повному обсязі необхідно, щоб КЗЗ забезпечував можливість реалізації канонічної форми матриці (2.1). Ця форма відповідає випадку коли кожен користувач має власний критичний об'єкт і хоче захистити його від доступу інших користувачів. У такому випадку при умові, що об'єкт K_i належить користувачу U_j, матриця приймає наступний вигляд:

$$\boldsymbol{D} = \begin{array}{c|cccccc} & U_1 & U_2 & \ldots & U_{k-1} & U_k \\ K_1 & 1 & 0 & \ldots & 0 & 0 \\ K_2 & 0 & 1 & \ldots & 0 & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ K_{r-1} & 0 & 0 & \ldots & 1 & 0 \\ K_r & 0 & 0 & \ldots & 0 & 1 \end{array}, \qquad (2.2)$$

Зрозуміло, що при цьому $r = k$.

У разі коли множина прав доступу вміщує більше ніж два елементи може бути побудована тривимірна матриця доступу, де третім виміром є права доступу. Позначимо множину цих прав $R = \{ R_1, \ldots, R_n \}$. При цьому матриця буде мати вигляд, що показаний на рис. 2.3.

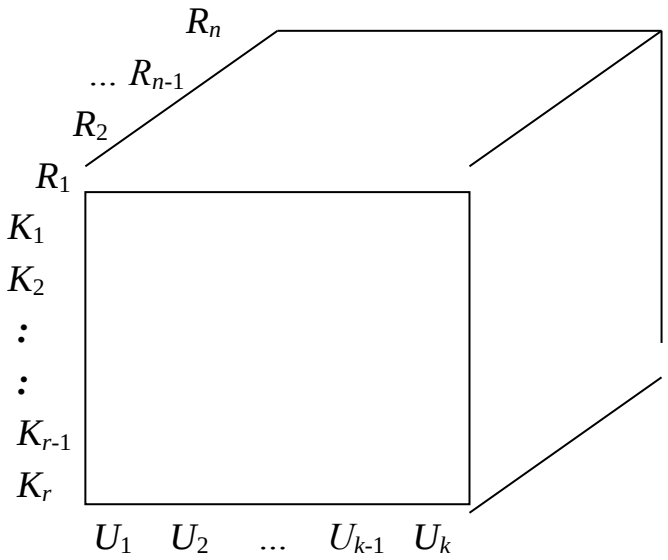

Рис. 2.3. Тривимірна матриця доступу.

Варіанти прав доступу являють собою перелік дій, що дозволені користувачу виконувати з критичним об’єктом. Серед таких дій можуть бути ознайомлення, модифікація, запуск на виконання, доповнення файлів баз даних без дозволу на ознайомлення зі змістом усієї бази.

Недоліком дискреційних (матричних) механізмів є складність підтримки в актуальному стані матриць в умовах великої кількості користувачів та об’єктів. Через це такі механізми використовують у системах з невеликою кількістю користувачів або в умовах необхідності максимальної гнучкості керування. За допомогою дискреційного механізму можна побудувати яку завгодно модель керування доступом.

Мандатні механізми обмежені у можливостях керування доступом, але набагато простіші у підтримці. Ці механізми побудовані на основі ієрархічних

міток безпеки. При цьому кожному користувачу та кожному критичному об’єкту призначається мітка, що відображає його місце у ієрархії захисту даних.

Розглянемо для прикладу систему з трьома наступними ієрархічними рівнями конфіденційності:

- цілком таємно (мітка 3);
- таємно (мітка 2);
- для службового користування (мітка 1).

Якщо користувачу надано право доступу до об’єктів для службового користування і заборонено доступ до об’єктів вищого рівня, то йому слід призначити мітку 1. Користувач, якому призначено мітку 2, буде мати доступ до таємних об’єктів та до об’єктів для службового користування. Користувачі, яким призначено мітку 3, будуть мати повний доступ до всіх критичних об’єктів.

Для розгалуження прав на ознайомлення і на модифікацію об’єктів можуть бути використані додаткові рівні прав доступу користувачів.

Наприклад, у разі коли рівень доступу користувача дорівнює рівню конфіденційності об’єкта, можна дозволяти тільки ознайомлення, а для одержання права на модифікацію об’єкта необхідно, щоб рівень доступу користувача перевищував рівень конфіденційності об’єкта. При цьому для користувачів треба ввести додатковий рівень доступу з міткою 4, бо мітка 3 не буде надавати прав на модифікацію цілком таємних об’єктів і тільки мітка 4 надасть можливість повного доступу, що необхідно для підтримки актуальності даних найвищого рівня конфіденційності.

Можливе сумісне використання у АС дискреційних та мандатних механізмів керування доступом. При цьому користувач буде отримувати право доступу до об’єкта тільки у разі коли дозвіл надано як з боку дискреційного так і з боку мандатного механізмів. Такі комбіновані системи керування доступом надають змогу у деякій мірі поєднувати переваги як мандатних так і дискреційних механізмів.

Висновки

1. Дані, що підлягають захисту є пасивними об’єктами комп’ютерної системи (КС). Найчастіше вини являють собою файли на жорстких дисках. Підмножину цих об’єктів, що потребують захисту, називають критичними об’єктами.
2. Об’єкти, що можуть бути загрозою для безпеки даних, являють собою комп’ютерні програми, що знаходяться в оперативній пам’яті у стадії виконання. Ці об’єкти є активними, їх прийнято називати процесами.

3. Існує можливість завдання шкоди власникам інформації через неправомочні або безвідповідальні дії користувачів. Через такі дії користувачі можуть стати порушниками безпеки. Щоб уникнути загроз від порушників безпеки використовують метод розмежування доступу користувачів до інформаційних ресурсів комп'ютерної системи. Це розмежування робиться на основі політики безпеки, яку приймають власники інформації.
4. Реалізація політики безпеки інформації покладається на комплекс засобів захисту, який контролює та обмежує потоки інформації, що утворюються між об'єктами комп'ютерної системи. При цьому повинні бути ліквідовані усі ті можливості завдання шкоди інформаційним ресурсам, що передбачені у моделі загроз. У цій моделі повинні бути описані можливі шляхи здійснення загрози. Також слід передбачити можливість проникнення комп'ютерних вірусів. У разі особливо цінної інформації може бути розглянуто застосування закладних пристроїв чи програм.
5. Крім моделювання загроз створюють абстрактний формалізований або неформалізований опис можливого порушника. Цей опис прийнято називати моделлю порушника. У якості порушника розглядається особа, яка має доступ до роботи з КС. Порушники класифікуються за рівнем можливостей, що надаються штатними засобами КС.
6. Для технічного захисту даних застосовується спосіб їх приховування від можливих порушників. Можливо також використовувати спосіб технічної дезінформації. У всіх випадках заходи захисту повинні бути відповідними загрозам. Вартість цих заходів не повинна перевищувати розмір шкоди від реалізації можливих загроз.
7. Нормативні документи ТЗІ забезпечують загальну технічну політику безпеки інформації, створення і розвиток єдиної термінології систем ТЗІ, сертифікацію, ліцензування й атестацію систем захисту, розвиток послуг у галузі ТЗІ та порядок підготовки кадрів для цієї галузі.
8. Методологічною базою для визначення вимог з захисту інформації в комп'ютерних системах, створення захищених комп'ютерних систем і засобів захисту від несанкціонованого доступу, оцінки захищеності інформації в комп'ютерних системах і їх придатності для обробки критичної інформації є нормативний документ Критерії оцінки захищеності інформації в комп'ютерних системах від несанкціонованого доступу.

9. В контексті Критеріїв система захисту розглядається як набір функціональних послуг. Кожна послуга являє собою набір функцій, що дозволяють протистояти певній множині загроз. Чим вище рівень послуги, тим більш повно забезпечується захист від певного виду загроз. Критерії розбиті на чотири групи, кожна з яких описує вимоги до послуг, що забезпечують захист від загроз конфіденційності, цілісності, доступності та спостереженості.
10. Загрози, що відносяться до несанкціонованого ознайомлення з інформацією, становлять загрози конфіденційності, яку забезпечують послуги довірчої та адміністративної конфіденційності, безпечного повторного використання об'єктів, аналізу прихованих каналів і конфіденційності при обміні.
11. Загрози, що відносяться до несанкціонованої модифікації інформації, становлять загрози цілісності, яка забезпечується послугами довірчої та адміністративної цілісності, відкатом і цілісністі при обміні.
12. Загрози, що відносяться до порушення можливості користування комп'ютерними системами або доступу до інформації, становлять загрози доступності, яка забезпечується послугами використання ресурсів, стійкості до відмов, гарячої заміни та відновлення після збоїв.
13. Ідентифікація і контроль за діями користувачів, керованість комп'ютерною системою становлять предмет послуг спостереженості, яка забезпечується реєстрацією, ідентифікацією і автентифікацією, наданням достовірного каналу, розподілом обов'язків, цілісністю комплексу засобів захисту, самотестуванням, автентифікацією при обміні, автентифікацією відправника (невідмовою від авторства), автентифікацією одержувача (невідмовою від одержання).
14. Крім функціональних критеріїв, що дозволяють оцінити наявність послуг безпеки в комп'ютерній системі, є критерії гарантій, що дозволяють оцінити коректність реалізації послуг. Критерії гарантій включають вимоги до архітектури комплексу засобів захисту, середовища захисту, середовища функціонування і експлуатаційної документації. Існують сім рівнів гарантій (Г-1, ..., Г-7). Ієрархія рівнів гарантій відбиває поступово наростаючу міру певності в тому, що реалізовані в комп'ютерній системі послуги захисту дозволяють протистояти загрозам.
15. Основою захисту інформації в комп’ютерних системах є механізми керування доступом, що розв’язують задачу розмежування доступу до критичних об’єктів для суб’єктів з різними повноваженнями. У системі

обов’язково повинен бути хоч один суб’єкт з правами адміністратора, який може керувати комплексом засобів захисту. Згідно нормативним документам для розмежування доступу до конфіденційної інформації слід використовувати дискреційні механізми керування доступом, а для таємної інформації – мандатні механізми керування доступом.

16. Дискреційні механізми керування доступом створюють на основі матриць доступу, які можуть бути двовимірними або тривимірними. Необхідними двома вимірами матриці доступу є множина суб’єктів доступу та множина критичних об’єктів. У разі коли множина прав доступу вміщує більше ніж два елементи може бути побудована тривимірна матриця доступу, де третім виміром є права доступу. Недоліком дискреційних механізмів є складність підтримки в актуальному стані матриць в умовах великої кількості користувачів та об’єктів.
17. Мандатні механізми обмежені у можливостях керування доступом, але набагато простіші у підтримці. Ці механізми побудовані на основі ієрархічних міток безпеки. При цьому кожному користувачу та кожному критичному об’єкту призначається мітка, що відображає його місце у ієрархії захисту даних. Можливе сумісне використання дискреційних та мандатних механізмів керування доступом. При цьому користувач буде отримувати право доступу до об’єкта тільки у разі коли дозвіл надано як з боку дискреційного так і з боку мандатного механізмів.

Запитання та завдання для самоперевірки

1. Покажіть на структурній схемі автоматизованої системи (див. рис. 2.1) які об’єкти і від чого слід захищати та надайте пояснення діям, що можуть відбуватися у системах захисту даних.
2. Поясніть, що являє собою модель загроз та модель порушника безпеки.
3. Які способи використовують для технічного захисту інформації від несанкціонованого доступу?
4. У чому полягає реалізація політики безпеки інформації?
5. Які функціональні послуги забезпечують захист від загроз конфіденційності інформації?
6. Які функціональні послуги забезпечують захист від загроз цілісності інформації?
7. Які функціональні послуги забезпечують захист від загроз доступності інформації?

8. Які функціональні послуги забезпечують захист від загроз спостереженості?
9. Які вимоги до КЗЗ необхідно враховувати щоб забезпечити відповідність критерію гарантій?
10. Поясніть особливості дискреційних та мандатних механізмів керування доступом.

РОЗДІЛ 3. КРИПТОГРАФІЧНІ МЕТОДИ ЗАХИСТУ ДАНИХ

3.1. Основні поняття криптографії

Слово криптографія походить від двох грецьких слів κρυπτός (прихований) та γράφω (писання). Сьогодні це є назва науки про методи захисту інформації від перехоплення та підробки сторонніми особами та унеможливлення відмови від авторства.

Криптографія є однією із найстаріших наук, яка нараховує декілька тисячоліть. Але у давні часи вона обмежувалась забезпеченням конфіденційності повідомлень з використанням таємного ключа або знання як перетворювати повідомлення із відкритої форми у закриту та навпаки. За останні десятиріччя криптографія набула стрімкого розвитку як у розробці нових методів перетворення повідомлень так і у застосуванні своїх досягнень.

Перший патент на пристрій (роторну машину) для криптографії отримав у 1918 році Едвард Х. Хеберн (Edward H. Hebern) з Каліфорнії. Роторну машину було зроблено на замовлення військових і довгий час роботи в галузі криптографії були засекречені. Усі військові криптографічні пристрої часів другої світової війни були побудовані на принципах цієї роторної машини. Вихід криптографії з таємниці розпочався в 1949 році коли вийшла в світ математична робота Клода Шеннона “The Communication Theory of Secrecy Systems” (Теорія зв’язку між секретними системами) [10]. Але широка зацікавленість читачів у методах криптографії виникла як наслідок публікації у 1967 році книжки Давида Кана “The Codebreakers: The Story of Secret Writing” (Дешифрувальники. Історія секретної переписки) [11]. Ця книжка висвітила ідеї криптоаналізу або вчення про те як долати криптографічні перепони і розкривати таємниці. У наступні 20 років відбувся бурхливий розвиток робіт у галузі криптології, яка поєднала два протилежні наукові напрямки криптографію і криптоаналіз. За цей період було закладено теоретичний фундамент цієї науки і фактично вирішено усі основні задачі стосовно створення засобів, які гарантують неможливість дешифрування секретних повідомлень. Докладний опис усіх цих досягнень криптографії надано в енциклопедичному творі Брюса Шнайера Прикладна криптографія [12].

Головні поняття криптографії представлено українською мовою та мовою оригіналу на рисунку (рис. 3.1).

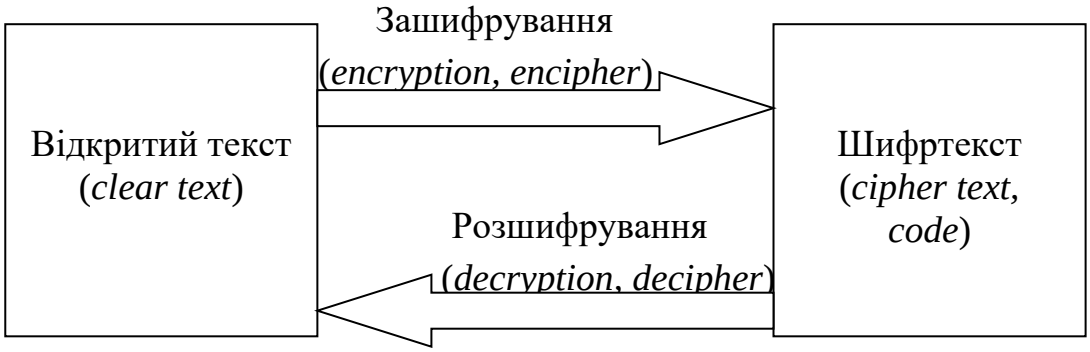

Рис. 3.1. Термінологія криптографічних перетворень.

Фахівців, що займаються криптографією з метою забезпечення захисту даних, називають криптографами, а фахівців, які займаються розкриттям (дешифруванням) зашифрованих текстів, називають криптоаналітиками. Фахівців, які поєднують обидві ці професії називають криптологами.

Сучасну криптографію використовують не тільки для забезпечення конфіденційності інформації під час передавання, але і для підтвердження цілісності повідомлень та дійсності особи відправника. Також за допомогою криптографії відправник повідомлення може впевнитись в тому, що одержувач є саме тою особою, до якої було відправлено повідомлення.

Алгоритми криптографічних перетворень можуть бути закритими (які слід зберігати у таємниці) або відкритими. У наш час перевагу надають відкритим алгоритмам, які широко відомі, ретельно перевірені на надійність та стандартизовані. У відкритих алгоритмах для здійснення процедури перетворень використовують ключ, що являє собою конкретний стан параметрів алгоритму і забезпечує вибір одного перетворення з певної множини можливих. Чим більше ця множина, тим складніше підібрати ключ і тим більшою може бути надійність захисту.

Алгоритми перетворень називають симетричними у разі коли для зашифрування і розшифрування використовують однакові ключі. Такі алгоритми пояснюються схемою перетворень на рисунку (рис. 3.2).

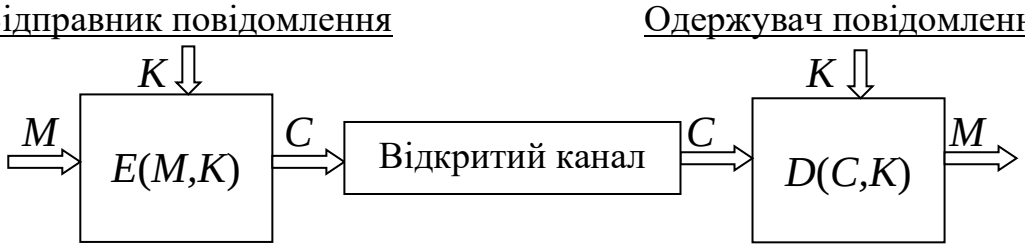

Рис. 3.2. Схема симетричної криптосистеми:

M – відкритий текст повідомлення; *K* – ключ; *C* – зашифрований текст повідомлення (шифртекст), де *C*=*E*(*M*,*K*); *E*(*M*,*K*) – криптографічний

перетворювач з відкритого тексту у шифртекст; $D(C,K)$ – криптографічний перетворювач з шифртексту у відкритий текст $M=D(C,K)$.

Криптосистема являє собою сукупність повідомлень, ключів та алгоритмів. Повідомлення надаються у вигляді послідовності бітів або байтів. Ключ являє собою число, що обирається з певної множини і надається у двійковому форматі. Алгоритми можуть бути блоковими або потоковими. Найчастіше використовують блокові алгоритми, у яких повідомлення розподіляють на блоки певної довжини і кожен із блоків шифрується окремо. У потокових алгоритмах процедура шифрування відбувається без розподілу на блоки.

У симетричних криптосистемах ключ необхідно зберігати у таємниці. Щоб забезпечити надійний захист ключів від компрометації їх періодично змінюють. При цьому для пересилання ключів слід користуватись тільки захищеними від прослуховування каналами. Саме це і є недоліком симетричних криптосистем.

Асиметричні криптосистеми використовують два різні ключі. Один ключ відкритий, яким зашифровують повідомлення, а другий – закритий, який призначений для розшифрування. При цьому немає необхідності у захищеному каналі для пересилання ключів. У цих системах одержувач генерує пару ключів (відкритий і закритий). Відкритий ключ він пересилає відправнику у відкритому вигляді. Цей ключ можна не зберігати в таємниці, бо він не дає змогу розшифрувати повідомлення. Закритий ключ одержувач зберігає в себе для розшифрування. Асиметричні алгоритми пояснюються схемою перетворень на рис. 3.3.

