Вінницький національний технічний університет

Факультет менеджменту та інформаційної безпеки

Кафедра менеджменту та безпеки інформаційних систем

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

Підвищення достовірності автентифікації користувача на основі захищеного електронного ключа та аналізу ентропії рухів миші

Виконав: ст. 2-го курсу, групи УБ-21м
спеціальності 125– Кібербезпека
Освітня програма – Управління інформаційною безпекою
Берестенко М.О.
Керівник: д.ф., ст.викл. каф. МБІС
Салієва О.В.
«15» грудня 2022 р.
Опонент: к.т.н., доц., доцент каф. ОТ
Савицька Л.А.
«15» грудня 2022 р.

Допущено до захисту
Голова секції УБ кафедри МБІС
Юрій ЯРЕМЧУК
“15” грудня 2022 р.

Вінниця ВНТУ - 2022 рік

Вінницький національний технічний університет
Факультет менеджменту та інформаційної безпеки
Кафедра менеджменту та безпеки інформаційних систем

Рівень вищої освіти ІІ-й (магістерський)
Галузь знань 12 – Інформаційні технології
Спеціальність 125 – Кібербезпека
Освітньо-професійна програма – Управління інформаційною безпекою

ЗАТВЕРДЖУЮ
Голова секції УБ, кафедра МБІС
_____ **Юрій ЯРЕМЧУК**
“ 15 ” вересня 2022 р.

З А В Д А Н Н Я

на магістерську кваліфікаційну роботу студенту

Берестенку Михайлу Олександровичу

1. Тема роботи «Підвищення достовірності автентифікації користувача на основі захищеного електронного ключа та аналізу ентропії рухів миші»

Керівник роботи Салієва О.В. д.ф., ст. викл. затверджені наказом вищого навчального закладу від «14» вересня 2022 року № 203

2. Строк подання студентом роботи за тиждень до захисту

3. Вихідні дані до роботи: нормативно-правова база, монографії та сучасні наукові статті по темі, Інтернет-ресурси, стандарти, існуюче ПЗ.

4. Зміст текстової частини:

– в першому розділі проаналізувати сучасні методи автентифікації користувачів, особливості двофакторної автентифікації;

– в другому розділі здійснити вдосконалення методу, провести проектування розробки, розробити алгоритми програмної частини;

– в третьому розділі здійснити програмну реалізацію розробки та аналіз результатів;

– в четвертому розділі проаналізувати економічну ефективність розробленого програмного забезпечення.

5. Перелік ілюстративного матеріалу (з точним зазначенням обов`язкових креслень):

– у першому розділі наведено 7 рис., 2 табл.;

– у другому розділі наведено 6 рис.;

– у третьому розділі наведено 29 рис.;

– у четвертому розділі наведено 1 рис. та 6 табл.

6. Консультанти розділів магістерської кваліфікаційної роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1	д.ф., ст. викл. каф. МБІС Салієва О.В.		
2	д.ф., ст. викл. каф. МБІС Салієва О.В.		
3	д.ф., ст. викл. каф. МБІС Салієва О.В.		
4	зав. каф. ЕПВМ, к.т.н. Лесько О.Й.		

7.Дата видачі завдання 15 вересня 2022 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів МКР	Строк виконання етапів МКР	Примітка
1.	Визначення напрямку МКР, формулювання теми	15.09.2022	30.09.2022
2.	Аналіз предметної області обраної теми	01.10.2022	15.10.2022
3.	Розробка алгоритму роботи	16.10.2022	31.10.2022
4.	Написання МКР на основі розробленої теми	01.11.2022	15.11.2022
5.	Розробка економічної частини	15.11.2022	23.11.2022
6.	Попередній захист МКР	24.11.2022	25.11.2022
7.	Виправлення, уточнення, коригування роботи	26.11.2022	15.12.2022
8.	Захист МКР	19.12.2022	21.12.2022

Студент Берестенко М.О.

Керівник МКР Салієва О.В.

АНОТАЦІЯ

УДК 621.391

Берестенко М.О. Підвищення достовірності автентифікації користувача на основі захищеного електронного ключа та аналізу ентропії рухів миші. Магістерська кваліфікаційна робота зі спеціальності 125 – «Кібербезпека», освітня програма «Управління інформаційною безпекою». Вінниця: ВНТУ, 2022. 116 с.

На укр. мові. Бібліогр.: 57 назв; рис.: 42; табл. 8.

У магістерській кваліфікаційній роботі здійснено підвищення достовірності автентифікації користувача на основі захищеного електронного ключа та аналізу ентропії рухів миші.

В роботі досліджено сучасні методи автентифікації користувачів, здійснено аналіз особливостей двофакторної автентифікації, описано алгоритми автентифікації користувачів з використанням електронного ключа та поведінкових характеристик.

Удосконалення алгоритму автентифікації користувача у захищеному додатку полягає у реалізації двофакторної автентифікації та застосуванні захищеного електронного ключа з використанням алгоритму AES та аналізу ентропії рухів миші, що розраховуються за певними параметрами і є індивідуальною характеристикою для кожного користувача.

Здійснено практичну реалізацію програмного додатку, проведений аналіз програмної розробки показав, що додаток працює коректно, стійкість до зламу електронного ключа зумовлена обраною технологією JWT, аналіз дослідження показників FAR та FRR показав підвищення достовірності розпізнавання на 11%. Економічна доцільність розробки підтверджена проведеним аналізом.

Ключові слова: електронний ключ, ентропія рухів мишки, двофакторна автентифікація, доступ, захист від НСД.

ABSTRACT

Mykhailo Berestenko. Increasing the authenticity of user authentication based on a secure electronic key and mouse movement entropy analysis. Master's thesis in specialty 125 – «Cyber Security», Education Program «Information Security Management». Vinnitsa: VNTU, 2022. – 116 p.

In Ukrainian language. Bibliographer: 57 titles; fig.: 42; tabl. 8.

In the master's qualification work, the reliability of user authentication was improved based on a secure electronic key and analysis of the entropy of mouse movements.

The work examines modern user authentication methods, analyzes the features of two-factor authentication, describes user authentication algorithms using an electronic key and behavioral characteristics.

The improvement of the user authentication algorithm in the secure application consists in the implementation of two-factor authentication and the application of a secure electronic key using the AES algorithm and the analysis of the entropy of mouse movements, which are calculated according to certain parameters and are an individual characteristic for each user.

The practical implementation of the software application was carried out, the analysis of the software development showed that the application works correctly, the resistance to breaking the electronic key is due to the selected JWT technology, the analysis of the FAR and FRR indicators showed an increase in the recognition reliability by 11%. The economic feasibility of the development is confirmed by the conducted analysis.

Keywords: electronic key, entropy of mouse movements, two-factor authentication, access, protection against NSD.

ЗМІСТ

ВСТУП

Актуальність. У зв'язку з широким розповсюдженням електронних ресурсів у всіх сферах людської діяльності гостро стоїть завдання забезпечення інформаційної безпеки в таких системах. Одним з основних заходів щодо захисту даних є забезпечення надійної автентифікації користувача. На даний момент існує багато підходів до автентифікації і ще більше реалізацій цих підходів.

Збільшення використання функції авторизації та автентифікації в різних сервісах, порталах спричинило пожвавлення інтересу до удосконалення методів авторизації та автентифікації.

При цьому не всі класичні рішення задачі автентифікації підходять для використання у всіх веб-додатках. А різні типи систем пред'являють свої унікальні вимоги до підсистем автентифікації. Крім того, активний розвиток обчислювальної техніки дозволяє легко зламувати алгоритми автентифікації, які ще 10-15 років тому вважалися надійними. У зв'язку з цим ведеться безперервна робота в області дослідження і розробки методів автентифікації.

Дослідженнями методів автентифікації займалися такі вчені як: Альфонс Бертільйон, Ерік Гроссе, М. Болл, Дж. Х. Коннел та інші [1 – 5].

Актуальність роботи визначається тим, що проблеми пов’язані з розмежуванням доступу і автентифікацією користувачів є критичними і вдосконалення механізмів вирішення цих задач необхідне для забезпечення коректної роботи сучасних програмних додатків.

Реалізоване у роботі вдосконалення алгоритму автентифікації користувача у захищеному додатку полягає у реалізації двофакторної автентифікації та застосуванні захищеного електронного ключа з використанням алгоритму AES та аналізу ентропії рухів миші, що розраховуються за певними параметрами і є індивідуальною характеристикою для кожного користувача.

Здійснена практична реалізацію програмного додатку, проведений аналіз

програмної розробки показав, що додаток працює коректно, стійкість до зламу електронного ключа зумовлена обраною технологією JWT, аналіз дослідження показників FAR та FRR показав підвищення достовірності розпізнавання на 11%. Економічна доцільність розробки підтверджена проведеним аналізом.

Мета і задачі дослідження. Метою роботи є розробка вдосконаленого методу для підвищення достовірності автентифікації користувача на основі захищеного електронного ключа та аналізу ентропії рухів миші.

Задачами дослідження є:

– дослідити поняття та види автентифікації користувача, здійснити аналіз існуючих методів автентифікації;

– проаналізувати особливості двофакторної автентифікації;

– вдосконалити метод автентифікації користувача на основі використання електронних ключів та ентропії рухів миші;

– розробити алгоритм роботи програми на основі вдосконаленого методу;

– обґрунтувати вибір середовища розробки та мови програмування;

– здійснити програмну реалізацію додатку на основі вдосконаленого методу;

– здійснити тестування розробки та дослідити підвищення достовірності автентифікації користувача;

– економічно обґрунтувати доцільність здійсненої розробки.

Об’єкт дослідження – удосконалений метод багатофакторної автентифікації користувачів.

Предмет дослідження – процес вдосконалення методу для підвищення достовірності автентифікації користувача на основі захищеного електронного ключа та аналізу ентропії рухів миші.

Наукова новизна – вдосконалення автентифікації користувача на основі захищеного електронного ключа та аналізу ентропії рухів миші.

Практична цінність. Розроблено програмний продукт, який реалізує удосконалений метод автентифікації користувача на основі захищеного електронного ключа та аналізу ентропії рухів миші.

1 ЗАГАЛЬНИЙ АНАЛІЗ МЕТОДІВ ТА ПРОЦЕСІВ АВТЕНТИФІКАЦІЇ КОРИСТУВАЧІВ

Підсистема аутентифікації користувачів – один із найважливіших компонентів корпоративної системи інформаційної безпеки, значення якої важко переоцінити. Підсистема автентифікації підтверджує особу користувача інформаційної системи і тому має бути надійною та достовірною, тобто виключати всі помилки у наданні доступу.

У даному розділі розглянемо сучасні методи автентифікації користувачів, суть двофакторної авторизації, а також способи її реалізації, більш детальніше опишемо методи авторизації користувачів на основі електронних ключів та дослідженню поведінкових характеристик, а саме ентропії рухів миші.

1.1 Аналіз сучасних методів автентифікації користувачів

Процес реєстрації користувача в системі складається з трьох взаємопов'язаних процедур, що виконуються послідовно:

– ідентифікації;

– автентифікації;

– авторизації.

Ідентифікація – це процедура розпізнавання суб'єкта щодо його ідентифікатора [6]. У процесі реєстрації суб'єкт пред'являє системі свій ідентифікатор і вона перевіряє його у своїй базі даних. Суб'єкти з відомими системою ідентифікаторами вважаються легальними (законними), інші суб'єкти належать до нелегальних.

Автентифікація – процедура перевірки справжності суб'єкта, що дозволяє достовірно переконатися, що суб'єкт, надавши свій ідентифікатор, є саме тим суб'єктом, ідентифікатор якого він використовує [7]. Для цього він повинен підтвердити факт володіння деякою інформацією, яка може бути доступна лише йому одному (пароль, ключ тощо).

Авторизація – процедура надання суб'єкту певних прав доступу до ресурсів системи після проходження ним процедури автентифікації [8]. Для кожного суб'єкта в системі визначається набір прав, які може використовувати при зверненні до її ресурсів.

Далі розглянемо поширені на сьогодні методи автентифікації користувачів (рис. 1.1)

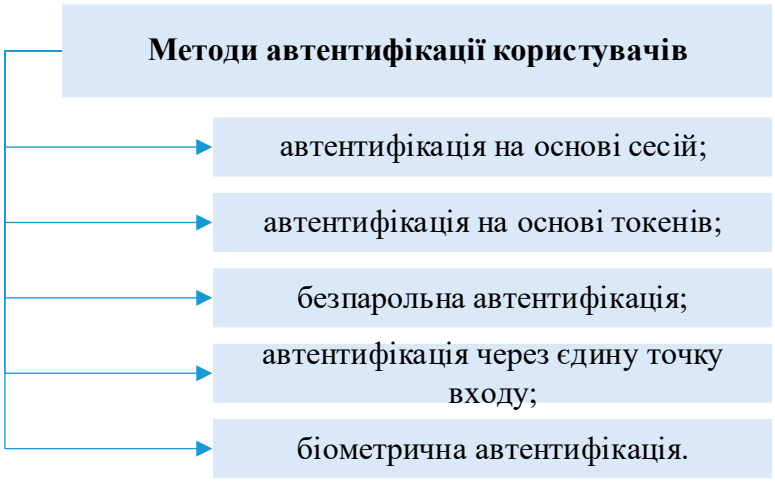


Рисунок 1.1 – Сучасні методи автентифікації користувачів (за [9])

Автентифікація на основі сесій [9]. Автентифікаційний запис або сесія зберігаються на сервері та на клієнті. Сервер повинен відстежувати активні сесії у базі даних чи пам'яті, але в фронтенді створюється кук-файл, у який зберігається ідентифікатор сесії.

Процедура аутентифікації на основі сесій реалізовується наступним чином (рис. 1.2):

– користувач вводить у браузері своє ім'я та пароль, після чого клієнтська програма надсилає на сервер запит;

– сервер перевіряє користувача, аутентифікує його, відсилає додатку унікальний токен (зберігши його в пам'яті або базі даних);

– клієнтська програма зберігає токени в куках і відправляє їх при кожному наступному запиті;

– сервер отримує кожен запит, що вимагає аутентифікації, за допомогою токена автентифікує користувача та повертає запитані дані клієнтської

програми;

– коли користувач виходить, клієнтська програма видаляє його токен, тому всі наступні запити від цього клієнта стають неавтентифікованими.

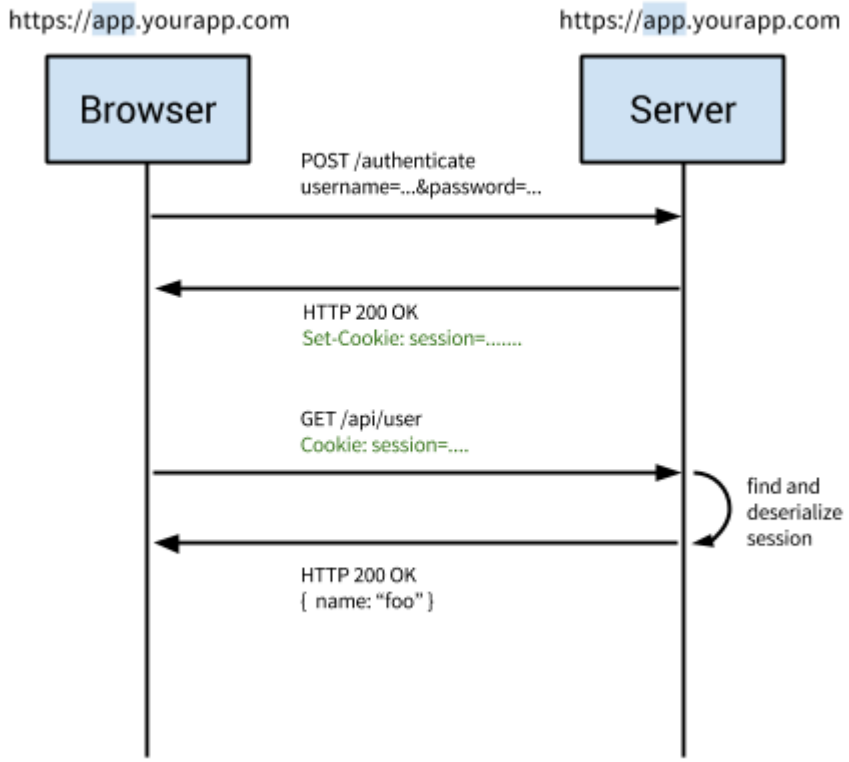


Рисунок 1.2 – Процес автентифікації на основі сесій (за [9])

Автентифікація на основі сесій має певні недоліки, зокрема при кожній автентифікації користувача сервер повинен створювати запис. Зазвичай вона зберігається в пам'яті, і за великої кількості користувачів є ймовірність надто високого навантаження на сервер.

Оскільки сесії зберігаються у пам'яті, масштабувати не так просто. Якщо багаторазово реплікується сервер, то на всі нові сервери доведеться реплікувати і всі сесії користувача. Це ускладнює масштабування.

Наступний тип автентифікації – на основі токенів [9].

Процедура автентифікації на основі токенів опишемо наступним чином (рис. 1.3):

– користувач вводить логін;

– сервер перевіряє їх і повертає токен, який може містити метадані;

– токен зберігається на клієнтській стороні, найчастіше в локальному сховищі, але може лежати і в сховищі сесій або кук-файлах;

– наступні запити до сервера зазвичай містять цей токен як додатковий заголовок авторизації;

– сервер розшифровує токен, якщо токен вірний, сервер обробляє запит.

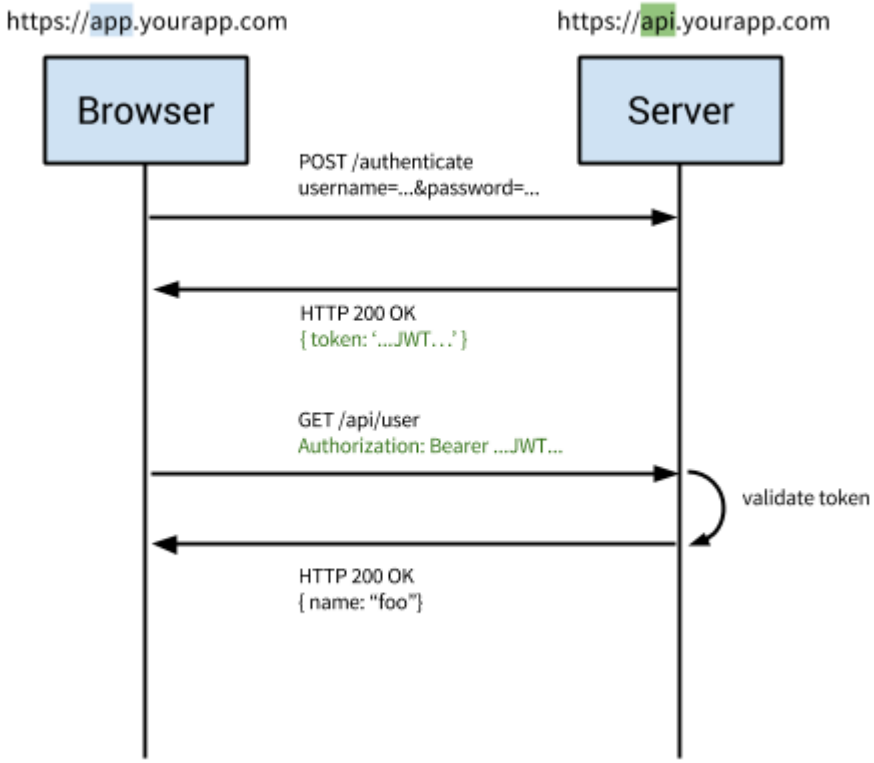


Рисунок 1.3 – Процес автентифікації на основі токенів (за [9])

Перевагою даного методу є та особливість, що серверу не потрібно зберігати записи з токенами або сесіями. Кожен токен самодостатній, містить всі необхідні для перевірки дані, а також передає затребувану інформацію користувача. Тому токени не ускладнюють масштабування. Крім того, у куках зберігається ID сесій, а JWT дозволяє зберігати метадані будь-якого типу, якщо це коректний JSON. Також, при використанні кук-файлів бекенд повинен виконувати пошук за традиційною SQL-базою або NoSQL-альтернативою, і обмін даними, триватиме довше, ніж зчитування токена. Крім того, якщо зберігати всередині JWT додаткові дані на кшталт користувацьких дозволів, то можете заощадити і додаткові звернення пошукові запити на отримання та

обробку даних.

Наступний метод – безпарольна автентифікація. Працює на основі одноразових посилань. Вводиться лише пошта/телефон. Програма відправляє одноразове посилання, користувач по ньому переходить і автоматично входить на сайт/додаток. При безпарольній автентифікації додаток вважає, що до скриньки надійшов лист із посиланням, якщо користувач написав свою, а не чужу адресу.

Проте вагомим недоліком даного методу є те, що якщо хтось отримає доступ до пошти користувача, він отримає і доступ до додатків і сайтів.

Перевагою методу є відсутність необхідності реалізовувати механізм відновлення паролів.

Ще один популярний на сьогодні метод автентифікації – аутентифікація через єдину точку входу – Single Sign On (SSO) [10]. Існують різноманітні реалізації даного методу. Розглянемо його на прикладі Google Accounts. Коли користувач авторизується в одному з Google-сервісів, наприклад Gmail, а потім отримує доступ до інших Google-сервісів без аутентифікації, то використовує єдину точку входу від Google.

Процес автентифікації на Google Accounts (SSO) можна описати наступним чином:

– користувач входить до одного із сервісів Google;

– користувач отримує згенерований в Google Accounts кук-файл;

– користувач переходить до іншого сервісу Google;

– користувач знову перенаправляється в Google Accounts;

– Google Accounts бачить, що користувачеві вже присвоєно кук-файл, і перенаправляє користувача на запитаний продукт.

У даній процедурі використовується три сутності: user; identity provider; service provider.

Користувач вводить пароль у постачальника ідентифікаційної інформації (identity provider, IDP), щоб отримати доступ до постачальника послуги (service provider (SP). Користувач довіряє IDP, і SP довіряє IDP, тому SP може довіряти

користувачеві.

Ще один метод – автентифікація з протоколом OAuth [11], яка використовується при реєстрації/вході до програми через соціальні мережі. Має перевагу у тому, що користувачі можуть увійти у програму одним кліком, якщо у них є обліковий запис в одній із соцмереж. Не потрібно пам'ятати логіни та паролі. Більшість соцмереж як механізм аутентифікації використовують авторизацію через OAuth2.

Наступний, один із найбільш поширених методів, біометрична автентифікація [12], що передбачає перевірку автентичності на основі біометричних показників. При біометричній аутентифікації, секретними даними користувача можуть бути як сітківка ока, так і відбиток пальця.

Сучасна біометрична автентифікація ґрунтується на двох методах:

– статичний метод аутентифікації – розпізнає фізичні параметри людини, якими вона володіє протягом всього життя: від свого народження і до самої смерті (відбитки пальців, відмінні характеристики райдужної оболонки ока, малюнок сітківки очей, термограма, геометрія обличчя, геометрія кисті руки і навіть фрагмент генетичного коду);

– динамічний метод – аналізує характерні риси, особливості поведінки користувача, які демонструються в момент виконання якоїсь звичайної повсякденної дії (підпис, клавіатурний почерк, голос та інше). Детальніше опис даного методу наведемо у таблиці 1.1.

Таблиця 1.1 – Реалізація динамічних біометричних характеристик [13]

Біометрична х-ка	Реєструючий пристрій	Зразок	Досліджувані риси
Голос	Мікрофон, телефон	Запис голосу	Частота, модуляція і тривалість голосового образу
Підпис	Планшет для підпису, перо для введення даних	Зображення підпису і значення відповідних динамічних вимірів	Швидкість, порядок ліній, тиск і зовнішній вигляд підпису

Продовження таблиці 1.1

Динаміка натискання клавіш	Клавіатура	Ритм машинопису	Час затримки (проміжок часу, протягом якого користувач утримує конкретну клавішу), час «польоту» (проміжок часу, який потрібний користувачеві для переходу з однієї клавіші на іншу)
Динаміка роботи з комп. мишкою	Маніпулятор «миша»	Образ характерної траєкторії	Характерні точки траєкторії та інші параметри траєкторії

Оскільки біометричні образи є унікальними для кожного користувача, то забезпечується високий рівень захисту доступу до інформації. Тому біометрична ідентифікація є одним із найзатребуваніших методів розпізнавання.

Проте, в сучасних умовах для підвищення достовірності автентифікації користувачів доцільно застосовувати двофакторну автентифікацію. Більш детальніше розглянемо її у наступному підрозділі.

1.2 Аналіз особливостей двофакторної авторизації користувачів та програмних аналогів на її основі

Системи двофакторної аутентифікації є найбільш поширеними та використовують два різних фактори при аутентифікації користувачів. Вона використовується в основному в мережевих сервісах, однак і для операційних систем є не менш актуальною.

Проблема витоку інформації актуальна в усьому світі та застосування двофакторної автентифікації для захисту інформації стане додатковим бар'єром для зловмисників.

Методи двофакторної автентифікації розглядаються як механізми посилення стійкості автентифікаторів. Двофакторний захист досить надійний

бар'єр, що серйозно ускладнює доступ до чужих даних і в якійсь мірі нівелює недоліки класичного парольного захисту [14].

Для реалізації посиленої аутентифікації (в т.ч. та багатофакторної аутентифікації) найзручніше використовувати засоби трьох груп (рис. 1.4):

– апаратно-програмні модулі довіреного завантаження;

– програмні комплекси з метою захисту від НСД;

– програмні комплекси, розраховані лише на реалізацію двофакторної аутентифікації.

Однак важливо не кількість факторів і не типи пред'явлених ознак, а якість реалізації механізму на обох сторонах взаємодії – як в частині користувача, так і в частині, що знаходиться у сторони, що перевіряє.

Рисунок 1.4 – Схема реалізації двофакторної авторизації (за [15])

Алгоритм двофакторної автентифікації передбачає, що користувач повинен надати два з трьох:

– те, що користувач знає (пароль або PIN-код);

– те, що користувач має (смартфон, usb-ключ, програмний ключ);

– унікальна характеристика користувача (біометрична характеристика).

До переваг двофакторної автентифікації можна віднести її здатність захистити інформацію як від внутрішніх загроз, так і від зовнішніх вторгнень. Певним недоліком можна вважати необхідність використання додаткових програмно-апаратних комплексів, пристроїв зберігання та зчитування даних. В той же час, зараз статистика зламів систем, які застосовують двофакторну автентифікацію, відсутня або вкрай низька.

Крім того, аналіз сучасних інформаційних систем підтверджує необхідність використання двофакторної аутентифікації, як додаткового бар'єру захисту доступності, цілісності та конфіденційності збережених та оброблюваних даних в інформаційній системі.

Основними характеристиками додатків, що слугують для автентифікації користувачів можна вважати [16]:

– незалежність – передбачає існування каналу зв'язку, яким може бути інтернет або оператор стільникового зв'язку;

– потреба синхронізації – ця якість передбачає потреба декодування із сервером аутентифікації;

– автоматичне відновлення – дозволяє визначити доступність поновлення до сервісу автентифікаційного приладу, не вдаючись до допомоги IT фахівців;

– захист програми паролем – відображає, чи реалізований механізм парольного захисту для самого додатка;

– введення даних з клавіатури – наявність цієї властивості говорить про те, що сервіс вимагає введення будь-яких даних за допомогою клавіатурного введення;

– застосування до інших систем – цей пункт говорить про те, що одна програма може використовуватися для автентифікації на різних ресурсах;

– необхідність реєстрації – якщо наявна дана характеристика, то додаток відповідно передбачає попередню реєстрацію у ньому;

– передача даних сервісу – цей параметр свідчить, що сервіс зберігає

секретну інформацію користувача, що використовується для його автентифікації;

– підтримка на операційних системах (Android, iOS, Windows, macOS, Chrome).

