**МІНІСТЕРСТВО ОСВІТИ І НАУКИ КРАЇНИ**

**Західноукраїнський національний університет**

**Фаультет комп’ютерних інформаційних технологій**

Кафедра комп’ютерної інженерії

**РАТУШНЯК Юрій Васильович**

**Алгоритми аналізу політики безпеки підприємства / Algorithms for analysis enterprise security policy**

спеціальність; 123 – Комп’ютерна інженерія

освітньо-професійна програма - Комп’ютерна інженерія

Кваліфікаційна робота

Виконав студент групи КІм-22

Ю.В.Ратушняк

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Науковий керівник:

к.т.н., доцент Л.О.Дубчак

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кваліфікаційну роботу

допущено до захисту

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_2021 р.

Завідувач кафедри КІ

О.М.Березький

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Тернопіль - 2021

ВСТУП

З розвитком комп’ютерної техніки та поширенням її в повсякденному житті особиста інформація багатьох людей стала вільно доступною у мережі Internet та різних базах даних. Цей факт справедливо обурював тих, кому належала конфіденційна інформація, тому потрібно було припинити вільне поширення особистих даних.

Кожна людина вирішує особисто яку інформацію необхідно отримати і яка інформація не повинна бути доступна іншим.

Для запобігання втрати інформації розробляються різні механізми її захисту, які використовуються на всіх етапах роботи з нею. Захищати від пошкоджень і зовнішніх впливів треба і пристрої, на яких зберігається таємна та важлива інформація, і канали зв'язку.

Пошкодження можуть бути викликані поломкою обладнання або каналу зв'язку, підробкою або розголошенням таємної інформації.

Для збереження інформації використовують різні способи захисту:

* безпека будівель, де зберігається таємна інформація;
* контроль доступу до таємної інформації;
* розмежування доступу;
* дублювання каналів зв'язку та підключення резервних пристроїв;
* криптографічні перетворення інформації [1].

В даній дипломній роботі розглянуто захист файлів та особистої інформації на основі криптографічного перетворення інформації та надання доступу до даних.

Метою дипломного проектування є розробка програмного засобу захисту файлів на основі нечіткої логіки. Застосування механізмів нечіткої логіки дозволяють:

* оперувати нечіткими вхідними даними: наприклад, значеннями, що безперервно змінюються в часі (динамічні задачі), значення, які неможливо задати однозначно (результати статистичних опитувань рекламних компаній тощо);
* оперувати нечіткою формалізацією критеріїв оцінки і порівняння: оперування критеріями "більшість", "можливо", "переважно" і т.д.;
* проводити якісні оцінки як вхідних даних, так і вихідних результатів: можна оперувати не тільки значеннями даних, але і їх ступенем достовірності та її розподілом;
* проводити швидке моделювання складних динамічних систем та їх порівняльний аналіз із заданим ступенем точності: оперуючи принципами поведінки системи, описаними fuzzy-методами, можна, по-перше, не витрачати багато часу на з'ясування точних значень змінних і складання описуючих рівнянь, по-друге, можна оцінити різні варіанти вихідних значень [2].

1. СУЧАСНІ АПАРАТНО-ПРОГРАМНІ ЗАСОБИ ЗАХИСТУ ФАЙЛІВ
   1. Аналіз існуючих апаратно-програмних засобів захисту інформації

Захист даних за допомогою шифрування є одним із можливих рішень проблеми безпеки даних. Зашифровані дані стають доступними лише для тих, хто вміє їх розшифрувати, і тому крадіжка зашифрованих даних не має сенсу для неавторизованих користувачів.

Коди і шифри використовувалися задовго до появи комп'ютерів. З теоретичної точки зору чіткої різниці між кодами та шифрами немає. Однак у сучасній практиці різниця між ними зазвичай досить очевидна. Коди працюють з лінгвістичними елементами, поділяючи зашифрований текст на семантичні елементи, такі як слова та склади. У шифруванні завжди є два елементи: алгоритм і ключ [1].

Алгоритм дозволяє використовувати відносно короткий ключ для шифрування будь-якого великого тексту. Шифри в основному використовуються для захисту даних, тому ми поговоримо про це пізніше.

Шифрування даних — це процес перетворення відкритих даних у зашифровані із шифруванням, а дешифрування — процес перетворення закритих даних у відкриті дані із шифруванням [1].

Шифрування — це процес шифрування даних.

Дешифрування — це процес перетворення закритих даних у відкриті з невідомим ключем і невідомим алгоритмом.

Ключ – це специфічний секретний стан певних параметрів алгоритму перетворення криптографічних даних, що дає можливість вибрати для цього алгоритму один варіант з набору з кількох [1].

Криптографічний захист — це захист даних шляхом криптографічного перетворення, тобто перетворення даних шляхом шифрування та/або імітації.

Рівняння шифрування – відношення, яке описує процес формування зашифрованих даних із відкритих даних у результаті перетворень, заданих алгоритмом криптографічного перетворення.

Шифрування — це набір зворотних перетворень набору відкритих даних у набір зашифрованих даних, заданих за допомогою алгоритму криптографічного перетворення.

Криптозахист — це характеристика числа, яка визначає його стійкість до розшифровки. Звичайно, ця характеристика визначається часом, необхідним для декодування [3].

Для шифрування даних використовуються асиметричні та симетричні алгоритми шифрування інформації.

«… Симетричні шифри - алгоритми, що використовуються для шифрування інформації, особливість яких полягає в тому, що ключ шифрування і дешифрування однаковий, тобто з його допомогою можна як зашифрувати, так і розшифрувати (відновити) повідомлення. Ці алгоритми шифрування були єдиними, відомими до липня 1976 року» [4].

Алгоритми симетричного шифрування поділяються на потокові та масові.Алгоритми потокового шифрування обробляють текст повідомлення послідовно. «… У потокових шифрах потік зашифрованого тексту розраховується на основі внутрішнього стану алгоритму, який змінюється під час роботи. Зміна стану контролюється за допомогою ключа і, в деяких алгоритмах, потоку чистого тексту.

Блочні алгоритми працюють з блоками фіксованого розміру. Зазвичай довжина блоку становить 64 біти. У блоковому шифрі отримують фрагмент відкритого тексту і ключ, а також генерують шифртекст того ж розміру. Оскільки повідомлення зазвичай довші за один блок, потрібен метод вставки послідовних блоків. Було розроблено кілька методів, які відрізняються в різних аспектах. Це способи дії блочних шифрів, і їх необхідно ретельно вибирати при використанні блочного шифру в криптосистемі» [4].

Алгоритм симетричного шифрування вимагає одного ключа для шифрування та дешифрування повідомлень. Цей ключ називається спільним секретом, оскільки всі користувачі, які беруть участь в обміні даними, мають однаковий ключ.

Поняття стабільності шифрування пов’язане з симетричними алгоритмами шифрування. Стійкість — це показник стійкості до криптоаналітичних атак. Стабільність алгоритму визначається розміром використовуваного ключа.

Для підвищення стабільності шифрування можна використовувати кілька ключів або реалізувати алгоритм шифрування кілька разів поспіль [5].

До переваг симетричної системи можна віднести:

* відносно висока швидкість (приблизно на 3 порядки вище, ніж асиметричні системи);
* простота виконання (завдяки більш простим операціям);
* менша довжина ключа, необхідна для належної стабільності.

Але є й істотні недоліки, які практично призводять до того, що ця система в даний час майже не використовується:

* складність управління ключами у великій мережі. Це означає квадратичне збільшення кількості ключів, які необхідно генерувати, зберігати, передавати та знищувати в мережі;
* складність обміну ключами. Для використання симетричної системи необхідно вирішити проблему надійної передачі ключа кожному абоненту, оскільки для передачі кожного ключа обом сторонам необхідний секретний канал [6].

«… Одним із конкретних рішень недоліків симетричної системи шифрування стала поява асиметричної криптографії. Ця галузь криптографії дуже молода в порівнянні з іншими представниками. Перша схема, яка мала практичне значення, була запропонована лише близько 20 років тому. Але за цей час асиметрична криптографія стала однією з основних галузей криптології і використовується в сучасному світі так само часто, як і симетричні схеми» [7].

Асиметричні криптографічні алгоритми використовують один ключ для шифрування повідомлення, а інший — для його дешифрування. Ключ шифрування відомий кожному, але перетворення необоротне, тому зашифрований текст може прочитати тільки одержувач - тільки йому відомий другий (приватний) ключ [8].

Крім того, процедура шифрування вибирається таким чином, щоб вона була незворотною навіть з відомим ключем шифрування - це друга передумова асиметричної криптографії. Тобто, знаючи ключ шифрування і зашифрований текст, відновити вихідне повідомлення неможливо – прочитати його можна лише за допомогою другого ключа – ключа дешифрування. Таким чином, ключ шифрування для надсилання електронної пошти кому-небудь взагалі неможливо приховати - знаючи, що зашифроване повідомлення все ще неможливо прочитати. Тому ключ шифрування в асиметричних системах називають «відкритим ключем», але ключ дешифрування повинен зберігатися в секреті одержувачем повідомлення – він називається «приватним ключем».

Загалом система відповідності з використанням асиметричного шифрування виглядає наступним чином. Для кожного з N абонентів, які відповідають, вибирається пара ключів: «відкритий» і «закритий», де j — номер абонента. Усі користувачі мережі знають усі відкриті ключі, а кожен закритий ключ зберігає лише той абонент, якому він належить. Якщо абонент (наприклад, абонент номер 7) хоче надіслати інформацію абоненту номер 9, він використовує ключ шифрування для шифрування даних і надсилає його абоненту 9. Незважаючи на те, що всі користувачі мережі знають ключ і можуть отримати доступ до каналу. Після шифрування повідомлення вони не можуть прочитати оригінальний текст, оскільки процес шифрування є незворотним для відкритого ключа. І тільки абонент 9, який отримав повідомлення, використовує ключ, який тільки він знає, щоб здійснити його перетворення та відновити текст повідомлення. Якщо повідомлення надсилається в протилежному напрямку (від абонента 9 до абонента 7), вам потрібно буде використовувати іншу пару ключів.

Тому, перш за все, в асиметричних системах кількість існуючих ключів пов’язана з кількістю абонентів лінійно, а не квадратично, як у симетричних системах. По-друге, коли порушується конфіденційність k-ї робочої станції, зловмисник дізнається лише ключ розшифровки: це дозволяє йому прочитати всі листи, отримані абонентом k, але не дозволяє йому цього робити. pass для нього при відправці листів .

Загальна схема асиметричної криптосистеми проілюстрована на рисунку 1.1 [7].

Алгоритми відкритого ключа розроблені таким чином, що ключ, який використовується для шифрування, відрізняється від ключа дешифрування. Крім того, ключ дешифрування не може бути, принаймні, протягом розумного інтервалу часу, обчислений на ключі шифрування.

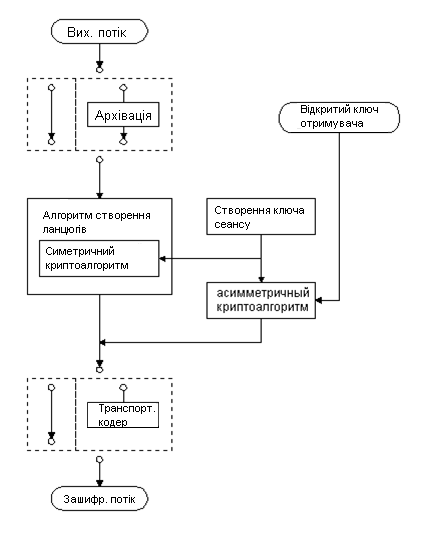


Рисунок 1.1 - Загальна схема асиметричної криптосистеми

Схематично шифрування та дешифрування за допомогою асиметричних криптографічних систем показано на рисунку 1.2 [7].