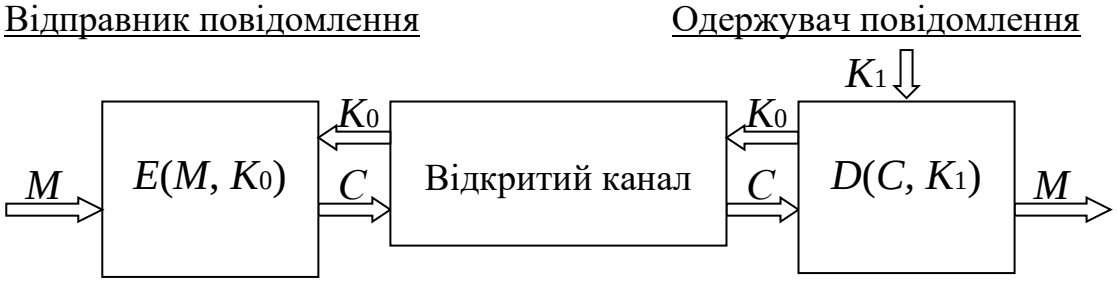

Рис. 3.3. Схема асиметричної криптосистеми:

M – відкритий текст повідомлення; K_0 – ключ для зашифровування повідомлення (відкритий ключ); C – зашифрований текст повідомлення (шифртекст), де $C=E(M,K_0)$; $E(M,K_0)$ – перетворювач з відкритого тексту у шифртекст; K_1 – закритий ключ для розшифровування повідомлення; $D(C,K_1)$ – перетворювач з шифртексту у відкритий текст $M=D(C,K_1)$.

Основним недоліком асиметричних алгоритмів є висока складність криптографічних процедур, які потребують у сотні і навіть тисячі разів більше

часу ніж у симетричних криптосистемах. Через цей недолік їх використовують тільки для передавання коротких повідомлень, наприклад, паролів або ключів для симетричних криптосистем.

У значній мірі стимулювання розвитку криптографії було пов'язано з досягненнями криптоаналітики. Так під час другої світової війни польським криптоаналітикам вдалося дешифрувати німецькі секретні повідомлення, що були зашифровані за допомогою шифрувальної машини Енігма. Алгоритми дешифрування поляки передали англійцям, які перехопили багато німецьких шифровок і змогли не тільки розкривати зміст секретних наказів, але й підробляти ці накази [13].

Розглянемо основні типи задач, які розв'язують криптоаналітики.

Задача 1. Розкриття з використанням виключно шифрованих текстів.

У математичному плані така задача є найскладнішою, але вона має велике практичне значення. Формулюється ця задача наступним чином.

Перехоплено певну множину шифрованих текстів $\{C_1,C_2,\ldots C_i,\ldots C_k\}$. При цьому відомо, що $C_i = E(M_i,K)$. Треба знайти множину відкритих текстів $\{M_1,M_2,\ldots M_i,\ldots M_k\}$ або ключ K разом із алгоритмом перетворення $D(C,K)$.

Цю задачу у сучасних системах шифрування розв'язати дуже складно, але існує багато випадків коли криптоаналітикам вдавалося успішно розв'язати таку задачу. Це часто відбувалося у минулі часи через недосконаліть криптографічних перетворень. Таке може трапитись і зараз у випадках коли захисту даних приділяють недостатньо уваги.

Задача 2. Розкриття з використанням відкритих текстів.

Крім перехоплених шифрованих текстів криптоаналітикам надано один або декілька відкритих текстів. Для цього за допомогою своїх агентів підкидають спеціальні повідомлення, які вимагають термінової доставки за допомогою системи секретного зв'язку. При цьому наперед знають, що ці повідомлення будуть передані у зашифрованому вигляді. Задача, яку розв'язують криптоаналітики формулюється так.

Відомо множину відкритих текстів $\{M_1,M_2,\ldots,M_i\}$ і відповідну множину шифрованих текстів $\{C_1,C_2,\ldots,C_i\}$. При цьому відомо, що $C_i = E(M_i,K)$. Треба знайти ключ K та алгоритм перетворення $M_i = D(C_i,K)$.

Задача 3. Розкриття з можливістю вибору відкритого тексту.

Ця задача відрізняється від попередньої тільки тим, що самому криптоаналітику надають змогу вибирати текст повідомлення, яке буде передано у зашифрованому вигляді. При цьому він має можливість підібрати таке повідомлення, яке полегшує йому розв'язання задачі.

Задача 4. Адаптивне розкриття з можливістю вибору текстів.

У цій задачі полегшення розв'язання у порівнянні з попередньою задачею полягає в тому, що криптоаналітику надається декілька спроб. Після кожної спроби він має можливість проаналізувати свої попередні результати і з врахуванням результатів цього аналізу обрати текст наступного повідомлення.

Одним з основних висновків сучасної криптографії є ствердження про те, що системи, які побудовані на основі закритих алгоритмів, у більшості випадків є слабо захищеними. За виняток можна вважати те, що у американській армії використовують закриті алгоритми шифрування, але там працюють найкращі криптологи світу.

Доведено, що існує спосіб шифрування, який неможливо розкрити за допомогою криптоаналізу. Цей спосіб полягає в тому, що до кожного байта або біта повідомлення додається випадкове число. Наприклад, у разі бітової послідовності виконується додавання за модулем 2. Тобто до кожного біта відкритого тексту додається випадковий біт. При цьому довжина випадкової послідовності дорівнює довжині повідомлення. Шенноном доведено, що у разі, коли випадкова послідовність менша за відкритий текст, захист не буде абсолютним. Головною умовою є те, що випадкова послідовність не повинна ніколи використовуватись повторно. Таким чином для кожного повідомлення випадкову послідовність слід спеціально генерувати за допомогою генератора справжніх випадкових чисел. Для утворення послідовності таких чисел не можна використовувати звичайні генератори випадкових чисел, які існують у стандартному програмному забезпеченні, бо такі генератори дають послідовність чисел, яка насправді не є випадковою і її можна відтворити. Розшифровування полягає в додаванні до закритого тексту тієї самої випадкової послідовності, яку використовували для зашифровування. Цей спосіб називають одноразовим блокнотом, бо він потребує наявності в одержувача і відправника блокнотів із записом однакових випадкових послідовностей, які можуть бути використані тільки один раз. Ці блокноти слід зберігати в абсолютній таємниці. Єдиним методом розкриття такого шифру є одержання копії блокноту.

Через надзвичайні практичні ускладнення одноразовими блокнотами користуються дуже обмежено. Усі інші методи, які ми будемо розглядати не дають абсолютної захищеності від можливого розкриття, але у реальних випадках ймовірність розкриття є на стільки малою, що користувачі цих методів можуть бути цілком задоволені результатами захисту.

Головний аргумент, який обумовлює недоцільність надмірного ускладнення технічних засобів захисту, полягає в тому, що основною причиною

розкриття таємниць є людський фактор. У переважній більшості випадків розкриття таємниць використовують підкуп, шантаж або людську необачливість.

Для оцінки якості криптографічних засобів захисту використовують поняття трудомісткості розкриття. У разі використання досконалих алгоритмів криптографічних перетворень цю трудомісткість розраховують для випадку розкриття за допомогою методу “грубої сили”. Цей метод полягає в повному переборі усіх можливих варіантів ключа та перевірки результатів кожної спроби розшифрування на схожість зі зрозумілим текстом. В умовах сучасного швидкого розвитку високопродуктивних комп’ютерних систем метод “грубої сили” з кожним роком набуває все більше можливостей у розкритті зашифрованих повідомлень.

3.2. Стандартизовані симетричні алгоритми шифрування

Для побудови симетричних алгоритмів шифрування використовують два методи, які відомі ще з давніх часів. Перший метод називають **заміною**, а другий – **перестановкою**. Під заміною розуміють дію, коли окремі символи або групи символів замінюють на інші за допомогою таблиць заміни. Наприклад, слово карусель шляхом заміни кожної букви на наступну за алфавітом можна записати у зашифрованому вигляді як лбсфтємю. При цьому таблиця заміни буде мати вигляд, який надано у таблиці 3.1.

Таблиця 3.1.

Перші рядкі шифрувальної таблиці

Символ у відкритому тексті	Символ у шифртексті
а	б
б	в
в	г

Ця таблиця являє собою ключ, який дозволяє зашифровувати і розшифровувати текстові повідомлення, що є примітивним прикладом симетричного алгоритму шифрування.

Можна побудувати набагато складнішу шифрувальну таблицю, у якій підмінятимуться не окремі символи, а їх послідовності, але все одно для сучасного криптоаналітика розкриття таких шифрів не є проблемою.

Другий метод, що називають перестановкою, полягає в тому, що в межах кожного блоку (частини повідомлення певної довжини) символи переставляють з одного місця на інше. Наприклад, якщо довжину блоків обрати 4, а кожен

символ в межах блоку переставити на місце наступного символу, а останній символ блоку переставляти на місце першого, то слово карусель у зашифрованому вигляді буде виглядати як укарьсел. При цьому таблиця перестановки буде мати вигляд, який надано у таблиці 3.2.

Таблиця 3.2.

Приклад шифрувальної таблиці за методом перестановки

Місце символу у відкритому тексті	Місце символу у шифртексті
перший	другий
другий	третій
третій	четвертий
четвертий	перший

Для покращення захисту можна використовувати комбінацію методів заміни та перестановки.

З першого погляду здається, що створення надійного симетричного алгоритму з використанням цих двох методів не є складною задачею. Але багаторічний досвід криптографів та криптоаналітиків доводить, що це не так. Навіть дуже заплутані алгоритми можуть виявитись ненадійними через непрофесійність розробників. Ті алгоритми, які розглядатимуться у цьому розділі є результатом багаторічної праці фахівців високого рівня.

Фактори, від яких залежить захищеність симетричної системи шифрування, — це надійність алгоритму та розмір ключа. При умові надійного алгоритму захищеність системи зростає експоненціально зі збільшенням довжини ключа. Алгоритм вважається надійним у разі неможливості розкриття шифру методом простішим, ніж “груба сила”, а саме тільки перебираючи можливі варіанти ключів. Легко підрахувати ймовірність подолання системи захисту методом “грубої сили” в залежності від довжини ключа.

Для того, щоб підібрати ключ використовують фрагмент відкритого тексту та відповідний до нього шифртекст. У разі, коли довжина ключа дорівнює 10 біт, достатньо перебрати 1024 можливих варіантів ключа. Після кожної спроби підбору ключа порівнюють результат дешифрування з фрагментом відкритого тексту. Вже за 512 спроб з ймовірністю 0,5 необхідний ключ може бути знайдено. Задачу такого підбору, з метою прискорення процесу пошуку ключа, можна вирішувати одразу на багатьох комп’ютерах. Враховуючи високі темпи зростання потужності комп’ютерної техніки, які можуть використовуватись для розкриття шифрів, мінімальна рекомендована довжина ключа становить 56 біт.

Щоб підібрати ключ довжиною 56 біт необхідно перебрати до 2^{56} варіантів. Якщо задіяти для вирішення цієї задачі мільйон комп'ютерів, кожен з яких за секунду буде робити мільйон перевірок, то для перебору всіх варіантів ключа необхідно витратити більше ніж 18 годин. При збільшенні ключа до 64 бітів цей час перевищуватиме 190 днів, а коли ключ збільшити до 128 біт, то час виходить за межі реального, бо становить близько 10^{19} років.

Перші кроки у напрямку стандартизації алгоритмів шифрування були зроблені в Америці Національним бюро стандартизації (National Bureau of Standards, NBS) ще на початку сімдесятих років. Здається незрозумілим сам той факт, що результати деяких секретних розробок у разі стандартизації необхідно буде оприлюднити, після чого цими результатами зможуть скористатись супротивники. Але плани були, як потім стало відомо, цілком зрозумілі. Через те, що у ті часи ще не розпочався період широкомасштабного розповсюдження обчислювальної техніки, для шифрування слід було створювати спеціальну апаратуру. Виробництво і впровадження цієї апаратури можна було утримувати під контролем з боку спеціалізованих державних органів. Більш того, можна було б примусити виробника апаратури закладати в неї спеціальні пристрої для розкриття шифрованих повідомлень контролюючими органами. Тоді мало хто міг передбачити, що комп'ютерна техніка і засоби програмування дуже швидко досягнуть рівня, коли алгоритми шифрування можна буде реалізувати у вигляді програми на звичайному персональному комп'ютері.

У 1973 році NBS запропонувало наступні вимоги до алгоритму, який можна було б стандартизувати:

— алгоритм повинен забезпечувати високий рівень безпеки;

— алгоритм повинен бути чітко визначеним та зрозумілим;

— безпечність алгоритму повинна бути побудована на ключах і не повинна залежати від знання самого алгоритму;

— алгоритм повинен бути доступним для всіх користувачів;

— алгоритм повинен адаптуватись до різних застосувань;

— алгоритм повинен забезпечити можливість економічної реалізації у вигляді електронного приладу;

— алгоритм повинен бути ефективним у користуванні;

— алгоритм повинен забезпечити можливість перевірки;

— алгоритм повинно бути дозволено експортувати.

Для перевірки відповідності алгоритмів переліченим вимогам NBS звернулось по допомогу до Агенції національної безпеки NSA (National Security Agency). У наступні роки відбувалися розгляди та обговорення різних

пропозицій і тільки 23 листопада 1976 року був прийнятий у якості федерального стандарту алгоритм під назвою DES (Data Encryption Standard). Цей стандарт офіційно було введено в дію у 1977 році, а у 1981 році Американський національний інститут стандартів ANSI (American National Standards Institute) запропонував цей стандарт для приватного користування під назвою DEA (Data Encryption Algorithm).

За весь цей період від прийняття стандарту до наших днів ставлення до алгоритму DES багато в чому змінилось, але і зараз цей алгоритм є дуже популярним у використанні.

Алгоритм DES є симетричним (однаковим для використання у процедурах зашифровування та розшифровування) та блоковим.

Для шифрування необхідно розподілити відкритий текст на блоки довжиною по 64 біти. Кожен блок у зашифрованому вигляді також має довжину у 64 біти. Ключ також має довжину у 64 біти, але з них тільки 56 слід задавати під час шифрування, бо кожний восьмий біт є бітом парності або являє собою суму за модулем 2 попередніх семи бітів.

Існує певна підмножина ключів, які не можна використовувати через суттєве послаблення захищеності, але перелік цих ключів наперед відомий.

Алгоритм являє собою комбінацію двох відомих методів перестановки і заміни, які виконуються 16 разів з використанням ключа. Крім того на початку і в кінці робиться додаткова перестановка бітів не залежно від ключа так як показано у таблиці 3.3.

Такі перестановки дуже просто реалізовувати апаратно і набагато складніше у програмному забезпеченні. Вважаючи, що ці перестановки робляться завжди за відомими правилами, то ніякого додаткового захисту від них не одержують. Єдина причина, через яку не відмовляються від цих перестановок, – дотримання стандарту. Схоже на те, що внесення до стандарту цієї перестановки було обумовлено метою ускладнення реалізації алгоритму програмними засобами без ускладнення у апаратній реалізації.

Перетворення, які відбуваються на кожному із 16 етапів (ітерацій), пояснюються схемою, яку наведено на рисунку (рис.3.4).

Таблиця 3.3.

Початкова перестановка бітів за алгоритмом DES

Номер біта		Номер біта	
на вході	на виході	на вході	на виході
1	58	33	57
2	50	34	49
3	42	35	41
4	34	36	33
5	26	37	25
6	18	38	17
7	10	39	9
8	2	40	1
9	60	41	59
10	52	42	51
11	44	43	43
12	36	44	35
13	28	45	27
14	20	46	19
15	12	47	11
16	4	48	3
17	62	49	61
18	54	50	53
19	46	51	45
20	38	52	37
21	30	53	29
22	22	54	21
23	14	55	13
24	6	56	5
25	64	57	63
26	56	58	55
27	48	59	47
28	40	60	39
29	32	61	31
30	24	62	23
31	16	63	15
32	8	64	7

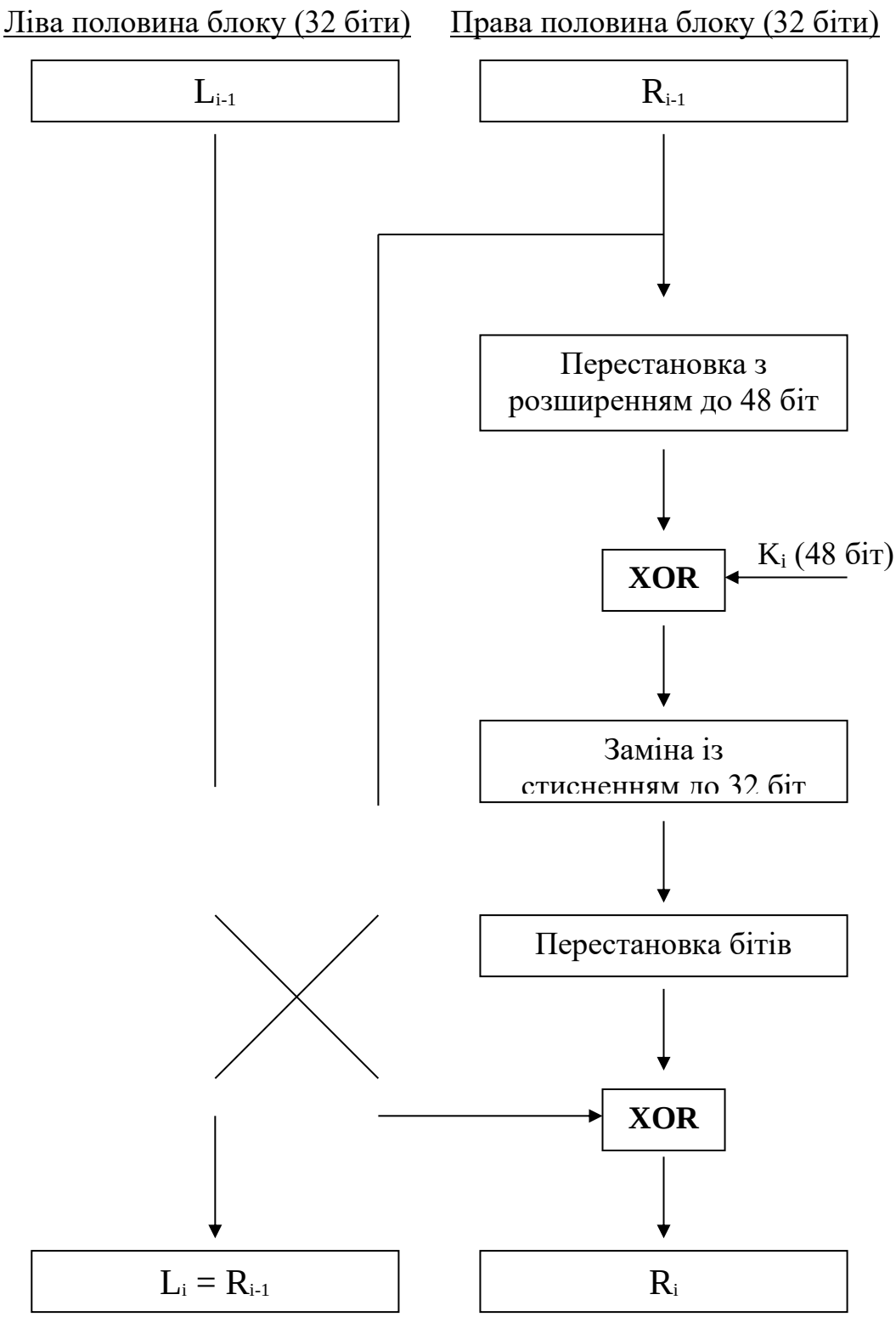

Рис. 3.4. Схема алгоритму і-го етапу перетворень за стандартом DES:

L_{i-1} – ліва половина 64-бітного блоку на початку і-го етапу перетворень; R_{i-1} – права половина 64-бітного блоку на початку і-го етапу перетворень; K_i – варіант ключа для і-го етапу перетворень; XOR – блок додавання бітів за модулем 2.