Проаналізуємо обрані аналоги додатків, що використовують двофакторну автентифікацію за вище наведеними характеристиками (табл. 1.2).

Таблиця 1.2 – Аналіз додатків двофакторної аутентифікації [17 – 21]

Характерис-тика	Google Authentic ator	Duo Mobile	Microsoft Authenticator	Free OTP	Authy
Незалежність	+	+	+	+	+
Потреба синхронізації	+	+	+	+	+
Автоматичне відновлення	–	+	+	+	–
Захист програми паролем	–	–	–	+	–
Введення даних з клавіатури	+	+	+	+	+
Застосування до інших систем	–	–	+	–	–
Необхідність реєстрації	+	+	+	+	+
Передача даних сервісу	+	+	+	+	+
Підтримка усіх ОС	+/–	+/–	+/–	+/–	+

Розглянувши наведені аналоги додатків із двофакторною автентифікацією, можна зробити висновок, що головним недоліком використання готових рішень є використання одного набору генерації для всіх підключень, а також можливість контролю та доступом до інформації фірмою розробником. Тому розробка програмного додатку з модифікованими засобами реалізації двофакторної автентифікації є актуальною в даний час.

В роботі пропонуються такі методи – автентифікатори як перевірка входу за допомогою біометричних характеристик та електронних ключів, що є більш захищеними, практичними та зручними для програмної реалізації інформаційної системи двофакторної аутентифікації. Використання цих методів дозволить посилити захист інформації, що зберігається в інформаційній системі та підвищити достовірність автентифікації користувача.

1.3 Авторизація користувачів на основі електронних ключів

Аутентифікація на основі електронних ключів полегшує процес аутентифікації для вже відомих користувачів. Для початку роботи користувач надсилає запит до сервера, вказавши користувацькі дані. Потім сервер підтверджує їх на підставі значень, зареєстрованих у базі даних ідентифікаційної інформації. Якщо ідентифікаційні дані підтверджені, сервер повертає ключ аутентифікації (який також зберігається у базі даних) [22].

Коли той самий користувач надалі надсилає запити на доступ до захищених ресурсів, ці запити можуть бути авторизовані за допомогою електронного ключа автентифікації замість пароля. Сервер звіряє ключ із зареєстрованим у базі даних ключем та надає доступ. Аутентифікацію можна реалізувати на основі різних типів токенів, наприклад, OAuth та JSON Web Tokens (JWT) [23].

Наприклад, JWT використовує безпечний спосіб, заснований на підписаних токена, що дозволяє з легкістю виявляти модифікації. Апаратні токени можуть містити ідентифікаційні дані або генерувати одноразовий пароль.

Існує безліч способів надання користувачам токенів аутентифікації – апаратні токени, одноразові паролі (які зазвичай передаються через мобільний телефон) і програмні токени, які зазвичай базуються на стандарті JWT.

Усі електронні ключі безпечно зберігають ідентифікаційну інформацію та дані користувача. Ключ також може підтвердити, що дані вірні та їх не модифікували – важлива вимога безпеки з урахуванням багатьох сучасних

законів про конфіденційність.

Також вони значно підвищують зручність роботи користувача, оскільки дозволяють користувачам виконувати вхід без необхідності запам'ятовування паролів.

Аутентифікація на основі токенів зазвичай складається із чотирьох етапів (рис. 1.5).

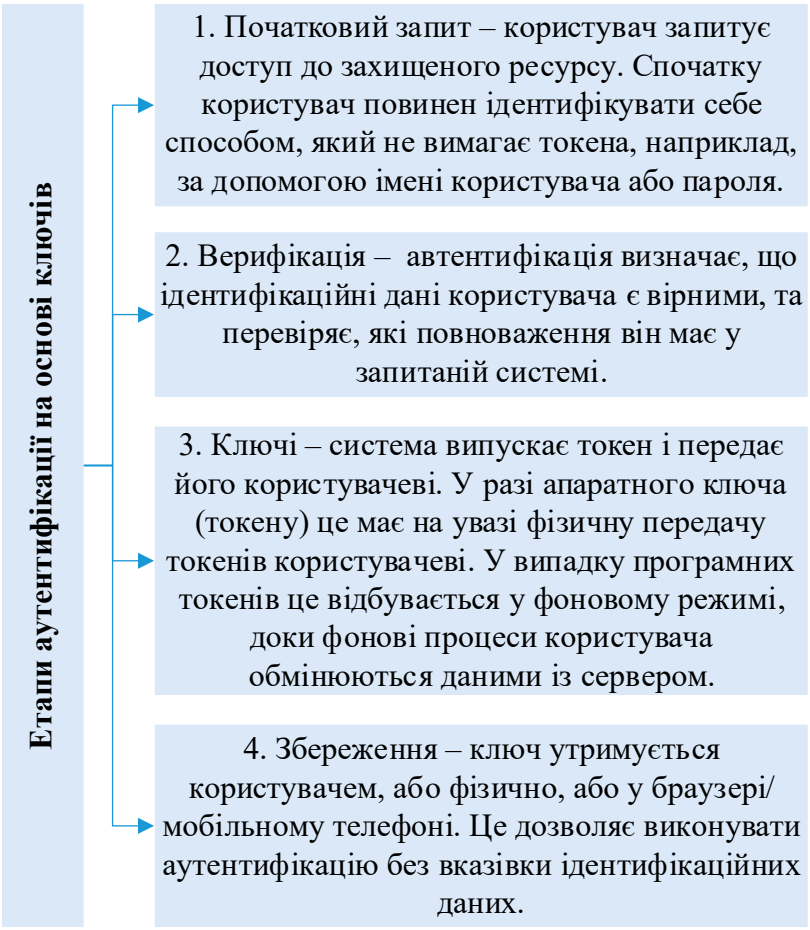


Рисунок 1.5 – Етапи аутентифікації на основі електронних ключів (за [24])

Далі розглянемо основні типи токенів аутентифікації, що використовуються розробниками для автентифікації користувачів або облікових записів сервісів.

Апаратні токени (USB) [25]. Апаратні токени – це фізичні пристрої, які забезпечують авторизацію користувачів для доступу до захищених мереж. Також іноді їх називають токенами автентифікації чи безпеки.

Завдання апаратного токена – забезпечення додаткового шару захисту завдяки двофакторної або багатофакторної автентифікації (2FA або MFA).

Власник токена прив'язує токен до системи чи сервісу, доступ якого йому необхідний.

Апаратні токени спроектовані з урахуванням зручності для користувачів та можливості налаштування, тому вони можуть мати різні формати. Найпоширенішими типами токенів є брелоки, USB та бездротові токени. Апаратні токени можна поділити на три категорії:

– безконтактні – такий токен не вимагає введення коду або з'єднання з пристроєм. Цей тип токена використовує бездротове підключення для доступу до системи, яке може надавати або відхиляти доступ на основі ідентифікаційних даних, пов'язаних зі з'єднанням;

– без підключення – токени без підключення не потрібно фізично вставляти до системи, до якої виконується доступ. Пристрій генерує одноразові коди доступу, що використовуються як частина 2FA або MFA;

– з підключенням – щоб забезпечити доступ токен з підключенням повинен фізично підключатися до системи. Токен сканується зчитуючим пристроєм, що отримує ідентифікаційні дані аутентифікації. Це може бути USB-токен або брелок.

JSON Web Tokens (JWT) (JWT) – це відкритий стандарт (RFC 7519). Він визначає простий автономний спосіб захищеної передачі між сторонами. Стандарт JWT використовує об'єкти JavaScript Object Notation (JSON) для передачі токенів між сторонами. Ці токени можуть використовуватися для аутентифікації, а також для передачі додаткової інформації про користувача або обліковий запис.

Завдяки своєму малому розміру, JWT можуть передаватися як URL, параметри POST або заголовки HTTP і доставлятися швидко. JWT містить всю необхідну інформацію про сутність, щоб уникнути багаторазових запитів до бази даних. Одержувачу JWT не потрібно викликати сервер, щоб перевірити токен.

JWT складається з трьох частин:

– заголовка, що містить тип токена і алгоритм шифрування, що використовується;

– корисне навантаження, що надає ідентифікаційні дані аутентифікації та іншу інформацію про користувачів або обліковий запис;

– підпис, що містить криптографічний ключ, який можна використовувати для підтвердження істинності інформації у корисному навантаженні.

Проаналізуємо безпечність автентифікації на основі токенів. Кіберзлочини стають дедалі витонченішими, тому постачальники сервісів з віддаленим управлінням повинні безперервно оновлювати методики та політики безпеки. Останнім часом зросла кількість атак, націлених на ідентифікаційні дані за допомогою таких способів, як фішинг, брутфорс та атаки за словником. Це означає, що автентифікація більше не може використовувати лише паролі [26].

Аутентифікація на основі токенів у поєднанні з додатковими техніками аутентифікації може створити складніший бар'єр, щоб завадити розумним хакерам використовувати вкрадені паролі. Токени можна отримувати тільки з унікального пристрою, який їх створив (наприклад, смартфона або брелока), завдяки чому вони сьогодні стають високоефективною методикою авторизації.

Хоча платформи токенів автентифікації зробили великий прогрес, загроза частково зберігається. Токени, що зберігаються в мобільних пристроях, легко використовувати, але вони можуть виявитися доступними через вразливість пристрою. Якщо токени надсилаються текстовим повідомленням, їх можна легко перехопити під час передачі. Якщо пристрій викрадено або втрачено, зловмисник може отримати доступ до токенів, що зберігаються на ньому.

Однак завжди потрібно пам'ятати про те, що ніколи не варто покладатися на один спосіб автентифікації. Аутентифікація токенами повинна вважатися лише одним із компонентів стратегії двофакторної або багатофакторної аутентифікації.

Перевагами використання токенів є [27]:

– ефективність – програмні токени ефективні та масштабовані. Сервер може легко створювати та перевіряти будь-яку потрібну кількість токенів, що спрощує збільшення кількості користувачів, які отримують доступ до веб-сайту або додатку;

– гнучкість – програмні токени можна використовувати на декількох серверах, і вони можуть одночасно забезпечувати автентифікацію для кількох веб-сайтів та програм. Вони зазвичай використовуються для реалізації single sign on (SSO), що зручно для користувачів та підвищує безпеку;

– безпека – токени на основі стандартів типу JWT не зберігають стану і можуть перевірятися тільки коли приватний ключ отримано серверним додатком, який використовувався для його генерації. Тому вони вважаються надійним та безпечним способом аутентифікації.

Таким чином, проаналізувавши можливості та переваги застосування токенів, в даній роботі буде опрацьовано алгоритм розробки електронного ключа, суть роботи якого буде заснована на технологіях виконання програмних токенів.

1.4 Авторизація на основі аналізу ентропії рухів миші

Хоча користувачу здається, що курсор миші переміщається плавно, для комп'ютера рух миші є послідовним набором дискретних даних. Одна одиниця даних містить положення курсору, стан кнопок миші і поточний час події. Під подією миші розуміється як натискання кнопки, і рух самої миші.

Таким чином, одним із факторів авторизації може бути зчитування поведінкових характеристик користувача та подальший аналіз ентропії. Опишемо більш детальніше особливості застосовуваного методу.

Особливості біометричних даних користувача.

Біометрія на основі даних миші та клавіатури розглядається у науковій спільноті з 2007 року [28]. Перші дослідження, згідно з опублікованою дисертацією на здобуття ступеня кандидата наук Діденка Сергія Михайловича,

було розпочато ще 2003 року [29].

Різні автори використовують різні набори ключових характеристик, проте загальна методологія збору та аналізу схожа.

На основі аналізу наукових праць [30 – 35] з цієї тематики є можливим виділення наступних ключових етапів: збір та передача сирих даних; зберігання сирих даних; виділення ключових характеристик даних; навчання нейронної мережі; зберігання стану нейронної мережі; класифікація (визначення) користувачів за допомогою навченої нейронної мережі.

Є варіанти щодо обробки сирих даних. Одні системи на комп'ютері користувача збирають дані та відправляють їх на сервер для аналізу, тоді як інші обчислюють ключові характеристики безпосередньо на комп'ютері користувача та відправляють на сервер вже оброблені дані.

Вже 2007 року системи визначення користувача з урахуванням біометричних даних, отриманих з допомогою миші давали точність 95%. Пізніше вченим вдалося збільшити точність до 99,7% (три помилкові спрацьовування на 1000 порцій даних) [36].

Характеристики персональних даних, що збираються. В основному вчені ставлять собі завдання ідентифікації користувача за отриманими біометричними даними. Однак ідентифікація – не єдина можлива сфера застосування цього підходу.

У ряді робіт є заяви про отримання таких типів інформації: ідентифікація користувача; визначення настрою; визначення емоційної реакції на матеріал, що вивчається; визначення психофізичних характеристик (наприклад, втома); визначення фізичних характеристик (зростання, вага).

Даний список може бути розширений за допомогою аналізу факторів організму та особистості, що впливають на аналізовану біометрію.

Особливість відстежування біометричних поведінкових характеристик користувача полягає у тому, що користувач переміщає курсор миші по складній кривій і зі швидкістю, що змінюється в процесі переміщення.

Як сама крива руху курсора, і швидкість руху обумовлені низкою

фізіологічних і психологічних чинників.

Рух миші залежить не тільки від таких факторів як розмір і маса руки, положення руки та всього тіла, але й від стану нервової системи та звичок користувача.

Рух курсору миші можна порівняти з почерком: як текст папері складається з ліній, і з рухів миші складається цифровий почерк. Кожен, хто використовує мишу, залишає підпис таким почерком у пам'яті комп'ютера [37].

Аналіз ентропії рухів миші.

Біометричні системи дозволяють ідентифікувати людину на основі фізіологічних або поведінкових характеристик. Особливості біометричної інформації можуть бути розраховані за допомогою відносної ентропії.

Ентропія [38] може інтерпретуватися як міра невизначеності (невпорядкованості) чи складності деякої системи, наприклад, будь-якого досвіду (випробування), який може мати різні результати, отже, і кількість інформації. Таким чином, іншою інтерпретацією ентропії є інформаційна ємність системи [39].

Враховуючи поставлену мету роботи, а саме підвищення достовірності автентифікації користувача на основі захищеного електронного ключа та аналізу ентропії рухів миші опишемо більш детальніше процес обрахунку ентропії. На рисунку 1.6 наведемо алгоритм для обрахування біометричних даних на основі використання біометричної ентропії [40].

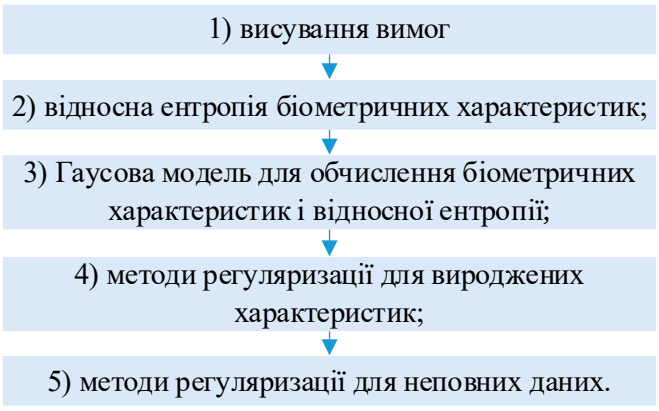


Рисунок 1.6 – Алгоритм для обрахування відносної ентропії за біометричними характеристиками (за [40])

Застосовуючи ентропію для обрахування біометричних даних важливо, аби при цьому дотримувались основні вимоги (рис. 1.7).

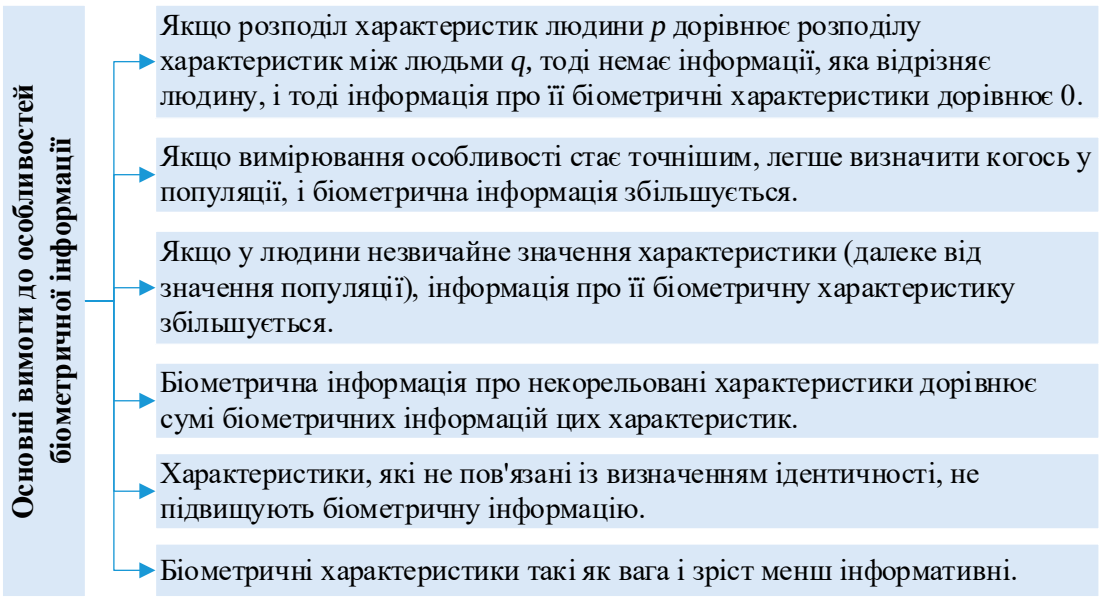


Рисунок 1.7 – Основні вимоги до біометричних даних (за [40])

Використовуючи ентропію ($D(p||q)$) як міру біометричної інформації, вважаймо $p(x)$ і $q(x)$ розподілами біометричних даних користувача. $D(p||q)$ – це відстань Кульбака-Лейблера, що визначається як додаткові біти інформації, що потрібні для представлення $p(x)$ відносно $q(x)$ [40]. Дану відстань можна рахувати за формулою:

$$D(p||q) = \int_x p(x)\, log_2 \frac{p(x)}{q(x)} dx$$

В біометричній системі необхідні характеристики S фіксуються для створення вектора біометричних характеристик $x(S \cdot 1)$ для кожного користувача [40].

Для користувача p формується N_p зразків характеристик, а для населення – N_q зразків. Визначивши x в якості випадкової величини X, відбувається обчислення середнього значення населення μ_q. Обрахунок даного значення суспільства здійснюється за формулою [40]:

$$\mu_q = E_q\,[X] = \frac{1}{N_q} \sum_{i=1}^{N_q} x_i$$

Обрахунок середнього значення для користувача здійснюється аналогічним чином, замінюючи q на p.

Матриця коваріації населення Σ_q обчислюється за наступною формулою:

$$\sum\nolimits_q = \frac{1}{N_q - 1}\sum_{i=1}^{N_q}(x_i - \mu_q)^T\big((x_i - \mu_q\big).$$

Матриця коваріації для користувача обчислюється аналогічним чином.

Проте, суттєвою проблемою під час здійснення прямих вимірювань інформації є придатність даних. Важко оцінити розподіл точно, особливо на кінцях, а тому для невеликих значень $p(x)$ та $q(x), log_2\,(p(x)\,/q(x))$ буде мати абсолютне значення [40]. Доречним рішенням у даній ситуації є застосування моделі з невеликою кількістю параметрів, зокрема, найбільш доцільною моделлю є Гаусовий розподіл.

Використання обраної моделі дозволяється для обчислення верхньої оцінки ентропії. Застосовуючи Гаусову модель та відповідні значення p і q, здійснимо обчислення біометричних характеристик окремого користувача та населення за наступними формулами [40]:

$$p(x) = \frac{1}{\sqrt{|2\pi\Sigma_p|}}\exp\,(-\frac{1}{2}(x - \mu_p)^T\sum\nolimits_p^{-1}(x - \mu_p))$$

$$q(x) = \frac{1}{\sqrt{|2\pi\Sigma_p|}}\exp\,(-\frac{1}{2}(x - \mu_q)^T\sum\nolimits_q^{-1}(x - \mu_q))$$

Враховуючи наведені вище формули, визначимо значення ентропії $D(p||q)$ наступним чином:

$$D(p||q) = \int p(x)\big(log_2 p(x) - log_2 q(x)\big)dx$$

$$= -k(\ln\big|2\pi\Sigma_p\big| - \ln\big|2\pi\Sigma_q\big| + 1$$

$$- E_p[\Big(\mathrm{x} - \mu_q\Big)^T\sum\nolimits_q^{-1}(x - \mu_q)] \; = k(ln\frac{\big|2\pi\Sigma_q\big|}{\big|2\pi\Sigma_p\big|} + trace((E_p + T)\sum\nolimits_q^{-1} - I))$$

При цьому:

$$T = \big(\mu_p - \mu_q\big)^t(\mu_p - \mu_{q),}k = log_2\sqrt{e}.$$

Таким чином, даний вираз обраховує ентропію для Гаусового розподілу $p(x)$ і $q(x)$, а також відповідає більшості поставлених вимог до особливостей біометричних даних, що були описані вище.

1.5 Висновки до Розділу 1 та постановка задачі

Отже, в даному розділі було проведено теоретичний огляд галузі, в якій проводиться розробка. Досліджено сучасні методи автентифікації користувачів, здійснено аналіз особливостей двофакторної автентифікації, описано алгоритми автентифікації користувачів з використанням електронного ключа та поведінкових характеристик.

В ході роботи досліджено, що багатофакторна автентифікація є найбільш ефективним методом захисту від несанкціонованого доступу, оскільки використання кількох незалежних факторів значно зменшує ймовірність, що вони будуть використані одночасно.

В результаті проведеного аналізу теоретичного матеріалу, були поставлені наступні задачі подальшої роботи:

– вдосконалити метод автентифікації користувача на основі використання електронних ключів та ентропії рухів миші;

– розробити алгоритм роботи програми на основі вдосконаленого методу та здійснити програмну реалізацію додатку та здійснити тестування розробки та дослідити підвищення достовірності автентифікації користувача;

– економічно обґрунтувати доцільність здійсненої розробки.

Надійність обраного рішення складається із надійності його елементів. Використання двофакторної системи на основі обраних методів значно збільшує її ефективність. В результаті виконання поставлених задач, можливо досягти основної мети роботи, а саме – підвищення достовірності автентифікації користувача на основі захищеного електронного ключа та аналізу ентропії рухів миші.

2 ПІДВИЩЕННЯ ДОСТОВІРНОСТІ АВТЕНТИФІКАЦІЇ КОРИСТУВАЧІВ НА ОСНОВІ ЕЛЕКТРОННОГО КЛЮЧА ТА АНАЛІЗУ ЕНТРОПІЇ РУХІВ МИШІ

В даному розділі здійснимо практичну розробку алгоритмів спрямованих на підвищення достовірності автентифікації користувачів на основі електронного ключа та аналізу ентропії рухів миші.

Для підвищення достовірності ідентифікації користувача застосуємо двофакторну авторизацію, що базується на основі технології формування ключа JWT та унікальних поведінкових характеристик людини.

2.1 Удосконалення алгоритму автентифікації користувача

Застосування програмного електронного ключа для поставленої задачі зумовлено такими його перевагами як зниження витрат, підвищення зручності користування, зниження ймовірності втрати чи крадіжки та менший ризик атак через посередника.

Застосування аналізу ентропії рухів миші дозволить здійснювати автентифікацію користувача на основі біометричних поведінкових характеристик.

Удосконалення алгоритму автентифікації користувача у захищеному додатку полягає у реалізації двофакторної автентифікації та застосуванні захищеного електронного ключа з використанням алгоритму AES [41] та аналізу ентропії рухів миші, що розраховуються за певними параметрами і є індивідуальною характеристикою для кожного користувача.

До розроблюваного програмного додатку на основі двофакторної автентифікації висуваються наступні вимоги:

– підтримка функції авторизації та реєстрації користувачів;

– налаштування розмежування доступу на основі рольової моделі;

– реалізація розділів введення даних при авторизації користувача;

– можливість формування електронного ключа за заданим алгоритмом;

– можливість здійснення аналізу ентропії рухів миші за заданим алгоритмом;

– аналіз отриманих даних автентифікації, перевірка та прийняття рішення про надання або відмову в доступі;

– обмеження кількості спроб некоректного входу;

– можливість поновлення електронного ключа при його втраті або блокуванні.

Детальний опис застосовуваних алгоритмів наведемо у підрозділах 2.2 та 2.3. Розглянемо загальний алгоритм авторизації користувача у захищеному додатку.

Крок 1. Запуск виконуваного файлу додатку.

Крок 2. Здійснення процесу автентифікації користувача.

Крок 2.1 Якщо користувач вже має обліковий запис в системі – перехід до кроку ?.

Крок 2.2 Якщо користувачеві необхідно отримати доступ до системи – перехід до кроку 3.

Крок 3. Реєстрація користувача.

Крок 3.1 Заповнення користувачем полів інформацією про особисті дані.

Крок 3.2 Здійснення процесу аналізу рухів миші.

Крок 3.3 Підтвердження процесу реєстрації.

Крок 4. Перевірка заповнених форм користувачем.

Крок 4.1 У випадку, якщо всі дані введені вірно, зразок рухів миші сформовано – запит на реєстрацію підтверджується.

Крок 4.2 У випадку, якщо форма реєстрації заповнена невірно (вказаний логін існує, не всі поля заповнені, аналіз рухів миші не здійснено) користувачеві відкривається відповідне сповіщення, а форма реєстрації потребує повторного заповнення.

Крок 5. Підтвердження реєстрації користувача з боку адміністратора.

Крок 5.1 Адміністратор перевіряє нового користувача за його обліковими

даними.

Крок 5.2 Якщо такому користувачеві необхідно надати доступ до системи – ставить відповідну відмітку у певних полях ролей для розмежування доступу.

Крок 5.3 Адміністратор формує електронний ключ, що надсилається користувачеві.

Крок 5.4 Збереження внесених змін.

Крок 6. Здійснення процесу авторизації користувача.

Крок 6.1 Заповнення користувачем поля для введення логіну.

Крок 6.2 Завантаження електронного ключа.

Крок 6.3 Здійснення процесу аналізу рухів миші

Крок 6.4 Підтвердження процесу авторизації.

Крок 7. Перевірка надання доступу до системи користувачеві.

Крок 7.1 У випадку, якщо логін та ключ користувача коректні, зразок ентропії рухів миші відповідає зареєстрованому – користувач отримує доступ.

Крок 7.2 У випадку, якщо один із факторів не відповідає вимогам – користувачеві відкривається сповіщення про невдалу спробу реєстрації.

Кількість спроб авторизації обмежена трьома. У випадку, якщо користувач тричі здійснюватиме некоректний вхід – доступ до облікового запису буде заблоковано, електронний ключ буде недійсним.

Для поновлення доступу користувачеві слід звернутись до адміністратора з електронної пошти вказаної при реєстрації та отримати новий електронний ключ на основі refresh-токену.

Крок 8. Після успішної авторизації користувачеві надається доступ до системи із правами, що були надані адміністратором системи.

Тобто, забезпечивши два фактори авторизації, які залежать виключно від кожного користувача (оскільки і ключ, і ентропія рухів миші індивідуальні для кожного) маємо змогу підвищити достовірність автентифікації користувачів при здійснення функції реєстрації та авторизації в системі.

Загальний алгоритм роботи користувача з додатком на основі обраних методів наведено на рис. 2.1.

