Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра захисту інформації

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему:
«МЕТОД ТА ЗАСІБ ПІДВИЩЕННЯ СТІЙКОСТІ ПАРОЛІВ»

Виконав: студент 2-го курсу, групи 1БС-20м
спеціальності 125 – Кібербезпека
(шифр і назва напряму підготовки, спеціальності)
_____ Кохан О.В.
(прізвище та ініціали)
Керівник к.т.н., доц. каф. ЗІ
_____ Баришев Ю.В.
(прізвище та ініціали)
Опонент: к.т.н., доц. каф . ОТ
_____ Савицька Л.А.
(прізвище та ініціали)
«____» _____ 2021 р.

Допущено до захисту
Завідувач кафедри
д. т. н., проф.
_____Лужецький В. А.
«___» _____2021 р.

Вінниця ВНТУ – 2021 рік

Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра захисту інформації
Освітньо-кваліфікаційний рівень магістр
Спеціальність 125 Кібербезпека
ОПП Безпека інформаційних і комунікаційних систем

ЗАТВЕРДЖУЮ
Завідувач кафедри ЗІ д.т.н., проф.
_____ В.А. Лужецький
_____ _____ 2021 року

ЗАВДАННЯ

НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

Кохану Олександру Володимировичу

1. Тема роботи: «Метод та засіб підвищення стійкості паролів»,
керівник роботи: Баришев Юрій Володимирович, к.т.н., доц. каф. ЗІ,
затверджена наказом ректора ВНТУ 24 вересня 2021 року №277.р.
2. Строк подання студентом роботи 23 грудня 2021 р.
3. Вихідні дані до роботи:
   - довжина згенерованих паролів – 65535 символів;
   - генерація на основі слова або словосполучення користувача;
   - мова програмування Java;
   - захист еталонних значень факторів автентифікації.
4. Зміст розрахунково-пояснювальної записки: Вступ. Аналіз методів автентифікації. Метод підвищення стійкості паролів. Алгоритм роботи засобу підвищення стійкості паролів. Розробка та тестування засобу підвищення стійкості паролів. Економічна частина. Висновки. Перелік використаних джерел. Додатки.
5. Перелік ілюстративного матеріалу.
Результат аналізу методів генерування паролів (плакат, А4). Математичний опис автентифікації користувача (плакат, А4). Структура засобу підвищення стійкості паролів (плакат, А4). Алгоритм роботи засобу підвищення стійкості паролів (плакат, А4). Таблиця замін символів (плакат, А4). Результати тестування (плакат А4).

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанти	Підпис, дата	
		завдання видав	завдання прийняв
1	Баришев Ю. В., к. т. н., доцент каф. ЗІ		
2	Баришев Ю. В., к. т. н., доцент каф. ЗІ		
3	Баришев Ю. В., к. т. н., доцент каф. ЗІ		
4	Баришев Ю. В., к. т. н., доцент каф. ЗІ		
5	Лесько О. Й., к.е.н., проф., зав. каф. ЕПВМ		

7. Дата видачі завдання _____ 2021 року

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів бакалаврської дипломної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз завдання. Вступ	01.09.2021 – 07.09.2021	
2	Розробка технічного завдання	08.09.2021 – 9.09.2021	
3	Аналіз літературних джерел за напрямком магістерської кваліфікаційної роботи	20.09.2021 – 26.09.2021	
4	Науково-технічне обгрунтування.	27.09.2021 – 4.10.2021	
5	Розробка алгоритму, програмна реалізація та тестування засобу підвищення стійкості паролів	05.10.2021 – 9.10.2021	
6	Аналіз ТЗ, висновки	30.10.2021 – 5.11.2021	
7	Оформлення пояснювальної записки	16.11.2021 – 0.11.2021	
8	Попередній захист МКР	03.12.2021	
9	Представлення МКР до захисту, рецензування	05.12.2021 – 20.12.2021	
10	Захист МКР	21.12.2021 – 23.12.2021	

Студент _____ О. В. Кохан

Керівник роботи _____ Ю.В. Баришев

АНОТАЦІЯ

УДК 004.056

Кохан О. В. Метод та засіб підвищення стійкості паролів. Магістерська кваліфікаційна робота зі спеціальності 125 – кібербезпека, освітня програма – Безпека інформаційних і комунікаційних систем. Вінниця: ВНТУ, 2021. 85 с.

Укр. мовою. Бібліогр.: 33 назв; рис.: 18; табл. 15.

Магістерська кваліфікаційна робота присвячена розробці методу та засобу підвищення стійкості паролів використання таблиці заміни символів.. Підготовлено науково-дослідне та техніко-економічне обґрунтування доцільності досліджень. У роботі здійснено аналіз існуючих методів автентифікації користувачів і обґрунтовано вибір для реалізації методу підвищення стійкості паролів. Розроблено математичну модель процесу генерування паролів. Розроблено програмний засіб, який дозволив протестувати запропонований метод.

Ілюстративна частина складається з 6 плакатів з демонстрацією результатів моделювання і проведених досліджень.

В економічному розділі оцінено витрати на розробку.

Ключові слова: генерування паролів, стійкий пароль, автентифікація, авторизація

ABSRACT

Kokhan O.V. Method and means of increasing password strength. Master's thesis in specialty 125 – cybersecurity. Vinnytsia: VNTU, 2021. – 85 p.

In Ukrainian language. Bibliographer: 33 titles; fig.: 18; tabl. 15.

The master's qualification work is devoted to developing a method and means to increase the stability of passwords using the table of symbol substitution. Prepared research and feasibility studies. The analysis of the existing user authentication methods is carried out in work, and the choice for the realization of the method of increased password stability is substantiated. A mathematical model of the password generation process has been developed. A software tool has been developed that allows testing the proposed method.

The graphic part consists of 6 posters demonstrating the results of modeling and research.

The economic section estimates the development costs.

Keywords: password generation, strong password, authentication, authorization

ЗМІСТ

ВСТУП

Разом із активним переходом людства до комп'ютерних технологій та перенесенням інформації в цифровий світ, зростає важливість питання обмеження доступу до персональних даних користувачів та варіанти захисту цих даних. Оскільки разом з розвитком інформаційних технологій, розвиваються і методи, що дозволяють отримати несанкціонований доступ до приватної інформації.

Одним з варіантів покращення захисту інформації користувача є використання стійких паролів в процесі їх авторизації, що дозволить збільшити час підбору пароля зловмисником та зменшить рентабельність зламу.

Однак для того щоб забезпечувати ефективний захист, пароль повинен відповідати певним критеріям[1, 2].

Для полегшення вибору парольної фрази, в користувачів є можливість використовувати спеціальні сервіси, а саме – генератори паролів.

Такі генератори допомагають створити справді безпечну та унікальну комбінацію для всіх типів облікових записів з різним рівнем секретності даних. У більшості служб є можливість налаштувати додаткові функції під час процесу генерації, наприклад, додати великі літери та спеціальні символи чи числа. Також можна зазначити рівень складності пароля - простий, середній і складний. Відповідно результатом буде стійкий пароль, що буде містити набір випадкових символів з максимальною ентропією. Це дозволить унеможливити атаки за словником і значно ускладнити підбір пароля грубою силою.

Проте чим більша логічна незалежність символів паролю один від одного і їх випадковість, тим важче користувачам запам'ятовувати і використовувати таке ключове слово. В результаті чого надається перевага поширеним паролям, що містять очевидний вміст (дату народження, імена членів сім'ї). Відповідно виникає необхідність поєднання цих двох факторів, щоб знайти середнє значення та дати можливість користувачам генерувати стійких пароль з можливістю запам'ятати його.

Об’єктом магістерської кваліфікаційної роботи є процес автентифікації користувачів в цифровому середовищі.

Предметом магістерської кваліфікаційної роботи є методи автентифікації користувачів в системі.

Метою дипломної роботи є удосконалення методів генерування стійких паролів.

Для досягнення мети необхідно виконати наступні завдання:

- проаналізувати методи автентифікації користувачів в інформаційних системах;
- проаналізувати існуючі методи підвищення стійкості паролів;
- провести дослідження для визначення методів розмиття паролів;
- розробити метод генерування стійких паролів;
- розробити алгоритм засобу генерування паролів;
- розробити та протестувати засіб генерування стійких паролів.

Наукова новизна магістерської кваліфікаційної роботи полягає в тому, що удосконалено метод генерування стійких паролів, який на відміну від відомих базується на використанні декількох методів розмивання парольної фрази, згенерованої користувачем, що дозволяє підвищити стійкість паролів зі збереженням комфорту щодо запам’ятовуваності для користувачів.

Практична цінність: засіб підвищення стійкості паролів.

Результати магістерської роботи доповідалися на таких конференціях:

XLIX регіональна н. т. к. професорсько-викладацького складу, співробітників та студентів університету з участю працівників науково-дослідних організацій та інженерно-технічних працівників підприємств м. Вінниці та області відбулась 18–29 травня 2020 р.

В результаті виконання магістерської кваліфікаційної роботи розроблено метод генерування паролів, що на відміну від інших дає можливість створювати паролів на основі ключового слова користувача для кращої запам’ятовуваності ключової фрази, з використанням різних методів розмивання паролів.

1 АНАЛІЗ ЛІТЕРАТУРНИХ ДЖЕРЕЛ

1.1 Аналіз методів автентифікації

Автентифікація - це процес перевірки ідентичності користувача. Користувачі ідентифікуються за допомогою різних механізмів автентифікації. У системі безпеки процес автентифікації перевіряє інформацію, надану користувачем з базою даних. Якщо інформація збігається з інформацією бази даних, користувачеві надається доступ до системи безпеки. Використовуються три типи механізму автентифікації [1].

Перевірка — це початкова фаза контролю доступу, і для перевірки використовуються три звичайні змінні — те, що користувач знає, те, що у користувача є, і те, що користувач є.

Знання користувачем чого небудь – коли система вимагає від особи ввести ім’я користувача та пароль, для надання доступу до цієї системи. Те, що користувач має – певна фізична цінність, що користувач має, наприклад, смарт-картку для автентифікації. Чим користувач є – певні фізіологічні особливості, такі як відбитки пальців, голос.

Усі типи механізмів автентифікації дозволяють користувачеві отримати доступ до системи, але всі вони працюють по-різному. Існує багато методів автентифікації, розроблених для користувачів, щоб отримати доступ до системи. У автентифікації паролем є дві форми – автентифікація зі слабким паролем і надійна автентифікація пароля.

Кожен контроль доступу має три процеси – ідентифікацію, автентифікацію, авторизацію [1]. Ідентифікація відбувається, коли користувач вводить ідентифікатор і ідентифікатор перевіряється системою безпеки. Деякі системи безпеки генерують випадкові ідентифікатори для захисту від зловмисників. Існує три процеси автентифікації. Авторизація – це перевірка та співставлення автентифікованого об’єкта інформації з рівнем доступу. Процес авторизації здійснюється трьома способами: авторизація виконується для автентифікованого користувача, авторизація виконується для членів групи,

авторизація виконується в кількох системах, а підзвітність — це процес, який веде системні журнали. Надалі буде розглянуто види автентифікації.

Автентифікація паролем вимагає від заявника згадати те, що він знає. Спочатку заявник вводить ім'я користувача, а по-друге, пароль. Пароль - це секретна комбінація букв, цифр та символів.

Однією з переваг є те, що довший пароль дуже важко зламати. Під час використання паролів необхідно використовувати надійні паролі. Міцний секретний ключ містить великі літери, нижній регістр, цифри та унікальні символи. Тепер адміністратори безпеки рекомендують паролі з 12 символів. Пароль із 12 символів із потужністю 94 та ентропією 78,7 біт знадобиться 55 днів, щоб зламати за допомогою суперкомп’ютерів. А за допомогою ПК для зламу знадобиться 3018 років [2].

Викриття паролю є найбільшою проблемою починаючи з моменту, коли користувач вводить пароль. Зловмисник може дізнатися пароль на різних етапах спілкування. Навіть якщо пароль надійний, він може бути легко відомий зловмиснику. Ключова проблема з іменем користувача та паролем, людський фактор:

- пароль легко вгадати, якщо він простий і його просто запам’ятати
- пароль можна дізнатися, якщо користувач його записав
- користувачі можуть випадково поділитися паролем
- користувач може забути складний пароль

З більш слабким паролем зловмисники зможуть легко зламати систему за допомогою методу грубої сили. Більшість засобів контролю доступу має пароль із восьми символів.

Є три фактори, які визначають міцність пароля – довжина, потужність та ентропія. Число 94 означає, що пароль був створений із пулу з 94 символів, включаючи верхній регістр, нижній регістр, цифри та спеціальні символи. Ентропія — це обчислена міцність пароля в бітах, тобто на скільки незалежними і неочевидними будуть набори символів у парольній фразі. Наприклад, пароль із восьми символів і потужністю 94 еквівалентний ентропії 52,4 біта. Звичайний

ПК зможе зламати пароль із 94-ма численністю за 20 хвилин за допомогою грубої сили. Використовуючи суперкомп’ютери, для зламу знадобиться 0,07 секунди. Отже, ентропія 52,4 біта або довжина 8 символів є слабким паролем [2]. Ще одним недоліком є соціальна інженерія, оскільки вона заманює користувачів на інший сайт, де зловмисники збирають особисту інформацію, включаючи паролі та ім’я користувача.

Автентифікація смарт-картки [2] є фактором, «що є у користувача». Смарт-картка – це картка розміром з кредитну картку, яка має вбудований сертифікат, який використовується для ідентифікації власника. Користувач може вставити картку в пристрій зчитування смарт-карт для автентифікації особи, смарт-картки зазвичай використовуються з PIN-кодом, що забезпечує багатофакторну автентифікацію. Іншими словами, користувач повинен щось мати (смарт-карту) і щось знати (PIN-код).

Однією з переваг смарт-карти є те, що вона поставляється в двох різновидах. По-перше, поставляється з картою пам'яті з даними, що забезпечує двофакторну автентифікацію. По-друге, оснащена мікропроцесором, що робить більш надійною двофакторну автентифікацію.

Смарт-карта з мікропроцесором зберігає сертифікати відкритого та приватного ключа. Смарт-картка блокується, якщо PIN-код введено неправильно після кількох спроб. Смарт-карта запобігає атак на словники. Вона портативний і її легко носити з собою користувачі.

Деякі причини автентифікації смарт-карткою полягають у тому, що деяким користувачам важко запам'ятати PIN-код, і вони вводять PIN-код на зворотному боці картки. Якщо картку вкрадено, її можна легко атакувати. Смарт-карту можна заблокувати після певної кількості неправильних спроб. Оскільки вона портативна, її також можна вкрасти. Деякі користувачі, які часто здійснюють покупки в Інтернеті, можуть стати жертвами фішингу. Іноді PIN-код можна дізнатися за допомогою плечового серфінгу.

Рішення вищевказаної проблеми полягає в тому, що картку потрібно тримати під рукою та не ділитися з іншими. Приховувати документи або

клавіатуру від поля зору, використовуючи тіло або тримаючи руку. Найкращий спосіб уникнути фішингу — не розкривати секрети [3]. Іншим рішенням може бути поєднання смарт-карт із RFID (радіочастотна ідентифікація). Крім того, комбінація двофакторної автентифікації (смарт-карти та біометричні дані) підвищує безпеку. Кількість спроб введення PIN-коду має бути обмежена.

Біометричні методи забезпечують автентифікацію на основі параметрів або властивостей тіла користувача. Біометрична автентифікація користувача – це метод, який ідентифікує користувача та/або перевіряє його особу на основі вимірювання його унікальних фізіологічних рис або поведінкових характеристик. Фізіологічна біометрика – це відбитки пальців, розпізнавання обличчя, райдужна оболонка ока, геометрія кисті, сканування сітківки ока.» Поведінкова біометрія – це розпізнавання голосу, ходи, сканування натискання клавіш і сканування підпису [4]. Відбитки пальців і рук є найпоширенішим біометричним методом, який використовується сьогодні. Багато ноутбуків оснащені зчитувачами відбитків пальців, а зчитувачі відбитків пальців також доступні на USB-флеш-накопичувачах.

Біометрична автентифікація широко використовуються і мають велику силу:

- позбавляє користувача від необхідності запоминання паролів
- біометрія унікальна і проста.
- дуже важко відтворити біометричні дані
- біометричні характеристики не можна втратити
- сканування відбитків пальців недороге [4].

Сканування відбитків пальців надійне, оскільки важко вгадати малюнок відбитків пальців. Хоча біометричні дані забезпечують найнадійнішу автентифікацію, вони схильні до помилок. Помилка першого роду виникає, коли система помилково відхиляє відомого користувача і вказує, що користувач невідомий. Помилкова другого роду виникає, коли система помилково ідентифікує невідомого користувача як відомого користувача. Біометричні системи, як правило, можна налаштувати на чутливість, але чутливість впливає

на точність». Інша проблема полягає в відсутності реального стандарту через специфічні формати постачальника. Також існують проблеми з прийняттям користувачами, оскільки користувач може відчувати себе злочинним, коли знімає сканування відбитків пальців. Крім того, травми на пальцях можуть заважати процесу сканування.

Автентифікація цифровим сертифікатом. Цифровий сертифікат — це технологія шифрування, яка працює подібно до Інтернет-версії паспорта [5]. Використовуючи інформацію відкритого та закритого ключа, цифрові сертифікати по суті гарантують одержувачу повідомлення, що повідомлення надходить від конкретної особи. Цифровий сертифікат підтверджує особу відправника, щоб забезпечити безпечніший зв’язок і запобігти шахрайству в Інтернеті. Найбільшими перевагами автентифікації на основі цифрових сертифікатів є конфіденційність. Шифруючи ваші комунікації — електронну пошту, логіни або транзакції онлайн-банкінгу — цифрові сертифікати захищають приватні дані та запобігають видимій інформації стороннім оком. Системи цифрових сертифікатів також зручні для користувачів, зазвичай вони працюють автоматично і вимагають мінімальних дій або участі відправника або одержувача.

Органи, які видають цифровий сертифікат, піддаються атаці зловмисників, і інформація про сертифікат змінюється. Зловмисники створюють фішинговий сайт і надсилають електронні листи та веб-сайти, які виглядають як оригінальні та проходять перевірку.

Найкращим рішенням є те, що центри цифрової сертифікації повинні оновити своє програмне забезпечення, щоб мінімізувати загрозу безпеці. Також розміщення цифрового сертифіката на токені забезпечить більш надійний захист.

Одноразовий пароль [6]. У роботі системи одноразових паролів OTP є дві системи. Генератор повинен створити відповідний одноразовий пароль із секретної парольної фрази користувача та інформації, наданої у запиті від сервера. Сервер повинен надіслати запит, який містить відповідні параметри

генерації, до генератора, повинен перевірити отриманий одноразовий пароль, повинен зберегти останній дійсний одноразовий пароль, який він отримав, і повинен зберегти відповідний порядковий номер одноразового пароля. Сервер також повинен сприяти зміні секретної парольної фрази користувача в безпечний спосіб.

Системний генератор OTP передає секретну парольну фразу користувача разом із початковим кодом, отриманим від сервера як частину завдання, через кілька ітерацій безпечної хеш-функції для створення одноразового пароля. Після кожної успішної автентифікації кількість ітерацій безпечної хеш-функції зменшується на одиницю. Таким чином генерується унікальна послідовність паролів. Сервер перевіряє одноразовий пароль, отриманий від генератора, обчислюючи безпечну хеш-функцію один раз і порівнюючи результат з раніше прийнятим одноразовим паролем.

Отже, хоча існує багато методів автентифікації, найпоширенішим та найзручнішим являється метод парольної автентифікації [7]. Паролі давно вбудовані в операційні системи й інші сервіси. При правильному використанні паролі можуть забезпечити прийнятний для багатьох систем рівень безпеки. Саме тому питання підвищення стійкості паролів і зручності їх використання досі актуальне.

1.2 Аналіз методів зламу паролів

Типовий спосіб зламу паролів - це отримати файл, що містить гешовані паролі користувачів, а потім запустити процес розшифрування для файлу, щоб спробувати отримати відповідності для всіх гешів, таким чином отримуючи всі паролі у файлі. Хоча остання частина, як правило, швидка, перша може бути дуже складною, і може знадобитися багато підходів, щоб проникнути в систему безпеки та отримати файл паролів. Однак за допомогою простих цільових пошуків у Google стало легше збирати незахищені геші користувачів.