Схему такого вигляду було вперше запропоновано Фейстелем і зараз такі схеми називають схемами Фейстеля.

Розпочнемо з розгляду перетворень, які виконуються над ключем K з метою отримання 16 варіантів 48-бітних ключів для кожного з 16 етапів перетворення.

Ключ K спочатку має вигляд 64-бітного слова, у якому кожний восьмий біт є доповненням до парності. Біти парності є цілком залежними від інших бітів і тому кожний восьмий біт просто відкидають у першому блоці перестановок, як зображено на рисунку (рис.3.5).

Рис. 3.5. Схема алгоритму перетворень ключа для зашифровування

Для кожного із 16 варіантів 48-бітних ключів є відповідне значення кількості N, яке надано у таблиці 3.4, а порядок першої та другої перестановки з відкиданням бітів надано у таблицях 3.5 та 3.6 відповідно.

Таблиця 3.4.

Залежність індексу ключа для зашифровування K_i від значення N

i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
N	1	2	4	6	8	10	12	13	15	17	19	21	23	25	27	28

Таблиця 3.5.

Перша перестановка з відкиданням бітів парності

Номер біта		Номер біта	
на виході	на вході	на виході	на вході
1	57	29	63
2	49	30	55
3	41	31	47
4	33	32	39
5	25	33	31
6	17	34	23
7	9	35	15
8	1	36	7
9	58	37	62
10	50	38	54
11	42	39	46
12	34	40	38
13	26	41	30
14	18	42	22
15	10	43	14
16	2	44	6
17	59	45	61
18	51	46	53
19	43	47	45
20	35	48	37
21	27	49	29
22	19	50	21
23	11	51	13
24	3	52	5
25	60	53	28
26	52	54	20
27	44	55	12
28	36	56	4

Таблиця 3.6.

Друга перестановка з відкиданням 8 бітів

Номер біта		Номер біта	
на виході	на вході	на виході	на вході
1	14	25	41
2	17	26	52
3	11	27	31
4	24	28	37
5	1	29	47
6	5	30	55
7	3	31	30
8	28	32	40
9	15	33	51
10	6	34	45
11	21	35	33
12	10	36	48
13	23	37	44
14	19	38	49
15	12	39	39
16	4	40	56
17	26	41	34
18	8	42	53
19	16	43	46
20	7	44	42
21	27	45	50
22	20	46	36
23	13	47	29
24	2	48	32

Як бачимо з таблиці 3.6 біти із номерами 9, 16, 18, 22, 25, 38, 43 та 54 відкинуто.

Перестановка 32-бітної правої половини блоку (див. рис.3.4) з розширенням до 48 біт показано у таблиці 3.7.

Таблиця 3.7.

Перестановка з розширенням до 48 біт

Номер біта		Номер біта	
на виході	на вході	на виході	на вході
1	32	25	16
2	1	26	17
3	2	27	18
4	3	28	19
5	4	29	20
6	5	30	21
7	4	31	20
8	5	32	21
9	6	33	22
10	7	34	23
11	8	35	24
12	9	36	25
13	8	37	24
14	9	38	25
15	10	39	26
16	11	40	27
17	12	41	28
18	13	42	29
19	12	43	28
20	13	44	29
21	14	45	30
22	15	46	31
23	16	47	32
24	17	48	1

Заміну із стисненням зі 48-бітного до 32-бітного слова виконують за різними таблицями для кожної з восьми груп по 6 бітів. Ці групи називають S-блоками. До першого S-блоку входять біти з 1 до 6, до другого – з 7 до 12, а до восьмого S-блоку входять біти від 43 до 48.

В межах кожного S-блоку біти на вході позначають b_1, b_2, b_3, b_4, b_5 та b_6. Заміну для кожного з блоків показано у таблицях 3.8, 3.9, 3.10, 3.11, 3.12, 3.13, 3.14 та 3.15.

Таблиця 3.8.

Значення бітів 1-4, що є виходами першого S-блоку

Значення бітів b_2, b_3, b_4, b_5	b_1=0, b_6=0	b_1=0, b_6=1	b_1=1, b_6=0	b_1=1, b_6=1
0000	**1110**	**0000**	**0100**	**1111**
0001	**0100**	**1111**	**0001**	**1100**
0010	**1101**	**0111**	**1110**	**1000**
0011	**0001**	**0100**	**1000**	**0010**
0100	**0010**	**1110**	**1101**	**0100**
0101	**1111**	**0010**	**0110**	**1001**
0110	**1011**	**1101**	**0010**	**0001**
0111	**1000**	**0001**	**1011**	**0111**
1000	**0011**	**1010**	**1111**	**0101**
1001	**1010**	**0110**	**1100**	**1011**
1010	**0110**	**1100**	**1001**	**0011**
1011	**1100**	**1011**	**0111**	**1110**
1100	**0101**	**1001**	**0011**	**1010**
1101	**1001**	**0101**	**1010**	**0000**
1110	**0000**	**0011**	**0101**	**0110**
1111	**0111**	**1000**	**0000**	**1101**

Таблиця 3.9.

Значення бітів 5-8, що є виходами другого S-блоку

Значення бітів b_2, b_3, b_4, b_5	b_1=0, b_6=0	b_1=0, b_6=1	b_1=1, b_6=0	b_1=1, b_6=1
0000	**1111**	**0011**	**0000**	**1101**
0001	**0001**	**1101**	**1110**	**1000**
0010	**1000**	**0100**	**0111**	**1010**
0011	**1110**	**0111**	**1011**	**0001**
0100	**0110**	**1111**	**1010**	**0011**
0101	**1011**	**0010**	**0100**	**1111**
0110	**0011**	**1000**	**1101**	**0100**
0111	**0100**	**1110**	**0001**	**0010**
1000	**1001**	**1100**	**0101**	**1011**
1001	**0111**	**0000**	**1000**	**0110**
1010	**0010**	**0001**	**1100**	**0111**
1011	**1101**	**1010**	**0110**	**1100**
1100	**1100**	**0110**	**1001**	**0000**
1101	**0000**	**1001**	**0011**	**0101**
1110	**0101**	**1011**	**0010**	**1110**
1111	**1010**	**0101**	**1111**	**1001**

Таблиця 3.10.

Значення бітів 9-12, що є виходами третього S-блоку

Значення бітів b_2, b_3, b_4, b_5	b_1=0, b_6=0	b_1=0, b_6=1	b_1=1, b_6=0	b_1=1, b_6=1
0000	**1010**	**1101**	**1101**	**0001**
0001	**0000**	**0111**	**0110**	**1010**
0010	**1001**	**0000**	**0100**	**1101**
0011	**1110**	**1001**	**1001**	**0000**
0100	**0110**	**0011**	**1000**	**0110**
0101	**0011**	**0100**	**1111**	**1001**
0110	**1111**	**0110**	**0011**	**1000**
0111	**0101**	**1010**	**0000**	**0111**
1000	**0001**	**0010**	**1011**	**0100**
1001	**1101**	**1000**	**0001**	**1111**
1010	**1100**	**0101**	**0010**	**1110**
1011	**0111**	**1110**	**1100**	**0011**
1100	**1011**	**1100**	**0101**	**1011**
1101	**0100**	**1011**	**1010**	**0101**
1110	**0010**	**1111**	**1110**	**0010**
1111	**1000**	**0001**	**0111**	**1100**

Таблиця 3.11.

Значення бітів 13-16, що є виходами четвертого S-блоку

Значення бітів b_2, b_3, b_4, b_5	b_1=0, b_6=0	b_1=0, b_6=1	b_1=1, b_6=0	b_1=1, b_6=1
0000	**0111**	**1101**	**1010**	**0011**
0001	**1101**	**1000**	**0110**	**1111**
0010	**1110**	**1011**	**1001**	**0000**
0011	**0011**	**0101**	**0000**	**0110**
0100	**0000**	**0110**	**1100**	**1010**
0101	**0110**	**1111**	**1011**	**0001**
0110	**1001**	**0000**	**0111**	**1101**
0111	**1010**	**0011**	**1101**	**1000**
1000	**0001**	**0100**	**1111**	**1001**
1001	**0010**	**0111**	**0001**	**0100**
1010	**1000**	**0010**	**0011**	**0101**
1011	**0101**	**1100**	**1110**	**1011**
1100	**1011**	**0001**	**0101**	**1100**
1101	**1100**	**1010**	**0010**	**0111**
1110	**0100**	**1110**	**1000**	**0010**
1111	**1111**	**1001**	**0100**	**1110**

Таблиця 3.12.

Значення бітів 17-20, що є виходами п’ятого S-блоку

Значення бітів b_2, b_3, b_4, b_5	b_1=0, b_6=0	b_1=0, b_6=1	b_1=1, b_6=0	b_1=1, b_6=1
0000	**0010**	**1110**	**0100**	**1011**
0001	**1100**	**1011**	**0010**	**1000**
0010	**0100**	**0010**	**0001**	**1100**
0011	**0001**	**1100**	**1011**	**0111**
0100	**0111**	**0100**	**1010**	**0001**
0101	**1010**	**0111**	**1101**	**1110**
0110	**1011**	**1101**	**0111**	**0010**
0111	**0110**	**0001**	**1000**	**1101**
1000	**1000**	**0101**	**1111**	**0110**
1001	**0101**	**0000**	**1001**	**1111**
1010	**0011**	**1111**	**1100**	**0000**
1011	**1111**	**1010**	**0101**	**1001**
1100	**1101**	**0011**	**0110**	**1010**
1101	**0000**	**1001**	**0011**	**0100**
1110	**1110**	**1000**	**0000**	**0101**
1111	**1001**	**0110**	**1110**	**0011**

Таблиця 3.13.

Значення бітів 21-24, що є виходами шостого S-блоку

Значення бітів b_2, b_3, b_4, b_5	b_1=0, b_6=0	b_1=0, b_6=1	b_1=1, b_6=0	b_1=1, b_6=1
0000	**1100**	**1010**	**1001**	**0100**
0001	**0001**	**1111**	**1110**	**0011**
0010	**1010**	**0100**	**1111**	**0010**
0011	**1111**	**0010**	**0101**	**1100**
0100	**1001**	**0111**	**0010**	**1001**
0101	**0010**	**1100**	**1000**	**0101**
0110	**0110**	**1001**	**1100**	**1111**
0111	**1000**	**0101**	**0011**	**1010**
1000	**0000**	**0110**	**0111**	**1011**
1001	**1101**	**0001**	**0000**	**1110**
1010	**0011**	**1101**	**0100**	**0001**
1011	**0100**	**1110**	**1010**	**0111**
1100	**1110**	**0000**	**0001**	**0110**
1101	**0111**	**1011**	**1101**	**0000**
1110	**0101**	**0011**	**1011**	**1000**
1111	**1011**	**1000**	**0110**	**1101**

Таблиця 3.14.

Значення бітів 25-28, що є виходами сьомого S-блоку

Значення бітів b_2, b_3, b_4, b_5	b_1=0, b_6=0	b_1=0, b_6=1	b_1=1, b_6=0	b_1=1, b_6=1
0000	**0100**	**1101**	**0001**	**0110**
0001	**1011**	**0000**	**0100**	**1011**
0010	**0010**	**1011**	**1011**	**1101**
0011	**1110**	**0111**	**1101**	**1000**
0100	**1111**	**0100**	**1100**	**0001**
0101	**0000**	**1001**	**0011**	**0100**
0110	**1000**	**0001**	**0111**	**1010**
0111	**1101**	**1010**	**1110**	**0111**
1000	**0011**	**1110**	**1010**	**1001**
1001	**1100**	**0011**	**1111**	**0101**
1010	**1001**	**0101**	**0110**	**0000**
1011	**0111**	**1100**	**1000**	**1111**
1100	**0101**	**0010**	**0000**	**1110**
1101	**1010**	**1111**	**0101**	**0010**
1110	**0110**	**1000**	**1001**	**0011**
1111	**0001**	**0110**	**0010**	**1100**

Таблиця 3.15.

Значення бітів 29-32, що є виходами восьмого S-блоку

Значення бітів b_2, b_3, b_4, b_5	b_1=0, b_6=0	b_1=0, b_6=1	b_1=1, b_6=0	b_1=1, b_6=1
0000	**1101**	**0001**	**0111**	**0010**
0001	**0010**	**1111**	**1011**	**0001**
0010	**1000**	**1101**	**0100**	**1110**
0011	**0100**	**1000**	**0001**	**0111**
0100	**0110**	**1010**	**1001**	**0100**
0101	**1111**	**0011**	**1100**	**1010**
0110	**1011**	**0111**	**1110**	**1000**
0111	**0001**	**0100**	**0010**	**1101**
1000	**1010**	**1100**	**0000**	**1111**
1001	**1001**	**0101**	**0110**	**1100**
1010	**0011**	**0110**	**1010**	**1001**
1011	**1110**	**1011**	**1101**	**0000**
1100	**0101**	**0000**	**1111**	**0011**
1101	**0000**	**1110**	**0011**	**0101**
1110	**1100**	**1001**	**0101**	**0110**
1111	**0111**	**0010**	**1000**	**1011**

Після заміни із стисненням виконують перестановку бітів в одержаному 32-бітному слові так як показано у таблиці 3.16.

Таблиця 3.16.

Перестановка бітів після заміни із стисненням

Номер біта		Номер біта	
на виході	на вході	на виході	на вході
1	16	17	2
2	7	18	8
3	20	19	24
4	24	20	14
5	21	21	32
6	29	22	27
7	12	23	3
8	28	24	9
9	17	25	19
10	1	26	13
11	15	27	30
12	23	28	6
13	26	29	22
14	18	30	11
15	31	31	4
16	10	32	25

Ця процедура є передостанньою на кожному із 16 етапів перетворень за алгоритмом DES. Завершується кожен етап процедурою додавання за модулем 2 бітів, що одержані після останньої перестановки, до бітів лівої половини 64-бітного блоку (див. рис. 3.4).

Результат останнього 16-го етапу ще один раз перетворюють за тою самою таблицею (див. табл. 3.3), яку використовували для початкової перестановки бітів, але у зворотному напрямку, тобто вхід і вихід міняють місцями.

Важливою властивістю алгоритму DES є можливість використання його як для зашифровування так і для розшифровування без суттєвих змін у схемі перетворень. Фактично для розшифровування повідомлень слід змінити тільки напрямок циклічної перестановки бітів ключа так як показано на рисунку (рис. 3.6).

Рис. 3.6. Схема алгоритму перетворень ключа для розшифровування

Для кожного із 16 варіантів 48-бітних ключів відповідне значення кількості N надано у таблиці 3.17.

Таблиця 3.17.

Залежність індексу ключа K_i для розшифровування від значення N

i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
N	0	1	3	5	7	9	11	13	14	16	18	20	22	24	26	27

При цьому, порівнюючи алгоритми для зашифровування і розшифровування (див. рис. 3.5 і рис. 3.6), та враховуючи значення N з таблиць 3.4 і 3.17 відповідно, можна встановити, що ключі K_1, K_2, ..., K_{16}, для зашифровування (див. рис. 3.5) співпадають з ключами K_{16}, K_{15}, ..., K_1, для розшифровування (див. рис. 3.6).

За стандартом DES шифрування здійснюється блоками по 64 біти, тому необхідно було розробити рекомендації для тих випадків, коли довжина повідомлення не є кратною 64 бітам. Розглянемо цей момент у запропонованих за стандартом найбільш відомих режимах шифрування.

Найчастіше використовують режим електронної шифрувальної книги (Electronic Code Book, ECB) у якому кожен блок повідомлення шифрують незалежно від інших блоків, а в останньому байті останнього блоку пересилають число, що означає кількість байт, які є доповненням і повинні відкидатись після

розшифровування. При цьому, у разі коли довжина повідомлення є кратною 64 бітам необхідно відкидати весь останній блок.

Режим зчеплення блоків шифру (Cipher Block Chaining, CBC) передбачає, що, починаючи з другого блоку, перед процедурою зашифровування, кожен біт відкритого тексту додається за модулем 2 до відповідного біту попереднього зашифрованого блоку. Таким чином кожен зашифрований блок є залежним від усіх попередніх блоків даного сеансу шифрування. Цей режим забезпечує більший захист від розкриття, але користуються меншим попитом через ускладнення у реалізації.

Оскільки алгоритм DES почав швидко розповсюджуватись, як стандарт, то одночасно поширювалась діяльність у пошуку слабких місць у цьому алгоритмі. Хоч до сьогоднішнього дня ще не знайдено простого метода для розкриття шифру DES, але з кожним роком час, за який можна подолати даний засіб захисту, невпинно зменшується. Для посилення захисту у 1987 році було прийнято стандарт ISO 8732 [14], який називають 3DES. У цьому стандарті пропонується тричі шифрувати кожен блок, використовуючи два або три різні ключі. Для зашифровування спочатку слід використовувати перший ключ, потім результат розшифрувати за допомогою другого ключа і остаточно зашифрувати за допомогою першого або третього ключа. Захист забезпечується дуже надійний, але процедура шифрування стає утричі довшою.

Хоч алгоритм DES і до сьогоднішнього дня має дуже широкий спектр застосувань, але вже на початку 90-х років минулого сторіччя для особливо важливих застосувань виникала необхідність у більш надійному захисті. Головними недоліками DES вважали недостатню довжину ключа та великі витрати часу на шифрувальні процедури. Існують відомості про те, що на спеціалізованих пристроях вже у ті часи за декілька годин можна було розкрити шифри DES.

Посилений стандарт блокового шифрування з’явився у 2001 році під назвою AES (Advanced Encryption Standard) [15]. Цей стандарт має класичну (за міжнародними традиціями) історію виникнення.

На початку 1997 року Національний інститут стандартів і технологій США (National Institute of Standards and Technology, NIST) проголосив про наміри прийняття нового стандарту блокового шифрування. Майже до кінця року розглядалися пропозиції про вимоги до нового стандарту. Ці вимоги було опубліковано і одночасно оголошено про конкурс, у якому будь-яка організація світу мала можливість прийняти участь.

Вимоги полягали у наступному. Довжина блоку повинна бути 128 біт. Ключі повинні мати три можливі значення довжини 128, 192 та 256 біт. Алгоритм повинен бути легко зрозумілим та мав можливість відтворення як у апаратному, так і у програмному вигляді без зайвих ускладнень.

За період з 1998 до 2001 року було проведено три конференції, на яких були присутні сотні фахівців з різних країн світу. Обговорення було відкритим, а рішення приймались на основі голосування. На першому етапі було відібрано 15 претендентів, на другому – 5. Кожного разу всі претенденти мали можливість вдосконалювати алгоритми та поєднувати найкращі досягнення один одного.

Програма, яка реалізує алгоритм AES, є відкритою. Повний текст цієї програми з поясненнями надано у додатку 1.

3.3. Алгоритм шифрування з відкритим ключем RSA

Важливим досягненням криптографії є винахід шифрування з відкритим ключем. Цей винахід дозволяє пересилати таємні повідомлення користуючись виключно відкритими каналами. Перший повноцінний алгоритм такого шифрування було розроблено у 1977 році. Цей алгоритм має назву RSA на честь трьох винахідників (Ron Rivest, Adi Shamir, Leonard Adleman) [16]. До цього часу нам невідомі докази надійності або ненадійності цього алгоритму, але саме це і є свідченням надзвичайної складності розкриття повідомлень, що зашифровані за допомогою алгоритму RSA. Багаторічний практичний досвід використання цього алгоритму свідчить про його високу надійність.

