МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ
Тернопільський національний економічний університет
Факультет комп'ютерних інформаційних технологій
Кафедра комп'ютерної інженерії

Славський Андрій Михайлович

Алгоритми вибору криптоалгоритмів захисту конфіденційної інформації / Algorithms for cryptoalgorithms selecting for confidential information protection

спеціальність:123 – Комп'ютерна інженерія
освітньо-професійна програма – Комп'ютерна інженерія

Випускна кваліфікаційна робота

Виконав студент групи КІм-21
А. М. Славський
_____

Науковий керівник:
к.т.н., Л. О. Дубчак
_____

ТЕРНОПІЛЬ - 2019

РЕЗЮМЕ

Випускна кваліфікаційна «Алгоритми вибору криптоалгоритмів захисту конфіденційної інформації» робота містить 77 сторінок пояснюючої записки, 20 рисунків, 3 таблиць, 1 додатків.

Метою кваліфікаційної роботи є розробка алгоритму вибору криптоалгоритмів захисту конфіденційної інформації.

Метод досліджень – алгоритм шифрування RSA, DES.

Середовище проектування – Matlab.

В даному випускному кваліфікаційному проекті розроблено та представлено алгоритм вибору криптоалгоритмів захисту конфіденційної інформації.

Методи дослідження – нечіткі алгоритми опрацювання інформації.

У кваліфікаційні роботі представляється загальна характеристика кібер-зловмисників або шкідливих програм, які спрямовані на захоплення інформаційних даних віддаленого комп'ютера, мотивації та прийняття рішень для надійного захисту конфіденційної інформації методом шифрування даних; перевірка та синтезування алгоритмів RSA та DES, алгоритм RSA є асиметричним, алгоритм DES є симетричним.

КЛЮЧОВІ СЛОВА: АЛГОРИТМ RSA, АЛГОРИТМ DES, MATLAB, МОДЕЛЬ, C++. КРИПТОАЛГОРИТМ ШИФРУВАННЯ.

RESUME

Final qualifying work «Algorithms for cryptoalgorithms selecting for confidential information protection» contains 77 pages of explanatory notes, 20 pictures, 3 tables, 1 applications.

Purpose of qualifying work is to develop selection algorithm encryption algorithms to protect confidential information.

The method of research - the encryption algorithm RSA, DES.

Environment Design - Matlab.

In this final qualifying project developed and presented selection algorithm encryption algorithms to protect confidential information.

Research Methods - fuzzy algorithms for processing information.

In qualifying work seems general description cyber attackers or malicious software aimed at remote data capture of information computer, motivation and decision making for reliable protection of sensitive information by encrypting data; verification and synthesis algorithms RSA and DES,RSA is an asymmetric algorithm, DES algorithm is symmetric.

KEYWORDS: ALGORITHM RSA, ALGORITHM DES, MATLAB, MODEL, C ++. ENCRYPTION ALGORITHM.

ЗМІСТ

ВСТУП

Сучасне суспільство неможливо уявити без інформаційних технологій. Комп'ютери, локальні та глобальні мережі, портативні пристрої – це все оточує нас в повсякденному житті.

Актуальність теми: алгоритми вибору криптоалгоритмів захисту конфіденційної інформації є актуально, оскільки захист інформації в сучасних комп’ютерних інформаційних системах є пріоритетним завданням. Викрадення конфіденційної інформації, знищення даних, виведення з ладу комп’ютерних систем – далеко не повний перелік усіх ризиків, що виникають у процесі експлуатації та використання сучасних інформаційних систем.

Доступ до цифрової інформації став однією із повсякденних потреб людини. Щоденно збільшується кількість тих, кому необхідно отримувати доступ до своєї електронної поштової скриньки, рахунку в банку, персональних сховищ файлів в мережі, соціальних мереж і т.д. Із збільшенням кількості інформації підвищується й актуальність її захисту від втрат та стороннього доступу, адже з кожним роком все більше зростає кількість вірусів, мережевих атак зловмисників, виникають загрози порушення конфіденційності інформації, що призводить до фінансових втрат, витоку особистих даних, завдання моральної шкоди. Новітні інформаційні системи будуються на основі мережевих технологій і все більш актуальними стають системи збереження інформації в "хмарних" сховищах, тобто в глобальній мережі "Інтернет". Навіть при цьому одним із найважливіших критеріїв оцінки сховища даних є параметри захисту інформації у ньому.

Метою випускної кваліфікаційної роботи є розробка нечіткого контролера вибору криптоалгоритму в комп’ютерній мережі. Контролер може бути використаний у комп'ютерних мережах як автономний пристрій або як частина комплексної системи захисту інформації. Вихідними значеннями криптоалгоритмів є RSA, DES та Ель-Гамаля, які можуть бути зміненими відповідно до проектованих систем.

Об’єкт дослідження – захист конфіденційно інформації.

Предмет дослідження – алгоритми вибору криптоалгоритмів захисту конфіденційної інформації.

Наукова новизна даної кваліфікаційної роботи полягає у вдосконаленні методу вибору криптоалгоритму, що базується на апараті нечіткої логіки.

Практична цінність – лежить у використанні на різних підприємствах, де стоїть питання захисту конфіденційної інформації.

Метод і завдання дослідження:

– проаналізувати сучасні засоби захисту інформації в комп’ютерній мережі;

– дослідити можливість застосування апарату нечіткої логіки для вибору крипто алгоритму;

– змоделювати нечітку систему вибору криптоалгоритму на основі механізму Мамдані;

– реалізувати контролер, що працює на основі розробленої нечіткої системи засобами Simulink;

– дослідити працездатність розробленого нечіткого контролера.

Публікація та апробація. Результати наукового дослідження опубліковано в матеріалах I та II науково-практичної конференції молодих вчених і студентів «Інтелектуальні комп’ютерні системи та мережі» (Тернопіль, 2019 р)[3,4].

У першому розділі проаналізовано сучасні засоби захисту в комп’ютерні мережі.Серед всього спектру методів захисту даних від небажаного доступу особливе місце займають криптографічні методи. Сучасні методи шифрування гарантують практично абсолютний захист даних, але завжди залишається проблема надійності їх стійкості до криптоаналізу, можливості зламу. Також однією із проблем криптографічних методів є використання ресурсів системи, тобто вплив на продуктивність. Визначення ефективності засобів захисту в тій чи іншій системі часто більш трудомістке, ніж їх розробка та реалізація.

У другому розділі проаналізовано сучасні алгоритми та нечіткі системи захисту інформації. Щоб зашифрувати відкритий текст, криптоалгоритм працює в

сполученні з ключем – словом, числом або фразою. Те саме повідомлення зашифроване одним алгоритмом, але різними ключами буде перетворюватися в різний шифртекст. Захищеність шифртекста цілком залежить від двох речей: стійкості криптоалгоритму і таємності ключа. Криптоалгоритм та ключі, протоколи, що приводять їх у дію, складають криптосистему. Відповідно, і атака зловмисників на систему захисту інформації в основному є спрямованою на отримання ключів дешифрування криптоалгоритму. Проблема системи може також полягати у значному зниженні продуктивності внаслідок використання потужного криптографічного алгоритму при великих обсягах інформації.

У третьому розділі побудовано нечіткий котролер та вдосконаленні алгоритму захисту інформації. Кроком до збільшення ефективності системи захисту інформації є вибір криптоалгоритму, який використовується в системі, тобто використання в системі декількох криптоалгоритмів.

Для вирішення даного завдання проект передбачає розробку контролера вибору криптоалгоритму в комп’ютерній мережі.

В основі переважної більшості інформаційних комп’ютерних систем покладено булеву алгебру – алгебру логіки. Але дедалі частіше розробники використовують також і системи, побудовані на можливостях нечіткої логіки, у якій існують перехідні величини (стани) на відміну від булевої алгебри, у котрій існує лише дві величини (0 та 1, правда чи неправда). Основна відмінність нечітких систем полягає у можливості задання вхідних параметрів системи нечіткими значеннями, що є ключовим, адже можливий вплив зловмисників на систему наперед не відомий, і дані про параметри неможливо чітко визначити. Контролери, що побудовані на основі нечіткої логіки, мають також значну перевагу у часі реагування на зміну вхідних параметрів.

Вступ випускної кваліфікаційної роботи оформлений згідно методички[1,2].

1 СУЧАСНІ ЗАСОБИ ЗАХИСТУ ІНФОРМАЦІЇ В КОМП'ЮТЕРНІЙ МЕРЕЖІ

1.1 Система передачі інформації в комп'ютерній мережі, методи захисту інформації

Сучасні мережні технології сприяли новій технічній революції. Створення мережі на підприємстві, в організації сприяє набагато кращому процесу обміну даними між різними структурними підрозділами, прискоренню документообігу, контролю за рухами тих чи інших матеріальних засобів, збільшенню й прискоренню передачі й обміну оперативною інформацією [5].

Комп'ютерна мережа – це сукупність комп'ютерів і різних пристроїв, що забезпечують інформаційний обмін між комп'ютерами в мережі без використання яких-небудь проміжних носіїв інформації.

Основні поняття, що використовуються в комп'ютерних мережевих технологіях описані нижче.

Комунікаційне чи мережеве устаткування – це периферійні пристрої, що здійснюють перетворення сигналів, які використовуються у комп'ютері, у сигнали, що передаються через лінії зв'язку, і навпаки. Такими пристроями є модеми і мережеві адаптери (мережеві карти). Модеми застосовуються при використанні телефонних ліній зв'язку, мережеві адаптери – при використанні інших ліній зв'язку – кабельних, оптоволоконних, безпровідних тощо [6].

Лінія зв'язку – це устаткування, за допомогою якого здійснюється з'єднання комп'ютерів у мережу.

Комунікаційне чи мережеве програмне забезпечення – це набір програм, що забезпечує роботу мережевого устаткування й обмін інформацією між комп'ютерами в мережі.

Усі комп'ютерні мережі розділяються на дві великі групи – локальні і глобальні [5].

Локальна мережа поєднує комп'ютери, що розташовані на невеликій відстані один від другого, і є замкнутою системою. Невеликі відстані між комп'ютерами

дають можливість використовувати в локальних мережах як лінії зв'язку звичайні лінії проводів.

Як правило, локальна мережа обмежена офісом, кабінетом, одним будинком. Найбільш розповсюдженими є локальні мережі, що включають в себе 5-25 персональних комп'ютерів, різні запам'ятовуючі пристрої, принтери, та інші периферійні пристрої. Локальні мережі повинні бути легко адаптованими, тобто мати гнучку архітектуру, що дозволяє довільно розташовувати робочі місця, чи додавати, переставляти персональні комп'ютери і периферійні пристрої. Якщо така мережа організована грамотно, то вихід з ладу однієї зі складових не впливає на роботу інших.

Ресурси мережі – це [5]:

– пристрої, що входять в апаратну частину одного з комп'ютерів мережі, доступні і можуть використовуватися будь-яким користувачем мережі. Ресурсами мережі можуть бути принтери, сканери, модеми, стримери, фотонабірні апарати, дискові нагромаджувачі великої ємності, пристрої резервного копіювання інформації, обладнання з програмним керуванням та ін.;

– дані і компоненти програмного забезпечення, що зберігаються на одному з комп'ютерів мережі.

За територіальною поширеністю мережі можуть бути локальними, глобальними й регіональними. Локальні – це мережі, що перекривають територію не більше 100 м^2, регіональні – це мережі, розташовані на території міста або області; глобальні – на території держави або групи держав, наприклад всесвітня мережа Інтернет.

За приналежністю також розрізняють відомчі й державні мережі. Відомчі належать одній організації і розташовуються на її території. Державні мережі використовуються в державних структурах.

За типом середовища передачі поділяються на мережі коаксіальні, на витій парі, оптоволоконні, із передачею інформації з радіоканалів, в інфрачервоному діапазоні [7].

Комп'ютери можуть з'єднуватися кабелями, утворюючи різну топологію мережі (зіркова, шинна, кільцева й т. ін.).

Слід розрізняти комп'ютерні мережі й мережі терміналів (термінальні мережі). Комп'ютерні мережі зв'язують комп'ютери, кожний з яких може працювати й автономно. Термінальні мережі зазвичай зв'язують потужні комп'ютери (мейнфрейми), а в окремих випадках і ПК із пристроями (терміналами), що можуть бути досить складними, але поза мережею їхня робота або неможлива, або взагалі втрачає зміст. Наприклад, мережа банкоматів чи терміналів поповнення різноманітних електронних рахунків.

У класифікації мереж існує два основні терміни: LAN і WAN. LAN (Local Area NeTworK, локальна обчислювальна мережа, ЛОМ) – локальні мережі, що мають замкнуту інфраструктуру. WAN (Wide Area NeTworK) – глобальна мережа, що покриває великі регіони, які включають у себе як локальні мережі, так і інші телекомунікаційні мережі й пристрої [8].

Термін «корпоративна мережа» також використовується в літературі для позначення об'єднання кількох мереж, кожна з яких може бути побудована на різних технічних, програмних та інформаційних принципах.

Глобальні мережі орієнтовані на обслуговування будь-яких користувачів, незалежно від їх місця розташування.

Комп'ютер, підключений до мережі, називається робочою станцією (WorKsTaTion); комп'ютер, що надає свої ресурси, – сервером (Server); комп'ютер, що має доступ до спільно використовуваних ресурсів, – клієнтом (ClienT).

Локальні обчислювальні мережі поділяються на два кардинально різні класи: однорангові (одноpівневі, або Peer To Peer) й ієрархічні (багаторівневі) [7].

Одноранговa мережа являє собою мережу рівноправних комп'ютерів, кожний з яких має унікальне ім'я (ім'я комп'ютера) і, як правило, пароль для входу в нього під час завантаження ОС. Ім'я й пароль входу надаються власником ПК засобами ОС.

В ієрархічних локальних мережах є один або кілька спеціальних комп'ютерів, серверів, на яких зберігається інформація, яка спільно використовується різними користувачами.

Сервер в ієрархічних мережах – це постійне сховище спільних ресурсів. Сам сервер може бути клієнтом тільки сервера вищого рівня ієрархії. Тому ієрархічні

мережі іноді називаються мережами з виділеним сервером. Сервери зазвичай являють собою високопродуктивні комп'ютери, інколи з кількома паралельно працюючими процесорами, з жорсткими дисками великого об'єму, із високошвидкісною мережною картою (1000 МБіт/с і більше).

Локальні комп’ютерні мережі класифікуються за призначенням:

– мережі термінального обслуговування. У них включається ЕОМ і периферійне устаткування, що використовується в монопольному режимі комп'ютером, до якого воно підключається, або може бути і загальномережним ресурсом;

– мережі, на базі яких побудовані системи управління виробництвом і управлінською діяльністю. Вони об'єднуються групою стандартів MAP/TOP. У MAP описуються стандарти, що використовуються в промисловості. TOP описують і стандарти для мереж, що використовуються в офісних мережах;

– мережі, що поєднують системи автоматизації, проектування. Робочі станції таких мереж, як правило, базуються на досить потужних персональних ЕОМ, наприклад фірми Sun MicrosysTeMs;

– мережі, на базі яких побудовані розподільні обчислювальні системи.

За ознакою швидкості мережі поділяються на низькошвидкісні (до 10 МБіт/с), середньошвидкісні (до 100 МБіт/с) і високошвидкісні (понад 100 МБіт/с).

Для того, щоб комп’ютери могли обмінюватися інформацією (повідомленнями), мережевих адаптерів і кабелів недостатньо, необхідно мати ще «мову», за допомогою якої вони могли б «розмовляти». Така «мова» називається протоколом.

Протокол – формалізовані правила прийому і передачі повідомлень між хостами (активними складовими мережі).

Усі протоколи можна розділити за семирівневою мережевою моделлю OSI.

Модель OSI (англ. Open SysTeMs InTerconnecTion Reference Model – модель взаємодії відкритих систем) – абстрактна модель для мережевих комунікацій і розробки мережевих протоколів. Представляє рівневий підхід до мережі. Кожен рівень обслуговує свою частину процесу взаємодії. Завдяки такій структурі спільна

робота мережевого обладнання й програмного забезпечення стає набагато простішою й зрозумілішою.

Модель OSI складається з 7-ми рівнів, розташованих вертикально один над іншим. Кожен рівень може взаємодіяти тільки зі своїми сусідами й виконувати відведені тільки йому функції (таблиця 1.1).

Таблиця 1.1 – Рівні OSI

Дані	Рівень OSI	Протоколи
Дані	Прикладний (доступ до мережних служб)	HTTP, gopHer, TelneT, DNS, SMTP, SNMP, CMIP, FTP, TFTP, SSH, IRC, AIM, NFS, NNTP, NTP, SNTP, XMPP, FTAM, APPC, X.400, X.500, AFP, LDAP, SIP, IETF, RTP, RTCP, ITMS, ModBus TCP, BACneT IP, IMAP, POP3, SMB, MFTP, BiTTorrenT, e2K, PROFIBUS та інші
Дані	Представлення (представлення і кодування даних)	ASN.1, XML, TDI, XDR, NCP, AFP, ASCII, Unicode
Дані	Сеансовий (керування сеансом зв'язку)	ASP, ADSP, DLC, NaMed Pipes, NBT, NeTBIOS, NWLinK, PrinTer Access ProTocol, Zone InforMaTion ProTocol, SSL, TLS, SOCKS, PPTP
Блоки	Транспортний (безпечне та надійне з'єднання «точка - точка»)	TCP, UDP, NeTBEUI, AEP, ATP, IL, NBP, RTMP, SMB, SPX, SCTP, DCCP, RTP, STP, TFTP
Пакети	Мережний (визначення маршруту та IP (логічна адресація))	IPv4, IPv6, ICMP, IGMP, IPX, NWLinK, NeTBEUI, DDP, IPSec, ARP, SKIP
Кадри	Канальний (MAC та LLC) (фізична адресація)	ARCneT, ATM, DTM, SLIP, SMDS, ETHerneT, FDDI, FraMe Relay, LocalTalK, ToKen Ring, PPP, PPPoE, STarLan, WiFi, PPTP , L2F, L2TP, PROFIBUS
Біти	Фізичний (кабель, сигнали, бінарна передача)	RS-232, RS-422, RS-423, RS-449, RS-485, ITU-T, RJ-11, T-carrier (T1, E1), модифікації стандарту ETHerneT: 10BASE-T, 10BASE2, 10BASE5, 100BASE-TX, 100BASE-FX, 100BASE-T, 1000BASE-T, 1000BASE-TX, 1000BASE-SX

Зараз основним протоколом, що використовується у комп’ютерних мережах є TCP/IP. TCP/IP – це абревіатура терміну TransMission ConTrol ProTocol / InTerneT ProTocol (протокол керування передачею / міжмережевий протокол). Фактично, TCP/IP це не один протокол, а декілька. Саме тому його часто називають набором, або комплектом протоколів, серед яких TCP і IP – два основні. TCP/IP представляє цей базовий набір протоколів, відповідальний за розбивання вихідного повідомлення на пакети (TCP), доставку пакетів на вузол адресата (IP) і збирання (відновлення) вихідного повідомлення з пакетів (TCP).

Крім протоколу обміну необхідна програма, яка взаємодіє з операційною системою і мережевим обладнанням.

TCP/IP представляє собою сімейство протоколів, що були розроблені в сімдесятих роках у рамках спеціального проекту Управління перспективних досліджень і розробок Міністерства оборони США з метою розвитку системи зв'язку між навчальними закладами і науково-дослідними інститутами. Розроблялися ці протоколи для Unix-систем, при цьому основні дослідження проводилися в Каліфорнійскому університеті (м. Берклі). Саме ці протоколи застосовуються в InTerneT і багатьох локальних мережах та для об’єднання комп’ютерів та мереж.

Відповідно моделі OSI інформація у комп’ютерних мережах передається пакетами даних. Пакетна передача – це технологія передачі цифрової інформації, що передбачає розподіл потоку інформаційних сигналів на частини, кожна з яких утворює окремий пакет, після чого пакети у визначеній послідовності передаються у телекомунікаційні мережі [10].

Вся інформація, що передається по мережі: файли, звук, відео і т. д., являє собою масив цифрових даних. На вихідному сервері ці дані розділяються на окремі «порції» заздалегідь обумовленої довжини (наприклад, по 256 байт), причому кожна з них забезпечується індивідуальним «заголовком». Така «порція» називається пакетом.

У заголовку пакета міститься інформація про місце призначення (наприклад, адресу в Інтернеті комп'ютера користувача, що подав запит), про ім'я файлу, до

якого належить цей пакет, і про порядковий номер даного пакету (тобто про те, з якого місця файлу він був «вирізаний»), а також контрольна сума – деяке число, що служить для перевірки правильності передачі.

Пакети пересилаються по мережі Інтернет, іноді навіть по різних маршрутах, залежно від завантаженості тих чи інших ліній зв'язку. Маршрут проходження кожного пакета визначають спеціальні комп'ютери – IP-маршрутизатори. Така технологія передачі даних називається динамічною маршрутизацією. На комп'ютері користувача для кожного пакета після його отримання підраховується окремо один від одного контрольна сума і звіряється з тим значенням, яке зберігається в заголовку. Якщо два значення контрольної суми збігаються, то пакет вважається прийнятим без помилок. В іншому випадку він повторно запрошується з сервера (тільки цей пакет, а не увесь файл). Коли ж усі пакети «зібрані», вони автоматично об'єднуються в файл, який є точною копією вихідного повідомлення [11]. Політика безпеки – це визначення того, що означає бути безпечним для системи , організації чи іншої організації. Для організації вона розглядає обмеження на поведінку своїх членів, а також обмеження, що накладаються на противників такими механізмами, як двері, замки, ключі та стіни. Для систем політика безпеки спрямована на обмеження функцій та потоку між ними, обмеження доступу зовнішніх систем та супротивників, включаючи програми та доступ до даних людьми.

До адміністративного рівня інформаційної безпеки відносяться заходи загального характеру, що реалізуються керівництвом організації.

Головна мета заходів адміністративного рівня – сформувати програму робіт у галузі інформаційної безпеки і забезпечити її виконання, виділяючи необхідні ресурси і контролюючи стан подій.

Основою програми є політика безпеки, що відображає підхід організації до захисту своїх інформаційних активів. Керівництво кожної організації повинне усвідомлювати необхідність підтримки режиму безпеки і виділення на ці цілі значних ресурсів.

Політика безпеки будується на основі аналізу ризиків, які визнаються реальними для ІС організації. Коли ризики проаналізовано і стратегію захисту

визначено, тільки тоді складається програма забезпечення інформаційної безпеки. Під цю програму виділяються ресурси, призначаються відповідальні, визначається порядок контролю виконання програми тощо.

Термін “політика безпеки” є не зовсім точним перекладом англійського словосполучення “SecuriTy policy”, проте в даному випадку калька краще відображає сенс цього поняття, ніж лінгвістично правильний переклад “правила безпеки”. Тут мова йде не про окремі правила або їх набори, а про стратегію організації у галузі інформаційної безпеки. Для вироблення стратегії і втілення її в життя потрібні політичні рішення, що приймаються на найвищому рівні керівництва організації, установи чи підприємства.

