Хмельницький національний університет
Факультет програмування
та комп'ютерних і телекомунікаційних систем
Кафедра кібербезпеки та комп'ютерних систем і мереж

КВАЛІФІКАЦІЙНА РОБОТА

бакалавр
Освітній рівень

Система захисту інформаційно-комунікаційної мережі ДП "Новатор" Державного концерну "Укроборонпром", м.Хмельницький
Назва теми

КвРКБ.170141.17.02.15 ПЗ
Шифр

Галузь знань *12 «Інформаційні технології»*
Шифр, назва

Спеціальність *125 «Кібербезпека»*
Шифр, назва

Освітня програма *«Кібербезпека»*
Назва

Виконав: студент IV курсу, група КБ–17–1 _____ Беркута Я.О.
Підпис Ініціали, прізвище

Керівник _____ В.М. Чешун
Підпис, дата Ініціали, прізвище

Нормоконтролер _____ І.В. Муляр
Підпис, дата Ініціали, прізвище

До захисту допускаю:
Зав. кафедри кібербезпеки та
комп'ютерних систем і мереж _____ Ю.П. Кльоц
Підпис Ініціали, прізвище

«____» червня 2021 р.

Хмельницький 2021

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет ПРОГРАМУВАННЯ ТА КОМП'ЮТЕРНИХ І ТЕЛЕКОМУНІКАЦІЙНИХ СИСТЕМ

Кафедра КІБЕРБЕЗПЕКИ ТА КОМП'ЮТЕРНИХ СИСТЕМ І МЕРЕЖ

Освітній рівень БАКАЛАВР

Галузь знань 12 ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

Спеціальність 125 КІБЕРБЕЗПЕКА

Освітня програма ОСВІТНЬО-ПРОФЕСІЙНА ПРОГРАМА ПІДГОТОВКИ БАКАЛАВРІВ

ЗАТВЕРДЖУЮ

Завідувач кафедри Кльоц Ю.П

05. • 02 2021 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Беркута Ярослав Олександрович

Прізвище, ім'я, по батькові студента

1 Тема роботи Система захисту інформаційно-комунікаційної мережі ДП" Новатор" Державного концерну "Укроборомпром" , м.Хмельницький

Керівник роботи Чешун Віктор Миколайович

Прізвище, ім'я, по батькові, науковий ступінь, вчене звання

Затверджено наказом ректора університету від 05 02 2021р. № 11 додаток 9

2 Строк подання студентом роботи на кафедру: _____

3 Вихідні дані до роботи системи захисту інформаційно-комунікаційної мережі ДП "Новатор" Державного концерну "Укроборомпром", м.Хмельницький

4 Зміст пояснювальної записки (перелік питань, які потрібно розробити) Аналіз об'єкта захисту, обґрунтування вибору засобів для побудови системи безпеки, проектування системи безпеки, реалізація роботи

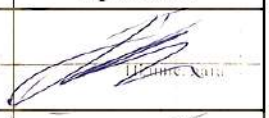
5 Перелік графічного матеріалу(із зазначенням обов'язкових креслень)

Впровадження розподіленого режиму

Інформаційна модель підприємства

Схема роботи програми, що взаємодіє з користувачем

6 Консультанти розділів курсового проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Муляр І.В., доцент кафедри КбКСМ		
Антиплагіат	Муляр І.В., доцент кафедри КбКСМ		

7 Дата видачі завдання 5 02 2021р.

КАЛЕНДАРНИЙ ПЛАН

№з/п	Назва етапів (розділів) дипломного проекту (роботи)	Термін виконання етапів проекту (роботи)	Примітка
1	Вибір та затвердження теми кваліфікаційної роботи.	Січень	–
2	Аналіз об'єкта захисту.	Січень–лютий	–
3	Проектування та розробка загальної архітектури і структури системи захисту.	Лютий–березень	–
4	Програмна реалізація запропонованого рішення та тестування системи; аналіз результатів і оцінювання прийнятих рішень.	Квітень	–
5	Написання тексту пояснювальної записки та розробка графічних матеріалів.	Травень	–
6	Остаточне коригування кваліфікаційної роботи з урахуванням зауважень керівника.		–
7	Оформлення кваліфікаційної роботи як документа відповідно до вимог.		–
8	Отримання супровідних документів. Нормоконтроль.	Червень	–
9	Підготовка до захисту та захист кваліфікаційної роботи.		–

Студент  Беркута Я.О.
Підпис Ініціали, прізвище

Керівник проекту (роботи)  В.М. Чешун
Підпис Ініціали, прізвище

АНОТАЦІЯ

Тема кваліфікаційної роботи: «Система захисту інформаційно-комунікаційної мережі ДП "Новатор" Державного концерну "Укроборомпром", м.Хмельницький».

Автор роботи: Беркута Ярослав Олександрович.

Керівник роботи: Чещун Віктор Миколайович.

Обсяг – 70 с., 9 рис., 2 додатка, 33 джерел.

Графічна частина: 9 презентаційних слайдів, 3 плакати.

Система захисту інформаційно-комунікаційної мережі ДП "Новатор"Державного концерну "Укроборомпром", м.Хмельницький.

Метою роботи є вивчення та аналіз типових проблем інформаційно-комунікаційної безпеки пов'язаних із витоком даних, побудова системи захисту від витоку даних.

У роботі було проаналізовано та досліджено проблеми інформаційно-комунікаційної безпеки, особливу увагу було зосереджено на проблемі витоків даних із конфіденційною інформацією причиною якого є внутрішній порушник інформаційної безпеки.

В рамках кваліфікаційної роботи була розроблена система захисту від витоків даних, спроектована із врахуванням побажань співробітників підприємства ДП "Новатор".