Неможливо уявити собі можливість використання алгоритму RSA в умовах відсутності високопродуктивної комп'ютерної техніки, бо цей алгоритм побудовано на добутку двох простих чисел довжиною від 384 до 1024 бітів. Найбільш розповсюдженою є довжина 512 біт. Обидва ключі відкритий, який пересилають відправнику у відкритому вигляді для зашифровування повідомлень, і закритий, який слід зберігати в таємниці для розшифровування цих повідомлень, генерує одержувач за наступним алгоритмом.

1. Обирають два великі прості числа q і p так, щоб вони були достатньо віддалені одне від одного, бо для близьких чисел існує метод розкриття шифру;
2. Обчислюють добуток $n = qp$, який і являє собою головну частину відкритого ключа;
3. Обирають число e, яке є другою частину відкритого ключа за умовою, щоб e і $(q-1)(p-1)$ були взаємно простими числами (число e вибирають

із наступного ряду 3, 5, 17, 257, ... так, щоб воно було простим і у двійковому форматі мало тільки дві одиниці – на першому і останньому місцях);

4. Пару чисел n і e пересилають відправнику для зашифровування (це є відкритий ключ);

5. За допомогою розширеного алгоритму Евкліда знаходять число d, що являє собою закритий ключ, таким чином, щоб виконувалась умова $ed \equiv 1 \pmod{(q-1)(p-1)}$, де знак потрійного рівняння означає конгруентність;

В залежності від значення n відправник розділяє повідомлення для шифрування на блоки такої довжини, щоб вони не перевищували довжину n. У разі, якщо блок коротший за n, то він доповнюється нулями у лівій частині до довжини n.

Для зашифровування повідомлення M у шифртекст C відправник повинен виконати наступну математичну дію

$$C \equiv M^e \pmod{n}.$$

Для розшифровування повідомлення M одержувач за допомогою закритого ключа d виконує дію

$$M \equiv C^d \pmod{n}.$$

Легко довести, що після виконання цієї дії буде відтворено саме те повідомлення M, яке було зашифроване.

Доказ полягає у наступному. Підставимо M^e замість C у останній вираз. Після цього одержимо $M \equiv M^{ed} \pmod{n}$, а за умовою вибору ключів $ed \equiv 1 \pmod{(q-1)(p-1)}$, де добуток $(q-1)(p-1)$ являє собою функцію Ейлера $\varphi(n) = \varphi(q)\varphi(p)$. Таким чином цю умову вибору ключів можна записати як $ed = k\varphi(n) + 1$, де k – ціле число. Далі, скориставшись цією заміною, запишемо $M^{ed} \equiv M^{k\varphi(n)+1} \pmod{n}$.

За теоремою Ейлера $M^{\varphi(n)} \equiv 1 \pmod{n}$, що означає $M^{k\varphi(n)+1} = M(M^{\varphi(n)})^k \equiv M \pmod{n}$. Саме це і треба було довести.

Нагадаємо, що функція Ейлера $\varphi(n)$ від числа n являє собою кількість чисел в інтервалі від 1 до $n-1$, які є взаємно простими до n. Для довільного простого числа p функція Ейлера $\varphi(p) = p - 1$. Ця функція є мультиплікативною, тобто $\varphi(qp) = \varphi(q)\varphi(p)$. Взаємно простими називають числа, які не мають спільних дільників, крім одиниці. За теоремою Безу числа a та b є взаємно простими тоді і тільки тоді, коли існують цілі x та y, для яких $ax+by = 1$.

Розглянемо простий приклад перетворень за алгоритмом RSA.

Візьмемо два простих числа $p = 5$ та $q = 11$. Знайдемо їх добуток n=55. Функція Ейлера $\varphi(n) = (p - 1)(q - 1) = 40$. Взаємно просте до $\varphi(n)$ число $e = 3$, що легко перевірити за теоремою Безу $3(-13) + 40 1 = 1$.

Пара чисел 3 та 55 є відкритим ключем для зашифровування повідомлень.

Закритий ключ d підбираємо за умовою $ed \equiv 1 \pmod{40}$. За цієї умови добуток ed може прийняти значення $40k + 1$, де k – натуральне число. У нашому випадку підійде значення 81, при якому ключ $d = 27$.

Зашифруємо повідомлення $M = 13$. Для цього обчислимо

$$C \equiv M^3 \pmod{55}.$$

Обчислення за модулем можна виконувати поетапно, щоб результат ніколи не перевищував 55. Спочатку $13^2 13 = 169 13$, де $13^2 \equiv 4 \pmod{55}$, а потім $4 13 \equiv 52 \pmod{55}$.

Число 52 у нашому прикладі є зашифрованим повідомленням.

Для розшифровування слід обчислити значення

$$M \equiv C^{27} \pmod{55}.$$

Для зручності розрахунків запишемо $C^{27} = C^{16} C^8 C^2 C$.

Легко обчислити всі ці значення $52^2 \equiv 9 \pmod{55}$, $52^4 \equiv 26 \pmod{55}$, $52^8 \equiv 16 \pmod{55}$, $52^{16} \equiv 36 \pmod{55}$, $36 16 9 52 \equiv 13 \pmod{55}$.

Щоб розкрити шифр, знаючи варіанти повідомлень, один з яких може бути передано у зашифрованому вигляді, можна зашифрувати всі ці варіанти та порівняти свої результати із переданим шифртекстом. Якщо серед відомих варіантів є той, який було передано, то вони співпадуть. Щоб унеможливити розкриття цим методом, кінцівки повідомлень доповнюють випадковим текстом.

Основним методом розкриття для алгоритму RSA залишається підбір двох простих чисел, що є дільниками числа n, але ця задача для значень n, які рекомендовані сьогоднішніми стандартами, не може бути розв'язана в умовах сучасної дійсності. У майбутньому, за умов розвитку комп'ютерної техніки, підвищуватимуться можливості генерування простих чисел більшої довжини для шифрування за алгоритмом RSA, залишаючи позаду можливості розкриття цих шифрів.

Фактором, який обмежує використання алгоритму RSA, є значне збільшення часу у порівнянні із симетричними алгоритмами. Витрати комп'ютерного часу, що пов'язані зі знаходженням великих простих чисел, не є критичним, бо ці числа можна підготувати заздалегідь. Головне те, що процедури шифрування за алгоритмом RSA у сотні разів довші, ніж за алгоритмами DES або AES.

У сучасних системах секретного зв'язку за допомогою алгоритму RSA пересилають ключі до симетричних систем шифрування, а потім передають дані за допомогою симетричних алгоритмів. Такий підхід забезпечує високу ефективність використання ресурсів комп'ютерних систем.

Алгоритм RSA також застосовують для цифрових підписів, які унеможливлюють підробку документів та відмову від авторства. Такі підписи утворюють за допомогою геш-функцій, які ми розглянемо у наступному підрозділі.

3.4. Поняття геш-функції та цифровий підпис

Геш-функція $H=h(M)$ перетворює рядок бітів M довільної довжини у рядок бітів H, що має фіксовану довжину m.

З метою ефективного використання геш-функцій у задачах, що пов'язані із підтвердженням особи відправника (цифровий підпис), зберігання паролів та перевірки цілісності повідомлень, до цих функцій висувають наступні вимоги.

- Перетворення із M у H не повинно бути складним.
- Перетворення із H у M повинно бути складним або неможливим.
- Не знаючи секретної частини (ключа) дуже складно підібрати такі M та H, щоб $H=h(M)$.
- Дуже складно підібрати друге значення M' для якого $h(M')=h(M)$.

Найкращі з багатьох запропонованих алгоритмів щодо утворення геш-функцій були стандартизовані або запатентовані. Найвідоміші з них MD-4, MD-5, SHA-1, SHA-2, ГОСТ-34.311. Скорочення MD означає Message Digest (стиснене повідомлення). Як результат вдосконалення алгоритму MD-4 з'явились алгоритми MD-5 та SHA (Secure Hash Algorithm – Алгоритм безпечного гешування). В Україні прийнято використовувати геш-функцію за російським стандартом ГОСТ-34.311. Порівняння деяких параметрів перелічених функцій наведено у таблиці 3.18.

Таблиця 3.18.

Параметри найбільш відомих геш-функцій

Найменування	Рік випуску	Довжина, біт	Швидкодія, Мбіт/с*
MD4	1990	128	5,66
MD5	1992	128	4,18
ГОСТ 34.311	1994	256	0,26
SHA-1	1995	160	1,8
SHA-2	2002	256/224; 512/384	?

* Швидкодію надано для процесора i486SX з частотою 100 МГц

За допомогою геш-функцій можна утворювати імітовставки з метою перевірки цілісності повідомлень, а також формувати цифровий підпис.

Для утворення цифрового підпису до появи сучасних стандартів широко використовували алгоритм RSA, який було розглянуто у попередньому підрозділі.

Для того, щоб отримати цифровий підпис за цим алгоритмом виконуються наступні дії.

Повідомлення, яке потребує цифровий підпис, розділяють на блоки по 512 біт та зашифровують, використовуючи режим зчеплення блоків шифру (Cipher Block Chaining, CBC). Наприклад, якщо M_1, M_2, ... M_k є блоки повідомлення M, то спочатку зашифровують блок M_1 за алгоритмом RSA. Результат цього шифрування $C_1 \equiv M_1^e \pmod{n}$ додають по бітам за модулем 2 до блоку M_2. Результат додавання зашифровують за тим самим алгоритмом RSA і одержують C_2. Цей результат додають по бітам за модулем 2 до блоку M_3. Останній блок M_k, доповнений нулями до 512 біт, після додавання по бітам за модулем 2 до C_{k-1} зашифровують за тим самим алгоритмом RSA. Останній результат C_k і являє собою цифровий підпис, який пересилають після останнього блоку повідомлення.

Алгоритм описаних перетворень для випадку k=3 пояснюються рисунком (рис. 3.7).

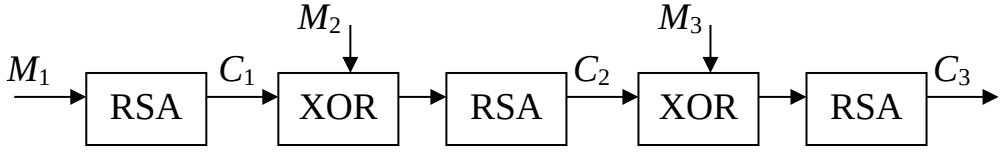

Рис. 3.7. Схема формування цифрового підпису за алгоритмом RSA:
RSA – блок перетворень за алгоритмом RSA; XOR – блок додавання бітів за модулем 2.

Особливість розглянутого алгоритму полягає в тому, що ключі (відкритий і закритий) формує відправник. Для утворення цифрового підпису він використовує закритий ключ e, а одержувач перевіряє отримане повідомлення за допомогою відкритого ключа d. Взагалі у алгоритмі RSA відкритий і закритий ключі e і d можна міняти місцями.

Для перевірки вірності отриманого повідомлення, що являє собою послідовність блоків M_1, M_2, ... M_k та C_k, одержувач повинен за допомогою

відкритого ключа d виконати послідовність процедур, яка представлена на рисунку (рис. 3.8).

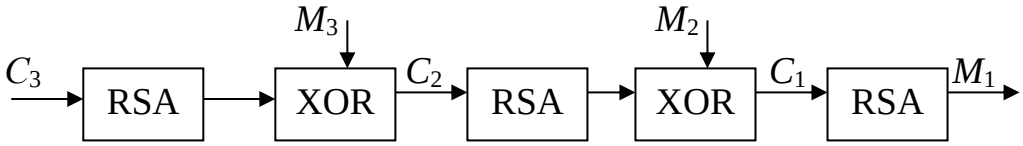

Рис. 3.8. Схема перевірки цифрового підпису за алгоритмом RSA

Одержане після цих перетворень значення M_1 повинно співпасти із початком повідомлення. Тільки в цьому випадку можна вважати, що повідомлення дійсно не пошкоджено і відправлено саме тією особою, від якої ми його очікуємо.

Хоч використання алгоритму RSA для цифрового підпису було дуже поширено, його навіть вважали стандартом де-факто, але у 1991 році NIST (National Institute of Standards and Technology) запропонував стандарт цифрового підпису DSS (Digital Signature Standard), який було побудовано без використання алгоритму RSA. Ця пропозиція була сприйнята з обуренням з боку прихильників RSA, але NIST пропонував надійний та безкоштовний засіб утворення цифрового підпису, який можна широко використовувати для міжнародного листування, у той час коли для RSA необхідно було купляти ліцензію.

Остаточно стандарт DSS було видано 19 травня 1994 року. В основу стандарту покладено схему Ель-Гамаля [17], криптографічна стійкість якої заснована на складності розв'язання задачі дискретного логарифмування. Ця задача полягає у знаходженні значення x з виразу

$$y \equiv g^x \pmod{p}$$

де y, g та p – цілі числа.

При тому що можна без ускладнень знайти значення y, знаючи значення x, зворотна задача для великих чисел не є простою. Є випадки коли для деяких комбінацій чисел x, g та p існує спрощене розв'язання цієї задачі. Щоб уникнути таких випадків і забезпечити високу криптографічну стійкість даної схеми, під час вибору значень x, g та p накладають спеціальні умови. Протягом останніх десятиріч ці умови вдосконалювались. На початку, так як було запропоновано Ель-Гамалем, умови формулювались наступним чином.

- Число p повинно бути простим і більшим за x та g.
- Число x повинно бути взаємно простим до $(p - 1)$.
- Числа y, g та p являють собою відкритий ключ, що потрібен для перевірки цифрового підпису.

- Число x слід зберігати у таємниці як частину закритого ключа для формування цифрового підпису.

Для отримання цифрового підпису необхідно за допомогою генератора випадкових чисел обрати ще одне число k, яке також як x повинно бути взаємно простим до $(p - 1)$ і меншим за p. Цифровий підпис складається з двох цілих чисел a та b, які знаходять з наступних виразів

$$a \equiv g^k \bmod p, \quad M \equiv (xa - kb) \bmod (p - 1),$$

де M – повідомлення, яке підписують.

Для кожного підпису слід обирати нове число k, а число x можна залишати те саме. Маючи два різні підписані документи з однаковими значеннями k, одержувач може вирахувати значення x і скомпрометувати закритий ключ. Тому важливою умовою користування таким цифровим підписом є необхідність обирати різні значення числа k для кожного нового документу. Пара чисел x та k являють собою закритий ключ.

Перевірка цифрового підпису одержувачем полягає у перевірці наступного рівняння

$$y^a a^b \bmod p = g^M \bmod p.$$

Значним кроком щодо вдосконалення систем цифрового підпису є використання криптографічних алгоритмів, що базуються на еліптичних кривих над полями Галуа. Цей метод було запропоновано у роботах Міллера [18] та Кобліца [19]. Задачу дискретного логарифмування було перекладено на більш стійку з точки зору криптографії основу. Цією основою стала множина точок заданої еліптичної кривої над полем Галуа. Суттєва перевага, яку при цьому було отримано, виявилась у значному зменшенні необхідної довжини ключа для забезпечення рівноцінної криптографічної стійкості у порівнянні з існуючими методами. У новій системі ключ довжиною 160 біт забезпечує таку ж криптографічну стійкість, як ключ із 1024 бітів у інших відомих системах. Зараз в усіх провідних країни світу для цифрового підпису прийнято стандарти, що базуються на використанні еліптичних кривих над полем Галуа. В Америці стандарт ANSI x9.62, що базується на ECDSA (Elliptic Curve Digital Signature Algorithm – Алгоритм цифрового підпису на еліптичних кривих) було прийнято у 1998 році. На Україні подібний стандарт прийнято у 2003 році [20].

Формування цифрового підпису за стандартом [20] складається з двох етапів. На першому етапі відбувається обчислення геш-функції для повідомлення, яке треба підписувати. У стандарті пропонується використовувати геш-функцію, яка описана у ГОСТ 34.311, або будь-яку іншу

функцію гешування, рекомендовану уповноваженим органом державної влади у сфері криптографічного захисту інформації.

Розглянемо алгоритм гешування, що пропонують у ГОСТ 34.311-95. За цим алгоритмом повідомлення M розподіляють на блоки $m_1, m_2,\ldots, m_n$ довжиною 256 біт. Останній блок доповнюють нулями до 256 бітів. Кожен блок, починаючи з першого, перетворюють у проміжне значення геш-функції $H_{i+1} = f(H_i, m_i)$. Значення H_1 обирають довільно. Після обчислення H_{n+1} обчислюють $H_{n+2} = f(H_{n+1}, L)$, де L – довжина повідомлення M у бітах за модулем 2^{256}, та $h = f(H_{n+2}, K)$, де K – контрольна сума повідомлення M, а h – остаточне значення геш-функції. Контрольну суму обчислюють сумуючи блоки $m_1, m_2,\ldots, m_n$ за модулем 2^{256}.

Алгоритм цих перетворень пояснюються рисунком (рис. 3.9).

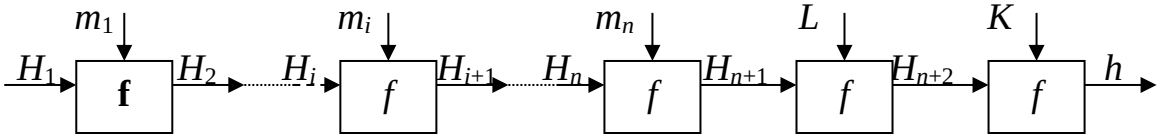

Рис. 3.9. Схема формування геш-функції за алгоритмом ГОСТ 34.311

Функція $f(H, m)$ являє собою перетворення, що складається з наступних трьох етапів:

- Формування 256-бітних ключів K_1, K_2, K_3, K_4 з використанням значень H та m;
- Шифрування за алгоритмом ГОСТ 28147-89 кожної з чотирьох 64 бітних частин 256-бітного значення H за допомогою відповідного ключа;
- Перетворення за допомогою модифікованої схеми Фейстеля.

Усі ці етапи докладно розглянуто у роботі Дениса Шефановського, з якою можна ознайомитись у мережі Інтернет на загальнодоступному сайті www.ssl.stu.neva.ru/psw/cripto/GOST_R_34.11-94_Pub.pdf.

За допомогою геш-функції одержують стисле відображення повідомлення. Таке відображення називають MD-підсиленням, де MD означає Message Digest. Важливою властивістю цього відображення є лавинний ефект, який полягає в тому, що заміна хоч одного біта у повідомленні викличе заміну багатьох бітів у відображенні.

Для обчислення геш-функції за стандартом [20] використовують відкритий алгоритм. Цей алгоритм і усі параметри щодо обчислення геш-функції відкриті для ознайомлення. Відправник повідомлення шифрує геш-функцію за допомогою свого особистого закритого ключа і таким чином створює цифровий підпис. Цифровий підпис пересилається в кінці повідомлення.

Для зашифровування геш-функції у стандарті для цифрового підпису рекомендовано використовувати алгоритм, що базується на еліптичних кривих над основним полем Галуа GF(2^m). Рекомендовано ряд конкретних значень степені основного поля m. Ці значення повинні бути простим числом від 163 до 509 в залежності від необхідного рівня захисту. Також стандартизовано формулу еліптичної кривої

$$y^2 + xy = x^3 + Ax^2 + B,$$

де значення A та B рекомендовано обирати з наданих у стандарті таблиць для різних показників степені m від 163 до 509. Значення змінних x та y повинні бути елементами поля Галуа GF(2^m).