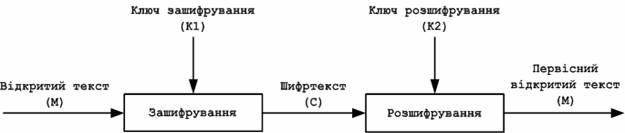
на Allbest.ru 

Рисунок 1.2 – Шифрування та дешифрування в асиметричних криптосистемах

Після винаходу криптографії було запропоновано багато криптографічних алгоритмів з відкритим ключем, що використовують відкриті ключі. Багато з них нестабільні. Багато з цих довговічних рішень не придатні для практичної реалізації. Вони або використовують дуже великі ключі, або розмір створюваного зашифрованого тексту набагато більше, ніж розмір простого тексту. Деякі алгоритми одночасно безпечні та практичні. Зазвичай ці алгоритми базуються на одній із складних завдань. Деякі з цих безпечних і зручних алгоритмів підходять лише для розповсюдження ключів. Інші застосовуються до шифрування (і розповсюдження ключів). Третій корисний лише для цифрових підписів [7].

Гібридні криптосистеми можуть прискорити роботу: для шифрування повідомлень використовуються алгоритми симетричного випадкового ключа, а для шифрування випадкових ключів сеансу використовуються алгоритми з відкритим ключем.

Оскільки асиметричні криптографічні алгоритми дуже повільні, реальні системи використовують швидкі, надійні та симетричні криптографічні алгоритми зі схемою ключа сеансу. Але сам ключ сеансу кодується асиметричним криптографічним алгоритмом з використанням відкритого ключа одержувача. Така система має всі властивості асиметричного та дуже швидкодіючого криптоалгоритму [7].

Криптографія з відкритим ключем (асиметрична) і симетрична криптографія мають свої переваги і недоліки, але вони призначені для вирішення різного роду проблем, і тому неправильно намагатися визначити, який із цих двох типів криптографічних алгоритмів є кращим чи гіршим.

Симетрична криптографія найкраща для шифрування даних. Це на порядки швидше і стійкіше до розкриття за допомогою виділеного зашифрованого тексту.

Криптографія з відкритим ключем дозволяє робити речі, недоступні для симетричної криптографії; він найкраще підходить для розподілу ключів і керування безліччю протоколів.

Розглядаючи ці два типи криптографічних алгоритмів з точки зору можливості їх використання для апаратного шифрування каналів зв’язку, можна сказати, що симетричні криптографічні алгоритми є найбільш підходящими для цього, оскільки мають суттєві переваги [7 ]:

1. позбавлений складних математичних обчислень – для шифрування використовуються прості логічні операції;
2. працювати на кілька порядків швидше;
3. більш стійкий до поломок.

Враховуючи той факт, що канали зв'язку передають дуже великі обсяги даних з високою швидкістю, а також те, що шифрування має бути апаратним, криптографічний алгоритм повинен відповідати наступним вимогам [7]:

1. орієнтація на виконання матеріалу;
2. шифрувати дані з високою швидкістю без зайвих затримок;
3. Бажано не використовувати буферизацію даних через критичні затримки.

Враховуючи вищезазначені вимоги, для апаратного шифрування більше підходять обтічні симетричні криптографічні алгоритми, які відповідають всім вимогам швидкості та стабільності шифрування, а також відсутності буферизації інформації, оскільки вони не круглі [7].

Кількість інформації в ключі зазвичай вимірюється в бітах. Для сучасних симетричних алгоритмів головною характеристикою криптовалюти є довжина ключа. Шифрування з ключами довжиною 128 біт і більше вважається надійним, оскільки для розшифровки інформації без ключа потрібні роки роботи найпотужніших суперкомп'ютерів. Для асиметричних алгоритмів, заснованих на задачах теорії чисел, через їх характеристики мінімальна надійна довжина ключа на даний момент становить 1024 біти. Для асиметричних алгоритмів, заснованих на теорії еліптичних кривих, мінімальна надійна довжина ключа становить 163 біти, але рекомендована довжина 191 біт і більше [8].

У таблиці 1.1 наведено дані про еквівалентність довжин ключів [9].

Таблиця 1.1 - Еквівалентність довжин ключів

|  |  |
| --- | --- |
| Довжина симетричного ключа | Довжина асиметричного ключа |
| 56 біт | 384 біти |
| 64 біти | 512 біт |
| 80 біт | 768 біт |
| 112 біт | 1792 біти |
| 128 біт | 2304 біти |

З таблиці 1.1 можна зробити висновок, що симетричні алгоритми мають більшу надійність у захисті інформації, з меншою довжиною ключа, ніж асиметричні алгоритми.

Щоб зашифрувати один файл і передати його комусь, або просто приховати від сторонніх людей, конфіденційність можна досягти двома способами: за допомогою апаратних шифраторів або спеціальних програм шифрування.

У сучасному світі існує ряд програмних засобів для шифрування інформації. Нижче наведено аналіз кодерів PPT (Bestty Privacy) і BestCryrt [9].

PPT (Retty Good Privacy) є одним із найвідоміших і найбільш надійних шифраторів. Ця програма містить ряд дуже потужних утиліт. Учені зазначають: «… Спочатку потрібно створити пару ключів - відкритий і приватний. Ключі можуть бути згенеровані за допомогою алгоритмів DSS і RSA. Довжина ключа може дорівнювати 1024-4096 біт. При створенні ключів необхідно ввести рядок символів, який буде використовуватися для розшифровки даних. Рекомендована кількість символів – від 100 до 200, але пам’ятайте, що програма не працює з буфером обміну, тому цю послідовність потрібно вводити вручну щоразу, коли файл розшифровується» [9].

Програма працює таким чином: відбувається обмін відкритими ключами. Маючи відкритий ключ відправника, одержувач може відкрити лист, але зі своїм приватним кодом. При створенні зашифрованого файлу необхідно вказати, хто зможе відкрити цей файл. У вікні потрібно ввести ключ відправника, а також ключі тих, хто також повинен мати доступ до зашифрованої інформації. З програмних засобів можна шифрувати (шифрування здійснюється за допомогою алгоритмів CAST, IDEA і TriPe DES), підписувати, розшифровувати файли, запускати ключі РГР і Wiе. Утиліта Wire видаляє файли без можливості відновлення [9].

Усі ключі, крім DES, мають довжину 256 біт, довжина ключа DES — 64 біти. Наразі довжини цих ключів недостатньо для забезпечення дійсно високого рівня конфіденційності. Ці ключі можуть зупинити хакера, але якщо комусь дійсно потрібно отримати доступ до інформації, він її отримає. Ви можете встановити дисковод у режим «тільки для читання», який не дозволить будь-якій програмі записувати на нього інформацію. Ви можете налаштувати програму на монтування диска під час кожного запуску та відключення через певний час або за допомогою комбінації клавіш [9].

У програмі є ще одна чудова утиліта під назвою BestCryter Service Manager. Він розроблений для того, щоб мати можливість додавати необхідні алгоритми шифрування та модулі генерації ключів і видаляти непотрібні модулі генерації ключів. Але це необхідно лише тим, хто не довіряє алгоритму шифрування DES.

Відомо, що алгоритми захисту інформації, зокрема шифрування, можуть бути реалізовані як програмними, так і апаратними методами. Апаратні кодери вважаються більш надійними і пропонують кращий захист.

Апаратний кодер за зовнішнім виглядом і по суті є звичайним комп'ютерним «залізом», найчастіше це платою розширення, яка вставляється в ISA, або PCI, роз'єм материнської плати ПК. Є й інші варіанти, наприклад, USB-накопичувач із криптографічними функціями [9].

Використання всієї карти лише для функцій шифрування є нераціональним використанням ресурсів, тому виробники апаратних кодерів зазвичай намагаються наситити їх різними додатковими функціями, зокрема [9]:

1. генерація випадкових чисел. Це потрібно в першу чергу для отримання криптографічних ключів;
2. перевірка підключення до комп’ютера. Коли ви вмикаєте ПК, пристрій просить користувача ввести особисту інформацію, наприклад, вставити дискету з ключами;
3. перевірка цілісності файлів операційної системи. Це не дозволить зловмиснику змінювати дані, поки вас немає. Шифрувальник зберігає список усіх важливих файлів із попередньо розрахованими контрольними сумами або значеннями хешування, і якщо контрольна сума не відповідає хоча б одному з них під час наступного запуску, комп’ютер буде заблоковано.

Картка з усіма цими функціями називається пристроєм захисту криптографічних даних (CRPD).

Кодер, який контролює введення на ПК і перевіряє цілісність операційної системи, також називається «електронним замком».

Одним із прикладів апаратних кодерів є кодер виробництва ЗАТ НІП «Інформзащита» СКЗІ (Система криптографічного захисту інформації) М-506 [9].

М-506 — комплексне програмно-апаратне забезпечення захисту криптографічної інформації, що реалізує алгоритм шифрування даних за ГОСТ 28147-89. У цьому комплексному засобі захисту інформації можливості апаратного кодувальника доповнюються широким набором інших функцій інформаційної безпеки.

СКЗІ М-506 складається з наступних компонентів:

* сервер безпеки. Встановлений на виділеному комп’ютері або контролері домену, він збирає та обробляє інформацію про стан усіх захищених робочих станцій та зберігає дані про налаштування всієї системи безпеки;
* Засіб захисту інформації від несанкціонованого доступу (ГІС) Secret Net NT 4.0. Усі робочі місця обладнані цією ГІС, на якій також встановлений електронний замок «Соболь». Це захищає ресурси комп'ютера і дозволяє реєструвати з'єднання користувача, представляти незареєстроване ім'я користувача, вводити неправильний пароль, перевищувати кількість спроб підключення;
* підсистема управління. Цей програмний компонент встановлюється на робочому місці системного адміністратора і дозволяє йому налаштовувати GIS Secret Net, а також керувати вбудованими можливостями операційної системи, відстежувати всі події, що впливають на безпеку системи, і ' ' відповідати на них в режимі реального часу. час;
* менеджер шифрування. Він встановлюється на автономний комп’ютер і виконує такі функції: створення ключів шифрування, створення дискет з ключами, ведення бази даних ключів, згенерованих на жорсткому диску комп’ютера.

Реалізована в архітектурі клієнт-сервер SKZI M-506, ви можете зосередити функції управління безпекою корпоративної мережі в одному місці та підвищити живучість всієї системи безпеки: навіть відмова безпеки сервера не знижує рівень безпеки. Розроблені можливості захисту від несанкціонованого доступу інтегровані в криптографічні механізми: електронний цифровий підпис, «прозоре» шифрування файлів та шифрування мережевого трафіку.

Для юридичного користувача, який має право працювати із зашифрованим мережевим ресурсом, цей ресурс знаходиться у звичному незашифрованому вигляді («прозорий» режим криптографічного перетворення). Але це не означає, що конфіденційна інформація передається по мережі на відкритому повітрі: мережевий трафік шифрується, а розшифровка відбувається лише на робочій станції користувача. Усі компоненти СКЗІ М-506 працюють у закритому програмному середовищі, недоступному для вірусів [9].

* 1. Виберіть спосіб захисту файлу

Інформаційна безпека – це діяльність, спрямована на запобігання витоку, несанкціонованих та ненавмисних дій із захищеною інформацією.

У той же час інформаційна безпека організації - стан безпеки ІТ-середовища організації, що гарантує її формування, використання та розвиток.

Зараз політика безпеки є абсолютно необхідною для більшості організацій. Перш за все, це визначає ставлення організації до безпеки та необхідні дії організації для захисту своїх ресурсів і даних. На основі політики безпеки встановлюються засоби та процедури, необхідні для забезпечення безпеки, а також визначаються ролі та відповідальність працівників організації щодо забезпечення безпеки.

Політика безпеки організації включає [10]:

* базова політика безпеки;
* спеціалізовані політики безпеки;
* порядок безпеки (рисунок 1.3).