Якщо важливо бути захищеним, тоді важливо бути впевненим, що вся політика безпеки застосовується механізмами, які є достатньо сильними. Існує багато організованих методологій та стратегій оцінки ризиків для забезпечення повноти політики безпеки та забезпечення їх повної реалізації. У складних системах, таких як інформаційні системи, політика може бути розбита на суб-політику, щоб полегшити розподіл механізмів безпеки для забезпечення виконання субполітики. Однак ця практика має підводні камені. Занадто легко просто перейти безпосередньо до суб-політик, які по суті є правилами роботи та не відповідають політиці вищого рівня. Це дає помилкове відчуття, що правила роботи вирішують якесь загальне визначення безпеки, якщо вони цього не роблять. Через те, що настільки важко чітко продумати повну інформацію про безпеку, правила роботи, що визначаються як "субполітики", які не мають "суперполітики", зазвичай виявляються незрозумілими правилами, які не досягають нічого з повнотою. Отже, політика безпеки на вищому рівні має важливе значення для будь-якої серйозної схеми безпеки, і підполітики та правила роботи без цього не мають сенсу.

Політика комп'ютерної безпеки визначає цілі та елементи комп'ютерних систем організації. Визначення може бути дуже формальним або неформальним. Політика безпеки забезпечується організаційної політикою або механізмами безпеки. Технічна реалізація визначає, чи є комп'ютерна система безпечною або небезпечною . Ці формальні моделі політики можна розділити на основні принципи безпеки: конфіденційність, цілісність та доступність. Наприклад, модель Белл-Ла-

Падула є моделлю політики конфіденційності, тоді як модель Біба є моделлю політики цілісності .

Модель Bell-LaPadula (BLP) – це модель державної машини, яка використовується для забезпечення контролю доступу у державних та військових програмах. Він був розроблений Девідом Елліоттом Беллом та Леонардом Дж. Ла Падулою, після чіткого керівництва Роджера Шелла, з метою формалізації політики багаторівневої безпеки Міністерства оборони США (МОП). Модель являє собою формальну перехідну модель політики комп'ютерної безпекиякий описує набір правил контролю доступу, які використовують мітки безпеки на об'єктах та пропусках для предметів. Мітки безпеки охоплюють найбільш чутливі (наприклад, "найпотаємніші"), найменш чутливі (наприклад, "не класифіковані" або "загальнодоступні").

Модель Bell-LaPadula зосереджується на конфіденційності даних та контрольованому доступу до секретної інформації , на відміну від моделі BiBa InTegriTy, яка описує правила захисту цілісності даних . У цій формальній моделі об'єкти в інформаційній системі поділяються на предмети та об'єкти. Поняття " безпечного стану " визначено, і доведено, що кожен перехід держави зберігає безпеку, переміщаючись з безпечного стану до безпечного стану, тим самим індукційно доводячи, що система задовольняє цілям безпеки моделі. Модель Bell-LaPadula побудована на концепції державної машиниз набором допустимих станів у комп'ютерній системі. Перехід від одного стану до іншого стану визначається перехідними функціями.

Стан системи визначається як "безпечний", якщо єдині дозволені режими доступу суб'єктів до об'єктів відповідають політиці безпеки . Щоб визначити, чи дозволено певний режим доступу, кліренс об'єкта порівнюється з класифікацією об'єкта (точніше, з комбінацією класифікації та набору відсіків, що складають рівень безпеки), щоб визначити, чи є суб'єкт авторизації для конкретного режиму доступу. Схема оформлення / класифікації виражається у вигляді решітки. Модель визначає один режим дискреційного контролю доступу (DAC) та два правила обов'язкового контролю доступу (MAC) з трьома властивостями безпеки:

– проста безпека об'єктів встановлює, що суб'єкт на заданому рівні безпеки може не читати об'єкт на більш високому рівні безпеки;

– властивість * (зірка) вказує, що об'єкт на заданому рівні безпеки може не писати жодного об'єкта на більш низькому рівні безпеки;

– дискреційна безпечна нерухомість говорить, що використання матриці доступу для вказівки дискреційного контролю доступу.

Передача інформації з документа високої чутливості на документ з меншою чутливістю може статися в моделі Белл-Ла-Падула за допомогою концепції довірених предметів. Довірені предмети не обмежуються властивістю "Зірка". Надійні предмети повинні бути визнані надійними щодо політики безпеки. Ця модель безпеки спрямована на контроль доступу і характеризується фразою: "читайте, записуйте". Порівняйте модель BiBa , то модель Кларка-Вілсона і китайська стіна модель.

За допомогою Bell-LaPadula користувачі можуть створювати вміст тільки на власному рівні безпеки або вище (тобто секретні дослідники можуть створювати секретні або секретні файли, але не можуть створювати публічні файли, ніяких записів). І навпаки, користувачі можуть переглядати вміст лише на власному рівні безпеки (тобто, секретні дослідники можуть переглядати загальнодоступні або секретні файли, але не можуть переглядати секретні файли, а не переглядати).

Модель Bell-LaPadula чітко визначила його масштаб. Це не розглядало наступне:

– приховані канали – коротко розповідається про передачу інформації за допомогою попередньо впорядкованих дій;

– мережі систем – пізніше модельна робота звернулася до цієї теми;

– політика поза багаторівневою безпекою – робота на початку 1990-х років показала, що MLS є однією з версій логічної політики , як і всі інші опубліковані політики.

Сила зоряної власності – це альтернатива * -ProperTy, в якій об'єкти можуть писати на об'єкти з відповідним рівнем безпеки. Таким чином, операція запису, дозволена в звичайному * -ProperTy, відсутня, лише операція wriTe-To-saMe.

Майно STrong STar, як правило, обговорюється в контексті багаторівневих систем управління базами даних і обумовлено проблемами цілісності. Ця сильна зірка властивість була передбачена в моделі Біба, де було показано, що сильна цілісність у поєднанні з моделлю Белл-Ла-Падула призвела до читання та письма на одному рівні.

Принцип спокою моделі Белл-Ла-Падула свідчить, що класифікація предмета або об'єкта не змінюється, коли на неї посилаються. Для принципу спокою є дві форми: "принцип сильного спокою" свідчить, що рівень безпеки не змінюється під час нормального функціонування системи. "Принцип слабкого спокою" вказує, що рівень безпеки ніколи не може змінюватися таким чином, щоб порушувати певну політику безпеки. Слабка спокій бажана, оскільки це дозволяє системам дотримуватися принципу найменшої привілеї. Тобто процеси починаються з низького рівня клірингу незалежно від їх оформлення власників і поступово накопичують більший рівень кліренсу, оскільки це вимагає дії.

Обмеження:

– відноситься лише конфіденційність, контроль написання (одна форма цілісності), власність та дискреційний контроль доступу;

– таємні канали згадуються, але не розглядаються всебічно;

– принцип спокою обмежує його придатність до систем, де рівні безпеки не змінюються динамічно. Це дозволяє керувати копіюванням від високого до низького за допомогою надійних предметів;

– модель державного переходу не містить жодних інваріантів стану;

– загальний процес може зайняти більше часу.

Модель Біба або Біба Цілісність Модель, розроблена KenneTH J. BiBa в 1975 році є формальним стан системи переходу з комп'ютерної безпеки політики, яка описує набір контролю доступу правил, призначених для забезпечення цілісності даних. Дані та предмети згруповані в упорядкований рівень цілісності. Модель розроблена таким чином, щоб об'єкти не могли пошкоджувати дані на рівні, що перевищує предмет, або може бути пошкоджено даними з нижчого рівня, ніж предмета.

Загалом, модель була розроблена для вирішення цілісності як основного принципу, що є прямим зворотним для моделі Белла-Ла-Падула .

Загалом, збереження цілісності даних має три цілі:

– запобігання модифікації даних неавторизованими сторонами;

– запобігання неавторизованій модифікації даних уповноваженими сторонами;

– підтримка внутрішньої та зовнішньої послідовності (тобто дані відображають реальний світ).

Ця модель безпеки спрямована на цілісність даних (а не конфіденційність) і характеризується фразою: "прочитати, записати". Це на відміну від моделі Белл-Ла-Падула, яка характеризується фразою "читати, писати".

У моделі BiBa користувачі можуть створювати вміст лише на рівні власної цілісності (або монашество може написати молитовну книгу, яку можуть прочитати простолюдини, але не можна читати первосвящеником). І навпаки, користувачі можуть переглядати лише вміст на рівні їхньої власної цілісності або вище (монах може читати книгу, написану первосвящеником, але не може прочитати брошуру, написану слабким простором). Інша аналогія, яку слід враховувати – це військовий ланцюг командування. Генерал може написати розпорядження полковнику, який може видати ці замовлення на майор. Таким чином, первинні розпорядження Генерального секретаря залишаються незмінними, і місія військових захищена (таким чином, "читати" цілісність). І навпаки, Приватний ніколи не може видати розпорядження своєму сержанту, який ніколи не може надсилати замовлення лейтенанту, а також захищати цілісність місії ("записувати").

Модель BiBa визначає набір правил безпеки, перші два з яких схожі на модель Bell-LaPadula . Ці перші два правила є зворотним правилами Белла-Ла-Падула:

– прості інтегральні властивості стверджують, що суб'єкт на заданому рівні цілісності не повинен читати дані на нижчому рівні цілісності (read up);

– властивість цілісності * (зірка) встановлює, що суб'єкт на заданому рівні цілісності не повинен записувати дані на більш високий рівень цілісності (записати);

– invoKe ProperTy встановлює, що процес нижче може не вимагати більш високого доступу; лише з предметами на рівні або нижчому рівні.[13]

З практичної точки зору політику безпеки доцільно розглядати на трьох рівнях деталізації.

До верхнього рівняналежать рішення, що стосуються організації в цілому. Вони мають загальний характер і, як правило, виходять від керівництва організації. Список подібних рішень може складатися з таких елементів:

– рішення про формування або перегляд комплексної програми забезпечення інформаційної безпеки, призначення відповідальних за реалізацію програми;

– формулювання цілей, до яких прагне організація у галузі інформаційної безпеки, визначення загальних напрямків досягнення цих цілей;

– забезпечення бази для дотримання законів і правил;

– формулювання адміністративних рішень з тих питань реалізації програми безпеки, які повинні розглядатися на рівні організації в цілому.

Для політики верхнього рівня цілі організації в галузі інформаційної безпеки формулюються у термінах цілісності, доступності і конфіденційності. Якщо організація відповідає за підтримку критично важливих баз даних, на першому плані може стояти зменшення кількості втрат, пошкоджень або спотворень даних. Для організації, що займається продажем комп’ютерної техніки, ймовірно, важлива актуальність інформації про послуги і ціни та її доступність максимальній кількості потенційних покупців. Керівництво режимного підприємства в першу чергу піклується про захист від несанкціонованого доступу, тобто про конфіденційність.

На верхній рівень виносять управління захисними ресурсами і координація використання цих ресурсів, виділення спеціального персоналу для захисту критично важливих систем і взаємодія з іншими організаціями, що забезпечують або контролюють режим безпеки.

Політика верхнього рівня повинна чітко окреслювати сферу свого впливу. Можливо, це будуть всі комп'ютерні системи організації (або навіть більше, якщо політика регламентує деякі аспекти використання співробітниками своїх домашніх комп'ютерів). Можлива, проте, і така ситуація, коли до сфери впливу включаються лише найважливіші системи.[13]

У політиці ІБ повинні бути визначені обов'язки посадовців щодо створення програми безпеки і впровадження її в життя.

Політика верхнього рівня має справу з трьома аспектами законопокірності і виконавської дисципліни:

– організація повинна дотримуватися існуючих законів;

– слід контролювати дії осіб, відповідальних за створення програми безпеки;

– необхідно забезпечити певний ступінь старанності персоналу, а для цього потрібно створити систему заохочень і покарань.

На верхній рівень слід виносити тільки ті питання, які забезпечують значну економію засобів, або без яких неможливо обійтися.

Британський стандарт BS 7799:1995 рекомендує включати до документа, що характеризує політику безпеки організації, такі розділи:

– вступний, який підтверджує заклопотаність вищого керівництва проблемами інформаційної безпеки;

– організаційний, що містить опис підрозділів, комісій, груп і т.д., які відповідають за роботи у галузі інформаційної безпеки;

– класифікаційний, що описує наявні в організації матеріальні та інформаційні ресурси і необхідний рівень їх захисту;

– штатний, що характеризує заходи безпеки, вживані до персоналу (опис посад з погляду інформаційної безпеки, організація навчання і перепідготовки персоналу, порядок реагування на порушення режиму безпеки і т.п.);

– розділ, який висвітлює питання фізичного захисту;

– розділ, який описує підхід до управління комп'ютерами і комп'ютерними мережами;

– розділ, що описує правила розмежування доступу до виробничої інформації;

– розділ, що характеризує порядок розробки і супроводу систем;

– розділ, що описує заходи, спрямовані на забезпечення безперервної роботи організації;

– юридичний розділ, який підтверджує відповідність політики безпеки чинному законодавству.[15]

До середнього рівнявідносять питання, що стосуються окремих аспектів інформаційної безпеки, але важливі для різних експлуатованих організацією систем. Приклади таких питань – ставлення до передових (але, можливо, недостатньо перевірених) технологій, доступ до InTerneT (як сумістити свободу доступу до інформації із захистом від зовнішніх загроз?), використання домашніх комп’ютерів, застосування користувачами неліцензійного програмного забезпечення і т.д.

Останнім часом повідомлення про атаки на інформацію, про хакерів і комп'ютерні зломи наповнили всі засоби масової інформації.

Дати визначення атаці на інформацію складно, оскільки інформація, особливо в електронному вигляді, представлена багатьма різними видами. Інформацією можна вважати і окремий файл, і базу даних, і один запис у ній, і цілком програмний комплекс. І всі ці об'єкти можуть піддатися і піддаються атакам з боку деякої соціальної групи осіб.

При зберіганні, підтриманні і надання доступу до будь-якого інформаційного об'єкту його власник, або уповноважена ним особа, накладає явно або самоочевидно набір правил по роботі з нею [17]. Умисне їх порушення класифікується як атака на інформацію.

З масовим впровадженням комп'ютерів в сфери діяльності людини обсяг інформації, що зберігається в електронному вигляді виріс в тисячі разів. А з появою комп'ютерних мереж навіть відсутність фізичного доступу до комп'ютера перестала бути гарантією збереження інформації.

Наслідками атаки на інформацію у першу чергу є, звичайно, економічні

втрати, а саме [16]:

– розкриття комерційної інформації може призвести до серйозних прямих збитків на ринку;

– звістка про крадіжку великого обсягу інформації звичайно серйозно впливає на репутацію фірми, приводячи побічно до втрат в обсягах торгових операцій;

– фірми-конкуренти можуть скористатися крадіжкою інформації, якщо та залишилася непоміченою, для того, щоб повністю розорити фірму, нав'язуючи їй фіктивні або свідомо збиткові угоди;

– підміна інформації як на етапі передачі, так і на етапі зберігання може привести до величезних збитків, пов'язаних, наприклад, із некоректною інформацією про номери рахунків, суми коштів тощо;

– багаторазові успішні атаки на фірму, яка надає певні інформаційні послуги, знижують довіру до фірми в клієнтів, дають право сумніватися в коректності отриманої інформації.

1.2 Ідентифікація клієнтів в комп’ютерній мережі

Клієнти в комп'ютерній мережі – це комп'ютери чи мережеві пристрої, що мають доступ до спільно використовуваних ресурсів цієї мережі. В ширшому розумінні клієнт – це апаратний або програмний компонент обчислювальної системи, який надсилає запити серверу [18].

Програма-клієнт взаємодіє з сервером, використовуючи певний протокол. Вона може запитувати з сервера будь-які дані, маніпулювати даними безпосередньо на сервері, запускати на сервері нові процеси і т. п. Отримані від сервера дані клієнтська програма може надавати користувачеві або використовувати як-небудь інакше, в залежності від призначення програми. Програма-клієнт і програма-сервер можуть працювати як на одному і тому ж

комп'ютері, так і на різних. У другому випадку для обміну інформацією між ними використовується мережеве з'єднання.

Різновидом клієнтів є термінали – робочі місця на багатокористувацьких ЕОМ, обладнані монітором та клавіатурою, і не здатні працювати без сервера. У 1990-і роки з'явилися мережеві комп'ютери – щось середнє між терміналом і персональним комп'ютером. Мережеві комп'ютери мають спрощену структуру і багато в чому залежать від сервера. Іноді терміналом називають будь-який клієнт або тільки окремий клієнт.

Тим не менш, не завжди під клієнтом мається на увазі комп'ютер зі слабкими обчислювальними ресурсами. Найчастіше поняття «клієнт» і «сервер» описують розподіл ролей при виконанні конкретного завдання, а не обчислювальні потужності. На одному і тому ж комп'ютері можуть одночасно працювати програми, що виконують як клієнтські, так і серверні функції. Наприклад, веб-сервер може в якості клієнта отримувати дані для формування сторінок від SQL-сервера.

Для ідентифікації клієнтів в комп’ютерних мережах використовуються ідентифікатори клієнтів, що є унікальними в усій мережі. Одними із найбільш часто використовуваних ідентифікаторів є IP-адреси та MAC-адреси [7].

IP-адреса (InTerneT ProTocol address) – це ідентифікатор (унікальний числовий номер) мережевого рівня, що використовується для адресації комп'ютерів чи пристроїв у мережах, що побудовані з використанням протоколу TCP/IP (наприклад, Інтернет). IP-адреси є двох версій: версія 4 та версія 6.

IP адреса версії 4. Це унікальна адреса, що ідентифікує окремий комп’ютер мережі. Прийнята в мережах IP адреса є 32-бітним числом. Для спрощення IP-адреса розбита на чотири 8-ми бітних числа, сегменти і має свою структуру, в якій одна частина визначає адресу мережі, до якої підключений даний комп’ютер, а інша означує конкретний хост-комп’ютер в цій мережі. Мережева частина адреси займає перші три сегменти, а адреса машини – останній сегмент. У сукупності ці сегменти утворять унікальну адресу, за допомогою якого ідентифікується будь який комп'ютер у мережі, що працює по протоколах TCP/IP. Наприклад, у IP-адресі 194.1.1.3 мережева частина – 194.1.1, а машинна частина – 3. Даний комп'ютер є

частиною мережі, адреса якої 194.1.1.0.

Якщо локальна мережа не має TCP/IP з'єднання з іншими мережами, можна використовувати будь-які адреси, але все таки бажано користуватися загальноприйнятими правилами про призначення адреси.

Для чіткої структуризації існує декілька класів мереж, від яких залежить максимальна кількість адрес для хостів [19]:

– мережі класу A – включає мережі з 1.0.0.0 до 127.0.0.0. Номер мережі знаходиться в першому байті октету. Це забезпечує 24-ох розрядну частину для означення хостів. Дозволяє використання приблизно 1,6 мільйони хостів у мережі;

– мережі класу B – включає мережі з 128.0.0.0 по 191.255.0.0; номер мережі знаходиться в перших двох байтах октету. Це нараховує 16320 мереж з 65024 хостом в кожній;

– мережі класу C – включає мережі від 192.0.0.0 по 223.255.255.0; номер мережі – перших три числа в октеті. Нараховує 2 мільйони мереж з 254 хостами в кожній;

– мережі класу D, E, F – адреси, що підпадають в діапазон з 224.0.0.0 по 254.0.0.0 є або експериментальними, або збережені для використання у майбутньому і не описують будь-якої мережі.

Згідно нашого прикладу адреса 194.1.1.3 належить хосту з номером 3, мережі класу C з номером 194.1.1.

Адреси: 0.0.0.0 та 127.0.0.0 є зарезервовані і їх використовувати неможна при присвоєнні адрес хостам. Перша називається маршрутом за замовчуванню, а друга - адресою локального інтерфейсу.

Для адрес хостів локальних мереж, які не мають адреси в Інтернет (приватні мережі) зарезервовані наступні адреси, які не транслюються в Інтернет:

10.0.0.0 – 10.255.255.255 – 1 мережа класу A

172.16.0.0 – 172.31.255.255 – 16 мереж класу B

192.168.0.0 – 192.168.255.255 – 256 мереж класу C

Адресу мережі можна легко встановити за адресою хост-комп’ютера. Адреса мережі – це мережева частина адреси хоста плюс нуль, наприклад, у хост-адресі

194.1.1.3 адреса мережі – 194.1.1.0 [18].

Системи визначають адресу мережі за адресою хост-комп'ютера за допомогою маски мережі, виконуючи порозрядну операцію «І» з маскою мережі й адресою хост-комп'ютера, отримуючи обнулення машинної частини адреси й одержанню його мережевої частини.

Маска мережі (NeTMasK) використовується для одержання адреси мережі. При визначенні маски мережі адреса хост-комп'ютера виступає в ролі трафарету. Всі числа в мережевій частині хост-адреси встановлюються рівними 255, а в машинній частині ставиться нуль. Це і є маска мережі. Маска мережі для адреси 194.1.1.3 – 255. 255. 255.0.

Мережева частина, 194.1.1, замінена на 255. 255. 255, а машинна частина 3, замінена нулем. За допомогою цієї маски системи визначають по вашій хост-адресі адресу мережі.

Наведений приклад маски належить мережі класу С, для мереж класу В мережева маска відповідно буде 255.255.0.0. Для loopBacK мережева маска завжди 255.0.0.0 [18].

Широкомовна (BroadcasT) адреса дозволяє системі посилати повідомлення одночасно всім системам у мережі. Як і мережева адреса, широкомовну адресу можна легко визначити за адресою хост-комп'ютера; машинна частина в ньому встановлена рівною 255, а мережева частина не змінюється.

IP адреса версії 6 (англ. InTerneT ProTocol version 6) – нова версія протоколу. Розробка протоколу IP версії 6 почалася 1992 року, а з 2003 р. його підтримку забезпечують виробники більшості телекомунікаційного устаткування (корпоративного рівня). IP версії 6 – новий крок у розвитку Інтернету. Цей протокол розроблено з урахуванням вимог до Глобальної мережі, що постійно зростають. 3 лютого 2011 року IANA виділила останні п'ять блоків IP-адрес /8 (IP версії 4). В технічних документах протоколи IP версії 4 та версії 6 пишуть як IPv4 та IPv6, відповідно [18].

Найбільш суттєва різниця між IPv4 та IPv6 полягає в тому, що раніше на інтернет-адресу виділяли 4 байти (32 біта), що відповідає стандартній на сьогодні чотирьохблоковій адресі IP, а протокол IPv6 виділяє на адресу 16 байтів (128 біт).

Це відповідає 340 трильйонам трильйонів трильйонів адрес ($3{,}4x10^{38}$) або по $5x10^{28}$ адрес на кожну людину.

5 лютого 2008 року організація ICANN, яка наглядає за використанням інтернет-протоколів, почала додавати в DNS-сервери записи, що містять адреси у форматі протоколу IPv6. Це поклало початок переходу з нинішнього протоколу IPv4 на сучасніший IPv6 [20].

Розширення адресного простору скасовує необхідність використання NAT, оскільки на кожну людину припадає близько $3*10^8$ унікальних адрес. Принцип призначення хосту IPv6 адреси є ієрархічним. Мінімальний розмір підмережі – /64 (2^{64}). Молодша частина адреси (64 біти) використовується як унікальний ідентифікатор користувача, наступна частина визначає підмережу всередині оператора зв'язку, далі йде ідентифікатор самого оператора. Такий підхід значно спрощує маршрутизацію.