Оскільки, як правило, пароль, який перевіряється у файлі паролів, має виконуватися за одним і тим же алгоритмом гешування, спосіб зробити геші

більш безпечними – це використовувати алгоритм гешування, який займає більшу кількість часу обчислень, наприклад SHA-512. Таким чином, кожне вгадування пароля в програмі автоматичного злому пароля буде проходити через геш-алгоритм для перевірки. Поширеною технікою злому паролів є створення всіх гешів, які потрібно перевірити завчасно. Таким чином, все, що потрібно зробити, це порівняти всі геші у файлі паролів з тими, які він уже згенерував. Оскільки геші вже обчислені, час, необхідний для проходження пароля через геш, не має значення і робить безкорисною міцність самого гешу [8].

Звичайний спосіб обійти таку атаку використовувати щось, що називається сіль. Сіль — це рандомізований рядок, що додається в кінці пароля користувача і потім гешується разом. Окрім надання більш безпечного пароля, солі також гарантують, що геш, створений для двох користувачів із однаковим паролем, буде різним. Наприклад, пароль, як-от "passwd", можна гешувати разом із "QxLUF1bgIAdeQX", об'єднаним з ним для одного користувача, і "bv5PehSMfV11Cd" для іншого. Це дасть геші MD5: 7bc4372cb5ca16d37bf8d688d82a19b1 та 436b51ce8857e7487eedb9998b4ff50 відповідно.

Для перевірки пароля системою знадобиться сіль, що зазвичай зберігається в базі даних облікових записів користувачів або як частина самого рядка гешування [8].

Проблема солей полягає в тому, що вони можуть бути використані повторно або занадто короткими. При повторному використанні, зловмисник може просто застосувати сіль до кожного вгадування пароля, перш ніж він його гешує. Щоб вирішити цю проблему, рекомендується створити хорошу сіль за допомогою криптографічно захищеного генератора псевдовипадкових чисел. Він схожий на генератор псевдовипадкових чисел, за винятком того, що він криптографічно захищений і набагато більш випадковий [8].

Як правило, злом паролів витягує паролі з файлів словників, або генерує їх, а потім гешує їх або шукає геші в таблиці та порівнює їх.

Далі будуть розглянуті деякі з найбільш популярних інструментів для злому паролів, які сьогодні використовують як зловмисники, так і системні адміністратори для перевірки безпеки своєї системи.

JohnTheRipper. Це швидка та популярна локальна програма для аудиту паролів від Openwall. Вона була створена з метою виявлення слабких паролів UNIX, але також може легко зламати слабкі геші Windows LM [9].

LOphtCrack. Це універсальний інструмент для аудиту Windows, який може атакувати робочі станції Windows і сервери Windows. Програма перевіряє стійкість паролів за допомогою словника, грубої сили та попередньо обчислених гешів. Відомо, що він надзвичайно швидкий і має простий у використанні інтерфейс користувача [10].

Aircrack-NG. Це унікальний інструмент, оскільки він призначений для перевірки іншого типу паролів, паролів Wi-Fi. Він може аналізувати паролі WEP або WPA [11]. Він аналізує зашифровані пакети, захоплені через бездротову мережу, і намагається підібрати пароль за допомогою свого алгоритму злому[11]. Додаток використовує атаку FMS (Fluhrer, Mantin і Shamir), яка є особливим типом атаки криптоаналізу. FMS дозволяє зловмиснику відновити ключ після надсилання великої кількості пакетів [12].

Отже, розглянуто методи зламу паролів такі як використання попередньо обчислених гешів, атаки за словником та атака грубої сили. Для захисті від двох останніх необхідно використовувати спеціальні символи, цифри та літери різних регістрів, що унеможливить атаку за словником. А використання таких паролів рекомендованою довжиною від 10 символів значно збільшить час атаки грубої сили.

1.3 Аналіз відомих методів генерування паролів

Для того щоб забезпечувати ефективний захист, пароль повинен відповідати певним критеріям[13, 14].

Стійких пароль – це головний бар'єр, який заважає зламати більшість ваших облікових записів у мережі. Якщо користувач не користується сучасними

методиками створення паролів, цілком можливо, що шахраї зможуть підібрати їх буквально за кілька годин. Щоб не наражати себе на ризик крадіжки даних і не стати жертвою вимагання, потрібно створювати паролі, які можуть протистояти зусиллям хакерів, озброєних сучасними засобами злому.

До створення пароля необхідно підійти відповідально залежно від виду та важливості ресурсу, де він використовуватиметься. При створенні складного пароля варто дотримуватись наступних правил [14]:

- довжина пароля. Пароль повинен містити щонайменше 8 символів, а краще – 10 і більше.
- наявність цифр і букв верхнього та нижнього регістрів, що йдуть не поспіль – GUrhFn.
- наявність спеціальних знаків - "!", "$", "@" і тому подібні (відповідно, якщо допустима їхня присутність).

Незважаючи на те, що деякі ресурси при реєстрації примусово змушують придумати складний пароль (вимагаючи певну довжину пароля, наявність цифр і літер, а також спеціальних символів), користувачі просто намагаються виконати ці вимоги, не замислюючись про надійність такого пароля.

В результаті виходить щось схоже на P@ssword123. Використання паролів типу «qwerty12345», «abc12345», «P@ssword1234» не гарантує надійного захисту даних, оскільки програми підбору паролів, якими користуються зловмисники, насамперед перевіряють саме такі паролі.

Надійний пароль не повинен містити імена, прізвиська тварин або назви міст, а також цифри дати народження або номера телефонів. Пароль типу «M@sha1990» буде вгадано програмами з підбору паролів, оскільки містить досить поширену комбінацію букв.

Для кожного ресурсу потрібно мати свій пароль. Це обумовлено необхідністю захисту всіх інших ресурсів користувача при компрометації пароля одного ресурсу, оскільки зараз багато ресурсів між собою пов'язані, за допомогою з'єднання поштових сайтів і соціальних мереж.

Пароль повинен бути відомий лише користувачеві, інакше ніякого надійного захисту вже бути не може. Небажано записувати паролі на папері і тим паче тримати ці записи поруч із комп'ютером у відкритому доступі всім.

Крім того, не рекомендується вводити свої паролі на сайтах з перевірки надійності паролів, оскільки вони можуть виявитися пастками для паролів.

Для вирішення такої проблеми існують сервіси, які за запитом генерують максимально складну комбінацію. Її не обов'язково запам’ятовувати, для зберігання паролів використовують спеціальні менеджери паролів [15].

Однак при виборі менеджера варто враховувати його особливості та призначення. Наприклад, вбудовані в браузер менеджери зручні, зрозумілі більшості користувачів, але поки все ще програють комерційним аналогам по здатності генерувати складні паролі, не скрізь присутня двофакторна автентифікація, та й немає можливості перемикатися між браузерами. Через це більшість експертів схиляються до того, щоб розглядати такі менеджери швидше як розширення браузерів і не зберігати в подібного роду додатках важливі дані, наприклад для банківських сервісів [16].

Генератори паролів допомагають створити дійсно надійну і унікальну комбінацію для будь-якого типу облікових записів. У більшості сервісів в процесі генерації можна налаштувати додаткові функції, наприклад, додавання букв верхнього регістру, спеціальних символів. чисел. Також можна вказати ступінь складності пароля - простий, середній і складний (рис. 1.1).

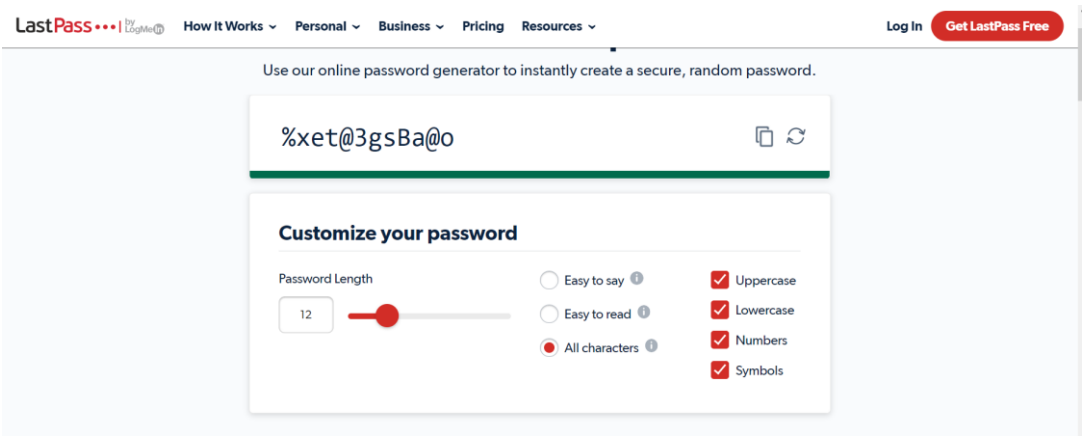


Рисунок 1.1 – Приклад онлайн генератора паролів

Проте використання менеджерів паролів несе додаткові труднощі. При втраті даних такого застосунку можуть бути втрачені доступи до облікових записів.

Існують різні методи генерування паролів (табл. 1.1) [17], проте всі вони відомі і мають ряд вразливостей, що дає можливість з легкістю підбирати пароль раніше перерахованими програмами, які можуть використовувати такі ж методи для підбору ключового слова.

Таблиця 1.1 – Методи генерування паролів

Назва методу	Суть методу	Характеристика
Спосіб заміни слів	базується на написанні паролю літерами з іншої розкладки клавіатури	нестійкий, оскільки більшість часто вживаних слів на певній мові використовуються для створення словників паролів
Спосіб випадкового натискання клавіш	базується на натисканні клавіш у певному порядку, не відриваючи руки	нестійкий, оскільки словники із «змієподібними» комбінаціями паролів (введеними затиснутим пальцем за певним маршрутом на клавіатурі) також існують
Спосіб «обчислюваних людиною» паролів	базується на формуванні матриці з символами, наприклад назви ресурсу, та подальшої заміни обраних символів, за певним правилом	може бути зручним для використання захисту онлайн-ресурсів, але генерування паролю для бездротової мережі з 63 символів таким способом не дасть необхідної стійкості та займе багато часу
Спосіб генерування паролів завдяки асоціаціям	базується на застосуванні технології асоціацій та замін певних літер символами	може бути легшим для запам’ятовування, але також не несе за собою необхідної стійкості
Спосіб створення паролів з декількох слів	базується на логічному поєднанні слів із зміною певних символів	може дати більш складний пароль, але його стійкість, також не можна вважати високою, а час на генерування може бути великим
Спосіб створення паролів завдяки генеруванню випадкових чисел	базується на використанні гральних кубиків для генеруванні випадкових чисел та подальшого вибору слів за таблицею	може займати менше часу для генерування паролю, але стійкість створеного паролю важко вважати високою, оскільки можливе використання словників для підбору

Продовження таблиці 1.1

Назва методу	Суть методу	Характеристика
Спосіб генерування випадкових паролів	базується на генеруванні паролів завдяки спеціальним командам та додаткам в операційній системі	може дати складніший результат та потребує набагато меншу часу для створення, але питання випадкового генерування такого ж самого паролю зловмисником, хоча й малоймовірне, але все ж таки можливе

Як показує досвід користувачі надають перевагу більш простим парольним фразам, які легко запам’ятати і які частіше всього пов’язані з особистою інформацією користувачів. Оскільки є можливість дізнатися достатньо велику кількість інформації про кожну людину в мережі інтернет, що відповідно значно облегшує шахраям отримати доступ до облікових записів.

Тому виникає необхідність покращення генераторів паролів, що дозволить користувачам не лише генерувати безпечні паролів, а й робити це на основі ключового слова. Це дозволить покращити захист облікового запису, та генерувати надійні паролі, які будуть зручними для запам’ятовування. А варіативність паролів дасть змогу обрати найзручнішу парольну фразу для кожного користувача.

Метод генерування паролів завдяки асоціаціям може бути легшим для запам’ятовування, але також не несе за собою необхідної стійкості. Для підвищення критерія стійкості буде запропоновано покращення даного методу з використанням генератора псевдовипадкових чисел, що дозволить зберегти суть даного підходу, унеможливити атаки за словником та ускладнити підбір ключового слова методом грубої сили.

1.4 Генератор псевдовипадкових чисел

Генератор псевдовипадкових чисел лежить в основі блоку генерування паролів. Тому необхідним є вибір генератора, який забезпечить відносно високу нелінійність та рівномірність при генеруванні чисел.

Найвідомішими генераторами ПВЧ є генератор на основі чисел Фібоначчі, лінійний та квадратичний конгруентні.

Методі лінійного конгруентного генератора полягає в обчислення послідовності випадкових чисел рекурентним співвідношенням [18]:

$$x_{i+1} = (ax_i + c) \bmod N \tag{1.1}$$

Квадратичний конгруентний генератор чисел описується наступною формулою:

$$x_{i+1} = (dx_i^2 + ax_i + c) \bmod N \tag{1.2}$$

Генератор на основі чисел Фібоначчі описаний наступним чином:

$$x_i = (x_{i-r} \Diamond x_{i-s}) \bmod N \tag{1.3}$$

В даному випадку $\Diamond$ - деяка бінарної операція [19].

Лінійний та квадратичний конгруентні генератори забезпечують рівномірність, проте згенерована послідовність чисел є лінійною. Генератор на основі чисел Фібоначчі забезпечує достатню нелінійність, проте згенерована послідовність нерівномірна. Тому вирішено використовувати удосконалений лінійний конгруентний генератор [20]:

$$\begin{cases} x_{i+1} = (ax_i + c) \bmod N \\ r_{i+1} = x_{i+1} \bmod M, x_{i+1} < (N - N \bmod M) \end{cases} \tag{1.4}$$

У даному генераторі $M < N$, варіюючи значення M можна отримати згенеровані числові послідовності, що розподілені рівномірно.

Відповідно до особливостей генерування модифікованих варіантів пароля користувачів при виборі генераторів випадкових чисел важливим показником є можливість рівномірного генерування чисел в довільному діапазоні чисел. Генератори Фібоначчі, лінійний та квадратичний конгруентні не володіють цієї властивістю для довільного діапазону, оскільки передбачають використання простих чисел як модуля.

Отже, для алгоритму генерування буде використано удосконалений лінійний конгруентний генератор, що володіє цією відомістю та дозволить рівномірно розподілити варіанти заміни символів.

1.5 Висновок до розділу

В розділі розглянуто методи автентифікації та визначено, що найпоширенішим та найзручнішим являється метод парольної автентифікації, оскільки паролі давно вбудовані в операційні системи й інші сервіси та є більш звичним та знайомим методом підтвердження особи для багатьох користувачів. При правильному використанні паролі можуть забезпечити прийнятний для багатьох систем рівень безпеки.

Проведено аналіз відомих методів генерування паролів, який показав, що метод генерування паролів завдяки асоціаціям легший для запам’ятовування користувачам, але також не несе за собою необхідної стійкості.

Розглянуто відомі методи зламу паролів, де визначено що найвідомішими методами є підбір геш значення паролів, атака за словником та атака підбором.

Тому, актуальним є розробка методів підвищення стійкості паролів, що допоможе унеможливити атаку за словником та атаку підбором.

Для реалізації запропонованих методів необхідним буде використання генератора псевдовипадкових чисел. Відповідно, був проведений аналіз на визначення генератора, що дасть можливість рівномірного генерування чисел в довільному діапазоні чисел для рівномірного розподілу в методах розмивання паролів.

2 МОДЕЛЬ ПІДВИЩЕННЯ СТІЙКОСТІ ПАРОЛІВ

2.1 Математичний опис процесу автентифікації

З метою узагальнення методів автентифікації необхідно розглянути процес автентифікації. Щоб описати систему використано теоретико-множинний підхід, що дозволить розглянути з чого складається процес автентифікації і що нового буде запропоновано.

$$\begin{aligned} UserAuthentication = \{&AuthenticationFactors, AuthenticationResults, \\ &retrieveFactor(\cdot), verifyFactor(\cdot)\} \end{aligned} \tag{2.1}$$

На вході є фактори автентифікації. Як було відображено в розділі 1.1 автентифікація може бути на основі знань Knowledge, на основі біометричних властивостей BiomentricFeature, за допомогою апаратних засобів Device. Тому множина факторів автентифікації буде складатися з наступних параметрів:

$$AuthenticationFactors = \{Knowledge, Biomentric Feature, Device\} \tag{2.2}$$

Відповідно результатами автентифікації можуть бути лише 2 наслідки – коли користувач автентифікувався і коли не автентифікувався.

$$AuthenticationResults = \{true, false\} \tag{2.3}$$

Таким чином, з точки зору аналізу можна описати будь-який з процесів автентифікації. Для парольної автентифікації буде пароль Passwords, результати автентифікації будуть такі ж – true або false, процес генерування пароля passwordGeneration та верифікація verifyPassword.

$$\begin{aligned} PasswordAuthentication = \{&Passwords, \{true, false\}, \\ &passwordGeneration(\cdot), verifyPassword(\cdot)\} \end{aligned} \tag{2.4}$$

Passwords – теоретично нескінченна множина, яка складається з комбінацій символів з алфавіту A, де A – множина символів, які можна ввести з

клавіатури визначеного стандарту. Тому, можна досягти будь-якої бажаної необхідної стійкості.

Проте існують певні обмеження, що не дозволяють використовувати паролі будь-якої довжини. Перш за все це можливості до запам'ятовування користувача. Відповідно існує дві складності при стандартній постановці задачі, щоб розв'язати задачу, поєднавши параметри досягання максимальної складності пароля та можливості його запам'ятовування звичайному користувачу.

Властивості пароля з точки зору автентифікації:

- складність для запам'ятовування rememberComplexity;
- складність для зламу breakingComplexity.

$$\begin{aligned} &\text{rememberComplexity} \to \min; \\ &\text{breakingComplexity} \to \max. \end{aligned} \tag{2.5}$$

Це суперечлива умова, оскільки breakingComplexity прямо пропорційний параметр до rememberComplexity. При збільшені довжини пароля збільшується складність його запам'ятовування і навпаки – закоротка ключова фраза не зможе надати необхідної стійкості.

Тому, виникає необхідність розробки утиліти, що допоможе користувачам з розв'язанням цієї задачі. Виникає постановка задачі магістерської кваліфікаційної роботи з точки зору математичного опису:

$$\begin{aligned} &\text{rememberComplexity} \le \text{const}; \\ &\text{breakingComplexity} \to \max. \end{aligned} \tag{2.6}$$

Це дозволить генерувати паролів певного розміру і дозволить збільшити стійкість.

Іншим варіантом постановки задачі може бути наступне:

$$\begin{aligned} &\text{rememberComplexity} \to \min; \\ &\text{breakingComplexity} \ge \text{const}. \end{aligned} \tag{2.7}$$

В даному випадку параметр складності запам’ятовування буде зводитися до мінімуму, а параметр стійкості буде обмежений певним мінімальним значенням.

Оскільки звичайні користувачі не обізнані стосовно стійкості паролів, в магістерській кваліфікаційній роботі пропонується обрати другий варіант, що дозволить задати певну межу мінімальної стійкості згенерованого пароля. Відповідно виникає необхідність додавання певних критеріїв перевірки, що дозволить відібрати найстійкіші з паролів поміж множини розмитих.

2.2 Метод підвищення стійкості паролів

Аналіз показав, що метод парольної авторизації є зручним та широко використовується в системах перевірки прав користувача. Однак такий метод передбачає наявність певних характеристик секретного слова користувача. Одним з головних таких характеристик є стійкість пароля та його зручність у використанні.

Як було визначено в розділі 2.1, постає завдання допомогти користувачам вирішити проблему вибору пароля, який матиме певне мінімальне значення стійкості та дасть змогу мінімізувати складність ключового слова, для більшої зручності запам’ятовування. Тому, розроблено метод, що дозволить вирішити ці проблему.

Спочатку користувачеві пропонується ввести слово, на основі якого буде надалі генеруватися парольна фраза. Оскільки, як зазначено в розділі 1.3, пароль повинен мати певну мінімальну довжину, слово відразу буде перевірятися на кількість літер. Якщо введене користувачем слово закоротке, буде повідомлено про це користувача та запропоновано обрати інший варіант.

Після виконання попередніх умов, пропонується зазначити щонайменше один метод розмивання паролю та значення імовірності виконання цього методу відповідно. Для вибору буде запропоновано три метода:

- заміна літер спеціальними символами;
- зміна регістра літери;

- додавання випадкової цифри в середину слова.