Підпис студента  Дата 5.06.21

Формат	Зона	Позиц	Позначення	Найменування	Кільк.	Прим.
A2		1	КвРКБ.170141.17.01.02 Е8	Впровадження розподіленого режиму	1	
A2		2	КвРКБ.170141.17.01.02 Е8	Інформаційна модель підприємства	1	
A2		3	КвРКБ.170141.17.01.02 Е8	Схема роботи програми, що взаємодіє з користувачем	1	

Зм.	Арк.	№ Докум.	Підп.	Дата	*КвРКБ.170141.17.01.02 ВП*		
Розробив		Беркута Я.О			Система захисту інформаційно-комунікаційної мережі ДП "Новатор" Державного концерну "Укроборонпром" Відомість проекту	Літера	Аркуш	Аркушів
Перев.		Чешун В.М.				н	1	2
Н. контр.		Муляр І.В.				ХНУ, КБ-17-1		
Затв.		Кльоц Ю.П.						

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

HDD	–	Hard Disk Drive
RBA	–	Remote Biometrical Assurance
SAML	–	Security Assertion Markup Language
SNMP	–	Simple Network Management Protocol
WAN	–	Wide Area Network
ЗІ	–	Захист інформації
ІКМ	–	Інформаційно-комунікаційні мережі
ІКС	–	Інформаційно-комунікаційні системи
ІКТ	–	Інформаційно-комунікаційні технології
ІТ	–	Інформаційні технології
ІТР	–	Інженерно-технічні робітники
ОС	–	Операційна система
ПЗ	–	Програмне забезпечення
ПК	–	Персональний комп’ютер
СЗІ	–	Система захисту інформації
СКД	–	Система контролю доступу
ТЗІ		Технічний захист інформації

ВСТУП

Актуальність. Напередодні 2020-х років уся планета виявилася обплетеною Всесвітнім павутинням Internet, інфраструктура якого утворюється шляхом об’єднання величезних регіональних мереж трансконтинентальними кабелями, а також численними каналами меншої перепускної здатності. Будь-які сучасні інформаційно-комунікаційні мережі (далі – ІКМ) можуть надавати своїм абонентам різноманітні, як завгодно складні, сервіси, однак, зазвичай, усі вони передбачають взаємодію двох абонентів між собою. Навіть, якщо разом одночасно спілкуються більше, ніж два абоненти (наприклад, як у телеконференціях із багатьма учасниками), все одно насправді за цією взаємодію приховується велика кількість взаємодій усіх абонентів поодинці із одним центральним сервером (тобто і тут мова іде про сукупність бінарних взаємодій типу «абонент-абонент»).

Важливо, що у переважній більшості усіх випадків взаємодії через ІКМ один абонент має виділене становище, може надавати у користування іншим абонентам свої ресурси (інформаційні, дискові, обчислювальні, алгоритмічні, і т.п.) і називається сервером, а інший підключається до нього і називається клієнтом. Під час підключення клієнта до сервера через ІКМ практично завжди виникає проблема санкціонованості цього процесу, адже абсолютно не будь-який клієнт може отримувати доступ до кожного ресурсу, який присутній в Інтернет (чи в ІКМ менших масштабів, зокрема регіональних – Wide Area Network, WAN). Коротко цю проблему можна охарактеризувати, як необхідність зведення відповідної системи захисту інформації (далі – СЗІ) в ІКМ, заснованої на певних протоколах автентифікації, що, відповідно, є актуальною задачею сучасної галузі кібербезпеки.

Відомі підходи до вирішення поставленої задачі. Очевидно, описана проблема виникла майже одразу із появою технології «клієнт-сервер» у територіально розподіленому варіанті її застосування (тобто фактично – з

появою перших ІКМ). За цей час розроблено чимало способів її вирішення, в результаті чого виникли як відомі надійні протоколи автентифікації типу Kerberos та OpenID Connect, так і поширені ненадійні механізми Cookie сесій, а також менш поширені у практичному використанні сертифікати X.509, можливості спеціальної мови SAML (Security Assertion Markup Language), а також механізм Secure SNMP (Simple Network Management Protocol) з використанням цифрового підпису.

Визначальною рисою усіх згаданих підходів є їх заснованість на знанні (інформації парольного характеру), або на атрибутах (деякі спеціальні файли складно, або неможливо сформувати вручну; вони повинні бути в наявності у законного користувача, тому можуть бути віднесені до атрибутів, хоча і нематеріального характеру). В той же час, у цих протоколах автентифікації відсутня прив'язка до біометричної інформації авторизованого користувача, що у деяких випадках може бути цілком неприйнятним через особливості системи захисту. Таким чином, актуальною є конкретна задача створення протоколу автентифікації, який включає біометричну інформацію клієнта, а отже дозволяє серверу більш надійним чином здійснювати його автентифікацію.

Таким чином, метою роботи є підвищення ступеня захисту ресурсів інформаційно-комунікаційних мереж за рахунок розробки та впровадження удосконаленого протоколу автентифікації.

Для досягнення поставленої мети слід вирішити наступні задачі:

- проаналізувати особливості існуючих протоколів автентифікації (як мережних, так і локальних);

- обрати один із протоколів, що може бути удосконалений, зокрема в частині використання біометричної інформації клієнта для його автентифікації;

- розробити удосконалений протокол автентифікації, який можна застосовувати в ІКМ;

- впровадити даний новий протокол у робочому програмному продукті та провести його тестування;

- проаналізувати результати, зробити висновки по роботі та оцінити перспективи її розвитку.

Об’єктом є процес автентифікації клієнта в інформаційно-комунікаційних мережах.

Предметом є алгоритми протоколів автентифікації клієнта, що можуть застосовуватися в ІКМ.

В роботі застосовуються методи криптографічних перетворень та біометричних досліджень (математичної статистики), а також елементи кластерного аналізу. Використано об’єктно-орієнтовану технологію програмування.