З IPv6 вилучено кілька функцій, що ускладнюють роботу маршрутизаторів:

– маршрутизатори більше не розбивають (фрагментують) пакет на частини (розбиття пакета можливо тільки на боці передавача). Відповідно, оптимальний MTU має визначатися за допомогою PaTH MTU discovery. Для покращення роботи протоколів, що потребують низького рівня втрати пакетів, мінімальний MTU збільшено до 1280 байт. Інформацію про фрагментацію пакетів перенесено з основного заголовку в розширені;

– зникла контрольна сума. Оскільки канальні (ETHerneT) та транспортні (TCP) протоколи також перевіряють коректність пакета, контрольна сума на рівні IP вважається зайвою. Крім того, кожен маршрутизатор зменшує Hop liMiT на одиницю, що призводить до потреби у перерахуванні суми в IPv4.

Незважаючи на суттєве збільшення розміру адреси IPv6, завдяки цим покращенням основний заголовок пакета збільшився лише у 2 рази: з 20 до 40 байт.

Покращення IPv6 у порівнянні з IPv4 [20]:

– в надшвидкісних мережах можлива підтримка надвеликих пакетів (джамбограм) – до 4 гігабайт;

– TiMe To Live перейменовано в Hop liMiT;

– з'явились відмітки потоків та класи трафіку;
– з'явилась багатоадресна передача;
– протокол IPsec з рекомендованого перетворився на обов'язковий.

У момент ініціалізації мережевого інтерфейсу йому призначується локальна IPv6-адреса, з префіксом fe80::/10, у молодшій частині адреси розміщується ідентифікатор інтерфейсу. У якості ідентифікатора інтерфейсу часто використовується 64-бітний розширений унікальний ідентифікатор EUI-64, що найчастіше формується з MAC-адреси. Локальна адреса дійсна тільки в межах мережевого сегменту канального рівня і використовується, в основному, для обміну інформаційними ICMPv6 пакетами [21].

Для отримання інших адрес вузол може здійснити запит про інформацію щодо налаштування мережі у маршрутизаторів за допомогою ICMPv6 повідомлення «RouTer SoliciTaTion». Цей запит відсилається на групову (MulTicasT) адресу маршрутизаторів. У відповідь маршрутизатори відсилають ICMPv6 повідомлення «RouTer AdverTiseMenT», що може містити інформацію про префікс мережі, адресу шлюзу, адреси рекурсивних серверів DNS, MTU та багато інших параметрів. Поєднуючи мережевий префікс та ідентифікатор інтерфейсу, вузол отримує нову адресу. Для захисту персональних даних ідентифікатор інтерфейсу може бути замінений на псевдовипадкове число.

Для більшого адміністративного контролю може бути використаний DHCPv6, що дозволяє адміністратору маршрутизатора призначати вузлам конкретні адреси.

IPv6 адреси показуються як вісім груп по чотири шістнадцяткові цифри, розділених двокрапками. Приклад адреси:

2001:0dB8:11a3:09d7:1f34:8a2e:07a0:765d

Якщо одна чи більше груп підряд дорівнюють 0000, то вони можуть скорочено записуватись як подвійна двокрапка (::). Наприклад, 2001:0dB8:0000:0000:0000:0000:ae21:ad12 може бути скорочена до 2001:dB8::ae21:ad12, 0000:0000:0000:0000:0000:0000:ae21:ad12 – до ::ae21:ad12. Скорочення не дозволяється у випадку, коли адреса містить 2 окремі нульові групи

через виникнення невизначеності [21].

При використанні IPv6-адреси в URL необхідно брати адресу в квадратні дужки:

HTTp://[2001:0dB8:11a3:09d7:1f34:8a2e:07a0:765d]/

Якщо потрібно вказати порт, то він пишеться після дужок:

HTTp://[2001:0dB8:11a3:09d7:1f34:8a2e:07a0:765d]:8080/

У багатьох мережах, включаючи InTerneT, є комп'ютери, що працюють як сервери доменних імен, для перетворення доменних імен мереж і хост-машин у IP-адреси. Це дозволяє ідентифікувати комп'ютер у мережі, користуючись не IP-адресою, а доменним ім'ям, так як IP-адреси досить важко запам’ятати, а ім’я комп’ютера значно легше: наприклад, wiKi.Tneu.edu.ua. До інших систем теж можна звертатися по доменних іменах, тому їхньої IP-адреси знати не обов'язково. Для цього, необхідно знати IP-адреси серверів доменних імен мережі.

MAC-адреса (Media Access ConTrol – управління доступом до носія) – це унікальний ідентифікатор, що зіставляється з різними типами устаткування для комп'ютерних мереж. Більшість мережевих протоколів канального рівня використовують один з трьох просторів MAC-адрес, керованих IEEE: MAC-48, EUI-48 і EUI-64. Адреси в кожному з просторів теоретично мають бути глобально унікальними. Не всі протоколи використовують MAC-адреси, і не всі протоколи, що використовують MAC-адреси, потребують подібної унікальності цих адрес.

У широкомовних мережах (таких, як мережі ETHerneT) MAC-адреса дозволяє унікально ідентифікувати кожен вузол мережі і доставляти дані тільки цьому вузлу. Таким чином, MAC-адреси формують основу мереж на канальному рівні, яку використовують протоколи вищого рівня. Для перетворення MAC-адрес в адреси мережевого рівня і назад застосовуються спеціальні протоколи (наприклад, ARP і RARP в мережах TCP/IP).

Адреси типу MAC-48 найпоширеніші, вони використовуються в таких технологіях, як ETHerneT, ToKen ring, FDDI тощо. Вони складаються з 48 бітів, таким чином, адресний простір MAC-48 налічує 248 (або 281 474 976 710 656) адрес. Згідно з підрахунками IEEE, цього запасу вистачить щонайменше до 2100 року.

EUI-48 відрізняється від MAC-48 лише семантично: якщо MAC-48 використовується для мережевого устаткування, то EUI-48 застосовується для інших типів апаратного і програмного забезпечення. Ідентифікатори EUI-64 складаються з 64 бітів і використовуються в FireWire, а також в IPv6 як молодші 64 біт мережевої адреси вузла. MAC-адреса – унікальний серійний номер пристрою, що однозначно ідентифікує його в мережі. MAC-адреса має довжину 6 байт і звичайно записується в шістнадцятковому вигляді, наприклад, 12:34:56:78:90:AB. Двокрапки можуть бути відсутні, але їхня наявність робить число більш читабельним. Кожен виробник привласнює адреси з приналежного йому діапазону адрес. Перші три байта адреси визначають виробника. У випадку виявлення двох пристроїв, у одній мережі з однаковою, MAC-адресою, що досить малоймовірно, необхідно змінити пристрій на інший або адресу пристрою за допомогою відповідних програм налагодження виробника.

У комп'ютерних мережах використовують поняття рівнів доступу до ресурсів. Рівні доступу застосовуються для клієнтів системи, користувачів баз даних чи програмного забезпечення, до клієнтів в системах клієнт-сервер. В комп'ютерних системах рівні доступу визначають адміністратори цих систем. В загальному варіанті використовується три рівні доступу [6]:

– найнижчий рівень доступу (нові клієнти; клієнти з неперевіреними чи невизначеними ідентифікаторами; клієнти, що отримують разовий доступ до відкритої інформації тощо);

– середній рівень доступу («постійні» клієнти системи, клієнти з визначеними перевіреними ідентифікаторами);

– високий рівень доступу (адміністратори системи, клієнти-аудитори системи, клієнти з управлінськими функціями тощо).

1.3 Визначення основних параметрів захисту інформації в комп'ютерній мережі

Інформація є одним з найбільш цінних ресурсів будь-якої компанії, тому забезпечення її захисту є одним з найважливіших завдань, адже в останні роки з розвитком комерційної і підприємницької діяльності збільшилося число спроб несанкціонованого доступу до конфіденційних даних [9].

Безпека інформаційної системи (ІС) – це властивість, що полягає в здатності системи забезпечити її нормальне функціонування, тобто забезпечити цілісність і секретність інформації.

Інформація, з точки зору інформаційної безпеки, наділена наступними характеристиками:

– конфіденційність – гарантія того, що конкретна інформація доступна тільки тому колу осіб, для кого вона призначена (порушення називається розкраданням або розкриттям інформації);

– цілісність – гарантія того, що інформація існує в її початковому вигляді, тобто при її зберіганні або передачі не було проведено несанкціонованих змін (порушення цієї характеристики називається фальсифікацією повідомлення);

– автентичність – гарантія того, що джерелом інформації є саме та особа, яка заявлена як її автор (порушення також називається фальсифікацією, але вже автора повідомлення);

– можливість апеляції – гарантія того, що при необхідності можна довести, що автором повідомлення є саме заявлена людина, і не може бути ніхто інший; відмінність цієї категорії від попередньої в тому, що при підміні автора, хтось інший намагається заявити, що він автор повідомлення, а при порушенні можливості апеляції – сам автор намагається приховати своє авторство.

У відношенні до інформаційних систем застосовуються інші категорії:

– надійність – гарантія того, що система поводиться в нормальному і позаштатного режимах так, як заплановано;

– точність – гарантія точного і повного виконання всіх команд;

– контроль доступу – гарантія того, що різні групи осіб мають різний доступ до інформаційних об'єктів і ці обмеження доступу постійно виконуються;

– контрольованість – гарантія того, що в будь-який момент може бути проведена повноцінна перевірка будь-якого компонента програмного комплексу;

– контроль ідентифікації – гарантія того, що клієнт, підключений у даний момент до системи, є саме тим, за кого себе видає;

– стійкість до умисним збоїв – гарантія того, що при умисному внесенні помилок система буде вести себе так, як обумовлено заздалегідь.

Для забезпечення цілісності і конфіденційності інформації необхідно забезпечити захист інформації від випадкового знищення або несанкціонованого доступу до неї [30].

Під захистом інформації від несанкціонованого доступу розуміється отримання доступу до ресурсів інформаційної системи шляхом виконання трьох процедур: ідентифікації, аутентифікації та авторизації.

Ідентифікація – привласнення користувачеві (об'єкту або суб'єктові ресурсів) унікальних імен і кодів (ідентифікаторів).

Аутентифікація – встановлення достовірності користувача, що представив ідентифікатор, або перевірка того, що особа або пристрій, що повідомив ідентифікатор є дійсно тим, за кого воно себе видає. Найбільш поширеним способом аутентифікації є надання користувачеві пароля і зберігання його в комп'ютері.

Авторизація – перевірка повноважень або перевірка права користувача на доступ до конкретних ресурсів і виконання певних операцій над ними. Авторизація проводиться з метою розмежування прав доступу до мережевих і комп'ютерних ресурсів.

Так як клієнти в комп’ютерній мережі отримують інформацію з сервера, від них надходять пакети-запити, на які сервер, в свою чергу, обробивши запит, повинен відправляти пакети-відповіді.

Запит – це формулювання своєї інформаційної потреби користувачем до

деякої бази даних. Для складання запиту використовується певний формат запиту. В загальних випадках в мережах біти формуються в пакети, тобто запити до сервера клієнти відправляють пакетами. Якість надходження таких пакетів також можна визначити. Як показник, що характеризує якість передачі пакетів, прийнято використовувати ймовірність прийому пакета з помилками або імовірність спотворення пакету (PacKeT Error RaTe, PER). Помилки в загальному випадку можуть призвести до різних наслідків. У деяких випадках пакети можуть бути втрачені, а в інших випадках – надходити не за призначенням. Втрата пакетів може відбуватися через технічні помилки при маршрутизації, внаслідок перевантажень або внаслідок втручання в роботу систем сторонніх осіб чи шкідливого програмного забезпечення (вірусів, програм перехоплення пакетів тощо). Імовірність втрати пакету (PacKeT Loss RaTe, PLR) є відношення кількості загублених пакетів до загальної кількості переданих за досить великий проміжок часу. Іноді пакети можуть надходити користувачеві, якому вони не призначені. Такі випадки назвиаються доставкою пакета не за адресою. Імовірність доставки пакета не за адресою (PacKeT InserTion RaTe, PIR) є кількість пакетів, доставлених не за адресою, за досить великий інтервал спостереження.

Природа цих помилок визначається засобами систем, в яких вони виникають. Помилки, залежні від систем передачі, визначаються в основному фізичним середовищем (коаксіальний кабель, волоконно-оптична лінія та ін.) і рядом інших факторів (видом кодування, адресування тощо).

Згідно із Законом України "Про інформацію" за режимом доступу інформація, поділяється на відкриту та з обмеженим доступом [34]. Відкрита інформація – це інформація, яка доступна для користування всіх, виходячи з ціни, зрозумілості і простоти у викладі. Ця інформація систематично публікується в офіційних друкованих виданнях (бюлетенях, збірниках), поширюється засобами масової комунікації, безпосередньо надається зацікавленим громадянам, державним органам та юридичним особам.

Інформація з обмеженим доступом – це інформація, яка має довірчий або секретний характер. У свою чергу, інформація з обмеженим доступом за своїм правовим режимом поділяється на конфіденційну, таємну, службову.

Конфіденційна інформація – це відомості, які знаходяться у володінні, користуванні та розпорядженні окремих фізичних або юридичних осіб і розповсюджуються за їх бажанням відповідно до передбачених ними умов.

У ст. 1 Закону України "Про державну статистику" дається таке визначення: "Конфіденційна інформація – статистична інформація, яка належить до інформації з обмеженим доступом і знаходиться у володінні, користуванні або розпорядженні окремого респондента та поширюється виключно за його згодою відповідно до погоджених з ним умов". До таємної інформації належить інформація, що містить відомості, які становлять державну та іншу передбачену законом таємницю, розголошення якої завдає шкоди особі, суспільству і державі [34].

В комп'ютерних системах інформацію можна аналогічно поділити на 2 групи за режимом доступу: відкрита інформація та інформація з обмеженим доступом. До групи з обмеженим доступом можна віднести конфіденційну інформацію, таємну інформацію та цілком таємну (або службову).

Службовою інформацією в комп'ютерних системах є інформація, до якої має доступ лише адміністратор системи; до цієї інформації відноситься інформація про функціонування системи, режими доступу до системи, розподілення прав доступу користувачів до інформаційних ресурсів, ідентифікатори (логіни, паролі) користувачів тощо.

Доступ до інформації з обмеженим доступом визначається заздалегідь розподілено між користувачами системи.

Захист інформації (англ. DaTa proTecTion) – сукупність методів і засобів, що забезпечують цілісність, конфіденційність і доступність інформації за умов впливу на неї загроз природного або штучного характеру, реалізація яких може призвести до завдання шкоди власникам і користувачам інформації.

Серед всього спектру методів захисту даних від небажаного доступу особливе місце займають криптографічні методи. Сучасні методи шифрування гарантують практично абсолютний захист даних, але завжди залишається проблема надійності їх реалізації. В даний час особливо актуальною стала оцінка вже використовуваних криптоалгоритмів.

Шифрування – це спосіб зміни повідомлення або іншого документа, що

забезпечує спотворення (захування) його вмісту. Кодування – це перетворення звичайного, зрозумілого, тексту в код (виконується без ключа). Шифрувати можна не тільки текст, але і різні комп'ютерні файли – від файлів баз даних і текстових процесорів до файлів зображень.

Ідея шифрування полягає в запобіганні прогляданню дійсного змісту повідомлення (тексту, файлу і т.п.) тими, у кого немає засобів його дешифрування. А прочитати файл зможе лише той, хто зможе його дешифрувати.

Шифрування з'явилося приблизно чотири тисячі років тому. Першим відомим застосуванням шифру (коду) вважається єгипетський текст, датований приблизно 1900 р. до н.е., автор якого використовував замість звичайних (для єгиптян) ієрогліфів не співпадаючі з ними знаки [25].

Криптографічний захист інформації – вид захисту інформації, що реалізується за допомогою перетворень інформації з використанням спеціальних даних (ключових даних) з метою приховування (або відновлення) змісту інформації, підтвердження її справжності, цілісності, авторства тощо.

Для сучасної криптографії характерне використання відкритих алгоритмів шифрування, що припускають використання обчислювальних засобів. Відомо більше десятка перевірених алгоритмів шифрування, які при використанні ключа достатньої довжини і коректної реалізації алгоритму, роблять шифрований текст недоступним для криптоаналізу. Широко використовуються такі алгоритми шифрування як TwofisH, IDEA, RC4 та ін.

Інформація, що може бути прочитана, осмислена і зрозуміла без яких-небудь спеціальних мір, називається відкритим текстом (plainTexT, clear TexT). Метод перекручування відкритого тексту таким чином, щоб сховати його суть, називається шифруванням (encrypTion або encipHering). Шифрування відкритого тексту приводить до його перетворення в шифртекст (cipHerTexT). Шифрування дозволяє сховати інформацію від тих, для кого вона не призначається, незважаючи на те, що вони можуть бачити сам шифртекст. Протилежний процес по звертанню шифртекста в його вихідний вид називається дешифруванням (decrypTion або decipHering) [25].

Криптографічна стійкість вимірюється тим, скільки знадобиться часу і

ресурсів, щоб із шифртексту відновити вихідний відкритий текст. Результатом стійкої криптографії є шифртекст, що винятково складно зламати без володіння визначеними інструментами по дешифруванню.

Криптографічний алгоритм, або шифр – це математична формула, що описує процеси шифрування і розшифрування. Щоб зашифрувати відкритий текст, криптоалгоритм працює в сполученні з ключем – словом, числом або фразою. Те саме повідомлення зашифроване одним алгоритмом, але різними ключами буде перетворюватися в різний шифртекст. Захищеність шифртекста цілком залежить від двох речей: стійкості криптоалгоритму і таємності ключа. У традиційній криптографії шифрування називається симетричним, коли той самий ключ використовується як для шифрування, так і для розшифрування даних. DaTa EncrypTion STandarT (DES) – приклад симетричного алгоритму, що широко застосовувався на Заході з 70-х років у банківській і комерційних сферах. В даний час його змінює Advanced EncrypTion STandard (AES) [31].

Симетричне шифрування має ряд переваг. Перше – швидкість даних криптографічних операцій. Однак, симетричне шифрування, використане саме по собі як засіб захисту даних, що пересилаються, може виявитися досить витратним просто через складність передачі таємного ключа. Для встановлення криптографічного зв'язку за допомогою симетричного алгоритму, відправникові й одержувачеві потрібно попередньо погодити ключ і тримати його в таємниці. Якщо вони знаходяться в географічно вилучених місцях, то повинні вдатися до допомоги довіреного посередника, наприклад, надійного кур'єра, щоб уникнути компрометації ключа в ході транспортування. Зловмисник, що перехопив ключ на шляху, зможе пізніше читати, змінювати і підробляти будь-яку інформацію, зашифровану або завірену цим ключем.

1.4 Постановка задачі

Клас систем, що описує принципи поведінки аналізованих моделей на основі

нечіткої логіки, порівняно молодий. Перші нечіткі множини були описані в роботах Лофті Заде в кінці 60-х рр. З тих пір суто математичне поняття завдяки працям Б. Коско перетворилося на самостійну концепцію і в новий підхід до вирішення багатьох завдань.

Системи, що реалізують механізми нечіткої логіки, в комерційному застосуванні з'явилися порівняно недавно. Але, звичайно ж, знайшли застосування в задачах керування і планування, причому фахівці швидко оцінили всі переваги такого підходу.

Діапазон можливих застосувань нечітких алгоритмів і систем управління, що реалізують ці алгоритми, надзвичайно широкий. Відомо, що NASA розглядає можливість застосування, а можливо, вже застосовує нечіткі системи для управління процесами стикування космічних апаратів. На іншому кінці цього діапазону знаходяться часто рекламовані пральні машини з Fuzzy Logic. Однак більш ґрунтовне застосування нечіткі системи все таки знаходять в традиційних областях управління промисловими об'єктами.

Одним з найзначніших застосувань нечіткого управління як за обсягом, так і по складності завдання є нечітка система управління доменною піччю.

Доменна піч повинна швидко реагувати на зміну виробничих планів і робочого режиму, при чому забезпечувати безперервне стабільне виробництво високоякісного чавуну. У зв'язку з цим обов'язковою умовою є належне підтримання нагріву, для чого необхідне постійне і точне управління цим процесом. Нагріванням печі можна управляти, змінюючи масу подачі, порядок завантаження матеріалів, обсяг, температуру, тиск та вологість гарячого дуття. Для цього фахівці на основі щоденних результатів роботи, показань великої кількості датчиків, що працюють в реальному часі, інформації, одержуваної по статистичній моделі, і практичних знань про роботу прогнозують нагрів і стан печі і передбачають оптимізацію завантаження сировини і нагріву дуття.

Система складається з керуючого комп'ютера (який здійснює збір даних від датчиків і їх попередню обробку для представлення у вигляді, зручному для логічних висновків), процесора штучного інтелекту (який, використовуючи базу знань, робить висновки про нагріванні печі) і цифрової контрольно-вимірювальної

апаратури, що управляє нагрівом за результатами висновків. Одна з проблем даної експертної системи, що містить емпіричні правила, уявлення нечіткостей в знаннях. Для її вирішення зазвичай використовують ступені достовірності виводу для кожного правила або нечіткі множини. При нечіткому управлінні, заснованому на теорії нечітких множин, з допомогою функцій належності, що дозволяють в природному вигляді представити суб'єктивні нечіткі поняття, властиві людині, описуються професійні знання кваліфікованого оператора і реалізується управління, аналогічне тому, яке він може виконувати. Разом з тим доменний процес є дуже складним процесом, в якому одночасно протікають реакції трьох фаз – газоподібної, твердої і рідкої. Тому професійні знання кваліфікованого оператора важко уявити тільки функціями належності і так само важко реалізувати керування у вигляді єдиної системи. У зв'язку з цим в даній системі в якості способу представлення обширних професійних знань використовували правила, а в якості засобу представлення нечіткостей ввели поняття теорії нечітких множин. Поряд з простотою представлення знань це дозволило уникнути збільшення числа правил і скоротити час виведення. У результаті з'явилася можливість оперативного управління в реальному часі на базі експертної системи. При цьому істотно підвищилися зручності технічного обслуговування [20].

На використання нечіткої логіки в автомобільній промисловості першими звернули увагу японські виробники. У 1981 році компанія «Nissan» вперше застосувала компоненти нечіткої логіки в системі управління п'ятиступінчастою коробкою передач, роком пізніше з'явилася аналогічна коробка на автомобілях «Honda», потім компанія «MiTsuBisHi MoTors» представила свою «розумну» коробку передач, компанія «RenaulT» спільно з «SieMens» розробила проактивну коробку перемикання передач, призначену для автомобіля «RenaulT Megane». На жаль, відкрита інформація про застосовувані в автомобільних системах контролерів з нечіткою логікою на сьогодні просто відсутня. Є тільки фрагментарна інформація про ідеї, закладені у цих системах. Перш за все, в системах нечіткого управління виконується аналіз манери водіння автомобіля, в основу якого покладені ідеї нечіткої логіки. У пам'ять комп'ютера закладається

лінгвістичний опис ступеня натискання педалі газу. Припустимо поняття «ступінь натиснення педалі газу велика». Якщо педаль натиснута на 60%, то це відповідає поняттю сильного натискання педалі на 40%. Виходячи з цього, виконується обмеження по рівню 40% функції приналежності, визначеною лінгвістичним описом манери водіння. Відповідно контролер визначає, на яких оборотах буде виконуватися перемикання передач. Для спокійної манери водіння перемикання виконується при менших оборотах, при спортивній – при великих. Використання нечіткої логіки дозволяє нечіткому контролеру визначати моменти, коли при скиданні газу потрібно увімкнути нижчу передачу, а коли ні, при цьому, під час аналізу використовується не один параметр (частота гальмування), а шість.