Наприклад, для степені $m = 163$ за стандартом рекомендовано обирати $A = 1$, $B =$ 5FF6108462A2DC8210AB403925E638A19C1455D21, а рекомендований порядок базової точки еліптичної кривої при цьому ж значенні степені $n =$ 400000000000000000002BEC12BE2262D39BCF14D.

Порядком базової точки еліптичної кривої є просте непарне число, для якого виконується умова $nP = O$, де P – базова точка еліптичної кривої, а O – нескінченно віддалена точка еліптичної кривої.

Координати (x_P, y_P) базової точки P еліптичної кривої можуть бути обчислені для заданого значення n.

Процедура формування підпису подібна до схеми Ель-Гамаля. На початку знаходять випадкове ціле число e, $0 < e <$ n та обчислюють точку еліптичної кривої $R = e\ P$ з координатами (x_R, y_R). Після цього перевіряють значення координати x_R. Якщо $x_R = 0$, то обирають інше випадкове число e. Після закінчення перевірок вважають обраним таємний параметр e та відповідний цифровий передпідпис $F_e = x_R$. Далі обчислюють добуток двох елементів поля Галуа GF(2^m), а саме F_e та h, де h – результат обчислення геш-функції, від якого обрано тільки молодшу частину із m бітів. Цей добуток у вигляді цілого числа r є одним з пари чисел цифрового підпису. Друге число цифрового підпису s обчислюють за формулою

$$s \equiv (e + dr) \bmod n,$$

де d – закритий персональний ключ ($0 < d < n$).

У випадку коли $s = 0$ слід переобрати випадкове число e та переробити усі обчислення, починаючи з точки R .

Одержану пару чисел r та s перетворюють на цифровий підпис.

Підписане повідомлення має наступний вигляд

$$iH \parallel M \parallel s \parallel r\ ,$$

де *iH* – ідентифікатор геш-функції, що являє собою послідовність із восьми бітів, які для функції за алгоритмом ГОСТ 34.311 дорівнюють 00000001. Цей ідентифікатор не є обов'язковим у випадках коли використовують тільки одну геш-функцію.

Знак ‖ означає конкатенацію бітових рядків.

Числа s та r у електронному підписі доповнюють у лівій частині нулями до наперед заданої довжини, яка повинна бути кратною 16 бітам. Рекомендована максимальна довжина підпису становить 512 біт. При цьому кожне з чисел s та r слід доповнити до 256 біт.

Для перевіряння цифрового підпису треба мати відкритий ключ Q і всі відкриті параметри цифрового підпису. При цьому послідовність дій повинна бути наступною.

- З прийнятої послідовності бітів виділяють повідомлення M та обчислюють геш-функцію.
- Обчислюють координати (x_R, y_R) для точки $R = sP + rQ$.
- Обчислюють елемент поля $p = h\, x_R$.
- У разі $p = r$ вважають, що підпис дійсний.

3.5. Алгоритм Діффі-Хелмана

Перший у світі алгоритм для отримання закритих ключів без використання закритого каналу зв'язку було запропоновано в роботі Діффі та Хелмана у 1976 році [21]. Таким чином, цей алгоритм з'явився на цілий рік раніше від алгоритму RSA.

Для того, щоб з обох кінців системи секретного зв'язку можна було б отримати однакові секретні ключі, за цим алгоритмом необхідно підготувати пару простих чисел q та p. Ці числа можуть бути відкритими і використовуватись багатьма користувачами необмежену кількість разів. При цьому, число p вибирають таким, щоб воно перевищувало розмір ключів, а число q може дорівнювати 2, 3, 5 або 7 за умовою, щоб воно являло собою твірний елемент мультиплікативної групи кільця лишків за модулем p. Це означає, що повинні виконуватись наступні дві умови:

$$q^{\varphi(p)} \equiv 1 \bmod p$$

та

$$q^{l} \neq 1 \bmod p, \quad 0 < l < \varphi(p),$$

де $\varphi(p)$ – функція Ейлера.

Для підвищення криптографічної стійкості алгоритму рекомендовано накладати на вибір числа p додаткову умову, а саме, щоб $(p - 1)/2$ було також простим числом. Витрати часу на пошук необхідних пар чисел q та p не є критичними, бо ці числа готуються заздалегідь. Можна підготувати декілька таких пар з різною довжиною числа p для забезпечення обміну ключами різної довжини. Зрозуміло, що зі збільшенням довжини ключів зростає криптографічна стійкість системи, але збільшуються витрати часу на шифрувальні процедури.

Алгоритм пошуку ключів складається з наступний дій.

Обидва абоненти незалежно один від одного обирають великі випадкові числа a та b відповідно, які повинні бути менші за p. Кожен з абонентів за допомогою свого випадкового числа обчислює допоміжний параметр $A = q^a \bmod p$ та $B = q^b \bmod p$ відповідно. Ці додаткові параметри абоненти пересилають один одному, після чого кожен з них може знайти значення секретного ключа K за однією з формул

$$K = B^a \bmod p$$

або

$$K = A^b \bmod p.$$

Легко довести, що $B^a \bmod p = A^b \bmod p$.

Для доведення замість A та B підставимо їх значення, після чого отримаємо рівняння $q^{ab} \bmod p = q^{ba} \bmod p$.

Криптографічна стійкість даного алгоритму заснована на складності розв’язання задачі дискретного логарифмування, яку ми розглянули у попередньому підрозділі 3.4.

Цей алгоритм широко використовується у системах для обміну конфіденційною інформацією в режимі точка-точка.

Недоліком розглянутого алгоритму є неможливість вибору значення ключа наперед, щоб не потрапити на який-небудь не рекомендований ключ зі списку слабих ключів.

Можливо розширити алгоритм для довільної кількості учасників обміну секретними даними. Розглянемо варіант алгоритму для трьох учасників. При цьому будемо вважати, що перший учасник обрав випадкове число a, другий обрав число b , а третій – число c. Кожен з них
обчислив допоміжний параметр $A=q^a \bmod p$, $B=q^b \bmod p$ та $C=q^c \bmod p$ відповідно. Всі вони відправляють отриманий параметр іншому учаснику. При цьому перший відправляє другому, другий – третьому, а третій – першому. Після цього кожен з них обчислює ще один додатковий параметр. Перший обчислює $A' = C^a \bmod p$ та відправляє другому, другий обчислює $B' = A^b \bmod p$ та

відправляє третьому, а третій обчислює $C' = B^c \bmod p$ та відправляє третьому. Після такого обміну по колу кожен учасник
може обчислити значення спільного секретного ключа. Цей ключ дорівнює $K = q^{abc} \bmod p$.

Існує ще один вдосконалений варіант алгоритму Диффі-Хелмана, який дозволяє обчислити закритий варіант ключа одному з учасників та надати можливість іншому учаснику знайти значення цього закритого ключа за допомогою наступної послідовності дій.

- Перший учасник обирає випадкове число a та обчислює закритий ключ за формулою $K = q^a \bmod p$.
- Другий учасник обирає випадкове число b, обчислює параметр $B=q^b \bmod p$ та відправляє цей параметр першому учаснику.
- Перший учасник обчислює додатковий параметр $A=B^a \bmod p$ та відправляє його другому учаснику.
- Другий учасник знаходить число c з виразу $cb=1 \bmod p$.
- Другий учасник обчислює закритий ключ $K = A^c \bmod p$.

Висновки

1. Криптографія є однією із найстаріших наук, яка за останні десятиріччя набула стрімкого розвитку. За цей період розроблено та впроваджено багато математичних методів захисту інформації від несанкціонованого розповсюдження, від перетворення даних під час зберігання або передавання. Вирішено проблему підтвердження особи відправника повідомлень та неможливості відмовлення від авторства.
2. Алгоритми криптографічних перетворень можуть бути закритими, які зберігають у таємниці, або відкритими. Перевагу надають відкритим алгоритмам, які широко відомі, ретельно перевірені на надійність та стандартизовані.
3. Для здійснення криптографічних перетворень використовують ключ, що являє собою конкретний стан параметрів алгоритму і забезпечує вибір одного перетворення з множини можливих. Чим більше ця множина, тим складніше підібрати ключ і тим більшою буде надійність захисту.
4. Алгоритми перетворень можуть бути симетричними (коли для зашифровування і розшифровування використовуються однакові ключі) або асиметричними, у яких використовують два різні ключі. Один ключ відкритий, яким зашифровують повідомлення, а другий – закритий, який призначений для розшифровування. У цих системах немає необхідності у

захищеному каналі для пересилання ключів. Одержувач генерує пару ключів і пересилає відправнику відкритий ключ. Цей ключ можна не зберігати в таємниці, бо він не дає змогу розшифровувати повідомлення. Закритий ключ одержувач зберігає в себе для розшифровування.

5. Недоліком асиметричних алгоритмів є висока складність процедур шифрування, які потребують у сотні і навіть у тисячі разів більше ресурсів ніж симетричні алгоритми. Через це асиметричні алгоритми використовують тільки для передавання коротких повідомлень, наприклад, паролів або ключів для симетричних криптосистем.
6. Криптоаналіз являє собою науковий напрямок, який вирішує задачі розкриття зашифрованих текстів. Успіхи криптоаналітиків примушують вдосконалювати системи криптографічного захисту. Наука криптологія поєднує два наукових напрямки: криптографію і криптоаналіз.
7. Доведено, що існує спосіб шифрування, який неможливо розкрити за допомогою криптоаналізу. Цей спосіб полягає в тому, що до кожного байта або біта повідомлення додається випадкове число. Шенноном доведено, що у разі, коли випадкова послідовність менша за відкритий текст, захист не може бути абсолютним.
8. Для оцінки якості криптографічних засобів захисту використовують поняття трудомісткості розкриття. У разі використання досконалих алгоритмів криптографічних перетворень цю трудомісткість розраховують для випадку розкриття за допомогою методу “грубої сили”. Цей метод полягає в повному переборі усіх можливих варіантів ключа та перевірки результатів кожної спроби розшифрування на схожість зі зрозумілим текстом.
9. Фактори, від яких залежить захищеність симетричної системи шифрування, – це надійність алгоритму та розмір ключа. При умові надійного алгоритму захищеність системи зростає експоненціально зі збільшенням довжини ключа. Алгоритм вважається надійним у разі неможливості розкриття шифру методом простішим, ніж “груба сила”.
10. Перший стандартний симетричний алгоритм шифрування під назвою DES (Data Encryption Standard) було запропоновано у 1976 році. Довжину ключа було обрано у 56 біт, а розмір блоків для шифрування – 64 біти. Завдяки швидкому розповсюдженню DES одночасно поширювалась діяльність з пошуку його слабких місць. Хоч до сьогоднішнього дня ще не знайдено простого метода для розкриття шифру DES, але з кожним роком час, за який можна подолати даний засіб захисту, невпинно зменшується. Для

посилення захисту у 1987 році було прийнято стандарт 3DES. У цьому стандарті пропонується тричі шифрувати кожен блок, використовуючи два або три різні ключі.

11. Посилений стандарт блокового шифрування з'явився у 2001 році під назвою AES (Advanced Encryption Standard). Довжина блоку AES дорівнює 128 біт, а ключі можуть мати три значення довжини (128, 192 або 256 біт), яку обирають в залежності від необхідного рівня захищеності.
12. Важливим досягненням криптографії є винахід шифрування з відкритим ключем. Перший повноцінний алгоритм для такого шифрування було розроблено у 1977 році. Цей алгоритм має назву RSA на честь трьох винахідників (Ron Rivest, Adi Shamir, Leonard Adleman). Алгоритм RSA побудовано на добутку двох простих чисел довжиною від 384 до 1024 бітів. Щоб розкрити шифр треба розкласти цей добуток на множники, що є дуже складною задачею. Багаторічний практичний досвід використання алгоритму RSA свідчить про його високу надійність.
13. Для розв'язання задач, що пов'язані із підтвердженням особи відправника (цифровий підпис), зберігання паролів та перевірки цілісності повідомлень використовують функції, що перетворюють рядок бітів довільної довжини у рядок бітів фіксованої довжини. Такі функції називають геш-функції. Найкращі з багатьох алгоритмів утворення геш-функцій були стандартизовані або запатентовані. Найвідоміші з них MD-4, MD-5, SHA-1, SHA-2, ГОСТ-34.311.
14. Перший стандарт для цифрового підпису DSS (Digital Signature Standard) було видано 19 травня 1994 року. В основу цього стандарту покладено схему Ель-Гамаля [17], криптографічна стійкість якої заснована на складності розв'язання задачі дискретного логарифмування. Ця задача полягає у знаходженні значення x з виразу $y \equiv g^x \pmod{p}$, де y, g та p – відомі цілі числа.
15. Значним кроком щодо вдосконалення систем цифрового підпису є використання криптографічних алгоритмів, що базуються на еліптичних кривих над полями Галуа. При цьому задача дискретного логарифмування перекладається на більш стійку з точки зору криптографії основу. Цією основою є множина точок еліптичної кривої над полем Галуа. Перевага такого рішення полягає у значному зменшенні необхідної довжини ключа для забезпечення рівноцінної криптографічної стійкості у порівнянні з існуючими методами.

16. Сучасні стандарти цифрового підпису базуються на еліптичних кривих над полями Галуа. Перший такий стандарт ANSI x9.62 в Америці було прийнято у 1998 році. На Україні стандарт цифрового підпису ДСТУ 4145-2002, що ґрунтується на еліптичних кривих прийнято у 2003 році.
17. Перший у світі алгоритм для отримання закритих ключів без використання закритого каналу зв'язку було запропоновано в роботі Диффі та Хелмана у 1976 році. Криптографічна стійкість даного алгоритму заснована на складності розв'язання задачі дискретного логарифмування. Цей алгоритм широко використовується у системах обміну конфіденційною інформацією в режимі точка-точка. Існує розширення цього алгоритму для довільної кількості учасників обміну секретними даними. Недоліком розглянутого алгоритму є неможливість вибору значення ключа наперед.

Запитання та завдання для самоперевірки

1. Надайте розширене поняття криптографії, як одного з актуальних сучасних наукових напрямків.
2. Чому слід надавати перевагу відкритим алгоритмам шифрування, які широко відомі та стандартизовані?
3. Від чого залежить криптографічна стійкість симетричних алгоритмів шифрування?
4. Сформулюйте задачі, які вирішують криптоаналітикі?
5. Поясніть особливості алгоритмів шифрування із закритим та відкритим ключем.
6. Наведіть приклади режимів шифрування ECB та CBC.
7. Опишіть особливості стандартів шифрування DES, 3DES та AES.
8. Що являє собою алгоритм RSA?
9. Які особливості та можливості використання геш-функцій?
10. Як сформувати та перевірити цифровий підпис за алгоритмом Ель-Гамаля?
11. На яких принципах криптографічного захисту засновано сучасні стандарти цифрового підпису?
12. Яким чином відбувається формування закритих ключів за алгоритмом Диффі-Хелмана?

РОЗДІЛ 4. ЗАХИСТ ДАНИХ В IP-МЕРЕЖАХ

4.1. Особливості мережевих технологій захисту даних

Територіальне розміщення комп'ютерів у мережах накладає особливі вимоги до архітектури систем захисту. При цьому виникає питання про вибір місця для засобів захисту. Існують наступні три можливі варіанти архітектури системи захисту даних у комп'ютерній мережі.

- Розподілена архітектура.
- Централізована архітектура.
- Централізовано-розподілена архітектура.

Недоліком розподіленої архітектури є відсутність оперативного контролю над системою захисту в цілому. Такий варіант архітектури використовують у мережах з відсутністю явно визначеного центру. Через даний недолік можуть виникати проблеми з виявленням слабких місць у системі захисту та визначенні відповідальних осіб за витік конфіденційної інформації.

Централізована архітектура базується на спеціалізованому сервері безпеки, де розміщені усі засоби захисту. При цьому усі потоки інформації комп'ютерної мережі повинні контролюватись цим сервером. Такий варіант архітектури може перевантажувати канали зв'язку та призводити до затримок в обслуговуванні користувачів.

З метою подолання ускладнень, що виникають через недоліки цих двох варіантів архітектури, може бути обрана централізовано-розподілена архітектура, при якій функції захисту розподіляються наступним чином. Адміністрування, реєстрацію та контроль покладають на центральний сервер, а захист клієнтських комп'ютерів відбувається децентралізовано.

Основними причинами порушень захисту у комп'ютерних мережах є приховані канали, комп'ютерні віруси, троянські коні, люки та програмні закладки. Атаки зловмисників можуть бути пасивними, що мають за мету моніторинг інформації, або активними, що знищують або модифікують дані. Для боротьби з цими явищами використовують засоби захисту на різних ієрархічних рівнях архітектури комп'ютерних мереж від канального до прикладного.

Захист на канальному рівні у IP-мережах складно реалізувати, бо шлях пакетів від відправника до одержувача може пролягати по різних маршрутах, які не завжди визначені наперед. При цьому для повноцінного захисту необхідно захищати канали між усіма маршрутизаторами, що діють на маршруті пакетів. У мережах загального користування такий засіб є малопридатним. На канальному

або фізичному рівні можуть бути захищені тільки окремі канали у разі точно визначеного маршруту пакетів.

Захист на прикладному рівні пов’язаний з конкретними протоколами прикладного рівня. Для цього більшість протоколів прикладного рівня було розроблено в захищеному варіанті з виділенням для цього окремих портів. Найбільш відомі з цих протоколів наведено у таблиці 4.1.

Таблиця 4.1.

Захищені варіанти протоколів прикладного рівня стеку TCP/IP

Назва основного протоколу з номером TCP-порта	Назва захищеного протоколу з номером TCP-порта
HTTP port 80	HTTPS port 443
FTP ports 20, 21	FTPS port 990
SMTP port 25	SMTPS port 465
POP3 port 110	POP3S port 995
IMAP port 143	IMAPS port 993

Таким чином, на прикладному рівні для забезпечення захисту даних необхідно встановлювати спеціальні захищені варіанти протоколів для кожної окремої служби.

Найкращим варіантом захисту даних в IP-мережі прийнято вважати захист на міжмережевому (або мережевому) рівні. При цьому захищаються не окремі канали або протоколи, а мережа в цілому, бо такий захист охоплює всі протоколи і всі канали кожної конкретної мережі. Саме на цьому рівні побудовано комплекс засобів захисту під назвою IPSec (Internet Protocol Security).

Комплекс IPSec ґрунтується на промислових стандартах, що створюються робочою групою IP Security, яка працює у складі IETF (Internet Engineering Task Force – Відкрите міжнародне співтовариство, що займається розвитком мережі Інтернет). У цих розробках активну участь приймають провідні компанії розробників з усього світу такі як Microsoft та Cisco Systems.

4.2. Принципи побудови комплексу засобів IPSec

У відкритих каналах зв’язку можуть відбуватись прослуховування або модифікація даних. Такі дії можна розглядати як перший крок до більш небезпечних загроз. Наступним кроком може бути атака зловмисника, що призведе до відмови в обслуговуванні клієнтів, заміни співбесідника на зловмисника, модифікації або підміни файлів на серверах та частковому або повному руйнуванні системи захисту.