Практичне значення роботи полягає у тому, що на базі розробленого протоколу створено працюючий програмний продукт, за допомогою якого можна виконувати віддалену автентифікацію клієнта за його біометричними показниками.

В перспективі до протоколу можна додати можливості по автентифікації сервера, які відсутні у версії протоколу RBA 1.0, щоби надати гарантій безпечного з’єднання і клієнтові також.

1 АНАЛІЗ СУЧАСНОГО СТАНУ ПРОБЛЕМИ ЗАХИСТУ ІНФОРМАЦІЇ В ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНІЙ МЕРЕЖІ НА БАЗІ ПРОТОКОЛІВ АВТЕНТИФІКАЦІЇ

1.1 Загальний опис проблеми ЗІ в ІКМ

Інформаційно-комунікаційні мережі в першу чергу призначені для обміну та зберігання інформації. Як відомо, інформація, що у тій, чи іншій мірі підлягає захисту, може бути класифікована за ступенем секретності наступним чином:

- цілком таємна;
- таємна;
- для службового користування;
- відкрита.

Однак, для цілей даної роботи більший інтерес представляє характеристика власника інформації, що захищається. Тут можна виділити наступні класи поділу:

- державна таємниця (що циркулює у державних установах, організаціях, підприємствах і безпосередньо відноситься або є спорідненою до сфери їх діяльності);
- службового характеру, причому така, що використовується на приватних підприємствах;
- особистого характеру, тобто секретна інформація, яка породжена або належить приватним особам.

Питання захисту державної таємниці в цілому відносяться до компетенції спеціальних структур (наприклад, Служби Безпеки України) і мають вирішуватися професіоналами найвищого рівня, причому з можливістю залучення необхідних, досить потужних матеріальних ресурсів. Взагалі кажучи, ресурси, що можуть бути відведені для забезпечення

захисту державної таємниці, теоретично можуть бути як завгодно великими. В принципі тут може навіть порушуватися відомий принцип галузі захисту інформації, за яким витрати на виконання захисту інформації не повинні перевищувати вартості самої інформації. Така ситуація може бути обумовлена політичними рішеннями, небажанням окремих високих посадовців нести навіть і незначні іміджеві втрати, іншими особистими причинами людей, в руках яких сконцентрована значна влада. Беручи до уваги ці обставини, розглядати у даній роботі захист державної таємниці не будемо.

З іншого боку, особиста інформація, яка є важливою для окремих приватних осіб, зазвичай не є досить цінною для широкої спільноти зловмисників (хакерів), що прагнули б нею заволодіти. Відповідно, зводити спеціальні СЗІ у цьому випадку м'яко кажучи не доцільно, тому у даній роботі захист персональних даних також не розглядається.

Таким чином, предметом захисту у даному дослідженні цілком обґрунтовано виступає комерційна інформація, яка є продуктом життєдіяльності приватних підприємств (широкого профіля).

Методологічно одним із перших кроків при вирішенні будь-якої задачі захисту інформації є аналіз заданої системи на предмет наявності об'єктів захисту, зокрема, для даної роботи, пошук слід проводити у складі інформаційно-комунікаційних мереж. Відповідно, навіть, до самої назви систем – інформаційно-комунікаційні – очевидно, усі їх об'єкти захисту можна розбити на два великих класи ([1]) – рис. 1.1:

- сховища даних, або засоби збереження інформації (цим терміном назвемо усі пасивні сутності, місця, де можуть зберігатися дані, що підлягають захисту; це, наприклад, файли, робочі місця, письмові столи, і т.п., залежно від рівня абстракції, що прийнятий на даному етапі аналізу);

- канали зв'язку, або засоби комунікації (усі зв'язки між суб'єктами та об'єктами системи у різних комбінаціях; це, наприклад, провідний телефон

в кабінеті директора підприємства, провідний телефон секретаря, ADSL-з'єднання у відділі технічних працівників, і т.п.).

Рисунок 1.1 - Найбільш укрупнена класифікація об'єктів захисту інформації

Обидва види об'єктів захисту з рис. 1.1 зручно унаочнюються на схемі інформаційних потоків системи: адже вхід та вихід кожного потоку може бути сховищем даних, а сам потік безпосередньо має реалізуватися якимсь каналом зв'язку.

Таким чином, для можливості виконання пошуку загроз (що є наступним кроком при зведенні СЗІ) для об'єктів захисту інформації обраної системи, необхідно провести аналіз існуючих (та, можливо, нових, які необхідно утворити шляхом впровадження системи захисту інформації) інформаційних потоків, тобто створити інформаційну модель підприємства [2]. Для цього необхідно виконати аналіз наявних бізнес-процесів підприємства, існуючих виробничих відносин на ньому та загальної методики його роботи. Оскільки усі підприємства мають власні особливості, а саме, профіль діяльності, кількість та якість працівників, наявні матеріальні активи, і т.д., і т.п., то зробити точну загальну модель, звичайно, неможливо. Однак, можна розглянути орієнтовну узагальнену структуру, що включає в себе різноманітні основні елементи діяльності та складу приватних підприємств: купівлю/продаж, виробництво, сервісні функції,

функціонування відділу інформаційних технологій, функціонування відділу головного інженеру (питання електрики, будівельні питання, пожежна безпека, механічні операції, тощо), і т.д. Таку узагальнену модель приватного підприємства та взаємодії між його різними структурними