Типовий приклад системи, що добре піддається реалізації за допомогою нечіткої логіки, – АБС – антиблокувальна гальмівна система. Реалізацій АБС існує досить багато, але в загальному випадку управління виконується по двом вхідним параметрам: прослизанню колеса (відношення швидкості автомобіля до миттєвої лінійної швидкості точки на зовнішньому радіусі колеса відносно її центра) і реальному прискоренню колеса. У нечітких АБС обидва параметри представляються у вигляді логічних змінних з набором з 5-8 термів, на підставі яких контролер, використовуючи набір правил, кількість яких дорівнює добутку кількості термів вхідних змінних, отримує значення тиску в гальмовому циліндрі, прагнучи до підтримання оптимального прослизання [20].

У системах захисту інформації на даний час в основному застосовуються асиметричні криптосистеми. Але використання криптоалгоритмів у криптосистемах є в основному наперед визначеним. Тобто в системі використовується наперед закладений алгоритм шифрування.

Застосування апарату нечіткої логіки при створенні апаратно-програмного засобу для здійснення розподілу доступу в комп’ютерній системі шляхом вибору оптимального алгоритму шифрування для кожного окремого клієнта та врахування поточних параметрів самої системи дозволить забезпечити стійкість криптосистеми в режимі реального часу.

Для вирішення цього завдання необхідно:

– проаналізувати сучасні засоби захисту інформації в комп’ютерній

мережі;

– дослідити можливість застосування апарату нечіткої логіки для вибору крипто алгоритму;

– змоделювати нечітку систему вибору криптоалгоритму на основі механізму Мамдані;

– реалізувати контролер, що працює на основі розробленої нечіткої системи засобами Simulink;

– дослідити працездатність розробленого нечіткого контролера.

2 НЕЧІТКА СИСТЕМА ВИБОРУ МЕТОДУ ЗАХИСТУ ІНФОРМАЦІЇ В КОМП'ЮТЕРНІЙ МЕРЕЖІ

2.1 Сучасні алгоритми захисту інформації

DES (DaTa EncrypTion STandard) – симетричний блочний алгоритм шифрування, розроблений фірмою IBM і затверджений у США в 1977 році як офіційний стандарт (FIPS 46-3). DES має блоки по 64 біта і 16-циклічну структуру мережі Фейстеля, для шифрування використовується ключ довжиною 56 біт. Алгоритм використовує комбінацію нелінійних (S-блоки) і лінійних (перестановки E, IP, IP-1) перетворень. Алгоритм DES у своєму складі містить засекречені елементи, що з самого початку його використання породило велику кількість побоювань, оскільки вони могли давати можливість Національному Агентству Безпеки США неправомірного контролю [25].

DES – блочний алгоритм шифрування, вхідними даними для коду є блок розміром n біт і K-бітний ключ (рисунок 2.1). На виході, після застосування шифруючого перетворення, отримується n-бітний блок, при чому навіть незначна зміна вхідних даних призводить до істотних змін зашифрованого блоку. Блочні алгоритми шифрування реалізуються методом багаторазового застосування до блоків вхідних даних деяких базових перетворень.

Першим етапом DES обробки вхідних даних є вхідна перестановка. Дана обробка класифікується як просте перетворення над частинами одного блоку. Проста перестановка без ключа – один з найпростіших алгоритмів шифрування. Символи вхідних даних перемішуються за певним попередньо домовленим законом, що не розголошується.

Наступним етапом шифрування є основний цикл обробки даних – 16-ти циклічне перетворення мережею Фейстеля з використанням секретного ключа, що класифікується як складне перетворення над локальною частиною у блоці. Мережа Фейстеля – криптографічне перетворення над блоками, що являють собою ліву і праву половини регістру зсуву. Аргументами функції шифрування є 32-бітний вектор вхідної послідовності R_i – 1 і 48-бітний ключ, що є результатом попередньої

обробки 56-бітного заданого ключа. На останньому етапі зашифровані дані знову перемішуються вихідною перестановкою [18].

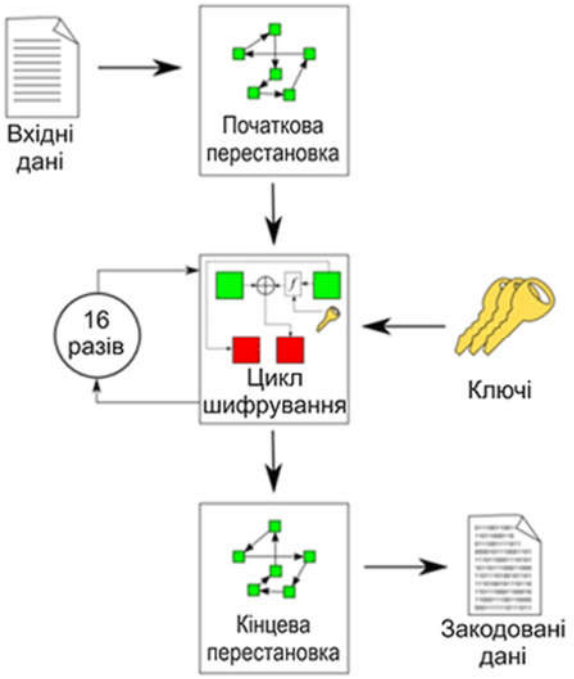


Рисунок 2.1 – Схема шифрування алгоритмом DES

В алгоритмі шифрування DES використовується пряме перетворення мережею Фейстеля при кодування і зворотне при декодуванні.

Алгоритм шифрування DES піддавався критиці за малу довжину ключа, що, врешті, не завадило йому стати загальноприйнятим стандартом. За історію свого існування алгоритм шифрування DES пережив декілька публічних криптографічних атак, що довели його невисоку надійність. Вперше код було розшифровано за допомогою використання мережі, що налічувала десятки тисяч комп'ютерів і на декодування знадобилося 39 днів. Пізніше, у 1998 у рамках досліджень DES CHallenge II, що проводила RSA LaBoraTory, алгоритм шифрування був зламано за допомогою суперкомп'ютера за 3 дні, що викликало значні побоювання щодо достатньої надійності міжнародного криптографічного стандарту шифрування. Остаточним доказом ненадійності DES стало публічне дешифрування коду у 1999 році, що зайняло лише 22 години 15 хвилин [18].

Зараз алгоритм шифрування DES вважається ненадійним в основному через

малу довжину ключа – 56 біт та розмір блоку – 64 біти. Вважається, що алгоритм шифрування достатньо надійний для застосування у модифікації.

3-DES є простим методом усунення недоліків DES – недостатньої криптостійкості. По суті дана модифікація шифрування є послідовним трьохциклічним DES з використанням 112- або 168-бітного ключа. Швидкість роботи даного алгоритму шифрування в три рази нижче, ніж у DES, але криптостійкість набагато краща, час, необхідний для криптоаналізу 3-DES, теоретично може в мільярд разів перевищити час злому попередника.

Хоча існують розроблені теоретичні атаки, про здійснені реальні розшифрування алгоритму 3-DES невідомо. Однак, низька швидкість та наслідування усіх інших недоліків алгоритму шифрування DES (наприклад, незручності для програмної реалізації, оскільки з самого початку алгоритм шифрування був розроблений для апаратної реалізації) зумовлюють неконкурентоспроможність алгоритму 3-DES порівняно з алгоритмом шифрування AES. Алгоритми DES та 3-DES поступово витісняється алгоритмом шифрування AES, що з 2002 року є стандартом США.

AES (Advanced EncrypTion STandard), також відомий під назвою Rijndael — симетричний алгоритм блочного шифрування з розміром блока 128 біт і ключем 128/192/256 біт. У результаті жорстокого відбору у рамка конкурсу AES, що проводився урядом США починаючи з 1997 року, був визнаний найкращим і прийнятий як державний стандарт шифрування Сполучених Штатів у 2002 році (FIPS 197). Розроблений Вінсентом Рейменом і Йоаном Дейменом алгоритм шифрування Рейндол найкраще відповідав висунутим умовам конкурсу [18].

По суті, алгоритм шифрування, запропонований авторами, і AES не є одне і те ж саме. Алгоритм шифрування Рейндол підтримує широкий діапазон розміру блоку та ключа. Алгоритм AES має фіксовану довжину у 128 біт, а розмір ключа може приймати значення 128, 192 або 256 біт. В той час як алгоритм Рейндол підтримує розмірність блоку та ключа із кроком 32 біт у діапазоні від 128 до 256. Через фіксований розмір блоку алгоритм шифрування AES оперує із масивом 4×4 байт, який називається станом (версії алгоритму із більшим розміром блоку мають додаткові колонки) (рисунок 2.2).

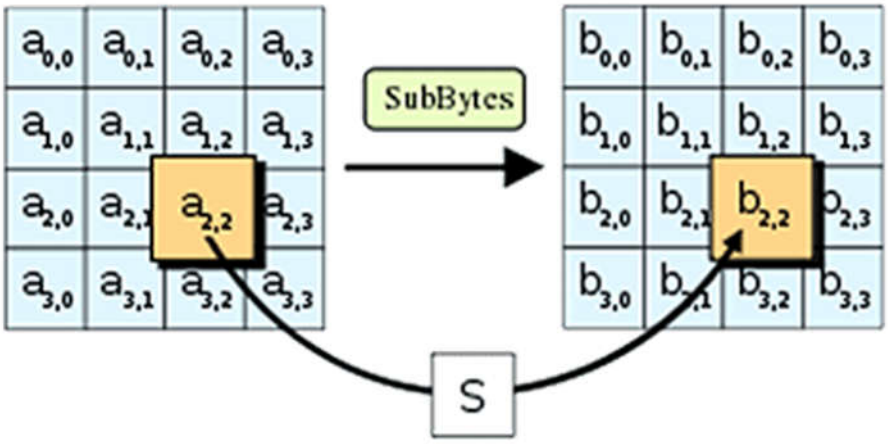


Рисунок 2.2 – Блочне шифрування AES

По принципу роботи алгоритм шифрування AES – підстановочно-перестановочна мережа. Особливістю криптографічного алгоритму AES є проста та доступна програмна реалізація, що розповсюджується у вигляді бібліотеки функцій. У складі бібліотеки основна функція шифрування та 8 допоміжних, а також оголошуються масиви змінних. У ході шифрування алгоритмом AES вхідні дані і ключ вносяться у таблиці (квадратні масиви), тобто стани, і трансформуються за допомогою перестановок, зсувів та взаємних перенесень за певними визначеними законами в декілька кроків (раундів).

Незважаючи на відкритість коду специфіка алгоритму шифрування AES не дозволяє декодувати таємні дані за прийнятний час. У червні 2003 року Агентство національної безпеки США постановило, що алгоритм AES з довжиною ключа 128 біт є достатньо надійним, щоб використовувати його для захисту інформації, що становить державну таємницю, а для найвищого рівня TOP SECRET – AES з ключем 192/256 біт.

На відміну від більшості інших алгоритмів шифрування AES має досить простий математичний опис. Це викликало досить негативні відгуки у наукових колах. Багато вчених висловлювали побоювання щодо безпечності алгоритму шифрування AES, що ґрунтується на неперевіреному стверджуванні про важкість розв’язання певних видів рівнянь, на яких базується код. Гіпотези і навіть їх доведення декілька разів публікувалися в наукових журналах. Так, наприклад, у роботі Ніколя Картуа і Йозефа Пепшика у 2002 році була описана теоретична

процедура під назвою XSL-атака, що могла б дозволити зламати алгоритм шифрування AES. Тим не менше ці дані не були підтверджені на практиці, тому не викликали значного резонансу. Через декілька років іншими дослідниками було доведено, що в описаному вигляді XSL-атака на алгоритм AES не може бути здійснена [36].

Значний удар по репутації стійкості алгоритму шифрування AES завдали так звані атаки по сторонніх каналах. Це процедури зламу коду, що не базуються на недоліках в його математичній моделі, а використовують специфіку реалізації захищеного протоколу. У 2005 році Даніель Бернштейн опублікував роботу з описом атаки, що використовує для злому інформацію про час виконання кожної операції шифрування. Для здійснення вдалої атаки знадобилося понад 200 мільйонів вибраних шифрованих текстів. У тому ж році Даг Арне Освік, Аді Шамір і Еран Трумер представили роботу з описом декількох аналогічним методик, одна з яких підбирала ключ лише за 800 циклів шифрування. Пізніше, у 2009 році були оприлюднені результати роботи по використанню диференціального аналізу помилок, що дозволило отримати ключ всього за 232 операції шифрування.

Станом на сьогодні AES є одним з самих розповсюджених алгоритмів симетричного шифрування. Підтримка криптографічного алгоритму AES на апаратному рівні введена фірмою InTel в сімейство високопродуктивних процесорів сімейства Sandy Bridge. Алгоритм шифрування використовується в сучасних захищених мережевих протоколах і платних сервісах.

Перший алгоритм кодування з відкритим ключем (puBlic-Key encrypTion (PKE) було запропоновано Вітфілдом Діффі та Мартіном Хелманом у Стенфордському університеті. Перевага PKE полягає у відсутності потреби секретної передачі ключа [37].

PKE базується на нерозв'язності проблеми розкладу натурального числа на прості множники.

RSA схему шифрування було запропоновано у 1978 році та названо іменами трьох його винахідників: Роном Рівестом, Аді Шаміром та Леонардом Адлеманом. RSA належить до класу алгоритмів кодування з відкритим ключем.

У 80-х роках криптосистема переважно використовувалася для забезпечення

секретності та достовірності цифрових даних. У сучасному світі RSA використовується в weB-серверах та браузерах для зберігання таємності даних, що передаються по мережі.

Схема RSA базується на обчисленні виразів зі степенями. Відкритий текст шифрується блоками, довжина кожного із яких менша за деяке число n.

Алгоритм генерації ключа:

– адресат *A* повинен згенерувати відкритий та секретний ключі:

1. згенерувати два великих простих числа *p* та *q* приблизно однакової довжини;

2. обчислити $n=p*q, f_i=(p-1)*(q-1)$;

3. вибрати натуральне *e*, $1<e<f_i$, взаємно просте з f_i;

4. використовуючи розширений алгоритм Евкліда, розв'язати рівняння $d*e^{1(Mod\ fi)}$

Відкритий ключ: (*n, e*). Секретний ключ: *d*.

Схема шифрування RSA:

Одержувач *B* шифрує повідомлення *M* та надсилає *A*.

1. Шифрування. *B* робить наступні дії:

– отримає відкритий ключ (*n, e*) від *A*;
– представляє повідомлення у вигляді натурального числа *M* з проміжку [1..*n*];
– обчислює $c = Me \text{ Mod } n$;
– надсилає шифртекст *c* до *A*.

2. Дешифрування. Для отримання повідомлення *K* із шифртексту *A* робить наступні дії: використовуючи секретний ключ *d*, обчислює $M = c^d \text{ Mod } n$.

За допомогою китайської теореми про лишки можна прискорити процес дешифрування, знаючи секретні прості числа *p* та *q*. Це відбувається наступним чином.

А має декодуючу експоненту *d*, а також *p* та *q* ($n = p * q$). *A* отримує від *B* шифр *c* та повинен виконати операцію $c^d \text{ (Mod } n)$.

– Обчислити $dp = d \text{ Mod } (p - 1)$, $dq = d \text{ Mod } (q - 1)$
– Обчислити $Mp = \text{Mod } p$, $Mq = \text{Mod } q$.

– Розв'язати систему лінійних порівнянь

Розв'язком системи буде декодоване повідомлення: $M = c^d$ (Mod n).

Повідомлення M називається неприхованим, якщо його шифр дорівнює самому повідомленню, тобто $M^e = M$ (Mod n).

Наприклад, повідомлення M=0 та M=1 завжди є неприхованими для довільних значень e та M.

Кількість неприхованих повідомлень в RSA системі дорівнює

$$(1+НСД(e\text{-}1, p\text{-}1)) * (1+НСД(e\text{-}1, q\text{-}1)) \qquad (2.1)$$

Оскільки значення e-1, p-1 та q-1 – парні, то $НСД(e\text{-}1, p\text{-}1)^2$, $НСД(e\text{-}1, q\text{-}1)^2$, а отже кількість неприхованих повідомлень завжди не менша за 9.

Алгоритм RSA використовується, як правило, як інструмент електронного підпису.

Наприкінці звичайного листа або документа виконавець чи відповідальна особа зазвичай ставить свій підпис. Подібна дія зазвичай переслідує дві мети.

По-перше, одержувач має можливість переконатися в автентичності листа, звіривши підпис з наявним у нього зразком.

По-друге, особистий підпис є юридичним гарантом авторства документа.

Із поширенням у сучасному світі електронних форм документів (у тому числі і конфіденційних) і засобів їхньої обробки особливо актуальною стала проблема встановлення дійсності та авторства непаперової документації. За всіх переваг сучасних систем шифрування вони не дозволяють забезпечити аутентифікацію даних. Тому засоби аутентифікації повинні використовуватися в комплексі з криптографічними алгоритмами.

Алгоритм RSA є найбільш простим і розповсюдженим інструментом електронного підпису.

Застосування алгоритму RSA для формування електронного цифрового підпису (далі – ЕЦП) на прикладі обчислення і перевірки електронного підпису (S) повідомлення M:

Перший крок – обчислення хеш-повідомлення M=H(M), що потім

шифрується на секретному ключі K_s.

Для алгоритму *ЕЦП* RSA $S=MK_s$ Mod N.

Одержувач, що бажає перевірити значення S повідомлення M, також обчислює хеш повідомлення по формулі $M = H(M)$ і розшифровує S за допомогою відкритого ключа Kp, використовуючи асиметричний алгоритм шифрування RSA, відповідно до виразу $M' = SK_p$ Mod N.

Якщо M' = M, ЕЦП повідомлення визнається вірним. У противному випадку підпис вважається підробленим і робиться висновок про те, що цілісність повідомлення порушена.

Отже, у криптосистемі RSA секретний ключ використовується для обчислення ЕЦП чи для розшифрування повідомлень, а публічний – для перевірки ЕЦП чи зашифрування повідомлень.

Слід зазначити і ряд недоліків, властивих формуванню ЕЦП із використанням RSA, причому частина з них успадковані від використовуваного алгоритму шифрування RSA.

Серед останніх варто згадати про те, що ЕЦП RSA уразлива до мультиплікативної атаки, тобто алгоритм ЕЦП RSA дозволяє зловмиснику, навіть не знаючи секретний ключ K_s, обчислити підпис повідомлень, результат хешування яких збігається з добутком результатів хешування підписаних раніше повідомлень.

Найбільша швидкість реалізації RSA в 1000 разів повільніша, ніж DES. В 1989 році найбільші швидкості VKSI- реалізації дорівнювали приблизно 64 Кб/с. Зараз швидкість ~1 Мб/с (для порівняння швидкість DES – від 10 до 100 Мб/с) [24].

Числа можуть змінюватись, але швидкість RSA ніколи не досягне швидкості симетричних алгоритмів. Ось чому на практиці більшість систем використовує RSA виключно для обміну DES-івськими ключами, а потім шифруються повідомлення DES-ом. Такий підхід називається гібридними криптосистемами.

За останні роки поняття еліптичних кривих (ellipTic curve (EC)) знайшло своє застосування у криптографії. Причиною цього є те, що еліптичні криві над скінченими полями утворюють скінчені групи, на яких (навіть для кривих досить великих розмірів) легко визначити арифметичні операції завдяки багатій структурі груп. Дотепер у криптографії працювали з мультиплікативними групами над

деякими скінченими полями. За своїми властивостями еліптичні криві дещо нагадують ці групи, але їхня перевага полягає в тому, що існує більша свобода вибору еліптичної кривої, ніж вибору скінченого поля. Крім того, еліптичні криптосистеми забезпечують кращий захист інформації [39].

У криптографії використовуються групи еліптичних кривих над полем Zn.

Група еліптичних кривих над полем Zn визначається рівнянням

$$y^2 \,(\text{Mod } n) = x^3 + ax + B \,(\text{Mod } n), \qquad (2.2)$$

де a,$B \in Zn$ та $a^3+27B^2 \neq 0(\text{Mod } n)$, разом з точкою O, що називається точкою на нескінченості (n – просте число, більше за 3). Множина $E(Zn)$ складається з усіх точок $x \in Zn$, $y \in Zn$, що задовільняють рівняння (2.2), включаючи точку O.

Арифметичні операції над точками еліптичної кривої визначені наступним чином:

1. $P+O=O+P=P$ для всіх $P \in E$ (E – еліптична крива);
2. Якщо $P=(x,y) \in E$, тоді $P+(-P)=0$, де $-P=(x,-y)$;
3. Нехай $P=(x_1,y_1)$, $Q=(x_2,y_2) \in E$, де $P \neq -Q$

Тоді $P+Q=(x_3,y_3)$, де $x_3=\lambda^2-x_1-x_2(\text{Mod } n)$, $y_3=\lambda(x_1-x_3)-y_1(\text{Mod } n)$,

$$\lambda = \frac{y_2 - y_1}{x_2 - x_1}(\text{mod } n), \quad \text{якщо } P \neq Q,$$

$$\lambda = \frac{3x_1^{\,2} + a}{2y_1}(\text{mod } n), \quad \text{якщо } P = Q.$$

Введена таким чином операція додавання надає еліптичним кривим структури абелевої групи, що дає можливість використовувати їх у криптосистемах.

Як правило, еліптичних криві використовуються у схемі вдосконаленого алгоритму цифрового підпису ECDSA (ellipTic curve digiTal signaTure algoriTHM). Цей алгоритм був затверджений як стандарт ANSI X9.62 в 1997р., і він є в певному сенсі аналогом алгоритму DSA (digiTal signaTure algoriTHM), але з використанням еліптичних кривих.

Алгоритм ECDSA успадкував від DSA ту особливість, щовін базується на схемі цифрового підпису Ель Гамала і використовує те саме підписуюче рівняння:

$$S=K^{-1}\{H(M)+xr\}\text{Mod } n. \qquad (2.3)$$

В обох алгоритмах величини, що важко генеруються, є системними параметрами, вони є загальновідомими і генерувати їх можна незалежно. В сучасній версії DSA, як і в ECDSA, використовують SHA-1 (secure HasH algoriTHM).

Проте, існують вагомі переваги ECDSA над DSA. По-перше, приватний ключ d та число K в ECDSA є статистично унікальними і непередбачуваними, а не лише випадковими, як в DSA, що покращує надійність алгоритму. Крім того, завдяки складності проблеми дискретного алгоритму систему ECDSA важче зламати.

В ECDSA метод “стиснення точки” дозволяє компактно представити точку на кривій за допомогою одного елемента поля (а не двох) та одного додаткового біта. Наприклад, якщо $p=2^{160}$ (це означає, що елементи поля Zn є 160-бітовими рядками), тоді відкритий ключ можна представити 161-бітовим рядком. Це приводить до суттєвого зменшення розміру відкритого ключа (якщо порівнювати з іншими асиметричними алгоритмами, виграш становить приблизно 25%).

2.2 Нечітка система на основі механізму Мамдані

Розробка нечіткого контролера вибору криптоалгоритму в комп’ютерній мережі полягає у використанні апарату нечіткої логіки.

Основи нечіткої логіки були закладені наприкінці 60-х років у працях відомого американського математика Латфі Заде. Соціальне замовлення на дослідження подібного роду було викликано зростаючим незадоволенням експертними системами. "Штучний інтелект", що легко справлявся із задачами керування складними технічними комплексами, був безпорадним при найпростіших висловленнях повсякденного життя, типу "Якщо машиною перед тобою керує недосвідчений водій – тримайся від неї подалі". Для створення дійсно

інтелектуальних систем, здатних адекватно взаємодіяти з людиною, необхідний був новий математичний апарат, що переводить невиразні і неоднозначні життєві твердження в мову чітких і формальних математичних формул [40].