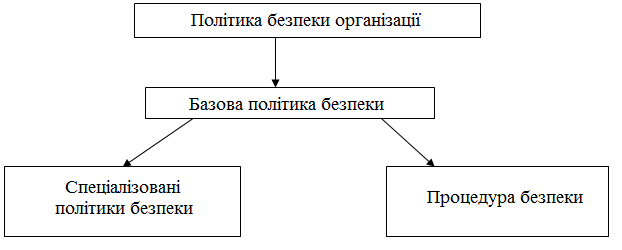


Рисунок 1.3 - Схема політики безпеки в організації

Основна політика безпеки — це те, як організація обробляє інформацію, хто може отримати до неї доступ і як. Підхід, реалізований базовою політикою безпеки, дає змогу виконувати роботу зі створення системи безпеки поступово та послідовно, не прагнучи до її негайного впровадження. Основна політика є основоположною для організації чи компанії. У свою чергу він визначає:

* вартість ділової документації;
* ризик втрати або пошкодження цих файлів;
* рівень безпеки, який визначає захист кожного файлу, тобто заходи безпеки, які можна вважати вигідними для застосування;
* політика безпеки організації на основі попередніх кроків;
* потреба у фінансових ресурсах для реалізації політики безпеки, придбання та встановлення необхідних засобів безпеки;
* потреба в обізнаності та навчанні персоналу для підтримки необхідних заходів безпеки з боку персоналу та керівництва;
* необхідність регулярних перевірок безпеки для виявлення поточних проблем, врахування змін у зовнішньому середовищі та проведення необхідних кадрових змін.

Існують десятки спеціалізованих політик, які можуть застосовувати більшість середніх і великих організацій. Деякі з них призначені для будь-якої організації, інші специфічні для певних середовищ.

З огляду на особливості застосування спеціалізованих політик безпеки можна розділити на дві групи [10]:

* політики, які впливають на значну кількість користувачів;
* політики, пов'язані з конкретними технічними галузями.

Спеціалізовані політики, які впливають на значну кількість користувачів, включають:

* Політика прийнятного використання - встановлення стандартних стандартів безпечного використання ІТ-обладнання та послуг у компанії, а також відповідних заходів безпеки для співробітників для захисту даних компанії та власної інформації;
* політика віддаленого доступу до мережевих ресурсів - встановлення стандартних правил безпечного віддаленого підключення будь-якого хоста до корпоративної мережі;
* політика захисту інформації;
* політика захисту паролем тощо.

Процедури безпеки є необхідним і важливим доповненням до політики безпеки для будь-якої організації. Оскільки політики безпеки описують лише те, що потрібно захищати та які основні правила захисту, процедури безпеки визначають, як захистити ресурси та які механізми застосування політики використовувати, іншими словами, як реалізувати політику безпеки.

Процедури безпеки:

* визначити дії, які необхідно вжити у відповідь на конкретні події;
* забезпечити швидке реагування в критичній ситуації;
* допомогти усунути проблему відмови від роботи, якщо, наприклад, під час кризи працівник раптово залишає робоче місце або його немає.

Існує багато процедур безпеки, які повинні бути стандартними в будь-якому бізнесі [10].

Говорячи про інформаційну безпеку, а в даному випадку файлів, розглянемо також поняття ідентифікації та аутентифікації [11].

Ідентифікація- присвоєння суб'єктів або об'єктів, доступ до ідентифікатора або порівняння представленого ідентифікатора зі списком призначених ідентифікаторів.

Ідентифікація предмета — це його розпізнавання, ототожнення з чимось. Якщо ми говоримо про сферу інформаційних технологій, то цей термін загалом означає встановлення особистості користувача. Цей процес необхідний для того, щоб система могла вирішити, чи надавати людині дозвіл на роботу за комп’ютером, доступ до конфіденційної інформації тощо. Таким чином, ідентифікація є одним із основних понять інформаційної безпеки.

Існує кілька способів ідентифікації користувачів. Кожен з них має свої переваги і недоліки, тому одні технології можна використовувати в одних системах, інші – в інших.

Існує три найбільш поширені типи ідентифікації [11]:

1. ідентифікація пароля –
2. Ідентифікація пристрою
3. біометрична ідентифікація.

Аутентифікаціяназивається процедурою перевірки ідентичності суб’єкта.

«… Аутентифікація здійснюється на основі того чи іншого секретного елемента, доступного суб'єкту та інформаційній системі. Звичайно, інформаційна система має не сам секретний елемент, а деяку інформацію про нього, з якої приймається рішення про відповідність суб’єкта ідентифікатору. Наприклад, перед інтерактивним сеансом більшість операційних систем запитують у користувача ім’я користувача та пароль. Введене ім’я є ідентифікатором користувача, а його пароль — аутентифікатором. Операційна система зазвичай зберігає не сам пароль, а його хеш-суму, що ускладнює отримання пароля.

В інформаційних технологіях використовуються такі методи аутентифікації» [11]:

* одностороння аутентифікація, коли клієнт системи доступу до інформації доводить її достовірність;
* Двостороння аутентифікація, коли, крім клієнта, система (наприклад, банк) також повинна підтвердити свою автентичність;
* тристороння аутентифікація, коли для підтвердження автентичності кожного з партнерів при обміні інформацією використовується так звана послуга нотаріальної аутентифікації.

Методи аутентифікації можна розділити на один фактор і два фактори.

Односторонні методи поділяються на:

• Логіка (паролі та ключові фрази, що вводяться з клавіатури комп'ютера або спеціального обладнання);

• Ідентифікація (основними носіями інформації є фізичні об'єкти: дискети, магнітні картки, чіп-карти, картки зі штрих-кодом тощо);

• Біометрія, яка базується на аналізі унікальних характеристик людини, таких як відбитки пальців, дизайн райдужної оболонки, голос і обличчя.

Двофакторні методи аутентифікації отримують в результаті поєднання двох різних однофакторних методів, найчастіше ідентифікаційного та логічного. Наприклад: «пароль + дискета», «магнітна карта + PIN-код».

Кожен клас методів має свої переваги і недоліки. Майже всі методи аутентифікації мають один недолік: вони фактично не аутентифікують конкретну сутність, а лише фіксують той факт, що аутентифікатор сутності відповідає її ідентифікатору. Тобто не всі відомі методи захищені від компромісу аутентифікатора [11].

Засоби захисту інформації слід вибирати не тільки на основі переліку їх абстрактних властивостей і переваг один перед одним, а й виходячи з потреб системи, в якій вони призначені для використання.

Якщо говорити про криптографічні засоби, то багато технічних сумнівів зводяться до того, які алгоритми реалізовані в пристрої (або програмі), програмні чи апаратні обчислення, звідки вони беруться, де зберігаються і як обробляються ключі, і, нарешті, чи сертифікати захисту інструменту та які.

Це означає, що на перший план виходить відповідність засобів захисту завданням конкретної системи.

Раніше інформація захищалася насамперед для відповідності нормативним вимогам. Багато людей вже роблять це, щоб захистити свій бізнес.

Криптографія використовується в трьох випадках [12]:

1. коли інші заходи не можуть виключити несанкціонований доступ до інформації під час зберігання;
2. коли необхідно захистити інформацію під час передачі;
3. коли необхідно захистити інформацію, що обробляється – інформаційні технології.

Усі три справи мають значення для організації захисту файлів компанії.

Використання одних і тих же засобів захисту для взаємодії технічних засобів і для взаємодії людини з технічними засобами може бути не оптимальним рішенням.

Взаємодія апаратних засобів, надійність виконуваних ними процедур над електронним файлом повинні забезпечуватися вбудованими фіксованими засобами безпеки, які контролюють середовище файлу як набір програмно-технічних засобів, що забезпечують достовірність оригіналу (перетвореного) .

Обробка файлу за безпосередньої участі людини повинна контролюватися особистими засобами, жорстко пов'язаними не тільки з тими чи іншими комп'ютерними засобами, але і з цілком конкретним користувачем системи.

Загалом, файл повинен передбачати:

1. індикація цілісності файлу;
2. доступність вмісту файлу для об'єктів бізнес-лінії;
3. апріорна недоступність вмісту файлу для об'єктів з інших сфер діяльності (конфіденційність);
4. автентифікація автора файлу;
5. автентифікація користувача або особи, яка переглядає вміст файлу;
6. множинність файлів програми.

У загальному випадку вимоги 4 і 5 забезпечуються засобами індивідуального захисту, а 2 і 6 - стаціонарними засобами.

Вимоги 1 і 3 - цілісність і конфіденційність - можуть відповідати обома або працювати разом, залежно від характеристик архітектури та функціонування системи [12].

Виходячи з вищевикладеного, після аналізу переваг та недоліків методів захисту інформації було прийнято рішення використовувати багатофакторний захист у розробленому програмному забезпеченні.

І ці фактори можна поєднувати в будь-якому порядку. Однак сьогодні в переважній більшості випадків використовується лише одна пара: захист паролем і маркером. У цьому випадку користувач може не боятися відновлення пароля від зловмисника – без ключа це не вийде, як і крадіжки токенів – без пароля не вийде. Однак деякі системи використовують більш надійні процедури ідентифікації. Вони одночасно використовують паролі, токени та біометричні характеристики людини [11].

Після аналізу цих фактів в якості вхідних даних було обрано наступне:

* логін, який не пропонується самою системою захисту файлів, але який вводиться вручну. Тому, якщо зловмиснику відомий лише пароль, він не зможе захопити інформацію, оскільки цей пароль дійсний лише з відповідним логіном;
* пароль, який буде дійсним лише для відповідного логіна користувача;
* токен - компактний пристрій, реалізований у вигляді USB-брелока, що використовується для авторизації користувачів, безпечного віддаленого доступу до інформаційних ресурсів, а також безпечного зберігання всіх персональних даних.

Захист файлу буде здійснюватися наступним чином: при спробі доступу до файлу система попросить ввести дані для перевірки, щоб здійснити ідентифікацію співробітника.

Люди, які хочуть прочитати, скопіювати, створити, змінити чи видалити файл, повинні спочатку ввести комбінацію особистого ідентифікатора та пароля та вставити маркер у порт USB. Цей токен працює за таким принципом: він має односторонню функцію f (x) і початкове число k. Сервер також знає, яка функція та який номер використовуються в маркері, пов’язаному з обліковим записом. Кожного разу, коли ви успішно входите в систему, це число k змінюється згідно з алгоритмом, відомим як маркеру, так і серверу. У цьому випадку після завершення ідентифікації токена вам буде запропоновано ввести цифровий PIN-код. Після успішного входу в систему працівникові надається доступ для виконання відповідних операцій.

Залежно від довжини з’єднання, пароля та PIN-коду токена буде три рівні доступу:

1. низький (тільки для читання);
2. середній (читання, копіювання, створення);
3. високий (читати, копіювати, створювати, змінювати, видаляти).

Отже, з вищесказаного можна зробити висновок, що вихідні дані при виконанні операцій ідентифікації, аутентифікації забезпечать певний рівень доступу.

* 1. Аналіз технічного завдання та постановка проблеми

У цьому дипломному проекті розроблено програмні засоби захисту файлів на основі нечіткої логіки. Це програмне забезпечення має забезпечувати надійний захист файлів і дозволяти доступ лише тим, хто має відповідні права. Доступ буде здійснюватися шляхом введення комбінації відповідного логіна та пароля, і якщо вам потрібен доступ з високим пріоритетом, що дозволяє виконувати операції створення, редагування, копіювання, переміщення файлу, вам, крім входу, знадобиться і пароль, вставте в роз’єм токена USB.

У сучасному світі захист інформації, а особливо конкретних файлів, дуже важливий, оскільки ці файли можуть містити конфіденційну інформацію, яка, потрапляючи в руки зловмисників, може завдати серйозної шкоди та незручностей їх власнику, а також тим, чиї дані може зберігатися у файлах даних.