Метод, що заміняє літери на спеціальні символи буде використовуватися відповідно роботи «Засіб генерування стійких паролів» [21]. Коли користувач зазначає ймовірність заміни символа, перевіряється кожна літера слова і для неї генерується псевдовипадкове число. Якщо воно потрапить у проміжок, зазначений користувачем – літера буде замінена, якщо ні – літера залишається і процедура повторюється для наступної літери.

Для заміни будуть використовуватися варіанти символів (табл. 2.1), що були відібрані в результаті опитування.

Таблиця 2.1 – Варіанти заміни символів

Буква	Символи для заміни
A	@ 4 (L /\
B	8 I3 |3
C	< (
H	# /-/]-[}{
I	1 !
K	1<]< |<
M	/v\ /\/\
O	0 () []
P	|* |o /* /o
Q	(_)_ ()_ 0_
S	5 $
W	\/\/ vv \x/ uu
X	>< }{][ )(

Оскільки для кожної літери буде декілька варіантів набору символів, що можна використати, більшу імовірність можливої заміни будуть мати варіанти, що набрали більшу кількість голосів за результатами опитування. Для кожного набору символів сформовано проміжок відповідно до частки набраних голосів. Вибір буде здійснюватися за допомогою генератора псевдовипадкових чисел. Тобто, в який проміжок потрапить згенероване число, такий варіант заміни буде використано.

Для методу заміни регістра літери користувач також надає вхідні лані у вигляді ймовірності заміни регістра кожної літери. Кожен символ буде перевірятися чи є він літерою. Потім, подібно тому, як у попередньому випадку, буде генеруватися псевдовипадкове число і, у разі потрапляння в проміжок зазначений користувачем, буде відбуватися заміна регістра літери.

Третім методом розмивання паролів буде вставлення випадкової цифри в середину слова, для чого користувач також зазначає ймовірність потрапляння цифри в кожну парольну фразу і, у випадку потрапляння в проміжок вказаний користувачем згенерованого числа, в пароль буде додано випадкову цифру.

Відповідно на виході буде отримано варіанти розмитого ключового слова (рис. 2.1).

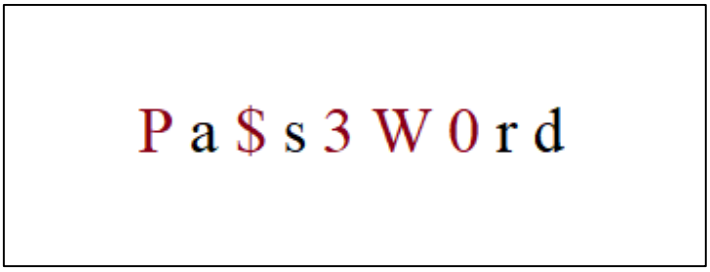


Рисунок 2.1 – Результат розмивання ключового слова

Наступним кроком буде перевірка варіантів згенерованих паролів, що дозволить відкинути варіанти, які не відповідають мінімальним вимогам, описаних в розділі 1.3. Кожна парольна фраза буде перевірена щодо виконання таких параметрів:

- перевірка довжини ключового слова;
- наявність спеціальних символів;

- наявність цифр;
- використання літер верхнього і нижнього регістрів.

Надто слабкі варіанти будуть видалені, а решта – виведені на екран користувача.

Оскільки методи розмивання паролів базуються на генераторі псевдовипадкових чисел, важливим показником є можливість рівномірного генерування чисел в довільному діапазоні чисел. Тому, буде використано удосконалений лінійний конгруентний генератор, який був описаний в розділі 1.5, що володіє цією властивістю та дозволить рівномірно розподілити варіанти заміни символів

2.3 Висновок до розділу

У розділі зроблено математичний опис використовуючи теоретико-множинний підхід, що дозволило розглянути з чого складається процес автентифікації та описати парольну автентифікацію.

Розроблено загальну структуру засобу підвищення стійкості паролів та описано методи, що будуть використовуватися для розмивання паролів.

Описано блок перевірки паролів, що дозволить відкинути найслабші варіанти можливих паролів та задати певне мінімальне значення стійкості пароля.

3 РОЗРОБКА ЗАСОБУ

3.1 Архітектура засобу підвищення стійкості паролів

Розроблено структуру засобу підвищення стійкості паролів (рис. 3.1), що дасть змогу користувачам використовувати даний підхід.

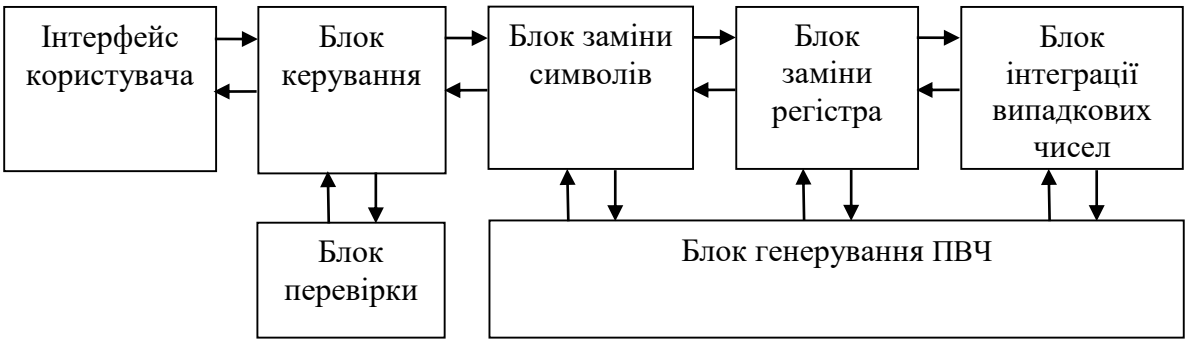


Рисунок 3.1 – Структура засобі підвищення стійкості паролів

Засіб підвищення стійкості паролів складається з інтерфейса користувача, блоку керування, блоку заміни символів, блоку заміни регістра, блоку інтеграції випадкових чисел, блоку генерування псевдовипадкових чисел та блоку перевірки.

Блок керування відповідає за формування команд іншим блокам та передачу даних між ними.

Щоб згенерувати пароль, користувач повинен через інтерфейс ввести ключове слово та задати параметри генерування, а саме – методи розмивання паролю, та значення ймовірності, з якою будуть замінюватися символи.

Алгоритми розмивання паролю працюють на основі генератора псевдовипадкових чисел, що був описаний в розділі 1.4, який дозволить рівномірно розподілити згенеровані числа на заданому проміжку та, відповідно, дасть змогу ефективного використання методів підвищення стійкості паролю.

Оскільки даний модуль генерування паролів спрямований не лише на підвищення стійкості, а й на підвищення комфорту користувачів при використанні парольної фрази, було використано таблицю заміни символів за

результатами опитування. Де також було відібрано символи, які можна ввести з клавіатури визначеного стандарту. Це дозволить використовувати варіанти заміни (табл. 2.1), що найбільше візуально схожі на відповідні їм літери на думку націлених споживачів, а саме – потенціальних користувачів цим засобом.

Лише використання заміни символів не дозволяє надати необхідної стійкості парольній фразі, що описано в результатах аналізу в розділі 1.3.

Тому пропонується використовувати декілька методів розмивання паролів, які дозволять унеможливити атаку за словником та ускладнити атаку грубої сили. Для цього додано блок заміни регістра літери та блок інтеграції випадкових чисел.

Блок заміни регістра також працює на основі генератора псевдовипадкових чисел. При введенні ключового слова для розмивання, користувач задає відсоток як у варіанті заміни літер на символи, після чого регістр кожної літери може бути змінений на протилежний із заданою імовірністю.

Блок інтеграції випадкових чисел буде додавати випадкове число в середину слова за бажанням користувача, що дасть змогу унеможливити атаку зі словником.

Оскільки звичайні користувачі можуть бути не обізнані стосовно стійкості паролів, додано блок перевірки ключового слова, для того, щоб відкинути найслабші варіанти можливих паролів та задати певне мінімальне значення стійкості пароля.

Щоб не витрачати великий час на перевірку стійкості набору паролів, що може не сподобатися користувачам, буде використано базові перевірки на дотримання рекомендацій, описаних в розділі 1.3 [14].

Надто слабкі варіанти будуть відкинуті, а решта – виведені на екран користувача.

3.2 Узагальнений алгоритм засобу підвищення стійкості паролів

Узагальнений алгоритм роботи засобу підвищення стійкості паролів зображено на рисунку 3.2.

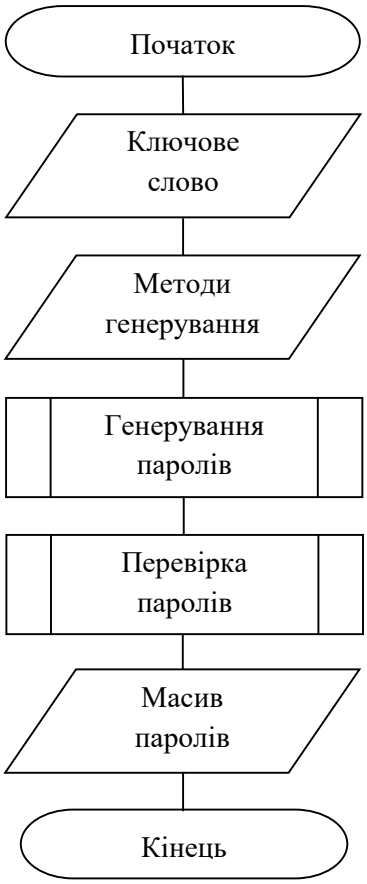


Рисунок 3.2 – Узагальнений алгоритм роботи засобу підвищення стійкості паролів

При запуску застосунку клієнту пропонується ввести ключове слово, на основі якого буде відбуватися генерування паролів.

Надалі необхідно обрати методи генерування, серед яких – заміна літер спеціальними символами, зміна регістру літери та вставлення випадкового числа всередину слова, та вхідні параметри генерування – ймовірність спрацювання для кожного з перерахованих методів.

Після чого відбувається процедура розмивання слова, що було введено користувачем за методами, які були зазначені.

Наступним кроком роботи застосунку є перевірка згенерованих паролів щодо забезпечення вимог до стійкості. Оскільки варіанти заміни літер можуть складатися не лише з одного символа, а й з двох чи трьох, відповідно парольні фрази, утворені після розмивання слова користувача, можуть бути різної

довжини, що впливає на стійкість. Також, оскільки методи розмивання паролів базуються на використанні генератора псевдовипадкових чисел, в деяких випадках вони можуть бути не застосовані для генерування того чи іншого слова. Тому, необхідно також перевіряти вихідні фрази на наявність літер різних регістрів та цифр. Це дасть змогу відкинути найслабші варіанти парольних фраз та допомогти користувачам, які можуть бути не обізнані в області стійкості паролів, обрати пароль з певним обмеженням мінімальної стійкості, використовуючи прості та швидкі методи перевірки надійності ключового слова.

3.3 Алгоритм роботи генерування паролів

Один з методів, що буде використовуватися для заміни літер на подібні символи, взято з бакалаврської роботи. Алгоритм методу зображено на рис. 3.3.

На вході подаються дані, що користувач вводив, а саме слово, яке буде розмиватися надалі, та значення ймовірності зміни символів у слові.

Заміна кожної літери буде відбуватися у циклі, кількість ітерацій якого залежить від кількості символів слова, що подавалося на вході, де буде визначено чи заміниться літера на символ.

Заміна побудована на генераторі псевдовипадкових чисел, межі генерування якого будуть визначатися від 0 до 100. Якщо користувач вкаже частку заміни літер наприклад 40% і згенероване випадкове число потрапить в проміжок від 0 до 40, відповідна літера у слові буде замінюватися на спеціальний символ.

Для різноманітності парольних фраз кожна літера, що буде замінятися, може мати декілька варіантів символів для заміни.

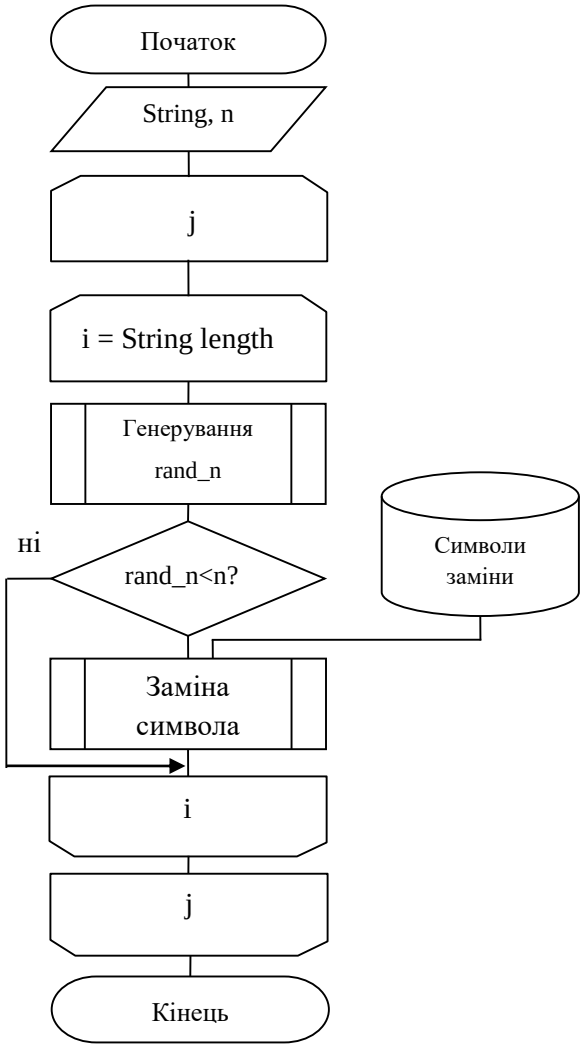


Рисунок 3.3 – Алгоритм роботи методу заміни літер на спеціальні символи

Щоб згенерувати пароль будуть використовуватися набори символів, що належать до алфавіту А, тобто, що можна ввести з певного стандарту клавіатури, та попередньо відібрані результатами опитування. Вибір символів, що будуть використовуватися замість букв, також залежить від генератора ПВЧ. З відібраних варіантів утворюється співвідношення на проміжку від 0 до 100. Буква буде замінена на символ, в проміжок якого входить згенероване число.

Другим методом, що користувач зможе обрати для розмивання паролів буде заміна регістра літери (рис. 3.4).

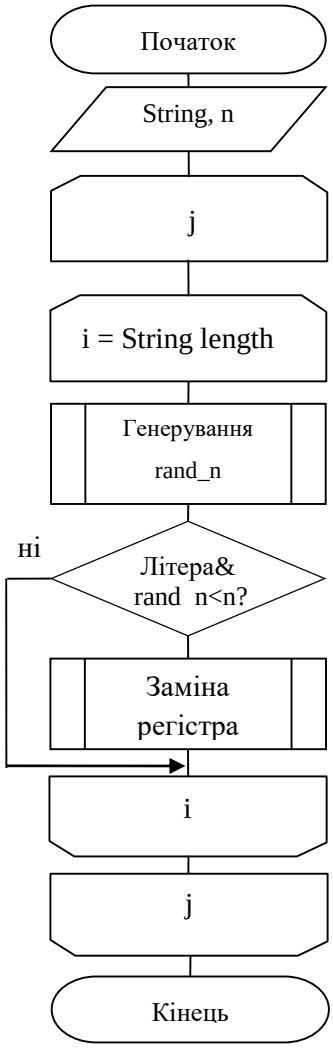


Рисунок 3.4 – Алгоритм роботи методу зміни регістру літер

Даний метод також побудований на генераторі псевдовипадкових чисел. На вході користувач вказує значення ймовірності заміни регістра кожної літери. Відповідно після генерування псевдовипадкового числа, відбувається перевірка чи являється даний символ літерою і чи входить згенероване число в проміжок, вказаний користувачем. Якщо обидві умови виконуються, відбувається зміна регістра літери.

Третім методом, що буде використовуватися при підвищенні стійкості парольної фрази, є вставлення випадкового числа в середину слова (рис. 3.5).

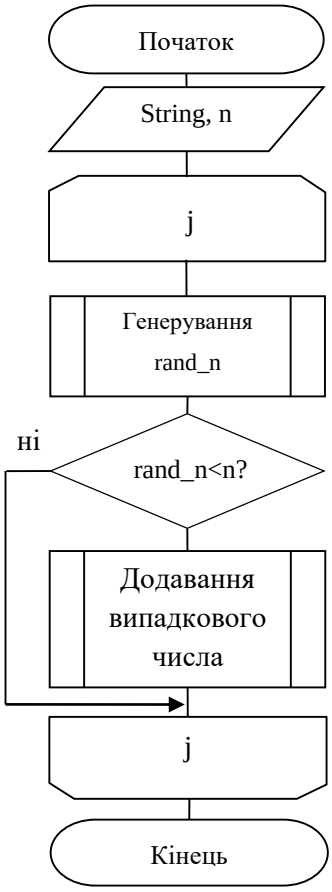


Рисунок 3.5 – Алгоритм роботи методу вставлення випадкового числа

Даний метод має лише один цикл, оскільки визначення того, чи буде використовуватися цей вид розмивання слова, визначатиметься один раз для кожної парольної фрази. Як і попередні два методи, працюватиме на генераторі псевдовипадкових чисел, де користувач вказуватиме відсоток потрапляння випадкового числа в середину слова.

Для того, щоб згенеровані числа були рівномірними на заданому проміжку, буде використовуватися удосконалений лінійний конгруентний генератор. Алгоритм роботи удосконаленого генератора зображено на рисунку 3.6.

Щоб забезпечити послідовність повного періоду генератора, потрібно знайти початкові значення. Стартове значення x0 дозволяється брати довільне в межах від 0 до N, N – найбільше ціле число, яке можливо відтворити на комп’ютері [22].

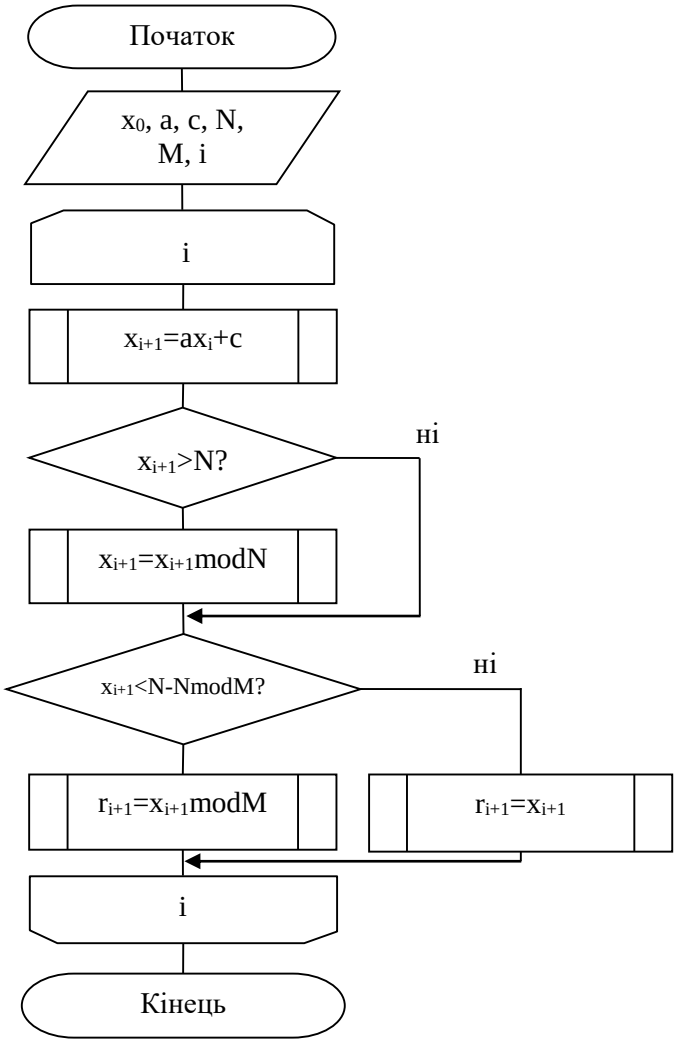


Рисунок 3.6 – Алгоритм роботи удосконаленого генератора ПВЧ

Щоб отримати послідовність чисел з повним періодом константа повинна бути непарною, число a - 1 повинно ділитися на 4.

Оскільки методи розмивання паролів побудовані на генераторі псевдовипадкових чисел, можуть траплятися варіанти паролів, коли ні один з методів не був застосований, що відповідно зменшить значення стійкості ключової фрази. Тому, після отримання множини розмитих паролів, виникає необхідність відібрати найкращі результати роботи застосунку.