Першим серйозним кроком у цьому напрямку стала теорія нечітких множин, розроблена Заде. Його робота "Fuzzy SeTs", що з'явилася в 1965 році в журналі "InforMaTion and ConTrol", заклала основи моделювання інтелектуальної діяльності людини і явилася початковим поштовхом до розвитку нової математичної теорії. Він же дав і назву для нової області науки – "fuzzy logic" (fuzzy нечіткий, розмитий, м'який).

Апарат теорії нечітких множин, продемонструвавши ряд багатообіцяючих можливостей застосування – від систем керування літальними апаратами до прогнозування підсумків виборів, виявився разом з тим надмірно складним для втілення, враховуючи наявний на той час рівень технології.

Своє друге народження теорія нечіткої логіки пережила на початку вісімдесятих років, коли відразу кілька груп дослідників (в основному в США і Японії) всерйоз зайнялися створенням електронних систем різного застосування, що використовують нечіткі керуючі алгоритми. Теоретичні основи для цих спроб були закладені в ранніх працях Коско й інших учених.

Третій період почався з кінця 80-х років і дотепер. Цей період характеризується бумом практичного застосування теорії нечіткої логіки в різних сферах науки і техніки. До 90-го року з'явилося близько 40 патентів, що відносяться до нечіткої логіки (30 з яких – японських). Сорок вісім японських компаній утворили спільну лабораторію LIFE (LaBoraTory for InTernaTional Fuzzy Engineering), японський уряд фінансував 5-річну програму по нечіткій логіці, що включає 19 різних проектів – від систем оцінки глобального забруднення атмосфери і передбачення землетрусів до автоматизованих систем управління заводських цехів і складів. Результатом виконання цієї програми з'явилася поява цілого ряду нових масових мікрочіпів, заснованих на нечіткій логіці. Сьогодні їх можна знайти в пральних машинах і відеокамерах, цехах заводів і моторних відсіків автомобілів, у системах керування складськими роботами і бойовими гелікоптерами.

У США розвиток нечіткої логіки йде по шляху створення систем, що потрібні великому бізнесу і військовим. Нечітка логіка застосовується при аналізі нових ринків, біржовій грі, оцінці політичних рейтингів, виборі оптимальної цінової стратегії і т.п. З'явилися і комерційні системи масового застосування.

Найбільш важливим застосуванням є контролери нечіткої логіки. Їх функціонування дещо відрізняється від роботи звичайних контролерів; для опису системи замість диференційних рівнянь використовуються знання експертів. Ці знання можуть бути виражені за допомогою лінгвістичних змінних, які описані нечіткими множинами.

В інженерних задачах застосовується, як правило, механізм нечіткого висновку Мамдані [37]. В ньому використовується мінімаксна композиція нечітких множин. Даний механізм включає наступну послідовність дій [41]:

1) процедура фазифікації: визначаються степені істинності, тобто значення функцій належності $MF_i(x)$ для лівих частин кожного і-го правила (передумов);

2) нечіткий висновок. Спочатку визначаються мінімальний рівень "відсічення" для лівої частини кожного з правил $A_i = \min(MF_i(x))$, а потім знаходяться "усічені" функції належності висновку $B_i = \min(A_i, B_i)$;

3) композиція або об'єднання отриманих "усічених" функцій, для чого використовується композиція нечітких множин $MF(y) = \max(B_i(y))$;

4) дефазифікація або приведення до чіткості. Існує декілька методів дефазифікації. Наприклад, метод середнього центру або центроїдний метод. Геометричний зміст такого значення – центр ваги для кривої функції належності отриманого виходу.

На рисунку 2.3 зображено логічний висновок за механізмом Мамдані на прикладі двох правил R1 та R2, в яких знаходяться мінімальні площі в зображеннях функцій належності трьох змінних, після чого здійснюється об'єднання усічених площ за максимальним законом і, нарешті, знаходиться центр ваги остаточної фігури, абсциса якого і є висновком нечіткої системи [37].

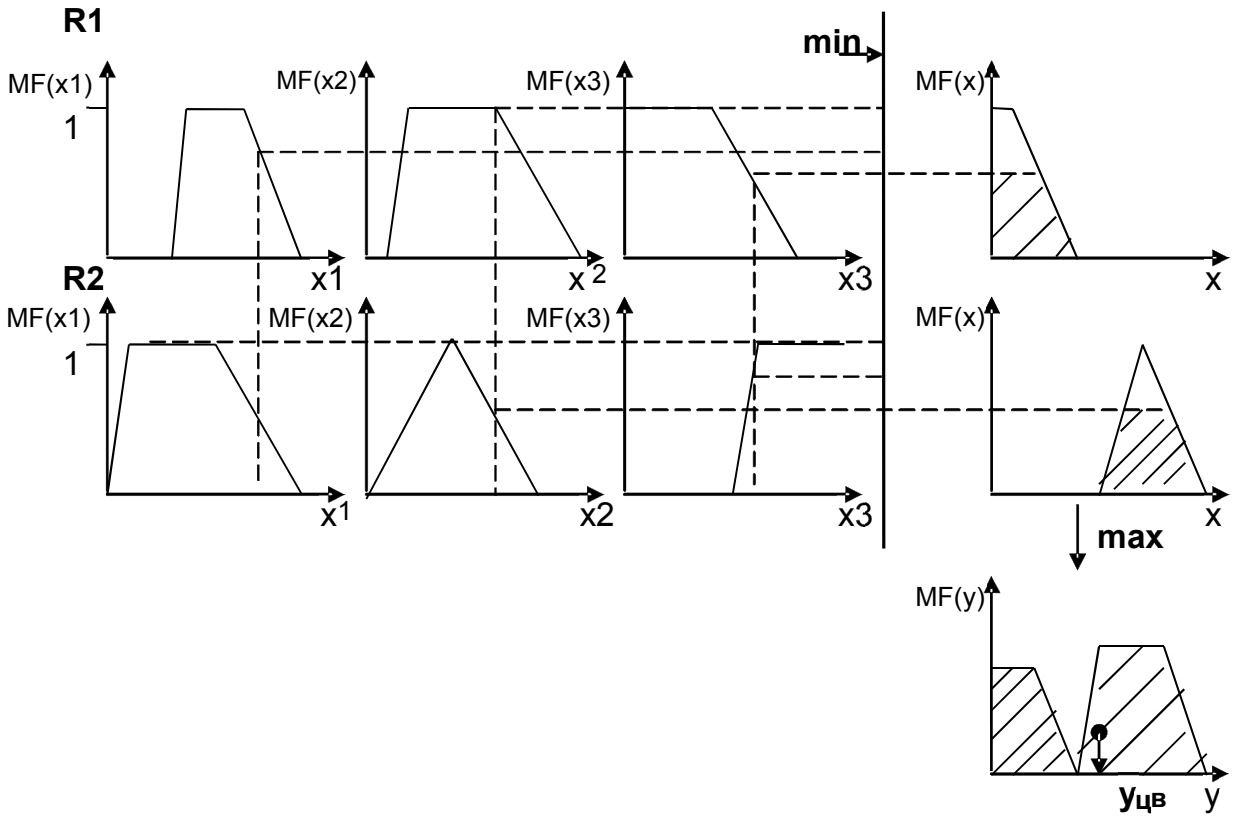


Рисунок 2.3 – Механізм нечіткого висновку Мамдані

На блок оброблення інформації сервера системи захисту поступає нечітка інформація про необхідний рівень продуктивності та рівень доступу клієнта.

В якості експертної оцінки вхідних даних можна застосувати значення продуктивності, яке належить відрізку від 0 до 10^5 тактів, а значення рівня доступу – від 0 до 3. Відповідно до вхідних нечітких значень рівня доступу клієнта та необхідного рівня продуктивності система видає значення, яке відповідає необхідному для застосування криптоалгоритму, а саме: DES, RSA чи на основі еліптичних кривих (EC) (рисунок 2.4).

Застосовуючи засіб Fuzzy Logic ToolBox середовища MATLAB 7.7.0 (R2008B) [40], можна побудувати нечітку систему вибору криптографічного алгоритму (algoriTHM – криптоалгоритм) залежно від значень продуктивності (perforMance), та рівня доступу клієнта (access).

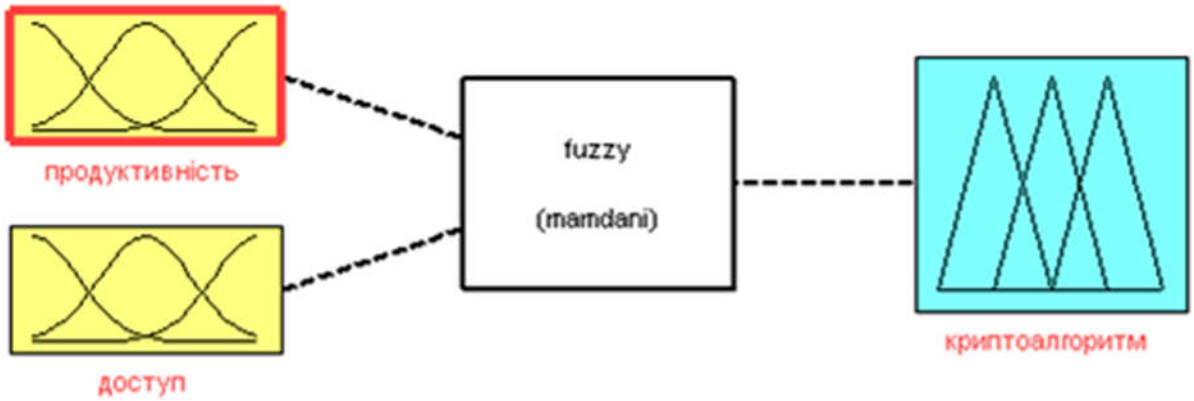


Рисунок 2.4 – Загальна схема нечіткої системи вибору криптоалгоритму

Значення функцій належності вхідної змінних продуктивність та вихідної криптоалгоритм задається трапецевидною функцією, що визначається четвіркою чисел (a,B,c,d), які позначають абсциси вершин трапеції,

$$MF(x)=\begin{cases}\dfrac{b-x}{b-a}, a\le x\le b\\ 1, b\le x\le c\\ \dfrac{x-c}{d-c}, c\le x\le d\\ 0, \text{в інших випадках}\end{cases}, \qquad (2.4)$$

а вхідної змінної доступ – дзвоноподібною функцією [12]

$$MF(x)=\frac{1}{1+\left|\dfrac{x-c}{a}\right|^{2b}}, \qquad (2.5)$$

яка задається трьома числами (a,B,c), що відповідають абсцисам крайніх точок та центру кривої.

Моделювання нечіткого висновку [44] здійснюється по типу Мамдані [40], описаному вище (див. п.1.4).

Функції належності для змінних доступ та продуктивність, подані на рисунках 2.5 та 2.6 відповідно.

Вони поділені на три інтервали кожна для точного опису змінних, зокрема, для опису рівня доступу застосовується змінні низький ∈ [0 , 1.4], що позначає низький рівень доступу клієнта, середній ∈ [0.6 , 2.4] – сер едній рівень та високий ∈ [1.6 , 3] – в исокий рівень (див.рисунок 2.7).

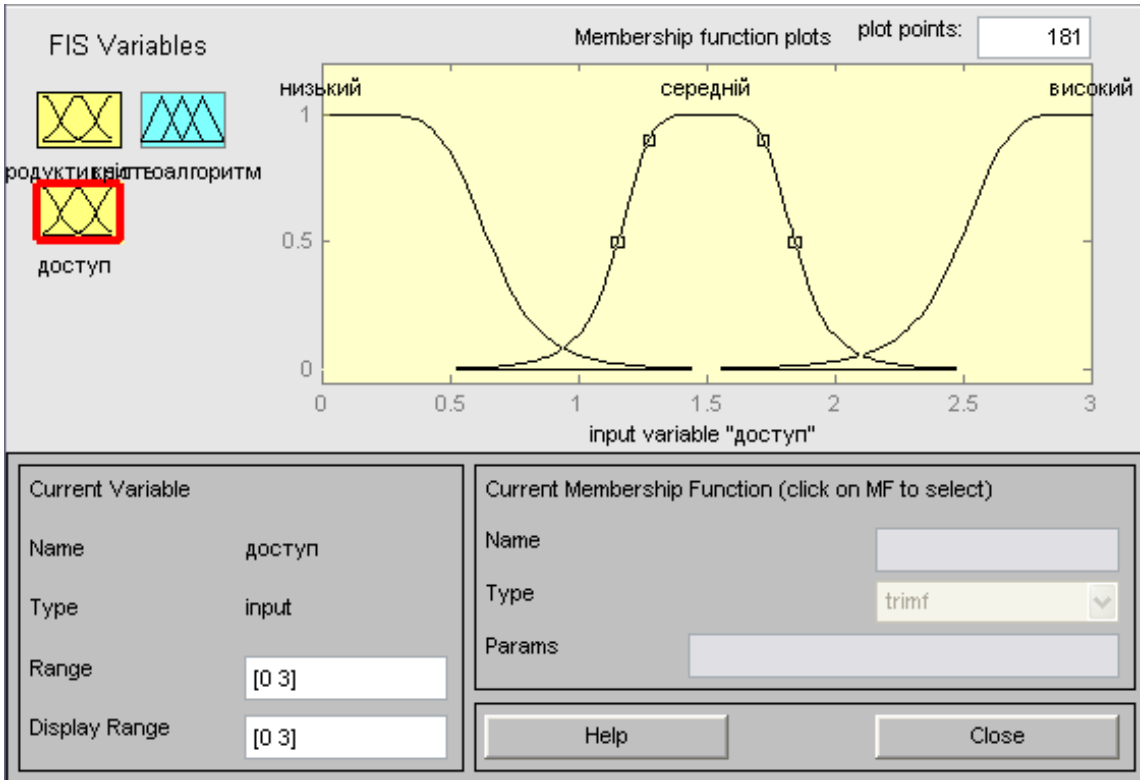


Рисунок 2.5 – Функції належності змінної доступ

Для задання продуктивності пропонуються змінні висока ∈ [0,30000] середня ∈ [20000,80000] та низька ∈ [70000,100000] (див.рисунок 2.6).

Функції належності для вихідної змінної алгоритм зображено на рисунку 2.7. Вони позначаються однаковими інтервалами на осі ординат для точного визначення центру ваги, що позначає нечіткий висновок системи [46]. None позначає відсутність криптоалгоритму, DES, RSA та EC – криптоалгоритми DES, RSA та на основі еліптичних кривих, відповідно.

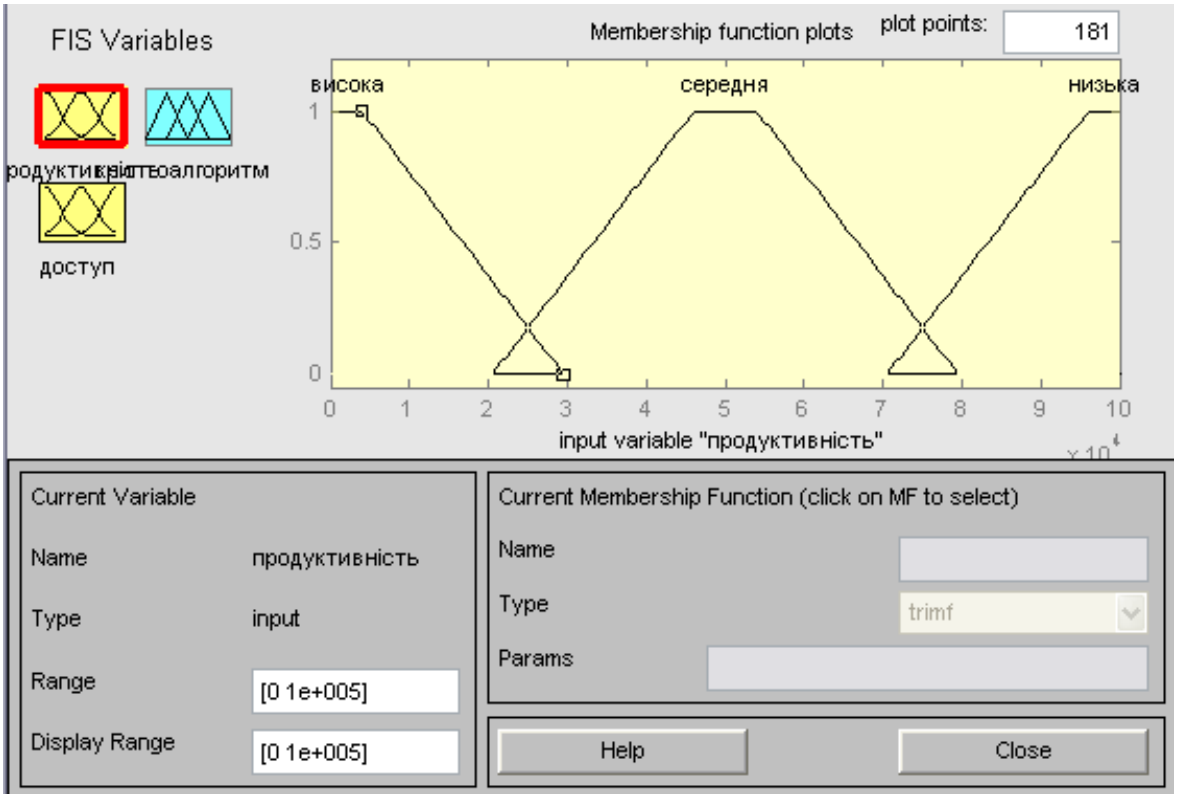


Рисунок 2.6 – Функції належності змінної продуктивність

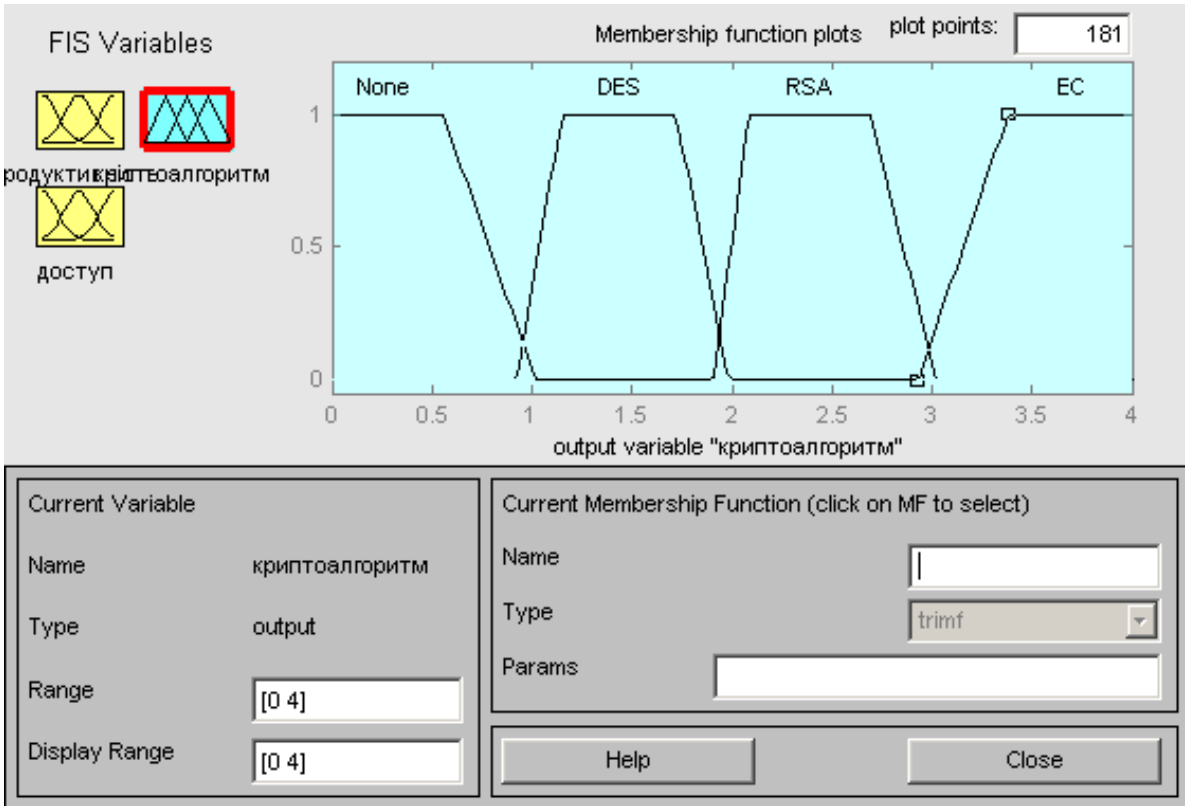


Рисунок 2.7 – Функції належності змінної криптоалгоритм

База знань для побудови даної нечіткої моделі складається з правил типу

«якщо – то» [11], усі вхідні змінні мають по три нечітких стани і ще один стан none, коли значення вхідної змінної не задане системою. Випадок, коли значення усіх вхідних змінних не задані, на практиці неможливий, тому кількість правил нечіткого висновку досліджуваної системи N=4*4-1=15.

Система правил нечіткого висновку:

1. If (продуктивність is висока) and (доступ is низький) THen (криптоалгоритм is RSA) (1)
2. If (продуктивність is висока) and (доступ is середній) THen (криптоалгоритм is DES) (1)
3. If (продуктивність is висока) and (доступ is високий) THen (криптоалгоритм is None) (1)
4. If (продуктивність is середня) and (доступ is низький) THen (криптоалгоритм is RSA) (1)
5. If (продуктивність is середня) and (доступ is середній) THen (криптоалгоритм is DES) (1)
6. If (продуктивність is середня) and (доступ is високий) THen (криптоалгоритм is None) (1)
7. If (продуктивність is низька) and (доступ is низький) THen (криптоалгоритм is EC) (1)
8. If (продуктивність is низька) and (доступ is середній) THen (криптоалгоритм is RSA) (1)
9. If (продуктивність is низька) and (доступ is високий) THen (криптоалгоритм is DES) (1)
10. If (доступ is низький) THen (криптоалгоритм is EC) (1)
11. If (доступ is середній) THen (криптоалгоритм is RSA) (1)
12. If (доступ is високий) THen (криптоалгоритм is DES) (1)
13. If (продуктивність is висока) THen (криптоалгоритм is DES) (1)
14. If (продуктивність is середня) THen (криптоалгоритм is RSA) (1)
15. If (продуктивність is низька) THen (криптоалгоритм is EC) (1)

2.3 Дослідження побудованої нечіткої системи

Нечіткий висновок моделі вибору методу криптоалгоритму, побудованого на основі заданих 15 правил з поточними значеннями змінних продуктивність та доступ має вигляд, представлений на рисунку 2.8 [17].