Програмне забезпечення захисту файлів розроблено на основі нечіткої логіки. Хоча ця галузь широко використовується лише в ІТ, вона досить перспективна і відкриває багато можливостей для розробників, у тому числі й у сфері інформаційної безпеки.

Нечітка логіка розв’язує протиріччя між суворістю математики та невизначеністю реального світу. Апарат нечіткої, нечіткої логіки дає змогу формалізувати нечіткі поняття і знання, оперувати ними і, отже, робити нечіткі висновки.

Різниця між нечіткою логікою та класичною логікою полягає в тому, що вона працює, крім значень «правда» та «брехня», також з проміжними значеннями. Основою цього пристрою є математична теорія нечітких множин, яка оперує такими поняттями, як нечіткі множини, нечіткі змінні або лінгвістичні змінні, нечіткі відношення та інші.

Нечіткі методи характеризуються використанням лінгвістичних змінних замість числових, або на додаток до них описуються прості зв’язки за допомогою нечітких виразів, а більш складні – нечітких алгоритмів. Відповідні правила дозволяють швидко обробляти складні комбінації, що є важливою перевагою нечіткої логіки [13].

Для подальшої побудови системи захисту файлів на основі нечіткої логіки розроблено дерево рішень (рисунок 1.4).



Рисунок 1.4 – Дерево рішень

На першому етапі проводиться детальний аналіз системи, в якому здійснюється чітке виділення властивостей системи, що розвивається, та її основних властивостей.

Далі ставиться задача, в якій проводиться опис проблеми, виділення мети розробки.

Після цього здійснюється вибір методу розробки. У цій дипломній роботі в якості методу розробки було обрано нечітку логіку.

Передостанній крок – вибір системи автоматизованого проектування. В даному випадку був обраний Matlab CAD, а точніше пакет, що входить до нього - Simulink.

Останній крок – тестування розробленої системи. На цьому етапі система тестується на запропонованих тестах, в результаті чого стає зрозуміло, чи відповідають очікуванням, сформульовані під час виконання завдання, результати тестування, отримані під час роботи системи.

1. РОЗРОБКА ТА ДОСЛІДЖЕННЯ
   1. Розробка структурної схеми пристрою

Основи нечіткої логіки були закладені наприкінці 1960-х років у працях відомого американського математика Латфі Заде. Соціальний порядок досліджень такого типу був викликаний зростанням незадоволеності експертними системами. «Штучний інтелект», який легко справлявся із завданнями управління складними технічними комплексами, був безсилий у найпростіших виразах повсякденного життя. Щоб створити справді розумні системи, здатні адекватно взаємодіяти з людьми, потрібен був новий математичний апарат, який перекладав нечіткі й неоднозначні твердження про життя мовою чітких і формальних математичних формул.

Першим серйозним кроком у цьому напрямку стала теорія нечітких множин, розроблена Л. Заде. Його праця «Нечіткі множини» заклала основу для моделювання інтелектуальної діяльності людини і стала початковим поштовхом для розвитку нової математичної теорії. Він також дав назву новій галузі науки – «нечіткої логіки» [14].

Якщо в булевій алгебрі є тільки дві величини, 1 і 0 - істина або хибна, то в нечіткій логіці існують також перехідні величини, які називаються станами.

Одним з основних понять, що є частиною нечіткої логіки, є поняття нечітких множин.

Нехай E — універсальна множина, x — елемент E, а R — визначена властивість. Звичайна (чиста) підмножина A універсальної множини E, елементи якої задовольняють властивість R, визначається як множина впорядкованої пари A = {µA (x) / x}, де µA (x ) — характеристика функції, яка приймає значення 1, коли х задовольняє властивостям R, і 0 - інакше.

Операції над нечіткими множинами узагальнюють операції над звичайними множинами. Деякі з цих операцій позначаються так само, як і у випадку з класичними очисними наборами. Крім того, операції над нечіткими множинами відносять до лінгвістичної інтерпретації.

Комбінуйте нечіткі множини. Об’єднанням нечітких множин і  називається множиною , функція належності якого дорівнює . Ця операція відповідає висловлюванню. Це твердження розкриває лінгвістичний сенс цієї операції стосовно нечітких множин.Наприклад, комбінація наборів «Приблизно 32» і «Приблизно 64» буде набором «Приблизно 32 або 64». Графік функції належності цієї множини показано на рисунку 2.1 [17].



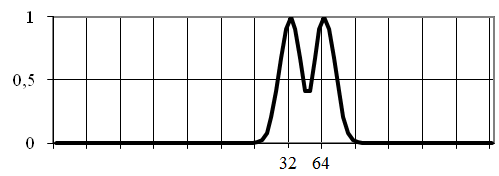


Рисунок 2.1 – Функція членства в нечітких наборах "Приблизно 32 або приблизно 64"

Перетин нечітких множин.Перетин нечітких множин  і  називається множиною , функція належності якого дорівнює . Ця операція відповідає висловлюванню «А і Б», що розкриває його мовне значення. Наприклад, вибираючи довжину пароля, ми можемо вважати, що пароль до 32 байт є середнім, а більше 64 байт – великим, тому як для «великого», так і для «середнього» перетином буде набір «не великий і не середній». Графік функцій цей набір показаний на малюнку 2.2 [17].



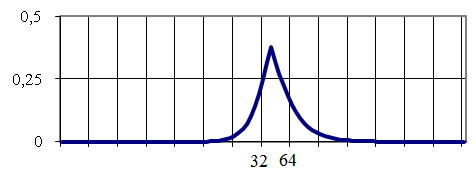


Рисунок 2.2 – Функція належності до нечіткої множини «ні великий, ні середній»

Слід зазначити, що оскільки функція належності цієї множини стає меншою за одиницю, вона є субнормальною. Для цього набору. Поділітьсяза цим числом цей набір перетворюється в номінальний. 2.3 зображено графік функції належності нормованої множини [17].

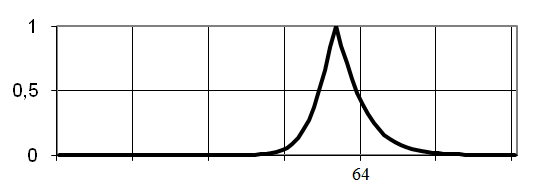


Рисунок 2.3 - Стандартний набір

Додавання.Додавання розмитий набір  у множині  називається множиною , функція належності якого дорівнює . Мовний сенс цієї операції визначається твердженням «". Наприклад, для набору «великий» доповненням є набір «ніgros ". Графік функції належності цієї множини показаний на рисунку 2.4 [17].



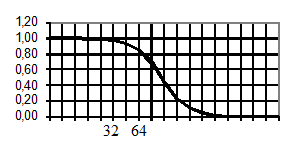


Рисунок 2.4 - Стандартний набір

Концентрація.Для нечіткого цілого  концентрація — це ціле , функція членства якого визначається зі звіту .

Лінгвістичний зміст цієї операції полягає в наступному. Наприклад, для лінгвістичної змінної «великий» концентрація відповідає виразу «дуже велика». На рисунку 2.5 наведено графіки функцій належності «великої» нечіткої множини та її «дуже великої» концентрації.

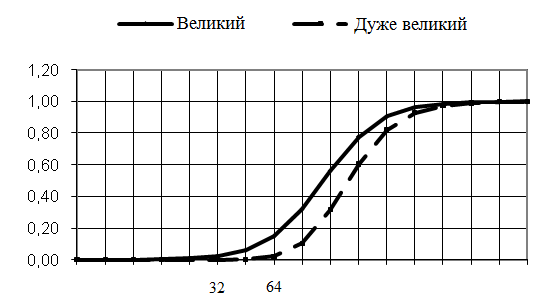


Рисунок 2.5 – Функції належності «великих» і «дуже великих» нечітких множин

Ерозія.Розмиття розмитого набору  називається множиною , функція членства якого визначається зі звіту .

Лінгвістичнийсенс цієї операції «не дуже». Наприклад, для лінгвістичної змінної «великий» fuzzy є «не дуже великий». На рисунку 2.6 наведено графіки функцій належності для «великих» нечітких множин та їх «не дуже великої» нечіткості [17].

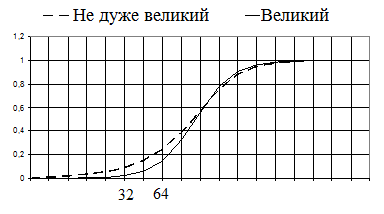


Рисунок 2.6 – Функції приналежності нечіткої множини «великі» та «не дуже великі»

Останні дві операції можна використовувати лише до нечітких множин.

На нечітких множинах також визначаються операції алгебраїчних добутків і сум. У цьому випадку алгебраїчний добуток нечітких множин і  називають множиною , функція належності якого визначається за формулою [17]:

(2.1)



А алгебраїчна сума — це множина з функцією належності [17]:

(2.2)



Використання алгебраїчного добутку дає змогу визначити поняття ступеня нечіткої множини , функція належності якого дорівнює .

Дозволяє  нечіткі підмножини базових множин , відповідно, то декартовий добуток (прямий) підмножин  буде називатися підмножиною  множина  з такою функцією членства:

(2.3)



Для визначення співвідношень на нечітких множинах використовується Декарт або прямий добуток нечітких множин [17].

Нечіткі вхідні значення системи перетворюються у вихідні на основі правил нечіткої логіки, яка реалізується на основі моделі логічного висновку.

Загалом, правило може включати всі можливі комбінації лінгвістичних термінів для всіх вхідних змінних, об’єднаних логічними операціями.

Використовуючи перетворення нечітких наборів, будь-яке правило, що містить як кон’юнкції, так і диз’юнкції з лівого боку, можна перетворити на систему правил, у лівій частині якої будуть або тільки кон’юнкції, або тільки диз’юнкції. Для визначення нечіткої кон’юнкції можна використати пошук мінімуму, а нечіткої диз’юнкції – пошук максимуму двох функцій належності [18].

У загальному випадку механізм логічного висновку складається з чотирьох етапів: введення нечіткого висновку (фасифікація), нечіткого висновку, композиції та чіткості, або дефазифікації (рисунок 2.7).

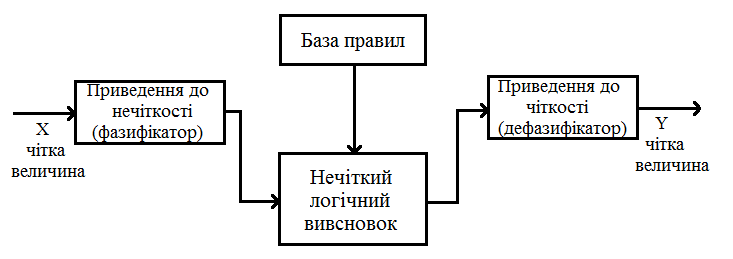


Рисунок 2.7 – Система нечіткіх висновків

Існують такі основні моделі логічного висновку: «Мамдані» (Mamdani) і «Takagi-Sugeno» (Takagi-Sugeno) [19].

Алгоритм Мамдані описує кілька послідовних кроків (рисунок 2.8). У цьому випадку кожен наступний крок отримує вхідні значення, отримані на попередньому кроці.



Рисунок 2.8 – Схема алгоритму нечіткого висновку Мамдані

Алгоритм відрізняється тим, що працює за принципом «чорного ящика». Кількісні значення вводяться, а також виводяться. На проміжних етапах використовується апарат нечіткої логіки та теорія нечітких множин. Це елегантність використання нечітких систем. Ви можете маніпулювати звичайними числовими даними, але використовувати гнучкі можливості, надані системами нечіткого висновку.

База правил — це набір правил, де кожен підвисновок порівнюється з певним ваговим коефіцієнтом.