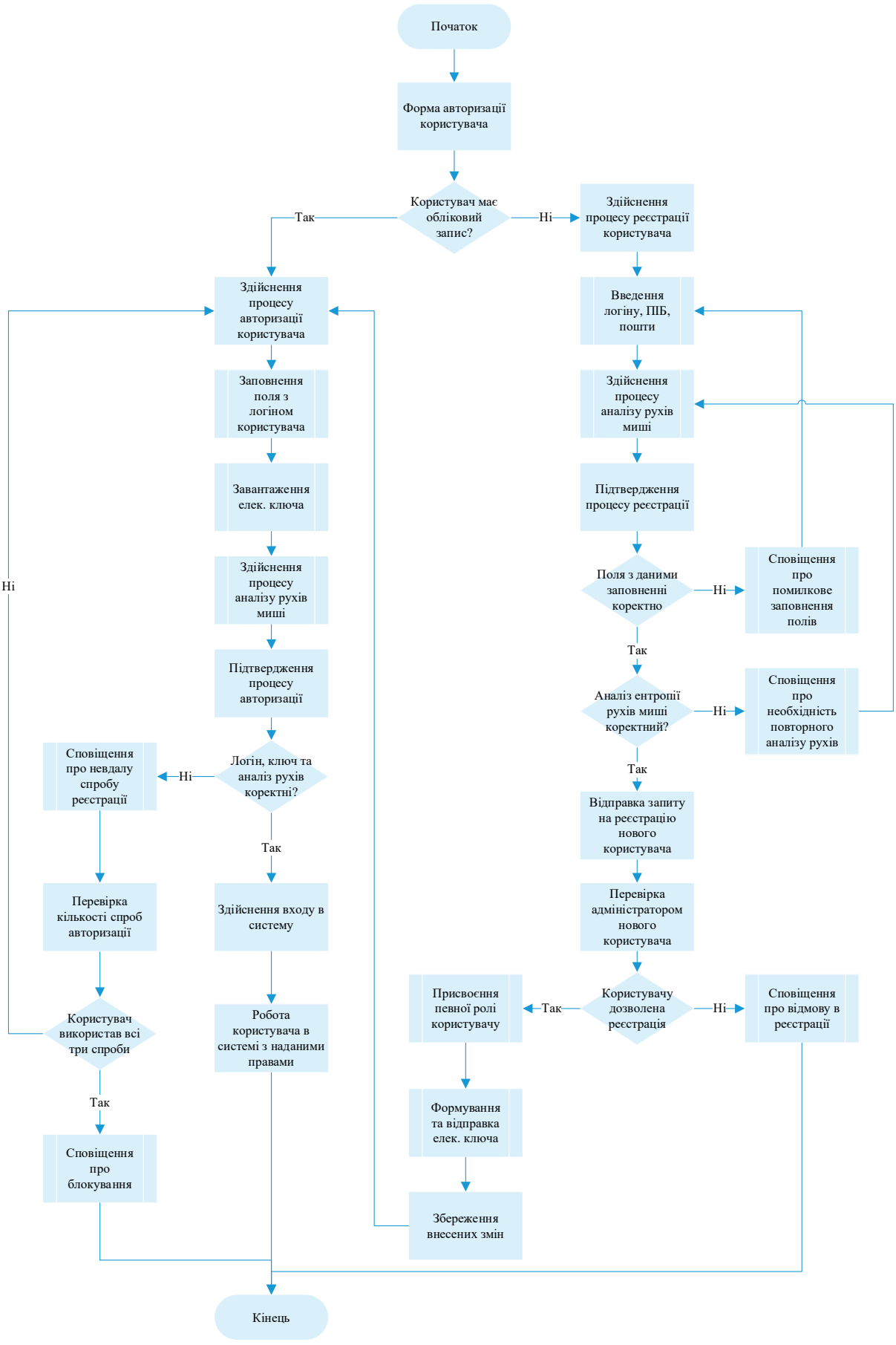


Рисунок 2.1 – Алгоритм роботи з додатком

Таким чином, на основі запланованого вдосконалення методу автентифікації, описаних вимог до розроблюваного додатку та алгоритму його роботи, далі більш детальніше опишемо застосовувані методи та їх математичну суть.

2.2 Алгоритм формування захищеного електронного ключа

Першим етапом автентифікації користувача у захищеному додатку є застосування електронного ключа. В даній роботі розробка електронного ключа здійснюється з використанням технології JWT (JSON Web Tokens), яка дозволяє уникнути помилок авторизації, збільшити продуктивність та збільшити масштабованість додатку.

Структурно JWT складається із трьох частин:

– header – заголовок;

– payload – корисне навантаження;

– signature – підпис.

Опишемо дані структури більш детальніше.

Заголовок – це службова частина токена, яка призначена для визначення типу обробки отриманого токену.

Ця частина є JSON-об'єктом і має наступний формат:

```
{
"тип": "JWT",
"alg": "HS512"
}
```

Структура заголовку містить поля:

– typ – тип токена, наприклад JWT;

– alg – алгоритм, використаний для генерації підпису.

Значення поля typ часто ігнорується програмами, проте стандарт не рекомендує відмовлятися від нього для забезпечення зворотної сумісності.

Поле alg є обов'язковим для заповнення.

В даній роботі для реалізації технології застосуємо алгоритм HS512, що також допустимий для використання. Для генерації та перевірки підпису

використовується єдиний секретний ключ.

Розділ «корисне навантаження» слугує для передачі будь-якої інформації, що допомагає додатку так чи інакше ідентифікувати користувача.

В роботі для розділу «корисного навантаження» застосуємо дані користувача, а саме його прізвище, ім’я та по батькові, а також роль в системі.

Варто зауважити, що в даному розділі можуть міститись і додаткові поля, проте вони не є обов’язковими. Застосування лише основних полів дозволяє кожного разу не звертатись до бази даних для отримання даних про користувача, відповідно, разом з цим, пришвидшує процес його авторизації системою..

В такому випадку, в даній роботі розділ «корисне навантаження» буде містити наступний JSON-об'єкт:

```
{
"id": "001",
"username": "test_name",
"iat": 4155486500,
"role": "user"
}
```

Структура містить наступні поля:

– id – унікальний ідентифікатор користувача;

– username – ПІБ користувача;

– iat – службове поле, час генерації токена у форматі Unix time;

– role – роль користувача, наприклад admin, user.

Пароль – третя структурна частина. Генерується із використанням алгоритму HS512.

Усі три структурні частини токену розділені між собою крапкою. Після того, як стрічка токену сформована, відбувається її шифрування за алгоритмом AES та передача користувачеві, що пройшов етап реєстрації.

В межах роботи механізм формування токену вбудовуваний в основний програмний додаток, проте за необхідності, дану реалізацію можна представити окремим модулем. Загальна схема алгоритму формування електронного ключа за технологією JWT наведена на рис. 2.2.

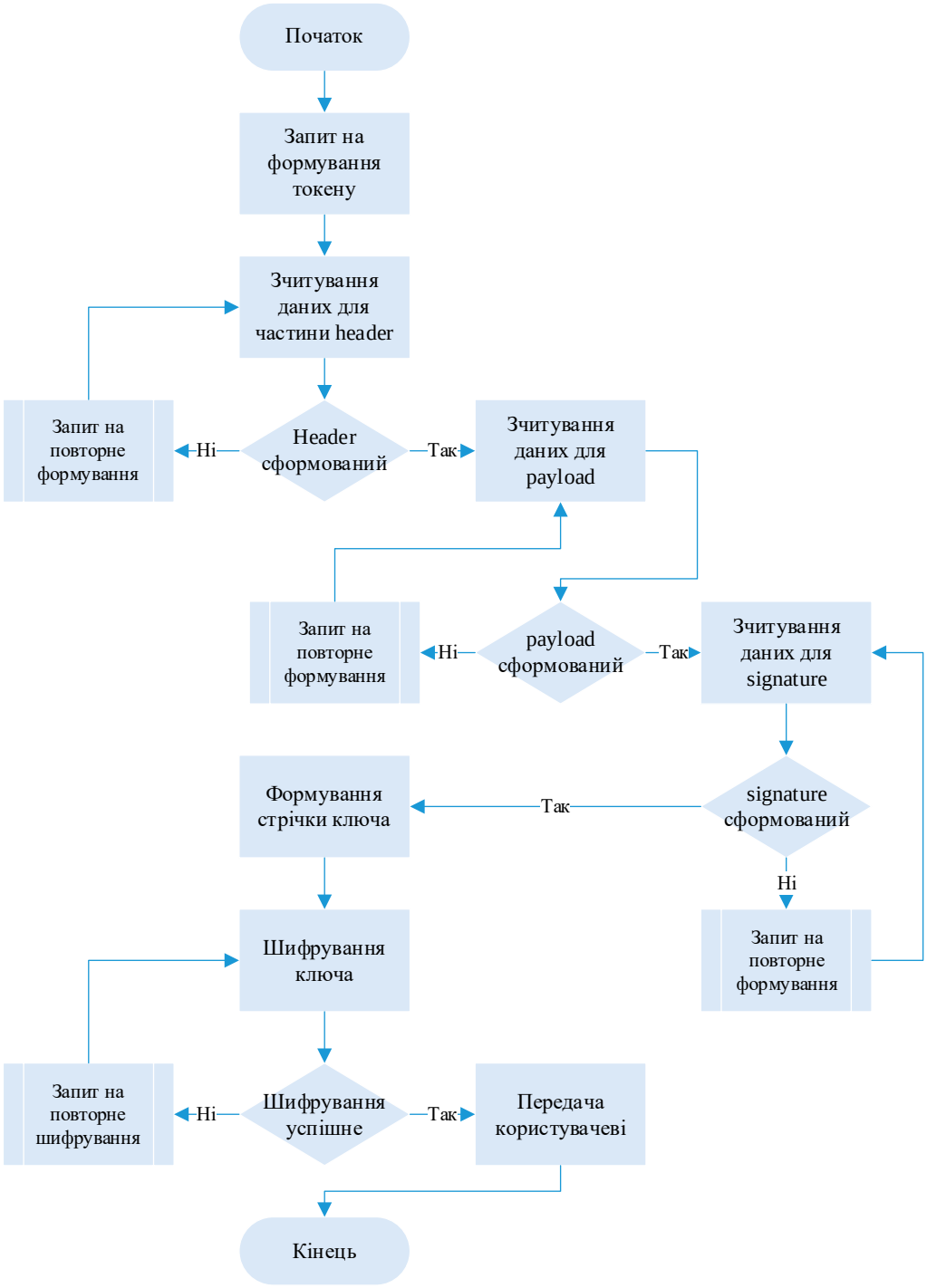


Рисунок 2.2 – Алгоритм формування електронного ключа

У випадку втрати токена користувачем, йому слід звернутись до адміністратора. Для відновлення токену адміністратор повинен застосувати refresh token [42], що формується з довільних даних при створенні основного токену авторизації. Refresh token є унікальним для кожного користувача та прив’язаний лише до одного конкретного токену авторизації.

При втраті токену авторизації, на основі refresh token адміністратор

формує новий токен авторизації для користувача.

Таким чином, використання електронного ключа на основі технології JWT можемо описати наступним чином:

Крок 1. Застосування даних користувача для запиту на формування токену.

Крок 2. Перевірка інформації користувача.

Крок 3. Формування електронного ключа та його шифрування.

Крок 4. Надання ключа користувачеві.

Крок 5 Користувач зберігає токен та прикріплює значення токена до кожного запиту.

Крок 6. Надання доступу користувачеві після перевірки ключа системою.

Крок 7. Можливість відновити доступ на основі електронного ключа з використанням refresh-токену.

Таким чином, запропонований алгоритм дозволяє не зберігати інформацію про всі видані токени, як при класичній схемі. Коли користувач звертається до програми, він передає йому свій токен. Додатку необхідно лише перевірити підпис і витягти необхідні поля з розділу «корисного навантаження».

У випадку реалізації окремого модуля сервісу аутентифікації за даним алгоритмом стає можливим створити єдину точку входу в різні сервіси з однаковими обліковими даними. Пройшовши процедуру аутентифікації, користувач зможе отримати доступ зі своїм токеном до тих ресурсів, які вважають довіреним даний сервіс аутентифікації.

Додатково створений refresh token дозволяє відновлювати втрачений токен авторизації користувача на основі його унікальних даних.

Додаткове шифрування токену при передачі користувачеві підвищує рівень захисту переданої послідовності та не викриває структуру даних, що розміщені у файлі. Для шифрування даних токену застосуємо алгоритм AES – симетричний ітеративний алгоритм шифрування [43, 44, 45].

2.3 Алгоритм процесу аналізу ентропії рухів миші

В межах даної роботи для визначення ентропії [46 – 47] рухів комп'ютерної миші досліджуватись будуть наступні показники:

– траєкторія руху комп'ютерної миші;

– відхилення руху комп'ютерної миші від зафіксованої траєкторії;

– швидкість руху комп'ютерної миші;

– прискорення руху комп'ютерної миші;

– нетипові рухи комп'ютерної миші;

– кліки комп'ютерної миші.

Опишемо більш детальніше кожен із наведених параметрів.

Для дослідження траєкторії руху комп'ютерної миші застосуємо метод відслідковування кривих та розбиття їх на окремі вектори.

Алгоритм дослідження даного показника складається із п'яти кроків.

Крок 1. Спрощення кривої (траєкторії за якою рухалась мишка) та розбиття на окремі вектори на основі використання алгоритму Дугласа-Пекера [48].

Крок 2. Обрахунок довжини векторів.

Для цього застосуємо математичну формулу:

$$b = \frac{d(P_0, P(b))}{d\ (P_0, P_1)} = \frac{|w|cos\theta}{v_1} = \frac{w \cdot v_1}{|v_1|^2} = \frac{w \cdot v_1}{v_1 \cdot v_1};$$

де P_0, P_1 проекція вектора P_0, P на відрізок P_0P_1.

$$v_1 = (P_1 - P_0), w = (P - P_0).$$

Крок 3. Перетворення векторів в косинус кута нахилу відносно осі x та y:

$$< L = (cos\ \alpha, cos\ \beta)$$

Дане перетворення дозволить отримати набір даних в діапазоні [–1, 1], який доречно використати для обробки. Якщо вектор перетворити таким чином, то він не буде залежати від масштабу.

Крок 4. Розпізнавання образу за допомогою багатошарового персептрона. Для цього застосуємо математичну формулу:

$$f_n = \frac{\left|\frac{g_1}{a_1}\right| + \left|\frac{g_2}{a_2}\right| + |\dots| + \left|\frac{g_m}{a_m}\right|}{m}$$

де m – число параметрів образу; g – параметри оброблюваного образу; a – параметри еталонних образів.

Варто зазначити, що в даній операцій кількість входів в багатошаровий персептрон дорівнює кількості образів в програмі. Серед отриманих значень знаходиться деяке максимальне число і відповідний до нього номер виходу. У випадку, якщо дане число перевищує поріг розпізнавання, то задана траєкторія визначається як образ, що відповідає номеру виходу.

Крок 5. Виконання дій, асоційованих з розпізнаним образом.

Якість розпізнавання за даним алгоритмом залежить від ступеню відмінності використовуваних зразків, тобто чим менша кількість схожих зразків в програмі, тим вище якість розпізнавання.

Для дослідження відхилення руху комп'ютерної миші від зафіксованої траєкторії застосуємо формулу для розрахунку евклідової відстані.

При розрахунку даної відстані застосуємо координати x_1 та y_1, що являються координатами курсора та враховуються для дослідження горизонтальних та вертикальних відхилень у момент часу.

Координатами x_2 та y_2 позначимо координати цільової точки.

Обрахунок даної відстані здійснимо за формулою:

$$length\left((\mathrm{x}_1, \mathrm{y}_1), (\mathrm{x}_2, \mathrm{y}_2)\right) = \sqrt{(\mathrm{x}_1 - \mathrm{x}_2)^2 + (\mathrm{y}_1 - b\mathrm{y}_2)^2},$$

Пропорційна близькість обчислимо за формулою:

$$proximity = 1 - length / max\,(length)$$

За отриманими показниками далі надається можливість відстежувати відстань миші до від еталонної точки до отриманої в кожному коректному випадку.

Відносна різниця між довжиною пройденої курсором комп'ютерної миші траєкторії та довжиною лінії, що поєднує початкову та кінцеву точки переміщення курсору миші позначається δD та рахується за формулою:

$$\delta D = \frac{D - D_{min}}{D_{min}}$$

Показники горизонтальної, вертикальної та загальної швидкості переміщення курсору обчислимо за формулою:

$$V_{X_i} = \frac{\delta x_i}{\delta t_i},\ \ V_{Y_i} = \frac{\delta y_i}{\delta t_i},\ \ V_i = \sqrt{V_{X_i}^2 + V_{Y_i}^2},\ \ i = 2 \dots n,$$

де $\delta t_i = t_i - t_{i-1},\ \ t_1 = 0,\ \ V_{X_1} = 0,\ \ V_{Y_1} = 0,\ \ V_1 = 0.$

Показники прискорення, ривку та кутової швидкості обчислимо за формулами:

$$a_i = \frac{\delta V_i}{\delta t_i},\ \ j_i = \frac{\delta a_i}{\delta t_i},\ \ \omega_i = \frac{\delta \Theta_i}{\delta t_i},\ \ i - 2 \dots n,$$

де $a_1 = 0,\ \ j_1 = 0,\ \ \omega_1 = 0.$

$$a(t) = \left(\frac{a_x(t)}{\sqrt{1 + \left(\frac{a_y(t)}{a_x(t)}\right)^2}} + \frac{a_y(t)}{\sqrt{1 + \left(\frac{a_x(t)}{a_y(t)}\right)^2}} \right)$$

де $a(t) -$ загальне прискорення переміщення курсору.

Окрім того, для представлення даних використаємо додаткові параметри:

$$r = \sum_{i=1}^{n} \sqrt{\delta x_i^2 + \delta y_i^2}$$

де $r -$ довжина траєкторії;

$$L = \sqrt{(x_n - x_1)^2 + (y_n - y_1)^2}$$

де $L -$ довжина відрізка між двома кінцевими точками траєкторії;

$$s = \frac{L}{r}$$

де $s -$ прямолінійність траєкторії;

$$T = t_n - t_1$$

де $T -$ час виконання дії.

Далі опишемо оцінку нетипових рухів комп’ютерної миші для визначення

складності траєкторії. Даний аналіз можна здійснювати на основі отриманих параметрів траєкторій.

У випадку наявності усіх необхідних для обрахунку параметрів x та y за певний час, можна порахувати $x-$ та $y-$ зрушення між кожним кроком за нормалізований час відповідно:

$$\Delta x = x_{timestep+1} - x_{timestep}$$

$$\Delta y = y_{timestep+1} - y_{timestep}$$

Наступним кроком слід обчислити кількість однакових вікон x-зрушення розміром m і $m+1$ (M_m та M_{m+1}), далі обчислити середню кількість схожих вікон для усіх вікон розміром m і m+1.

Ентропію обчислимо за наступною математичною формулою:

$$\Delta S = -In\frac{M_{m+1}}{M_{m+1}}$$

$$\Delta S = In[M_m] - In[M_{m+1}]$$

Для дослідження кліків комп'ютерної миші користувачем в роботі застосовується зональний клік-тест. При здійсненні певного алгоритму дій фіксується кількість та зона кліків кожним користувачем. Далі за отриманими даними формується карта координат кліків, яка потім порівнюється із отриманими новими даними при авторизації користувача.

Метод дослідження клік-тесту можна описати наступними кроками:

Крок 1. Здійснення користувачем кліків мишкою у певних зонах.

Крок 2. Фіксування параметрів кліків у вигляді карти координат.

Крок 3. Формування звіту характеру кліків для кожного користувача.

Крок 4. Співставлення еталонних значень із отриманими під час кожного випадку авторизації користувача.

Крок 5. Визначення ентропії.

Таким чином, етап авторизації користувача на основі методу дослідження ентропії рухів миші з можливістю отримання точних результатів, дозволяє здійснити програмну реалізацію додатку із підвищеною достовірністю автентифікації користувачів.

2.4 Обґрунтування вибору засобів програмування

Виходячи із поставлених задач роботи для програмної реалізації алгоритму спрямованого на підвищення достовірності автентифікації користувачів на основі електронного ключа та аналізу ентропії рухів миші застосуємо мову об’єктно-орієнтованого програмування C# [49 – 50], середовище програмування Visual Studio [51 – 52] з використанням інтерфейсу Windows Forms [53 – 54].

Опишемо детальніше обрані засоби програмування та визначимо їх переваги для даної роботи.

C# – універсальна мова програмування, що дозволяє реалізувати складні завдання.

Будучи об'єктно-орієнтованою мовою, він багато перейняв у Java та C++. Як і Java, C# спочатку призначався для веб-розробки, і приблизно 75% синтаксичних можливостей такі мови ж, як Java.

C# також вважають «очищеною версією Java». Ще 10% мова запозичила із C++ і 5% – з Visual Basic. 10% C#, що залишилися, – це реалізація власних ідей розробників. Об'єктно-орієнтований підхід дозволяє будувати за допомогою C# великі, але в той же час гнучкі, масштабовані та розширювані додатки (рис. 2.3).

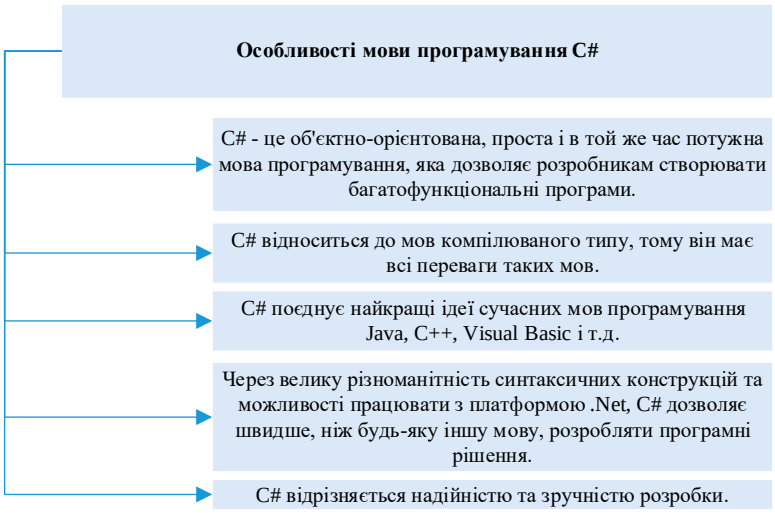


Рисунок 2.3 – Особливості мови програмування C#

Microsoft Visual Studio є середовищем програмування, розробленим компанією Microsoft. Дане середовище дозволяє створювати кросплатформні проекти різними мовами програмування, таких як Visual Basic, Visual C#, Visual C++, Visual F# та інші.

Також середовище надає можливість створювати програми, що використовують у своїй роботі платформу .NET, яка дозволяє використовувати великий набір сервісів, що реалізуються у вигляді проміжного коду, що не залежить від базової архітектури (рис. 2.4).

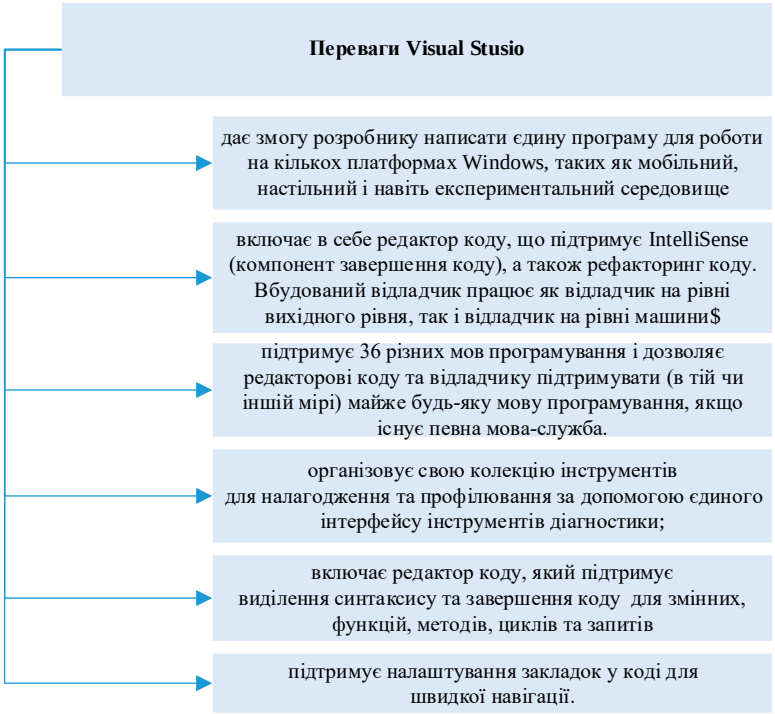


Рисунок 2.4 – Переваги Microsoft Visual Studio

Для створення інтерфейсу (GUI додатку) в Microsoft .NET обрано застосовувати Windows Forms.

Windows Forms – це новий стиль розробки програми на базі класів .NET Framework class library. Дані класи мають власну модель програмування, яка є більш досконалою, порівняно з моделями, що засновані на Win32 API або MFC, та використовуються в керованому середовищі .NET Common Language Runtime (CLR).

У Windows Forms термін "форма" – є синонімом вікна верхнього рівня.

Головне вікно програми – це форма. Всі інші вікна верхнього рівня, що присутні в додатку, – це також форми. Діалогові вікна також вважаються формами.

Незважаючи на назву «форми», програми, що використовують Windows Forms, не виглядають як форми. Подібно до традиційних Windows-додатків програми дозволяють здійснювати повний контроль над подіями у власних вікнах (рис 2.5).

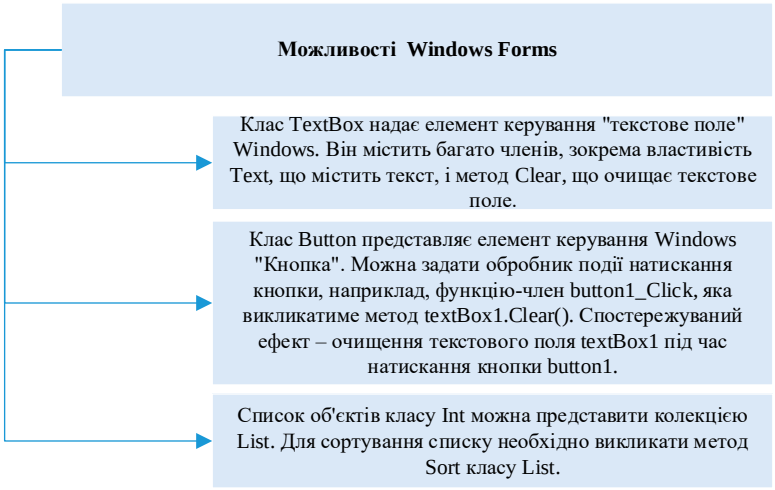


Рисунок 2.5 – Можливості Windows Forms використані в роботі

Таким чином, на основі обраних засобів програмування, таких як мова об’єктно-орієнтованого програмування C#, середовище програмування Visual Studio з використанням інтерфейсу Windows Forms, в наступному розділі здійснимо практичну програмну реалізацію описаного методу для підвищення достовірності автентифікації користувача на основі електронного ключа та аналізу ентропії рухів миші.

2.5 Висновки до Розділу 2

Таким чином, у даному розділі з метою підвищення достовірності автентифікації користувачів на основі електронного ключа та аналізу ентропії рухів миші було здійснено удосконалення методу автентифікації.

Удосконалення алгоритму автентифікації користувача у захищеному додатку полягає у реалізації двофакторної автентифікації та застосуванні захищеного електронного ключа з використанням алгоритму AES та аналізу ентропії рухів миші, що розраховуються за певними параметрами і є індивідуальною характеристикою для кожного користувача.

Реалізація електронного ключа зумовлена такими його перевагами як зниження витрат, підвищення зручності користування, зниження ймовірності втрати чи крадіжки та менший ризик атак через посередника. Аналіз ентропії рухів миші дозволить здійснювати автентифікацію користувача на основі біометричних поведінкових характеристик.