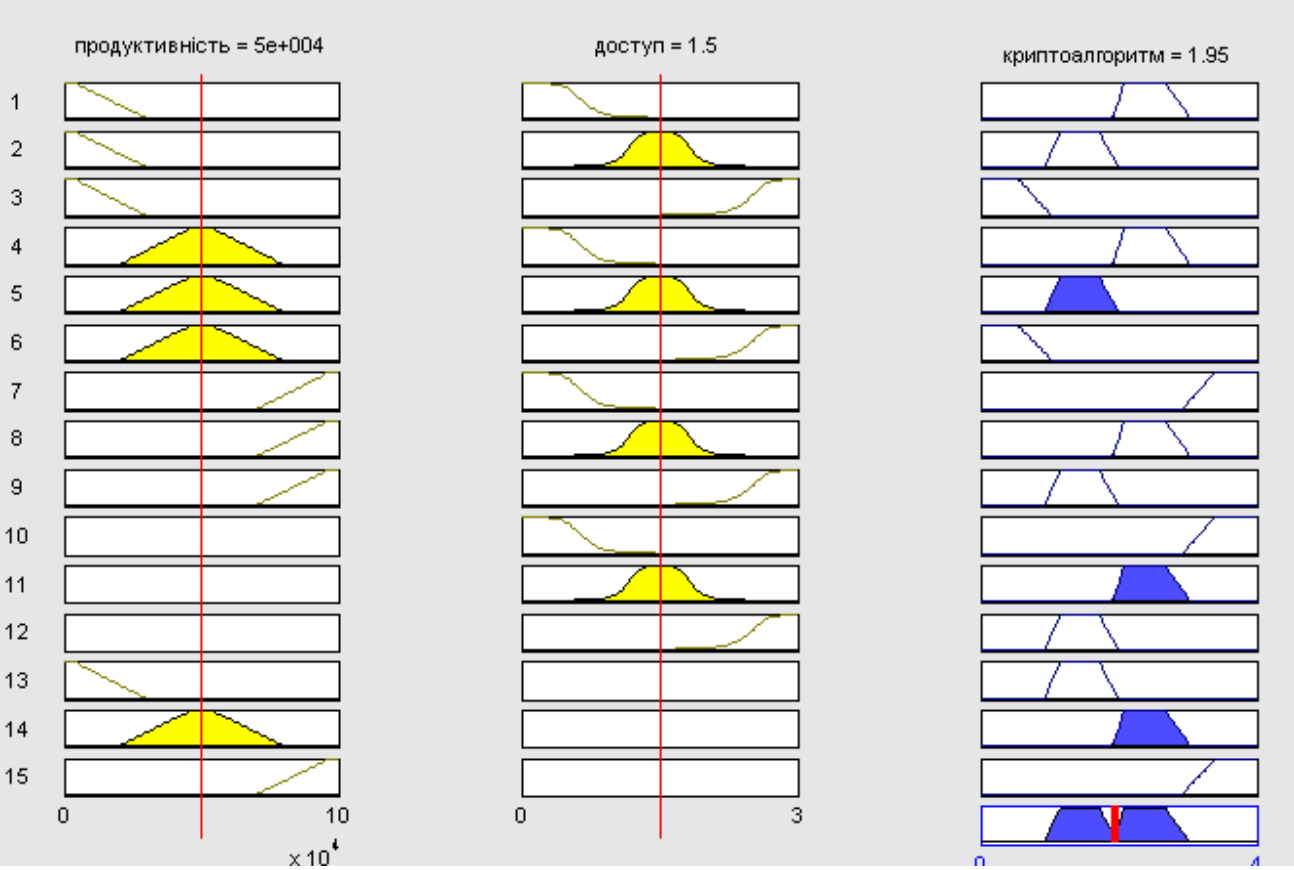


Рисунок 2.8 – Нечіткий висновок моделі вибору методу криптоалгоритму

Поверхня значень нечіткої системи на основі механізму Мамдані подана на рисунку 2.9 [17]. Вона підтверджує правильність побудови бази правил нечіткого висновку.

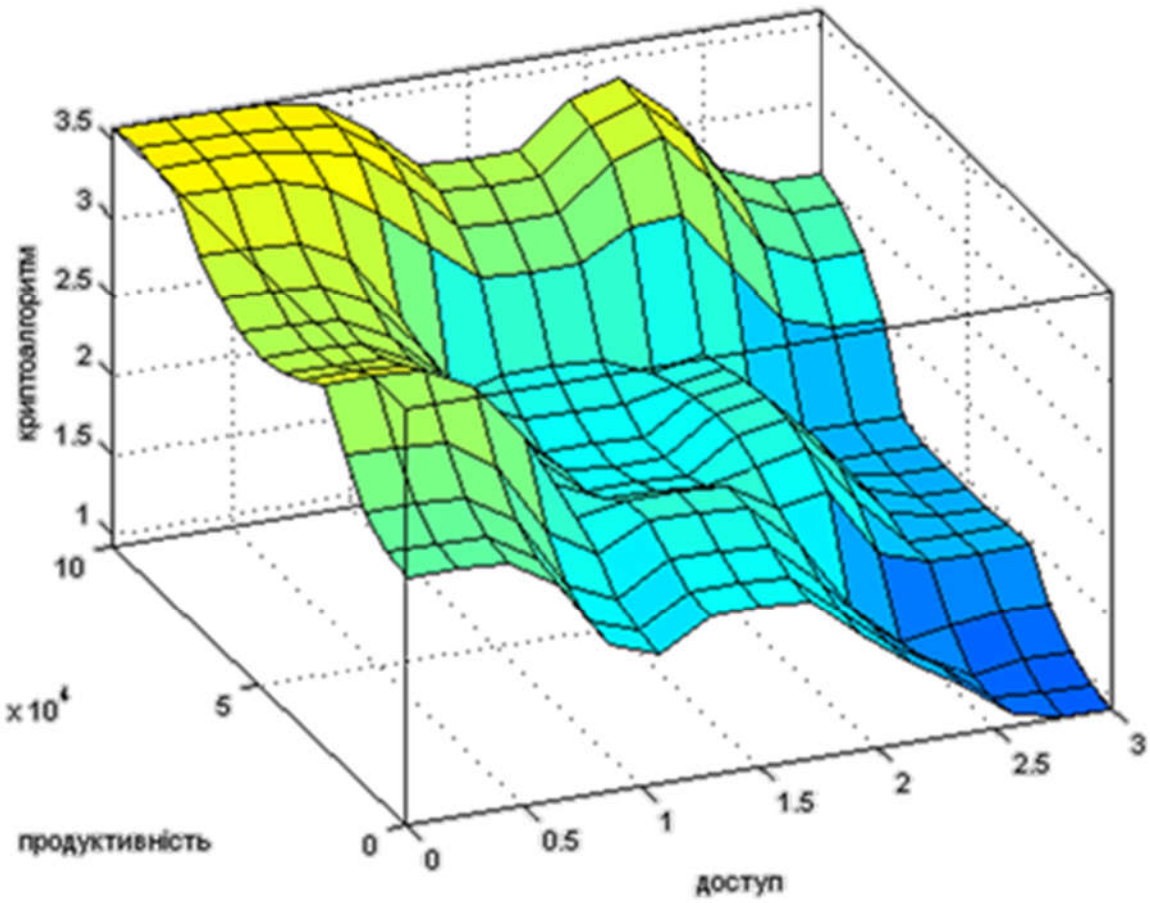


Рисунок 2.9 – Поверхня значень виходу нечіткої системи
на основі механізму Мамдані

Отже, побудована нечітка система вибору криптоалгоритму дозволяє за поточними значеннями рівня доступу клієнта та необхідного рівня продуктивності комп'ютерної системи вибрати адекватний в даному випадку алгоритм шифрування інформації.

Для практичної реалізації системи вибору криптоалгоритму доцільно побудувати нечіткий контролер.

3 НЕЧІТКИЙ КОНТРОЛЕР ВИБОРУ МЕТОДУ ЗАХИСТУ ІНФОРМАЦІЇ

3.1 Побудова нечіткого контролера

Для реалізації розробленої нечіткої системи вибору методу захисту інформації в комп'ютерній мережі варто розробити контролер, який працює на нечіткій логіці.

Прикладом програмного забезпечення, що реалізує механізм нечіткої логіки є електронна таблиця – пакет Fuzi Calc фірми FuziWare.

Програмний пакет Fuzi Calc відноситься до добре відомого класу програм – електронних таблиць. Перш за все він призначений для зберігання даних і їх обробки, а також для виконання простих розрахунків і оцінок. Даний пакет абсолютно унікальний, оскільки заснований на нетрадиційних принципах (багатозначною логікою) та орієнтований на широке коло користувачів, не спокушених у сучасній математиці і програмуванні. А ще він дозволяє працювати з нечітко визначеними даними, як із звичайними числами.

Основна перевага пакета – робота з розмитими, невизначеними фінансовими даними. Але, звичайно ж, з цього однозначно випливають і його недоліки. Це випливає хоча б з тієї ж складної природи самих даних. Результати розрахунку Fuzi Calc є тільки наближенням, апроксимацією дійсності і можуть досить сильно відрізнятися від реальності або від інтуїтивних уявлень. Якщо придивитися уважніше, стає зрозуміло, що виникаючі проблеми пов'язані не стільки з пакетом, скільки з недостатністю наших уявлень про суть фінансової невизначеності, складності природи даних. Доводиться з цим миритися і використовувати для різних завдань різні наближення.

Не обійшли засоби нечіткої логіки і програмні системи, що обслуговують великий бізнес. Першими, зрозуміло, були фінансисти, задачі яких вимагають щоденного прийняття правильних рішень у складних умовах непередбаченого ринку.

Слідом за фінансистами, стурбовані успіхами японців і втратою стратегічної

ініціативи, когнітивними нечіткими схемами зацікавилися промислові гіганти США. MoTorola, General ElecTric, OTis ElevaTor, Pacific Gas & ElecTric, Ford і інші на початку 90-х почали інвестувати в розробку виробів, що використовують нечітку логіку.

Серед лідерів нового ринку виділяється американська компанія Hyper Logic, заснована в 1987 році Фредом Уоткінсом (Fred WaTKins). Спочатку компанія спеціалізувалася на нейронних мережах, однак незабаром цілком сконцентрувалася на нечіткій логіці. Недавно вийшла на ринок друга версія пакета CuBiCalc фірми HyperLogic, яка є однією з найбільш могутніх експертних систем на основі нечіткої логіки. Пакет містить інтерактивну оболонку для розробки нечітких експертних систем і систем керування, а також run-TiMe модуль, що дозволяє оформляти створені користувачем системи у виді окремих програм.

Крім Hyper Logic серед "патріархів" нечіткої логіки можна також назвати такі фірми як InTelligenceWare, InfraLogic, ApTronix. Усього ж на світовому ринку представлено більш 100 пакетів, які тим чи іншим видом використовують нечітку логіку. У трьох десятках СУБД реалізована функція нечіткого пошуку. Власні програми на основі нечіткої логіки анонсували такі гіганти як IBM, Oracle і інші.

Нечітка логіка дає нам чудовий інструмент для вирішення завдань з даними, що динамічно змінюються.

Коротко перелічимо відмітні переваги fuzzy-систем у порівнянні з іншими:

– над змінними, заданими у нечіткому вигляді, можна проводити обчислення і отримувати відповідь із заданим ступенем точності;

– можливість нечіткої формалізації критеріїв оцінки і порівняння: оперування критеріями "більшість", "можливе", переважно" і т.д.;

– при використанні нечіткого опису процесу надається можливість не тільки кількісного, але й якісного аналізу як вхідних даних, так і виведених результатів;

– можливість проведення швидкого моделювання складних динамічних систем і їхній порівняльний аналіз із заданим ступенем точності;

– в порівнянні з класичними інструментами даний метод сильно скорочує

кількість проміжних обчислень, що суттєво за умови жорстких часових рамок в ухваленні рішення.

Розробка моделі на основі теорії нечітких множин може бути виконана в різних системах програмування, наприклад, DelpHi, C-Builder та ін. Але системи об'єктно-орієнтованої мови мають тільки можливості побудови моделей нечіткого висновку, що вимагає значних витрат при розробці. В даний час до стандартних систем створено низку програм для практичного використання нечітких множин, одна з них – це MaTLaB, яка забезпечена найбільш розвиненими пакетами Simulink і Fuzzy Logic ToolBox.

MaTLaB – це назва продукту для числового аналізу та також мова програмування. Створена компанією THe MaTHWorKs, це досить простий засіб для роботи з математичними матрицями, малювання функцій, роботи з алгоритмами, створення робочих оболонок (user inTerfaces) з програмами в інших мовах програмування. Хоча цей продукт спеціалізується на чисельному обчисленні, спеціальні інструментальні засоби працюють з програмним забезпеченням Maple, що робить його повноцінною системою для роботи з алгеброю.

MaTLaB отримав назву від «MATrix LABoraTory» яка була заснована у пізніх 1970-х Клівом Молером, який пізніше став керівником департаменту обчислювальних наук університету Нового Мексико. Він розробив його, щоби надати своїм студентам доступ до пакетів LINPACK та EISPACK без необхідності опановувати Фортран. MaTLaB став дуже скоро популярним в інших університетах і привернув особливу увагу прикладних математиків. Інженер Джон Літтл закохався у цей продукт, коли відвідав Молера у Станфордському університеті у 1983-му році. Прогнозуючи комерційний успіх MaTLaB, він приєднався до Молера і Стіва Бангерта. Вони переписали MaTLaB на Сі і заснували компанію THe MaTHWorKs у 1984-му році. Переписані бібліотеки стали відомими як JACKPAC [17]. Поза визнанням викладачів лінійної алгебри та числового аналізу MaTLaB визнали де факто спеціалісти по роботі з цифровими зображеннями (напр. томографія).

MaTLaB має більше ніж мільйон користувачів на виробництвах і науковців.

Бібліотека MaTLaB складається з розділів, кожен із яких є набором інструментів, призначених для вирішення певного кола проблем. Бібліотека містить не тільки стандартні для математичних пакетів засоби, але й надає можливість цифрової обробки зображень, пошуку рішень на основі нечіткої логіки, містить апарат побудови та аналізу нейронних мереж, засоби фінансового аналізу та інші – загалом біля 30 наборів інструментів, зокрема:

– матриці та лінійна алгебра – алгебра матриць, лінійні рівняння, власні значення і вектори, сингулярності, факторизація матриць та інше;

– многочлени та інтерполяція – корені многочленів, операції над многочленами та їх диференціювання, інтерполяція та екстраполяція кривих;

– математична статистика та аналіз даних – статистичні функції, статистична регресія, цифрова фільтрація, швидке перетворення Фур'є та інші;

– обробка даних – набір спеціальних функцій, включаючи побудову графіків, оптимізацію, пошук нулів, чисельне інтегрування та інше;

– диференціальні рівняння – вирішення диференціальних і диференціально-алгебраїчних рівнянь, диференціальних рівнянь із запізнюванням, рівнянь з обмеженнями, рівнянь в часткових похідних та інше;

– розріджені матриці – спеціальний клас даних пакету MaTLaB, що використовується у спеціалізованих додатках;

– цілочисельна арифметика – виконання операцій цілочисельної арифметики в середовищі MaTLaB.

Операції з нечіткою логікою у пакеті MaTLaB дозволяє виконувати модуль Fuzzy Logic ToolBox. Він дозволяє створювати системи нечіткого логічного виведення і нечіткої класифікації в рамках середовища MaTLaB, з можливістю їхнього інтегрування в Simulink.

Fuzzy Logic ToolBox містить наступні категорії програмних інструментів: функції; інтерактивні модулі з графічним користувальницьким інтерфейсом (з GUI); блоки для пакета Simulink; демонстраційні приклади.

Базовим поняттям Fuzzy Logic ToolBox є FIS-структура – система нечіткого виведення (Fuzzy Inference SysTeM). FIS-структура містить усі необхідні дані для

реалізації функціонального відображення “входи-виходи” на основі нечіткого логічного виведення відповідно до схеми, приведеної на рисунку 3.1.

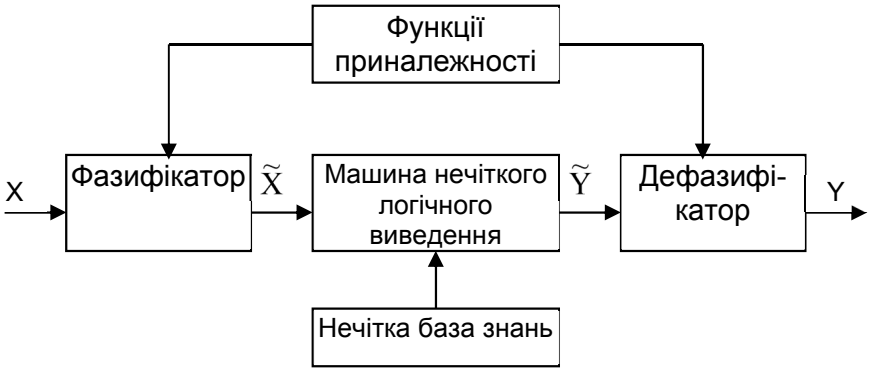


Рисунок 3.1 – Нечітке логічне виведення

Позначення: X – вхідний чіткий вектор; $\tilde{X}$ – вектор нечітких множин, що відповідає вхідному вектору X; $\tilde{Y}$ – результат логічного виведення у виді вектора нечітких множин; Y – вихідний чіткий вектор.

MaTlaB в останні роки став робочим інструментом інженерів, студентів, керівників, фізиків, зв'язківців. Серед великого спектра засобів, що надає MaTlaB користувачеві для рішення різноманітних задач у різних областях людської діяльності особливе місце займає підсистема Simulink.

Simulink – це інтерактивне середовище для моделювання й аналізу широкого класу динамічних систем за допомогою блок-діаграм.

Основні властивості підсистеми Simulink:

– містить у собі велику бібліотеку блоків (безупинні елементи, дискретні елементи, математичні функції, нелінійні елементи, джерела сигналів, засоби відображення, додаткові блоки), які можна використовувати для графічного збирання систем;

– надає можливість моделювання лінійних, нелінійних, безупинних, дискретних і гібридних систем;

– блок-діаграми можуть бути об'єднані в складені блоки, що дозволяє використовувати ієрархічне представлення структури моделі, тим самим забезпечуючи спрощений погляд на компоненти і підсистеми;

– містить засоби для створення користувальницьких блоків і бібліотек блоків;

– підтримує підсистеми, що працюють за умовами, тригерів.

Simulink забезпечує інтерактивне середовище для моделювання, при цьому поводження моделі і результати її функціонування відображаються в процесі роботи, і існує можливість змінювати параметри моделі навіть у той момент, коли вона виконується. Simulink дозволяє створювати власні блоки і бібліотеки блоків з доступом із програм на MaTlaB, ForTran чи C, зв'язувати блоки з розробленими раніше програмами, що містять вже перевірені моделі.

Так, наприклад, у Simulink інтегровано спеціалізовані додатки, що значно збільшили потужність даного середовища моделювання:

– STaTeflow – графічний інструментарій для проектування складних систем керування. STaTeflow дає можливість моделювати поводження складних подійно-керувальних систем, базуючись на теорії кінцевих автоматів. Це дозволяє користувачам Simulink додавати подійно-керувальне поводження до їхніх моделей;

– STaTeflow Coder – генерація коду на мові C для діаграм STaTeflow. Використовуючи STaTeflow і STaTeflow Coder, користувач може генерувати код винятково для STaTeflow-частин моделі Simulink;

– Real-TiMe WorKsHop – доповнює Simulink і STaTeflow Coder, забезпечуючи автоматичну генерацію коду на мові C з моделей Simulink. За допомогою Real-TiMe WorKsHop можна легко генерувати код для дискретних, безупинних і гібридних систем, включаючи системи, що містять підсистеми працюючі при виконанні визначених умов;

– DSP BlocKseT – бібліотеки блоків Simulink для створення, моделювання і макетування цифрових систем обробки сигналів;

– Nonlinear ConTrol Design BlocKseT – інтерактивний підхід до автоматизованого проектування систем керування;

– Fixed-PoinT BlocKseT – бібліотеки блоків Simulink для моделювання поводження систем керування і динамічних фільтрів з фіксованою крапкою;

– Simulink ReporT GeneraTor – дозволяє створювати і будувати звіти з

моделей Simulink і STaTeflow у різних форматах, серед яких HTML, RTF, XML і SGML.

Взаємодія Fuzzy Logic MaTlaB з Simulink відбувається через бібліотеку fuzBlocK, що містить такі блоки:

– Fuzzy Logic ConTroller – нечіткий контролер;

– Fuzzy Logic ConTroller wiTH Ruleviewer – нечіткий контролер з виведенням вікна правил під час моделювання в пакеті Simulink;

– MeMBersHip FuncTions – бібліотека Simulink-блоків для функцій приналежностей та реалізацій логічних операцій.

Автоматичний синтез моделі відбувається за допомогою FIS Wizard. Синтезовані моделі складаються лише зі вбудованих Simulink-модулів, тому нечітке виведення виконується дуже швидко, навіть якщо модель виходить громіздкою [47].

Розширення сфери застосування автоматизованих систем управління, підвищення рівня відповідальності цих систем при вирішенні різних завдань управління в промисловій та непромисловій сфері поставило перед розробниками ряд складних і практично суперечливих проблем. З одного боку, необхідно посилювати можливості цих систем, реалізовувати все більш складні алгоритми керування, з іншого – все більш жорсткі вимоги пред'являються до рівня їх надійності. Відомо, що підвищення гнучкості систем управління за рахунок ускладнення апаратної реалізації призводить до погіршення відмовостійкості та живучості систем. Істотне зростання складності об'єктів управління створило певні труднощі і теоретичного характеру. Розробка систем управління неможлива без побудови моделі об'єкта управління. Основний зміст даної задачі полягає у вирішенні наступних проблем:

– описі процесів, що відбуваються в системах управління;

– виборі відповідних методів формалізації та встановленні адекватності одержуваних моделей з вихідним об'єктом, а також з методами дослідження (в залежності від рівня фізичної і математичної строгості).

Слід зазначити, що традиційний підхід до вирішення завдань теорії

управління на основі існуючих в прикладній математиці формально-логічних методів ставить своєю метою створення точних (в широкому сенсі) моделей, строгих міркувань і висновків. Основну увагу при цьому доводиться приділяти питанням коректності, повноти, несуперечності, замкнутості, стійкості, керованості і багатьом іншим якісним аспектам опису моделей об'єктів і алгоритмів управління.

Слід зазначити, що процес побудови моделей фізичних процесів носить складний еволюційний характер, пов'язаний з неминучою апроксимацією реального об'єкта і призводить до втрати інформації при його описі. При цьому гіпотези і аксіоми, з яких здійснюються апроксимація та представлення реального об'єкта відповідною моделлю, можуть не враховувати всій реальної сутності фізичного процесу, що призводить до додаткового приросту ризику і невизначеності в описі об'єкта управління. Тут доречно нагадати принцип несумісності Л.Заде: "У міру зростання складності системи наша здатність формулювати точні, що містять сенс, твердження про її поведінку, зменшуються аж до деякого порогу, за яким точність і зміст стають взаємовиключними".

Одним з можливих напрямів вирішення перерахованих вище проблем є створення "інтелектуальних" систем управління, в яких управління і прийняття рішень реалізуються на моделях нечітких регуляторів і експертних систем з використанням нечітких індукцій і алгоритмів управління, узагальнених правил нечіткого логічного висновку. З точки зору нечітких моделей розглядаються два варіанти:

1) нечіткість опису як апроксимація слабоструктурованої моделі реального об'єкта управління через його складність і нечіткості інформації про його властивості;

2) реальний об'єкт володіє об'єктивною внутрішньої нечіткістю опису функціонування[48].