Цей крок часто називають розмиттям. Вхідними є згенерована база даних правил і масив вхідних даних. Цей масив містить значення всіх вхідних змінних. Мета цього кроку — отримати значення істинності для всіх припущень із бази правил.

Умова правила може бути складеною, тобто включати підумови, пов’язані одна з одною за допомогою логічної операції «І». Метою цього кроку є визначення ступеня істинності умов для кожного правила системи нечіткого висновку. Простіше кажучи, для кожного стану існує мінімальна цінність істини з усієї підсвідомості.

На етапі активізації підвисновків відбувається перехід від умов до підвисновків. Метою цього кроку є отримання набору «активованих» нечітких наборів для кожного з підвисновків у певній базі правил.

Метою кроку накопичення висновків є отримання нечіткої множини (або їх комбінації) для кожної з вихідних змінних. Об’єднання двох нечітких множин – це третя нечітка множина.

Метою фазового зсуву є отримання кількісного значення для кожної з вихідних лінгвістичних змінних [20].

Механізм Мамдані відповідає розглянутому прикладу, показаному на малюнку 2.9.

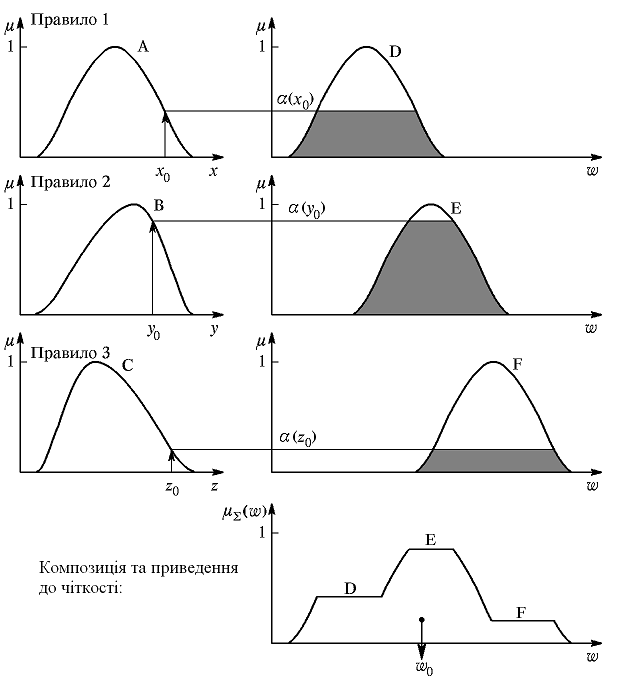


Рисунок 2.9 - Висновок механізму Мамдані

У цій ситуації математично це можна описати так:

1. нечітко: знайти ступені істинності для передумов кожного правила:;
2. нечіткий висновок: знайдіть рівні «відсікання» для передумов кожного з правил (за допомогою операції МІНІМУМ)

,,

;

де через "«Операція позначена логічним мінімумом (мін), потім знайдіть «урізані» функції членства:

,,

.

1. композиція: з використанням операції МАКСИМУМ (макс, далі іменується «") Поєднує знайдені усічені функції, що призводить до остаточної нечіткої підмножини для вихідної змінної з функцією належності

;

1. ясність (знайти) здійснюється, наприклад, методом центроїда [17].

Розроблений інструмент захисту файлів буде реалізований за допомогою програмного забезпечення Simulink, яке інтегровано в програмне середовище Matlab.

Matlab- пакет програм для чисельного аналізу, а також мова програмування, що використовується в цьому пакеті. Система створена The MathWorks і є зручним інструментом для роботи з математичними матрицями, функціями малювання, роботи з алгоритмами, створення робочих оболонок з програмами на інших мовах програмування. Хоча цей продукт спеціалізується на чисельних обчисленнях, з програмним забезпеченням Maple працюють спеціальні інструменти, що робить його повноцінною системою для роботи з алгеброю [21].

Simulink є інтерактивним інструментом для моделювання, моделювання та аналізу динамічних систем, включаючи дискретні, безперервні та гібридні, нелінійні та розривні системи. Розроблено The MathWorks. Надає можливість будувати графічні функціональні діаграми, моделювати динамічні системи, досліджувати продуктивність системи та покращувати проекти. Simulink повністю інтегрований з MATLAB, забезпечуючи швидкий доступ до широкого спектру інструментів аналізу та проектування.

Simulink також інтегрується з Stateflow для моделювання поведінки, викликаної подіями. Ця перевага робить Simulink найпопулярнішим інструментом для проектування систем керування та комутації, цифрової обробки та інших прикладних програм моделювання [22].

На основі аналізу загальна схема розробленого програмного забезпечення захисту файлів матиме такий вигляд (рисунок 2.10).

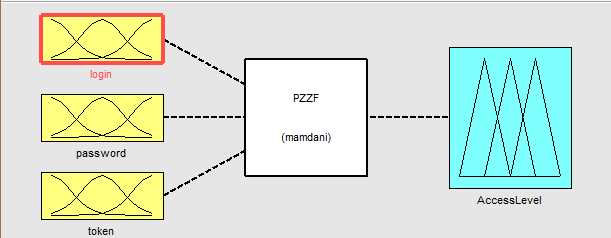


Рисунок 2.10 - Схема розробленого пристрою

У додатку А показана функціональна схема розробленого пристрою. З цієї діаграми можна зрозуміти, що люди, які запитують операції з читання, копіювання, створення, редагування або видалення файлу, повинні спочатку ввести комбінацію персонального ідентифікатора та пароля, вставити маркер у роз’єм USB, ввести PIN-код маркера та залежно від цього за довжиною логіна, пароля та ПІН-коду, після успішного входу працівник отримує рівень доступу для виконання відповідних операцій.

* 1. Реалізація проекту MATLAB

Об’єкт системи захисту файлів отримує нечітку інформацію про рівень доступу, необхідний для читання, копіювання, створення, зміни або видалення файлу.

Для виконання моделювання нечіткого висновку використовується алгоритм Мамдані, який описано в розділі 2.1.

Функції членства для змінних логіна, пароля та токена наведені відповідно на рисунках 2.11, 2.12 та 2.13 [26]. Вони розділені на три інтервали, кожен для точного опису змінних, зокрема, для вказівки довжини зв'язку використовується змінна, що вказує на малу довжину входу,  - середній і  велика довжина з’єднання (рисунок 2.11).

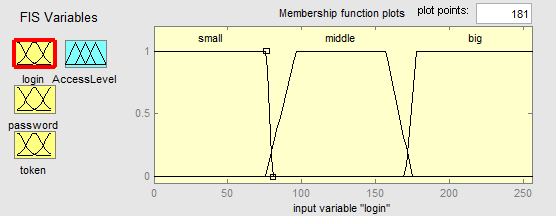
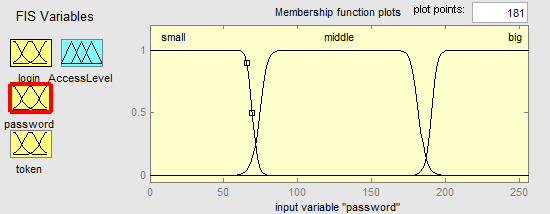


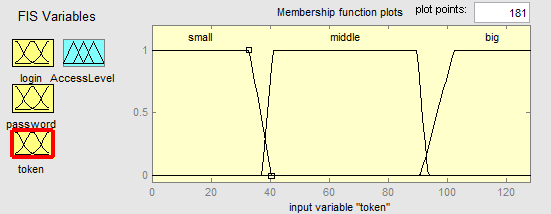
Рисунок 2.11‑ Функції зв'язку змінної зв'язку

Подібні змінні використовуються для встановлення довжини пароля вказує невелику довжину ключа, середній і  велика довжина ключа (рисунок 2.12).



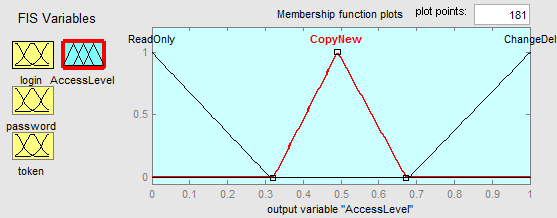
Графік 2.12‑ Функції членства пароля

Довжина PIN-коду, який вводиться при використанні токена, визначається значеннями   і , що відповідає відповідно малій, середній та великій довжині PIN-коду (рисунок 2.13).



Графік 2.13‑ Функції членства в маркерах

На малюнку 2.14 показані функції членства для вихідної змінної AccessLevel. Вони позначені відповідними інтервалами на осі y для точного визначення центру ваги, що вказує на нечіткий висновок системи [24]. ReadOnly означає доступ лише для читання, CopyNew означає доступ лише для читання, копіювання, створення файлів, а ChangeDel означає доступ лише для читання, копіювання, створення, зміну та видалення.



Графік 2.14‑ Функції членства змінних AccessLevel

У Додатку B перераховано модель, побудовану на класичному механізмі Мамдані, розробленому MATLAB.

База знань для побудови цієї нечіткої моделі складається з таких правил, як «якщо - то» [27], всі вхідні змінні мають три нечітких стани і один стан – жодного, коли значення вхідної змінної не має, не задається системою. Кількість правил нечітких висновків досліджуваної системи, тому що об’єднання, коли значення всіх вхідних змінних не вказані, з практичної точки зору неможливе.

Нижче наведено деякі правила, які формують базу знань для побудови нечіткої моделі:

1. Якщо (логін невеликий) і (пароль невеликий), то (Рівень доступу лише для читання) (1)
2. Якщо (логін невеликий) і (токен невеликий), то (Рівень доступу лише для читання) (1)
3. Якщо (пароль невеликий) і (токен невеликий), то (Рівень доступу лише для читання) (1)
4. Якщо (логін невеликий) і (пароль невеликий) і (токен знаходиться посередині), то (Рівень доступу лише для читання) (1)
5. Якщо (логін невеликий) і (пароль знаходиться посередині) і (токен невеликий), то (Рівень доступу лише для читання) (1)
6. Якщо (логін знаходиться посередині) і (пароль невеликий) і (токен невеликий), то (Рівень доступу лише для читання) (1)
7. Якщо (логін невеликий) і (пароль невеликий) і (токен великий), тоді (Рівень доступу – CopyNew) (1)
8. Якщо (логін невеликий) і (пароль великий) і (токен невеликий), то (Рівень доступу — CopyNew) (1)
9. Якщо (великий логін) і (пароль малий) і (токен малий), тоді (Рівень доступу — CopyNew) (1)
10. Якщо (логін посередині) і (пароль посередині), то (Рівень доступу лише для читання) (1)
11. Якщо (логін знаходиться посередині) і (токен знаходиться посередині), то (Рівень доступу лише для читання) (1)
12. Якщо (пароль посередині) і (токен посередині), то (Рівень доступу лише для читання) (1)
13. Якщо (логін знаходиться посередині) і (пароль знаходиться посередині) і (токен невеликий), то (Рівень доступу – CopyNew) (1)
14. Якщо (логін знаходиться посередині) і (пароль невеликий) і (токен знаходиться посередині), то (Рівень доступу — CopyNew) (1)
15. Якщо (логін невеликий) і (пароль знаходиться посередині) і (токен знаходиться посередині), тоді (Рівень доступу — CopyNew) (1)

Додаток B містить систему правил нечіткого висновку.

* 1. Функціональне моделювання проекту

Нечіткий висновок моделі забезпечення рівня доступу, побудованої на основі 63 правил, заданих з поточними значеннями логіна, пароля, токена та змінних AccessLevel, має вигляд, показаний на рисунку 2.15 [27].