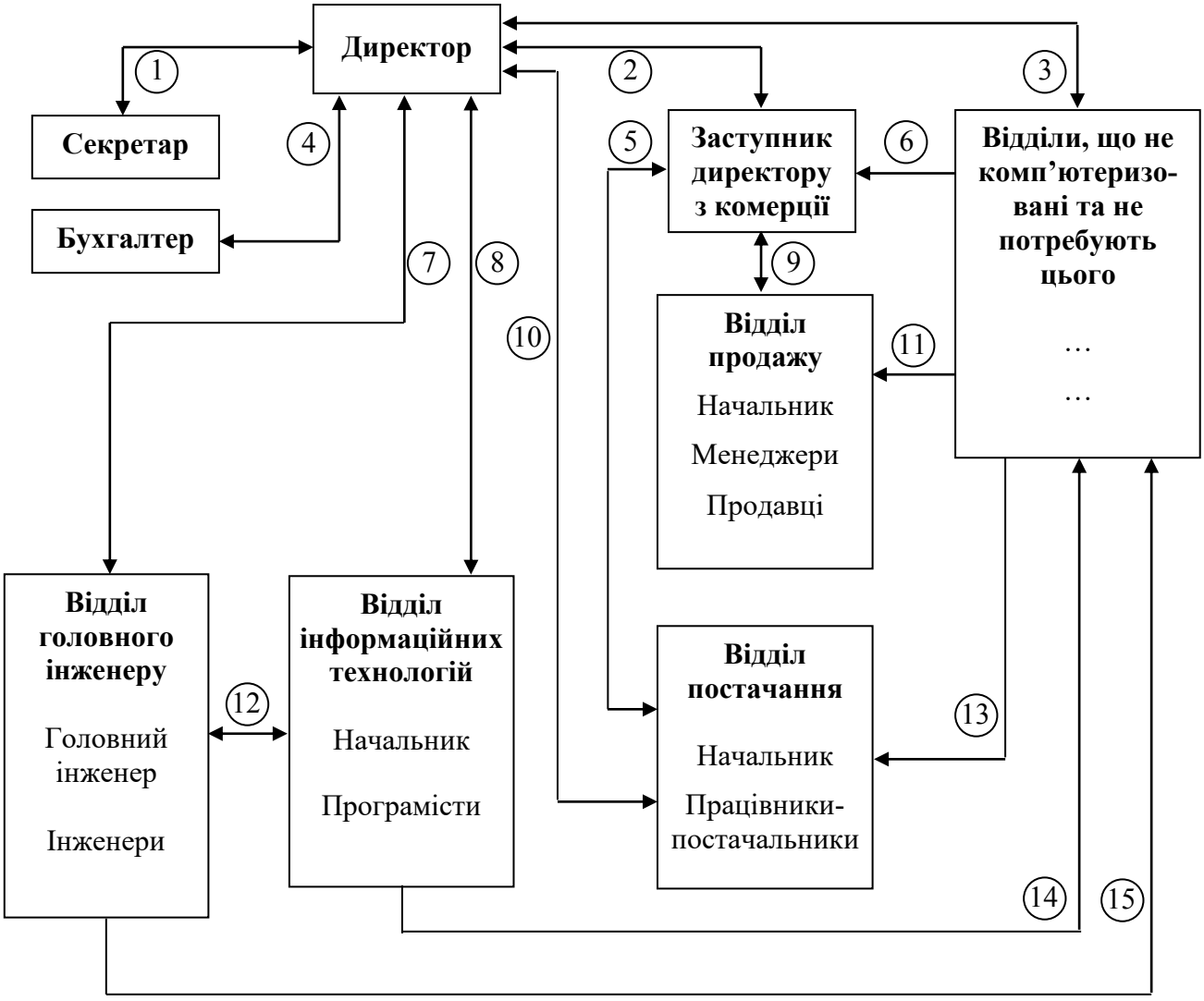


Рисунок 1.2 - Інформаційна модель підприємства у вигляді схеми інформаційних потоків.

ланками, зобразимо у вигляді схеми на рис. 1.2.

Для повного і правильного розуміння структури і значення кожного інформаційного потоку необхідно розшифрувати його семантичний зміст, що виконано в наступному тексті.

Дамо розшифровку наведених на рис. 1.2 позначень інформаційних потоків:

1 – інформація від секретаря про запити зовнішніх суб'єктів до директора; зворотний потік вказівок про поточні завдання; фізична суть інформації: мовна голосова;

2 – інформація про важливі замовлення, проблемні ситуації, звітування про хід комерційної діяльності виробництва; зворотний потік наказів з підвищення рівня комерційної діяльності підприємства;

3 – будь-яка безпосередня інформація від працівників до директора; зворотній потік заохочень та стягнень, відповідь на запити робітників;

4 – інформація про фінансовий стан підприємства; зворотний потік наказів про проплати, розподіл та управління коштами;

5 – інформація до заступника директору про необхідні закупки товарів та комплектуючих, запит фінансів; зворотний потік рекомендацій про оптимальний розподіл коштів на закупки, управління закупками;

6 – інформація від працівників про необхідність великих придбань, ремонту обладнання, вирішення довільних виробничих питань, суперечок;

7 – інформація від інженерно-технічних робітників (далі – ІТР) про нові перспективні розробки, удосконалення наявної продукції, вирішення принципових питань з проектування виробничого процесу (сюди ж включаються питання про апаратні засоби захисту критичної інформації приватного підприємства); зворотній потік даних про необхідність нових розробок та удосконалення старих, спрямовування у потрібному напрямі розробок продукції;

8 – інформація про нові інформаційні технології, які знаходяться на різних стадіях впровадження на підприємстві, в т.ч. потребують впровадження (сюди ж включаються програмні засоби захисту критичної інформації приватного підприємства); зворотний потік інформації рекомендаційного характеру з питань ІТ;

9 – інформація відділу продаж про хід продажу, запит готової продукції, аналітична інформація різного характеру; зворотний потік інформації про ціноутворення, планування рекламних подій, підвищення обізнаності потенційних покупців про продукцію підприємства;

10 – зведена інформація відділу постачання про проведені та плановані закупки, про необхідні великі закупівлі, відомості про контрагентів-постачальників; зворотний потік відомостей про пріоритетні місця закупок, управління процесом купівлі;

11 – інформація про хід виробництва готової продукції;

12 – обмін технічними відомостями про інформаційно-комунікаційні технології (ІКТ) та споріднені процеси;

13 – запит потрібних для нормального протікання виробничого процесу товарів працівниками;

14 – надання робочої документації по створенню продукції, яка містить елементи ІКТ;

15 – надання усієї робочої документації, консультації робітників, проведення навчання та роз'яснень.