У першому випадку оцінка адекватності моделі реальному об'єкту встановлюється нечіткої мірою відносини між досліджуваними об'єктами або системами та методами імітаційного моделювання. Особливе значення такий підхід має при побудові промислових інтелектуальних систем автоматичного

керування, заснованих на знаннях і використовують у структурі контурів управління відповідні пристрої з штучним інтелектом. У другому випадку дослідження повноти відповідної оцінки адекватності об'єкта управління показали, що існує клас динамічних систем, для яких істинність суджень про адекватність систем, що належать цьому класу, своїм моделям, не може принципово приймати булеві значення {0,1}. Сформульовані п'ять принципів організації інтелектуальної керуючої структури.

Перший принцип. Наявність взаємодії керуючих систем з реальним зовнішнім світом з використанням інформаційних каналів зв'язку.

Другий принцип. Принципова відкритість систем з метою підвищення інтелектуальності і вдосконалення власної поведінки.

Третій принцип. Наявність механізмів прогнозу змін зовнішнього світу і власного поведінки системи в динамічно мінливому зовнішньому світі.

Четвертий принцип. Наявність у керуючої системи багаторівневої ієрархічної структури, побудованої у відповідності з правилом підвищення інтелектуальності і зниження вимог до точності моделей у міру підвищення рангу ієрархії в системі (і навпаки).

П'ятий принцип. Збереження функціонування (можливо, з деякою втратою якості або ефективності, інакше з деякою деградацією) при розриві зв'язків або втрати керуючих впливів від вищих рівнів ієрархії керуючої структури [20].

У відповідності з цими принципами визначені два типи інтелектуальних керуючих систем.

Управляючі системи, які організовані і функціонують відповідно до сформульованих п'яти принципів (в повному їх обсязі), називаються керуючими системами, що володіють властивістю "інтелектуальність у великому".

Керуючі системи, структурно не організовані у відповідності з наведеними вище п'ятьма принципами, але використовують при функціонуванні знання (наприклад у вигляді правил) як засіб подолання невизначеності вхідної інформації, моделі керованого об'єкта або його поведінки, називаються керуючими системами, що володіють властивістю "інтелектуальність в малому". Прикладом керуючих систем з властивістю "інтелектуальність в малому" служать нечіткі

контролери.

Нечітким регулятором (контролером) називається ієрархічна дворівнева система управління, "інтелектуальна в малому", на нижньому (виконавчому) рівні якої знаходиться традиційний контролер, а на верхньому координаційному рівні використовується база знань (з блоком нечіткого виведення у вигляді продукційних правил з нечіткою імплікацією) і пристрої перекладу в лінгвістичні і в чіткі значення (фазифікатор і дефазифікатор відповідно). У наведена функціональна схема нечіткого контролера. Як правило, у всіх нечітких контролерах використовується основний принцип регулювання – принцип регулювання по відхиленню.

Вихідна змінна об'єкта управління порівнюється із заданим значенням *xr*, помилка неузгодженості *E=xr-x* зазвичай піддається різним масштабним перетворенням. Крім самого значення неузгодженості обчислюється швидкість зміни неузгодження. Отримані числові значення перетворюються фазифікатором у відповідні лінгвістичні значення. Використовуючи ці значення і знання, що зберігаються в базі знань, процесор виведення визначає лінгвістичний еквівалент керуючого впливу, який за допомогою дефазифікатора перетвориться в числову форму. У всіх цих операціях бере участь база знань контролера, яка, можна вважати, складається з двох частин: бази даних і бази правил. База даних містить лінгвістичні значення всіх використовуваних змінних і відповідні базові множини. При роботі фазифікатора визначається приналежність помилки неузгодженості і швидкість зміни неузгодження конкретним базовим множинам з відповідними лінгвістичними значеннями. Дефазифікатор вирішує зворотну задачу, при якій для лінгвістичного значення управління визначається базова множина і знаходиться точка єдиного керуючого впливу [49].

База правил нечітких регуляторів будується на основі продукційної моделі знань, що має конструкцію виду "якщо ..., то ...". Кожна продукція представляється у вигляді безлічі пар "ситуація – дія" і дозволяє ставити у відповідність зі сформованою ситуацією дію регулятора у вигляді значення регулюючого впливу.

Основною проблемою створення нечіткого регулятора є конструювання бази знань, що містить досвід і знання людини – оператора.

Заповнення бази знань може виконуватися різними способами: оператор експерт управляє технологічним процесом, за яким "спостерігає" регулятор, запам'ятовуючи всі дії експерта і заповнюючи свою базу знань; оператор-експерт описує свою дію при кожній ситуації, що спостерігається у вигляді продукції "якщо ... , то ... ", які й будуть утворювати базу знань регулятора; перед самоорганізованим нечітким регулятором ставиться мета забезпечити бажану перехідну характеристику проектованої системи і одночасно повідомляється деяка інформація про технологічний процес (об'єкт управління). Регулятор самостійно (методом проб і помилок) накопичує знання без експерта. Для побудови бази знань нечіткого контролера може використовуватися і більш складна структура: "ситуація - стратегія управління – дія". Крім розглянутої структури можливі і складніші, здатні адаптувати до змін навколишнього середовища шляхом перемикань на інші безлічі лінгвістичних значень, продукційні правила, бази знань.

Сукупність – фазифікатор, процес виведення, дефазифікатор – можна розглядати як нечіткий процесор. У реалізації останнього в даний час можливі наступні напрямки:

– програмна емуляція нечітких процесорів на персональних комп'ютерах або промислових контролерах;

– застосування плат-прискорювачів для ПК, що реалізують нечіткі алгоритми управління;

– створення спеціалізованих нечітких процесорів.

Емуляція нечітких процесорів на персональних комп'ютерах не викликає в даний час будь-яких ускладнень. Апаратні і програмні можливості сучасних ПК дозволяють відтворювати всі необхідні операції [10]. Однак вимога управління в реальному масштабі часу може бути забезпечена традиційними ПК на обмеженій кількості продукційних правил. Значно кращі перспективи тут мають ЕОМ на RISC процесорах. Загальним недоліком цих рішень є апаратна і програмна надмірність подібних рішень. У той же час високий рівень розвитку ПК робить їх зручним засобом розробки систем, що використовують методи теорії нечітких множин.

Плати-прискорювачі могли б взяти на себе основні функції по виконанню

операцій фазифікації, обробки продукційних правил і дефазифікації, при цьому ПК використовувався б як інтелектуальний пристрій вводу-виводу і місцезнаходження бази даних і правил. При такому рішенні велике значення має раціональна організація обміну між ПК і платою-прискорювачем. Плати-прискорювачі мають являти собою співпроцесори, орієнтовані на виконання певного класу операцій з нечіткими множинами (числами). В першу чергу це мають бути системи, що працюють з продукційними правилами.

При розробці відповідних плат, мабуть, варто вирішити кілька проблем, пов'язаних:

1) з самою структурою продукційних правил, маючи на увазі, що правила можуть бути:

– простими: якщо <проста умова>, то <дія>;

– складними: якщо <проста умова>, то <дія1>, інакше <дія2>;

– складовими: якщо <проста умова1>, то якщо <проста умова2>, то <дія>.

2) зі структурою причинної частини продукційних правил, яка може являти собою деяку композицію довільної складності простих умов;

3) рішення може вибиратися по одному продукційному правилу, що має найбільший рівень виконання або по комбінації декількох продукційних правил;

4) з типом алгоритму обробки продукційного правила: максимінний алгоритм, імплікаційний (Заде, Лукасевич, Мамдані).

У той же час, незалежно від конкретної реалізації нечіткого процесора і змісту пунктів 1-4, необхідна реалізація основних етапів:

– фазифікація – відображення множини значень вхідних змінних на множину їх лінгвістичних значень;

– процедура виведення, коли за встановленими лінгвістичним значенням з бази правил вибирається продукційне правило або правила з найбільшим рівнем істинності і потім обчислюється результуюча згортка функцій належностей, що входять в правило (правила);

– дефазифікація, коли відбувається визначення єдиного керуючого впливу.

Слід мати на увазі, що ці етапи послідовно виконуються в кожному циклі

управління та ефективність їх реалізації буде безпосередньо позначатися на процесі управління.

3.2 Структурна схема нечіткого контролера, побудованого засобами Matlab Simulink

Побудову моделі контролера вибору криптоалгоритму у комп'ютерній мережі можна здійснити засобами Simulink. Дана модель нечіткого висновку за класичним механізмом Мамдані, описаним у 3.2 [40].

Входами нечіткого контролера (Fuzzy Logic ConTroller), який працює за механізмом Мамдані є значення продуктивності (perforMance) та рівня доступу (access), а виходом – значення криптоалгоритму (algoriTHM), описаного в таблиці 3.1, відповідно до задання нечіткого висновку на рисунку 2.7.

Таблиця 3.1 – Нечіткі множини виходу

Алгоритм захисту інформації	Нечітка множина
Відсутній	[0,1]
DES	[1,2]
RSA	[2,3]
EC	[3,4]

Найбільш важливим застосуванням теорії нечітких множин є контролери нечіткої логіки. Їх функціонування дещо відрізняється від роботи звичайних контролерів; для опису системи замість диференційних рівнянь використовуються знання експертів. Ці знання можуть бути виражені за допомогою лінгвістичних змінних, які описані нечіткими множинами.

Загальна схема нечіткого контролера містить три блоки опису функцій належності вхідних змінних (блоки InpuT MF), блок опису функцій належності

виходу (OuTpuT MF), виходи яких поступають на вхід 15 правил (блоки Rule 1 … 15).

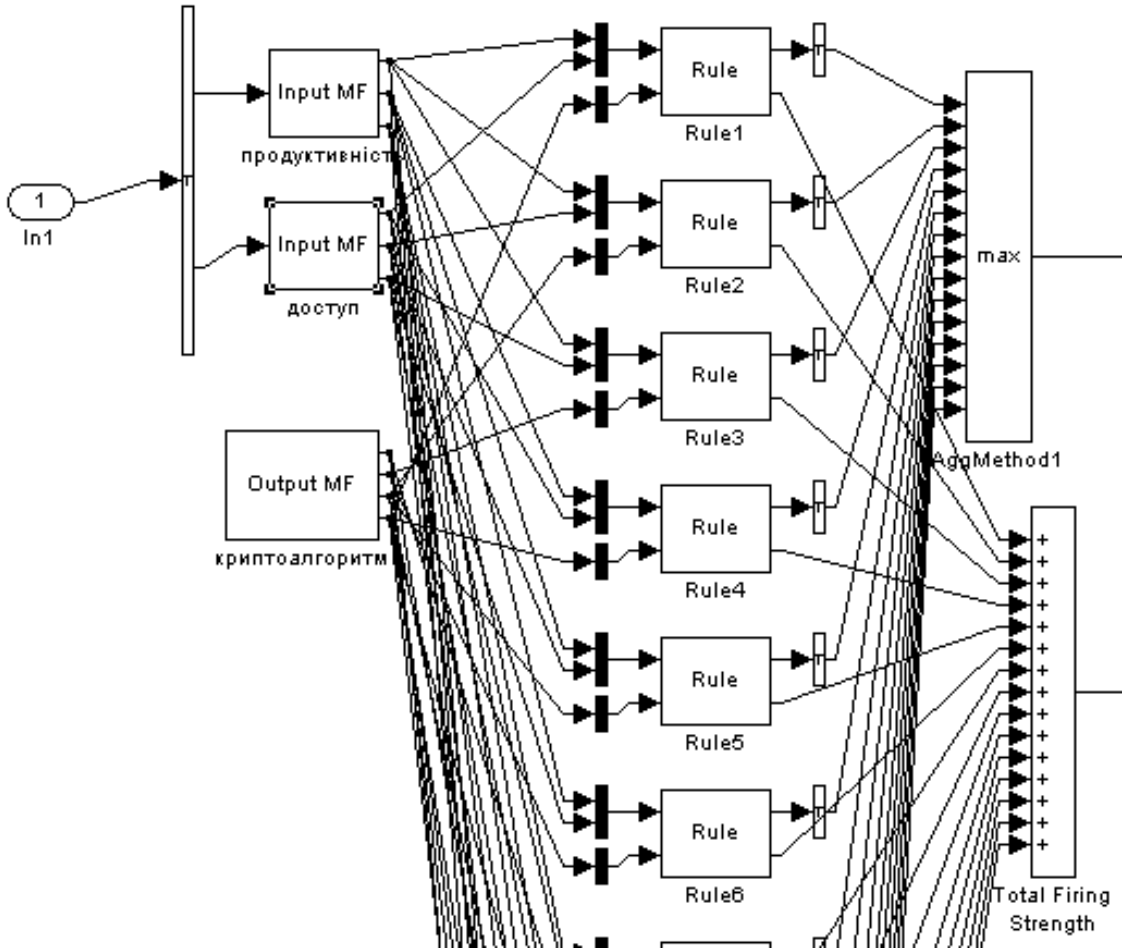
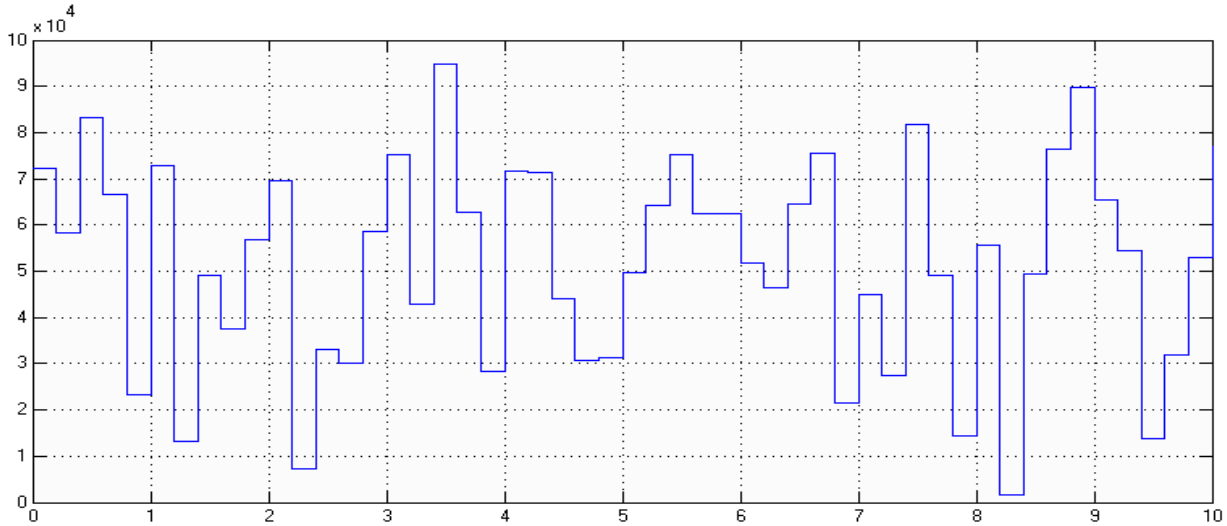
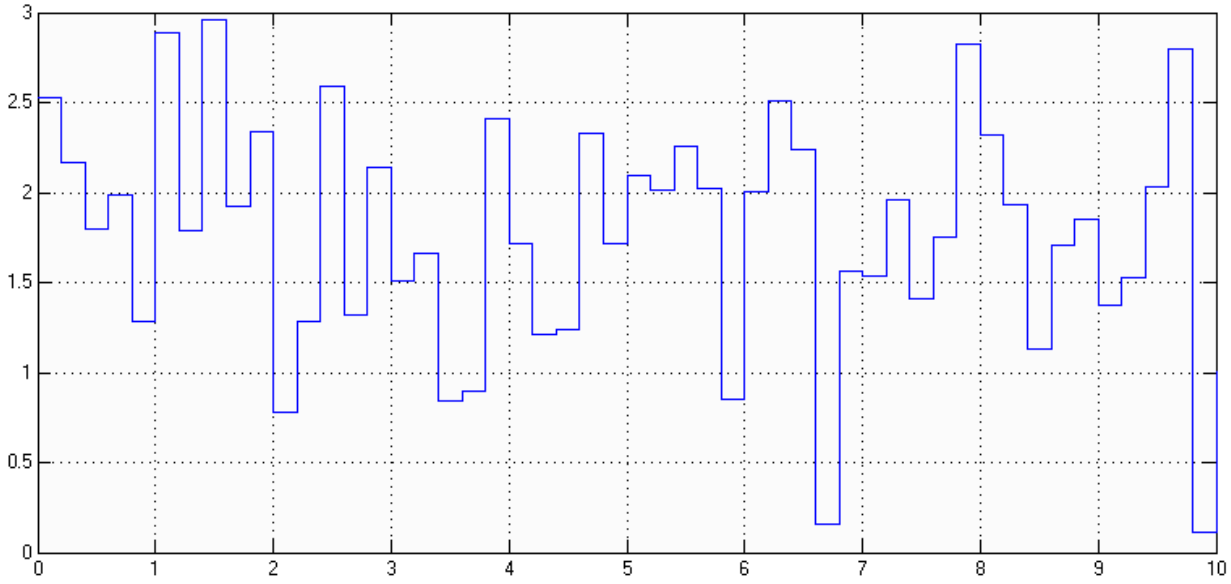


Рисунок 3.2 – Фрагмент схеми розробленого нечіткого контролера

Вхідні змінні задаються випадковим чином з рівномірним розподілом, що зображено на рисунку 3.3.



а)



б)

Рисунок 3.3 – Рівномірно розподілене задання випадкових значень вхідних змінних: а) продуктивності; б) рівня доступу

Схема обчислення функцій належності вхідних та вихідної змінних, побудована системою Simulink, подано на рисунках 3.4, 3.5, 3.6.

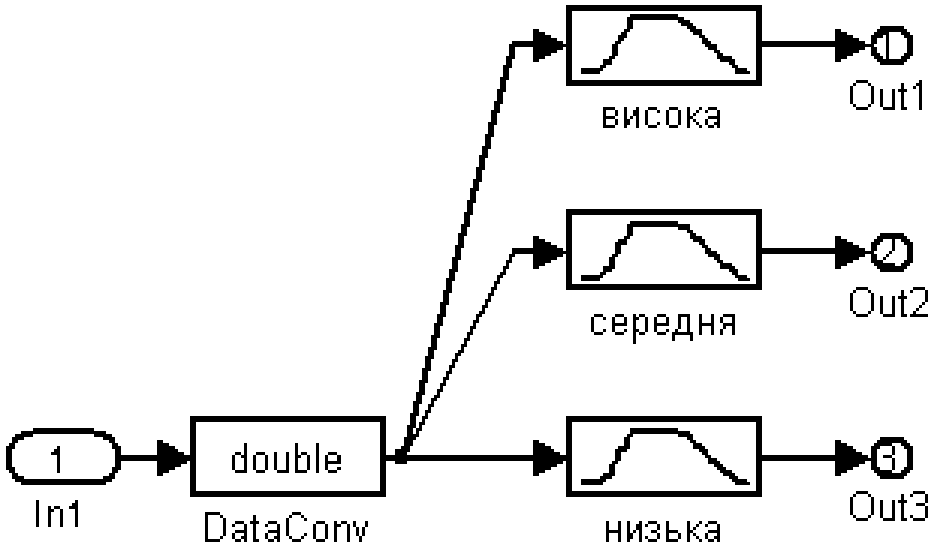


Рисунок 3.4 – Схема визначення функцій належності вхідної змінної «продуктивність» нечіткого контролера

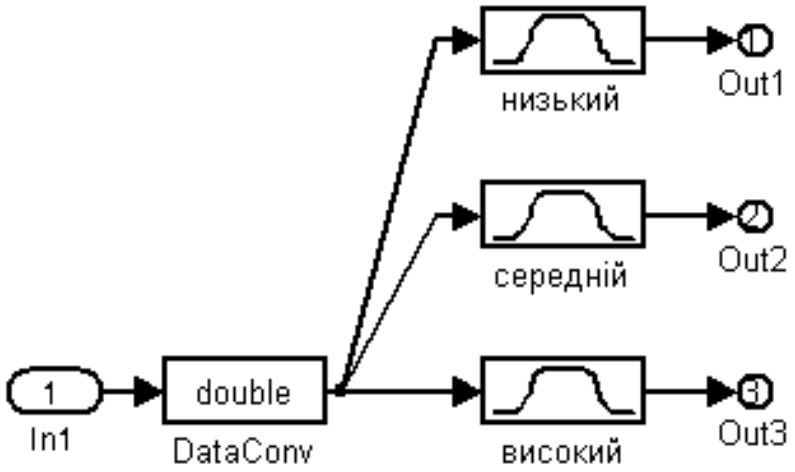


Рисунок 3.5 – Схема визначення функцій належності
вхідної змінної «доступ» нечіткого контролера

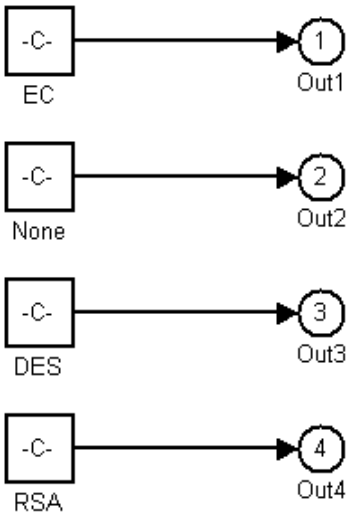


Рисунок 3.6 – Схема визначення функцій належності вихідної змінної
«криптоалгоритм» нечіткого контролера

Опрацювання нечітких змінних та вибір криптоалгоритму, що є нечітким висновком за механізмом Мамдані, здійснюється за схемою, яка опрацьовує правила з бази знань. Входами правила є значення вхідних змінних рівня доступу та продуктивності (вхід 1) та відповідне їм значення криптоалгоритму (вхід 2). Опрацювання цих даних відбувається за мінімальним законом (блок Min). Виходами даної схеми є значення функції належності виходу algoriTHM (вихід 1) та послідовність, що відображає інтервал задання цього виходу (вихід 2).

3.3 Дослідження режимів роботи нечіткого контролера

Для здійснення висновку за механізмом Мамдані нечіткий контролер здійснює дефазифікацію, тобто знаходження центру ваги кінцевої фігури, що утворюється в результаті сумування виходів 15 правил. Схема дефазифікації, подана на рисунку 3.7, реалізує формулу [16]:

$$r_{цв} = \frac{\sum_{j=1}^{m} r_j \mu(r_j)}{\sum_{j=1}^{m} \mu(r_j)}, \tag{3.1}$$

де m – кількість прямокутників, на які поділено кінцеву фігуру,

r_j – значення абсциси,

$\mu(r_j)$ – значення ординати j-ї фігури.