3.4 Алгоритм тестування паролів

Після розмивання слова, що вводив користувач, утворюється масив з варіантів згенерованих паролів, серед яких користувач може обрати парольну фразу, що сподобалася йому найбільше. Проте, як було описано в розділі 2.2,

користувач може бути не обізнаним в сфері стійкості паролів, а наявність генератора псевдовипадкових чисел допускає можливість, що не всі описані методи розмивання паролів будуть використані, в результаті чого отримані результати не будуть відповідати мінімальним вимогам стійкості паролів. Тому, в застосунку буде використовуватися перевірка паролів (рис. 3.7), що дозволить відсіяти слабкі варіанти паролів та дати змогу користувачам обирати парольну фразу серед більш стійких варіантів.

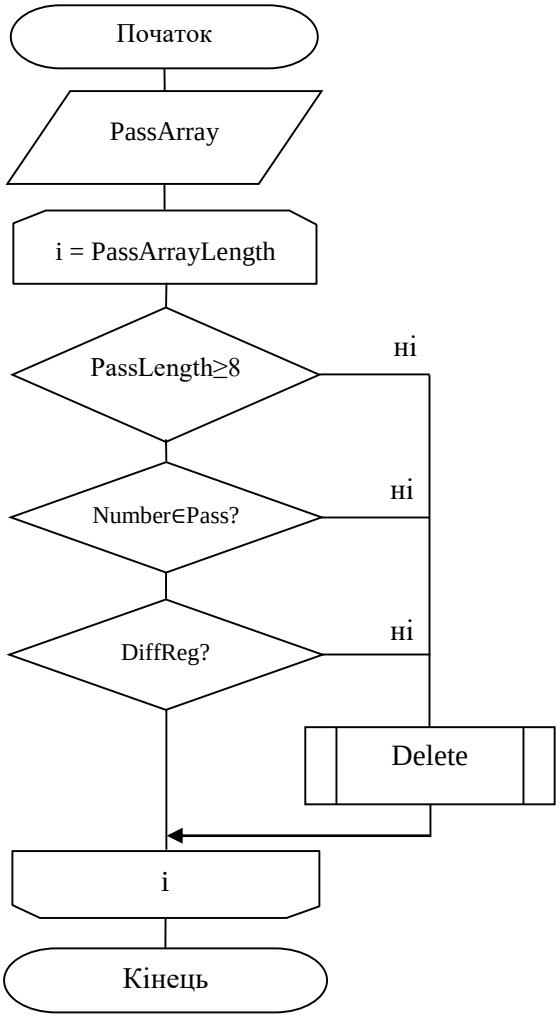


Рисунок 3.7 – Алгоритм роботи перевірки паролів

На вході подається масив парольних фраз, що попередньо були згенеровані. Далі кожен пароль проходить перевірку на довжину, наявність цифри та різних регістрів відповідно. Якщо одна з перевірок не пройдена, варіант пароля видаляється з масиву.

3.5 Висновок до розділу

У розділі було розроблено загальну архітектуру засобу підвищення стійкості паролів, що показує основні модулі програми та їх взаємодію між собою. Засіб підвищення стійкості паролів складається з інтерфейса користувача, блоку керування, блоку заміни символів, блоку заміни регістра, блоку інтеграції випадкових чисел, блоку генерування псевдовипадкових чисел та блоку перевірки.

На основі архітектури розроблено загальний алгоритм роботи засобу підвищення стійкості паролів. Для генеруваня паролів створено алгоритми методів розмивання паролів, а саме - алгоритм роботи методу заміни літер на спеціальні символи, алгоритм роботи методу зміни регістру літер та алгоритм роботи методу вставлення випадкового числа, а також алгоритм роботи генератора псевдовипадкових чисел, шо використовується для всіх методів розмивання парольної фрази.

Розроблено алгоритм методу перевірки згенерованих паролів, що дозволить відсіяти слабкі варіанти паролів та дати змогу користувачам обирати парольну фразу серед більш стійких варіантів.

4 ТЕСТУВАННЯ ЗАСОБУ

4.1 Обгрунтування вибору засобів розробки

Щоб розробити засіб підвищення стійкості паролів було обрано мову програмування Java. Дана мова програмування не являється найзручнішим засобом для реалізації криптографічних перетворень, проте має ряд інших переваг. Можливість використовувати застосунок на різних пристроях з різними операційними системами, що забезпечує Java, покращить конкурентоспроможність застосунку [23].

Більше того, Java менш складна, ніж такі мови, як C і C++, оскільки багато складних функцій цих мов вилучено з Java, наприклад, концепція явних покажчиків, класи зберігання, перевантаження операторів і багато іншого [23].

Java - об'єктно-орієнтована мова, яка допомагає підвищити гнучкість і можливість повторного використання коду. Використовуючи концепцію ООП, можна легко повторно використовувати об’єкт в інших програмах.

Це також допомагає підвищити безпеку, об’єднуючи дані та функції в єдиний блок і не даючи до них доступу назовні. Це також допомагає розділити більші модулі на менші, для кращого розуміння структури.

Java зменшує загрози безпеці та ризики, уникаючи використання явних покажчиків. Покажчик зберігає адресу пам'яті іншого значення, яке може спричинити несанкціонований доступ до пам'яті.

Це питання вирішується шляхом видалення концепції покажчиків. Крім того, у Java для кожної програми є менеджер безпеки, який дозволяє визначати правила доступу до класів [23].

Підтримується автоматичне керування пам’яттю, яке керується віртуальною машиною Java. Щоразу, коли об’єкти більше не використовуються програмами і вони не посилаються ні на що, вони не потребують видалення за допомогою явного програмування. Java автоматично видаляє невикористані об’єкти за допомогою автоматичного процесу збирання сміття.

Для використання додатку користувачам, необхідно розробити графічний інтерфейс. Для цього був використаний JavaFX [24], в якому доступний багатий графічний та мультимедійний API.

JavaFX дозволяє створювати програми Java з сучасним апаратним прискореним інтерфейсом користувача, який є дуже портативним [24].

Щоб створити зрозумілий користувачу інтерфейс використовувався застосунок SceneBuilder [25]. Це інструмент візуального компонування, який дозволяє користувачам швидко проектувати інтерфейси користувача програми JavaFX без кодування. Користувачі можуть перетягувати компоненти інтерфейсу користувача в робочу область, змінювати їх властивості, застосовувати таблиці стилів, а код FXML для макета, який вони створюють, автоматично генерується у фоновому режимі. Результатом є файл FXML, який потім можна об’єднати з проектом Java, прив’язавши інтерфейс користувача до логіки програми.

4.2 Основні семантичні структури програми

Засіб підвищенян стійкості паролів реалізовано у вигляді 7 файлів. Класи Main, Controller, Parameters, Result – описують логіку, а в файлах FXML sample, passParameters та passResult описано інтерфейс користувача.

Клас Main містить метод Main, який використовується для виклику першого вікна інтерфейсу, де користувач вказує слово, на основі якого буде генеруватися пароль (рис. 4.1). Методи, що використовуються, описані в класі Controller. У файлі sample описано графічний інтерфейс вікна. Методами класу Controller є recheckWord та initialize.

Метод recheckWord використовується для перевірки кількості символів в слові, що вводить користувач. Функція отримує значення String та повертає результат true або false.

Метод initialize використовується для ініціалізації класу FXML-контролера та зв’язування з файлом опису інтерфейсу користувача sample. В якості

аргументів середовище програмування передає URL-адресу опису графічного інтерфейса та ресурси.

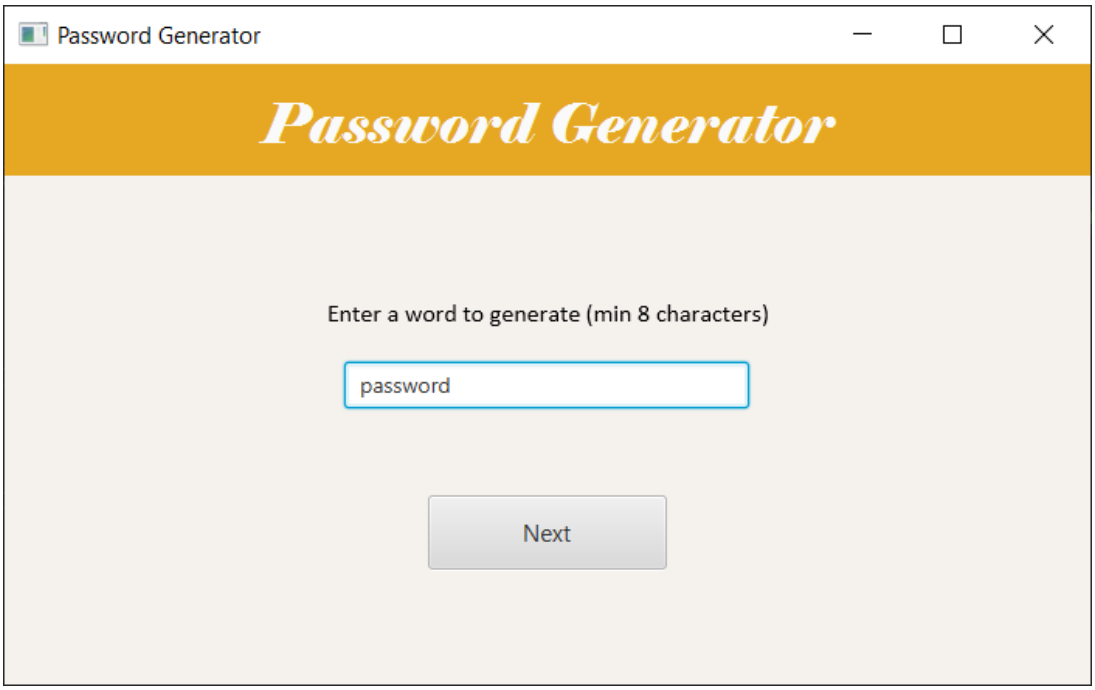


Рисунок 4.1 – Початкове вікно програми

При натисканні користувачем на кнопку «Next», у випадку якщо обране слово підходить за кількістю символів, викликається вікно, де обираються методи розмивання паролю та їх параметри (рис. 4.2).

Логіка та графічний інтерфейс описані в класі Parameters та файлі passParameters відповідно. В класі використовуються методи initialize, Rand_N, ReplacingLettersSymbols, ChangeCase, AddNumber та RecheckPass.

Метод Rand_N використовується для генерування випадкового числа і викликається методами ReplacingLettersSymbols, ChangeCase, AddNumber. Початкове значення x_0 визначається за допомогою currentTimeMillis.

ReplacingLettersSymbols метод, що використовується для заміни літер на спеціальні символи. На вхід подаються String - введене слово користувача, на основі якого буде будуватися пароль, та число, що також задає користувач, яке буде порівнюватися з псевдовипадковим числом та відповідати за відсоток символів, що будуть замінені в слові.

Рисунок 4.2 – Вікно параметрів генерування паролів

Метод ChangeCase відповідає за зміну регістра символів. На вхід подаються String та число, вказав користувач, яке буде порівнюватися з псевдовипадковим числом.

AddNumber метод, який додає всередину слова випадкову цифру. На вхід подаються String та число – обране користувачем, що також буде порівнюватися з псевдовипадковим згенерованим числом.

Метод RecheckPass перевіряє паролі на наявність параметрів, що були описані в розділі 1.3. Для цього на вхід подається String, після чого перевірений пароль видаляється з масиву парольних фраз, якщо не відповідає критеріям.

Метод initialize як і в попередньому класі використовується для ініціалізації класу FXML-контролера та зв’язування з файлом опису інтерфейсу користувача.

Після натискання на кнопку «Generate», викликається вікно, де виводяться результати генерування паролів (рис. 4.3). Логіка та графічний інтерфейс описані в класі Result та файлі passResult відповідно. В класі використовується метод initialize для ініціалізації класу FXML-контролера та зв’язування з файлом опису інтерфейсу користувача.

Рисунок 4.3 – Вікно результатів генерування паролів

В полі виводяться варіанти згенерованих та перевірених парольних фраз, серед яких користувач може обрати пароль.

4.3 План тестування

Для тестування застосунка необхідно розробити план [26], обрати відповідний функціонал та визначити функції, що буде перевірено, задля правильної роботи програми. Для цього було вирішено створити чек-лист [27, 28], що дозволить перерахувати пункти програми, які необхідно перевірити, та відкинути варіант, де буде не враховано певного функціоналу програми та не перевірено коректність його роботи.

Для формування чек-листа необхідно визначити що буде тестуватися. Оскільки застосунок буде складатися з трьох вікон, вирішено розділити список перевірки відповідно до кожного (табл. 4.1).

Чек листу подано у вигляді таблиці, де названо критерій, що має перевірятися, та короткий опис цього критерію, тобто що має відбуватися у випадку правильної роботи застосунку.

Таблиця 4.1 – План тестування застосунку

Критерій	Опис
Вікно для введення слова	
Застосунок відкривається	Застосунок відкривається
Форма доступна для введення слова	У вікні додатку існує форма, де користувач задає слово, на основі якого буде згенеровано варіанти паролів
Виконується перевірка обраного слова	Після того, як користувач задає слово для генерування, виконується перевірка на кількість символів, яка має складати не менше восьми. В іншому випадку виводиться повідомлення про помилку
Присутня та працює кнопка для переходу до наступного вікна	Присутня кнопка, що дозволяє користувачеві перейти до наступного вікна програми та запускає перевірку заданого слова
Вікно задання параметрів генерування	
Для кожного методу розмивання паролів існують чекбокси, які користувач може обрати	Для генерування паролів використовується три методи розмивання, які користувач може обрати за власним бажанням
Присутні випадні меню з відсотковими значеннями для кожного методу розмивання паролів	Відповідно до описаних в 2.2 розділі методів розмивання, для кожного методу необхідно задати ймовірність спрацювання

Продовження таблиці 4.1

Критерій	Опис
Виконується перевірка вказаних методів	Для генерування паролів користувач повинен вказати щонайменше 1 методі розмивання ключового слова. В іншому випадку буде виведене повідомлення про помилку вказаних даних
Присутня та працює кнопка генерування паролів	Присутня кнопка, що запуска перевірку заданих методів розмивання, дозволяє перейти на наступне вікно програми та згенерувати варіанти парольних фраз
Вікно результатів генерування	
Виконується перевірка отриманих результатів генерування	Після отримання варіантів згенерованих паролів, виконується перевірка, описана в розділі 2.2. Варіанти паролів, що не пройшли перевірку, видаляються
Присутнє поле для виведення результатів генерування	Присутнє поле де виводяться всі варіанти паролів, що пройшли перевірку

Тестування відповідно до розробленого чек-листа буде проведено та описано в наступному підрозділі.

4.4 Сценарій тестування

Для проведення тестування розроблено два тестові сценарії [29, 30], що представляє собою послідовність дій користувача для двох результатів тестування – позитивного і негативного.

Позитивним результатом тестування буде вважатися отримання на виході роботи програми набір варіантів паролів, що відповідають критеріям перевірки описаних в розділі 2.2.

Негативним результатом буде вважатися, коли всі згенеровані паролі не пройдуть перевірку після методів розмивання і відповідно користувач не отримає варіантів паролів.

Сценарій тестування для позитивного результату роботи застосунку:

Крок 1. Користувач запускає додаток, вводить слово більше довжиною 8 або більше символів та натискає кнопку «Next».

Крок 2. Користувач задає два або три методи розмивання паролів та ймовірності спрацювання кожного обраного методу.

Крок 3. Користувач натискає кнопку «Generate» та отримує варіанти парольних фраз.

Сценарій тестування для негативного результату роботи застосунку:

Крок 1. Користувач запускає додаток, вводить слово менше восьми символів.

Крок 2. Виводиться повідомлення про помилку, після чого користувачеві необхідно обрати слово довше або рівне восьми символам для продовження роботи програми.

Крок 3. Користувач не задає методи розмивання паролів та натискає кнопку «Generate».

Крок 4. Виводиться повідомлення про помилку, після чого необхідно задати щонайменше один метод розмивання та ймовірність його спрацювання.

Крок 5. Користувач вказує метод додавання цифр в середину слова та натискає кнопку «Generate».

Крок 6. Варіанти паролів не виводяться, оскільки всі парольні фрази не пройшли перевірку описану в розділі 2.2.

Відповідно, розроблені сценарії тестування можна зобразити у вигляді загального алгоритму (рис. 4.4). Даний алгоритм відображає всі можливі кроки користування застосунком користувача як для позитивного так і негативного вихідних результатів.

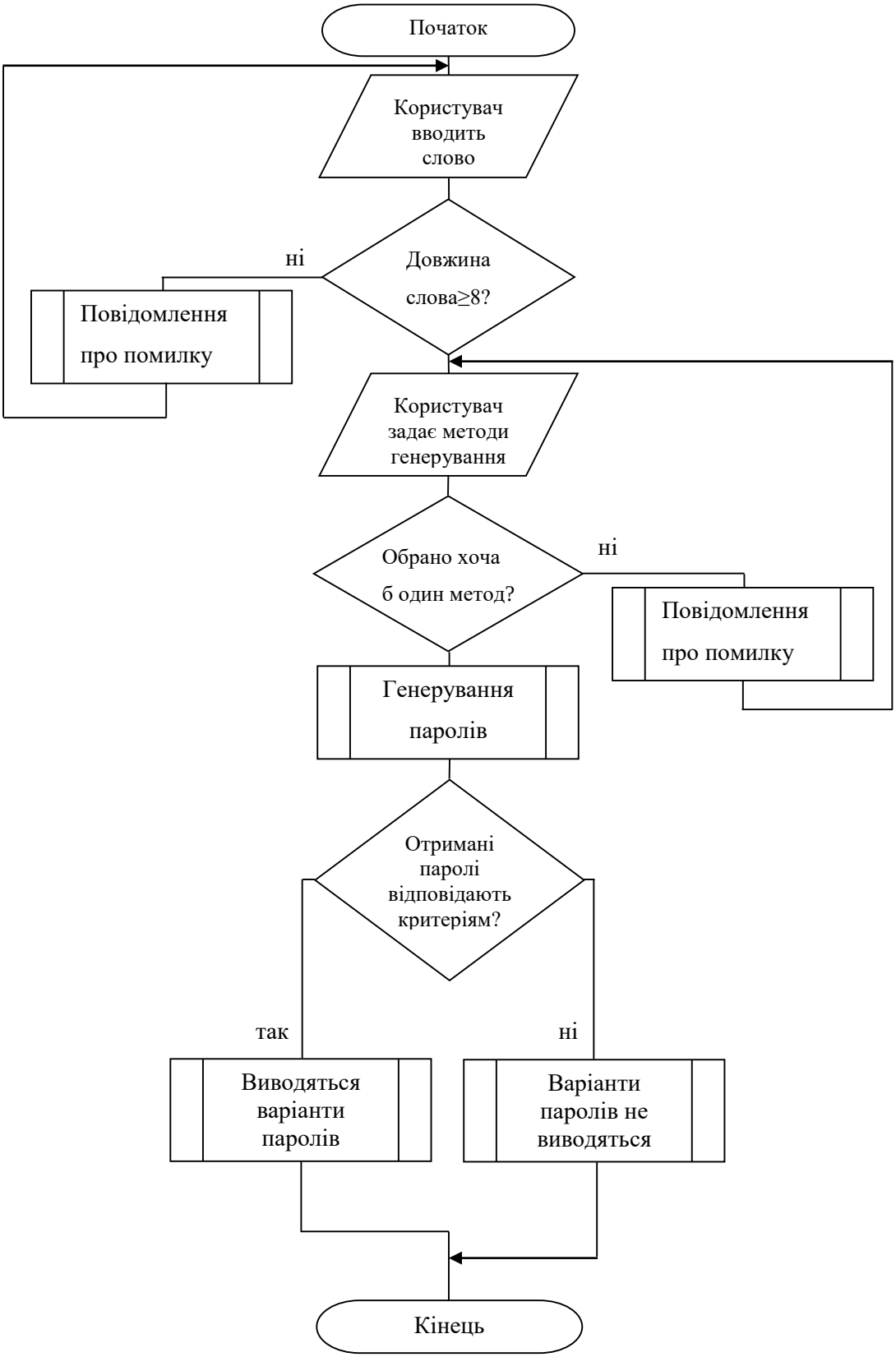


Рисунок 4.4 – Загальний алгоритм тестування

Тепер можна переходити власне до тестування.

4.5 Функціональне тестування

Після того як користувач запускає застосунок, з’являється початкове вікно програми (рис. 4.1), де користувачу пропонується ввести слово, на основі якого буде генеруватися пароль.