Переважна більшість інформаційних потоків мають бути захищеними у тій, чи іншій мірі, причому засоби захисту потоків приватного підприємства, що циркулюють в інформаційно-комунікаційних мережах, будуть розглянуті нижче, тому тут розглянемо лише потоки, що є некомп'ютеризованими, аби до них більше не повертатися - табл. 1.1.

Таблиця 1.1 – Матриця інформаційних потоків, що відображує їх можливу фізичну природу.

№ потоку	Мовна голосова	Мовна текстова	Графічна	Таблична	Відео
1у	розмова телефонна, усна	SMS, записки	факс		
2	розмова телефонна, усна	Записки	факс	звіти	
3	розмова усна	Заяви		табель	
4	розмова телефонна, усна	Рахунки	факс	звіти	
5	розмова телефонна, усна	службові записки		звіти відомість	
6	розмова усна				
7	розмова усна	службові записки, заяви	креслення	специфікації, відомості	моделювання, записи випробувань
8	розмова усна	службові записки, заяви, тексти програм	креслення, схеми алгоритмів	специфікації програм	моделювання, записи випробувань
9	розмова усна, телефонна	SMS, розпоря-дження	рекламні зображення	прайс-листи, звіти	рекламні ролики
10	розмова усна, телефонна	накази, службові записки, заяви	фотографії	відомості товарів	
11	розмова телефонна, усна	описи особливос-тей продукції	пояснюючі фотографії		відеозаписи роботи продукції
12	розмова усна	текстові документи	креслення, схеми	таблиці команд і станів	
13	розмова усна	службові записки, замовлення		перелік закупних товарів	
14	розмова усна	специфікації електроніки	схеми підклю-чення	таблиці електричних контактів	
15	розмова усна	текстові описи, технологія	креслення, схеми розміщення	специфікації	навчальні матеріали

При обранні вказаних восьми потоків були прийняті наступні допущення:

- у всіх можливих випадках усну розмову можна замінити (повністю чи частково) на електронні комп'ютерні засоби зв'язку;

- усі потоки, що пов'язані із відділами, які не підлягатимуть комп'ютеризації, не слід розглядати в контексті проблеми захисту інформації приватного підприємства в ІКМ, адже у цих відділах власне відсутні засоби інформаційно-комунікаційного характеру;

- деякі інформаційні потоки не можна комп'ютеризувати з нетехнічних причин (наприклад, усну або телефонну розмову секретаря і директора не можна комп'ютеризувати через необхідність збереження певної субординації у їхніх відносинах).

В цілому, оскільки сама комп'ютеризація інформаційних потоків не є основною темою даної роботи, то після обрання інформаційних потоків, які підлягають комп'ютеризації, вважатимемо, що вона вже зроблена і будемо у наступному розглядати для цих потоків ті ж засоби захисту, що й для інших інформаційних потоків, що з самого початку забезпечувалися засобами ІКТ.

Наступним кроком при зведенні СЗІ є виконання аналізу загроз, які існують для даної системи [3]. Як відомо, усі загрози поділяються на три класи, в залежності від тієї якості інформації, на порушення якої вони напрямлені. Так, розрізняють три основні якості інформації, що є важливими для галузі кібербезпеки:

- цілісність;

- конфіденційність;

- доступність.

Розглянемо докладніше ці три варіанти порушення критичних властивостей інформації приватного підприємства, що циркулює в ІКМ.

Загрози цілісності [4] звичайно виникають у наступних випадках:

- відмова певного обладнання, що пов’язане із обробкою чи передачею даних;

- виконання людиною-оператором помилок («людський фактор»);

- деструктивна дія програм-вірусів;

- злочинне пошкодження інформації людиною-зловмисником.

Частою і дуже неприємною проблемою як персональних настільних комп’ютерів, так і ноутбуків, а іноді навіть і серверів, є відмова жорстких дисків, виготовлених по типу HDD (Hard Disk Drive), у яких над тонкими магнітними пластинами, виконаними з феромагнетику літає (в прямому сенсі, дуже швидко переміщуючись поруч із пластинами) голівка, що читає та пише інформацію. Фактично, HDD є єдиною критичною частиною комп’ютера, у якій здійснюється механічний рух (вентилятори систем охолодження можна не рахувати через їх дуже низьку вартість та легкість заміни), а як відомо, електронні компоненти на порядки надійніше, ніж механічні. В цілому, відновити дані зі зламаного жорсткого диску можна тільки, якщо причиною є вихід з ладу електронного контролеру HDD, а якщо причина – у механічному пошкодженні пластин, то відновити інформацію взагалі буде неможливим (принаймні традиційним обладнанням сервісних центрів, без залучення «шпіонських» технологій). Отже, загроза втрати даних є цілком реальною і для її уникнення обов’язковим має бути резервування інформації, причому особливо критичної – навіть більш, ніж двократне (зазвичай виконується за розкладом, спеціальними утилітами).

Помилки людини є ще однією важливою загрозою цілісності, що може мати суттєві наслідки, причому мова йде саме про ненавмисні дії (злочинні наміри розглянемо пізніше). Імовірність реалізації цієї загрози напряму залежить від кваліфікації користувачів, що здійснюють обробку інформації: чим вона вища, тем менше можливість несвідомої модифікації інформації. Очевидним шляхом нейтралізації цієї загрози є наскрізний процес

підвищення кваліфікації персоналу шляхом проведення тренінгів, курсів, окремих занять.