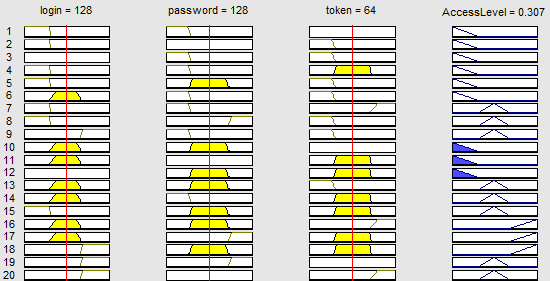


Рисунок 2.15 ‒ Нечіткий висновок моделі рівня доступу

Перевірити правильність функціонування системи можна шляхом аналізу наданих даних на основі розроблених правил, які наведені в таблиці 2.1.

Таблиця 2.1 - Перевірка правильності виконання правил

|  |  |  |  |
| --- | --- | --- | --- |
| підключення | пароль | токен | Рівень доступу |
| 54 | 41 | 19 | 0,101 |
| 87 | 64 | 33 | 0,122 |
| 122 | 125 | 64 | 0,307 |
| 205 | 193 | 96 | 0,449 |
| десять | 217 | 9 | 0,307 |
| 184 | 199 | 106 | 0,5 |
| 256 | 256 | 128 | 0,501 |

Поверхні значень нечіткої системи на основі механізму Мамдані представлені на малюнках 2.16, 2.17, 2.18.

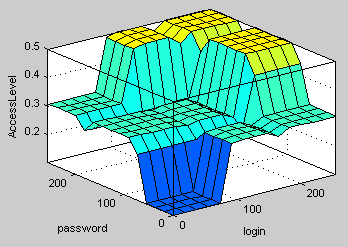


Рисунок 2.16 - Поверхневі значення виходу нечіткої системи на основі механізму Мамдані як функція пароля та логіну

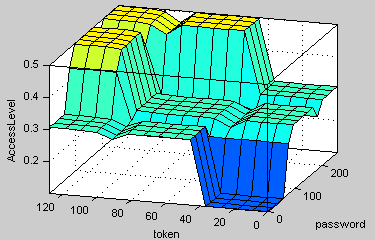


Рисунок 2.17 - Поверхня вихідних значень нечіткої системи на основі механізму Мамдані як функція токена та пароля

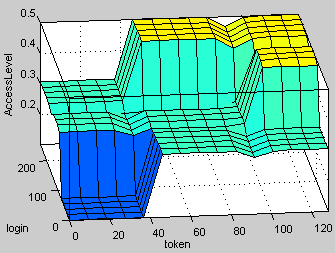


Рисунок 2.18 - Значення площ виводу нечіткої системи на основі механізму Мамдані як функція логіна та токена

На основі цих поверхонь і таблиці 2.1 можна перевірити правильність побудови бази правил нечіткого висновку.

Основний недолік нечіткого висновку, побудованого на класичному механізмі Мамдані, полягає в тому, що для будь-яких вхідних даних необхідно розробити всю базу правил, тобто виконати три кроки. Однак такий спосіб побудови засобу захисту файлів дозволяє працювати в системі реального часу, що відповідає вимогам сучасних комп’ютерних систем.

1. РЕАЛІЗАЦІЯ ПРОЕКТУ ЗАХИСТУ ФАЙЛІВ
   1. Розробка моделі нечіткого контролера за допомогою «Simulink»

Щоб побудувати модельпрограмне забезпеченняспособи захист файлів на основі нечіткої логікиВи можете використовувати інтерактивний інструмент Simulink, повністю інтегрований з Matlab. Це дозволяє моделювати, моделювати та аналізувати динамічні системи.

Під час виконання інструмент моделюваннязахист файлів на основі нечіткої логіки За допомогою інтерактивного інструменту Simulink були використані наступні основні компоненти: джерело випадкових чисел (випадкове число), осцилограф (осцил), контролер нечіткої логіки та мультиплексор.

Метою джерела випадкового сигналу з нормальним розподілом (випадковим числом) є формування випадкового сигналу з нормальним розподілом рівня сигналу (рисунок 3.1).

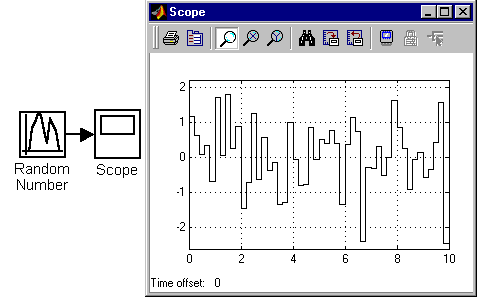


Рисунок 3.1 - Джерело випадкового сигналу з нормальним розподілом

Його основними параметрами є:

* середнє - середнє значення сигналу;
* дисперсия – дисперсія (стандартне відхилення);
* насіння початкове - початкове значення [28].

Метою інструмента осцилу є побудова графіка досліджуваних сигналів як функції часу. Це також дозволяє відстежувати зміни сигналу під час процесу моделювання.

Зображення блоку та вікно відображення графіка показано на малюнку 3.2 [29].

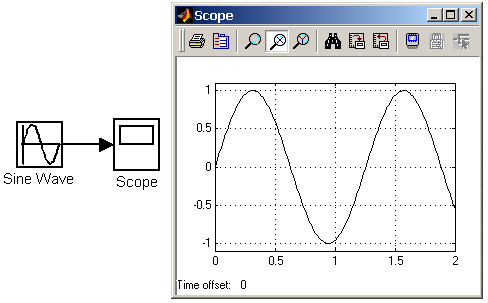


Рисунок 3.2 – Осцилограф

Щоб увімкнути систему нечіткого висновку в модулі Simulink, виберіть блок контролера нечіткої логіки, потім двічі клацніть цей блок і введіть ім’я файлу або ім’я змінної в діалоговому вікні, що відповідає логічним завершенням контролера нечіткої логіки. Зображення блоку контролера нечіткої логіки показано на рисунку 3.3.

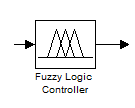


Рисунок 3.3 - Блок керування нечіткою логікою

Якщо система нечіткого висновку має кілька входів, то в модулі Simulink ці входи повинні бути мультиплексовані разом перед входом у нечіткий контролер. Аналогічно, якщо система нечіткого висновку має кілька виходів, то вихідні сигнали блоку будуть представлені однією мультиплексною лінією [30].

Щоб виконати цю дію, використовуйте блок мультиплексування. Він об’єднує вхідні сигнали у вектор. Приклад застосування цього блоку проілюстрований на малюнку 3.4.

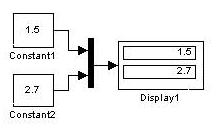


Рисунок 3.4 - Приклад використання блоку мультиплексора

Параметри цього блоку:

1. число записів - кількість записів;
2. Параметр відображення - спосіб відображення. Вибрано зі списку:

* смуга - вузький вертикальний прямокутник чорного кольору;
* сигнали - прямокутник з білим фоном і мітками вхідного сигналу;
* none - прямокутник з білим фоном без показу міток введення.

Вхідні сигнали блоку можуть бути скалярними та/або векторними. Якщо серед вхідних сигналів є вектори, то кількість входів можна задати як вектор, що вказує кількість елементів кожного вектора.

Наприклад, вираз [2 3 1] визначає три вхідних сигнали, перший сигнал є вектором двох елементів, другий сигнал є вектором трьох елементів, а останній сигнал є скаляром. Якщо розмірність вхідного вектора не відповідає кількості входів, зазначених у параметрі, Simulink видасть повідомлення про помилку після початку обчислення. Розмір вхідного вектора можна встановити на -1 (мінус один). При цьому розмірність вхідного вектора може бути довільною.

Параметр Number of inputs також можна визначити як список міток сигналу, наприклад: Vector1, Vector2, Scalar. У цьому випадку мітки сигналів будуть відображатися поруч з відповідними лініями підключення. Сигнали, що надходять на входи блоку, повинні бути однотипними (дійсними або комплексними) [31].

Виходячи з вищесказаного, на малюнку 3.5 показана модель нечіткої системи на основі класичного механізму Мамдані, який був описаний у розділі 2.1. [32].

Використовувати раніше створену систему нечіткого захисту файлів (AccessLevel), яка залежить від значень довжини з’єднання (логін), пароля (паролю) та PIN-коду токена (токена), здійснюваних за допомогою програмного забезпечення Fuzzy Logic Toolbox у нечіткому контролері вам потрібно відкрити налаштування контролер нечіткої логіки та введіть між одинарними лапками шлях до файлу, що містить зареєстровану нечітку систему, та ім’я файлу з відповідним розширенням.

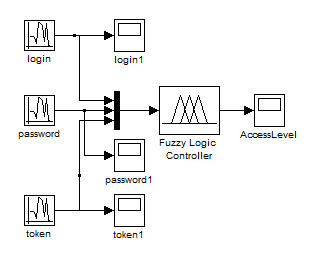


Рисунок 3.5 - Модель розробленого інструменту

Приклади параметрів налаштування контролера нечіткої логіки показаний на рисунку 3.6.

Розроблений нечіткий контролер (Fuzzy Logic Controller) працює за механізмом Мамдані. Його вхідними параметрами є довжина входу в систему, пароль і довжина PIN-коду маркера, а його вихід — значення центру ваги, яке інтерпретує рівень доступу (AccessLevel).

Схема контролер нечіткої логіки складається з трьох основних блоків:

* блок опису функцій належності вхідних змінних (блоки Input MF);
* блок опису вихідних функцій належності (Output MF);
* виходи двох попередніх блоків, які надходять на вхід 63 правила (блоки Rule 1…63).

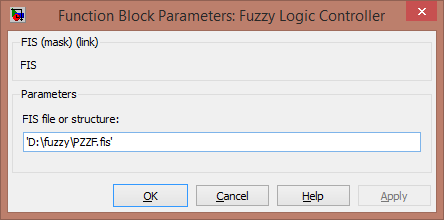


Рисунок 3.6 - Параметриконтролер нечіткої логіки

Випадково вхідна змінна підключення визначається рівномірним розподілом (рисунок 3.7).

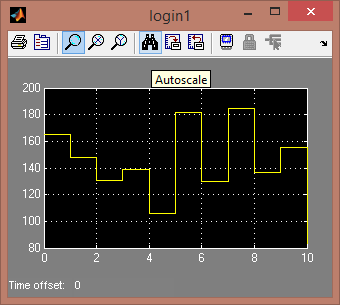


Рисунок 3.7 - Рівномірно розподілене призначення випадкових значень вхідної змінної login

Так само відповідно до рівномірного розподілу встановлюється змінна вхідного пароля (рисунок 3.8).

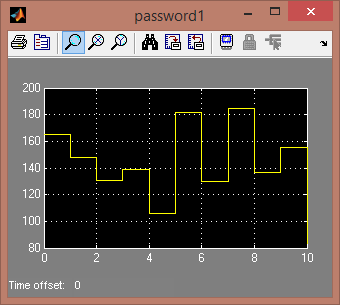


Рисунок 3.8 - Рівномірно розподілене призначення випадкових значень вхідної змінної пароля

Вхідна змінна токена також встановлюється на основі випадкових значень (рисунокнок 3.9).

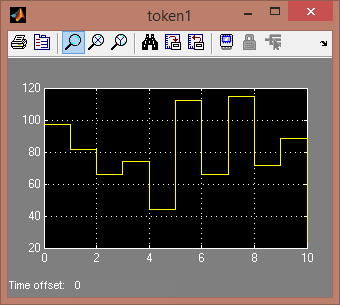


Рисунок 3.9 - Рівномірно розподілене призначення випадкових значень вхідної змінної токена

В результаті за допомогою джерела випадкового сигналу з нормальним розподілом були згенеровані випадкові сигнали з нормальним розподілом рівня сигналу для змінних логіну, пароля та токена.

Фрагмент розробленої схеми нечіткого регулятора показано на рисунку 3.10.