Для практичної реалізації поставленої мети роботи заплановано використати такі засоби програмування: мову об’єктно-орієнтованого програмування C#, середовище програмування Visual Studio з використанням інтерфейсу Windows Forms.

3 ПРАКТИЧНА РЕАЛІЗАЦІЯ ПРОГРАМНОГО ДОДАТКУ НА ОСНОВІ ДВОФАКТОРНОЇ АВТОРИЗАЦІЇ

У даному розділі здійснимо практичну реалізацію програмного додатку на основі алгоритму підвищення достовірності автентифікації користувача на основі захищеного електронного ключа та аналізу ентропії рухів миші.

Для реалізації даної задачі спроектуємо графічний користувацький інтерфейс додатку, опишемо програмну частину, сформуємо інструкцію користувача для роботи з додатком та проаналізуємо реалізований програмний засіб на основі описаного алгоритму автентифікації відповідними засобами тестування.

Програмна реалізація здійснена на мові програмування C# у середовищі Visual Studio з використанням інтерфейсу Windows Forms. Тестування виконано на основі unit-тестів та дослідження показників FAR та FRR.

3.1 Проектування користувацького інтерфейсу

Одним із етапів практично програмної реалізації розроблюваного програмного додатку є проектування графічного інтерфейсу. Хоча розроблюваний додаток призначений для демонстрації захищеного доступу на основі двофакторної автентифікації та не потребує втілення певних маркетингових рішень щодо зовнішнього вигляду, основні особливості побудови інтерфейсу повинні бути дотриманні.

На сьогодні існує багато правил та особливостей щодо розроблюваних додатків, в даній роботі застосуємо деякі з них, а саме:

– доступність – додаток має бути візуально зрозумілий для користувача функціональні можливості доступними, а пошук кнопки авторизації та полів введення даних не повинен викликати труднощів;

– мінімалізм – додатки призначенні для контролю доступу не повинні бути перенавантажені зайвим змістом (рекламою, інформацією фірми, що не є

актуальною для здійснення задачі авторизації і т.д.), діалогові вікна повинні бути лаконічними та відображати лише функціонал додатку;

– ефективність – інтерфейс користувача – це інструмент керування, який надає доступ до різних функцій програми. Правильний інтерфейс повинен давати можливість користувачеві з найменшими зусиллями виконати дію, що його цікавить;

– послідовність – функціонал додатку повинен бути представлений користувачеві послідовно, не слід на початкових сторінках демонструвати недіючі кнопки, що будуть доступні лише для авторизованого користувача чи пропонувати можливість реєстрації вже авторизованому користувачеві;

– розуміння – можливість повернутись до попереднього вікна, переглянути відомості дії користувача, надати коротку інформацію про обліковий запис, тобто розуміння того, що навіть у найпростіших додатках контролю доступу користувачеві може знадобитись переглянути елементарне.

Працюючи над досягненням однієї з цих характеристик, розробник може створити проблеми для досягнення іншої. Наприклад, намагаючись зробити інтерфейс зрозумілішим, можна додати багато описів і пояснень, що зрештою зробить інтерфейс ще більш громіздким і незручним. Або скорочуючи матеріал для досягнення мінімалізму, можна зробити деякі речі незрозумілими для пересічного користувача.

Для дотримання поставлених вимог та уникнення можливих недоліків із реалізацією користувацького інтерфейсу, спроектуємо деякі діалогові вікна розроблюваного додатку.

Одним із функціональних можливостей, що реалізуються – є реєстрація користувача в додатку. Доречно верхній частині вікна відображати назву процесу, що виконує користувач. Для здійснення реєстрації користувачеві слід вводити відповідні дані у поля «Логін», «ПІБ», «Пошта», розмістимо їх у лівій частині сторінки.

Для реалізації одного із факторів авторизації, алгоритмом передбачено аналіз ентропії рухів миші, виділимо окреме поле для демонстрації даного

процесу у правій частині вікна.

В нижній частині вікна розмістимо функціональні кнопки запуску аналізу ентропії рухів миші та підтвердження реєстрації.

Вигляд проектованого вікна наведено на рис. 3.1.

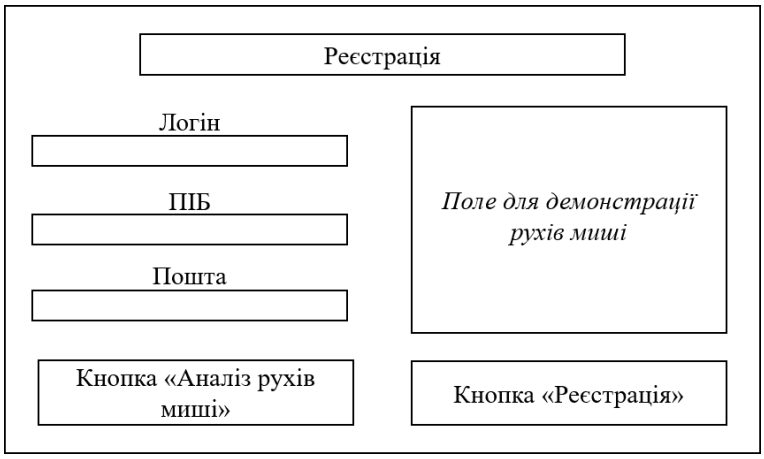


Рисунок 3.1 – Проектування діалогового вікна реєстрації

Вікно авторизації матиме схожу структуру, за відмінності відсутності полів для введення «ПІБ» та «Пошта» та з доданим розділом завантаження електронного ключа.

Наступним розглянемо вікно додатку авторизованого в системі користувача (на прикладі адміністратора).

У верхній частині вікна розмістимо основний функціонал наданий можливостями облікового запису. У випадку адміністратора – це кнопки:

– «Особистий кабінет»;

– «Адміністрування користувачів»;

– «Історія записів»;

– «Вихід»;

– поле для відображення логіна користувача.

Нижче на сторінці, під функціональними кнопками, розміщено поле із назвою сторінки.

Основна частина сторінки – стосується безпосередньо змісту системи, що потребує захисту, тобто у цій частині вже можуть бути розміщені інформація, що є актуальною для користувача з певним рівнем доступу, функціональні

кнопки, що дозволяють здійснювати функції передбачені захищуваною системою і т.д.

Вигляд проектованого вікна наведено на рисунку 3.2.

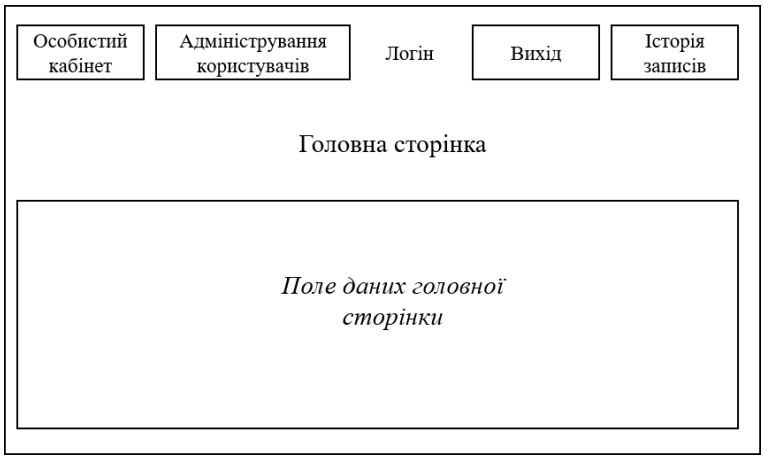


Рисунок 3.2 – Проектування діалогового вікна авторизованого користувача

Аналогічні вікна для інших ролей користувачів матимуть схожий вигляд, за винятком наявності функціональних кнопок у верхній частині вікна та змістом основної частини сторінки (якщо є така вимога від захищуваної системи).

Наступне діалогове вікно – це редагування даних користувача адміністратором системи, оскільки в роботі передбачено, що саме адміністратор підтверджує реєстрацію користувача та надсилає ключ.

Для виконання даних функцій адміністратору слід надати можливість переглядати основі дані користувача (логін, ПІБ, пошта), застосувати чек-бокс – для відмітки про надання ролі користувачеві та функціональні кнопки із можливістю згенерувати ключ, відмінити та зберегти внесені зміни.

Усі згадані поля розмістимо послідовному у діалоговому вікні (рис.3.3).

Для взаємодії додатку з користувачем передбачені спливаючі сповіщення про результат виконання певних дій (успішний чи неуспішний).

Вигляд вікна сповіщення схожий до системного, містить лише інформацію, що слід надати користувачеві та функціональну кнопку для підтвердження ознайомлення із сповіщенням.

Жодних інших даних у згадуваному вікні не повинно бути розміщено.

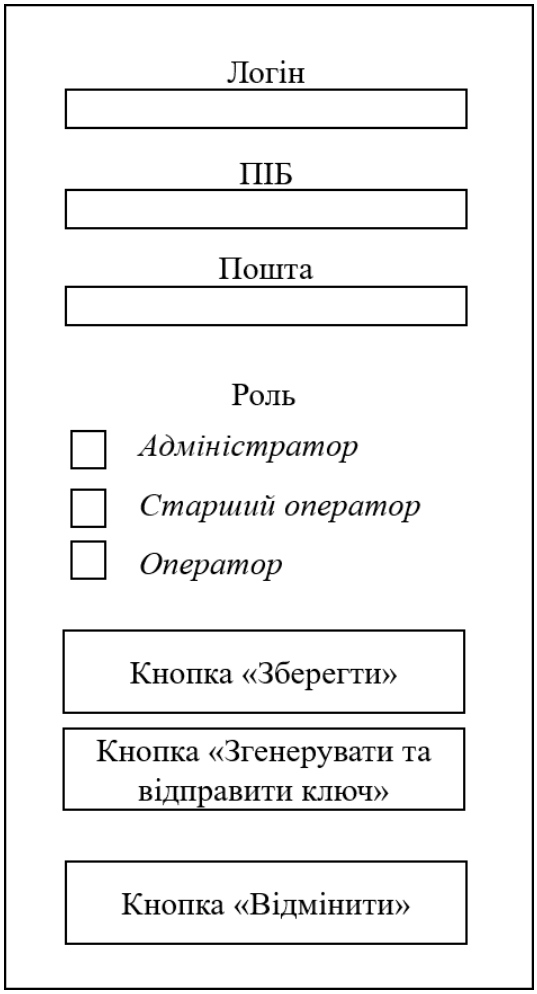


Рисунок 3.3 – Проектування діалогового вікна редагування даних користувача

Таким чином, на основі наведених особливостей розробки графічного користувацького інтерфейсу, далі в роботі здійснимо програмну реалізацію розроблюваного додатку, що матиме зовнішній вигляд, наведений у даному розділі.

3.2 Особливості програмної реалізації додатку

Враховуючи поставлені задачі роботи, було здійснено програмну реалізацію додатку на основі розробленого методу для підвищення достовірності автентифікації користувача на основі захищеного електронного ключа та аналізу ентропії рухів миші.

Розглянемо детальніше особливості даної реалізації на прикладі деяких

частин програмного коду.

Використані бібліотеки:

```
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Text;
using System.Windows;
using System.Windows.Controls;
using FaceRecognition.Models;
using Newtonsoft.Json;
using walletMouseEntropy;
```

Реалізовано клас користувач, що містить змінні (UserId, UserName, IsAdmin, MouseUp, MouseDow, IsRegistered) для надання їм певного значення:

```
public class User
  {
     public string UserId { get; set; }
     public string UserName { get; set; }
     public bool IsAdmin { get; set; }
     public string FIO { get; set; }
     public string Email { get; set; }
     public string Role { get; set; }
     public string Password { get; set; }
     public int? MouseUp { get; set; }
     public int? MouseDown { get; set; }
     public int? MouseLeft { get; set; }
     public int? MouseRight { get; set; }
     public bool IsRegistered { get; set; } = false;
  }
```

Реалізація кнопки Registration_Click, що викликає функцію реєстрації користувача:

```
private void Registration_Click(object sender, RoutedEventArgs e)
     {
        var w = Application.Current.Windows[0];
        this.Hide();
        Registration wfAbout = new Registration();
        wfAbout.ShowDialog();
        this.Close();
     }
```

Реалізація кнопки Login_Click, що викликає функцію авторизації користувача:

```
private void Login_Click(object sender, RoutedEventArgs e)
    {
      this.Hide();
      Login wfAbout = new Login();
      wfAbout.ShowDialog();
      this.Close();
    }
```

Фіксація авторизованого в системі користувача:

```
    private void Exit_Click(object sender, RoutedEventArgs e)
    {
      Registration.Visibility = Login.Visibility = Visibility.Visible;
      Exit.Visibility = Visibility.Hidden;
      Login.Visibility = Visibility.Visible;
      UserTable.Visibility = Visibility.Hidden;
      Helper.AddLog("User is logged out", AuthUserName.Content.ToString());
```

Та опублікування надпису «Не авторизовано», якщо користувач здійснив вихід з системи:

```
      UserInfo.Visibility = Visibility.Hidden;
      history.Visibility = Visibility.Hidden;
      AuthUserName.Content = "Не авторизовано";


    }
```

Створення класу для формування електронного ключа за технологією JWT:

```
public static void AddLog(string eventRequest, string user = null)
    {
      var log = GeLogFile();
      if (log == null)
        log = new List<string>();
      var userResp = "";
      if (!String.IsNullOrEmpty(user))
      {
        userResp = "\n user: " + user;
      }
      var requestString = $"{eventRequest} \n time: {DateTime.UtcNow} {userResp} \n";
      log.Insert(0, requestString);
      string json = JsonConvert.SerializeObject(log);
      File.WriteAllText($"{
Path.GetDirectoryName(System.AppDomain.CurrentDomain.BaseDirectory)}/LogFile.json",  json,
Encoding.UTF8);
    }
```

Зчитування файлу з електронним ключем при здійсненні авторизації

користувача:

```
{
        List<string> users = new List<string>();

        var path = $"{
Path.GetDirectoryName(AppDomain.CurrentDomain.BaseDirectory)}/LogFile.json";
        bool exists = System.IO.File.Exists(path);
        if (!exists)
          System.IO.File.Create(path);
        using (StreamReader r = new StreamReader(path))
        {
          string json = r.ReadToEnd();
          users = JsonConvert.DeserializeObject<List<string>>(json);
        }
        return users;
      }
```

Розробка кодової послідовності для шифрування сформованого електронного ключа:

```
private static byte[] AES_Decrypt(byte[] bytesToBeDecrypted, byte[] passwordBytes)
      {
        byte[] decryptedBytes = null;
        byte[] saltBytes = new byte[] { 2, 1, 7, 3, 6, 4, 8, 5 };
        using (MemoryStream ms = new MemoryStream())
        {
          using (RijndaelManaged AES = new RijndaelManaged())
          {
            AES.KeySize = 256;
            AES.BlockSize = 128;
            var key = new Rfc2898DeriveBytes(passwordBytes, saltBytes, 1000);
            AES.Key = key.GetBytes(AES.KeySize / 8);
            AES.IV = key.GetBytes(AES.BlockSize / 8);
            AES.Mode = CipherMode.CBC;
            using (var cs = new CryptoStream(ms, AES.CreateDecryptor(),
CryptoStreamMode.Write))
            {
              cs.Write(bytesToBeDecrypted, 0, bytesToBeDecrypted.Length);
              cs.Close();
            }
            decryptedBytes = ms.ToArray();
          }
        }
```

Реалізація класу для фіксування рухів миші:

```
private bool MouseEntropyAnalyze(User user)
      {
```

```
            var countDif = 30;
            var resCount = 1;
            if (Up - countDif >= user.MouseUp && user.MouseUp <= Up + countDif)
            {
               resCount += 1;
            }
            if (Down - countDif >= user.MouseDown && user.MouseDown <= Down + countDif)
            {
               resCount += 1;
            }
            if (Left - countDif >= user.MouseLeft && user.MouseLeft <= Left + countDif)
            {
               resCount += 1;
            }
            if (Right - countDif >= user.MouseRight && user.MouseRight <= Right + countDif)
            {
               resCount += 1;
            }
            return resCount >= 3;
         }
      return decryptedBytes;
         }
```

Перевірка заповнення полів при реєстрації та виведення відповідного сповіщення у випадку, якщо поле з логіном не заповнене:

```
      private bool VerifyData()
         {
            if (String.IsNullOrEmpty(UserName.Text))
            {
               MessageBox.Show("Введіть логін");
               return false;
            }
            return true;
         }
```

Перевірка даних аналізу рухів миші:

```
      private void Canvas_MouseDown_1(object sender,
System.Windows.Input.MouseButtonEventArgs e)
         {
            if (e.ButtonState == MouseButtonState.Pressed)
               lastPoint = e.GetPosition(this);
```

Зчитування даних про аналіз рухів миші та аналіз ентропії:

```
            private void Form1_MouseMove(object sender,
System.Windows.Input.MouseEventArgs e)
            {
               var points = e.GetPosition(this);
```

```
        if (e.LeftButton == MouseButtonState.Pressed && drawEnable)
        {
            if (lastPoint != null)
            {
                if (lastPoint.X < points.X)
                {
                    Right += 1;
                }
                if (lastPoint.X > points.X)
                {
                    Left += 1;
                }
                      if (lastPoint.Y > points.Y)
                {
                    Up += 1;
                }
                if (lastPoint.Y < points.Y)
                {
                    Down += 1;
                            }
            }
            Line line = new Line();
            line.Stroke = SystemColors.WindowFrameBrush;
            line.X1 = lastPoint.X;
            line.Y1 = lastPoint.Y;
            line.X2 = e.GetPosition(this).X;
            line.Y2 = e.GetPosition(this).Y;
            lastPoint = e.GetPosition(this);
            myCanvas.Children.Add(line);
        }
    }
```

Фіксування часу для аналізу даних рухів миші:

```
void timer_Tick(object sender, System.EventArgs e)
    {
        drawEnable = false;
        logButton.IsEnabled = true;
        timer.Stop();
        MessageBox.Show("Аналіз завершено та дані збережено!");
        StaerRecog.Content = "Розпочати знову?";
        myCanvas.Children.Clear();
    }
```

Сповіщення про невдалу спробу автентифікації:

```
if (!string.IsNullOrEmpty(eror))
        {
            --failCount;
            MessageBox.Show($"Невдала спроба ідентифікації. {eror}. Залишилося спроб -
```

```
{failCount}");
```

Сповіщення про успішний вхід користувача до системи:

```
enableAutoNavigate = false;
                MessageBox.Show("Спроба входу була вдалою!");

                commonpage.ShowDialog();
                this.Close();
```

Отже, на основі описаних класів було реалізовано основні ключові особливості програмного додатку, описано процес формування та шифрування електронного ключа, процес запису та аналізу рухів комп'ютерної миші, діалогові форми додатку для взаємодії з користувачем.

3.3 Інструкція користувача для роботи з додатком

В даному розділі опишемо інструкцію користувача для роботи з додатком, що реалізований для практичної реалізації підвищення достовірності автентифікації користувача на основі захищеного електронного ключа та аналізу ентропії рухів миші.

Для початку роботи з додатком, користувачеві слід запустити виконуваний файл, після чого відкривається головне вікно (рис. 3.4). В даному вікні розміщені кнопки «Вхід», «Реєстрація» та «Інформація».

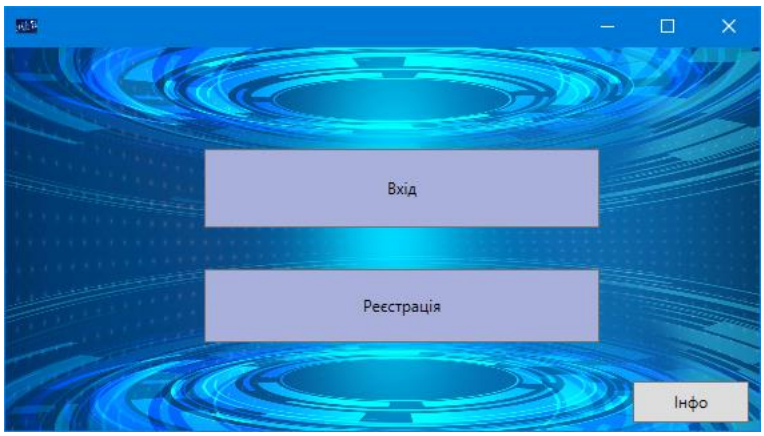


Рисунок 3.4 – Вигляд головного вікна програмного додатку

Натиснувши кнопку «Інформація», відкривається нове вікно.

В даному вікні може бути розміщена будь-яка інформація, що може бути доступною для неавторизованого в системі користувача.

Для прикладу, в даному вікні розмістимо тему роботи (рис. 3.5).

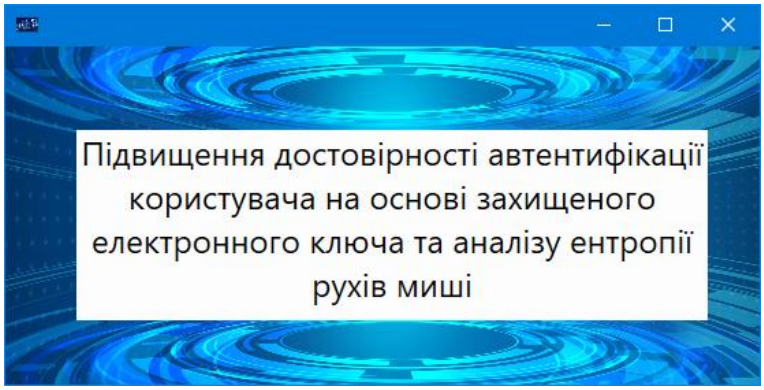


Рисунок 3.5 – Вигляд вікна при натисненні кнопки «Інформація»

Далі більш детальніше розглянемо основні функції додатку.

В додатку застосована рольова модель розмежування прав доступу та передбачено три основні ролі:

– адміністратор;

– старший оператор; оператор.

В залежності від потреб власника програмного додатку кількість ролей можна розширювати та кожній з них надавати окремі поля.

Керуючою роллю для даного додатку є адміністратор, який підтверджує реєстрацію користувача, формує та надсилає електронний ключ.

В головному вікні облікового запису адміністратора (рис. 3.6) розмішені кнопки «Обліковий запис», «Адміністрування користувачів» та «Історія записів» (журнал або лог-файл, що містить дані про звернення до додатку) (рис.3.7).

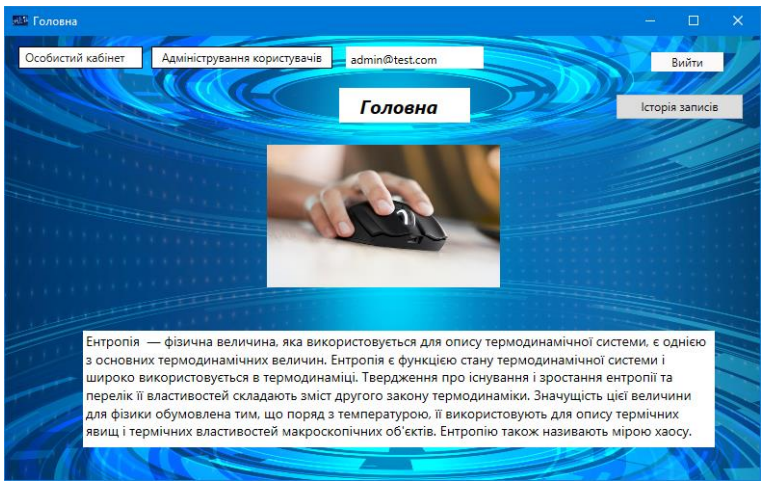


Рисунок 3.6 – Вигляд вікна облікового запису адміністратора

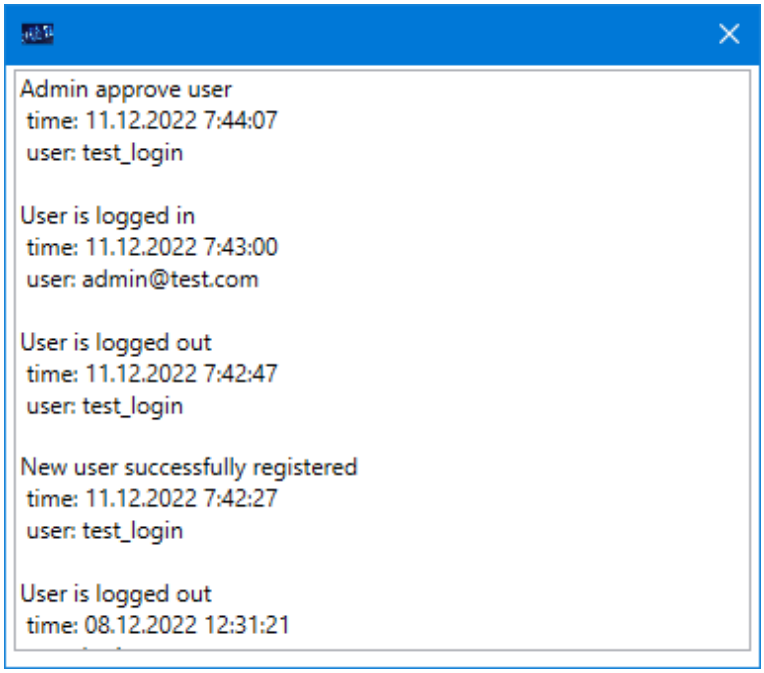


Рисунок 3.7 – Вигляд вікна журналу дій

Зокрема, у вікні «Адміністрування користувачів» адміністратор може переглядати та редагувати права доступу кожного користувача (рис.3.8). Функція підтвердження реєстрації користувача реалізована для того, щоб реєстрація нових користувачів була контрольованою та лише повноважені особи мали змогу отримати певний рівень доступу у захищуваній системі.

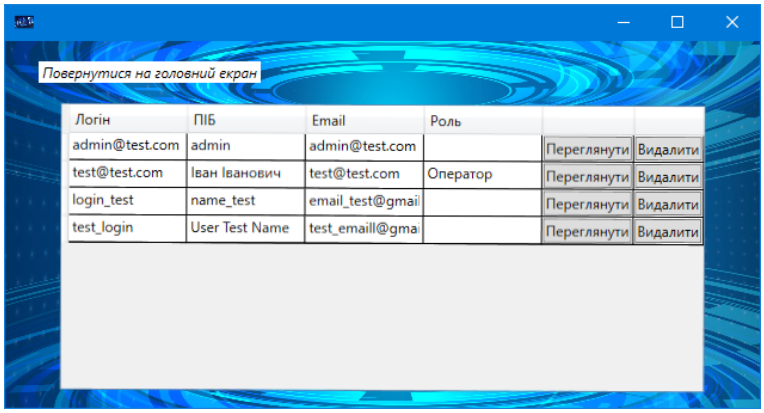


Рисунок 3.8 – Вигляд вікна «Адміністрування користувачів»

Обравши в переліку певного користувача, адміністратор може його видалити в системі, якщо є така необхідність, або ж натиснути кнопку «Переглянути», після чого відкривається нове вікно (рис. 3.9).

У даному вікні адміністратор може переглянути деякі користувацькі дані та надати (або змінити) користувачеві його роль, згенерувати ключ, зберегти внесені зміни.

Рисунок 3.9 – Вигляд вікна редагування даних користувача

При успішній генерації ключа, адміністратор отримує відповідне сповіщення (рис. 3.10). Безпосередньо процесом формування ключа адміністратор не керує.

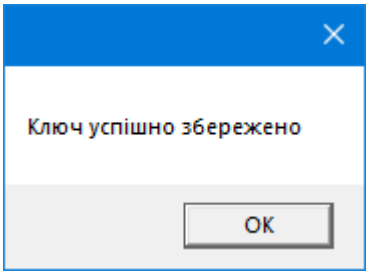


Рисунок 3.10 – Вигляд сповіщення про успішне формування ключа

Наступна функціональна можливість – реєстрація користувача. У випадку, якщо користувачеві необхідно отримати доступ до системи, на головному вікні додатку йому слід натиснути кнопку «Реєстрація» (рис. 3.11).