Ще однією проблемою цілісності даних є робота вірусів типу вандалів, які повністю знищують, або вносять невиправні зміни в інформації з метою простого завдання шкоди усім користувачам, компаніям та комп'ютерам підряд. Очевидним шляхом подолання такої загрози є використання перевірених (тобто таких, що використовуються відповідно до їх ліцензії) програм-антивірусів.

Також можливе пошкодження інформації людиною-зловмисником, який може бути як внутрішнім (незадоволеним або підкупленим, підмовленим) працівником компанії, так і зовнішнім (хакери, працівники фірм-конкурентів, і т.п.). Для уникнення несанкціонованого доступу (НСД) типу «запис/видалення інформації» слід проводити заходи з захисту інформації загального характеру, впроваджувати спеціальні апаратні та програмні засоби захисного характеру.

Для уникнення описаних нештатних ситуацій необхідно розробити і проводити комплекс заходів, направлених на збереження цілісності даних.

Наступним класом загроз є загрози доступності [4], які існують, коли:

- в мережі існує значне перевантаження каналів (наприклад, як наслідок DoS/DDoS-атаки, або у зв'язку із відмовою певної частини перепускних магістралей);

- тимчасово відмовив носій, на якому зберігаються дані, як частий випадок – недоступний (не вмикається) комп'ютер із необхідними даними;

- важливі дані стерто завдяки халатності, злого замислу, вірусом, тощо (загроза аналогічна по своїй суті модифікації).

Традиційно проблеми з доступністю інформації розглядаються в контексті атак на відмову у обслуговуванні, що є актуальним для приватного підприємства за умови, що у нього є сайт, на якому зберігається критична інформація. В цьому випадку, DoS-атака, а точніше розподілена

DDoS, становить реальну загрозу, зважаючи на те, що сайти приватних підприємств зазвичай не розміщуються на високопродуктивних серверах і ризик виходу з ладу всього інформаційного ресурсу при такій атаці є надзвичайно високим.

Для зменшення імовірності успішної DDoS-атаки треба намагатися якомога сильніше оптимізувати серверне ПЗ, особливо, якщо сайт є активним і використовує певний серверний код (на зразок PHP). При цьому усі складові сайту (в першу чергу, скрипти) мають бути оптимізовані, оскільки навіть різниця в 1 мілісекунду на одному запиті, що здається мізерною, при сотні тисяч запитів виростає у додаткові хвилинні затримки. Також суворому налаштуванню підлягають параметри самого веб-сервера, особливо в частині відсікання підозрілих, нестандартно сформованих запитів.

Нарешті, загрози конфіденційності [4] з'являються, якщо суб'єкт має доступ до інформації, яка вимагає вищого рівня секретності, ніж той, який початково був наданий суб'єктові (тобто поточний рівень доступу вище за документально дозволений). Частинним випадком такої ситуації є проникнення в мережу зовнішнього зловмисника (початковий рівень доступу – нульовий, тому будь-який доступ є незаконним і виникає загроза конфіденційності інформації). Цей варіант загрози приймаємо за основний у подальшому матеріалі, і саме боротьба із ним буде основною метою подальшої роботи. Існуючі загрози зведемо в таблицю 1.2.

Таблиця 1.2 - Матриця загроз локальної мережі приватного підприємства.

Небезпечний об’єкт, процес	Цілісність	Конфіденційність	Доступність
Внутрішній зловмисник	1.1. Помилкова модифікація даних працівником фірми. 1.2. Навмисна модифікація даних працівником фірми.	2.1. Отримання доступу до інформації більш високого рівня доступу.	3.1. Помилкове видалення інформації. 3.2. Навмисне видалення інформації.
Зовнішній зловмисник	1.3. Зміна даних через відкриті для повного доступу ресурси.	2.2. Несанкціонований доступ в локальну мережу підприємства. 2.3. Прослуховування трафіку мережі.	3.3. Організація відмов у обслуговуванні (DoS-атак).
Вірус	1.4. Модифікація файлів.	2.3. Ведення скритного нагляду за користувачем. 2.4. Перехоплення введених паролів. 2.5. Відправлення в мережу файлів з інформацією.	3.4. Організація розподілених відмов у обслуговуванні (DDoS-атак). 3.5. Видалення даних. 3.6. Руйнація даних на жорсткому диску.
Відмова апаратної частини	1.5. Некоректний запис у файл з даними.		3.7. Відсутність доступу до інформації, розміщеної на носієві, який відмовив.

Таким чином, у підрозділі розглянуто типи інформації, що підлягає захисту в ІКМ, та за основний об’єкт, що захищається, обрано службову інформацію приватних підприємств. Далі проаналізовано інформаційні потоки приватного підприємства, що підлягають захисту в ІКМ засобами програмного та апаратного характеру. Також виконано аналіз загроз та за основну прийнято загрозу несанкціонованого доступу клієнтів до серверної інформації з більш високим рівнем доступу, ніж є у клієнта (за умовчанням мається на увазі доступ на читання). Далі розглянемо, якими існуючими засобами цей захист може забезпечуватися.

1.2 Аналітичний огляд науково-технічних джерел, присвячених проблемі ЗІ в ІКМ на базі протоколів автентифікації

Як зазначалося у вступі, на сьогоднішній день існує певна кількість протоколів автентифікації, що мають забезпечувати захист інформації при організації зв’язку типу «клієнт-сервер» в ІКМ; розглянемо їх докладніше.

Так, головним засобом вирішення окреслених проблем автентифікації у мережі є протокол Kerberos, що є конкретною реалізацією протоколу Нідхема-Шредера, і описаний у багатьох джерелах, наприклад [5-7]. Kerberos – це програмний продукт, розроблений в середині 1980-х років у Масачусетському технологічному інституті і з тих пір він піддався ряду принципових змін. Клієнтські компоненти Kerberos присутні в більшості сучасних операційних систем (ОС).