Якщо дані не введено або користувач вказав закоротке слово, виведеться відповідне повідомлення (рис. 4.5) про те, що обране слово закоротке для подальшої роботи програми.

Для забезпечення більшої стійкості парольної фрази, слово перевіряється на кількість символів (не менше восьми).

У випадку якщо кількість символів більша, відкривається вікно застосунку, де необхідно обрати методи розмивання паролів та ймовірність застосування відповідного методу (рис. 4.2).

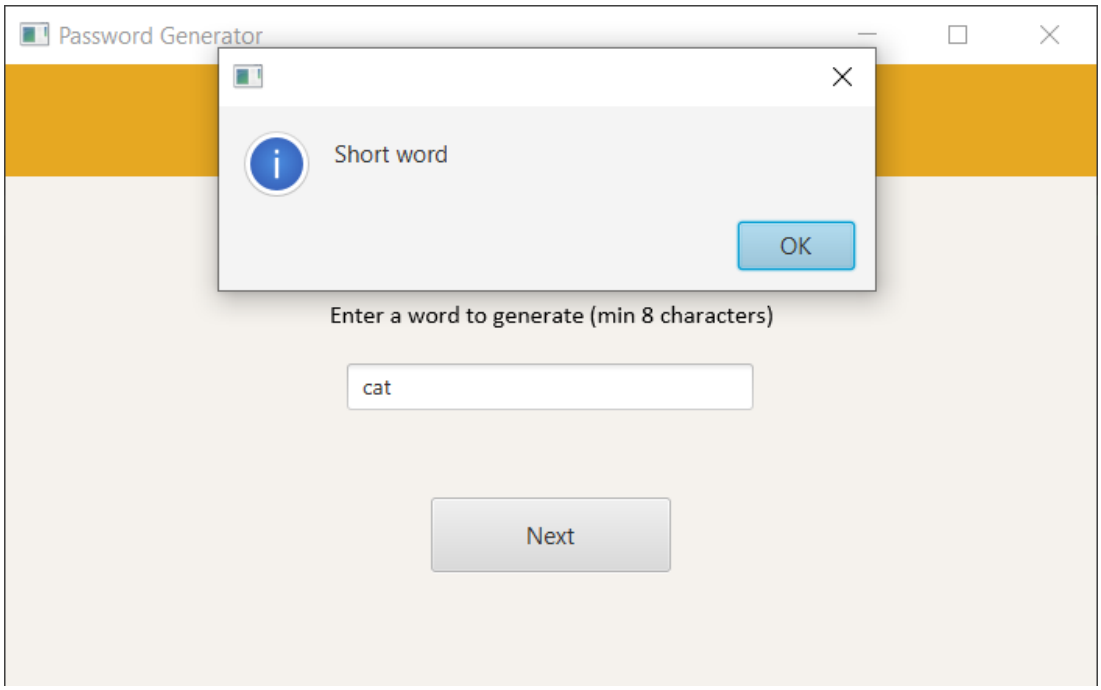


Рисунок 4.5 – Повідомлення про помилку вводу слова

У випадку коли користувач не обрав ні одного методу розмивання паролів, буде виведено повідомлення про те, що необхідно обрати хоча б одну опцію з перерахованих (рис. 4.6).

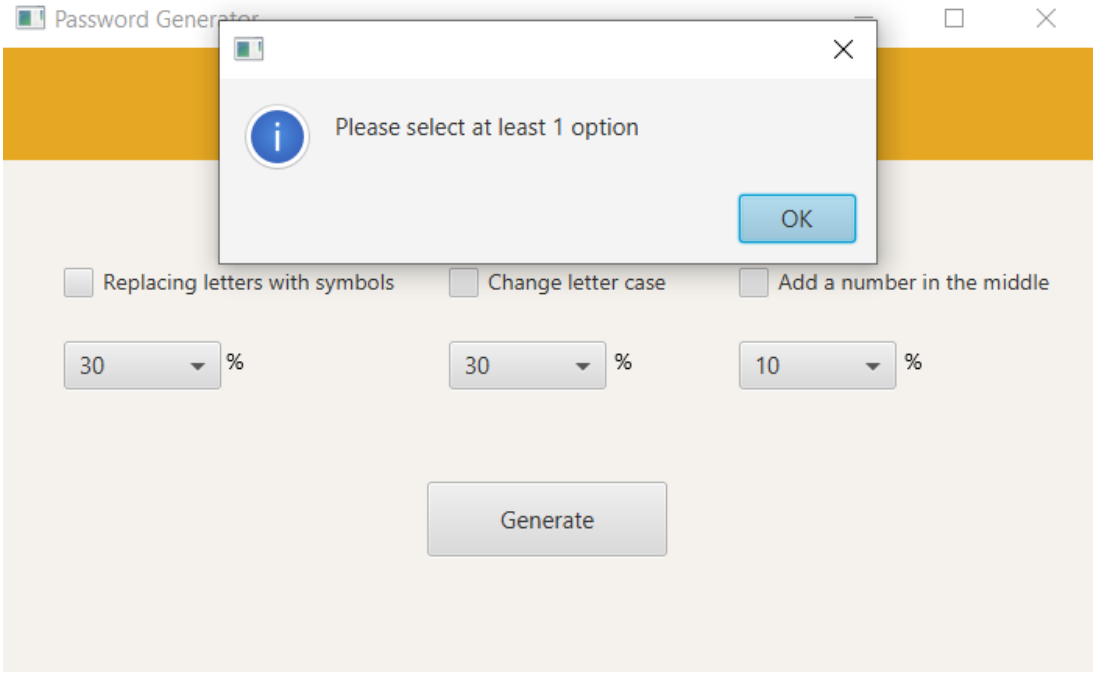


Рисунок 4.6 – Повідомлення про помилку обраних методів

Після правильного введення даних для генерування, відкривається наступне вікно, де виводяться всі варіанти згенерованих паролів, що пройшли тестування (рис. 4.7).

Рисунок 4.7 – Результати генерування паролів

З отриманих результатів користувач може обрати варіант, що найбільше його влаштовує.

4.6 Блокове тестування

Блокове тестування [31] здійснювалося за допомогою відкритої бібліотеки Junit [32, 33]. Для цього було створено новий клас з набором функцій, що викликали відповідні методи з класу реалізації модуля авторизації.

Всі методи, що необхідно протестувати, описані в класі Parameters, тому тестування проводилося саме для цього класу (рис. 4.7).

Основні методи, що реалізовані в застосунку підвищення стійкості паролів, це методи розмивання паролів ReplacingLettersSymbols, ChangeCase та AddNumber, метод генерування псевдовипадкових чисел Rand_N.

Методи розмивання паролів працюють відповідним чином, якщо вихідні паролі відрізняються один від одного та від початкового вхідного значення. Для перевірки двічі використано Assert.assertNotEquals, результатом якого буде успішне тестування, якщо згенерований пароль відрізняється від початкового введеного слова, та якщо згенеровані паролі відрізняються один від одного при однакових вхідних значеннях.

Метод генерування випадкових чисел Rand_N працює коректно, якщо кожне наступне згенероване число відрізняється від попередніх. Тому використано Assert.assertNotEquals, що порівнює отримані значення методу.

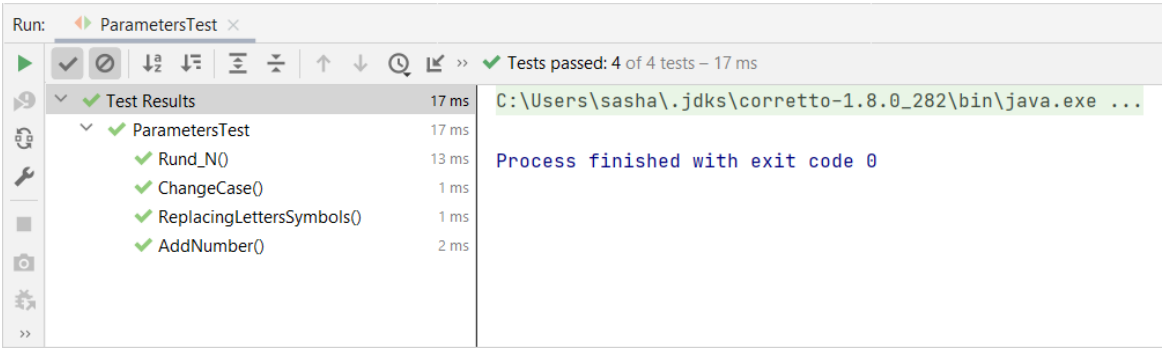


Рисунок 4.7 – Результати блокового тестування

Результати показали що методи розмивання паролів та генерування псевдовипадкових чисел працюють коректно.

4.7 Тестування методу заміни символів

За результатами опитування відібрано варіанти заміни для кожного символа та ймовірність використання кожного варіанту, залежно від кількості набраних голосів (табл. 4.2).

Таблиця 4.2 – Ймовірності заміни символів за результатами опитування

Буква	Символ для заміни	Ймовірність заміни, %	Символ для заміни	Ймовірність заміни, %
A	@	88	/\	12
B	8	63	I3	37
C	(	89	<	11
H	#	69	]-[	31
I	1	52	!	48
K	|<	66	1<	34
M	/\/\	57	/v\	43
O	0	77	()	23
Q	0_	72	()_	28
S	$	88	5	12
X	}{	62	><	38
W	vv	59	\/\/	41

Метод розмиванна працює на основі генератора псевдовипадкових чисел описаному в розділі 1.4. Тому доцільно перевірити, що справді заміни відбуваються з тією ймовірністю, яку закладали відповідно до результатів опитування.

Для перевірки на функцію буде подавати слово з набору символів, що входять до алфавіту заміни символів, та ймовірність заміни символа 100%. Для більшої точності кожна літера буде замінюватися тисячу разів. Алгоритм проведення тестування зображено на рисунку 4.8.

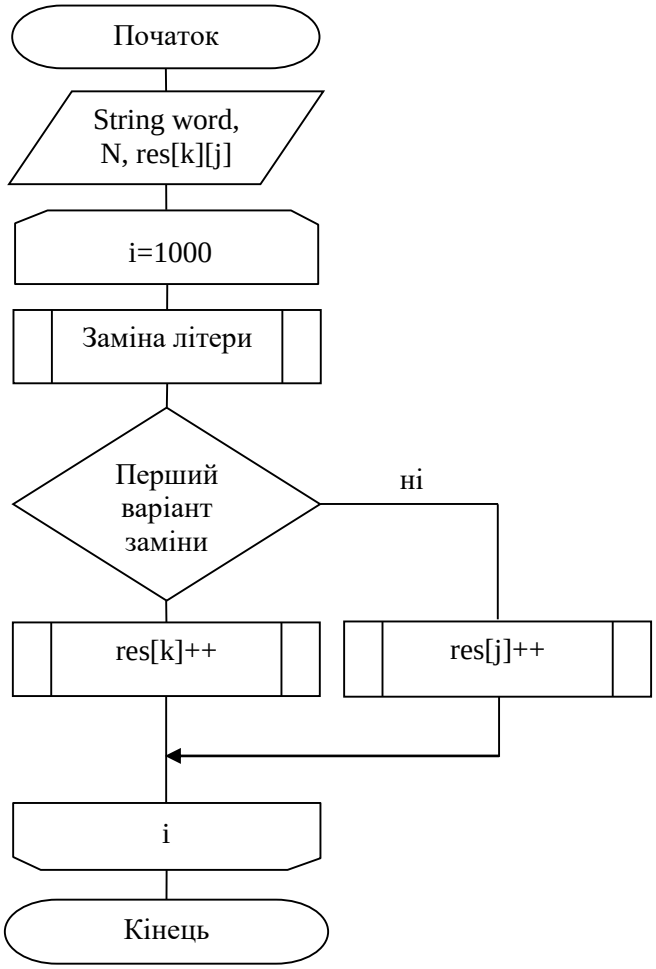


Рисунок 4.8 – Алгоритм тестування методу заміни літер на символи

Відповідно до алгоритму, на вхід подаються слово для генерування і ймовірність заміни символа.

Для кожної літери існує масив, до будуть відображені результати тестування у вигляді кількості раз, коли був використаний перший чи другий варіант заміни. Потім результати будуть приведені до вигляду відсоткових значень та порівняні зі значеннями, що були зібрані за допомогою опитування та вказані при реалізації програми.

Тестування вважатиметься успішним, якщо відсотки заміни символів будуть наближеними або співпадуть з ймовірностями, що задавалися (табл 4.2).

Результати тестування при ймовірності заміни сомволів сто відсотків відображені в таблиці 4.3.

Таблиця 4.3 – Результати тестування при ймовірності заміни 100%

Буква	Символ для заміни	Ймовірність заміни, %	Результат тестів %	Символ для заміни	Ймовірність заміни, %	Результат тестів %
A	@	88	90	/\	12	10
B	8	63	64	I3	37	36
C	(	89	90	<	11	10
H	#	69	70	]-[	31	30
I	1	52	54	!	48	46
K	|<	66	68	1<	34	32
M	/\/\	57	58	/v\	43	42
O	0	77	78	()	23	22
Q	0_	72	72	()_	28	28
S	$	88	88	5	12	12
X	}{	62	64	><	38	36
W	vv	59	60	\/\/	41	40

За результатами тестування видно, що ймовірність заміни літери на відповідний символ близька до ймовірності що була визначена за результатами опитування та не перевищує похибки 2%.

З метою перевірки роботи частини функціональності програми, пов’язаної з заданням ймовірності зміни символу проведено ще один тест, де для перевірки на функцію буде подавати слово з набору символів, що входять до алфавіту заміни символів, та ймовірність заміни символа 50%.

Тест буде вважатися успішним, якщо частота заміни знизиться приблизно вдвічі для кожної літери, а пропорція щодо частоти зустрічі кожного з варіантів заміни символів залишиться.

Результати тестування відображені в таблиці 4.4.

Таблиці 4.4 – Результати тестування при ймовірності заміни 50%

Буква	Символ для заміни	Ймовірність заміни, %	Резу-льтат тестів %	Символ для заміни	Ймовірність заміни, %	Резу-льтат тестів %	К-ть замін %
A	@	88	86	/\	12	14	53,8
B	8	63	60	I3	37	40	53,9
C	(	89	88	<	11	12	53,8
H	#	69	66	]-[	31	34	54
I	1	52	57	!	48	43	53,9
K	|<	66	60	1<	34	40	53,9
M	/\/\	57	52	/v\	43	48	53,7
O	0	77	77	()	23	23	53,9
Q	0_	72	69	()_	28	31	53,7
S	$	88	86	5	12	14	54,2
X	}{	62	57	><	38	43	53,7
W	vv	59	55	\/\/	41	45	53,9

Як видно з результатів, частота заміни символів знаходиться в межах 53-45% при заданому вхідному параметрі 50%. Тобто, частота заміни знизилася вдвічі для кожної літери.

Пропорція частоти для заміни кожного варіанта символа залишилася приблизно такою ж, хоча похибка зросла до 5% відносно попередніх результатів, де похибка була 2%.

4.8 Висновок до розділу

Розглянуто можливі засоби реалізації модуля авторизації користувачів. Обрано мову програмування Java, що дозволить використовувати застосунок на

різних операційних системах, JavaFX та SceneBuilder, що дозволять створити зрозумілий графічний інтерфейс.

Розроблено застосунок, який реалізує засіб підвищення стійкості паролів на основі розроблених алгоритмів.

Розроблено план тестування у вигляді чек-листа, що дозволило обрати відповідний функціонал та визначити функції, що буде перевірено, задля правильної роботи програми.

Для проведення тестування розроблено два тестові сценарії, що представляє собою послідовність дій користувача для двох результатів тестування – позитивного і негативного.

Виконано функціональне тестування, яке показало що всі функції програми працюють належним чином.

Проведено блокове тестування, результати якого показали що методи розмивання паролів та генерування псевдовипадкових чисел працюють коректно.

Розроблено алгоритм тестування методу заміни літер на спеціальні символи. Проведено тестування методу за даним алгоритмом. Результати показали, що метод працює коректно, а пропорції частоти для заміни кожного символа наближені до заданих.

5 ЕКОНОМІЧНА ЧАСТИНА

Метою економічної частини магістерської кваліфікаційної роботи є обґрунтування економічної доцільності розробки методу та засобу підвищення стійкості паролів. Для цього необхідно виконати такі етапи робіт:

- оцінити комерційний потенціал розробки;
- спрогнозувати витрати на виконання наукової роботи та впровадження її результатів;
- спрогнозувати комерційний ефект від реалізації результатів розробки;
- розрахувати ефективність вкладених інвестицій та період їх окупності.

5.1 Оцінювання комерційного потенціалу розробки (технологічний аудит розробки)

Об’єктом дослідження магістерської кваліфікаційної роботи є процес автентифікації користувачів в цифровому середовищі.

Для проведення технологічного аудиту було залучено трьох незалежних експертів: Баришев Ю. В., Войтович О. П., Шелепало Г. В. Кожен з експертів повинен ознайомитися з запропонованою розробкою та заповнити таблицю, яка визначає рекомендовані критерії оцінювання комерційного потенціалу розробки та їх можливу оцінку в балах. Після виконання цього, підраховується середньоарифметична сума балів та визначається який рівень комерційного потенціалу має нова розробка.

Здійснюємо оцінювання комерційного потенціалу розробки за 12-ю критеріями, наведеними в таблиці 5.1.

Таблиця 5.1 – Результати оцінювання комерційного потенціалу розробки

Критерії	Прізвище, ініціали експерта		
	Баришев Ю. В.	Войтович О. П.	Шелепало Г. В.
	Бали, виставлені експертами:		
1	3	2	3
2	2	3	2
3	4	3	3
4	2	1	2
5	2	2	2

Продовження таблиці 5.1

Критерії	Прізвище, ініціали експерта		
	Баришев Ю. В.	Войтович О. П.	Шелепало Г. В.
	Бали, виставлені експертами:		
6	3	3	3
7	2	3	3
8	4	3	3
9	4	3	4
10	4	3	4
11	3	4	4
12	3	3	3
Сума балів	$СБ_1$=36	$СБ_2$=33	$СБ_3$=36
Середньоарифметична сума балів $\overline{СБ}$	$\overline{СБ} = \frac{\sum_{1}^{i} СБ_i}{i} = \frac{36+33+36}{3} = 35$		

За отриманими результатами оцінки із таблиці 5.1, можна зробити висновок базуючись на таблиці 5.2 рівнів комерційного потенціалу розробки.

Таблиця 5.2 – Рівні комерційного потенціалу розробки

Середньоарифметична сума балів	Рівень комерційного потенціалу
0-10	Низький
11-20	Нижче середнього
21-30	Середній
31-40	Вище середнього
41-50	Високий

Відповідно до таблиці 5.2, рівень комерційного потенціалу розробки методу та засобу підвищення стійкості паролів вище середнього.

Оцінювання рівня якості розробки методу та засобу підвищення стійкості паролів здійснюється з метою порівняльного аналізу та визначення найбільш ефективного, з технічної точки зору, варіанта інженерного рішення.

Рівень якості – кількісна характеристика міри придатності певного виду продукції для задоволення конкретного попиту на неї при порівнянні з відповідними базовими показниками за фіксованих умов споживання.

Абсолютний рівень якості розробки методу та засобу підвищення стійкості паролів потрібно знайти за обчисленням обраних для вимірювання показників, при цьому не порівнюючи їх із відповідними показниками аналогічних засобів.

Для цього було обрано низку параметрів: варіативність паролів після генерування, обсяг даних для перевірки, кількість взаємодій між компонентами одного застосунку. Визначаємо величину параметрів якості в балах та встановлюємо граничні значення (гірші, кращі, середні). Дані для кожного параметра представлено у таблиці 5.3

Таблиця 5.3 – Основні параметри методу та засобу підвищення стійкості паролів

Параметри	Абсолютне значення параметру			Коефіцієнт вагомості параметру
	Краще +5…+4	Середнє +3	Гірше +1…+2	
Відповідність паролів певним вимогам стійкості	4			0,6
Варіативність паролів після генерування		3		0,3
Кількість взаємодій між компонентами одного застосунку			2	0,1

Із врахування коефіцієнтів вагомості відповідних параметрів можна визначити абсолютний рівень якості інноваційного рішення за формулою:

$$K_{я.а.} = \sum_{i=1}^{n} Рн_i * a_i, \text{ де} \quad (5.1)$$

$Рн_i$ – числове значення і-го параметра інноваційного рішення, n – кількість параметрів інноваційного рішення, що прийняті для оцінювання, аі – коефіцієнт вагомості відповідного параметра (сума коефіцієнтів вагомості всіх параметрів повинна дорівнювати 1).