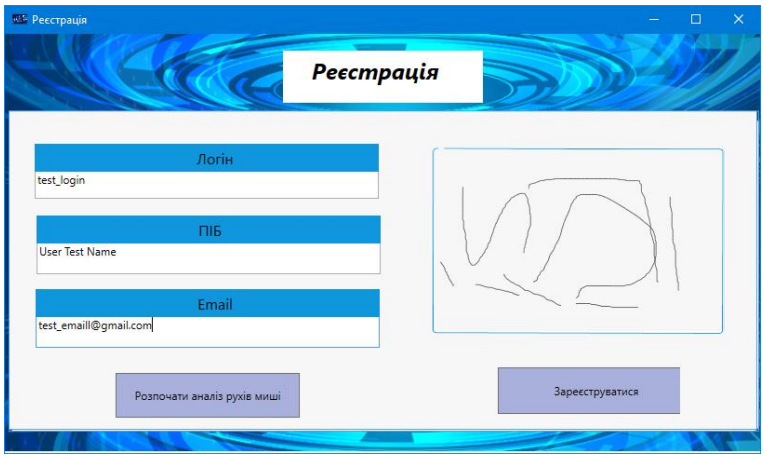


Рисунок 3.11 – Вигляд вікна реєстрації користувача

В даному вікні користувач вводить свій логін, повне ім’я, електронну пошту. Враховуючи, що додаток передбачає двофакторну авторизацію, другим фактором перевірки є аналіз ентропії рухів миші. В роботі дана функція продемонстрована наочно.

Отже, після заповнення необхідних полів, користувачеві слід натиснути кнопку «Розпочати аналіз рухів миші», після цього у вікні справа здійснювати звичайні рухи мишкою. Аналіз відбувається протягом 5 – 7 секунд, після того як аналіз буде здійснено, користувач отримає відповідне сповіщення (рис.3.12).

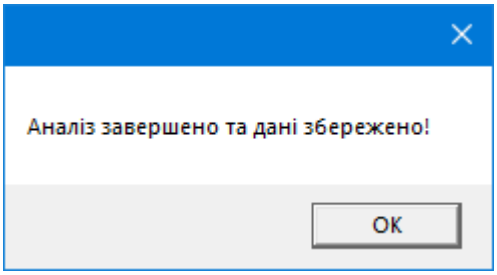


Рисунок 3.12 – Вигляд сповіщення про успішний аналіз рухів миші

У випадку, якщо аналіз не буде здійснено, користувач має можливість скористатись кнопкою повторного аналізу.

Наступним кроком є підтвердження реєстрації користувачем, у випадку коректного виконання усіх дій, користувач отримує відповідне сповіщення (рис. 3.13).

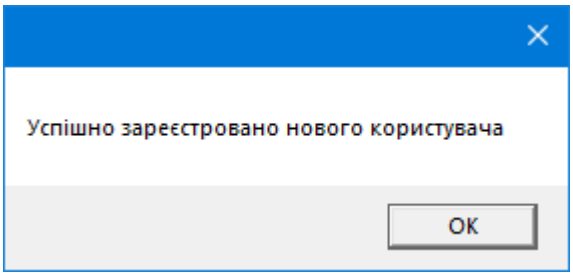


Рисунок 3.13 – Вигляд сповіщення про успішну реєстрацію

Після виконання даних дій. Користувач вже відомий системі, проте немає користувацьких прав та електронного ключа для авторизації. Дані можливості користувач отримує лише після підтвердження реєстрації адміністратором та наданим йому електронним ключем.

Далі розглянемо функціональну можливість – авторизація користувача. Для здійснення даної функції користувачеві необхідно скористатись кнопкою «Вхід» на головній сторінці додатку, після чого відкриється відповідне вікно.

У даному вікні авторизації користувачеві слід ввести лише логін, наданий ключ та здійснити аналіз рухів миші (рис. 3.14).

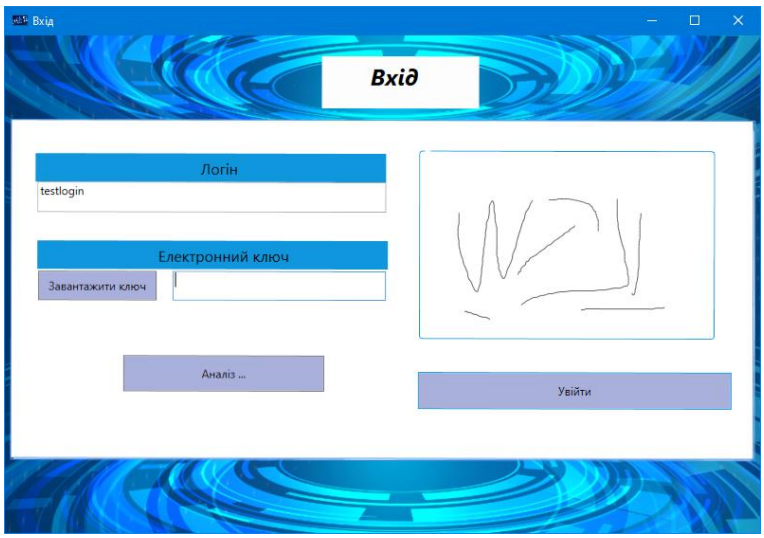


Рисунок 3.14 – Вигляд вікна авторизації користувача

Вважливо зауважити, що користувачеві надається лише три спроби входу (рис. 3.15). Якщо всі вони будуть некоректні – доступ до системи із діючим електронним ключем буде заблоковано. Для поновлення доступу користувачеві слід звернутись до адміністратора з електронної пошти, що була вказана при реєстрації.

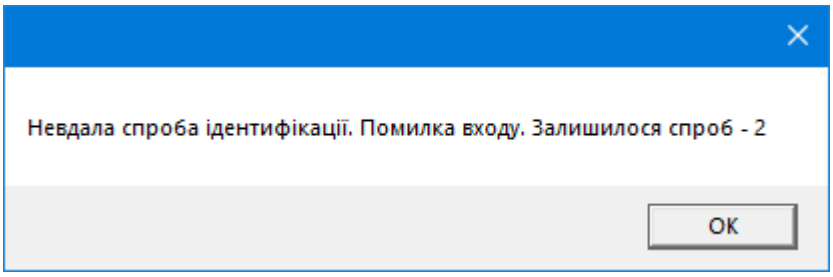


Рисунок 3.15 – Вигляд сповіщення про некоректний вхід та обмеження спроб

У випадку, якщо користувач коректно заповнив усі поля, додаток надає відповідне сповіщення (рис. 3.16) та доступ користувачеві до системи із тими правами, що були надані йому адміністратором (рис.3.17).

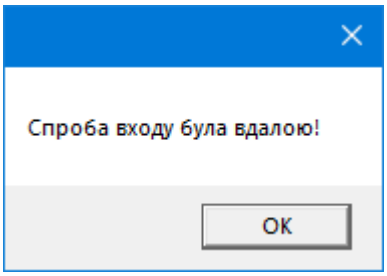


Рисунок 3.16 – Вигляд сповіщення про успішну спробу авторизації

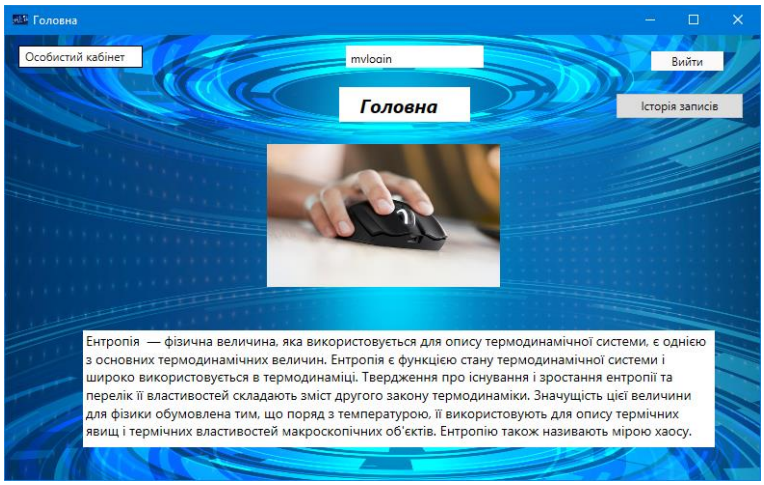


Рисунок 3.17 – Вигляд вікна авторизованого користувача

Авторизований користувач у системі також може зайти в особистий кабінет та переглянути основні відомості свого акаунта (рис. 3.18).

В особистому кабінеті користувачеві можуть бути доступні можливості передбачені системою та відповідальним за нею, а також роллю, що визначена для даного користувача.

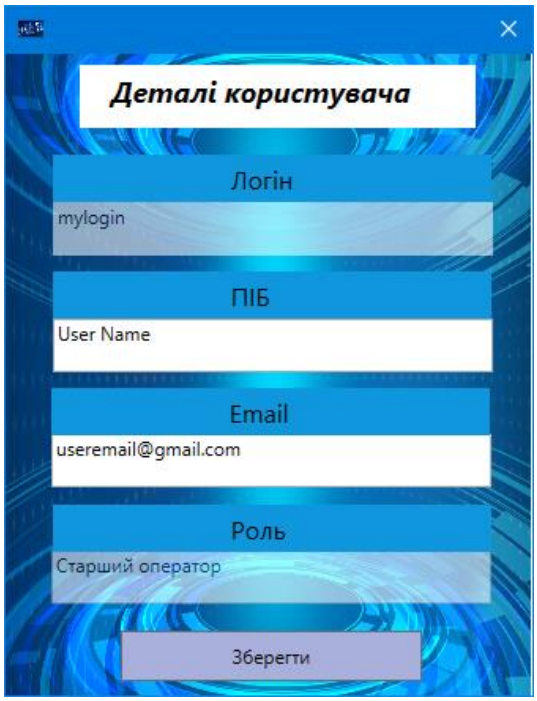


Рисунок 3.18 – Вигляд вікна даних користувача

Таким чином, на прикладі продемонстрованого додатку була представлена реалізація системи, достовірність входу до якої підвищена за рахунок використання двофакторної авторизації на основі електронного ключа та аналізу поведінкових характеристик користувачів.

3.4 Тестування та аналіз результатів розробки

Експериментально перевіримо функціональність розроблюваного додатку.

Зареєструємо нового користувача із обліковими даними:

– логін testlogin;

– ПІБ Test Name;

– пошта test1email@gmail.com

При аналізі рухів миші, здійснюватимемо рухи миші більш швидше, ніж звичайно, заповнюючи усю область для аналізу (рис. 3.19).

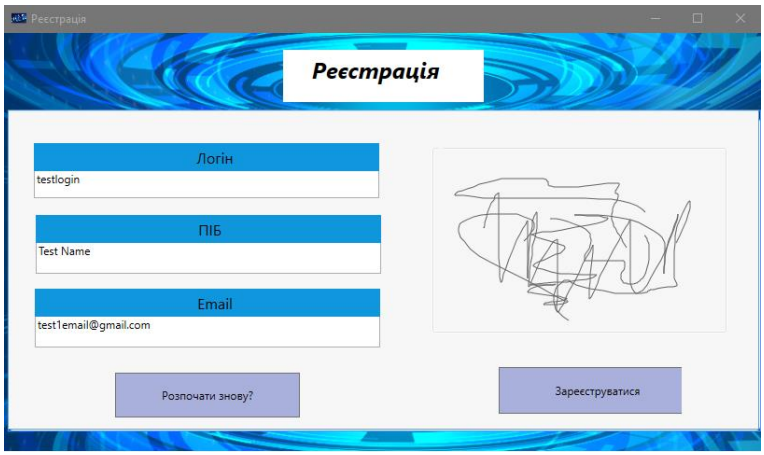


Рисунок 3.19 – Заповнення форми реєстрації користувача

Далі для підтвердження реєстрації користувача зайдемо в обліковий запис адміністратора та в розділі адміністрування користувачів (рис. 3.20) надамо користувачеві певну роль та згенеруємо ключ доступу (рис.3.21).

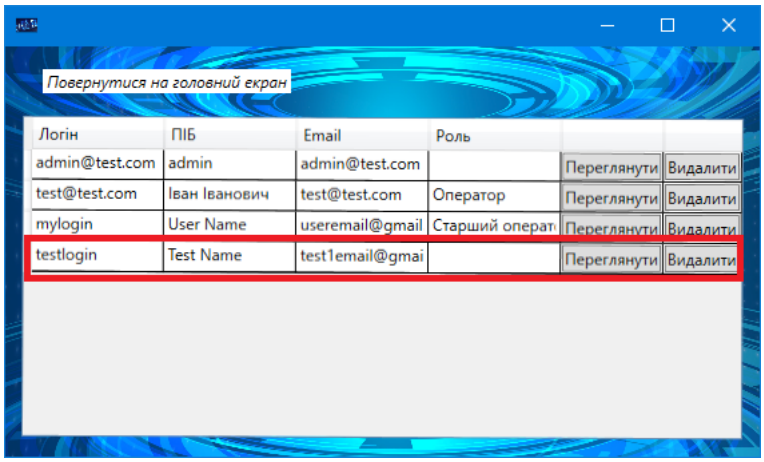


Рисунок 3.20 – Вікно «Адміністрування користувачів» у обліковому записі адміністратора

Після виконання даних дій, в таблиці адміністрування користувачів відповідному користувачеві надається відмічена (відповідні права доступу) та надсилається ключ.

Спробуємо здійснити спробу авторизації зареєстрованого користувача.

У формі авторизації введемо коректні дані, завантажимо коректний ключ. У полі аналізу ентропії рухів миші, рух мишкою здійснюватимемо у звичайному темні, не настільки інтенсивно як при реєстрації (рис. 3.22).

Рисунок 3.21 – Підтвердження реєстрації користувача (надання ролі та генерація ключа)

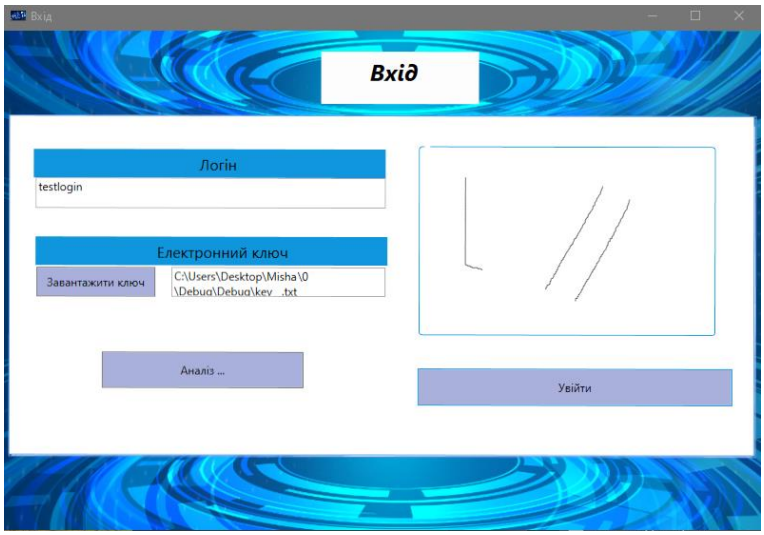


Рисунок 3.22 – Заповнення форми авторизації користувача

Після натиснення кнопки «Увійти», додаток видає відповідне сповіщення, що свідчить про неуспішну спробу входу.

Спробуємо здійснити спробу авторизації ще раз, здійснюючи рухи мишею аналогічно як при реєстрації, проте з іншим ключем доступу. Як

наслідок – невдала спроба входу. Дозволена кількість спроб входу для користувача – 1 (рис. 3.23).

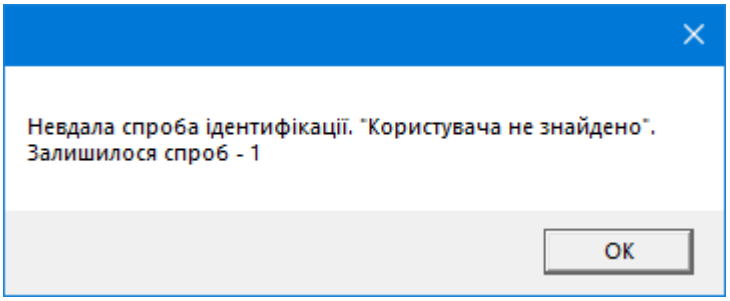


Рисунок 3.23 – Обмеження кількості спроб входу для користувача

Далі здійснимо ще одну спробу авторизації, при цьому рухи мишею будуть аналогічні як при реєстрації та завантажимо наданий користувачеві ключ доступу. Як наслідок – спроба входу успішна.

Застосуємо unit-тести [55] для тестової перевірки вище описаних сценаріїв, а саме:

– правильне введення логіну і невірний ключ;

– правильне введення логіну і некоректна ентропія;

– некоректний логін, ключ та вірна ентропія;

– некоректний логін, ентропія рухів та вірний ключ;

– коректне введення паролю, ключа, рухів миші.

Опишемо коротко процес проходження тесту.

Створимо новий тест для оцінки правильного написання коду (рис. 3.24):

```
namespace Catalog.Test
{
    [TestClass]
    public class UserModel_Tests
    {
        [TestMethod]
        public void TestMethod1()
    }
    }
```

Рисунок 3.24 – Тест для оцінки коду

Для перевірки правильності застосовуваного логіну, виключення ймовірності повторення логіну опишемо TestMetod (рис. 3.25):

```
{
    using Catalog;
    [TestClass]
    public class UserModel_Tests
    {
        [TestMethod]
        public void GetUserById_Test()
    }
}
```

Рисунок 3.35 – Тест для перевірки логіну

Після запуску описаних тестів отримуємо наступні результати (рис. 3.26):

```
1/passed/GetUserId_TestProc/CatalogTest
2/passed/GetUserById_Test/CatalogTest
3/passed/RolsCountInTable/CatalogTest
```

Рисунок 3.26 – Результати тесту

Наступним кроком тестування є перевірка фактору авторизації на основі аналізу рухів миші.

Для того аби дослідити надійність аналізу на основі рухів миші, здійснимо тестування десятьох учасників, мета яких – повторити задані рухи мишею еталонного користувача.

Досліджуватиметься шість показників за якими здійснюється аналіз:

– траєкторія руху комп’ютерної миші;

– відхилення руху комп’ютерної миші від зафіксованої траєкторії;

– швидкість руху комп’ютерної миші;

– прискорення руху комп’ютерної миші;

– нетипові рухи комп’ютерної миші;

– кліки комп’ютерної миші.

На рисунку 3.27 наведено діаграму, яка відображає відмінність показників десятьох учасників від еталонного зранку користувача.

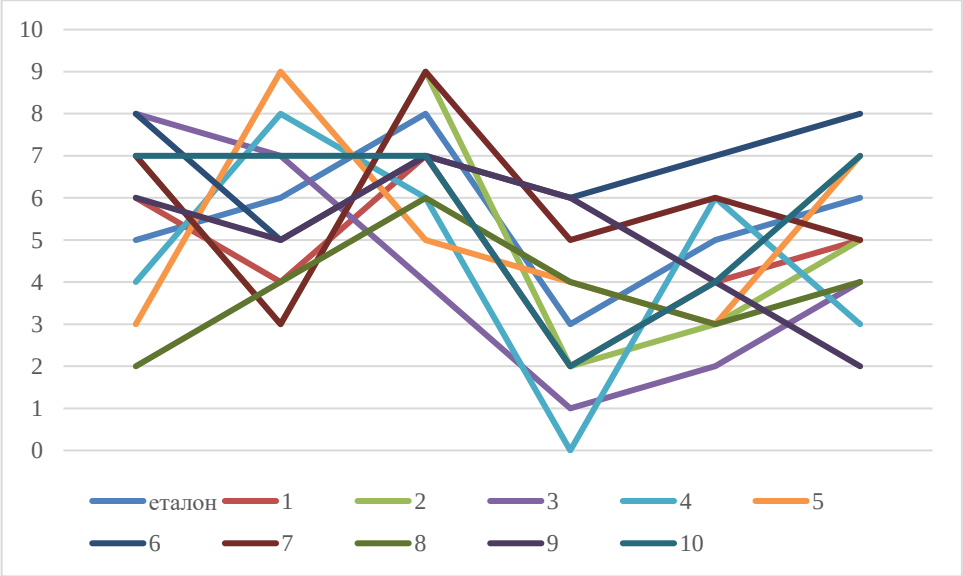


Рисунок 3.27 – Дослідження зміни показників

Як можемо побачити з діаграми, хоча і структура показників схожа, проте кількісні дані суттєво відрізняються, що є суттєвим при аналізі ентропії рухів миші. Даний дослід доводить, що поведінкова біометрична характеристика на основі рухів миші є індивідуальною для кожного користувача.

Для оцінки запропонованого методу аналізу рухів миші застосуємо показники оцінки біометричної ідентифікації [56]:

$$\mathrm{FAR} = \frac{\mathrm{NFA}}{\mathrm{NIVA}},$$

де FAR – ймовірність помилкового доступу;

NFA – кількість фактів помилкового доступу;

NIVA – загальна кількість звернень до системи.

$$\mathrm{FRR} = \frac{\mathrm{NFF}}{\mathrm{NEVA}},$$

де FRR – ймовірність помилкової відмови в доступі;

NFF – кількість фактів відмови в доступі;

NEVA – загальна кількість звернень до системи.

Порівняння реалізованого методу буде здійснюватися відносно аналізу рухів миші на основі нейронної мережі за участі десятьох учасників. Далі на

графіках наведемо результати проведеного дослідження (рис. 3.28, рис. 3.29).

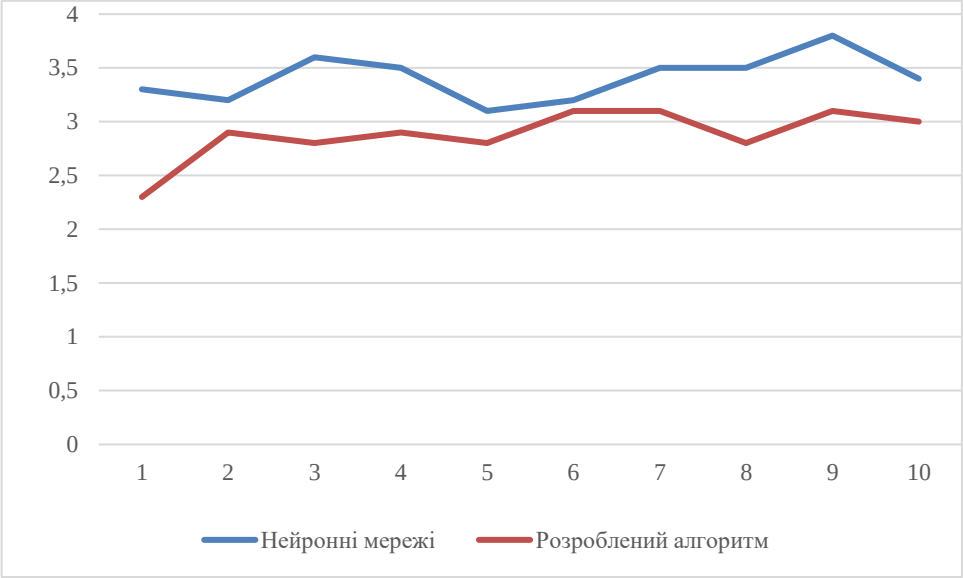


Рисунок 3.28 – Показники ймовірності помилкового доступу

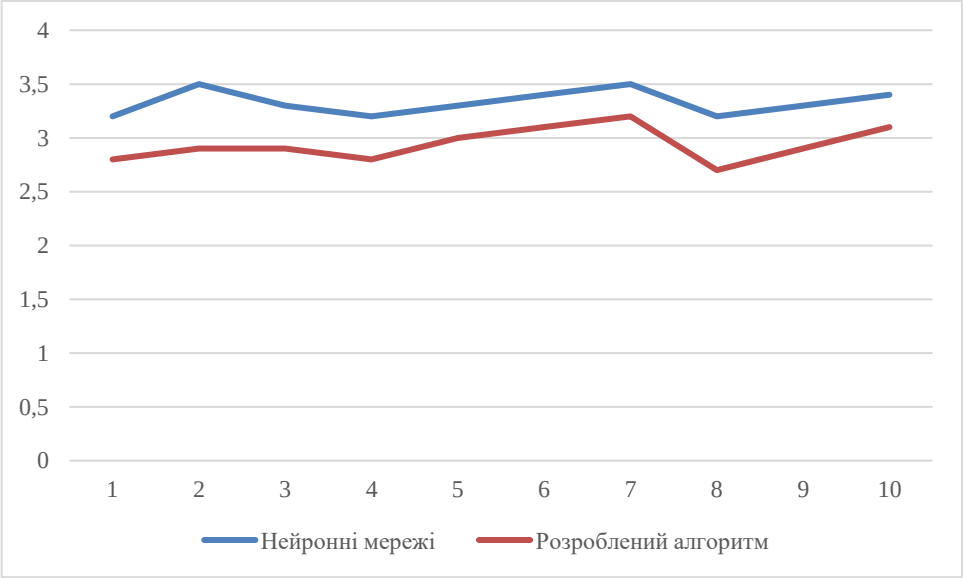


Рисунок 3.29 – Показники ймовірності помилкової відмови в доступі

З наведених графіків можемо побачити, що розроблюваний алгоритм на основі аналізу ентропії рухів миші має кращі показники біометричної ідентифікації, а також потребує менше часу на обробку даних. Враховуючи кількісні дані наведених показників можемо вважати, що достовірність автентифікації за даним методом підвищена орієнтовно на 11%.

Аналізуючи результати проведеного тестування, можемо зробити висновок, що програмно реалізований метод контролю доступу з

використанням двофакторної автентифікації користувачів на основі захищеного електронного ключа та аналізу ентропії рухів миші може бути застосований на практиці.

3.5 Висновки до Розділу 3

Отже, в даному розділі було здійснено практичну реалізацію програмного додатку на основі алгоритму підвищення достовірності автентифікації користувача на основі захищеного електронного ключа та аналізу ентропії рухів миші.

В ході виконання роботи було описано особливості програмної реалізації здійсненої на мові програмування C# у середовищі Visual Studio з використанням інтерфейсу Windows Forms; розроблено інструкцію користувача для роботи з додатком, проведено тестування та аналіз результатів розробки на основі unit-тестів та дослідження показників біометричної ідентифікації FAR та FRR.

Проведений аналіз програмної розробки показав, що додаток працює коректно, unit-тести мають позитивний результат, що свідчить про відсутність помилок у написаному коді, стійкість до зламу електронного ключа зумовлена обраною технологією JWT, аналіз дослідження показників FAR та FRR показав підвищення достовірності розпізнавання на 11%. Дані показники свідчать про можливість застосування розробленого додатку на основі двофакторної авторизації на практиці.

4 ЕКОНОМІЧНА ЧАСТИНА

В даному розділі проведемо економічний аналіз практичної доцільності розробки, для визначення її комерційного потенціалу та привабності для інвесторів.

Для цього опишемо оцінку комерційного потенціалу розробки програмного забезпечення; прогнозування витрат на виконання наукової роботи та впровадження її результатів; прогнозування комерційних ефектів від реалізації результатів розробки; розрахунок ефективності вкладених інвестицій та періоду їх окупності.

На основі відповідних результатів, зробимо висновки.

4.1 Оцінювання комерційного потенціалу розробки програмного забезпечення

Метою проведення технологічного аудиту є оцінювання комерційного потенціалу розробки, створеної в результаті науково-технічної діяльності [57].

Результатом магістерської кваліфікаційної роботи є розробка програмного засобу на основі алгоритму для підвищення достовірності автентифікації користувача на основі захищеного електронного ключа та аналізу ентропії рухів миші.