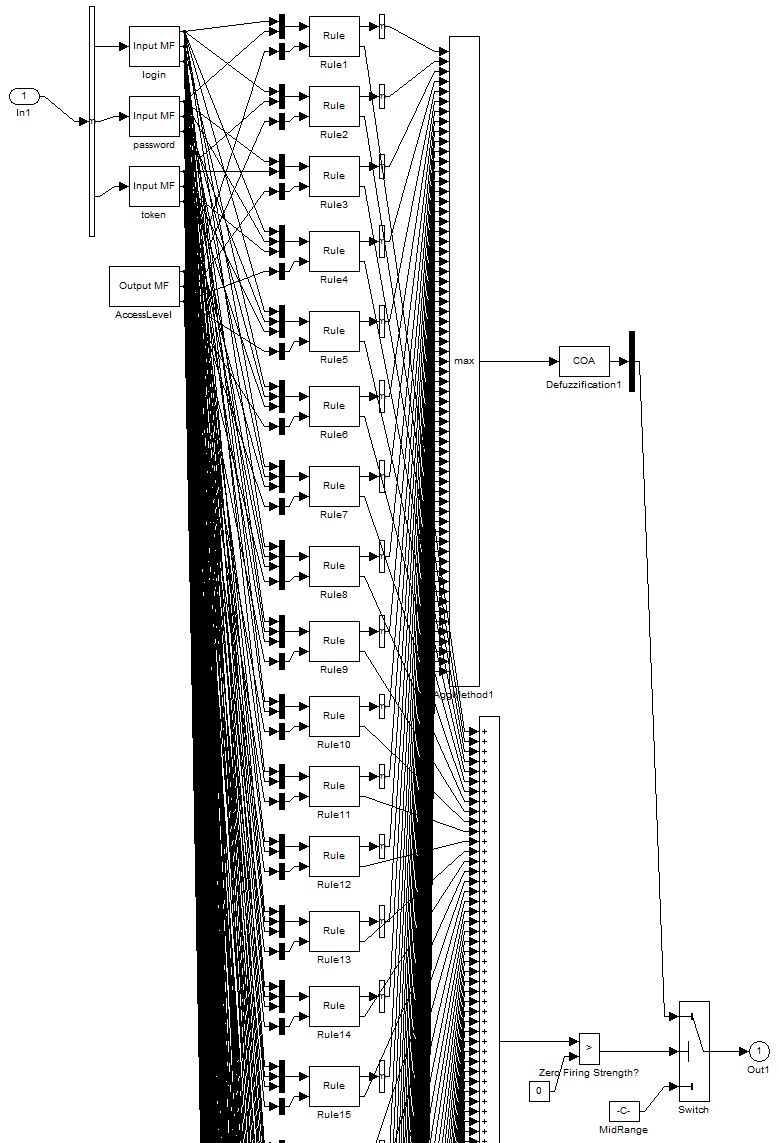


Рисунок 3.10 - Фрагмент розробленої схеми нечіткого контролера

* 1. Аналіз роботи розробленого нечіткого контролера

Нечітка підмножина відрізняється від звичайної підмножини тим, що для елементів x множини E не існує однозначної відповіді «ні» на властивість R. У зв’язку з цим нечітка підмножина A універсальної множини E визначається як множина впорядкована пара A = {µA (x) / x}, де µA (x) є характеристичною функцією належності (або просто функцією належності), яка приймає значення в упорядкованій множині M (наприклад, M = [0, 1 ]).

Функція належності вказує ступінь або рівень належності елемента x до підмножини A. Множина M називається набором аксесуарів. Якщо M = {0,1}, то нечітку підмножину A можна розглядати як звичайну або чітку множину.

Наприклад, ми можемо представити множину X усіх чисел у діапазоні від 0 до 10. Далі визначимо підмножину A множини X — усі дійсні числа від 5 до 8 (A = [5, 8]).

Функція належності до множини A відповідає значенню 0 або 1 для кожного елемента множини X, якщо цей елемент не належить або належить до підмножини A. Для кращого сприйняття результат показаний на рисунку 3.11 [16].

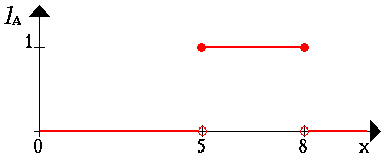


Рисунок 3.11 – Візуалізація функції належності множини A

Елементи, що відповідають значенню 1, можна вважати такими, що знаходяться в множині A, а елементи, що відповідають значенню 0, не знаходяться в множині A.

Цей підхід використовується в багатьох програмах. Однак є багато випадків, коли цей підхід не може бути використаний через його недостатню гнучкість.

Цей підхід використовує прямий метод визначення функції належності, в якому експерт визначає для кожного x *Е*значення µA (x) або сама функція належності. Як правило, прямі методи визначення функції належності використовуються для таких вимірювань, як час, швидкість, відстань, температура, тиск, тобто ті, які можна виміряти (полярні значення).



Характеризуючи об’єкт, у багатьох задачах можна вибрати набір певних характеристик і для кожної з них визначити полярні значення, які відповідають значенням функції належності, 1 або 0.

У таблиці 3.1 наведені полярні значення для розробленого елементапрограмаспособи захист файлів на основі нечіткої логіки [16].

Таблиця 3.1 - Полярні значення для програм захисту файлів

|  |  |  |  |
| --- | --- | --- | --- |
| ?? | ім'я | 1 | 0 |
| *X*1 | довжина з'єднання | Чудово | малий |
| *X*2 | довжина пароля | Чудово | малий |
| *X*3 | довжина PIN-коду маркера | Чудово | малий |

Для конкретного рівня доступу А експерт на основі таблиці фіксує µA (x) [0,1], утворюючи векторну функцію належності {µA (x1), µA (x2), µA (x3)}.



Непрямі методи визначення значень функції належності використовуються в тих випадках, коли для визначення нечіткої множини не вимірюються елементарні властивості. Як правило, це методи попарного порівняння. Якщо були відомі значення функцій належності, наприклад µA (xi) = wi, i = 1,2, ..., n, то попарні порівняння можна представити матрицею відношень A = {aij} , де aij = wi / wj (операція поділу)) [16].

На рисунку 3.12 наведена діаграма функцій належності вхідних і вихідних змінних, яка була побудована за допомогою інтерактивного інструменту Simulink.

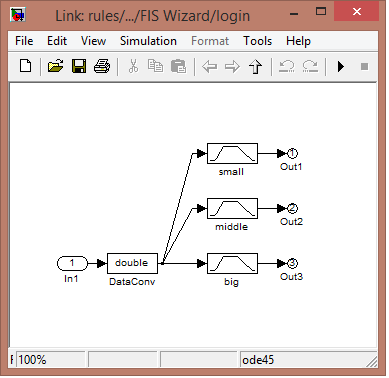


Рисунок 3.12 Діаграма визначення функцій належності вхідних і вихідних змінних нечіткого контролера

Побудова кінцевої фігури, а також обробка нечітких змінних для знаходження центру ваги, які є нечітким висновком за механізмом Мамдані, здійснюється за алгоритмом, представленим на рисунку 2.9.

На рисунку 3.13 показано обробку правил з бази знань за допомогою інструменту Simulink з урахуванням оцінки, яка відображається константою Weight.

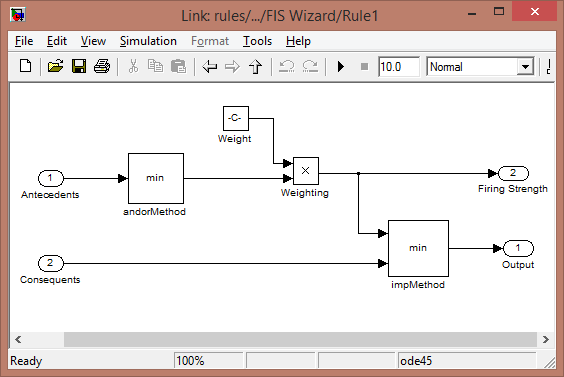


Рисунок 3.13 - Схема обробки вхідних нечітких значень за правилом «якщо то»

У Додатку D показана блок-схема обробки вхідних нечітких значень за правилом «якщо то».

Значення вхідних змінних логіна, пароля та PIN-коду токена є входами правила і відповідають входу 1, а вхід 2 відповідає наданню певного рівня доступу. У блоці min, за законом мінімуму, ці дані обробляються. Функції приналежності виходу AccessLevel відповідають виходу 1, а послідовність, яка відображає інтервал його роботи, відповідає виходу 2.

Нечіткий контролер виконує пошук центру ваги кінцевої цифри, яка формується в результаті підсумовування виходів 63 правил, тобто за механізмом Мамдані виконується процес фазового зсуву. Дефазіфікацію нечіткого висновку проводять за формулою:

,, (3.1)

або значення осі абсцис;

 значення ординати -я цифра;

 кількість прямокутників, на які розбивається остаточна фігура.

Фактична схема дефазифікації нечіткого висновку показана на рисунку 3.14.

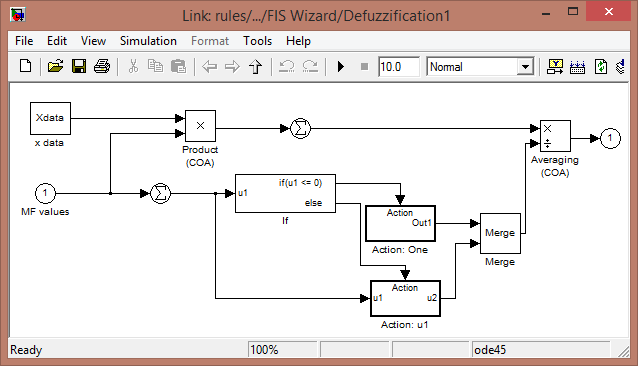


Рисунок 3.14 Діаграма фазового зсуву нечіткого висновку

У Додатку Е показана блок-схема фазового зсуву нечіткого висновку.

Значення центру ваги, тобто результат моделі при встановленні вхідних значень довжини входу, пароля та PIN-коду токена з рівномірним розподілом показано на рисунку 3.15.

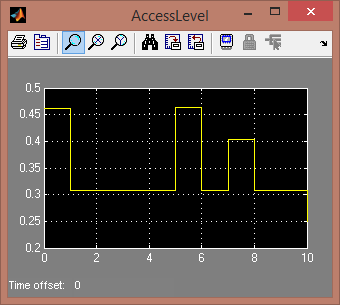


Рисунок 3.15 – Результати розробленої нечіткої моделі

Тестові значення вхідних і вихідних значень системи захисту нечітких файлів за механізмом Мамдані наведені в таблиці 3.2.

Таблиця 3.2 - Тестові значення змінних нечіткої системи захисту файлів, побудованої на механізмі Мамдані

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| \ п | Увійти | Пароль | Токен | Рівень доступу |
| 1 | 167 | 167 | 98 | 0,46 |
| 2 | 149 | 149 | 82 | 0,31 |
| 3 | 131 | 131 | 68 | 0,31 |
| 4 | 140 | 140 | 75 | 0,31 |
| 5 | 108 | 108 | 44 | 0,31 |
| 6 | 181 | 181 | 114 | 0,46 |
| 7 | 130 | 130 | 68 | 0,31 |
| 8 | 184 | 184 | 116 | 0,405 |
| 9 | 138 | 138 | 74 | 0,31 |
| десять | 156 | 156 | 89 | 0,31 |

Тому коректне функціонування інструменту захисту файлів, розробленого за класичним механізмом нечітких висновків Мамдані, підтверджується стандартними тестами середовища Simulink.

Відповідно до поточних вимог комп’ютерної системи щодо довжини входу, пароля та PIN-коду токена, для оптимальної роботи та надійного захисту файлів використовується механізм нечіткої логіки Мамдані, який діє за законом мінімуму.

* 1. Стійкість розробленого програмного забезпечення до несанкціонованого доступу

На основі ідентифікації користувачів буде забезпечена стійкість розробленого програмного забезпечення до несанкціонованого доступу.