Таким чином, абсолютний рівень якості методу та засобу підвищення стійкості паролів складає 3.7 бали. Тепер потрібно визначити відносний рівень якості та порівняти показники з абсолютними показниками якості найліпших аналогів (табл. 5.4).

Відносний рівень якості методу та засобу підвищення стійкості паролів визначаємо за формулою 5.2.

$$K_{я.в.} = \sum_{i=1}^{n} q_i * a_i, \text{ де} \quad (5.2)$$

Таблиця 5.4 – Порівняння основних параметрів методу та засобу підвищення стійкості паролів

Параметри	Варіанти		Відносний показник якості	Коефіцієнт вагомості параметра
	Базовий (конкурент)	Новий		
Відповідність паролів певним вимогам стійкості	4	4	2	0,6
Варіативність паролів після генерування	4	6	0,8	0,2
Кількість взаємодій між компонентами одного застосунку	3	3	1,6	0,1

За розрахунками відносний рівень якості методу та засобу підвищення стійкості паролів складає 1,9. Це означає, що нова розробка якісніша на 52% відносно товару-аналога.

У найширшому розумінні конкурентоспроможність товару – це можливість його успішного продажу на певному ринку і в певний проміжок часу. Водночас конкурентоспроможною можна вважати лише однорідну продукцію з технічними параметрами і техніко-економічними показниками, що ідентичні аналогічним показникам уже проданого товару. Для того, щоб високоякісний товар був одночасно і конкурентоспроможним, він має відповідати критеріям оцінювання споживачів конкретного ринку в конкретний період часу.

Дані для розрахунку загального показника конкурентоспроможності розробки необхідно занести до таблиці 5.5.

Таблиця 5.5 – Нормативні, технічні та економічні параметри засобу автентифікації з нульовим знанням та товару-конкурента

Параметри	Варіанти		Відносний показник якості	Коефіцієнт вагомості параметра
	Базовий (конкурент)	Новий		
ймовірність автентифікації порушника	2^{-20}	$2*2^{-20}$	2	0,6

Продовження таблиці 5.5

Параметри	Варіанти		Відносний показник якості	Коефіцієнт вагомості параметра
	Базовий (конкурент)	Новий		
швидкість автентифікації	0,25	0,2	0,8	0,2
кількість взаємодій	3	5	1,6	0,1
Ціна за продукт, грн	20000	15000	0,75	-

Загальний показник конкурентоспроможності розробки (K) з урахуванням вищезазначених груп показників визначаємо за формулою 5.3:

$$K = \frac{I_{Т.П.}}{I_{Е.П.}} = \frac{1{,}9}{0{,}75} = 2{,}53, \quad (5.3)$$

де $I_{Т.П.}$ – індекс технічних параметрів (відносний рівень якості інноваційного рішення); $I_{Е.П.}$ – індекс економічних параметрів розрахований нижче за формулою 5.4:

$$I_{Е.П.} = \frac{РН_{ЕІ}}{РБ_{ЕІ}} = \frac{15000}{20000} = 0{,}75, \quad (5.4)$$

де $РН_{ЕІ}$, $РБ_{ЕІ}$ – економічні параметри (ціна придбання та споживання товару) відповідно нового та базового товарів.

Згідно розрахунків загальний показник конкурентоспроможності 2,53, що свідчить про більшу конкурентну спроможність засобу автентифікації з нульовим знанням у порівнянні з товаром-аналогом.

5.2 Прогнозування витрат на виконання науково-дослідної та конструкторсько-технологічної роботи

5.2.1 Розрахунок витрат, що стосуються виконавців розробки методу та засобу підвищення стійкості паролів

Команда розробки методу та засобу підвищення стійкості паролів.

Основна заробітна плата для розробників (дослідників) Зо, якщо вони працюють в наукових установах бюджетної сфери визначається за формулою:

$$З_О = \frac{M}{T_р} t \quad (5.5)$$

де M – місячний посадовий оклад розробника;

T_p – кількість робочих днів у місяці, T_p = 22 дні; t – число днів роботи.

Таблиця 5.3 – Розрахунки основної заробітної плати спеціаліста

Працівник	Оклад М, грн.	Оплата за робочий день, грн.	Число днів роботи, t	Витрати на оплату праці, грн.
Керівник	25000	1136.36	7	7945.52
Розробник	20000	909.09	15	13648.5
Всього:				21594.02

Для робітників, які задіяні до даної розробки та працюють в наукових установах бюджетної форми за робочими професіями, заробітна плата розраховується за формулою:

$$З_P = \sum_{i=1}^{n} t_i \cdot C_i \quad (5.6)$$

де t_i – норма часу (трудомісткість) на виконання конкретної роботи, годин; n – число робіт по видах та розрядах; C_i – погодинна тарифна ставка робітника відповідного розряду, який виконує дану роботу. C_i визначається за формулою:

$$C_i = \frac{M_M * K_i}{T_p * T_{ЗМ}} \quad (5.7)$$

де M_M – розмір мінімальної заробітної плати за місяць, грн.; в 2021 році мінімальна заробітна плата становить – 6500 грн., K_i – тарифний коефіцієнт робітника відповідного розряду, T_p = 22 дні; $T_{зм}$ – тривалість зміни = 8 годин.

Таблиця 5.7 – Заробітна плата робітників

Найменування робіт	Трудомісткість, н-год.	Розряд роботи	Погодинна тарифна ставка	Тариф. коеф.	Величина, грн.
Розробка	8	5	56.88	1.54	455
Тестування	5	4	49.85	1.35	249.5
Впровадження	2	1	36.93	1	73.86
Всього					777.86

Потрібно розрахувати додаткову заробітну плату робітників. Вона розраховується як 10-12% від основної:

$$З_Д=(0,1...0,12)*З_О \quad (5.8)$$

$$З_Д=0.1*(З_О+З_Р)=0,1*(21594,02+777,86)=2167,188 \text{ (грн.)}$$

Нарахування на заробітну плату $Н_{ЗП}$ розраховується як 22% від суми основної та додаткової заробітної плати:

$$Н_{ЗП}=(З_О+З_Р+З_Д)*\frac{\beta}{100}=(21594,02+777,86+2167,18)*0,22=52599,78 \text{ (грн.)} \quad (5.9)$$

де $З_О$ – основна заробітна плата розробників, грн.;

$З_р$ – основна заробітна плата робітників, грн.;

$З_Д$ – додаткова заробітна плата розробників, грн.;

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

Розрахунок амортизаційних відрахувань виконується за такою формулою:

$$А=\frac{Ц\times Н_А}{100}*\frac{Т}{12} \quad (5.10)$$

де $Ц$ – балансова вартість обладнання, грн;

$Н_А$ – річна норма амортизаційних відрахувань. Для даного випадку можна прийняти, що $Н_А$ = 25%;

$Т$ – термін використання ($Т$=2 місяців);

$Т_В$ – корисний час використання ($Т_В$ для комп'ютера становить 4 роки).

Під час виконання розробки методики тестування безпеки веб-застосунку використовувався ноутбук ціною в 25000 грн. Амортизаційні відрахування для нього представлені у таблиці 5.8.

Таблиця 5.8 – Амортизаційні відрахування

Найменування	Ціна, грн.	Норма амортизації, %	Термін використання, міс.	Сума амортизації, грн.
Ноутбук	25000	25	2	1041,06
Всього	1041,06			

Витрати на силову електроенергію розраховуються за формулою:

$$В_Е=В*П*Ф*К_П \quad (5.11)$$

де $В$ – вартість 1кВт-години електроенергії (В=4,62 грн/кВт);

$П$ – установлена потужність комп'ютеру ($П$=0,74 кВт);

$Ф$ – фактична кількість годин роботи комп'ютеру ($Ф$=176=(7+15)*8 год);

$К_П$ – коефіцієнт використання потужності ($К_П$ <1, $К_П$ = 0,8).

Відповідно до формули 5.11 витрати на силову електроенергію:

$В_Е$=4,62*0,74*176*0,8=481,36 (грн.)

Інші витрати $В_{ін}$ можна прийняти як (100-300)% від суми основної заробітної плати розробників, які виконували роботу, тобто:

$$В_{ін}=1*(21594,02+777,86)=22371,88 \text{ (грн.)} \quad (5.11)$$

Сума усіх попередніх витрат дає загальні витрати на виконання роботи. Усі витрати складають:

В=21594,02+777,86+2167,18+481,36+5259,9+22371,88+1041,06 =52915,4 (грн.)

Розрахунок загальної вартості наукової розробки $В_{заг}$ за формулою:

$$В_{заг}=\frac{В}{\alpha}=52915,4 \quad (5.12)$$

де $\alpha=1$ – частка витрат, які безпосередньо здійснює виконавець даного етапу роботи, у відносних одиницях.

Прогнозування загальних витрат $ЗВ$ на виконання та впровадження результатів виконаної наукової роботи здійснюється за формулою:

$$ЗВ=\frac{В_{заг}}{\beta} \quad (5.13)$$

Розрахунок прогнозованих загальних витрат = 75593,42 (грн.)

5.2.2 Розрахунок собівартості розробки методики тестування безпеки для веб-застосунків

Витрати на силову електроенергію розраховуються за формулою:

$$В_Е=В*П*Ф*К_П \quad (5.14)$$

де $В$ – вартість 1кВт-години електроенергії (В=4,62 грн/кВт);

$П$ – установлена потужність комп’ютеру ($П$=0,74 кВт);

$Ф$ – фактична кількість годин роботи комп’ютеру ($Ф$=176=(7+15)*8 год);

$К_П$ – коефіцієнт використання потужності ($К_П$ <1, $К_П$ = 0,8).

Відповідно до формули 5.11 витрати на силову електроенергію:

$В_Е$=4,62*0,74*176*0,8=481,36 (грн.)

Для робітників, які задіяні до даної розробки та працюють в наукових установах бюджетної форми за робочими професіями, заробітна плата розраховується за формулою:

$$З_Р = \sum_{i=1}^{n} t_i \cdot C_i \quad (5.15)$$

де t_i – норма часу (трудомісткість) на виконання конкретної роботи, годин; n – число робіт по видах та розрядах; C_i – погодинна тарифна ставка робітника відповідного розряду, який виконує дану роботу. C_i визначається за формулою:

$$C_i = \frac{M_M * K_i}{T_p * T_{ЗМ}} \quad (5.16)$$

де M_M – розмір мінімальної заробітної плати за місяць, грн.; в 2021 році мінімальна заробітна плата становить – 6500 грн., K_i – тарифний коефіцієнт робітника відповідного розряду, T_p = 22 дні; $T_{зм}$ – тривалість зміни = 8 годин.

Таблиця 5.9 – Заробітна плата робітників

Найменування робіт	Трудомісткість, н-год.	Розряд роботи	Погодинна тарифна ставка	Тариф. коеф.	Величина, грн.
Розробка	8	5	56.88	1.54	455
Тестування	5	4	49.85	1.35	249.5
Впровадження	2	1	36.93	1	73.86
Всього					777,86

Потрібно розрахувати додаткову заробітну плату робітників. Вона розраховується як 10-12% від основної:

$$З_Д = (0,1...0,12) * З_О \quad (5.17)$$

$$З_Д = 0.1 * (З_О + З_Р) = 0,1 * 777,86 = 77,786 \text{ (грн.)}$$

Нарахування на заробітну плату $Н_{ЗП}$ розраховується як 22% від суми основної та додаткової заробітної плати:

$$Н_{ЗП} = (З_О + З_Р + З_Д) * \frac{\beta}{100} = (777,86 + 77,786) * 0,22 = 188,24 \text{ (грн.)} \quad (5.18)$$

де $З_О$ – основна заробітна плата розробників, грн.;

$З_р$ – основна заробітна плата робітників, грн.;

$З_Д$ – додаткова заробітна плата розробників, грн.;

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

Для розрахунку загальновиробничих витрат з рахунку на одиницю продукції буде використана наступна формула, яка відображає нормативи відносно до основної заробітної плати основних робітників, які виготовляють продукцію:

$$ЗВВ=Н_В*З_О \text{ (грн.)} \quad (5.19)$$

Норматив загальновиробничих витрат для програмних продуктів становить 230-270%.

ЗВВ=2,7*777,86=2100,22 (грн.)

Сума попередніх витрат утворює виробничу собівартість розробки.

$$S_В = 481,36+777,86+77,78+188,24+2100,22=3625,46 \text{ (грн.)} \quad (5.20)$$

5.3 Прогнозування комерційних ефектів від реалізації результатів розробки

Ціна – це грошовий вираз вартості товару (продукції, послуги). Вона завжди коливається навколо ціни виробництва (перетвореної форми вартості одиниці товару, що дорівнює сумі витрат виробництва й середнього прибутку) та відображає рівень суспільне необхідних витрат праці.

Виходячи з того, що розробки, як правило, приймаються та впроваджуються за завданням замовника, або коли результатом розробки є продукція, що підлягає державному регулюванню, то нижню межу ціни реалізації розробки можна розрахувати за формулою 5.21:

$$Ц = S_В \cdot (1+\frac{P}{100}) \cdot (1+\frac{\omega}{100}), \quad (5.21)$$

де $S_В$ – виробнича собівартість інноваційного рішення, грн.;

P – норматив рентабельності узгоджений із замовником або встановлений державою, (Р=30…60%);

ω – ставка податку на додану вартість, % (з осені 2021 року $\omega = 20\%$).

$$Ц = 4185 \cdot (1+\frac{60}{100}) \cdot (1+\frac{20}{100}) = 8035,2 \text{ (грн.)} \quad (5.21)$$

Із врахуванням коефіцієнта якості ціна розробки становить 28123 грн.

Чистий прибуток від реалізації розробки можна розрахувати за формулою:

$$П = (Ц - \frac{(Ц - МР)\cdot f}{100} - S_В - \frac{q\cdot S_В}{100})\cdot(1-\frac{h}{100})\cdot РП, \quad (5.22)$$

де *Ц* – ціна розробки, грн.; *МР* – вартість матеріальних та інших ресурсів, що були придбані виробником для виготовлення розробки (*МР*=(0,1…0,2) *Ц*), грн.;

f – зустрічна ставка податку на додану вартість, %; $S_В$ – виробнича собівартість розробки, грн.; *q* – норматив, який визначає величину адміністративних витрат, витрат на збут та інші операційні витрати, % (рекомендовано *q*=5…10%); *h* – ставка податку на прибуток, %, *РП* – прогнозований попит продажів.

$$П = (28123 - \frac{(28123 - 28123\cdot 0{,}2)\cdot 14}{100} - 4377{,}5 - \frac{5\cdot 4377{,}5}{100})\cdot(1-\frac{18}{100})\cdot 2 = 33418 \text{ (грн.)}, \quad (5.23)$$

Прогнозований чистий прибуток від реалізації розробки складає 33418 грн.

5.4 Розрахунок ефективності вкладених інвестицій та період їх окупності

Розрахунок ефективності вкладених інвестицій передбачає проведення таких робіт:

Термін окупності вкладених у реалізацію наукового проекту інвестицій розраховано за формулою 5.24:

$$T_{ОК} = \frac{ЗВ}{П} = \frac{75593{,}42}{33418} = 2{,}26 \text{ (роки)} \quad (5.24)$$

Оскільки $T_{ОК} < 3$ років, то фінансування наукової розробки методу та засобу автентифікації користувачів із нульовим знанням є доцільним.

5.5 Висновки розділу

Отже, у цьому розділі виконано обґрунтування економічної доцільності проведення наукового дослідження та розробки методу та засобу підвищення стійкості паролів.

Рівень комерційного потенціалу розробки методу та засобу підвищення стійкості паролів вище середнього.

На основі параметрів засобу підвищення стійкості паролів визначено абсолютний рівень якості методу та засобу підвищення стійкості паролів який складає 3,7 бали, при цьому відносний рівень якості розробки становить 1,9.

Також, було розраховано загальний показник конкурентоспроможності, який показав що дана методика є конкурентоспроможною, відображається в оцінці 2,53 бали. Загальні витрати для розробки методики, що стосуються виконавців розробки, склали 75593,42 грн, а собівартість розробки – 3625,46 грн.

Також, було розраховано прогнозований чистий прибуток розробки, який склав 33718 грн, та мінімальну ціну розробки 28123. На основі отриманих даних було розраховано термін окупності вкладених інвестицій, який складає 2,26 роки, що є меншим значенням за 3, а тому фінансування розробки методу та засобу підвищення стійкості паролів є доцільним.

ВИСНОВКИ

В результаті аналізу розглянуто методи автентифікації та визначено, що найпоширенішим та найзручнішим є метод парольної автентифікації. При правильному використанні паролі можуть забезпечити прийнятний для багатьох систем рівень безпеки. Аналіз відомих методів генерування паролів показав, що метод генерування паролів завдяки асоціаціям легший для запам’ятовування користувачам, але також не несе за собою необхідної стійкості. Розглянуто відомі методи зламу паролів, де визначено що найвідомішими методами є підбір геш значення паролів, атака за словником та атака підбором. Проведений аналіз на визначення генератора, що дасть можливість рівномірного генерування чисел в довільному діапазоні чисел для рівномірного розподілу в методах розмивання паролів. Обрано удосконалений лінійний конгруентний генератор, що володіє такою властивістю.

Розроблено математичний опис використовуючи теоретико-множинний підхід, що дозволило розглянути з чого складається процес автентифікації та описати парольну автентифікацію. Розроблено загальну структуру засобу підвищення стійкості паролів та описано методи, що будуть використовуватися для розмивання паролів. Описано блок перевірки паролів, що дозволить відкинути найслабші варіанти можливих паролів та задати певне мінімальне значення стійкості пароля.

Створено загальну архітектуру засобу підвищення стійкості паролів, що показує основні модулі програми та їх взаємодію між собою. Розроблено загальний алгоритм роботи застосунку. Створено алгоритми методів розмивання паролів, генерування псевдовипадкового числа та методу перевірки згенерованих паролів.

Розглянуто можливі засоби реалізації модуля авторизації користувачів. Обрано мову програмування Java, що дозволить використовувати застосунок на різних операційних системах, JavaFX та SceneBuilder, що дозволять створити зрозумілий графічний інтерфейс. Розроблено застосунок, який реалізує засіб підвищення стійкості паролів на основі розроблених алгоритмів.

Розроблено план тестування у вигляді чек-листа, що дозволило обрати відповідний функціонал та визначити функції, що буде перевірено, задля правильної роботи програми. Для проведення тестування розроблено два тестові сценарії, що представляє собою послідовність дій користувача для двох результатів тестування – позитивного і негативного. Виконано функціональне тестування, яке показало що всі функції програми працюють належним чином. Проведено блокове тестування, результати якого показали що методи розмивання паролів та генерування псевдовипадкових чисел працюють коректно.

Розроблено алгоритм тестування методу заміни літер на спеціальні символи. Проведено тестування методу за даним алгоритмом. Результати показали, що метод працює коректно, а пропорції частоти для заміни кожного символа наближені до заданих.

Виконано обґрунтування економічної доцільності проведення наукового дослідження та розробки методу та засобу підвищення стійкості паролів. Рівень комерційного потенціалу розробки методу та засобу підвищення стійкості паролів вище середнього. На основі параметрів засобу підвищення стійкості паролів визначено абсолютний рівень якості методу та засобу підвищення стійкості паролів який складає 3,7 бали, при цьому відносний рівень якості розробки становить 1,9. Також, було розраховано загальний показник конкурентоспроможності, який показав що дана методика є конкурентоспроможною, відображається в оцінці 2,53 бали. Загальні витрати для розробки методики, що стосуються виконавців розробки, склали 75593,42 грн, а собівартість розробки – 3625,46 грн.

Реалізований засіб підвищення стійкості паролів ускладнює атаки грубої сили та атаки за словником при цьому, залишаючи пароль простим для запам’ятовування користувачем.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. R. Housley, B. Aboba, Guidance for Authentication, Authorization, and Accounting (AAA) Key Management. – 2007. – 23 с.