Для проведення технологічного аудиту залучено трьох незалежних експертів. У межах даної роботи такими експертами є викладачі кафедри МБІС:

– Карпінець В. В. (к.т.н., доцент каф. МБІС ВНТУ);

– Салієва О.В. (д.ф., викл. каф. МБІС ВНТУ);

– Шиян А. А. (к.ф.-м.н., доцент каф. МБІС ВНТУ).

Оцінювання комерційного потенціалу здійснимо за критеріями, що наведені в таблиці 4.1

Таблиця 4.1 – Критерії оцінювання комерційного потенціалу розробки бальна оцінка

Критерії оцінювання та бали (за 5-ти бальною шкалою)					
Кри-терій	0	1	2	3	4
Технічна здійсненність концепції:					
1	Достовірність концепції не підтверджена	Концепція підтверджена експертними висновками	Концепція підтверджена розрахунками	Концепція перевірена на практиці	Перевірено роботоздатність продукту в реальних умовах
Ринкові переваги (недоліки):					
2	Багато аналогів на малому ринку	Мало аналогів на малому ринку	Кілька аналогів на великому ринку	Один аналог на великому ринку	Продукт не має аналогів на великому ринку
3	Ціна продукту значно вища за ціни аналогів	Ціна продукту дещо вища за ціни аналогів	Ціна продукту приблизно дорівнює цінам аналогів	Ціна продукту дещо нижче за ціни аналогів	Ціна продукту значно нижче за ціни аналогів

Продовження таблиці 4.1

Критерії оцінювання та бали (за 5-ти бальною шкалою)					
Крит-тер.	0	1	2	3	4
4	Технічні та споживчі властивості продукту значно гірші, ніж в аналогів	Технічні та споживчі властивості продукту трохи гірші, ніж в аналогів	Технічні та споживчі властивості продукту на рівні аналогів	Технічні та споживчі властивості продукту трохи кращі, ніж в аналогів	Технічні та споживчі властивості продукту значно кращі, ніж в аналогів
5	Експлуатацій ні витрати значно вищі, ніж в аналогів	Експлуатацій ні витрати дещо вищі, ніж в аналогів	Експлуатацій ні витрати на рівні експл. витрат аналогів	Експлуатацій ні витрати трохи нижчі, ніж в аналогів	Експлуатаційні витрати значно нижчі, ніж в аналогів
Ринкові перспективи					
6	Ринок малий і не має позитивної динаміки	Ринок малий, але має позитивну динаміку	Середній ринок з позитивною динамікою	Великий стабільний ринок	Великий ринок з позитивною динамікою
7	Активна конкуренція великих компаній на ринку	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкурентів немає

Продовження таблиці 4.1

Практична здійсненність					
8	Відсутні фахівці як з технічної, так і з комерційної реалізації ідеї	Необхідно наймати фахівців або витрачати значні кошти та час на навч. наявних фахівців	Необхідне незначне навчання фахівців та збільшення їх штату	Необхідне незначне навчання фахівців	Є фахівці з питань як з технічної, так і з комерційної реалізації ідеї
9	Потрібні значні фінансові ресурси, які відсутні. Джерела фінансування ідеї відсутні	Потрібні незначні фінансові ресурси. Джерела фінансування відсутні	Потрібні значні фінансові ресурси. Джерела фінансування є	Потрібні незначні фінансові ресурси. Джерела фінансування є	Не потребує додаткового фінансування
10	Необхідна розробка нових матеріалів	Потрібні матеріали, що використовуються у військово-промисловому комплексі	Потрібні дорогі матеріали	Потрібні досяжні та дешеві матеріали	Всі матеріали для реалізації ідеї відомі та давно використовуються у виробництві
11	Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років. Термін окупності інвестицій більше 10-ти років	Термін реалізації ідеї від 3-х до 5-ти років. Термін окупності інвестицій більше 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій від 3-х до 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій менше 3-х років
12	Необхідна розробка регламентних документів та отримання великої кількості дозвільних документів на виробництво та реалізацію продукту	Необхідно отримання великої к-ті дозвільних документів на вир-во та реалізацію продукту, що вимагає значних коштів та часу	Процедура отримання дозвільних документів для виробництва та реалізації продукту вимагає незначних коштів та часу	Необхідно тільки повідомлення відповідним органам про виробництво та реалізацію продукту	Відсутні будь-які регламентні обмеження на виробництво та реалізацію продукту

Результати оцінювання комерційного потенціалу експертами розробки зведено в таблицю 4.2.

Таблиця 4.2 – Результати оцінювання комерційного потенціалу розробки

Критерії	Прізвище, ініціали, посада експерта		
	1 – Карпінець В.В.	2 – Салієва О.В.	3 – Шиян А.А
1	4	4	4
Ринкові переваги (недоліки):			
2	2	3	3
3	4	4	4
4	3	4	3
5	3	4	4
Ринкові перспективи			
6	3	3	3
7	3	3	3
Практична здійсненність			
8	3	4	4
9	4	3	3
10	3	4	3
11	4	4	4
12	4	3	3
Сума балів	$СБ_1 = 40$	$СБ_1 = 43$	$СБ_1 = 41$
Середньоарифметична сума балів $\overline{СБ}$	$\overline{СБ} = 41{,}3$		

За даними таблиці 4.2 можна зробити висновок, щодо рівня комерційного потенціалу розробки. Зважимо на результат й порівняємо його з рівнями комерційного потенціалу розробки, що представлено в таблиці 4.3.

Таблиця 4.3 – Рівні комерційного потенціалу розробки

Середньоарифметична сума балів $\overline{СБ}$, розрахована на основі висновків експертів	Рівень комерційного потенціалу розробки
0 – 10	Низький
11 – 20	Нижче середнього
21 – 30	Середній
31 – 40	Вище середнього
41 – 48	Високий

Рівень комерційного потенціалу розробки, становить $41{,}3$ балів, що відповідає рівню «високий».

Проаналізуємо суть технічної проблеми та розглянемо аналоги. Наукова новизна розробки полягає в підвищенні достовірності автентифікації користувача на основі захищеного електронного ключа та аналізу ентропії рухів миші.

Розробка програмного засобу має на меті практичну реалізацію удосконаленого алгоритму автентифікації користувача у захищеному додатку на основі двофакторної автентифікації (застосуванні захищеного електронного ключа з використанням алгоритму AES та аналізу ентропії рухів миші, що розраховуються за певними параметрами і є індивідуальною характеристикою для кожного користувача). Застосування програмного електронного ключа для поставленої задачі зумовлено такими його перевагами як зниження витрат, підвищення зручності користування, зниження ймовірності втрати чи крадіжки та менший ризик атак через посередника. Застосування аналізу ентропії рухів миші дозволить здійснювати автентифікацію користувача на основі біометричних поведінкових характеристик. Програмний додаток розроблюється з розрахунком на користувачів системи, що володіють правом доступу до захищуваних даних, шляхом проходження двоетапної перевірки ідентифікації особи. Під час виконання роботи було здійснено тестування розробленого прорамного продукту, яке показало позитивні результати показники яких свідчать про можливість застосування розробленого додатку на основі двофакторної авторизації на практиці.

Враховуючи такі переваги розробленого методу, можемо порівняти його з одним із аналогів, а саме Authy (табл. 1.2). У таблиці 4.4 наведені основні технічні показники аналога і нового програмного продукту.

Таблиця 4.4 – Основні технічні показники аналога і нового програмного продукту

Показники, %	Аналог	Нова розробка	Відношення параметрів нової розробки до параметрів аналога
Функціональність	85	100	1,18
Надійність	65	100	1,54
Сумісність	90	100	1,11
Супровід	80	100	1,25
Економія ресурсів і часу	95	100	1,05
Простота використання	75	100	1,33

Системи двофакторної аутентифікації є найбільш поширеними та використовують два різних фактори при аутентифікації користувачів. Вона використовується в основному в мережевих сервісах, однак і для операційних систем є не менш актуальною.

Проблема витоку інформації актуальна в усьому світі та застосування двофакторної автентифікації для захисту інформації стане додатковим бар'єром для зловмисників.

Розроблений програмний додаток, порівняно з аналогами, має ряд переваг, зокрема: удосконалений метод підвищення достовірності на аналізі ентропії рухів миші. Слід зауважити, що дослідження рухів миші є поведінковою характеристикою, що відноситься до біометричної ідентифікації, яка має високі якісні показники розпізнавання.

Також достовірність ідентифікації підвищується за рахунок використання електронного ключа, який є також захищеним та у разі втрати або викрадення не послугує ключем доступу зловмисника до певних даних, додаткове шифрування захищає дані, що становлять електронний ключ.

На підставі вищевикладеного можна стверджувати, що нове технічне рішення, що пропонується для розробки, буде мати кращі показники, ніж у аналога та більшою мірою задовольнить потреби споживачів. Тому його розробка та впровадження є актуальним та доцільним.

Програмний засіб на сьогодні має перспективу та користь як для пересічних користувачів так і для спецслужб. Продукт, який пропонується є реалізованим засобом, що дозволяє проводити автентифікацію користувачів в системі.

Для реалізованого проекту є характерним можливість надання якісних послуг з технічної підтримки на всіх етапах використання програмного продукту користувачами.

Передбачається, що засобами розповсюдження та збуту розробки можуть бути інтернет-магазин з продажу програмного забезпечення та/або офіційний сайт додатку.

Доцільним є встановлення середньої ціни на програмний продукт задля якісного та швидкого виходу на ринок послуг. Також це може допомогти, як в боротьбі з конкурентами, так і в захопленні більшої частки ринку.

Також передбачається проведення рекламної компанії для залучення інвесторів та поширення інформації про продукт серед пересічних користувачів. Просування продукції на ринку планується здійснювати завдяки цільовій рекламі та на технічних форумах і сайтах технічних новинок.

Враховуючи, що на сьогодні питання забезпеченості високого рівня безпеки та захищеності даних в процесі автентифікації є актуальними – цілком ймовірний високий попит на запропонований програмний продукт.

4.2 Прогнозування витрат на виконання наукової роботи та впровадження її результатів

Прогнозування витрат на виконання науково-дослідної, дослідно-конструкторської та конструкторсько-технологічної роботи складається з таких етапів:

1-й етап: розрахунок витрат, які безпосередньо стосуються виконавців даного розділу роботи;

2-й етап: розрахунок загальних витрат на виконання даної роботи;

3-й етап: прогнозування загальних витрат на виконання та впровадження результатів даної роботи.

Виконаємо розрахунок витрат, які безпосередньо стосуються виконавців даного розділу роботи, за такими статтями та формулами, приймаючи до уваги те, що для розробки інформаційної технології було залучено одного розробника програмного забезпечення.

1. Основна заробітна $З_о$:

$$З_о = \frac{М}{Т_р} \cdot t, \text{грн.} \quad (4.1)$$

де $М$ – місячний посадовий оклад – $18\,000$ грн.;

$Т_р$ – число робочих днів в місяці; приблизно $Т_р = 20$ днів;

t – число робочих днів роботи – 35 днів.

Таким чином:

$$З_о = \frac{18000}{20} \cdot 35 = 31\,500\ (грн.)$$

Таблиця 4.5 – Витрати по заробітній платі

Найменування посади	Місячний посадовий оклад, грн.	Оплата за робочий день, грн.	Число днів роботи	Витрати на заробітну плату
Розробник	18 000	900	35	31 500
Всього				31 500

2. Додаткова заробітна плата $З_д$ працівників розраховується як 12% від основної заробітної плати:

$$З_д = 0{,}12 \cdot 31\,500 = 3\,780 (грн.) - для\ розробника$$

3. Нарахування на заробітну плату $Н_{зп}$ розробника становить:

$$Н_{зп} = (З_о + З_д) \cdot \frac{\beta}{100} \quad (4.2)$$

де $З_о$ – основна заробітна плата розробника;

$З_д$ – додаткова заробітна плата розробника;

β – ставка єдиного внеску на загальнообов'язкове державне соціальне страхування – 22%.

$$Н_{зп} = (31\,500 + 3\,780) \cdot 0{,}22 = 7\,761{,}6\ (грн.)$$

4. Амортизація обладнання, комп'ютерів та приміщень, які використовувались під час виконання даного етапу роботи розраховуємо за формулою:

$$А = \frac{Ц \cdot Т}{12 \cdot Т_В} \quad (4.3)$$

де Ц – загальна балансова вартість обладнання, приміщення тощо, грн.;

Т – фактична тривалість використання, міс;

$Т_В$ – термін використання обладнання, приміщень тощо, роки.

Розробка програмного забезпечення ведеться орієнтовно 1,5 місяці.

Для офісного приміщення $А = \frac{120\,000 \cdot 1,5}{12 \cdot 5} = 3\,000 грн.$; для комп'ютера $А = \frac{15\,000 \cdot 1,5}{12 \cdot 2} = 937,5 грн.$; для монітора $А = \frac{8\,000 \cdot 1,5}{12 \cdot 2} = 500 грн.$

Розрахунки зведено до таблиці 4.6:

Таблиця 4.6 – Амортизаційні відрахування

Наймену-вання	Балансова вартість (грн.)	Термін використання (років)	Фактична тривалість в-ня, (міс.)	Величина ам.. відрахувань, (грн..)
Офісне приміщення	120 000	5	1,5	3 000
Комп'ютер	15 000	2	1,5	937,5
Монітор	8 000	2	1,5	500
			Всього	4 437,5

5. Витрати на комплектуючі К, що були використані під час виконання даного етапу роботи, розраховуються за формулою:

$$К = \sum_{1}^{n} Н_i \cdot Ц_i \cdot К_i \ (грн.) \qquad (4.4)$$

де $Н_i$ – кількість комплектуючих *i*-го виду, шт.;

$Ц_i$– ціна комплектуючих *i*-го виду, грн.;

$К_i$ – коефіцієнт транспортних витрат, $К_i$ = (1,1…1,15);

n – кількість видів комплектуючих.

Таблиця 4.7 – Витрати на комплектуючі

Найменування комплектувальних	Кількість	Ціна за штуку, грн.	Сума, грн.	Примітка
Клавіатура	1	750 грн.	750 грн.	
Комп'ютерна мишка	1	350 грн.	350 грн.	
Всього:			$К_i = 1,2$	1 100 грн.

6. Витрати на силову електроенергію $В_е$ розраховуються за формулою:

$$В_е = В \cdot П \cdot Ф \cdot К_п \ (грн.) \qquad (4.5)$$

де В – вартість 1 кВт-год. (на сьогодні для підприємців вартість 3,45грн./кВт-год);

П – установлена потужність обладнання – 0,8 кВт;

Ф – фактична кількість годин роботи обладнання – 350 годин,

$К_п$ – коефіцієнт використання потужності.

$$В_е = 6{,}4 \cdot 0{,}8 \cdot 350 \cdot 0{,}14 = 250{,}9 \text{ (грн)}.$$

7. Інші витрати $В_{ін}$ охоплюють:

– витрати на управління організацією;

– оплату службових відряджень;

– витрати на утримання, ремонт та експлуатацію основних засобів;

– витрати на опалення, освітлення, водопостачання, охорону праці тощо.

Інші витрати $В_{ін}$ можна прийняти як 100% від суми основної заробітної плати розробника:

$$В_{ін} = 31\,500 \cdot 1 = 31\,500 \text{ (грн)}$$

Послуги Інтернету – 350 грн., канцтовари – 300 грн. Загальна вартість становить:

$$350 + 300 = 650 \text{ (грн.)}$$

8. Сума всіх попередніх статей витрат дає витрати на виконання даної частини роботи – В.

$$В = 31\,500 + 3\,780 + 7\,761{,}6 + 4\,437{,}5 + 1100 + 250{,}9 + 31\,500 + 650 = 80\,980 \text{ (грн.)}$$

9. Проведемо прогнозування загальних витрат ЗВ на виконання та впровадження результатів виконаної наукової роботи. Прогнозування здійснюється за формулою:

$$ЗВ = \frac{В_{заг}}{\beta}, \text{ грн.} \quad (4.6)$$

де β – коефіцієнт, який характеризує етап (стадію) виконання даної роботи.

Так, якщо розробка знаходиться:

– на стадії науково-дослідних робіт, то $\beta \approx 0{,}1$;

– на стадії технічного проектування, то $\beta \approx 0{,}2$;

– на стадії розробки конструкторської документації, то $\beta \approx 0{,}3$;

– на стадії розробки технологій, то $\beta \approx 0{,}4$;

– на стадії розробки дослідного зразка, то $\beta \approx 0{,}5$;

– на стадії розробки промислового зразка, $\beta \approx 0{,}7$;

– на стадії впровадження, то $\beta \approx 0{,}9$.

$В_{заг}$ – загальна вартість всієї наукової роботи.

$$В = 80\,980 \text{ (грн.)}$$

$$ЗВ = \frac{80\,980}{0{,}7} = 115\,685{,}7 \text{ (грн.)}$$

Отже, прогноз загальних витрат ЗВ на виконання та впровадження результатів виконаної наукової роботи складає $115\,685{,}7$ (грн.)

4.3 Прогнозування комерційних ефектів від реалізації результатів розробки

У даному підрозділі проведемо кількісне прогнозування, яку вигоду, зиск можна отримати у майбутньому від впровадження результатів виконаної наукової роботи.

В умовах ринку узагальнюючим позитивним результатом, що його отримує підприємство від впровадження результатів тієї чи іншої розробки, є збільшення чистого прибутку підприємства. Зростання чистого прибутку можна оцінити у теперішній вартості грошей.

Зростання чистого прибутку забезпечить інвестору надходження додаткових коштів, які дозволять покращити фінансові результати діяльності.

Виконання даної наукової роботи та впровадження її результатів складає приблизно 1 рік. Позитивні результати від впровадження розробки очікуються вже в перші місяці після впровадження.

Проведемо детальне прогнозування позитивних результатів та кількісне їх оцінювання по роках.

Обчислимо збільшення чистого прибутку підприємства $\Delta П_i$ для кожного із років, протягом яких очікується отримання позитивних результатів від впровадження розробки, розраховується за формулою:

$$\Delta П_i = \sum_{1}^{n} (\Delta П_я \cdot N + П_я \cdot \Delta N)_i \quad (4.7)$$

де $\Delta П_я$ – покращення основного якісного показника від впровадження результатів розробки у даному році;

N – основний кількісний показник, який визначає діяльність підприємства у даному році до впровадження результатів наукової розробки;

ΔN – покращення основного кількісного показника діяльності підприємства від впровадження результатів розробки;

$П_я$ – основний якісний показник, який визначає діяльність підприємства у даному році після впровадження результатів наукової розробки;

n – кількість років, протягом яких очікується отримання позитивних результатів від впровадження розробки.

Припустимо, що внаслідок впровадження результатів наукової розробки чистий прибуток підприємства збільшиться на 120 грн., а кількість одиниць реалізованої послуги збільшиться:

– протягом першого року – на 450 од.,

– протягом другого року – ще на 600 од.,

– протягом третього року – ще на 800 од.

Орієнтовно: реалізація продукції до впровадження результатів наукової розробки складала 1 шт., а прибуток, що його отримувало підприємство на одиницю продукції до впровадження результатів наукової розробки – 150 грн.

Потрібно спрогнозувати збільшення чистого прибутку підприємства від впровадження результатів наукової розробки у кожному році відносно базового.

Збільшення чистого прибутку підприємства $\Delta П_1$ протягом першого року складе:

$$\Delta П_1 = 150 \cdot 1 + (150 + 120) \cdot 450 = 121\,650 \text{ (грн.)}$$

Обчислимо збільшення чистого прибутку підприємства $\Delta П_2$ протягом другого року:

$$\Delta П_2 = 150 \cdot 1 + (150 + 120) \cdot (450 + 600) = 283\,650 \text{ (грн.)}$$

Збільшення чистого прибутку підприємства $\Delta П_3$ протягом третього року становитиме:

$$\Delta П_3 = 150 \cdot 1 + (150 + 120) \cdot (450 + 600 + 800) = 499\,650 \text{ (грн.)}$$

Отже, розрахунки показують, що відповідно прогнозуванню комерційний ефект від впровадження розробки виражається у значному збільшенні чистого прибутку підприємства.

4.4 Розрахунок ефективності вкладених інвестицій та періоду їх окупності

Основними показниками, які визначають доцільність фінансування наукової розробки певним інвестором, є абсолютна і відносна ефективність вкладених інвестицій та термін їх окупності.

Розрахунок ефективності вкладених інвестицій передбачає:

1-й крок. Розрахунок теперішньої вартості інвестицій PV, що вкладаються в наукову розробку.

Такою вартістю ми можемо вважати прогнозовану величину загальних витрат ЗВ на виконання та впровадження результатів НДДКР, тобто $ЗВ = PV = 115\,523{,}3$ (грн.)

2-й крок. Розрахуємо очікуване збільшення прибутку $\Delta П_i$, що його отримає підприємство (організація) від впровадження результатів наукової розробки, для кожного із років, починаючи з першого року впровадження. Таке збільшення прибутку також було розраховане нами раніше та становить:

$$\Delta П_1 = 121\,650 \text{ (грн)}, \Delta П_2 = 283\,650 \text{ (грн)}, \Delta П_3 = 499\,650 \text{(грн)}.$$

3-й крок. Будуємо вісь часу, на якій відображаємо всі платежі (інвестиції та прибутки), що мають місце під час виконання науково-дослідної роботи та впровадження її результатів. Рисунок 4.1 характеризує рух платежів (інвестицій

та додаткових прибутків).

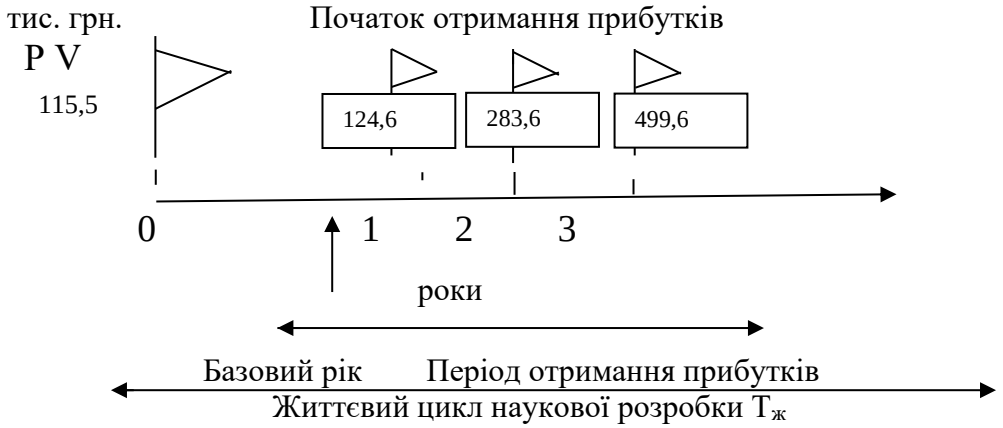


Рисунок 4.1 – Вісь часу з фіксацією платежів, що мають місце під час розробки та впровадження результатів НДДКР

4-й крок. Розрахуємо абсолютну ефективність вкладених інвестицій $Е_{абс}$ за формулою:

$$Е_{абс} = (ПП - PV), (грн.) \quad (4.8)$$

де ПП – приведена вартість всіх чистих прибутків, що їх отримає підприємство (організація) від реалізації результатів наукової розробки, грн.;

PV – теперішня вартість інвестицій $PV = ЗВ, грн.$

Приведена вартість всіх чистих прибутків ПП розраховується за формулою:

$$ПП = \sum_{1}^{Т} \frac{\Delta П_i}{(1+\tau)^t}, (грн) \quad (4.9)$$

де $\Delta П_i$ – збільшення чистого прибутку у кожному із років, протягом яких виявляються результати виконаної та впровадженої НДДКР, грн.;

Т – період часу, протягом якого виявляються результати впровадженої НДДКР, роки;

τ – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні – 0,1;

t – період часу (в роках) від моменту отримання чистого прибутку до точки «0»;

$$\text{ПП} = \frac{121\,650}{(1+0{,}1)^1} + \frac{283\,650}{(1+0{,}1)^2} + \frac{499\,650}{(1+0{,}1)^3} = 720\,406{,}8\ (\text{грн.})$$

$$\text{Е}_{\text{абс}} = 720\,406{,}8 - 115\,685{,}7 = 604\,721{,}1\ (\text{грн.})$$

Оскільки $\text{Е}_{\text{абс}} > 0$, результат від проведення наукових досліджень щодо розробки програмного продукту та їх впровадження принесе прибуток, тобто є доцільним, проте це ще не свідчить про те, що інвестор буде зацікавлений у фінансуванні даної програми.

5-й крок. Розрахуємо відносну (щорічну) ефективність вкладених в наукову розробку інвестицій $\text{Е}_{\text{в}}$ за формулою:

$$\text{Е}_{\text{в}} = \sqrt[\text{Т}_{\text{ж}}]{1 + \frac{\text{Е}_{\text{абс}}}{PV}} - 1 \qquad (4.10)$$

де $\text{Е}_{\text{абс}}$ – абсолютна ефективність вкладених інвестицій, грн.;

PV – теперішня вартість інвестицій $PV = \text{ЗВ}$, грн.

$\text{Т}_{\text{ж}}$ – життєвий цикл наукової розробки, роки.

$$E_{\text{в}} = \sqrt[3]{1 + \frac{604\,721{,}1}{115\,685{,}7}} - 1 = \sqrt[3]{6{,}2} - 1 = 0{,}82 \text{ або } 82\%$$

Порівняємо $E_{\text{в}}$ з мінімальною (бар'єрною) ставкою дисконтування τ_{min}, яка визначає ту мінімальну дохідність, нижче за яку інвестиції вкладатися не будуть.

Спрогнозуємо величину τ_{min}.

У загальному вигляді мінімальна (бар'єрна) ставка дисконтування τ_{min} визначається за формулою:

$$\tau_{min} = d + f \qquad (4.11)$$

де d – середньозважена ставка за депозитними операціями в комерційних банках;

$d = 0{,}2$;

f – показник, що характеризує ризикованість вкладень; величина $f = 0{,}5$.

$$\tau_{min} = 0{,}2 + 0{,}5 = 0{,}7$$

Оскільки $E_в = 82\% > \tau_{min} = 70\%$, то у інвестора є потенційна зацікавленість у фінансуванні даної наукової розробки.

6-й крок. Розрахуємо термін окупності вкладених у реалізацію наукового проекту інвестицій $Т_{ок}$ за формулою:

$$Т_{ок} = \frac{1}{Е_в}, \text{рік} \quad (4.12)$$

$$Т_{ок} = \frac{1}{0{,}82} = 1{,}2 \text{ (року)}$$

Оскільки термін окупності вкладених у реалізацію наукового проекту інвестицій менше трьох років ($Т_{ок} < 3$ років), то фінансування нової розробки є доцільним.

4.5 Висновки до розділу

В даному розділі було виконано оцінювання комерційного потенціалу розробки програмного засобу на основі алгоритму для підвищення достовірності автентифікації користувача на основі захищеного електронного ключа та аналізу ентропії рухів миші.

Проведено технологічний аудит з залученням трьох незалежних експертів. Визначено, що рівень комерційного потенціалу розробки вище середнього.

Проведено порівняння з аналогом. Згідно з проведеним оцінюванням нова розробка є якісною та конкурентоспроможною.

Рівень комерційного потенціалу розробки, становить 41,3 балів, що відповідає рівню «високий».

Згідно із розрахунками всіх статей витрат на виконання науково-дослідної, дослідно-конструкторської та конструкторсько-технологічної роботи загальні витрати на розробку складають 115 685,7 (грн.). Розрахована абсолютна ефективність вкладених інвестицій в сумі 604 721,1 (грн.) свідчить про отримання прибутку інвестором від комерціалізації програмного продукту.