Метою створення комплексу засобів захисту даних IPSec було забезпечення надійного захисту від усіх перелічених шкідливих явищ. Цей захист розподіляється за двома наступними напрямками:

- захист IP-пакетів від прослуховування та модифікації;
- захист від можливих атак зловмисників.

Для забезпечення всебічного захисту використовують спеціальні протоколи з шифруванням даних, перевіркою автентичності та зміною шифрувальних ключів. Захист засновано на моделі “точка-точка” (end-to-end), що означає забезпечення безпеки під час передавання даних від служби відправника до служби одержувача. При цьому кожен з абонентів мережі забезпечує захист тільки на своєму боці за умов, що середовище передавання даних залишається незахищеним.

Такий підхід забезпечує надійність та гнучкість захисту з’єднань між комп’ютерами або службами у локальних мережах усіх типів, глобальних мережах, а також для випадків доступу до Інтернет з боку клієнтів приватних мереж.

Стратегія безпеки комплексу засобів IPSec базується на захисті від зовнішніх втручань у захищену мережу за допомогою фільтрації пакетів і автентифікації користувачів. Ці засоби захисту не виключають потреби у дотриманні встановлених правил користувачами. Останні повинні зберігати в таємниці паролі і не залишати без нагляду комп’ютери з відкритим доступом до захищених ресурсів.

За допомогою засобів IPSec слід обрати потрібний рівень захисту, щоб зловмисник був не в змозі дешифрувати захищені дані. Це забезпечується вибором певної структури політики безпеки.

Для правильного вибору структури політики безпеки рекомендована наступна послідовність дій [5].

- Оцінка необхідного рівня захисту даних, що пересилаються у мережі.
- Визначення місця знаходження даних, які підлягають захисту і усіх шляхів їх пересилання у мережі.
- Визначення небезпечних ділянок мережі, на яких існує можливість реалізації атак.
- Розробка плану забезпечення інформаційної безпеки для мережі в цілому.
- Створення та тестування засобів захисту для кожного з можливих напрямків пересилання даних, що підлягають захисту.

Точно визначених стандартів для рівнів безпеки даних не існує. Найчастіше орієнтуються на наступні три рівні.

- **Мінімальна безпека**, що відповідає тим комп'ютерам, які не приймають участь в обміні даними, що потребують захисту. На цих комп'ютерах засоби IPSec не використовуються.
- **Стандартна безпека**, до якої належать сервери, на яких зберігається конфіденційна інформація, що не потребує максимального рівня захисту. Для цих комп'ютерів слід обирати засоби IPSec, що не відповідають максимальному рівню безпеки і забезпечують можливість найбільш ефективної роботи для легальних користувачів.
- **Висока безпека**, до якої належать комп'ютери, на яких зберігаються найважливіші дані і існує висока ймовірність спроби несанкціонованого доступу до них. Для цих комп'ютерів може бути застосовано засоби IPSec у повній мірі для всіх потоків даних. І тих даних, що відправляються з цього комп'ютера, і тих, що приходять на цей комп'ютер.

Всі IP-пакети, що потрапляють на захищений комп'ютер, аналізуються фільтром, який відпрацьовує правила, що занесені в базу даних політики безпеки. Перевіряються IP-адреси та параметри протоколів вищих рівнів. На основі цього аналізу приймається рішення про наступні дії щодо кожного пакету. Ці дії можуть бути наступними:

- пакет знищується;
- пакет обминає засоби IPSec;
- пакет обробляється засобами IPSec.

Структурна схема комплексу IPSec показана на рисунку (рис. 4.1).

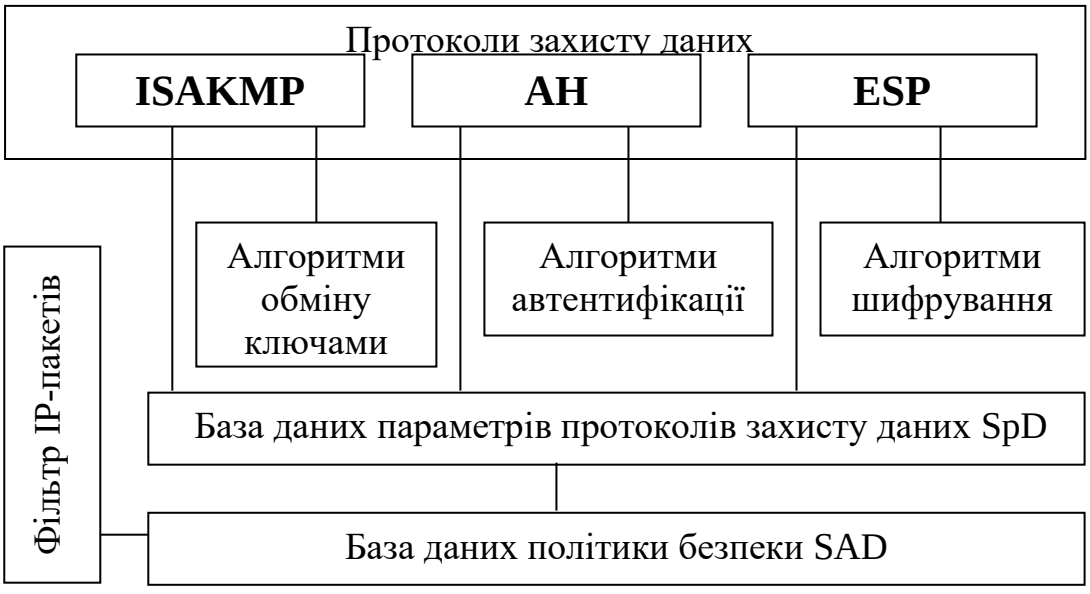

Рис. 4.1. Основні складові архітектури засобів захисту IPSec:

ISAKMP – протокол узгодження параметрів безпеки та управління ключами (Internet Security Association and Key Management Protocol); AH – автентифікуючий заголовок (Authentication Header); ESP – інкапсулюючий захист даних (Encapsulated Security Payload); SpD – Security policy Database; SAD – Security Association Database.

Для кожного з напрямків обміну даними можуть бути обрані свої параметри протоколів захисту, що зберігаються в базі даних SpD. На початку кожного захищеного сеансу обміну даними відбувається послідовність дій, яку називають SA (Security Association – домовленість про безпеку). При цьому вузол, що ініціює з'єднання, відправляє на вузол відповідача пакет за протоколом ISAKMP з пропозиціями щодо рівня безпеки, а саме про такі параметри:

- алгоритм шифрування (DES, 3DES або AES);
- алгоритм гешування (MD5, SHA-1 або SHA-2);
- метод автентифікації (сертифікат, загальний ключ або Kerberos);
- параметр для алгоритму Диффі-Хелмана (768 або 1024 бітів).

Вузол відповідача може або прийняти пропозиції про рівень безпеки або відправити від себе інші пропозиції. При цьому можливо до п'яти спроб домовитися про рівень безпеки.

Після узгодження перелічених параметрів відбувається формування ключів за алгоритмом Диффі-Хелмана. Для цього використовується той самий протокол ISAKMP.

Далі починається другий етап узгодження з використанням шифрувальних ключів. На цьому етапі сторони домовляються про параметри протоколів IPSec (AH та ESP) і в результаті обирають незалежні параметри безпеки для кожного з двох напрямків обміну даними. При цьому для кожного сеансу передавання даних у разі необхідності можуть формуватись нові ключі за алгоритмом Диффі-Хелмана.

Для кожного ключа під час формування встановлюється термін дії. У разі тривалості сеансу передавання даних 10000 секунд, а термін дії ключа встановлено тільки 2000 секунд, то протягом цього сеансу 5 разів будуть формуватись різні варіанти ключів. У разі розкриття якогось з ключів зловмисник отримає тільки п'яту частину інформації.

Кожному варіанту узгодження параметрів безпеки по кожному із напрямків передавання даних надається унікальний індекс SPI (Security Parameters Index – Індекс параметрів безпеки). Цей індекс пересилається у кожному захищеному пакеті. Згідно індексу одержувач обирає той чи інший варіант обробки даних.

4.3. Протоколи, що забезпечують безпеку передавання даних

У складі засобів IPSec використовується три протоколи.

- ISAKMP – протокол узгодження параметрів безпеки та управління ключами (Internet Security Association and Key Management Protocol);
- AH – протокол автентифікуючого заголовку (Authentication Header);
- ESP – протокол захисту даних за допомогою інкапсуляції (Encapsulated Security Payload).

Усі ці протоколи відносять до міжмережевого рівня стеку TCP/IP, який відповідає мережевому рівню моделі ISO/OSI.

Протокол ISAKMP, що описаний у RFC 2408, використовують в усіх процедурах узгодження параметрів безпеки та для формування ключів. Цей протокол іноді називають IKE (Internet Key Exchange) за RFC 2409, хоч фактично дані IKE вкладають після заголовку ISAKMP. Пакети цього протоколу пересилають у дейтаграмах протоколу транспортного рівня UDP (User Datagram Protocol) зі номерами 500 для портів відправника та одержувача.

Формат заголовка протоколу ISAKMP показано у таблиці 4.2.

Таблиця 4.2.

Структура заголовка протоколу ISAKMP

Найменування даних	Кількість біт даних	Значення даних
Initiator Cookie	64	Ідентифікуючі дані відправника
Responder Cookie	64	Ідентифікуючі дані одержувача
Next Payload	8	Код типу даних у цьому пакеті (перелік цих кодів надано у таблиці 4.3)
MjVer	4	Старша цифра версії ISAKMP (1 для RFC 2408)
MrVer	4	Молодша цифра версії ISAKMP (0 для RFC 2408)
Exchenge Tape	8	Код типу обміну даними (перелік цих кодів надано у таблиці 4.4)
Flags: E(Encription) C(Commit) A(Authentication only) None	1 1 1 5	1 – зашифровано; 1 – обмін ключами; 1 – перевіряється тільки цілісність; завжди нулі
Message ID	32	Ідентифікатор повідомлення ISAKMP
Length	32	Кількість байт у повідомленні ISAKMP, включаючи цей заголовок

За допомогою протоколу ISAKMP встановлюють домовленість про безпеку (SA – Security Association), а також підтверджують або ліквідують цю домовленість.

Перелік усіх варіантів повідомлень у протоколі ISAKMP із кодами, що фігурують у заголовку цього протоколу під назвою Next Payload, надано у таблиці 4.3.

Таблиця 4.3.

Перелік кодів типів даних у протоколі ISAKMP

Код типу даних	Значення коду даних
0	Немає
1	Домовленість про безпеку (SA - Security Association)
2	Пропозиція (P - Proposal)
3	Перетворення (T - Transform)
4	Обмін ключами (KE - Key Exchange)
5	Ідентифікація (ID - Identification)
6	Сертифікат (CERT - Certificate)
7	Запит сертифіката (CR - Certificate Request)
8	Геш-функція (HASH - Hash)
9	Цифровий підпис (SIG - Signature)
10	Відлік часу (NONCE - Nonce)
11	Оповіщення (N - Notification)
12	Видалення (D - Delete)
13	Ідентифікатор виробника (VID – Vendor ID)
14-127	Зарезервовано
128-255	Для приватного використання

Протокол ISAKMP дозволяє обмінюватись даними у зашифрованому вигляді або тільки з перевіркою цілісності, а також і без шифрування та перевірки.

Перелік усіх варіантів обміну даними за протоколом ISAKMP із кодами, що фігурують у заголовку цього протоколу під назвою Exchenge Tape, надано у таблиці 4.4.

Таблиця 4.4.

Перелік кодів при обміні даними за протоколом ISAKMP

Код	Значення коду при обміні даними
0	Немає
1	Базовий (Base)
2	Цілком захищений (Identity Protection)
3	Захищений тільки від перетворень (Authentication Only)
4	Активний (Aggressive)
5	Інформаційний (Informational)
6 - 31	Для майбутніх використань у ISAKMP
32 -239	Для спеціального використання
240 - 255	Для приватного використання

Структуру IP-пакета із повідомленням за протоколом ISAKMP зображено на рис. 4.1.

Рис. 4.1. Структура IP-пакета із повідомленням ISAKMP

У разі необхідності захисту цілісності (автентичності) повідомлення, яке не є конфіденційним, у засобах IPSec використовується протокол автентифікуючого заголовку AH.

Цей протокол не шифрує дані і тому вони є доступними для читання, але він утворює цифровий підпис за допомогою однієї з геш-функцій, яку обирають під час процедури узгодження параметрів безпеки SA.

Структура захищеного IP-пакета з використанням протоколу AH зображена на рис. 4.2.

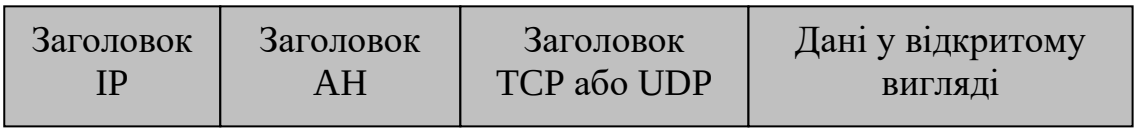

Рис. 4.2. Структура IP-пакета із автентифікуючим заголовком

Дія протоколу AH полягає у формуванні між IP-заголовком та заголовком транспортного рівня (TCP, UDP або іншого протоколу вищого рівня) ще одного заголовку. У цьому заголовку розміщується геш-функція, яка, з метою захисту цілісності всього IP-пакета, охоплює не тільки дані, але й усі заголовки,

включаючи IP-заголовок, крім полів TTL (час існування) та контрольної суми, що перераховуються у маршрутизаторах під час передавання пакету.

Структуру заголовку АН показано у таблиці 4.5.

Таблиця 4.5.

Структура автентифікуючого заголовку АН

Найменування даних	Кількість біт даних	Значення даних
Номер протоколу транспортного рівня	8	6 для TCP, 17 для UDP
Довжина заголовку	8	Кількість байт у заголовку АН
Резерв	16	
Індекс параметрів безпеки	32	Номер SPI, що відповідає даному варіанту SA (домовленості про безпеку)
Номер пакета	32	Номер IP-пакета від початку сеансу, починаючи з одиниці
Геш-функція	32*n	Значення n та алгоритм хешування відповідає даному варіанту SA

Протоколу автентифікуючого заголовку АН надано номер 51 за стандартом IETF. Цей номер встановлюється у IP-заголовку в полі номера протоколу транспортного рівня у разі використання протоколу АН.

Для збереження конфіденційності інформації під час передавання у комплексі засобів IPSec використовується протокол ESP, що захищає дані за допомогою шифрування.

Структура захищеного IP-пакета з використанням протоколу ESP зображена на рис. 4.3.

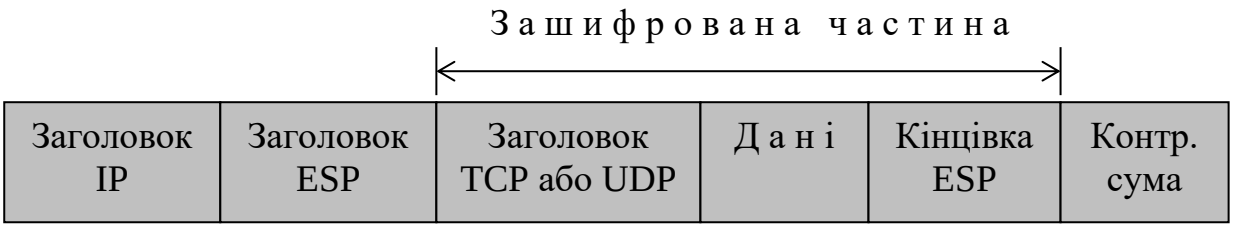

Рис. 4.3. Структура IP-пакета, що захищений за протоколом ESP

Протоколу ESP надано номер 50 за стандартом IETF. Цей номер встановлюється у IP-заголовку в полі номера протоколу транспортного рівня у разі захисту даних за допомогою протоколу ESP.

Структуру пакету ESP надано у таблиці 4.6.

Таблиця 4.6.

Структура пакета ESP протоколу

Найменування даних	Кількість біт даних	Значення даних
Індекс параметрів безпеки	32	Номер SPI, що відповідає варіанту SA (домовленості про безпеку)
Номер пакета	32	Номер IP-пакета від початку сеансу
Зашифрований зміст з доповненням	32*n+16	Доповнення повинно забезпечити ціле число блоків шифру
Довжина доповнення	8	Кількість байт доповнення
Номер протоколу транспортного рівня	8	6 для TCP, 17 для UDP
Контрольна сума	32*m	Формується після шифрування

Перші 64 біти пакету ESP (індекс параметрів безпеки та номер пакета), що являють собою ESP заголовок, не шифрують. Не шифрується також контрольна суму, бо вона формується вже після шифрування. Усі інші поля протоколу ESP, включаючи довжину доповнення і номер протоколу транспортного рівня, шифруються за алгоритмом, що відповідає узгодженому варіанту SA.

Протокол ESP можна використовувати ще й у режимі тунелювання. Цей режим також називають інкапсуляцією. Мета такого режиму полягає у збереженні в таємниці дійсних IP-адрес відправника і одержувача IP-пакету. Для зловмисника IP-адреса дає інформацію про те який комп'ютер бере участь у з'єднанні, а оскільки комп'ютери розподілені по робочих місцях з певними функціональними обов'язками, то це може сприяти розкриттю змісту повідомлення.

Структура захищеного IP-пакета з використанням протоколу ESP в режимі тунелювання зображена на рис. 4.4.

Рис. 4.4. Структура захищеного IP-пакета у режимі тунелювання за протоколом ESP

У новому заголовку IP-адреса відповідає серверу тунелювання захищеної мережі, а дійсна IP-адреса відправника або одержувача пакету знаходиться у зашифрованій частині.

Схему підключення захищеної мережі до відкритого каналу за допомогою сервера тунелювання зображено на рис. 4.5.

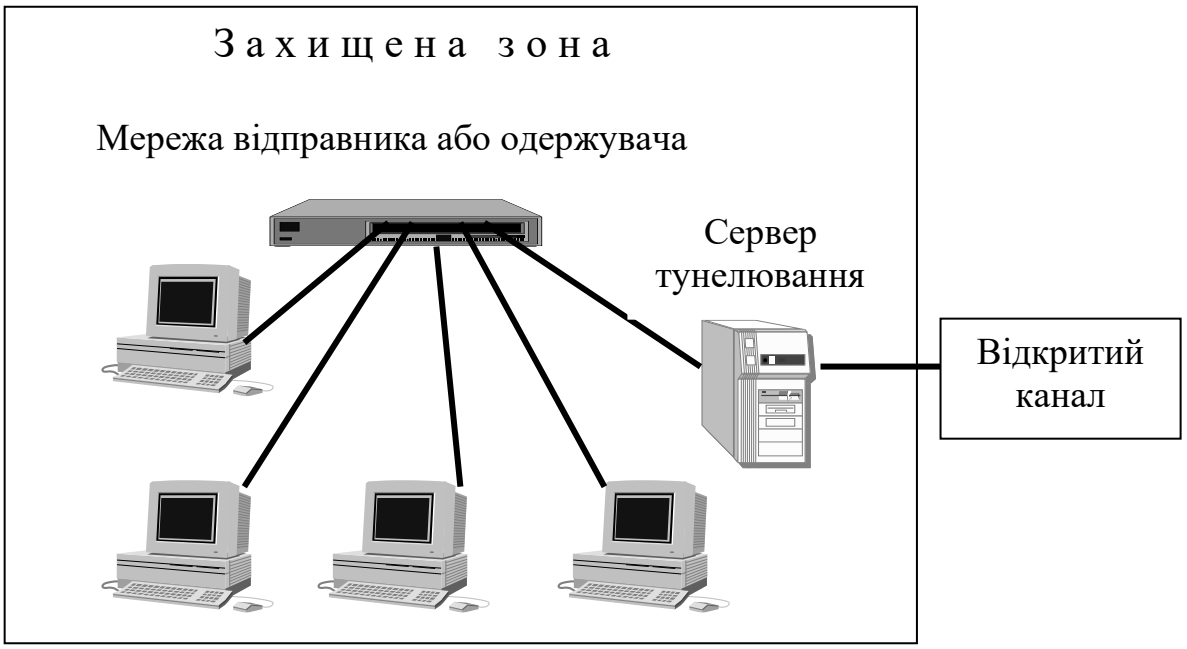

Рис. 4.5. Схема підключення захищеної мережі до відкритого каналу за допомогою сервера тунелювання

Можливо підключення за схемою, що зображено на рис.4.5, як з боку відправника, так і з боку одержувача, а можливо підключення за цією схемою мережі тільки з одного боку, а з другого — абонент зі одним комп'ютером.