При запуску програми або програми керування користувачами з’явиться вікно, в якому необхідно ввести ім’я користувача та пароль (рисунок 3.16).

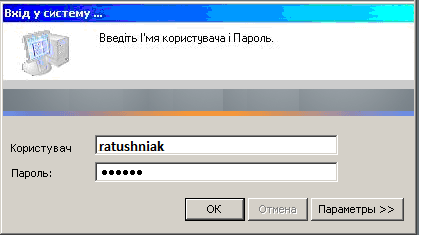


Рисунок 3.16 ‒ Запит пароля для входу

У разі збігу введених даних користувач входить в систему, в іншому випадку - здійснюється вихід. Після входу користувача в систему перевіряються всі права на використання певних системних об’єктів. Якщо ви спробуєте отримати доступ до об’єкта, на який користувач не має прав, з’явиться повідомлення «Доступ заборонено» (рисунок 3.17).

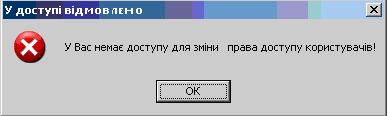


Рисунок 3.17 - Повідомлення «Доступ заборонено»

Адміністративні засоби, призначені для спільного доступу під час роботи в мережі персонального комп’ютера, а також для вирішення проблем інтеграції містяться в додатку «Менеджер користувачів» (рисунок 3.18).

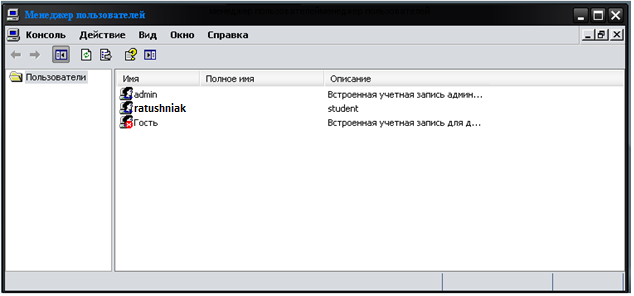


Рисунок 3.18 ‒ Головне вікно програми «Диспетчер користувачів»

Для входу кожному користувачеві можна призначити певний пароль.

Під час підключення виконуються такі дії:

1. перевірити, чи є користувач з таким іменем;
2. паролі перевіряються;
3. перевіряються права доступу користувача до дій та об'єктів системи.

Усі основні елементи керування програмою розташовані в головному вікні програми.

У цьому вікні ви можете виконувати всі дії з записами користувачів, наданими системою. Наприклад, ви можете видалити, створити новий, змінити пароль, а також переглянути та редагувати властивості потрібного користувача. Щоб створити нового користувача, виберіть контекстне меню у списках користувачів і виберіть «Новий користувач». Після цього з’явиться вікно, показане на рисунку 3.19

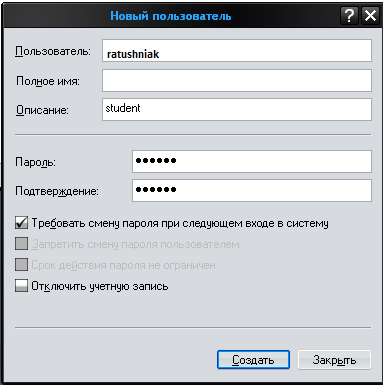


Рисунок 3.19 ‒ Створення нового користувача

Ви також можете виконати цю дію, вибравши опцію «Новий користувач» у підменю «Дії» головного меню «Менеджер користувачів». На екрані з'явиться вікно «Новий користувач кно». У відповідь на запит необхідно ввести ім’я (логін) нового користувача, яке не повинно перевищувати 24 символів, а також не повинно містити спеціальних символів або пробілів.

Ви також можете ввести повне ім’я користувача, яке не повинно перевищувати 24 символи.

При створенні нового користувача в цьому вікні можна ввести коротку інформацію про користувача (посада, опис виконаної роботи тощо).

Нижче потрібно ввести новий пароль користувача та підтвердити його, щоб перевірити правильність.

Якщо вибрано пункт «Вимагати зміни пароля під час наступного входу», під час першого входу вам потрібно буде зберегти поточний пароль, а потім змінити його на новий.

Видалення користувача схоже на створення, але ви повинні вибрати «видалити». Ви повинні підтвердити видалення користувача, натиснувши відповідну кнопку (рисунок 3.20).

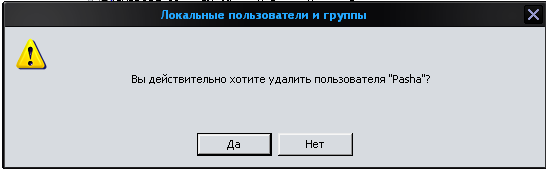


Рисунок 3.20 – Запит на видалення користувача

Слід зазначити, що цей додаток також має кілька обов’язкових опцій. Ці параметри знаходяться у властивостях користувача.

Ви можете призначити обмеження використання будь-якому користувачеві на основі дати та часу у вашому профілі. Наприклад, можна вказати час з 9:00 до 18:00, тобто ввести можна тільки в робочий час (рисунок 3.21).

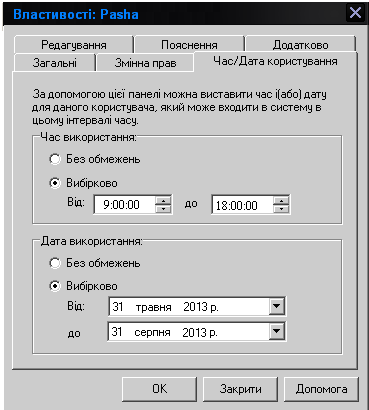


Рисунок 3.21 – Параметри виконання користувача

У випадку, якщо людина працює тимчасово, наприклад, стажується студентами, можна використовувати термін.

На додаток до цих функцій є й інші, не менш важливі, наприклад надання користувачам прав адміністратора. За допомогою параметра «адміністративний доступ», незалежно від інших груп прав доступу до певних програмних об’єктів, користувач матиме право доступу до інформації, створення файлів і папок, авторизації редагування, дозволу на видалення.

Вікно надання прав адміністратора користувачам показано на рисунку 3.22.

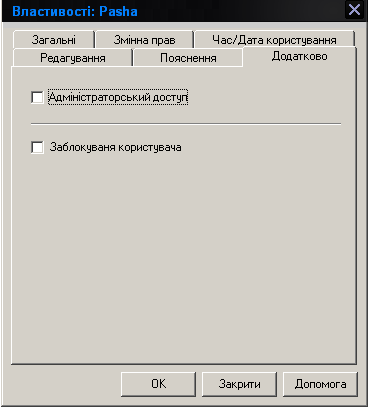


Рисунок 3.22 ‒ Інші налаштування для користувачів

Щоб заблокувати з’єднання, ви можете скористатися опцією «Блокувати користувача», поки його не розблокує адміністратор.

Ці права належать адміністратору. Якщо вам потрібно виконати дії власника пароля, ви можете скористатися цими властивостями користувача. Доступ до цієї опції має лише адміністратор. Це певний контроль над діями користувача.

Така стабільність програмного забезпечення з використанням прав доступу користувачів досить надійна для несанкціонованого доступу і може використовуватися в будь-якій комп’ютерній системі.

ВИСНОВКИ

В галузі обчислювальної техніки поняття безпеки є досить широким. Під ним розуміється і надійність роботи комп'ютера, і збереження цінних даних, і захист певної інформації від внесення до неї змін неуповноваженими особами.

У даному дипломному проекті було розроблено програмний засіб захисту файлів на основі нечіткої логіки шляхом:

1. виконання аналізу сучасних апаратно-програмних засобів захисту інформації;
2. вибору методу захисту файлів для розроблюваної системи;
3. проведення аналізу технічного завдання та постановки задачі;
4. розробки структурної схеми пристрою, його реалізації засобами MATLAB, а також виконанням функційної симуляції проекту;
5. здійснення реалізації проекту, яка включає в себе розробку моделі нечіткого контролера засобами «Simulink», перевірку правильності його роботи, а також дослідження стійкості розробленого програмного засобу до несанкціонованого доступу;
6. проведення аналізу санітарно-гігієнічних умов праці, а також стану пожежної безпеки приміщення, в якому виконувалась розробка програмного засобу захисту файлів.

Реалізований проект за допомогою засобів MATLAB працює коректно, в чому можна переконатись провівши функційну симуляцію проекту. Дані, отримані в результаті цієї симуляції, близькі до тих, які були визначені при перевірці правильності роботи нечіткого контролера засобами «Simulink».

Результати дипломного проектування використано практично.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. <http://www.bestreferat.ru/referat-42951.html>
2. <http://www.victoria.lviv.ua/html/oio/html/theme11_1_rus.htm#11_8>
3. <http://referats.allbest.ru/programming/9000152518.html>
4. <http://ref.rushkolnik.ru/v59466/?page=4>
5. <http://masters.donntu.edu.ua/2013/fknt/ippolitov/diss/indexu.htm#ref6>
6. <http://oim.asu.kpi.ua/files/DM/23_Cryptography.pdf>
7. <http://works.doklad.ru/view/Jl0lQEgYEMs.html>
8. <http://znaimo.com.ua/Ключ_(криптографія)>
9. ua-referat.com/Криптографічні\_методи\_захисту\_інформації.doc‎
10. <http://otherreferats.allbest.ru/management/d00203016.html>
11. <http://ref.rushkolnik.ru/v59466/?page=2>
12. <http://okbsapr.ru/konyavskaya_2010_1.html>
13. <http://www.pglu.ru/lib/publications/University_Reading/2009/XII/uch_2009_XII_00054.pdf>
14. <http://www.victoria.lviv.ua/html/oio/html/theme11.htm#11_1>
15. <http://ua.convdocs.org/docs/index-80899.html?page=11>
16. [http://www.victoria.lviv.ua/html/oio/html/theme11.htm#11](http://www.victoria.lviv.ua/html/oio/html/theme11.htm#11_2)
17. <http://otherreferats.allbest.ru/mathematics/00123125_0.html>
18. Нечеткая логика: алгебраические основы и приложения: Монография/С.Л. Блюмин, И.А. Шуйкова, П.В. Сараев, И.В. Черпаков. – Липецк: ЛЭГИ, 2002. – 113 с.
19. <http://www.basegroup.ru/library/analysis/fuzzylogic/math/>
20. <http://habrahabr.ru/post/113020/>
21. <http://uk.wikipedia.org/wiki/MATLAB>
22. <http://uk.wikipedia.org/wiki/Simulink>
23. <http://matlab.exponenta.ru/fuzzylogic/>
24. Штовба С.Д. Обеспечение точности и прозрачности нечеткой модели Мамдани при обучении по экспериментальным данным / С.Д.Штовба // Проблемы управления и информатики. – 2007. – №4. – С. 102–114.
25. Гостев В.И. Определение управляющих воздействий на выходе нечеткого регулятора при идентичных треугольных функциях принадлежности с увеличенным наклоном / В.И.Гостев, С.Н.Скуртов, И.В.Панченко // Вісник Хмельницького національного університету. Технічні науки. – 2007. - № 5. – С.253-256.
26. <http://matlab.exponenta.ru/simulink/book1/9_1.php>
27. <http://matlab.exponenta.ru/simulink/book1/9_2.php>
28. <http://matlab.exponenta.ru/fuzzylogic/book2/3/fuzblock.php>
29. <http://matlab.exponenta.ru/simulink/book1/9_7.php>
30. Дубчак Л.О. Спосіб обробки нечіткої інформації / Л.О.Дубчак // Вісник Східноукраїнського національного університету ім. В.Даля. – 2012. - № 8 (179), Ч.1. – С. 306-309.
31. <http://zakon4.rada.gov.ua/laws/show/2694-12>
32. <http://bibliofond.ru/view.aspx?id=6242#1>
33. <http://ref.rushkolnik.ru/v38127/?page=7>