Щорічна ефективність вкладених в наукову розробку інвестицій складає 82%, що вище за мінімальну бар'єрну ставку дисконтування, яка складає 70%. Це означає потенційну зацікавленість інвесторів у фінансуванні розробки.

Термін окупності вкладених у реалізацію проекту інвестицій становить 1,2 (року), що також свідчить про доцільність фінансування нової розробки.

Отже, проаналізувавши отримані економічні показники, можна вважати, що запропонована розробка програмного засобу має високий комерційний потенціал, а тому є доцільною для подальшого впровадження.

ВИСНОВОК

В магістерській кваліфікаційній роботі здійснювалась розробка в галузі підвищення достовірності автентифікації користувача на основі захищеного електронного ключа та аналізу ентропії рухів миші.

Багатофакторна автентифікація є найбільш ефективним методом захисту від несанкціонованого доступу, оскільки використання кількох незалежних факторів значно зменшує ймовірність, що вони будуть використані одночасно. Надійність обраного рішення складається із надійності його елементів.

Виходячи із поставлених задач роботи першому розділі роботи досліджено сучасні методи автентифікації користувачів, здійснено аналіз особливостей двофакторної автентифікації, описано алгоритми автентифікації користувачів з використанням електронного ключа та поведінкових характеристик. Для здійснення захисту даних електронних ключів обрано алгоритм симетричного шифрування AES.

В другому розділі було здійснено удосконалення методу для підвищення достовірності автентифікації користувача.

Удосконалення алгоритму автентифікації користувача у захищеному додатку полягає у реалізації двофакторної автентифікації та застосуванні захищеного електронного ключа з використанням алгоритму AES та аналізу ентропії рухів миші, що розраховуються за певними параметрами і є індивідуальною характеристикою для кожного користувача.

Застосування програмного електронного ключа для поставленої задачі зумовлено такими його перевагами як зниження витрат, підвищення зручності користування, зниження ймовірності втрати чи крадіжки та менший ризик атак через посередника. Застосування аналізу ентропії рухів миші дозволить здійснювати автентифікацію користувача на основі біометричних поведінкових характеристик.

До розроблюваного програмного додатку на основі двофакторної автентифікації висуваються наступні вимоги: підтримка функції авторизації та

реєстрації користувачів; налаштування розмежування доступу на основі рольової моделі; реалізація розділів введення даних при авторизації користувача; можливість формування електронного ключа за заданим алгоритмом; можливість здійснення аналізу ентропії рухів миші за заданим алгоритмом; аналіз отриманих даних автентифікації, перевірка та прийняття рішення про надання або відмову в доступі; обмеження кількості спроб некоректного входу; можливість поновлення електронного ключа при його втраті або блокуванні.

Враховуючи особливості поставлених задач роботи, для програмної реалізації, було обрано мову об’єктно-орієнтованого програмування C#, середовище програмування Visual Studio з використанням інтерфейсу Windows Forms.

В третьому розділі роботи описано здійснення практичної реалізації програмного додатку та її особливості.

Проведений аналіз програмної розробка показав, що додаток працює коректно, unit-тести мають позитивний результат, що свідчить про відсутність помилок у написаному коді, стійкість до зламу електронного ключа зумовлена обраною технологією JWT, аналіз дослідження показників FAR та FRR показав підвищення достовірності розпізнавання на 11%. Дані показники свідчать про

В четвертому розділі роботи було досліджено економічну доцільність розробки. Результати, отримані внаслідок написання розділу, є позитивними та свідчать про можливість застосування розробленого додатку на основі двофакторної авторизації на практиці.

Отже, за підсумками виконаних завдань та отриманих результатів можна вважати, що поставлена початкова мета роботи виконана, а саме – здійснено підвищення достовірності автентифікації користувача на основі захищеного електронного ключа та аналізу ентропії рухів миші.

ПЕРЕЛІК ПОСИЛАНЬ

1. Martin Fowler. Microservices. *Стаття:* веб-сайт. URL: https://martinfowler.com/articles/microservices.html (дата звернення: 12.10.2022).

2. AISO/IEC 27005:2005 «Інформаційні технологій. Методи захисту. Система управління інформаційною безпекою. Вимоги» *Електрон. ресурс:* веб-сайт. URL: http://www.dstszi.gov.ua/dstszi/control (дата звернення: 12.10.2022).

3. Самохвалов Ю.Я., Темпіков В.О., Хорошко В.О. Організаційно-технічне забезпечення захисту інформації: Навчальний посібник / За ред. проф. Хорошка В.О. – К.: НАУ, 2002. – С. 207

4. Security and Compliance: Customer Controls for Information Protection in Office 365. *Microsoft:* веб-сайт. URL: http://download.microsoft.com/download/ (дата звернення: 12.10.2022).

5. НД ТЗІ 1.1-002-99 Загальні положення щодо захисту інформації в комп'ютерних системах від несанкціонованого доступу

6. Ідентифікація Аутентифікація і авторизація: що це і в чому відмінність *Qagroup:* веб-сайт. URL: https://qagroup.com.ua/publications/autentyfikatciia-i-avtoryzatciia/ (дата звернення: 12.10.2022).

7. Способи аутентифікації в веб-додатках. *Dataart:* веб-сайт. URL: https://www.dataart.com.ua/news/ (дата звернення: 12.10.2022).

8. Авторизація (інформаційні технології) // Велика українська енциклопедія : у 30 т. / проф. А. М. Киридон (відп. ред.) та ін. – 2016. – Т. 1 : А – Акц. – 592 с. – ISBN 978-617-7238-39-2.

9. Захист програмного забезпечення від несанкціонованого доступу. *Студфайли:* веб-сайт. URL: https://studfiles.net/ (дата звернення: 16.10.2022).

10. What is single sign-on (SSO)? *Techtarget:* веб-сайт. URL: https://www.techtarget.com/searchsecurity/ (дата звернення: 16.10.2022).

11. B. Campbell. OAuth 2.0 Form Post Response Mode *Openid:* веб-сайт. URL: https://openid.net/specs/oauth-v2-form-post-response-mode-1_0.html (дата звернення: 16.10.2022).

12. Коваль Л.Г., Злепко С.М. Методи і технології біометричної ідентифікації за результатами літературних джерел. *Вчені записки ТНУ імені В.І. Вернадського. Серія: технічні науки.* 2019. Вип. 2. С. 104-112.

13. Ляшенко Г.Є., Астраханцев А.А. Дослідження ефективності методів біометричної автентифікації. *Захист інформації в інформаційних комунікаційних системах*. 2017. Вип. 2 (148). С. 111-114.

14. Двофакторна авторизація як спосіб додаткового захисту. *Стаття:* веб-сайт. URL: http://cikt.kubg.edu.ua/ (дата звернення: 16.10.2022).

15. Що таке двофакторна автентифікація або 2FA? *Dropbox:* веб-сайт. URL: https://experience.dropbox.com/uk-ua/resources/what-is-2fa (дата звернення 18.10.2022).

16. Опція двофакторної авторизації: переваги та недоліки. *Chvv:* веб-сайт. URL: http://chvv.com.ua/optsiya-dvofaktornoyi-avtorizatsiyi-perevagi-ta-nedoliki/ (дата звернення 18.10.2022).

17. Google Authenticator. *GooglePlay:* веб-сайт. URL: https://play.google.com/store/apps/details?id=com.google.android.apps.authenticator2&hl=uk&gl=US (дата звернення 18.10.2022).

18. Duo Mobile. *GooglePlay:* веб-сайт. URL: https://play.google.com/store/apps/details?id=com.duosecurity.duomobile&hl=uk&gl=US (дата звернення 18.10.2022).

19. Microsoft Authenticator. *GooglePlay:* веб-сайт. URL: https://play.google.com/store/apps/details?id=com.azure.authenticator&hl=uk&gl=US (дата звернення 18.10.2022).

20. Free OTP. *GooglePlay:* веб-сайт. URL: https://play.google.com/store/apps/details?id=org.liberty.android.freeotpplus&hl=uk&gl=AZ (дата звернення 18.10.2022).

21. Authy. *GooglePlay:* веб-сайт. URL: https://play.google.com/store/apps/details?id=com.authy.authy&hl=uk&gl=US (дата звернення 18.10.2022).

22. Бурячок В.Л. Політика інформаційної безпеки: навчальний посібник. / В.Л.Бурячок, Р.В.Грищук, В.О.Хорошко / За заг. ред. докт. техн. наук, проф. В.О. Хорошка. – К.: ПВП «Задруга», 2014. – 134 с

23. What is a JWT? Understanding JSON Web. *Supertokens:* веб-сайт. URL: Tokens https://supertokens.com/blog/what-is-jwt (дата звернення 18.10.2022).

24. Як проходити двохетапну перевірку за допомогою ключа безпеки *Support:* веб-сайт. URL: https://support.google.com/accounts/answer/6103523?hl=uk&co=GENIE.Platform%3DAndroid (дата звернення 21.10.2022).

25. Що таке апаратний токен? *Стаття:* веб-сайт. URL: https://hideez.com/uk/blogs/news/hardware-tokens (дата звернення 21.10.2022).

26. What is multi factor authentication. *Loginradius:* веб-сайт. URL: https://www.loginradius.com/blog/2019/06/what-is-multi-factor-authentication/ (дата звернення 21.10.2022).

27. Що таке токени і чому вони всім потрібні? *Актив:* веб-сайт URL: https://aktiv.ua/ua/materials/articles/chto-takoe-tokeny (дата звернення 13.10.2022)

28. Біометрія як універсальний спосіб ідентифікації людини. *Матеріали онлайн:* веб-сайт. URL: http://bablyukh.clan.su/publ/1-1-0-4 (дата звернення 20.10.2022).

29. Сарбуков А. , Грушо А. Автентифікація в комп’ютерних системах. *Системи безпеки.* 2003. Вип. 5 (53). С. 25-29.

30. Тарнавський Ю. А. Технології захисту інформації / Юрій Адамович Тарнавський. – Київ, 2018.

31. Технології захисту інформації : навчальний посібник / С. Е. Остапов,С. П. Євсеєв, О. Г. Король. – Х. : Вид. ХНЕУ, 2013. – 476 с.

32. Кібербезпека : сучасні технології захисту. Навчальний посібник для студентів вищих навчальних закладів. / С. Е. Остапов, С. П. Євсеєв, О.Г. Король. – Львів: «Новий Світ- 2000», 2020 . – 678 с.

33. Остапов С. Е. Технології захисту інформації : навчальний посібник / С. Е. Остапов, С. П. Євсеєв, О. Г. Король. – Х. : Вид. ХНЕУ, 2013. – 476 с.

34. Данченков Я.В. Теорія інформації: *навчальний посібник* / Я.В. Данченков. – Рівне : НУВГП, 2012. - 111 с.

35. Сорока Л.С. Основи теорії інформаці: *навчальний посібник* / Л.С. Сорока. - Х. : ХНУ ім. В.Н. Каразіна, 2007. –264 с.

36. Біометричні системи безпеки. *Навчальні матеріали:* веб-сайт. URL: https://uadoc.zavantag.com/ (дата звернення 16.05.2022).

37. Computer mouse movement patterns: A potential marker of mild cognitive impairment *Стаття:* веб-сайт. URL: https://www.sciencedirect.com/ science/article/pii/S2352872915000792 (дата звернення: 15.02.2021)

38. Ентропія як міра ступеня невизначеності. *Букліб:* веб-сайт. URL: https://buklib.net/books/23810/ (дата звернення 25.10.2022)

39. Николайчук Я.М. Теорія джерел інформації: *монографія* / Я.М. Николайчук. – Тернопіль: ТНЕУ, Економічна думка, 2008. – 396 с.

40. Олешко І.В. Порівняльний аналіз методів біометричної автентифікації на основі критерію відносної ентропії. *Вісник.* Автоматика вимірювання та керування. Харків: ХНУРЕ, 2015.

41. Biryukov, Alex and Khovratovich, Dmitry. Related-key Cryptanalysis of the Full AES-192 and AES-256. – *Advances in Cryptology* – ASIACRYPT 2009, 2009. – Vol. 5912.

42. Refresh access tokens: *Developer*: веб-сайт. URL: https://developer.okta.com/docs/guides/refresh-tokens/main/ (дата звернення: 20.10.2022).

43. Advanced Encryption Standard (AES) *Портал:* веб-сайт. URL: https://www.geeksforgeeks.org/advanced-encryption-standard-aes/ (дата звернення: 20.10.2022).

44. Опис прототипу *Studwood:* веб-сайт. URL: https://studwood.net/2354701/informatika/opis_prototipu (дата звернення: 20.10.2022).

45. Еволюція криптографічних алгоритмів шифрування *Isearch:* веб-сайт. URL: http://isearch.kiev.ua/uk/searchpractice/ (дата звернення: 20.10.2022).

46. Теорія інформації та кодування: Підруч. для студ. вищ. техн. навч. закл. / Ю. П. Жураковський, В. П. Полторак. – К. : «Вища шк.», 2001. – 255 с.

47. Теорія інформації та кодування: підручник / В. І. Барсов, В. А. Краснобаєв, З. В. Барсова, О. І. Тиртишніков, І. В. Авдєєв; ред.: В. І. Барсов; МОНМС України, Укр. інж.-пед. акад., Полтав. нац. техн. ун-т ім. Ю. Кондратюка. – Полтава, 2011. – 321 с.

48. Douglas-Packer algorithm. *Towardsdatascience:* веб-сайт. URL:https://towardsdatascience.com/ (дата звернення: 20.10.2022).

49. Вступ в C# *Programm:* веб-сайт. URL: https://programm.top/uk/c-sharp/tutorial/introduction/ (дата звернення: 20.10.2022).

50. Мова програмування C# *Навчальний портал:* веб-сайт. URL: http://www.znannya.org/?view=csharp (дата звернення: 20.10.2022).

51. Робота з Visual Studio. *Visual Studio*: веб-сайт. URL: https://visualstudio.microsoft.com (дата звернення: 22.10.2022).

52. Visual Studio Microsoft: веб-сайт. URL: https://visualstudio.microsoft.com (дата звернення: 22.10.2022).

53. Design Windows Forms. *Jetbrains:* веб-сайт. URL: https://www.jetbrains.com/help/rider/Working_with_Windows_Forms.html (дата звернення: 22.10.2022).

54. Windows Forms. *Docs:* веб-сайт. URL: https://docs.sentry.io/platforms/dotnet/ (дата звернення: 22.10.2022).

55. Unit testing. *Techtarget:* https://www.techtarget.com/searchsoftwarequality/definition/unit-testing (дата звернення: 22.10.2022).

56. FAR and FRR: security level versus user convenience. *Recogtech:* веб-сайт. URL: https://www.recogtech.com/en/knowledge-base/security-level-versus-user-convenience (дата звернення 21.10.2022).

57. Методичні вказівки до виконання економічної частини магістерських кваліфікаційних робіт / Уклад. : В. О. Козловський, О. Й. Лесько, В. В. Кавецький. – Вінниця : ВНТУ, 2021. – 42 с.

ДОДАТКИ

Додаток А. Технічне завдання

Вінницький національний технічний університет
Факультет менеджменту та інформаційної безпеки
Кафедра менеджменту та безпеки інформаційних систем

ЗАТВЕРДЖУЮ
Голова секції "Управління інформаційною безпекою" кафедри МБІС
д.т.н., професор
_____ Юрій ЯРЕМЧУК
"24" вересня 2022 р.

ТЕХНІЧНЕ ЗАВДАННЯ

до магістерської кваліфікаційної роботи на тему:

Підвищення достовірності автентифікації користувача на основі захищеного електронного ключа та аналізу ентропії рухів миші

08-72.МКР.001.00.000.ТЗ

Керівник магістерської кваліфікаційної роботи
д.ф., ст. викл. Ольга Салієва _____

Вінниця – 2022 р.

1. Найменування та область застосування

Підвищення достовірності автентифікації користувача на основі захищеного електронного ключа та аналізу ентропії рухів миші. Область застосування: виконувати процес двофакторної автентифікації користувачів у системі.

2.Підстава для розробки

Розробка виконується на основі наказу ректора ВНТУ №203 від 14. 09. 2022 р.

3. Мета та призначення розробки

3.1 Мета розробки: розробка вдосконаленого методу для підвищення достовірності автентифікації користувача на основі захищеного електронного ключа та аналізу ентропії рухів миші.

3.2 Призначення: розроблений програмний засіб виконує процес двофакторної автентифікації користувачів у системі.

4. Джерела розробки

4.1. Martin Fowler. Microservices. Стаття: веб-сайт. URL: https://martinfowler.com/articles/microservices.html (дата звернення: 12.10.2022).

4.2. AISO/IEC 27005:2005 «Інформаційні технологій. Методи захисту. Система управління інформаційною безпекою. Вимоги» Електрон. ресурс: веб-сайт. URL: http://www.dstszi.gov.ua/dstszi/control (дата звернення: 12.10.2022).

4.3. Самохвалов Ю.Я., Темпіков В.О., Хорошко В.О. Організаційно-технічне забезпечення захисту інформації: Навчальний посібник / За ред. проф. Хорошка В.О. – К.: НАУ, 2002. – С. 207

4.4. Security and Compliance: Customer Controls for Information Protection in Office 365. Microsoft: веб-сайт. URL: http://download.microsoft.com/download/ (дата звернення: 12.10.2022).

5. Вимоги до програми

5.1 Вимоги до функціональних характеристик:

5.1.1 Програмний засіб повинен мати зручний, легкий у використанні інтерфейс користувача;

5.1.2 Реалізація методу не повинна вимагати спеціальних ліцензійних програмних додатків;

5.1.3 Програмний засіб повинен виконувати процес двофакторної автентифікації користувачів у системі.

5.2 Вимоги до надійності:

5.2.1 Програмний засіб повинен працювати без помилок, у випадку виникнення критичних ситуацій необхідно передбачити виведення відповідних повідомлень;

5.2.2 Бази даних повинні бути налаштовані на автоматичне створення резервних копій;

5.2.3 Програмний засіб повинен виконувати свої функції.

5.3 Вимоги до складу і параметрів технічних засобів: процесор – Pentium 1500 МГц і подібні до них; оперативна пам’ять – не менше 512 Mb; середовище функціонування – операційна система сімейство Windows; вимоги до техніки безпеки при роботі з програмою повинні відповідати існуючим вимогам та стандартам з техніки безпеки при користуванні комп’ютерною технікою.

6. Вимоги до програмної документації

6.1 Обов’язкова поетапна інструкція для майбутніх користувачів, наведена у пункті 3.3

7. Вимоги до технічного захисту інформації

7.1 Необхідно забезпечити процес двофакторної автентифікації користувачів у системі.

7.2 Неможливість отримання доступу незареєстрованих користувачів до інформаційних ресурсів.

8. Техніко-економічні показники

8.1 Цінність результатів використання даного проекту повинна перевищувати витрати на його реалізацію.

8.2 Має бути реалізований таким чином, щоб підходити для використання широкого загалу.

9. Стадії та етапи розробки

№ з/п	Назва етапів магістерської кваліфікаційної роботи	Початок	Закінчення
1	Визначення напрямку магістерської роботи, формулювання теми	15.09.2021	30.09.2021
2	Аналіз предметної області обраної теми	01.10.2022	10.10.2022
3	Апробація отриманих результатів	11.10.2022	15.10.2022
4	Розробка алгоритму роботи	16.10.2022	31.10.2022
5	Написання магістерської роботи на основі розробленої теми	01.11.2022	15.11.2022
6	Розробка економічної частини	15.11.2022	23.11.2022
7	Передзахист магістерської кваліфікаційної роботи	24.11.2022	25.11.2022
8	Виправлення, уточнення, корегування магістерської кваліфікаційної роботи	26.11.2022	15.11.2022
9	Захист магістерської кваліфікаційної роботи	19.12.2022	21.12.2022

10. Порядок контролю та прийому

10.1 До приймання магістерської кваліфікаційної роботи надається:

– ПЗ до магістерської кваліфікаційної роботи;

– програмний додаток;

– презентація;

– відзив керівника роботи;

– відзив опонента

Технічне завдання до виконання прийняв _____ М.О. Берестенко

Додаток Б. Лістинг форм авторизації користувача

```
using Newtonsoft.Json;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.IO;
using System.Linq;
using System.Security.Cryptography;
using System.Text;
using System.Windows;
using System.Windows.Input;
using System.Windows.Shapes;
using walletMouseEntropy;
using Application = System.Windows.Application;
using MessageBox = System.Windows.MessageBox;
using OpenFileDialog = Microsoft.Win32.OpenFileDialog;
using Path = System.IO.Path;
using SystemColors = System.Windows.SystemColors;
using Timer = System.Timers.Timer;
using User = FaceRecognition.Models.User;

namespace FaceRecognition
{
   public partial class Login : Window, INotifyPropertyChanged
   {
      private System.Windows.Point lastPoint;
      private int Up = 0;
      private int Down = 0;
      private int Left = 0;
      private int Right = 0;
      private bool drawEnable = false;
      public Login()
      {
         InitializeComponent();
      }
      private int failCount = 3;
      public event PropertyChangedEventHandler PropertyChanged;
      private bool active = true;
      private bool enableAutoNavigate = true;
      private Timer captureTimer;
      public void OnLoad()
      {
      }
      private void Window_Loaded(object sender, RoutedEventArgs e)
      {
         OnLoad();
      }
```

```
        private void Button_Click_1(object sender, RoutedEventArgs e)
        {
            if (!VerifyData())
                return;

            var res = GetUsers();
            var eror = "";

            if (res == null)
            {
                eror = "\"Користувача не знайдено\"";

            }
            else
            {
                var f = lblFaceName.Content.ToString().ToLower();

                var user = res.FirstOrDefault(x => x.UserName == UserName.Text);
                if (user == null)
                {
                    eror = "\"Користувача не знайдено\"";
                }
                else
                {
                    if (user.IsAdmin || (user.UserId == Decript(FileName.Text) &&
MouseEntropyAnalyze(user)))
                    {
                        this.Hide();
                        Helper.AddLog("User is logged in", user.UserName);
                        CommonPage commonpage = new CommonPage();
                        commonpage.AuthUserName.Content = user.UserName;
                        if (user.IsAdmin || user.Role == "Адміністратор")
                        {
                            commonpage.UserTable.Visibility = Visibility.Visible;
                        }
                        else
                        {
                            commonpage.UserTable.Visibility = Visibility.Hidden;
                        }
                        commonpage.Registration.Visibility = commonpage.Login.Visibility =
Visibility.Hidden;
                        commonpage.Exit.Visibility = Visibility.Visible;
                        commonpage.UserInfo.Visibility = Visibility.Visible;
                        commonpage.history.Visibility = Visibility.Visible;
                        enableAutoNavigate = false;
                        MessageBox.Show("Спроба входу була вдалою!");
                        commonpage.ShowDialog();
```

```
                    this.Close();
                }

                else
                {
                    eror = $"Помилка входу";

                }
            }
        }

        if (!string.IsNullOrEmpty(eror))
        {
            --failCount;
            MessageBox.Show($"Невдала спроба ідентифікації. {eror}. Залишилося спроб -
{failCount}");
            if (failCount == 0)
            {
                this.Hide();
                Environment.Exit(1);
            }
            else
            {
                return;
            }
        }
    }
    private bool MouseEntropyAnalyze(User user)
    {
        var countDif = 30;
        var resCount = 1;
        if (Up - countDif >= user.MouseUp && user.MouseUp <= Up + countDif)
        {
            resCount += 1;
        }
        if (Down - countDif >= user.MouseDown && user.MouseDown <= Down + countDif)
        {
            resCount += 1;
        }
        if (Left - countDif >= user.MouseLeft && user.MouseLeft <= Left + countDif)
        {
            resCount += 1;
        }
        if (Right - countDif >= user.MouseRight && user.MouseRight <= Right + countDif)
        {
            resCount += 1;
        }
        return resCount >= 3;
    }
```

```
        private static string Decript(string path)
        {
            string encryptedResult = null;
            using (StreamReader r = new StreamReader(path))
            {
                encryptedResult = r.ReadToEnd();
            }
            byte[] bytesToBeDecrypted = Convert.FromBase64String(encryptedResult);
            byte[] passwordBytesdecrypt = Encoding.UTF8.GetBytes("Password");
            passwordBytesdecrypt = SHA256.Create().ComputeHash(passwordBytesdecrypt);
            byte[] bytesDecrypted = AES_Decrypt(bytesToBeDecrypted, passwordBytesdecrypt);
            return Encoding.UTF8.GetString(bytesDecrypted);
        }
        private static byte[] AES_Decrypt(byte[] bytesToBeDecrypted, byte[] passwordBytes)
        {
            byte[] decryptedBytes = null;
            byte[] saltBytes = new byte[] { 2, 1, 7, 3, 6, 4, 8, 5 };
            using (MemoryStream ms = new MemoryStream())
            {
                using (RijndaelManaged AES = new RijndaelManaged())
                {
                    AES.KeySize = 256;
                    AES.BlockSize = 128;
                    var key = new Rfc2898DeriveBytes(passwordBytes, saltBytes, 1000);
                    AES.Key = key.GetBytes(AES.KeySize / 8);
                    AES.IV = key.GetBytes(AES.BlockSize / 8);
                    AES.Mode = CipherMode.CBC;
                    using (var cs = new CryptoStream(ms, AES.CreateDecryptor(),
CryptoStreamMode.Write))
                    {
                        cs.Write(bytesToBeDecrypted, 0, bytesToBeDecrypted.Length);
                        cs.Close();
                    }
                    decryptedBytes = ms.ToArray();
                }
            }
            return decryptedBytes;
        }
        private bool VerifyData()
        {
            if (String.IsNullOrEmpty(UserName.Text))
            {
                MessageBox.Show("Введіть логін");
                return false;
            }
            return true;
        }
        private void newAccount_Click(object sender, RoutedEventArgs e)
        {
```

```
            active = false;
            var w = Application.Current.Windows[0];
            this.Hide();
            Registration wfAbout = new Registration();
            wfAbout.ShowDialog();
            this.Close();
        }
        private List<User> GetUsers()
        {
            List<User> users = new List<User>();
            var path = $"{Path.GetDirectoryName(AppDomain.CurrentDomain.BaseDirectory)}/ResourceFile.json";
            bool exists = System.IO.File.Exists(path);
            if (!exists)
                System.IO.File.Create(path);
            using (StreamReader r = new StreamReader(path))
            {
                string json = r.ReadToEnd();
                users = JsonConvert.DeserializeObject<List<User>>(json);
            }
            return users;
        }
        private void Login_OnClosed(object sender, EventArgs e)
        {
            captureTimer = null;
            active = false;
            if (enableAutoNavigate)
            {
                var w = Application.Current.Windows[0];
                this.Hide();
                CommonPage wfAbout = new CommonPage();
                wfAbout.ShowDialog();
                this.Close();
            }
        }
        private void Button_Click(object sender, RoutedEventArgs e)
        {       OpenFileDialog dlg = new OpenFileDialog();
            dlg.RestoreDirectory = true;
            if (dlg.ShowDialog() == true)
            {           string selectedFileName = dlg.FileName;
                FileName.Text = selectedFileName;
            }
        }
        private void Canvas_MouseDown_1(object sender, System.Windows.Input.MouseButtonEventArgs e)
        {
            if (e.ButtonState == MouseButtonState.Pressed)
                lastPoint = e.GetPosition(this);
        }
```

```
        private void Form1_MouseMove(object sender,
System.Windows.Input.MouseEventArgs e)
        {
            var points = e.GetPosition(this);
            if (e.LeftButton == MouseButtonState.Pressed && drawEnable)
            {
                if (lastPoint != null)
                {                       if (lastPoint.X < points.X)
                    {                   Right += 1;
                                        }
                                        if (lastPoint.X > points.X)
                    {                   Left += 1;
                                        }
                                    if (lastPoint.Y > points.Y)
                    {                   Up += 1;
                                        }
                                        if (lastPoint.Y < points.Y)
                    {
                        Down += 1;
                              }         }
                    Line line = new Line();
                    line.Stroke = SystemColors.WindowFrameBrush;
                    line.X1 = lastPoint.X;
                    line.Y1 = lastPoint.Y;
                    line.X2 = e.GetPosition(this).X;
                    line.Y2 = e.GetPosition(this).Y;

                    lastPoint = e.GetPosition(this);
                    myCanvas.Children.Add(line);
                }     }
            System.Windows.Forms.Timer timer = new System.Windows.Forms.Timer();
            private void Button_Click_2(object sender, RoutedEventArgs e)
            {
                drawEnable = true;
                StaerRecog.Content = "Аналіз ...";
                logButton.IsEnabled = false;
                timer.Interval = 10000;
                timer.Tick += timer_Tick;
                timer.Start();
            }      void tmer_Tick(object sender, System.EventArgs e)
            {          drawEnable = false;
                logButton.IsEnabled = true;
                timer.Stop();
                MessageBox.Show("Аналіз завершено та дані збережено!");
                StaerRecog.Content = "Розпочати знову?";
                myCanvas.Children.Clear();
            }
        }
    }
```