Цей протокол призначений для вирішення наступного завдання. В наявності відкрита (незахищена) мережа, у вузлах якої зосереджені суб'єкти – користувачі, а також клієнтські і серверні програмні компоненти. Кожен суб'єкт має секретний ключ. Щоб суб'єкт C міг довести свою справжність суб'єкту S (без цього S не стане обслуговувати C), він повинен не тільки назвати себе (ідентифікація), але і продемонструвати знання секретного ключа (автентифікація). C не може просто надіслати S свій секретний ключ, по-перше, тому, що мережа відкрита (доступна для пасивного і активного прослуховування), а, по-друге, тому, що S не знає (і не повинен знати) секретний ключ C. В системі потрібно використовувати менш прямолінійний спосіб демонстрації клієнтом знання свого секретного ключа для сервера.

Для цього система Kerberos включає довірену третю сторону (тобто сторону, якій довіряють усі), що володіє секретними ключами суб'єктів, які обслуговуються, і допомагає їм у попарній перевірці автентичності. Схема застосування всього протоколу наведена на рис. 1.3.

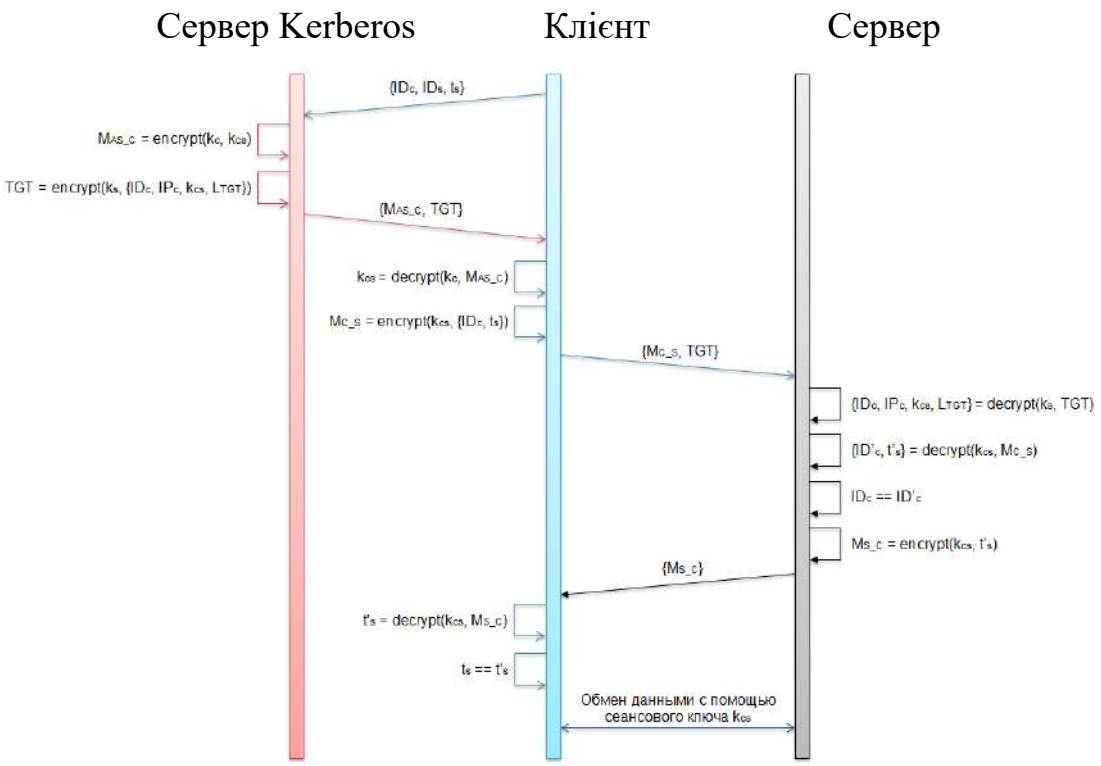


Рисунок 1.3 - Схема використання протоколу Kerberos для автентифікації у ІКМ.

Спрощено послідовність ідентифікації / аутентифікації за допомогою Kerberos версії 5.0 на основі специфікації RFC 1510 "Request for Comments: 1510. The Kerberos Network Authentication Service (V5)" виглядає наступним чином:

1) Клієнт посилає серверу Kerberos запит, який містить свій ідентифікатор IDc і ідентифікатор сервера даних (запитуваної послуги) IDs, а також мітку часу ts.

2) Сервер Kerberos виконує наступні дії:

- по мітці часу ts перевіряє синхронізацію свого годинника з годинником клієнта;

- формує повідомлення MAS_C - зашифровує сеансовий ключ kcs ключем клієнта kc;

- формує квиток TGT - зашифровує ідентифікатор клієнта IDc, його IP-адресу IPc, сеансовий ключ kcs і час життя квитка (сеансового ключа) LTGT ключем сервера ks;

- відсилає повідомлення MAS_C і квиток TGT клієнту.

3) Клієнт виконує наступні дії:

- розшифровує повідомлення MAS_C за допомогою свого ключа kc для отримання сеансового ключа kcs;

- формує повідомлення MC_S - зашифровує свій ідентифікатор IDc і мітку часу ts сеансовим ключем kcs;

- відсилає повідомлення MC_S і квиток TGT сервера.

4) Сервер виконує наступні дії:

- розшифровує квиток TGT за допомогою свого ключа ks для отримання ідентифікатора клієнта IDc, його IP-адреси IPc, сеансового ключа kcs і часу життя квитка LTGT;

- розшифровує повідомлення MC_S за допомогою сеансового ключа kcs для отримання ідентифікатора клієнта ID'c і мітки часу t's;

- для аутентифікації клієнта виконує порівняння ідентифікаторів IDc і ID'c, ойтриманих, відповідно, з квитка TGT і повідомлення MC_S;

- формує повідомлення MS_C - зашифровує мітку часу t's сеансовим ключем kcs;

- відсилає повідомлення MS_C клієнту.