4.4. Особливості політики IP-безпеки із засобами IPSec

Політика IPSec у кожному конкретному випадку базується на прийнятих в мережі правилах безпечної роботи. Кожна політика являє собою низку правил, що описують заборонені та/або дозволені дії користувачів щодо роботи у захищеній комп'ютерній мережі. Ця політика може розповсюджуватись на окремі групи комп'ютерів або на структурні підрозділи зі різними вимогами щодо рівня захисту даних.

Під час створення чи коригування політики для IPSec необхідно враховувати наступне.

- Політика IPSec, що призначається для домену, перетворює політику тільки на тих комп'ютерах, що належать даному домену.
- Групова політика має вищий пріоритет ніж політика доменів.
- Правила призначення групової політики може бути обрано або так, щоб зміна політики вищої за ієрархією групи перетворювала політику для усіх нижчих груп, або так, щоб зміна політики для групи, що нижче за ієрархією, перетворювала політику для вищих за ієрархією груп.

Правила, з яких складається політика IPSec, визначають яким чином і за яких умов відбувається захист з'єднань. Цей захист виконується виходячи з IP адрес відправника та одержувача. Кожне правило включає список фільтрів для IP-пакетів та перелік дій з цими пакетами, включаючи методи автентефікації та параметри тунелювання. Кожна політика може складатись з довільної кількості правил, при цьому не всі правила можуть бути активними. Одна політика може мати різні правила щодо різних з'єднань. Наприклад, для внутрішніх і зовнішніх з'єднань можуть бути призначені різні правила.

Кожен фільтр може перевіряти наступні параметри.

- IP-адреси комп'ютерів або мереж відправника та одержувача. Для випадків призначення адреси мережі необхідним атрибутом є маска мережі.
- Номери протоколів транспортного рівня. У разі замовчання дія фільтра розповсюджується на всі протоколи.
- Номери портів для протоколів TCP та UDP. У разі замовчання дія фільтра розповсюджується на всі номери портів.

Дія фільтрів полягає у виборі одного з трьох рішень по кожному з'єднанню:

- з'єднання забороняється;
- з'єднання захищається;
- з'єднання не підлягає захисту.

Останній варіант можливо використовувати для тих протоколів вищих рівнів, які мають власні засоби захисту.

Для автентифікації учасників обміну даними у засобах IPSec передбачено наступні три методи.

Для клієнтів, які підтримують протокол Kerberos, можна процедуру автентифікації здійснювати за допомогою сервера Kerberos. Цьому методу слід надавати перевагу перед двома наступними.

Для клієнтів, які не підтримують протокол Kerberos, автентифікація може бути здійснена за допомогою сертифікату відкритого ключа. Для цього

необхідно мати доступ хоч до одного центру сертифікації (Certificate Authority, CA).

У випадках неможливості автентифікації учасників обміну даними жодним з двох наведених вище методів можна скористатись спільним ключем. Цей ключ необхідно заздалегідь пересилати у закритому вигляді та зберігати у таємниці. Через те, що у засобах IPSec цей ключ зберігається без використання засобів захисту, користування таким ключем повинно бути обмеженим у часі.

Під час розробки політики IP-безпеки необхідно у кожному випадку відшукувати найкраще співвідношення між забезпеченням зручного доступу до інформаційних ресурсів для більшості легальних користувачів та захистом критичної частини інформації від несанкціонованого доступу.

4.5. Поглиблений аналіз потоків даних у IP-мережах

Широке розповсюдження комп'ютерних вірусів і небажаних листів від невідомих рекламодавців у мережі Інтернет утворюють паразитне навантаження на програмно-технічні засоби IP-мереж. Деякі шкідливі програми створюють потоки даних, які повністю блокують доступ до потрібних ресурсів.

З метою забезпечення нормальної роботи користувачів в умовах впливу цих завад, у мережах встановлюють засоби, що відслідковують інформаційні потоки більш детально та поглиблено, ніж це передбачено у комплексі IPSec.

Для боротьби зі переліченими паразитними явищами недостатньо аналізу IP-пакетів, а треба досліджувати повідомлення в цілому, що можливо тільки на вищому рівні архітектури комп'ютерних мереж. Серед багатьох програмних засобів, що призначені для боротьби з вірусами та іншими паразитними явищами, у кожному разі обирають найпридатніший, виходячи з конкретних умов власного застосування.

Виходячи зі нашого досвіду щодо створення та адміністрування корпоративної комп'ютерної мережі будівельного комплексу України, для досягнення вище вказаної мети ми використовували як існуючи програмні засоби, так і власні розробки. Далі наведемо їх перелік із стислим описом.

Для захисту від комп'ютерних вірусів ми обрали програму Clamav (Clam Anti Virus), яка використовується протягом багатьох років широко відомими фірмами і розповсюджується безкоштовно згідно ліцензії GNU (General Public License). Ця програма завжди доступна у вигляді відкритого коду на мові програмування C в Інтернеті за адресою www.clamav.net. Володіє цією

програмою фірма Sourcefire, яка є всесвітньо відомим розробником антивірусних програм.

Програма Clamav сканує файли, що пересилаються електронною поштою, на поштовому шлюзі під час передавання. Антивірусна база даних регулярно доповнюється та поновлюється автоматично кілька разів на добу. У цій базі налічується близько півмільйона комп'ютерних вірусів, троянів та черв'яків. Програма аналізує архівні файли у форматах Zip, RAR, Tar, Gzip, Bzip2, OLE2, Cabinet, CHM, BinHex, SIS та інших, а також програмні і замасковані файли, включаючи документи MS Office, MacOffice, HTML, RTF та PDF. Управління програмою у системі UNIX здійснюється за допомогою команд з консолі.

Для захисту від небажаних повідомлень рекламного характеру, які прийнято називати спамом, ми обрали широко відому програму Spam Assassin. У конфігураційному файлі цієї програми знаходиться набір правил, які дозволяють виявляти небажані повідомлення електронної пошти і відфільтровувати їх за відомим принципом чорного та білого списків. Є можливість автоматичного поновлення конфігураційного файлу із централізованих джерел, а також доповнювати набір правил за власним розсудом.

Для захисту мережі від порушників, які знаходяться в межах своєї мережі та не легалізовано розповсюджують повідомлення рекламного характеру, розроблено спеціальну програму для виявлення та блокування таких порушників. Через таких порушників спеціальні служби можуть заблокувати доступ усіх користувачів, які користуються даною IP-адресою, до мережі Інтернет. Ця програма підраховує кількість відправлень електронної пошти за певні проміжки часу для кожного відправника та автоматично блокує TCP-порт 25 даному відправнику у разі перевищення встановленого значення кількості поштових відправлень за 10 хвилин. Вихідний текст головного модуля цієї програми, яку розробив Тарасюк Д.М., надано у додатку 2.

Висновки

1. Існують три можливі варіанти архітектури систем захисту даних у комп'ютерних мережах: розподілена, централізована та централізовано-розподілена архітектура. Недоліком розподіленої архітектури є відсутність оперативного контролю над системою захисту в цілому. Централізована архітектура може призводити до затримок в обслуговуванні користувачів. З метою подолання ускладнень, що виникають через недоліки двох перших варіантів архітектури, обирають централізовано-розподілену архітектуру,

при якій адміністрування, реєстрацію та контроль покладають на центральний сервер, а захист клієнтських комп'ютерів відбувається децентралізовано.

2. Основними причинами порушень захисту у комп'ютерних мережах є приховані канали, комп'ютерні віруси, троянські коні, люки та програмні закладки. Атаки зловмисників можуть бути пасивними, що мають за мету моніторинг інформації, або активними, що знищують або модифікують дані. Для боротьби з цими явищами використовують засоби захисту на різних ієрархічних рівнях архітектури комп'ютерних мереж від канального до прикладного.
3. Найкращим варіантом захисту даних в IP-мережі прийнято вважати захист на міжмережевому (або мережевому) рівні. При цьому захищаються не окремі канали або протоколи, а мережа в цілому, бо такий захист охоплює всі протоколи і всі канали кожної конкретної мережі. Саме на цьому рівні побудовано комплекс засобів захисту під назвою IPSec, який ґрунтується на промислових стандартах, що створюються робочою групою IP Security, яка працює у складі IETF.
4. У відкритих каналах зв'язку можуть відбуватись прослуховування або модифікація даних. Такі дії можна розглядати як перший крок до більш небезпечних загроз. Наступним кроком може бути атака зловмисника, що призведе до відмови в обслуговуванні клієнтів, заміни співбесідника на зловмисника, модифікації або підміни файлів на серверах та частковому або повному руйнуванні системи захисту.
5. Для забезпечення всебічного захисту у IPSec використовують спеціальні протоколи з шифруванням даних, перевіркою автентичності та зміною шифрувальних ключів. Захист забезпечує безпеку під час передавання даних від служби відправника до служби одержувача. При цьому середовище передавання даних залишається незахищеним.
6. Стратегія безпеки комплексу засобів IPSec базується на захисті від зовнішніх втручань у захищену мережу за допомогою фільтрації пакетів і автентифікації користувачів. Ці засоби захисту не виключають потреби у дотриманні встановлених правил користувачами. Останні повинні зберігати в таємниці паролі і не залишати без нагляду комп'ютери з відкритим доступом до захищених ресурсів.
7. Для кожного з напрямків обміну даними можуть бути обрані свої параметри протоколів захисту, що зберігаються в базі даних. На початку кожного захищеного сеансу обміну даними відбувається послідовність

дій, яку називають SA (Security Association – домовленість про безпеку). При цьому вузол, що ініціює з'єднання, відправляє на вузол відповідача пакет з пропозиціями щодо рівня безпеки, у якому пропонуються конкретні параметри алгоритмів захисту.

8. Кожному варіанту узгодження параметрів безпеки по кожному із напрямків передавання даних надається унікальний індекс SPI (Security Parameters Index – Індекс параметрів безпеки). Цей індекс пересилається у кожному захищеному пакеті. Згідно індексу одержувач обирає той чи інший варіант обробки даних.
9. У складі засобів IPSec використовується три протоколи: протокол узгодження параметрів безпеки та управління ключами (ISAKMP – Internet Security Association and Key Management Protocol), протокол автентифікуючого заголовку (AH – Authentication Header) та протокол захисту даних за допомогою інкапсуляції (ESP – Encapsulated Security Payload). Усі ці протоколи відносять до міжмережевого рівня стеку TCP/IP, який відповідає мережевому рівню моделі ISO/OSI.
10. Протокол ISAKMP використовують в усіх процедурах узгодження параметрів безпеки та формування ключів. Цей протокол іноді називають IKE (Internet Key Exchange) за RFC 2409, хоч фактично дані IKE вкладають після заголовку ISAKMP. Пакети цього протоколу пересилають у дейтаграмах протоколу транспортного рівня UDP (User Datagram Protocol) зі номерами 500 для портів відправника та одержувача.
11. Дія протоколу AH полягає у формуванні між IP-заголовком та заголовком транспортного рівня (TCP, UDP або іншого протоколу вищого рівня) ще одного заголовку. У цьому заголовку розміщується геш-функція, яка, з метою захисту цілісності всього IP-пакета, охоплює не тільки дані, але й усі заголовки, включаючи IP-заголовок, крім полів TTL (час існування) та контрольної суми, що перераховуються у маршрутизаторах під час передавання пакету. Цей протокол не шифрує дані і тому вони є доступними для читання, але він утворює цифровий підпис за допомогою однієї з геш-функцій, яку обирають під час процедури узгодження параметрів безпеки SA.
12. Для збереження конфіденційності інформації під час передавання у комплексі засобів IPSec використовується протокол ESP, що захищає дані за допомогою шифрування. Протоколу ESP надано номер 50 за стандартом IETF. Цей номер встановлюється у IP-заголовку в полі номера протоколу транспортного рівня у разі захисту даних за допомогою протоколу ESP.

Перші 64 біти пакету ESP (індекс параметрів безпеки та номер пакета), що являють собою ESP заголовок, не шифрують. Не шифрується також контрольна суму, бо вона дописується в кінцівку вже після шифрування. Усі інші поля протоколу, включаючи довжину доповнення і номер протоколу транспортного рівня, шифруються.

13. Політика IPSec у кожному конкретному випадку базується на прийнятих в мережі правилах безпечної роботи. Кожна політика являє собою низку правил, що описують заборонені та/або дозволені дії користувачів щодо роботи у захищеній комп'ютерній мережі. Ця політика може розповсюджуватись на окремі групи комп'ютерів або на структурні підрозділи зі різними вимогами щодо рівня захисту даних.
14. Правила, з яких складається політика IPSec, визначають яким чином і за яких умов відбувається захист з'єднань. Цей захист виконується виходячи з IP адрес відправника та одержувача. Кожне правило включає список фільтрів для IP-пакетів та перелік дій з цими пакетами, включаючи методи автентефікації та параметри тунелювання. Кожна політика може складатись з довільної кількості правил, при цьому не всі правила можуть бути активними. Одна політика може мати різні правила щодо різних з'єднань. Наприклад, для внутрішніх і зовнішніх з'єднань можуть бути призначені різні правила.
15. Для автентифікації учасників обміну даними у засобах IPSec передбачено три методи. Для клієнтів, які підтримують протокол Kerberos, процедуру автентифікації рекомендовано здійснювати за допомогою сервера Kerberos. Цьому методу слід надавати перевагу. Для клієнтів, які не підтримують протокол Kerberos, автентифікація може бути здійснена за допомогою сертифікату відкритого ключа. У випадку неможливості автентифікації жодним з двох наведених вище методів можна скористатись спільним ключем, який необхідно заздалегідь пересилати у закритому вигляді та зберігати у таємниці. Через те, що у засобах IPSec цей ключ зберігається без використання засобів захисту, користування таким ключем повинно бути обмеженим у часі.
16. Під час розробки політики IP-безпеки необхідно у кожному випадку відшукувати найкраще співвідношення між забезпеченням зручного доступу до інформаційних ресурсів для більшості легальних користувачів та захистом критичної частини інформації від несанкціонованого доступу.
17. Широке розповсюдження комп'ютерних вірусів і небажаних листів від невідомих рекламодавців у мережі Інтернет утворюють паразитне

навантаження на програмно-технічні засоби IP-мереж. Деякі шкідливі програми здатні створюють потоки даних, які повністю блокують доступ користувачів до потрібних ресурсів. Для боротьби зі переліченими паразитними явищами недостатньо аналізу IP-пакетів, а треба досліджувати повідомлення в цілому, що можливо тільки на вищому рівні архітектури комп'ютерних мереж. Серед багатьох програмних засобів, що призначені для боротьби з вірусами та іншими паразитними явищами, у кожному разі обирають найпридатніший, виходячи з конкретних умов власного застосування.

Запитання та завдання для самоперевірки

1. Порівняйте та проаналізуйте можливі варіанти архітектури систем захисту даних у комп'ютерних мережах.
2. Назвіть причини порушень захисту даних у комп'ютерних мережах.
3. На якому рівні архітектури комп'ютерних мереж створення систем захисту вважається найдоцільнішим?
4. Які небезпечні дії зловмисників можуть відбуватись у відкритих каналах зв'язку?
5. Які засоби у комплексі IPSec використовуються для забезпечення всебічного захисту даних під час передавання?
6. На чому базується стратегія безпеки комплексу засобів IPSec?
7. Що являє собою процедура встановлення домовленості про безпеку SA (Security Association)?
8. Для чого надається індекс SPI (Security Parameters Index – Індекс параметрів безпеки)?
9. Які три протоколи використовується у складі засобів IPSec?
10. Яке призначення протоколу ISAKMP?
11. Який принцип дії протоколу AH?
12. Для чого використовується протокол ESP?
13. З яких правил складається політика IPSec?
14. Які варіанти автентифікації учасників обміну даними передбачено у засобах IPSec?
15. Виходячи з яких альтернатив слід обирати рівень захисту даних у IP-мережі?

Список літератури

1. *ДСТУ 3396.0-96.* Захист інформації. Технічний захист інформації. Основні положення. – Чинний з 01.01.1997.
2. *НД ТЗІ 1.1-002-99.* Загальні положення щодо захисту інформації в комп'ютерних системах від несанкціонованого доступу. – Чинний з 28.04.1999. – К.: ДСТСЗІ СБ України, 1999. – 14 с.
3. *НД ТЗІ 2.5-004-99.* Критерії оцінки захищеності інформації в комп'ютерних системах від несанкціонованого доступу. – Чинний з 28.04.1999. – К.: ДСТСЗІ СБ України, 1999. – 53 с.
4. *НД ТЗІ 1.1-003-99.* Термінологія в галузі захисту інформації в комп'ютерних системах від несанкціонованого доступу. – Чинний з 28.04.1999. – К.: ДСТСЗІ СБ України, 1999. – 24 с.
5. *Steve Friedl.* An Illustrated Guide to IPsec. [Електронний ресурс] http://unixwiz.net/techtips/iguide-ipsec.html
6. *Гайна Г.А.* Основи проектування баз даних. Навчальний посібник. – К.: Кондор, 2008. – 200 с.
7. *Вишняков В.М.* Захист даних в інформаційних системах. Навчальний посібник. – К.: КНУБА, 2010. – 128 с.
8. *Вишняков В.М.* Принципи побудови комп'ютерних мереж. Навчальний посібник. – К.: КНУБА, 2022. – 128 с.
9. *ДСТУ 3396.1-96.* Захист інформації. Технічний захист інформації. Порядок проведення робіт. – Чинний з 01.07.1997.
10. *C.E.Shannon.* The Communication Theory of Secrecy Systems // Bell System Technical Journal. – 1949 – v.28, n.4 – C.654-715
11. *D.Kahn.* The Codebreakers: The Story of Secret Writing. – New York: Macmillan Publishing Co., 1967. – 1164 с.
12. *Bruce Schneier.* Applied Cryptography: Protocols, Algorithms, and Source Code in C, Second Edition. – New York: Wiley, 1995. – 784 с.
13. *D.Kahn.* Seizing the Enigma. – Boston: Houghton Mifflin Co., 1991. – 640 с.
14. *ISO DIS 8732.* "Banking Key Management (Wholesaled)" Association for Payment Clearing Services, London, Dec 1987.
15. *Bruce Schneier.* Schneier on Security. – New York: Wiley, 2008. – 336 с.
16. *R.L.Rivest, A.Shamir, L.M.Adleman.* A Method of Obtaining Digital Signatures and Publie-Key Cryptosystem // Communications of the ASM, System Technical Journal. – 1978 – v.21, n.2 – C.120-126.