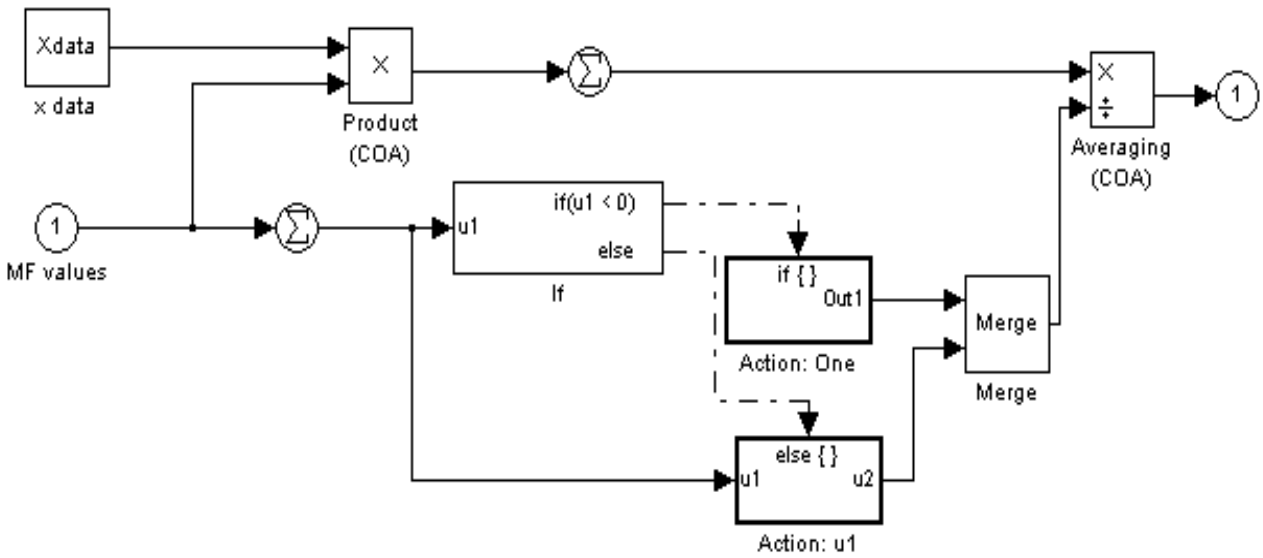


Рисунок 3.7 – Схема дефазифікації нечіткого висновку

Для перевірки правильності виведеного результату застосовується порівняння з нульовим значенням виходу (рисунок 3.8).

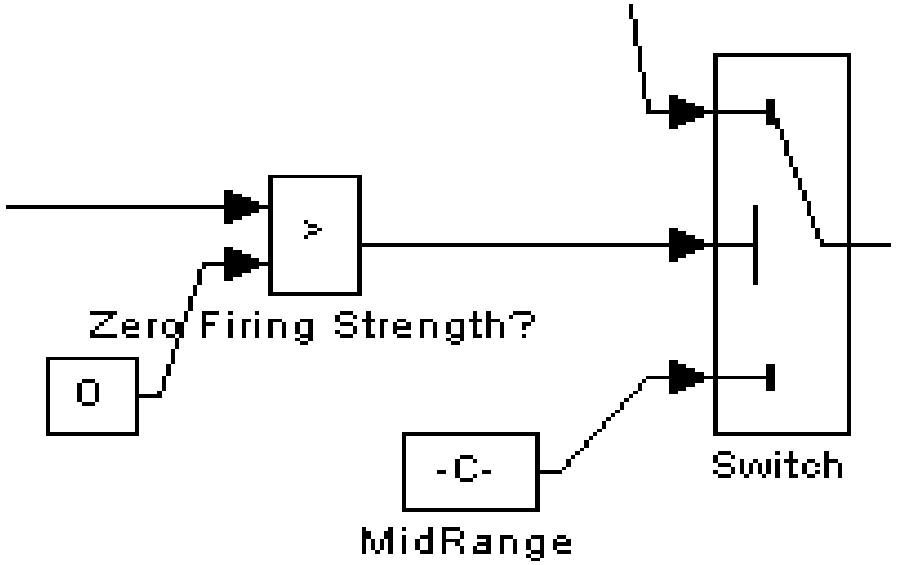


Рисунок 3.8 – Перевірка правильності виведення результату роботи нечіткого контролера

Результат роботи моделі при заданні вхідних значень стійкості до часової атаки, продуктивності та допустимих затрат пам’яті системи з однаковим розподілом, тобто значення центра ваги, зображено на рисунку 3.9.

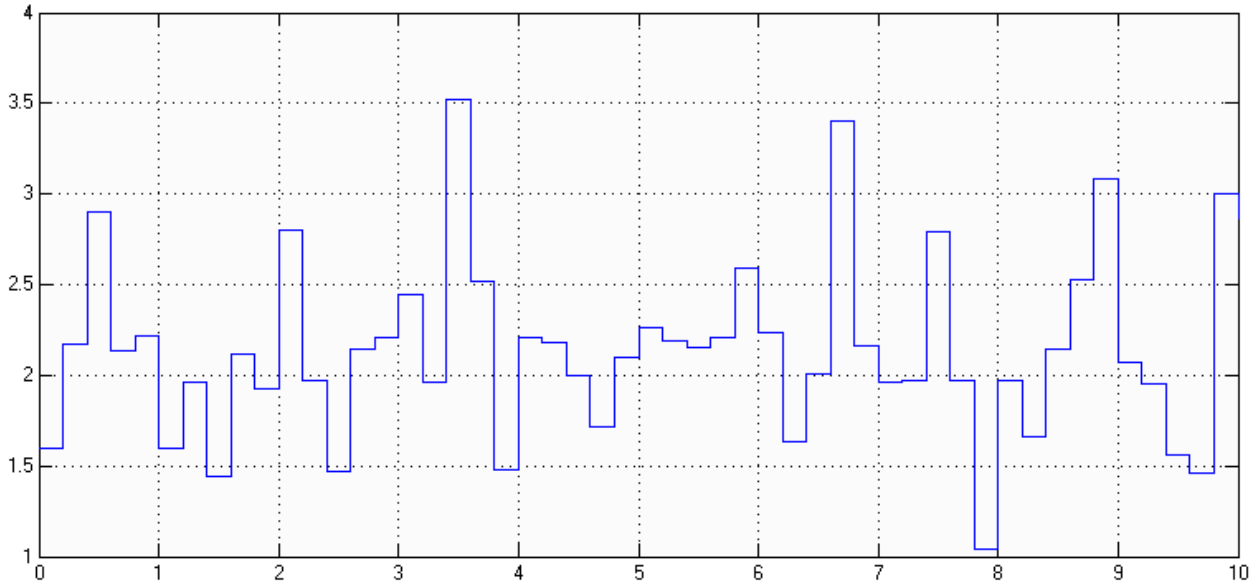


Рисунок 3.9 – Результати роботи розробленої нечіткої моделі

У таблиці 3.2 подано тестові значення вхідних та вихідних значень нечіткої системи вибору оптимального методу криптоалгоритму за механізмом Мамдані.

Таблиця 3.2 – Тестові значення змінних нечіткої системи вибору методу криптоалгоритму, побудованої за механізмом Мамдані

№ п\п	PerforMance	Access	AlgoriTHM
1	1.14e+004	2.68	0.99
2	3.13e+004	2.11	2.14
3	9.74e+004	0.673	3.56
4	5.18e+004	1.69	1.95
5	3.71e+004	0.982	2.38
6	7.54e+004	0.32	3.39
7	8.45e+003	2.85	0.924
8	2.32e+004	1.18	2.17
9	7.17e+004	0.585	3.17
10	5.99e+004	1.6	2

3.4 Захист розробленої нечіткої системи вибору криптоалгоритму

Заходи захисту інформації повинні:

– бути відповідними загрозам;

– бути розробленими з урахуванням можливої шкоди від їх реалізації і вартості захисних заходів та обмежень, що вносяться ними;

– забезпечувати задану ефективність захисту інформації на встановленому рівні протягом часу обмеження доступу до неї або можливості здійснення загроз.

Мінімально необхідний рівень захисту інформації забезпечують обмежувальними і фрагментарними заходами протидії найнебезпечнішій загрозі. Підвищення рівня захисту інформації досягається нарощуванням технічних заходів протидії безлічі загроз.

Розроблена нечітка система вибору криптоалгоритму у вигляді контролера може бути реалізована на програмованих логічних інтегральних схем (ПЛІС) чи програмованих логічних матриць (ПЛМ) [5].

Контролер може бути використаний у комплексній системі захисту

інформації в комп'ютерній мережі. Така система із нечітким контролером вибору криптоалгоритму забезпечує приватність, конфіденційність, цілісність даних, а також неможливість однозначного визначення криптоалгоритму, яким шифруються дані в системі. Такі контролери та системи виготовляються у вигляді закінчених апаратних продуктів50]. Приклад контролера в складі клієнт-серверної системи доступу до даних в комп’ютерній мережі зображено на рисунку 3.10.

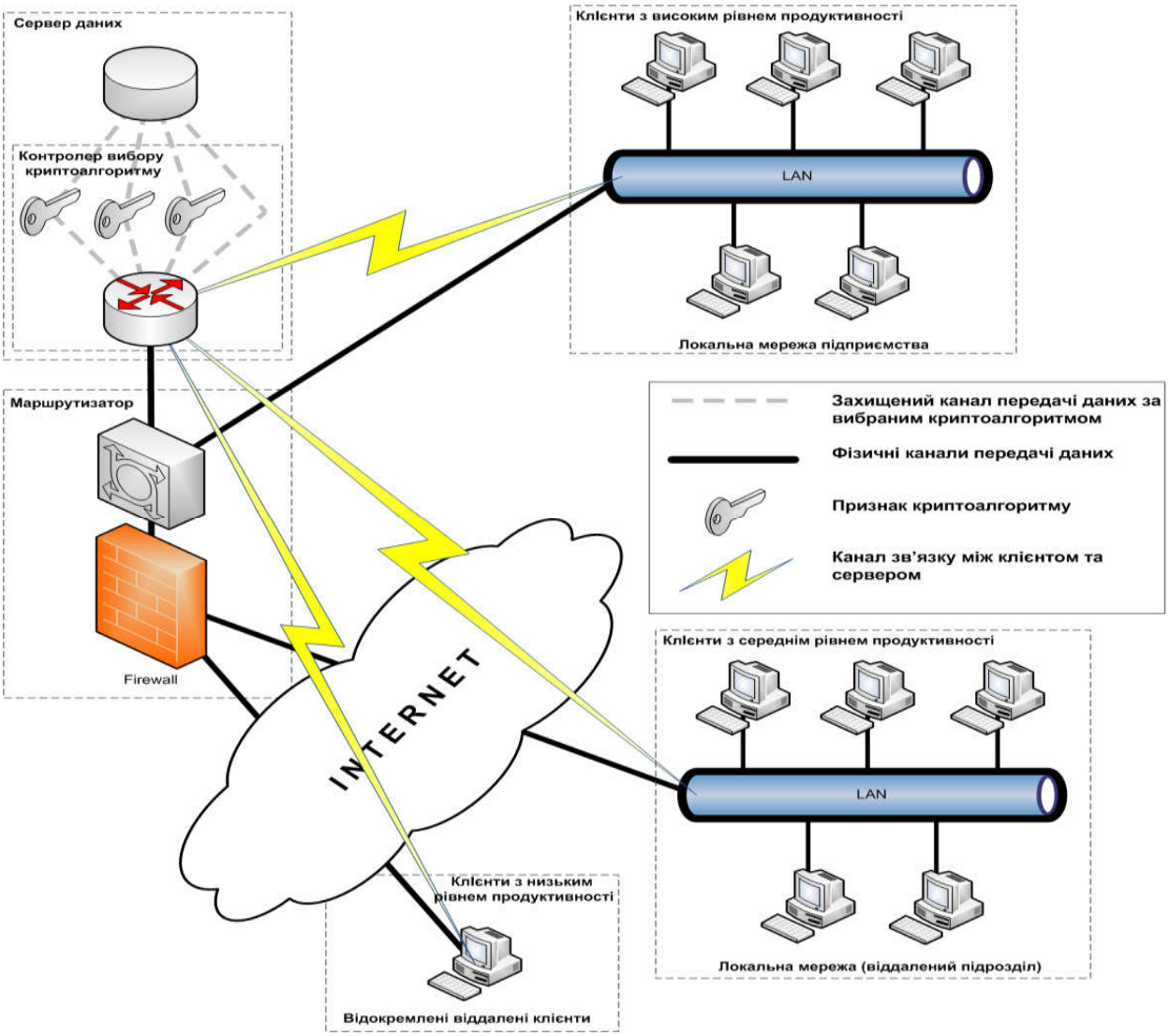


Рисунок 3.10 – Приклад використання контролера в клієнт-серверній системі доступу до даних

Хоча на сьогодні основна увага розробників систем захисту інформації приділяється програмним засобам, однак основними їх недоліками є недостатня стійкість до зламу та часто недостатня продуктивність, особливо при обробці

інтенсивних потоків даних, що може спричиняти відмову системи в цілому (наприклад при DDoS-атаці). Цих недоліків позбавлені апаратні засоби захисту інформації.

Сучасні апаратні контролери та системи захисту інформації традиційно реалізуються на базі універсальних ПЛІС, які застосовуються протягом декількох десятиліть для побудови різноманітних інтерфейсних вузлів, пристроїв керування, контролю і т.д. З появою швидкодіючих ПЛІС надвисокої інтеграції, що працюють на високих тактових частотах, їхня ніша на світовому ринку значно розширилася. Сучасні зразки ПЛІС, виконані по 0,22-мікронній технології, здатні працювати на частотах до 300 Мгц і реалізують до 3 млн. еквівалентних логічних вентилів. Настільки різке збільшення потужності ПЛІС дозволяє використовувати їх не тільки для реалізації простих контролерів і інтерфейсних вузлів, але і для цифрової обробки сигналів, складних інтелектуальних контролерів і нейрочипів.

Основні переваги ПЛІС:

– при створенні спеціалізованих логічних пристроїв розробник не обмежений можливостями наявної в його розпорядженні елементної бази – для більшості сучасних ПЛІС є бібліотеки, що містять усе необхідне, від найпростіших логічних елементів до мікропроцесорів;

– ПЛІС дозволяють скоротити терміни впровадження реалізованих на них пристроїв за рахунок спрощення процесу налагодження: розроблювач без сторонньої допомоги може багаторазово корегувати схему, не вносячи змін у друкований монтаж;

– застосування ПЛІС часто дозволяє істотно зменшити габарити апаратури в порівнянні з аналогічними пристроями, реалізованими на традиційних ВІС.

Апаратні засоби захисту інформації мають значно вищу стійкість до зламу в порівнянні з програмними, основними перевагами над програмними засобами є:

– захист від несанкціонованого доступу забезпечується неможливістю фізичного доступу до системи;

– неможливість будь-якого віддаленого внесення змін у алгоритм функціонування системи та неможливість зчитування цього алгоритму;

– стійкість до атак "на відмову";

– автономність – незалежність від паралельних задач, які виконуються у системі та впливають на продуктивність.

Також для забезпечення коректного функціонування апаратної системи необхідно дотримуватися правил її експлуатації – підтримувати відповідні умови зовнішнього середовища (температура, вологість повітря), забезпечувати безперебійну подачу електроживлення, наявність відповідного захисного заземлення, захист від зовнішніх впливів – електромагнітного випромінювання, фізичних пошкоджень тощо.

Тому варто зазначити, що розроблена нечітка система вибору криптоалгоритму забезпечує захист інформації, що передається по комп’ютерній мережі, за рахунок оптимальності вихідного алгоритму шифрування даних.

ВИСНОВКИ

Наукова новизна даної кваліфікаційної роботи полягає у вдосконаленні методу вибору криптоалгоритму, що базується на апараті нечіткої логіки.

В результаті проведених досліджень:

1. Проаналізовано системи передачі інформації, ідентифікацію клієнта та основні параметри захисту інформації в комп'ютерній мережі. Серед всього спектру методів захисту даних від небажаного доступу особливе місце займають криптографічні методи. Сучасні методи шифрування гарантують практично абсолютний захист даних, але завжди залишається проблема надійності їх реалізації. В даний час особливо актуальною стала оцінка вже використовуваних криптоалгоритмів.

2. Проаналізовано сучасні алгоритми захисту інформації, що дало змогу виділи найперспективніші симетричний криптоалгоритм DES та асиметричний – RSA.

3. Вдосконалено метод вибору криптоалгоритму шляхом застосування апарату нечіткої логіки, що дає можливість будувати системи реального часу.

4. Змодельовано нечітку систему вибору криптоалгоритму, що дало змогу здійснити подальшу симуляцію нечіткого контролера.

5. Побудовано та досліджено роботу нечіткого контролера вибору криптоалгоритму, який за поточними значеннями рівня доступу клієнта та необхідного рівня продуктивності комп'ютерної системи вибирає адекватний в даному випадку алгоритм шифрування інформації.

Результати проведених досліджень плануються до використання (додаток Б).

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Дубчак Л.О., Мельник Г.М. Методичні рекомендації до виконання магістерської роботи з освітнього ступеня “Магістр”. Спеціальність: 123 – Комп’ютерна інженерія. Магістерська програма – Комп’ютерна інженерія" / за ред. О.М. Березького. Тернопіль: ТНЕУ, 2018. С. 41.

2. Гураль І.В, Дубчак Л.О. Методичні вказівки до оформлення курсових проектів, звітів про проходження практики, випускних кваліфікаційних робіт для студентів спеціальності «Комп’ютерна інженерія» / за ред. О.М. Березького. Тернопіль: ТНЕУ, 2019. С. 33.

3. Славський А.М. Вибір криптоалгоритму захисту конфіденційної інформації: зб. публікацій І наук.практ. конф. «Інтелектуальні комп’ютерні системи та мережі». Тернопіль, 2019. С. 49.

4. Гулька О.О., Славський А.М. Захист конфіденційної інформації в телемедицині. зб. публікацій ІІ наук.практ. конф. «Інтелектуальні комп’ютерні системи та мережі». Тернопіль, 2019. С. 30.

5. Невмержицький О.В. Аналіз сучасних моделей, орієнтованих на знання, та методів прийняття рішень. Інформаційні технології проектування. М.: Вип. 13. 2013. С.119.

6. Knopfmacher J. On measures of fuzziness. Journal of Mathematical Analysis and Applications. 2005. Vol. 49. P.529.

7. Дубчак Л. О. Метод обробки нечітких даних на основі механізму Мамдані. Системи обробки інформації. Тернопіль: 2012. Вип. 7 (105). С.131.

8. Липаева В. В. 'Программная инженерия. Методологические основы. Инженерия программного обеспечения. М. : Издательский дом «Вильямс», 2002. 624 с.

9. Лавріщева К.М. Програмна інженерія. К.: Видавництво «Академперіодика», 2008.319 с.

10. Ларман К. Применение UML и шаблонов проектирования. М.: Издат. дом «Вильямс», 2002. 617 с.

11. Вендров А.М. CASE-технологии. Современные методы и средства проектирования информационных систем. М.: Финансы и статистика, 1998.

12. Лекае В. А. Некоторые аспекты разработки, создания и эксплуатации web-сайтов и порталов. Межотраслевая информационная служба. 2007. №2. С.44.

13. Menezes A. J., van Oorschot P. C., Vanstone S. A. Handbook of Applied Cryptography.CRC Press, 1996. 794 p.

14. Брассар Ж. Современная криптология. М. : Полимед, 1999. 176 с.

15. Шнайер Б. Прикладная криптография. Протоколы, алгоритмы, исходные тексты на языке Си. М.: Триумф, 2002. 816 с.

16. Ященко В. Введение в криптографию. М. : МЦНМО, 2012. 348 с.

17. Тилборг И.Основы криптологии. М. : Мир, 2006. 472 с.

18. Rivest R. Method for obtaining digital signatures and public-key cryptosystems. New York City: ACM, P. 356.

19. Boneh D. Twenty Years of attacks on the RSA Cryptosystem. AMS, 1999. P. 203.

20. Фергюсон Н. Практическая криптография. М. : Диалектика, 2004. 432 с.

21. Мирончук Ю., Купріненко О. Побудова функцій належності нечітких множин, які відповідають кількісним експертним оцінкам фізичних величин. Системи обробки інформації. К.: 2017. 207с.

22. Блюмин С., Шуйкова И., Сараев П. Нечеткая логика: алгебраические основы и приложения: Монография. К.: ЛЭГИ, 2002. 113 с.

23. Cordon O., Herrera F. A General study on genetic fuzzy systems. Genetic Algorithms in computer science. M.: Tante, 1995. P. 33.

24. Леоненков А. Нечеткое моделирование в MATLAB и fuzzyTECH. СПб.: БХВ-Петербург, 2005. 736 с.

25. Abadeh M., Habibi J., Lucas C. Intrusion Detection Using a Fuzzy Genetics-Based Learning Algorithm. Journal of Network and Computer Applications. 2007. P. 414.

26. Багрова І.В. Організація виробництва. Київ, ЦНЛ, 2005. 248 с.

27. Земор Ж. Курс криптографии. Ижевск : РХД, 2006. 256 с.

28. Ян С. Криптоанализ RSA. Ижевск : РХД, 2011. 312 с.

29. Коутинхо С. Введение в теорию чисел. Алгоритм RSA. М. : Постмаркет, 2001. 328 с.

30. Bakhtiari M., Maarof M. Serious Security Weakness in RSA Cryptosystem. IJCSI. 2012. P. 175.

31. Петюшкин А. В. HTML в дизайне. СПб .: БХВ-Петербург, 2004. 400с.

32. Чмора А.Л. Силовая атака на основе распределенных вычислений. Современная прикладная криптография. 2002.

33. Скляров Д. Искусство защиты и взлома информации. СПб.: БХВ-Петербург, 2004. 288 с.

34. Столлингс В. Операционные системы. Москва-Санкт-Петер-Киев: Вильямс, 2002. 845 с

35. Boneh D. Twenty Years of attacks on the RSA Cryptosystem. AMS, 1999. P. 203.

36. Зегжда Д.П., Івашко А.М. Основи безпеки інформаційних систем. М.: Гаряча лінія. Телеком, 2000. 134с.

37. Mamdani E. Application of fuzzy algorithms for the control of a simple dynamic plant. Proc. IEEE 121, 1974. P.1585.

38. Koo T. Analysis of a Class of Fuzzy Controllers, in Proc. 1st Asian Fuzzy Systems Sump. Singapore: Way, 1998. P. 35.

39. Passino K., Yurkovich S. Fuzzy Control. California: Addison-Wesley, 2001. P. 53

40. Iancu I. Extended Mamdani Fuzzy Logic Controller. California: ACTA Press, 2001. P. 143.

41. Ротштейн А., Штовба С. Идентификация нелинейной зависимости нечеткой базой знаний с нечеткой обучающей выборкой. Кибернетика и системный анализ. Х.: 2006. Вип. 2. С. 17.

42. Дьяконов В.М., Круглов В.О. Алгоритмы нечёткого вывода: алгоритм Мамдани и алгоритм Сугэно. Математические пакеты расширения MATLAB. Специальный справочник. Санкт-Петербург: Питер, 2001. 309 с.

43. Рижова В.А. Проектування і дослідження комплексних систем безпеки. СПб: НДУ ІТМО. 2012.

44. Novak V., Perfilieva I., Mockor J. Mathematical principles of fuzzy logic M: Kluwer Academic Publishers, 1999. P. 15.

45. Ворона В.А., Тихонов А. В. Система контролю і управління доступів. М.: Гаряча лінія. Телеком, 2010. 272с.

46. Classification of Network Traffic Using Fuzzy Clustering for Network Security: веб-сайт. URL: https://link.springer.com/chapter/10.1007/978-3-319-62701-4_22 (дата звернення: 05.03.2019).

47. Рутковская Д., Пилинский М., Рутковский Л. Нейронные сети, генетические алгоритмы и нечеткие системы. М.: Горячая линия. Телеком, 2004. 452 с.

48. Штовба С. Д. Введение в теорию нечетких множеств и нечеткую логику. Винница: Издательство винницкого государственного технического университета, 2001. 198 с.

49. Штовба С. Д. Проектирование нечетких систем средствами MATLAB. М: Горячая линия. Телеком, 2007. 288 с.

50. Expert evaluation model of the computer system diagnostic features: веб-сайт. URL: https://ieeexplore.ieee.org/document/7027101/metrics#metrics (дата звернення: 02.03.2019).