2. Nilesh A. Lal, Salendra Prasad, Mohammed Farik, A Review Of Authentication Methods. – 2016. P. 246-249. URL: https://www.researchgate.net/profile/Mohammed-Farik/publication/311514269_A_Review_Of_Authentication_Methods/links/584fbed808aed95c250b4915/A-Review-Of-Authentication-Methods.pdf (дата звернення 24.11.2021)

3. B. Schneier, Schneier on Security. URL: https://www.schneier.com/blog/archives/2010/02/man-in-the-midd_1.html. (дата звернення 25.11.2021)

4. Biometric Security. URL: http://www.bioelectronix.com/what_is_biometrics.html (дата звернення 25.11.2021)

5. H. Spector, Techwalla. URL: https://www.techwalla.com/articles/what-are-theadvantages-disadvantages-of-a-digital-certificate. (дата звернення 26.11.2021)

6. N. Haller, C. Metz, P. Nesser, M. Straw. A One-Time Password System». – 1998. – 25 с. URL: https://www.hjp.at/doc/rfc/rfc2289.html (дата звернення 26.11.2021)

7. Как пользователи воспринимают разные методы аутентификации URL: https://habr.com/ru/post/344406/ (дата звернення 27.11.2021)

8. CrackStation. URL: https://crackstation.net/hashing-security.htm (дата звернення 28.11.2021)

9. Openwall. URL: http://www.openwall.com/john/ (дата звернення 28.11.2021)

10. А.А. Калинин, «Слоговой метод генерации паролей». URL: http://cdosfera.rinethost.ru/userfiles/Aktyalnie_problemi.pdf#page=73 (дата звернення 30.11.2021)

11. В.Б. Лим, «Создание надёжных паролей». URL: https://cyberleninka.ru/article/n/sozdanie-nadezhnyh-paroley/viewer (дата звернення 30.11.2021)

12. Використання менеджера паролів URL: https://eset.ua/ua/blog/view/79/ispolzovaniye-menedzhera-paroley-zachem-nuzhen-i-kak-pravilno-vybrat (дата звернення 30.11.2021)

13. Сравнение встроенных в браузеры менеджеров паролей URL: https://www.anti-malware.ru/compare/Password-Managers#part5 (дата звернення 30.11.2021)

14. Бурячок В. Л., Платоненко А. В., Семко О. В. Вибір раціонального способу генерування паролів серед множини існуючих. Київ, 2019. – 6 с.

15. Баришев Ю. В. Методи формування псевдовипадкових чисел для псевдонедетермінованих геш-функцій. Баришев Ю. В., Кравчук Т. А.: тези доповідей П'ятої Міжнародної науково-практичної конференції «Методи та засоби кодування, захисту й ущільнення інформації» – Вінниця : ВНТУ, 2016. – С. 58-60.

16. Харин Ю.С. Математические и компьютерные основы криптологии: учеб. пособие / Ю.С. Харин, В.И. Берник, Г.В. Матвеев, С.В. Агиевич // МН: Новое знание – 2003. – 193 с.

17. Баришев Ю. В. Метод зав'язування блоків даних для псевдонедетермінованого гешування. Баришев Ю. В., Лавренюк Т. А.: матеріали XLVI науково-технічної конференції підрозділів ВНТУ, Вінниця, - 2017. – 2 с

18. О. В. Кохан, Ю. В. Баришев. Засіб генерування стійкого пароля.: матеріали XLIX науково-технічної конференції підрозділів ВНТУ, Вінниця, 2020 р.– 3с

19. Томашевський В. М. Моделювання систем. – Київ : Видавнича група BHV, 2005. – 352 с

20. Advantages and Disadvantages of Java. URL: https://techvidvan-com.translate.goog/tutorials/pros-and-cons-of-java/?_x_tr_sl=auto&_x_tr_tl=ru&_x_tr_hl =ru&_x_tr_pto=op (дата звернення 10.12.2021)

21. What Does JavaFX Mean For You? URL: https://dzone.com/articles/what-does-javafx-mean-you (дата звернення 10.12.2021)

22. JavaFX Scene Builder. URL: https://www.oracle.com/java/technologies/javase/javafxscenebuilder-info.html (дата звернення 10.12.2021)

23. Що таке тест-план, для чого він потрібен і з чого складається. URL: https://training. qatestlab.com/blog/technical-articles/test-plan/ (дата звернення 15.12.2021)

24. Чек-листы в тестировании: что нужно знать тестировщику! URL: https://qualitica.ru/blog/chek-list/ (дата звернення 15.12.2021)

25. Що таке чеклісти та як з ними працювати. URL: https://training.qatestlab.com /blog/technical-articles/work-with-checklist/ (дата звернення 16.12.2021)

26. Тестовый сценарий. URL: https://coderlessons.com/tutorials/kachestvo-programmnogo-obespecheniia/ruchnoe-testirovanie/testovyi-stsenarii (дата звернення 16.12.2021)

27. Что такое тестовый сценарий? URL: https://ru.education-wiki.com/2263266-what-is-test-scenario (дата звернення 17.12.2021)

28. Виды тестирования и подходы к их применению. URL: https://habr.com/ru/post/81226/ (дата звернення 17.12.2021)

29. Тестирование программы, Junit. URL: http://java-online.ru/blog-junit.xhtml (дата звернення 18.12.2021)

30. JUnit — Краткое руководство. URL: https://coderlessons.com/tutorials/java-tekhnologii/vyuchit-junit/junit-kratkoe-rukovodstvo (дата звернення 18.12.2021)

ДОДАТКИ

Міністерство освіти і науки України
Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра захисту інформації

ЗАТВЕРДЖУЮ
Завідувач кафедри ЗІ д. т. н., проф.
_____В. А. Лужецький
«____»_____2021 р.

ТЕХНІЧНЕ ЗАВДАННЯ

на виконання магістерської кваліфікаційної роботи

на тему: «МЕТОД ТА ЗАСІБ ПІДВИЩЕННЯ СТІЙКОСТІ ПАРОЛІВ»
08-20.МКР.005.00.000 ТЗ

Керівник магістерської кваліфікаційної роботи
к. т. н., доцент каф. ЗІ Баришев Ю. В.

Вінниця 2021

1 Підстави для проведення робіт

Робота проводиться на підставі наказу ректора ВНТУ від 1 вересня 2021 року № 207.

Дата початку роботи 01.09.2021 р.

Дата закінчення роботи 23.12.2021 р.

2 Мета та призначення НДР

Метою дипломної роботи є удосконалення методів генерування стійких паролів.

Об'єктом дослідження є процес автентифікації користувачів в цифровому середовищі.

Предметом є методи автентифікації користувачів в системі.

Актуальність теми. Разом із активним переходом людства до комп'ютерних технологій та перенесенням інформації в цифровий світ, зростає важливість питання обмеження доступу до персональних даних користувачів та варіанти захисту цих даних. Оскільки разом з розвитком інформаційних технологій, розвиваються і методи, що дозволяють отримати несанкціонований доступ до приватної інформації. Одним з варіантів покращення захисту інформації користувача є використання стійких паролів в процесі їх авторизації, що дозволить збільшити час підбору пароля зловмисником та зменшить рентабельність зламу.

Проте чим більша логічна незалежність символів паролю один від одного і їх випадковість, тим важче користувачам запам'ятовувати і використовувати таке ключове слово. В результаті чого надається перевага поширеним паролям, що містять очевидний вміст (дату народження, імена членів сім'ї). Відповідно виникає необхідність поєднання цих двох факторів, щоб знайти середнє значення та дати можливість користувачам генерувати стійких пароль з можливістю запам'ятати його.

3 Вихідні дані для проведення НДР

3.1 R. Housley, B. Aboba, Guidance for Authentication, Authorization, and Accounting (AAA) Key Management. – 2007. – 23 с.

3.2 Nilesh A. Lal, Salendra Prasad, Mohammed Farik, A Review Of Authentication Methods. – 2016. P. 246-249.

3.3 Баришев Ю. В. Метод зав'язування блоків даних для псевдонедетермінованого гешування. Баришев Ю. В., Лавренюк Т. А.: матеріали XLVI науково-технічної конференції підрозділів ВНТУ, Вінниця, - 2017. – 2 с

3.4 О. В. Кохан, Ю. В. Баришев. Засіб генерування стійкого пароля.: матеріали XLIX науково-технічної конференції підрозділів ВНТУ, Вінниця, 2020 р.– 3с

4 Виконавці НДР

Студент групи 1БС-20м Кохан Олександр Володимирович

5 Вимоги до виконання НДР

Для підвищення стійкості генерування паролів виконати наступні завдання:

- проаналізувати методи автентифікації користувачів в інформаційних системах;
- проаналізувати існуючі методи підвищення стійкості паролів;
- провести дослідження для визначення методів розмиття паролів;
- розробити метод генерування стійких паролів;
- розробити алгоритм засобу генерування паролів;
- розробити та протестувати засіб генерування стійких паролів.

6 Вимоги до супровідної документації

Графічна і текстова документація повинна відповідати діючим стандартам України.

7 Стадії та етапи розробки

Етап	Зміст	Початок	Закінчення	Результат
1	Аналіз завдання	01.09.2021	04.09.2021	Вступ, технічне завдання
2	Аналіз джерел та методів захисту	05.09.2021	05.10.2021	Звіт з аналізу літературних джерел
3	Обгрунтування програмних засобів реалізації, створення початкового варіанту програми, проведення тестувань	06.10.2021	24.10.2021	Схема структурна та алгоритм засобу підвищення стійкості паролів.

Етап	Зміст	Початок	Закінчення	Результат
4	Створення кінцевого варіанту програми, тестування, розробка інструкцій по роботі з програмою	25.10.2021	17.11.2021	Діюча програма і остаточний звіт у вигляді пояснювальної записки

8 Очікувані результати та порядок реалізації НДР

Передбачається розробка нових (удосконалення існуючих) методів підвищення стійкості паролів. Заплановане створення програмного засобу.

9 Матеріали які подаються після закінчення НДР

По завершенню роботи подається пояснювальна записка та ілюстративна частина.

10Порядок приймання НДР та її етапів

Апробація на науково-технічних конференціях та семінарах. Результати роботи будуть розглядатися на засіданні ДЕК із захисту магістерських кваліфікаційних робіт.

Попередній захист та доопрацювання МКР грудень 2021 р.

Представлення МКР до захисту 5 грудня 2021 р.

Захист МКР 21-23 грудня 2021 р.

11Вимоги до розроблення документації

Документація буде виконуватись за допомогою комп’ютерного набору у відповідності вимог ДСТУ 3008:2015 «Інформація та документація. Звіти у сфері науки і техніки. Структура та правила оформлювання».

12Вимоги щодо технічного захисту інформації з обмеженим доступом

У зв’язку з тим, що дана робота не містить інформації, що потребує захисту у відповідності до законів України, заходи з її технічного захисту не передбачаються.

Розробив студент групи 1БС-20м _____Кохан О. В.

Додаток Б

Код засобу

Метод ReplacingLettersSymbols

```
  public static String ReplacingLettersSymbols (String word, int n){
    char[] wordChars = word.toCharArray();
    StringBuffer newPassword = new StringBuffer();
    for(int i = 0; i<word.length(); i++) {
        if(Rand_N()<n){
            switch (wordChars[i]){
                case 'a': if(Rand_N()<89){newPassword.append("@");} else
newPassword.append("/\\"); break;
                case 'b': if(Rand_N()<64){newPassword.append("8");} else
newPassword.append("|3");break;
                case 'c': if(Rand_N()<90){newPassword.append("(");} else
newPassword.append(wordChars[i]);break;
                case 'h': if(Rand_N()<70){newPassword.append("#");} else
newPassword.append("]-[");break;
                case 'i': if(Rand_N()<53){newPassword.append("1");} else
newPassword.append("!");break;
                case 'k': if(Rand_N()<67){newPassword.append("|<");} else
newPassword.append("1<");break;
                case 'm': if(Rand_N()<58){newPassword.append("/\\/\\");} else
newPassword.append("/v\\");break;
                case 'o': if(Rand_N()<78){newPassword.append("0");} else
newPassword.append("()");break;
                case 'q': if(Rand_N()<73){newPassword.append("0_");} else
newPassword.append(wordChars[i]);break;
                case 's': if(Rand_N()<89){newPassword.append("$");} else
newPassword.append("5");break;
                case 'w': if(Rand_N()<60){newPassword.append("vv");} else
newPassword.append("\\/\\/");break;
                case 'l': if(Rand_N()<55){newPassword.append("|_");} else
newPassword.append(wordChars[i]);break;
                case 'v': if(Rand_N()<80){newPassword.append("\\/");} else
newPassword.append(wordChars[i]);break;
                case 'z': if(Rand_N()<63){newPassword.append("2");} else
```

```
newPassword.append(wordChars[i]);break;
                default:
                    newPassword.append(wordChars[i]);break;
            }
        } else newPassword.append(wordChars[i]);
    }
    return newPassword.toString();
}
```

Метод Rand_N

```
public static long seed = System.currentTimeMillis();
public static int Rand_N (){
    long m = 2147483647, a = 2622621, c = 11;
    seed = (seed * a + c) % m;
    if(seed<m-(m%100)) seed =seed%100;
    return (int) seed;
}
```

Додаток В

Код тестування

Клас ParametersTest

```
import org.junit.Assert;
import org.junit.Test;
import sample.SignUpController;
import java.security.NoSuchAlgorithmException;

public class SignUpControllerTest extends SignUpController{
    SignUpController test = new SignUpController();

      @Test
    public void Rand_N() {
        int n = test.Rand_N(123456789);
        Assert.assertNotEquals(n, test.Rand_N(n));
    }

    @Test
    public void createHash () throws NoSuchAlgorithmException {

        final String expected =
"43d05d6de8e0ea256293621b1c738c8d62ee68b3be47349121b41b4299e131d4";
        final String actual = test.createHash("communication");
        Assert.assertEquals(expected, actual);
    }

    @Test
    public void createHash () throws NoSuchAlgorithmException {

        final String expected = test.createHash("communicationn");
        final String actual = test.createHash("communication");
        Assert.assertNotEquals(expected, actual);
    }

    @Test
    public void RandPassword ()  {
        final String word = "password";
        Assert.assertNotEquals(test.RandPassword(word, 40),
test.RandPassword(word, 40));
    }

      @Test
          public void RandPassword ()  {
              final String word = "password";
              Assert.assertNotEquals(test.RandPassword(word, 40), word)
    }}
```

Клас Test

```
public class Main {
    public static void main(String[] args) {
        for (int i = 0; i < 1000; i++) {
            RandPassword("w", 50);
        }
        System.out.println("First: " + res[0]);
        System.out.println("Sec: " + res[1]);
        int a = res[0]+res[1];
        System.out.println("Sum " + a);
        float b = res[0]/a;
        System.out.println(b);
    }
    public static int[] res = new int[] { 0, 0};
    public static long seed = System.currentTimeMillis();
    public static int Rand_N (){
        long m = 2147483647, a = 2622621, c = 11;
        seed = (seed * a + c) % m;
        if(seed<m-(m%100)) seed =seed%100;
        return (int) seed;
    }
    public static String RandPassword (String word, int n){
        char[] wordChars = word.toCharArray();
        StringBuffer newPassword = new StringBuffer();
        for(int i = 0; i<word.length(); i++) {
            if(Rand_N()<n){
                switch (wordChars[i]){
                    case 'a':
if(Rand_N()<89){newPassword.append("@");res[0]++;} else
{newPassword.append("/\\");res[1]++;} break;
                    case 'b':
if(Rand_N()<64){newPassword.append("8");res[0]++;} else
{newPassword.append("|3");res[1]++;}break;
                    case 'c':
if(Rand_N()<90){newPassword.append("(");res[0]++;} else
{newPassword.append(wordChars[i]);res[1]++;}break;
```

```
                    case 'h':
if(Rand_N()<70){newPassword.append("#");res[0]++;} else
{newPassword.append("]-[");res[1]++;}break;
                    case 'i':
if(Rand_N()<53){newPassword.append("1");res[0]++;} else
{newPassword.append("!");res[1]++;}break;
                    case 'k':
if(Rand_N()<67){newPassword.append("|<");res[0]++;} else
{newPassword.append("1<");res[1]++;}break;
                    case 'm':
if(Rand_N()<58){newPassword.append("/\\/\\");res[0]++;} else
{newPassword.append("/v\\");res[1]++;}break;
                    case 'o':
if(Rand_N()<78){newPassword.append("0");res[0]++;} else
{newPassword.append("()");res[1]++;}break;
                    case 'q':
if(Rand_N()<73){newPassword.append("0_");res[0]++;} else
{newPassword.append(wordChars[i]);res[1]++;}break;
                    case 's':
if(Rand_N()<89){newPassword.append("$");res[0]++;} else
{newPassword.append("5");res[1]++;}break;
                    case 'w':
if(Rand_N()<60){newPassword.append("vv");res[0]++;} else
{newPassword.append("\\/\\/");res[1]++;}break;
                    case 'l':
if(Rand_N()<55){newPassword.append("|_");res[0]++;} else
{newPassword.append(wordChars[i]);res[1]++;}break;
                    case 'v':
if(Rand_N()<80){newPassword.append("\\/");res[0]++;} else
{newPassword.append(wordChars[i]);res[1]++;}break;
                    case 'x':
if(Rand_N()<63){newPassword.append("}{");res[0]++;} else
{newPassword.append("><");res[1]++;}break;
                    default:newPassword.append(wordChars[i]);break;}
            } else newPassword.append(wordChars[i]);}
        return newPassword.toString();
    }}
```

Додаток Г

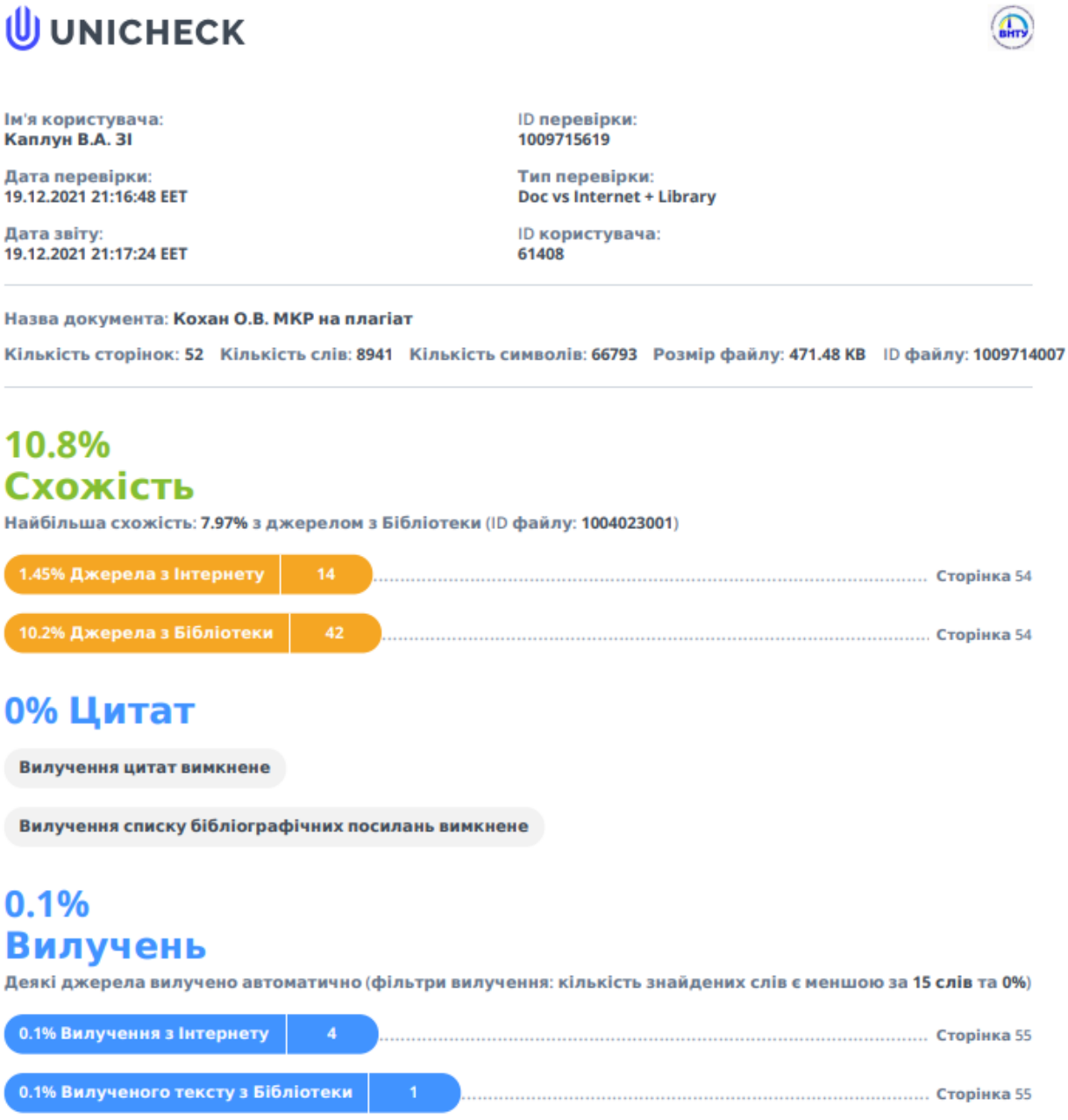
Критерії оцінювання комерційного потенціалу розробки