17. *T.ElGamal.* On Computing Logarithms Over Finite Fields // Advanced in Criptology CRIPTO'85 Proceedings, Springer-Verlag – 1986 – С. 396-402.
18. *V.S.Miller.* Use of Elliptic Curve in Cryptography // Advanced in Criptology CRIPTO'85 Proceedings, Springer-Verlag – 1986 – С. 417-426.
19. *N.Coblitz.* Elliptic Curve in Cryptosystems // Mathematics of Computation – 1987 – v.48, n.177 – С.203-209.
20. *ДСТУ 4145-2002.* Інформаційні технології. Криптографічний захист інформації. Цифровий підпис, що ґрунтується на еліптичних кривих. Формування та перевіряння. – Чинний з 01.07.2003.
21. *W.Diffie, M.E.Hellman.* New Direction in Cryptography // IEEE Transactions on Information Theory. – 1976 – v.IT-22, n.6 – С. 644-654.

Тексти програм у псевдокоді для реалізації алгоритму AES

```
/*************************************************************/
/*           Назви функцій та позначення змінних             */
/*  Cipher() – функція зашифровування                        */
/*  Nb – довжина блоку у 32-бітних словах Nb=4               */
/*  Nk - довжина ключа у 32-бітних словах Nk = 4, 6 або 8    */
/*  CiherKey – секретний ключ довжиною Nk*4 байт             */
/*  Nr - кількість раундів, Nr = 10, 12 або 14, Nr = Nk+6    */
/*  RoundKey – результат перетворення CipherKey для раунду   */
/*  KeyExpansion – функц.перетворення CipherKey у RoundKey   */
/*  w[] – масив для перетворених ключів до всіх раундів      */
/*  state – масив байтів на 4 рядки та Nb колонок            */
/*  AddRoundKey() – операція XOR між state та  RoundKey      */
/*  S-box – таблиця замін байтів у функції KeyExpansion      */
/*  Rcon[] – масив зі 32 бітів, що незмінні для раунду       */
/*  SubBytes() – перетворення масиву замінами через S-box    */
/*  ShiftRows() – циклічний зсув у 3-х нижніх рядках масиву  */
/*************************************************************/

Cipher(byte in[4*Nb], byte out[4*Nb], word w[Nb*(Nr+1)])
begin
   byte state[4,Nb]
   state = in
   AddRoundKey(state, w[0, Nb-1])
   for round = 1 step 1 to Nr-1
       SubBytes(state)
       ShiftRows(state)
       MixColumns(state)
       AddRoundKey(state, w[round*Nb, (round+1)*Nb-1])
   end for
   SubBytes(state)
   ShiftRows(state)
   AddRoundKey(state, w[Nr*Nb, (Nr+1)*Nb-1])
   out = state
end

KeyExpansion(byte key[4*Nk], word w[Nb*(Nr+1)], Nk)
begin
     word temp
     i=0
```

```
     while(i < Nk)
        w[i] = word(key[4*i], key[4*i+1], key[4*i+2], key[4*i+3])
      i = i+1
     end while
     i = Nk
     while ( i < Nb * (Nr+1))
          temp = w[i-1]
          if (i mod Nk = 0)
              temp = SubWord(RotWord(temp)) xor Rcon[i/Nk]
          else if (Nk > 6 and i mod Nk = 4)
            temp = SubWord(temp)
          end if
          w[i] = w[i-Nk] xor temp
          i = i+1
     end while
end
/*************************************************************/
/*  InvCipher() - функція розшифровування                    */
/*************************************************************/
InvCipher(byte in[4*Nb], byte out[4*Nb], word w[Nb*(Nr+1)])
begin
     byte state[4,Nb]
     state = in
     AddRoundKey(state, w[Nr*Nb, (Nr+1)*Nb-1])
     for round = Nr-1 step -1 downto 1
        InvShiftRows(state)
        InvSubBytes(state)
        InvAddRoundKey(state, w[round*Nb, (round+1)*Nb-1])
        InvMixColumns(state)
     end for
     InvShiftRows(state
     InvSubBytes(state
     InvAddRoundKey(state, w[Nr*Nb, (Nr+1)*Nb-1])
     out = state
end
```

Додаток 2

Текст PERL-програми для виявлення та блокування порушників

```
#!/usr/bin/perl -w

my $LOGFILE = "/var/log/maillog";
my $PERIOD = 10;          # minutes
my $INNERMAXEVENTS = 29; # писем, не более чем за 1 минуту или в
среднем за $PERIOD минут
my $OUTERMAXEVENTS = 15;
my $INNERIPFWTABLE = 2;
my $OUTERIPFWTABLE = 3;
my $OUTERIPFWTABLE2 = 4;
my $IPFW = "/sbin/ipfw";
my $MAIL = "/usr/bin/mail";
my $MAILRECIPIENT = 't@dim.kiev.ua';
my $MAILCC = 'Vladimir@ndiasb.kiev.ua';
my $MAILBC = '';
my $SPAMOFFLOG = "/var/log/spamoff.log";
my $LOCKFILE = "/var/run/spamoff.pid";
my $NSLOOKUP = "/usr/bin/nslookup";
my $WHOIS = "/usr/bin/whois";

# -<
#  Монитор спам-активности хостов.
#
#  Алгоритм. Лог файл для sendmail'a, /var/log/maillog, каждую
секунду
#  вычитывается в поиске новых поступивших записей об отправляемых
мейлах.
#  Для каждого IP адреса хоста, с которого отправляется почта,
#  за каждую минуту, но не позже $PERIOD минут, запоминается
количество
#  попыток отправить почту. Различаются IP адреса как
принадлежащие
#  локальной сети (LAN), так и внешнему интернету (WAN). Для LAN и
WAN
#  хостов используется раздельный учет.
#
#  Если за $PERIOD минут в среднем скорость получения почтовых
сообщений
#  превысит значение $INNERMAXEVENTS для LAN или $OUTERMAXEVENTS
для WAN,
```

```
#  ИЛИ за текущую минуту будет превышение 3/4 одного из этих
значений
#  (в зависимости от LAN/WAN хоста-источника), то
#
#       1. Соответствующий IP будет внесен в таблицу ipfw
#          номер $INNERIPFWTABLE для LAN-хоста или
#          таблицу номер $OUTERIPFWTABLE для WAN-хоста.
#
#       2. На емейл адреса $MAILRECIPIENT, $MAILCC, $MAILBC будут
#          высланы соответствующие информационные письма.
#
#  Программа не применяет никаких прямых действий, которые могут
прекратить
#  поток спам писем на сервер. Программа только регистрирует
хосты-источники
#  и, в зависимости от того, изнутри или извне источник -
заполняет
#  соответствующие таблицы ipfw, правила которого реализуют ту или
иную
#  политику. Например, среди правил ipfw могут быть такие:
#
#  (1) add deny tcp from table($INNERIPFWTABLE) to any 25
#  (2) add pipe 100 tcp from table($OUTERIPFWTABLE) to me 25
#  (3) pipe 100 config bw 16Kbit/s queue 5Kbytes
#  (4) add fwd 127.0.0.1,25 tcp from 10.0.0.0/8 to any 25
#
#  Правило (1) блокирует отправку почты, проходящую через
#  сервер в любых направлениях со спам-хостов изнутри сети.
#  Правило (2) ограничивает пропускную полосу для спам-хостов
#  из внешней сети Интернет. Правило (3) определяет такую полосу
#  Правило (4) задает, что все попытки отправить почту напрямую к
#  другим серверам, но которые проходят через данный сервер как
#  шлюз, должны быть перенаправлены к локальному sendmail'у
(который
#  должен быть соответствующим образом настроенным).
#
#  Newsyslog обычно в полночь сжимает и ротирует лог sendmail'а.
#  После ротации spamoff теряет поток сообщений о передаваемой
#  почте, получаемой через log-файл. Поэтому в конфигурационном
#  файле /etc/newsyslog.conf для /var/log/maillog в поле /PID_FILE
#  следует указать PID файл spamoff. Spamoff сохраняет свой PID
#  в файле $LOCKFILE. Тогда newsyslog будет отправлять сигнал
#  HUP для spamoff, и он восстановит поток сообщений.
#
```

```
#  По сигналу USR1 spamoff выводит в файл $SPAMOFFLOG счетчики для
#  всех известных на тот момент хостов.
#
#  По сигналу INT spamoff завершает свою работу.
# >-                                    Автор: Тарасюк Д.М.

use strict;
use Time::Local;
use IO::Seekable;

my %Activ;

# Проверка на иденичность процесса

my $mylockfile = 0;

END {
   # При выходе уничтожить pid-файл
   unlink $LOCKFILE if $mylockfile;
}

if (-s $LOCKFILE) {
   print "$0: one copy is worked now (PID can be found in
\"$LOCKFILE\")\n";
   exit
}

open(LOCK, ">", $LOCKFILE)
   or die "$0: cannot create lock file \"$LOCKFILE\": $!";
print LOCK "$$\n";
close LOCK;
$mylockfile = 1;

# Обработка сигналов по мере их поступления

sub handler {
   my $sig = $_[0];
   open (LOG, ">>", $SPAMOFFLOG)
       or die "$0: cannot open log file \"$SPAMOFFLOG\": $!";
   print LOG "\n*** [" . localtime() . "] SIG$sig ***\n" ;
   if (uc($sig) eq "HUP") {
       # Была ротация $LOGFILE - его нужно переоткрыть
       print LOG "Log file \"$LOGFILE\" is reopened\n";
       close FILE;
```

```
   }
   elsif (uc($sig) eq "USR1") {
       # Запрошена выдача в $SPAMOFFLOG счетчиков для всех..
       #.. зарегистрированных хостов
       my $text;
       my $num = 0;
       my ($innerflag, $lasttime, $pos, @acts);
       foreach my $ip (keys %Activ) {
           ($innerflag, $lasttime, $pos, @acts) = split(':',
$Activ{$ip});
           $text = "$1 %s %-15s Counters:"
               if localtime($lasttime * 60) =~
m{^\S+\s+(\S+\s+\d+\s+\d+:\d+):};

           my $events = 0;
           for (my $i = 0; $i < $PERIOD; ++$i) {
               $events += $acts[$pos];
               $text .= $acts[$pos--];
               $pos = $PERIOD - 1 if $pos < 0;
               $text .= $i == $PERIOD - 1 ? "=$events" : "+";
           }

           ++$num;
           printf LOG "%7u) $text\n", $num, $innerflag ? "LAN" :
"WAN", $ip;
       }
   }
   else {
       # Выход - там все файлы и закроются
       exit 0;
   }
   close LOG;
}

foreach my $sig ('HUP' , 'INT' , 'QUIT',
                 'TRAP', 'IOT' , 'BUS' ,
                 'USR1', 'SEGV', 'USR2',
                 'PIPE', 'TERM',  'FPE',
                )
{
   $SIG{$sig} = "handler";
}

xxxxx_REOPEN_AFTER_ROTATION_xxxxx:
```

```
# print "xxxxx_REOPEN_AFTER_ROTATION_xxxxx\n";

open(FILE, "< $LOGFILE") || die "*** $! ***\nFile does not
exist?\n";

my $fs = (stat(FILE))[7];
seek(FILE, $fs, SEEK_SET) or die "$!";

my $fd;
my $text;
my $countertodelold = 0;

while (1) {
   while (($fd = fileno(FILE)) && ($text = <FILE>)) {
       overlook($text);
   }
   unless (defined $fd) {
       sleep 1;
       goto xxxxx_REOPEN_AFTER_ROTATION_xxxxx;
   }
   sleep(1);
   ++$countertodelold;
}

sub overlook {
   my $text = $_[0];
   my $relay;
   my $time;
   my $ip;
   if ($text =~ m{^
            (\S+\s+\d+\s+\d+:\d+:\d+)\s+ \S+ \s+ \S+\[\d+\]:\s+
            .+
            relay=(?:(\S+) \s+)?
            \[(\d+\.\d+\.\d+\.\d+)\] (.*) $
            }ix) {
       ($time, $relay, $ip) = ($1, $2, $3);
       my $rest = $4;
       unless ($rest =~ m{stat=}) {
           my $timeinsec = backtime($time);
           check($ip, $time, $timeinsec);

           # Периодическое удаление информации о старых IP
           if ($countertodelold > $PERIOD) {
```

```
                $countertodelold = 0;
                deloldrecords($timeinsec);
            }

# print "$timeinsec($relay)[$ip]: $text\n";

        }
    }
}

sub deloldrecords {
    my $timeinsec = $_[0];
    my $lasttime;
    foreach my $ip (keys %Activ) {
        (undef,  $lasttime, undef) = split(':', $Activ{$ip});
        if ($lasttime < $timeinsec / 60 - 10) {
            delete($Activ{$ip});
        }
    }
}

sub takesteps($$$\@$) {
    my ($ip, $innerflag, $time, $pacts, $pos) = @_;

    ## Блокировка на уровне ipfw

    my $tbn = $OUTERIPFWTABLE;
    $tbn = $INNERIPFWTABLE if $innerflag;

    # Проверка, если ли $ip уже в списке блокируемых

    open(CMD, "$IPFW table $tbn list |")
        or die "$0: Cannot get list of ipfw table No $tbn: $!";
    my $ipexistflag = 0;
    while (<CMD>) {
        if (m{^$ip/32\s+\d+}) {
            $ipexistflag = 1;
            last;
        }
    }
    close CMD;

    $tbn = $OUTERIPFWTABLE2 if (!$innerflag && $ipexistflag);
```

```
    # Блокировать источник писем. Ipfw должен иметь правило
блокировки, ..
    #..например "deny tcp from table($tbn) to me 25"

    unless ($ipexistflag) {
        open(CMD, "$IPFW table $tbn add $ip |")
            or die "$0: Cannot add $ip to ipfw table No $tbn: $!";
        while(<CMD>) {;}
        close CMD;
    }

    # Безопасная отсылка email-уведомления о зарегистрированном
источнике спама

    my $counters = "";
    my $events = 0;
    my $doirep = 0;
    for (my $i = 0; $i < $PERIOD; ++$i) {
        if ($pacts->[$pos]) {
            $doirep = 0;
        }
        else {
            ++$doirep;
        }
        $events += $pacts->[$pos];
        $counters .= $pacts->[$pos--];
        $pos = $PERIOD - 1 if $pos < 0;
        $counters .= $i == $PERIOD - 1 ? "=$events" : "+";
    }

    my $subject = "SPAMOFF: $ip is blocked"
        . ($ipexistflag ? " again" : "")
        . " ($events mails/" . ($PERIOD - $doirep) . "min)";

    my $body = "$time: $counters";
    my $mailopts = "-s '$subject'";
    $mailopts .= " -c $MAILCC" if $MAILCC;
    $mailopts .= " -b $MAILBC" if $MAILBC;
    $mailopts .= " $MAILRECIPIENT";

    my $pid;
    if (!defined($pid = open(TOSEND, "|-"))) {
        die "Can't fork to send mail: $!";
    }
```

```
   else {
       if ($pid) { #-> родитель
           # передача тела письма mail'y
           print TOSEND "$body\n$subject\n";

           if (!$innerflag && open(CMD, "$NSLOOKUP $ip |")) {
               print TOSEND "\n" . " * * * NSLOOKUP INFO" . " *" x
25 . "\n\n";
               print TOSEND while (<CMD>);
               close CMD;
           }
           if (!$innerflag && open(CMD, "$WHOIS $ip |")) {
               print TOSEND "\n" . " * * * WHOIS INFO" . " *" x 26
. "\n";
               print TOSEND while (<CMD>);
               close CMD;
           }

           close TOSEND;
       }
       else { #-> потомок
           exec("$MAIL $mailopts")
               or die "Can't exec mail: $!";
       }
   }
}
```

Додаток З

Перелік позначень та скорочень

АС — автоматизована система
ДВ — відновлення після збоїв
ДЗ — гаряча заміна
ДР — використання ресурсів
ДС — стійкість до відмов
КА — адміністративна конфіденційність
КВ — конфіденційність при обміні
КД — довірча конфіденційність
КЗЗ — комплекс засобів захисту
КК — аналіз прихованих каналів
КО — повторне використання об'єктів
КС — комп’ютерна система
КСЗІ — комплексна система захисту інформації
НА — автентифікація відправника
НВ — автентифікація при обміні
НИ — ідентифікація і автентифікація
НК — достовірний канал
НО — розподіл обов'язків
НП — автентифікація одержувача
НР — реєстрація
НСД — несанкціонований доступ до інформації
НТ — само тестування
НЦ — цілісність КЗЗ
ПІН — персональний ідентифікаційний номер
ПРД — правила розмежування доступу
ТЗІ — технічний захист інформації
ЦА — адміністративна цілісність
ЦВ — цілісність при обміні
ЦД — довірча цілісність
ЦО — відкат
AES — посилений стандарт шифрування (Advanced Encryption Standard)
АН — протокол автентифікуючого заголовку (Authentication Header)

ANSI — Американський національний інститут стандартів (American National Standards Institute)
CBC — режим зчеплення блоків шифру (Cipher Block Chaining)
DEA — алгоритм шифрування даних (Data Encryption Algorithm)
DES — стандарт шифрування даних (Data Encryption Standard)
DSS — стандарт цифрового підпису (Digital Signature Standard)
ECB — режим електронної шифрувальної книги (Electronic Code Book)
ECDSA — Алгоритм цифрового підпису на еліптичних кривих (Elliptic Curve Digital Signature Algorithm)
ESP — протокол захисту даних за допомогою інкапсуляції (Encapsulated Security Payload)
GNU — універсальна загальнодоступна ліцензія (General Public License)
IETF — Відкрите міжнародне співтовариство, що займається розвитком мережі Інтернет (Internet Engineering Task Force)
IP — протоколу обміну даними між комп'ютерними мережами (Internet Protocol)
IPSec — комплекс засобів захисту до протоколу IP (Internet Protocol Security)
ISAKMP — протокол узгодження параметрів безпеки та управління ключами (Internet Security Association and Key Management Protocol)
MD — стиснене повідомлення (Message Digest)
NBS — Національне бюро стандартизації (National Bureau of Standards)
NIST — Національний інститут стандартів і технологій США (National Institute of Standards and Technology)
NSA — Агенція національної безпеки (National Security Agency)
RSA — алгоритм шифрування даних з відкритим ключем, що названий на честь трьох винахідників (Ron Rivest, Adi Shamir, Leonard Adleman)
SA — домовленість про безпеку (Security Association)
SHA — алгоритм безпечного гешування (Secure Hash Algorithm)
UDP — протокол дейтаграм користувача (User Datagram Protocol)

Навчальне видання

ВИШНЯКОВ Володимир Михайлович

ЗАХИСТ ІНФОРМАЦІЇ В КОМП'ЮТЕРНИХ СИСТЕМАХ

Навчальний посібник

Редагування та коректура *В.М. Вишняков*
Комп’ютерне верстання *М.М. Власенко*

Підписано до друку 27.12.2022. Формат 60х84 1/16.
Ум. друк. арк. 6,96. Обл.-вид. арк. 5,28.
Тираж 80 прим. Вид. № 16/І-17. Зам. № 15/1- 18

Видавець і виготовлювач
Київський національний університет будівництва і архітектури

Повітрофлотський проспект, 31, Київ, Україна, 03680

Свідоцтво про внесення до Державного реєстру суб’єкту
Видавничої справи ДК №808 від 13.02.2002