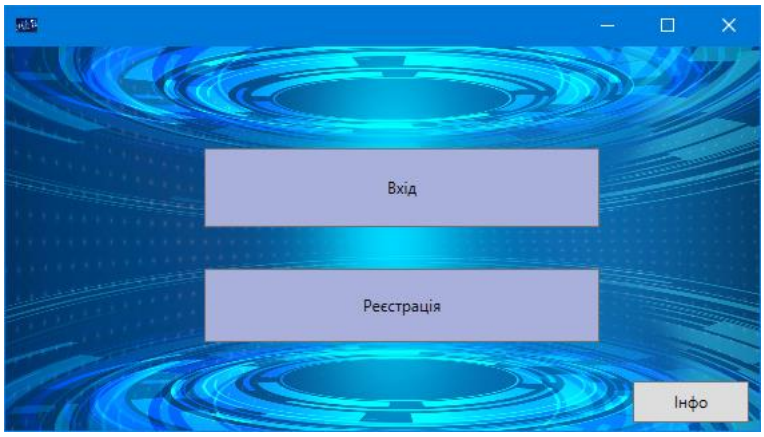
Додаток В. Лістинг аналізу ентропії та формування ключа

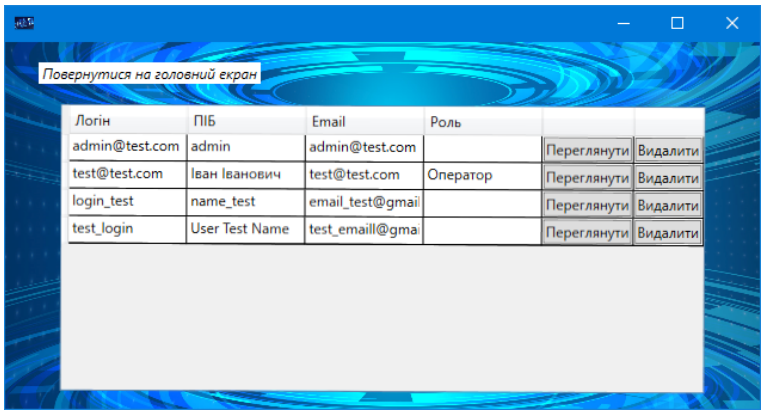
```
using FaceRecognition;
using Newtonsoft.Json;
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Security.Cryptography;
using System.Text;
using System.Windows;
using User = FaceRecognition.Models.User;

namespace walletMouseEntropy
{
   public partial class EditUser : Window
   {
      public EditUser()
      {
         InitializeComponent();
      }
      private void Window_Loaded(object sender, RoutedEventArgs e)
      {
      }
      private string role = "";
      private void Button_Click(object sender, RoutedEventArgs e)
      {
         var users = GetUsers();
         User user = users.FirstOrDefault(x => x.UserName == UserName.Text);
         if (user != null)
         {
            user.UserName = UserName.Text;
            user.FIO = FIO.Text;
            user.Email = Email.Text;
            user.Role = role;
         }
         else
         {
            users.Add(new User()
            {
               UserName = UserName.Text,
               FIO = FIO.Text,
               Email = Email.Text,
               Role = role
            });
         }
         UpdateUser(users);
         Helper.AddLog("Admin approve user", user.UserName);
```

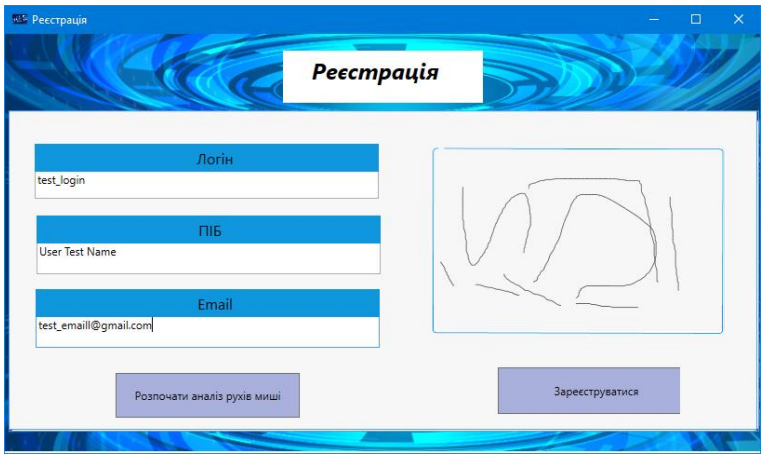
```
            this.Close();
        }
        private void UpdateUser(List<User> users)
        {
            string json = JsonConvert.SerializeObject(users);
            File.WriteAllText(
    $"{System.IO.Path.GetDirectoryName(System.AppDomain.CurrentDomain.BaseDirectory)}/
ResourceFile.json",
                json, Encoding.UTF8);
            var w = Application.Current.Windows[0];
            this.Hide();
            walletMouseEntropy.UserTable userTable = new walletMouseEntropy.UserTable();
            userTable.ShowDialog();
        }
        private List<User> GetUsers()
        {
            List<User> users = new List<User>();
            var path =
$"{System.IO.Path.GetDirectoryName(AppDomain.CurrentDomain.BaseDirectory)}/ResourceFile.js
on";

            using (var fs = new FileStream(path, FileMode.Open, FileAccess.Read,
FileShare.ReadWrite))
            using (var sr = new StreamReader(fs, Encoding.Default))
            {
                string json = sr.ReadToEnd();
                users = JsonConvert.DeserializeObject<List<User>>(json);
            }
            return users;
        }
        private void Button_Click_1(object sender, RoutedEventArgs e)
        {
            this.Close();
        }
        private void CheckBox_Checked_2(object sender, RoutedEventArgs e)
        {
            if (ad.IsChecked == true)
            {
                ad.IsChecked = true;
                upOper.IsChecked = false;
                oper.IsChecked = false;
                role = "Адміністратор";
            }
        }
        private void CheckBox_Checked(object sender, RoutedEventArgs e)
        {
            if (upOper.IsChecked == true)
            {
                ad.IsChecked = false;
```

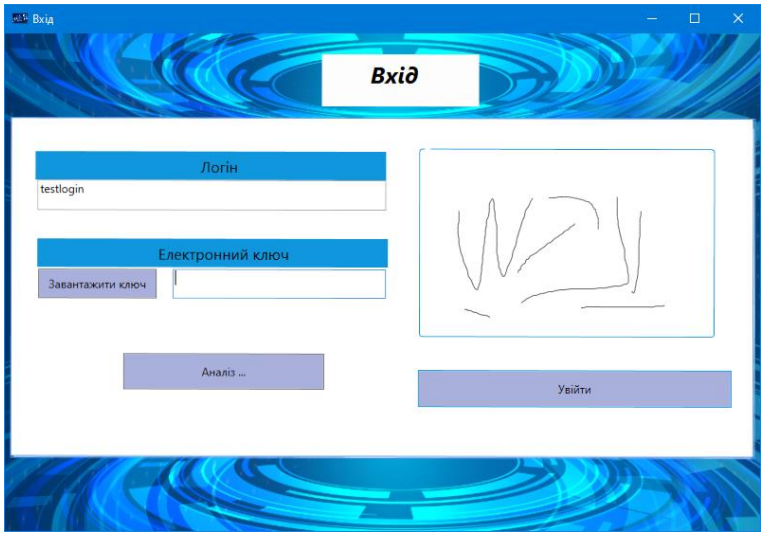
```
            upOper.IsChecked = true;
            oper.IsChecked = false;
            role = "Старший оператор";
        }
    }
    private void CheckBox_Checked_1(object sender, RoutedEventArgs e)
    {
        if (oper.IsChecked == true)
        {
            ad.IsChecked = false;
            upOper.IsChecked = false;
            oper.IsChecked = true;
            role = "Оператор";
        }
    }
    private void Button_Click_2(object sender, RoutedEventArgs e)
    {
        var users = GetUsers();
        User user = users.FirstOrDefault(x => x.UserName == UserName.Text);
        byte[] bytesToBeEncrypted = Encoding.UTF8.GetBytes(user.UserId);
        byte[] passwordBytes = Encoding.UTF8.GetBytes("Password");
        passwordBytes = SHA256.Create().ComputeHash(passwordBytes);
        byte[] bytesEncrypted = AES_Encrypt(bytesToBeEncrypted, passwordBytes);
        string encryptedResult = Convert.ToBase64String(bytesEncrypted);

File.WriteAllText($"{System.IO.Path.GetDirectoryName(AppDomain.CurrentDomain.BaseDirectory
)}/key__.txt", encryptedResult);
        MessageBox.Show("Ключ успішно збережено");
    }          public static byte[] AES_Encrypt(byte[] bytesToBeEncrypted, byte[]
passwordBytes)
    {       byte[] encryptedBytes = null;
           byte[] saltBytes = new byte[] { 2, 1, 7, 3, 6, 4, 8, 5 };
        using (MemoryStream ms = new MemoryStream())
        {          using (RijndaelManaged AES = new RijndaelManaged())
            {             AES.KeySize = 256;
                AES.BlockSize = 128;
                var key = new Rfc2898DeriveBytes(passwordBytes, saltBytes, 1000);
                AES.Key = key.GetBytes(AES.KeySize / 8);
                AES.IV = key.GetBytes(AES.BlockSize / 8);
                AES.Mode = CipherMode.CBC;
                using (var cs = new CryptoStream(ms, AES.CreateEncryptor(),
CryptoStreamMode.Write))
                {                cs.Write(bytesToBeEncrypted, 0, bytesToBeEncrypted.Length);
                    cs.Close();
                }             encryptedBytes = ms.ToArray();
            }       }
        return encryptedBytes;
    }   }
}
```

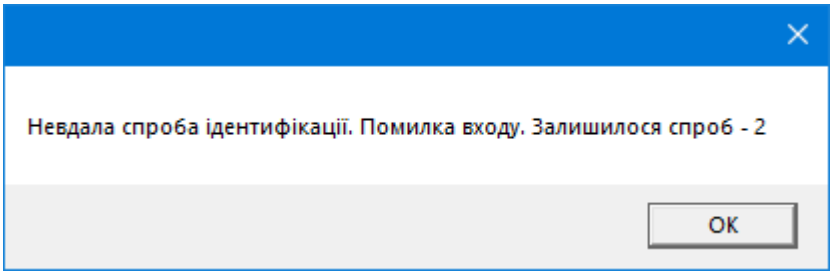
Додаток Г. Інтерфейс додатку

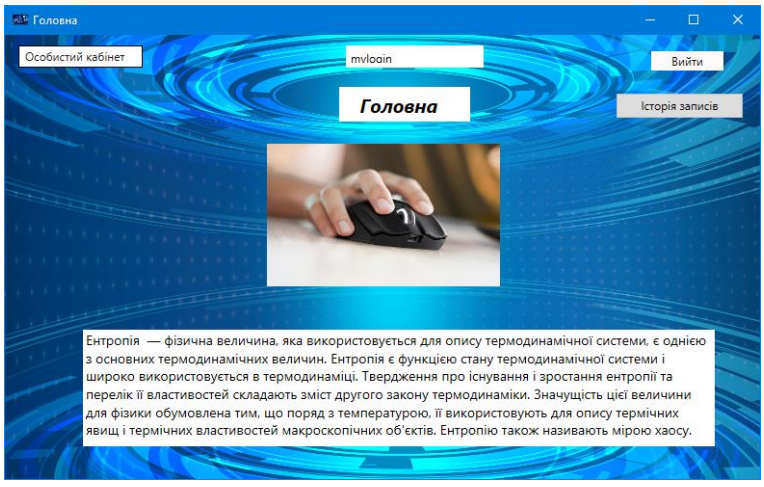





Реєстрація
Реєстрація
Логін
test_login
ПІБ
User Test Name
Email
test_emaill@gmail.com
Розпочати аналіз рухів миші
Зареєструватися


Вхід
Вхід
Логін
testlogin
Електронний ключ
Завантажити ключ
Аналіз ...
Увійти


Невдала спроба ідентифікації. Помилка входу. Залишилося спроб - 2
OK


Головна
Особистий кабінет
mvlogin
Вийти
Головна
Історія записів
Ентропія — фізична величина, яка використовується для опису термодинамічної системи, є однією з основних термодинамічних величин. Ентропія є функцією стану термодинамічної системи і широко використовується в термодинаміці. Твердження про існування і зростання ентропії та перелік її властивостей складають зміст другого закону термодинаміки. Значущість цієї величини для фізики обумовлена тим, що поряд з температурою, її використовують для опису термічних явищ і термічних властивостей макроскопічних об'єктів. Ентропію також називають мірою хаосу.

Додаток Д. Ілюстративний матеріал

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

Підвищення достовірності автентифікації користувача на основі захищеного електронного ключа та аналізу ентропії рухів миші

ВИКОНАВ: СТ. ГР. УБ-21М БЕРЕСТЕНКО М.О.

КЕРІВНИК: Д.Ф., СТ. ВИКЛ. САЛІЄВА О.В.

Вступ

- Актуальність роботи визначається тим, що проблеми пов'язані з розмежуванням доступу і автентифікацією користувачів є критичними і вдосконалення механізмів вирішення цих задач необхідне для забезпечення коректної роботи сучасних програмних додатків.
- Метою роботи є розробка вдосконаленого методу для підвищення достовірності автентифікації користувача на основі захищеного електронного ключа та аналізу ентропії рухів миші.
- Практична цінність. Розроблено програмний продукт, який реалізує удосконалений метод автентифікації користувача на основі захищеного електронного ключа та аналізу ентропії рухів миші.

Сучасні методи автентифікації

Процес реєстрації користувача в системі складається з трьох взаємопов'язаних процедур, що виконуються послідовно:

- ідентифікації;
- автентифікації;
- авторизації.

В сучасних умовах для підвищення достовірності автентифікації користувачів доцільно застосовувати двофакторну автентифікацію.

Аналоги додатків на основі двофакторної автентифікації

- Головним недоліком використання готових рішень є використання одного набору генерації для всіх підключень, а також можливість контролю та доступом до інформації фірмою розробником. Тому розробка програмного додатку з модифікованими засобами реалізації двофакторної автентифікації є актуальною в даний час.

Характеристика	Google Authenticator	Duo Mobile	Microsoft Authenticator	Free OTP	Authy
Незалежність	+	+	+	+	+
Потреба синхронізації	+	+	+	+	+
Автоматичне відновлення	–	+	+	+	–
Захист програми паролем	–	–	–	+	–
Введення даних з клавіатури	+	+	+	+	+
Застосування до інших систем	–	–	+	–	–
Необхідність реєстрації	+	+	+	+	+
Передача даних сервісу	+	+	+	+	+
Підтримка усіх ОС	+/–	+/–	+/–	+/–	+

Автентифікація користувачів на основі електронних ключів та аналізу ентропії рухів миші

Етапи аутентифікації на основі ключів

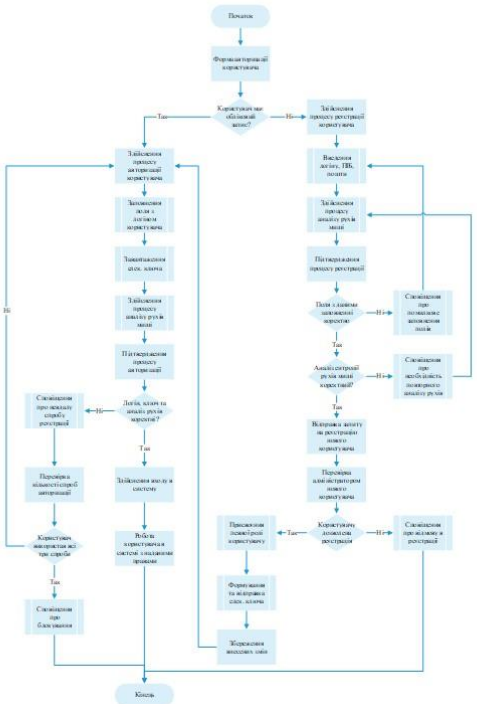
1. Початковий запит – користувач запитує доступ до захищеного ресурсу. Спочатку користувач повинен ідентифікувати себе способом, який не вимагає токена, наприклад, за допомогою імені користувача або пароля.
2. Верифікація – автентифікація визначає, що ідентифікаційні дані користувача є вірними, та перевіряє, які повноваження він має у запитаній системі.
3. Ключі – система випускає токен і передає його користувачеві. У разі апаратного ключа (токену) це має на увазі фізичну передачу токенів користувачеві. У випадку програмних токенів це відбувається у фоновому режимі, доки фонові процеси користувача обмінюються даними із сервером.
4. Збереження – ключ утримується користувачем, або фізично, або у браузері/ мобільному телефоні. Це дозволяє виконувати аутентифікацію без вказівки ідентифікаційних даних.

1) висування вимог
2) відносна ентропія біометричних характеристик;
3) Гаусова модель для обчислення біометричних характеристикі відносної ентропії;
4) методи регуляризації для вироджених характеристик;
5) методи регуляризації для неповних даних.

- *Алгоритм для обрахування відносної ентропії за біометричними характеристиками*

Удосконалення алгоритму автентифікації користувача

- Удосконалення алгоритму автентифікації користувача у захищеному додатку полягає у реалізації двофакторної автентифікації та застосуванні захищеного електронного ключа з використанням алгоритму AES та аналізу ентропії рухів миші, що розраховуються за певними параметрами і є індивідуальною характеристикою для кожного користувача.



- *Алгоритм роботи додатку на основі вдосконаленого методу*

Алгоритм формування захищеного електронного ключа та аналізу ентропії рухів миші

- Для аналізу ентропії рухів миші:
- розрахунок відстані здійснимо за формулою:

$$length\left((x_1, y_1), (x_2, y_2)\right) = \sqrt{(x_1 - x_2)^2 + (y_1 - by_2)^2},$$

- пропорційна близькість обчислюється за формулою:

$$proximity = 1 - length / max\,(length)$$

- показники горизонтальної, вертикальної та загальної швидкості переміщення курсору обчислимо за формулою:

$$V_{x_i} = \frac{\delta x_i}{\delta t_i}, \quad V_{y_i} = \frac{\delta y_i}{\delta t_i}, \quad V_i = \sqrt{V_{x_i}^2 + V_{y_i}^2}, \quad i = 2 \dots n,$$

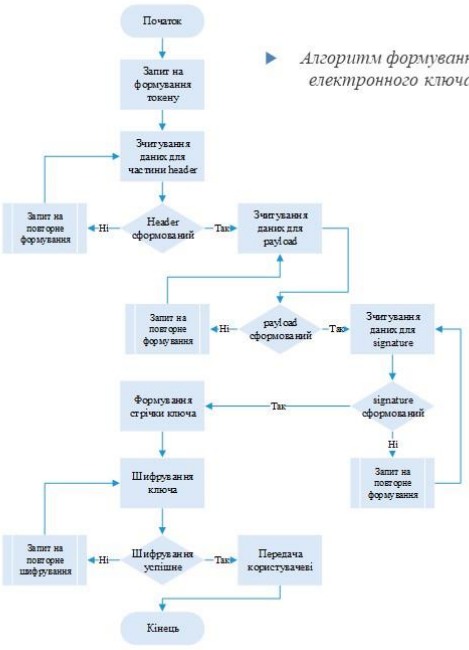
- показники прискорення, ривку та кутової швидкості обчислимо за формулами:

$$a_i = \frac{\delta V_i}{\delta t_i}, \quad j_i = \frac{\delta a_i}{\delta t_i}, \quad \omega_i = \frac{\delta \Theta_i}{\delta t_i}, \quad i - 2 \dots n,$$

- Ентропію обчислимо за наступною математичною формулою:

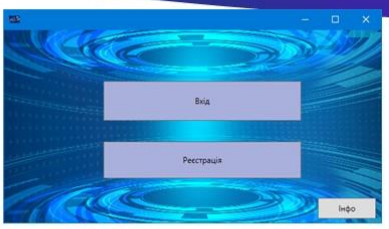
$$\Delta S = -In\frac{M_{m+1}}{M_{m+1}}$$

$$\Delta S = In[M_m] - In[M_{m+1}]$$



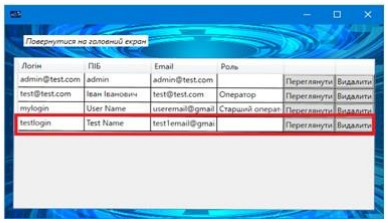
- *Алгоритм формування електронного ключа*

Тестування роботи додатку



- *Вигляд головного вікна програмного додатку*

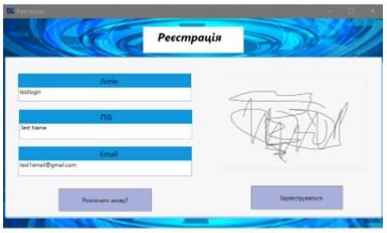
- *Вигляд вікна облікового запису адміністратора*



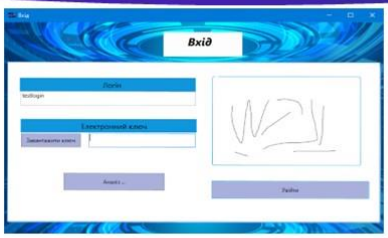
- *Вигляд вікна «Адміністрування користувачів»*

- *Вигляд вікна редагування даних користувача*

Авторизація та реєстрація

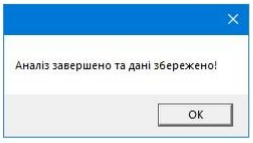


▶ *Вигляд вікна реєстрації користувача*

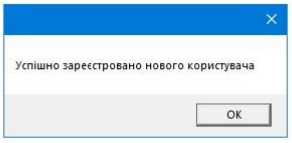


▶ *Вигляд вікна авторизації користувача*

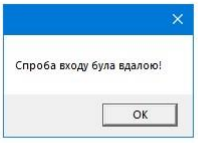
▶ *Вигляд вікна даних користувача*



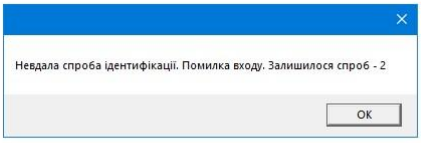
▶ *Вигляд сповіщення про успішний аналіз рухів миші*



▶ *Вигляд сповіщення про успішну реєстрацію*



▶ *Вигляд сповіщення про успішну спробу авторизації*

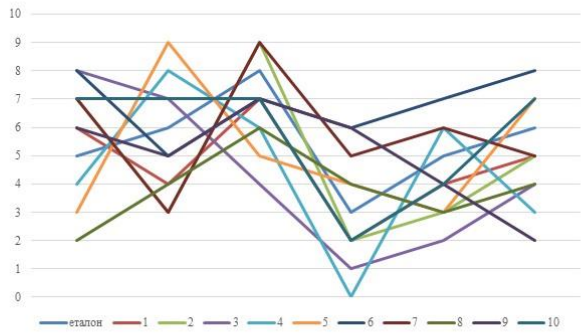


▶ *Вигляд сповіщення про некоректний вхід та обмеження спроб*

Аналіз отриманих результатів

Дослідимо шість показників за якими здійснюється аналіз:

- траєкторія руху комп'ютерної миші;
- відхилення руху комп'ютерної миші від зафіксованої траєкторії;
- швидкість руху комп'ютерної миші;
- прискорення руху комп'ютерної миші;
- нетипові рухи комп'ютерної миші;
- кліки комп'ютерної миші.

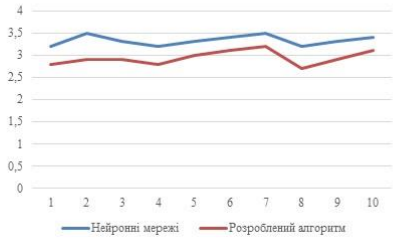


Аналіз отриманих результатів

Для оцінки запропонованого методу аналізу рухів миші застосуємо показники оцінки біометричної ідентифікації:

- $FAR = \frac{NFA}{NIVA}$, де FAR – ймовірність помилкового доступу;
- $FRR = \frac{NFF}{NEVA}$, де FRR – ймовірність помилкової відмови в доступі;

у порівнянні з розпізнаванням на основі нейронних мереж.



- *Показники ймовірності помилкового доступу*

- *Показники ймовірності помилкової відмови в доступі*

Висновки

- В магістерській кваліфікаційній роботі здійснювалась розробка в галузі підвищення достовірності автентифікації користувача на основі захищеного електронного ключа та аналізу ентропії рухів миші.
- Удосконалення алгоритму автентифікації користувача у захищеному додатку полягає у реалізації двофакторної автентифікації та застосуванні захищеного електронного ключа з використанням алгоритму AES та аналізу ентропії рухів миші, що розраховуються за певними параметрами і є індивідуальною характеристикою для кожного користувача.
- Враховуючи особливості поставлених задач роботи, для програмної реалізації, було обрано мову об'єктно-орієнтованого програмування C#, середовище програмування Visual Studio з використанням інтерфейсу Windows Forms.
- Проведений аналіз програмної розробки показав, що додаток працює коректно, unit-тести мають позитивний результат, що свідчить про відсутність помилок у написаному коді, стійкість до зламу електронного ключа зумовлена обраною технологією JWT, аналіз дослідження показників FAR та FRR показав підвищення достовірності розпізнавання на 11%.
- Економічний аналіз показників свідчить про доцільність впровадження розробленого додатку на основі двофакторної авторизації на практиці.
- Отже, за підсумками виконаних завдань та отриманих результатів можна вважати, що поставлена початкова мета роботи виконана, а розроблений додаток за результатами тестування можливий до застосування на практиці.

Дякую за увагу!

Додаток Е. Протокол перевірки на антиплагіат

ПРОТОКОЛ
ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ
НА НАЯВНІСТЬ ТЕКСТОВИХ
ЗАПОЗИЧЕНЬ

Назва роботи: Підвищення достовірності автентифікації користувача на основі захищеного електронного ключа та аналізу ентропії рухів миші

Тип роботи: магістерська кваліфікаційна робота
(БДР, МКР)

Підрозділ: Кафедра менеджменту та безпеки інформаційних систем
Факультет менеджменту та інформаційної безпеки
(кафедра, факультет)

Показники звіту подібності Unicheck

Оригінальність 97% Схожість 3%

Аналіз звіту подібності (відмітити потрібне):

1. Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату.

2. Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і відсутності самостійності її виконання автором. Роботу направити на розгляд експертної комісії кафедри.

3. Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень.

Особа, відповідальна за перевірку _____ Коваль Н.П.
(підпис) (прізвище, ініціали)

Ознайомлені з повним звітом подібності, який був згенерований системою Unicheck щодо роботи.

Автор роботи _____ Берестенко М.О.
(підпис) (прізвище, ініціали)

Керівник роботи _____ Салієва О.В.
(підпис) (прізвище, ініціали)