Критерії оцінювання та бали (за 5-ти бальною шкалою)					
Критерій	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 аналогам	Ціна продукту дещо нижче за ціни аналогів	Ціна продукту значно нижче за ціни аналогів
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	Експлуатаційні витрати значно вищі, ніж в аналогів	Експлуатаційні витрати дещо вищі, ніж в аналогів	Експлуатаційні витрати на рівні витрат аналогів	Експлуатаційні 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

Критерії оцінювання та бали (за 5-ти бальною шкалою)					
Критерій	0	1	2	3	4
8	Відсутні фахівці як з технічної, так і з комерційної реалізації ідеї	Необхідно наймати фахівців або витрачати значні кошти та час на навчання 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. Джерела фінансування 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 більше 10-ти років	Термін реалізації ідеї від 3-х до 5-ти років. Термін окупності більше 5-ти р	Термін реалізації ідеї менше 3-х років. Термін окупності від 3-х до 5-ти р.	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 кількості дозвільних документів на виробництво 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

Додаток Д

Результати перевірки роботи на плагіат



UNICHECK

Ім'я користувача:
Каплун В.А. ЗІ

ID перевірки:
1009715619

Дата перевірки:
19.12.2021 21:16:48 EET

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
19.12.2021 21:17:24 EET

ID користувача:
61408

Назва документа: Кохан О.В. МКР на плагіат

Кількість сторінок: 52 Кількість слів: 8941 Кількість символів: 66793 Розмір файлу: 471.48 KB ID файлу: 1009714007

10.8%
Схожість

Найбільша схожість: 7.97% з джерелом з Бібліотеки (ID файлу: 1004023001)

1.45% Джерела з Інтернету 14 ..... Сторінка 54

10.2% Джерела з Бібліотеки 42 ..... Сторінка 54

0% Цитат

Вилучення цитат вимкнене

Вилучення списку бібліографічних посилань вимкнене

0.1%
Вилучень

Деякі джерела вилучено автоматично (фільтри вилучення: кількість знайдених слів є меншою за 15 слів та 0%)

0.1% Вилучення з Інтернету 4 ..... Сторінка 55

0.1% Вилученого тексту з Бібліотеки 1 ..... Сторінка 55

Назва документа: **Кохан О.В. МКР на плагіат**

ID файлу: **1009714007**

Схожість

Джерела з Інтернету 14

3	https://hyperhost.ua/info/uk/navishcho-vikoristovuvati-generator-paroliv-5-prichin	0.56%
5	http://docplayer.net/36229840-Moloda-nauka-2012-tom-i.html	0.45%
8	https://iq.vntu.edu.ua/fdb/1888/%D0%93%D1%83%D1%86%D0%B0%D0%BB%D1%8E%D0%BA_%D0%9E.%D0%92./are 11 джерел	0.25%
10	http://www.dut.edu.ua/uploads/l_1365_25736977.doc	0.2%

Джерела з Бібліотеки 42

1	**БДР_Кохан_Р** ID файлу: **1004023001** Навчальний заклад: **Vinnytsia National Technical University**		7.97%
2	**Магістерська Алла Штокал-на плагіат (1)** ID файлу: **1008247529** Навчальний заклад: **Vinnytsia National Tec...**		0.66%
4	**МКР_Середа** ID файлу: **1009655065** Навчальний заклад: **Vinnytsia National Technical University**		0.47%
6	**Панченко** ID файлу: **1009708141** Навчальний заклад: **Vinnytsia National Technical University**	30 Джерело	0.45%
7	**144_MKR_Mazur_VI_2021** ID файлу: **1008144564** Навчальний заклад: **Vinnytsia National Technical Univer**	6 Джерело	0.4%
9	**БДР_2017_Лавренюк** ID файлу: **2017387** Навчальний заклад: **Vinnytsia National Technical University**		0.22%
11	**ЗІ_Рудик** ID файлу: **1000764602** Навчальний заклад: **Vinnytsia National Technical University**	2 Джерело	0.2%

Назва документа: **Кохан О.В. МКР на плагіат**

ID файлу: **1009714007**

Вилучення

Вилучення 4

http://www.rusnauka.com/9_NND_2014/Informatica/4_163542.doc.htm — 3 джерела — 0.12%

https://iq.vntu.edu.ua/repository/getfile.php/1745.pdf — 0.1%

Вилучення по Бібліотеці акаунту 1

073СахаровВВ2019 ID файлу: **1000713592** Навчальний заклад: **Vinnytsia National Technical University** — 0.1%

ІЛЮСТРАТИВНА ЧАСТИНА

РЕЗУЛЬТАТ АНАЛІЗУ МЕТОДІВ ГЕНЕРУВАННЯ ПАРОЛІВ

Назва методу	Суть методу	Характеристика
Спосіб заміни слів	базується на написанні паролю літерами з іншої розкладки клавіатури	нестійкий, оскільки більшість часто вживаних слів на певній мові використовуються для створення словників паролів
Спосіб випадкового натискання клавіш	базується на натисканні клавіш у певному порядку, не відриваючи руки	нестійкий, оскільки словники із «змієподібними» комбінаціями паролів (введеними затиснутим пальцем за певним маршрутом на клавіатурі) також існують
Спосіб «обчислюваних людиною» паролів	базується на формуванні матриці з символами, наприклад назви ресурсу, та подальшої заміни обраних символів, за певним правилом	може бути зручним для використання захисту онлайн-ресурсів, але генерування паролю для бездротової мережі з 63 символів таким способом не дасть необхідної стійкості та займе багато часу
Спосіб генерування паролів завдяки асоціаціям	базується на застосуванні технології асоціацій та замін певних літер символами	може бути легшим для запам’ятовування, але також не несе за собою необхідної стійкості
Спосіб створення паролів з декількох слів	базується на логічному поєднанні слів із зміною певних символів	може дати більш складний пароль, але його стійкість, також не можна вважати високою, а час на генерування може бути великим
Спосіб створення паролів завдяки генеруванню випадкових чисел	базується на використанні гральних кубиків для генеруванні випадкових чисел та подальшого вибору слів за таблицею	може займати менше часу для генерування паролю, але стійкість створеного паролю важко вважати високою, оскільки можливе використання словників для підбору
Спосіб генерування випадкових паролів	базується на генеруванні паролів завдяки спеціальним командам та додаткам в операційній системі	може дати складніший результат та потребує набагато меншу часу для створення, але питання випадкового генерування такого ж самого паролю зловмисником, хоча й малоймовірне, але все ж таки можливе

					08–20.МКР.005.00.000 ІЧ1		
Змн.	Арк.	№ докум.	Підпис	Дата			
Розроб..		Кохан О.В.			Метод та засіб підвищення стійкості паролів. Результат аналізу методів генерування паролів	Літ.	Арк. 1 / Аркушів 1
Перевір.		Баришев Ю.В.					
Реценз.		Савицька Л.А.				ВНТУ, 1БС-20м	
Н. Контр.		Баришев Ю.В.					
Затверд.		Лужецький В.А.					

МАТЕМАТИЧНИЙ ОПИС АВТЕНТИФІКАЦІЇ КОРИСТУВАЧА

Для парольної автентифікації буде пароль Passwords, результати автентифікації будуть такі ж – true або false, процес генерування пароля passwordGeneration та верифікація verifyPassword.

$$\text{PasswordAuthentication} = \{\text{Passwords}, \{\text{true}, \text{false}\}, \\ \text{passwordGeneration}(\cdot), \text{verifyPassword}(\cdot)\}$$

Passwords – теоретично нескінченна множина, яка складається з комбінацій символів з алфавіту A, де A – множина символів, які можна ввести з клавіатури визначеного стандарту.

Властивості пароля з точки зору автентифікації:

- складність для запам’ятовування rememberComplexity;
- складність для зламу breakingComplexity.

$$\text{rememberComplexity} \to \min; \\ \text{breakingComplexity} \to \max.$$

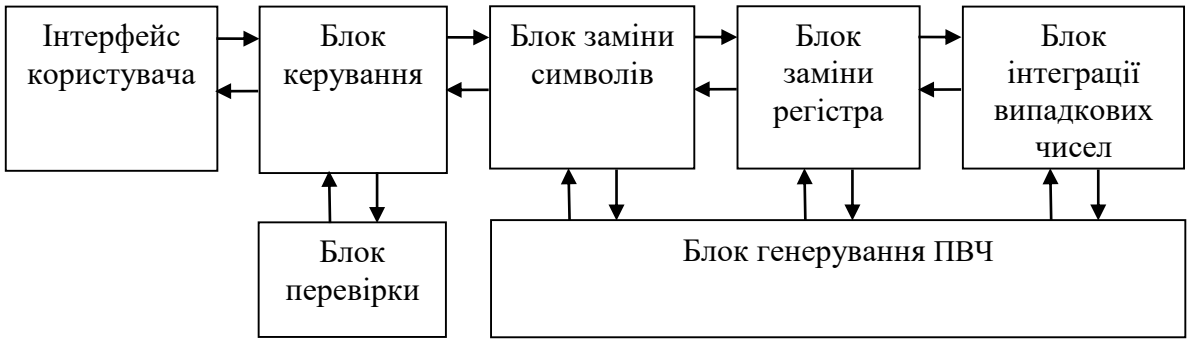
При збільшені довжини пароля збільшується складність його запам’ятовування і навпаки – закоротка ключова фраза не зможе надати необхідної стійкості.

Виникає постановка задачі магістерської кваліфікаційної роботи з точки зору математичного опису:

$$\text{rememberComplexity} \to \min; \\ \text{breakingComplexity} \geq \text{const}.$$

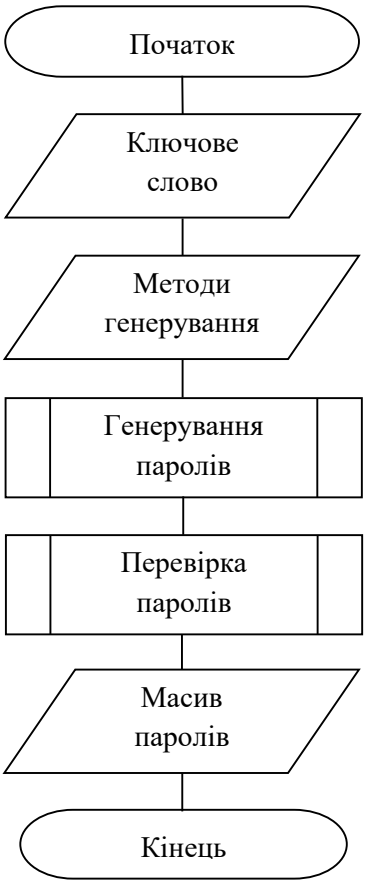
					08–20.МКР.005.00.000 ІЧ2		
Змн.	Арк.	№ докум.	Підпис	Дата			
Розроб..		Кохан О.В.			Метод та засіб підвищення стійкості паролів. Математичний опис автентифікації користувача	Літ.	Арк. 1 / Аркушів 1
Перевір.		Баришев Ю.В.					
Реценз.		Савицька Л.А.				ВНТУ, 1БС-20м	
Н. Контр.		Баришев Ю.В.					
Затверд.		Лужецький В.А.					

СТРУКТУРА ЗАСОБУ ПІДВИЩЕННЯ СТІЙКОСТІ ПАРОЛІВ



					08–20.МКР.005.00.000 ІЧЗ		
Змн.	Арк.	№ докум.	Підпис	Дата			
Розроб..		Кохан О.В.			Метод та засіб підвищення стійкості паролів. Структура засобу підвищення стійкості паролів	Літ.	Арк. 1 / Аркушів 1
Перевір.		Баришев Ю.В.					
Реценз.		Савицька Л.А.				ВНТУ, 1БС-20м	
Н. Контр.		Баришев Ю.В.					
Затверд.		Лужецький В.А.					

АЛГОРИТМ РОБОТИ ЗАСОБУ ПІДВИЩЕННЯ СТІЙКОСТІ ПАРОЛІВ



					08–20.МКР.005.00.000 ІЧ4		
Змн.	Арк.	№ докум.	Підпис	Дата			
Розроб..		Кохан О.В.			Метод та засіб підвищення стійкості паролів. Алгоритм роботи засобу підвищення стійкості паролів	Літ.	Арк. 1 / Аркушів 1
Перевір.		Баришев Ю.В.					
Реценз.		Савицька Л.А.				ВНТУ, 1БС-20м	
Н. Контр.		Баришев Ю.В.					
Затверд.		Лужецький В.А.					

ТАБЛИЦЯ ЗАМІН СИМВОЛІВ

Буква	Символ для заміни	Ймовірність заміни, %	Символ для заміни	Ймовірність заміни, %
A	@	88	/\	12
B	8	63	I3	37
C	(	89	<	11
H	#	69	]-[	31
I	1	52	!	48
K	|<	66	1<	34
M	/\/\	57	/v\	43
O	0	77	()	23
Q	0_	72	()_	28
S	$	88	5	12
X	}{	62	><	38
W	vv	59	\/\/	41

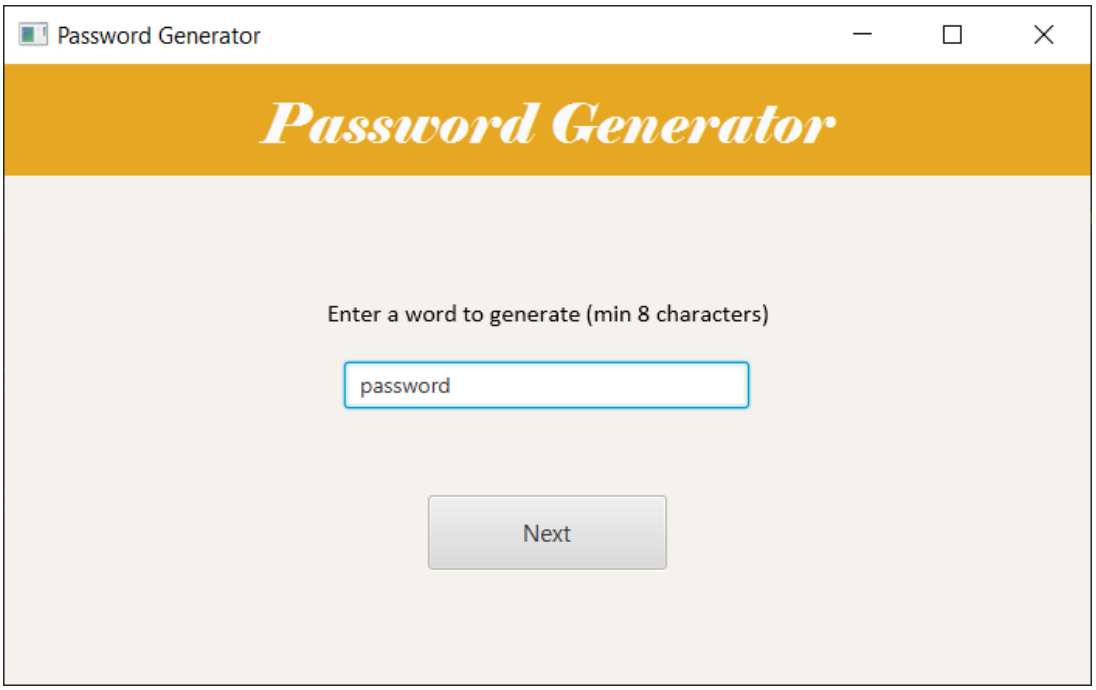
					08–20.МКР.005.00.000 ІЧ5		
Змн.	Арк.	№ докум.	Підпис	Дата			
Розроб..		Кохан О.В.			Метод та засіб підвищення стійкості паролів. Таблиця заміни символів	Літ.	Арк. 1 / Аркушів 1
Перевір.		Баришев Ю.В.					
Реценз.		Савицька Л.А.				ВНТУ, 1БС-20м	
Н. Контр.		Баришев Ю.В.					
Затверд.		Лужецький В.А.					

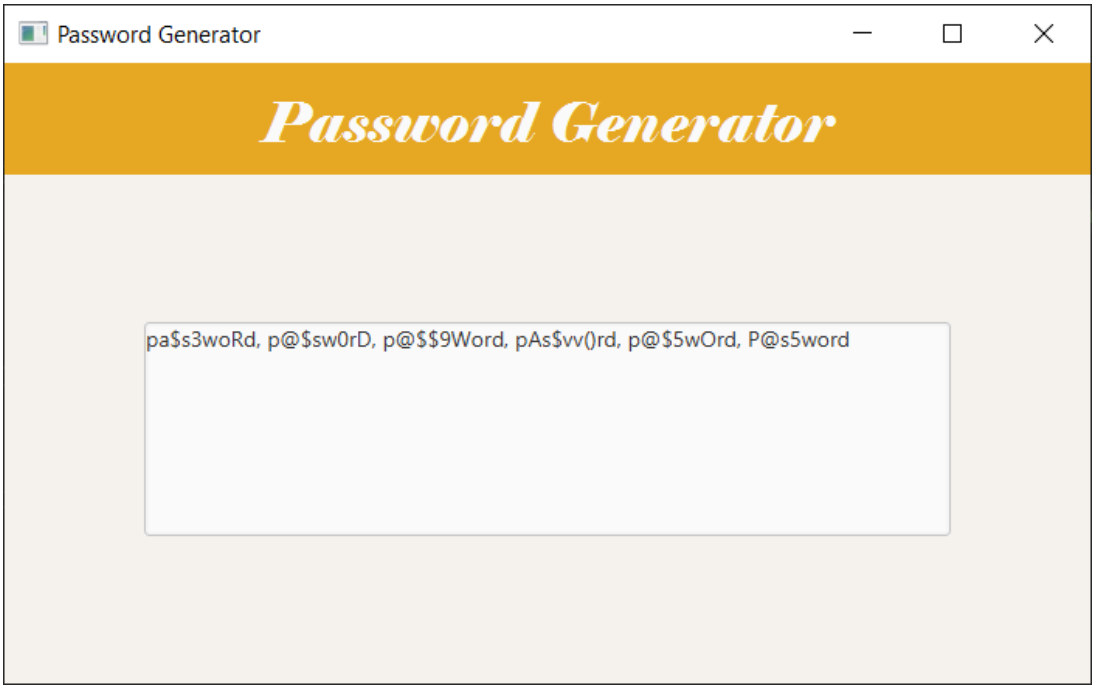
РЕЗУЛЬТАТИ ТЕСТУВАННЯ МЕТОДУ ЗАМІНИ

Буква	Символ для заміни	Ймовірність заміни, %	Резу-льтат тестів %	Символ для заміни	Ймовірність заміни, %	Резу-льтат тестів %
A	@	88	90	/\	12	10
B	8	63	64	I3	37	36
C	(	89	90	<	11	10
H	#	69	70	]-[	31	30
I	1	52	54	!	48	46
K	|<	66	68	1<	34	32
M	/\/\	57	58	/v\	43	42
O	0	77	78	()	23	22
Q	0_	72	72	()_	28	28
S	$	88	88	5	12	12
X	}{	62	64	><	38	36
W	vv	59	60	\/\/	41	40

					08–20.МКР.005.00.000 ІЧ6		
Змн.	Арк.	№ докум.	Підпис	Дата			
Розроб..		Кохан О.В.			Метод та засіб підвищення стійкості паролів. Результати тестування методу заміни	Літ.	Арк. 1 / Аркушів 1
Перевір.		Баришев Ю.В.					
Реценз.		Савицька Л.А.				ВНТУ, 1БС-20м	
Н. Контр.		Баришев Ю.В.					
Затверд.		Лужецький В.А.					

ТЕСТУВАННЯ ЗАСОБУ





					08–20.МКР.005.00.000 ІЧ7		
Змн.	Арк.	№ докум.	Підпис	Дата			
Розроб..		Кохан О.В.			Метод та засіб підвищення стійкості паролів. Тестування засобу	Літ.	Арк. 1 / Аркушів 1
Перевір.		Баришев Ю.В.					
Реценз.		Савицька Л.А.				ВНТУ, 1БС-20м	
Н. Контр.		Баришев Ю.В.					
Затверд.		Лужецький В.А.					