5) Клієнт прикінцево:

- розшифровує повідомлення MS_C за допомогою сеансового ключа kcs для отримання мітки часу t's;

- для аутентифікації сервера виконує порівняння міток часу ts і t's.

6) Обмін даними між клієнтом і сервером виконується за допомогою сеансового ключа kcs протягом часу життя квитка (сеансового ключа) LTGT.

Як видно з даного протоколу, крім ідентифікації / аутентифікації, паралельно вирішується питання з обміном сеансовим ключем.

Ярким недоліком розглянутого підходу є відсутність аутентифікації самого користувача. Фактично аутентифікації підлягає набір даних (ключів), яким може заволодіти зловмисник, а потім, як результат, і отримати доступ до віддалених ресурсів серверу.

Набагато більш сучасна надбудова OpenID Connect над протоколом OAuth 2.0 має абсолютно той же самий недолік. OAuth – це відкритий протокол (схема) авторизації, що дозволяє надати третій стороні обмежений доступ до захищених ресурсів користувача без необхідності передавати їй (третій стороні) логін і пароль [8-9]. Робота над протоколом почалася в листопаді 2006 року, а остання версія OAuth 1.0 була затверджена 4 грудня 2007 року. Як подальший розвиток в 2010 році з'явився протокол OAuth 2.0, остання версія якого в якості в RFC 6749 опублікована в жовтні 2012 року [10].

OpenID Connect - відкритий стандарт децентралізованої системи автентифікації, що надає користувачеві можливість створити єдиний обліковий запис для автентифікації на безлічі не пов'язаних один з одним Інтернет-ресурсів, використовуючи послуги третіх осіб [11]. Базовою функцією OpenID є надання портативного, клієнт-орієнтованого, цифрового ідентифікатора для вільного і децентралізованого використання [12]. Таким чином, дійсно доступ забезпечується за допомогою цифрового атрибуту – ідентифікатора (спеціального файлу), яким може заволодіти зловмисник, і, отже, отримати доступ до віддалених ресурсів користувача.

Інші протоколи автентифікації в ІКМ (тобто такі, що використовуються у віддаленому режимі) мають аналогічний недолік: вони не беруть до уваги біометричні показники клієнта, хоча очевидно така можливість безперечно є наявною.

1.3. Постановка задачі

Беручи до уваги усі особливості інформації, наведеної у попередньому розділі, можна сформулювати наступну задачу: розробити протокол автентифікації в інформаційно-комунікаційних мережах та систему захисту інформації на основі його, що використовували би у якості об’єкта контролю біометричні показники особи, яка потребує надання доступу. При цьому мають бути вирішені наступні питання:

- який протокол автентифікації брати за основу;
- в якій саме частині вносити зміни в обраний протокол автентифікації;
- які саме біометричні показники повинні підлягати контролю;
- яким способом здійснювати порівняння поточного образу клієнта та еталону;
- як реалізувати конкретну СЗІ на базі розробленого протоколу;
- якими є результати роботи створеної СЗІ.

Результатом роботи має бути завершений програмний продукт у вигляді розподіленого програмного комплексу, який забезпечує виконання автентифікації за розробленим протоколом.

Відповіді на ці запитання у комплексі визначають особливості даної роботи і будуть надані та обґрунтовані в наступних розділах.

1.4 Висновок

В першому розділі роботи були досліджені основні проблеми ЗІ в ІКМ. Був проведений аналітичний огляд присвячених проблемі ЗІ в ІКМ на базі протоколів автентифікації. Також був запущений та затверджений процес постановки задачі для виконання технічної частини.

2 АНАЛІЗ ТА ОБГРУНТУВАННЯ ВИБОРУ ЗАСОБІВ ТА ТЕХНОЛОГІЙ, ЩО МОЖУТЬ БУТИ ЗАСТОСОВАНІ ДЛЯ ЗАХИСТУ ІНФОРМАЦІ В ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНІЙ МЕРЕЖІ З ВИКОРИСТАННЯМ ПРОТОКОЛІВ АВТЕНТИФІКАЦІЇ

2.1 Аналіз існуючих протоколів автентифікації

Недоліком існуючих протоколів автентифікації в ІКМ у попередньому розділі було названо відсутність урахування біометричних параметрів суб'єкта, що авторизується. Відповідно, для початку слід розглянути протоколи біометричної аутентифікації, що виконується локально (на тому ж комп'ютері, доступ до якого контролюється), наприклад, [13] або [14].

Найбільш укрупнено усі протоколи (способи) біометричної аутентифікації діляться на статичні (по фізіологічним характеристикам тіла людини) та динамічні (по оцінці поведінки людини у певних, стандартизованих ситуаціях). Розглянемо їх особливості докладніше.

Почнемо з автентифікації по фізіологічним характеристикам людини, які змінюються дуже повільно (істотні зміни видно на інтервалах часу, що значно перевищують час між послідовними автентифікації), або взагалі не підлягають змінам.

В першу чергу, під такий тип аутентифікації підпадає використання відбитків пальців людини, унікальність яких використовується вже більше ста років в криміналістиці (метод, названий «дактилоскопія»). Згідно з різними оцінками, ймовірність збігу відбитків пальців у двох різних людей становить від 10-10 (тобто у двох осіб з не менше ніж 10 млрд. людей) до 10-4 (у двох із 10 тис.). Навіть при розгляді самої песимістичної оцінки, даний рівень традиційно вважається достатнім для використання відбитків пальців у якості 100% -ного доказу в суді.

Для цілей даної роботи інтерес представляє можливість розширення протоколу доступу до ресурсів локального ПК на основі такої і всіх наступних

біометричних систем, до контролю доступу віддалено. Очевидним недоліком саме цього варіанту СКД є необхідність наявності спеціалізованого апаратного рішення - сканера відбитків пальців - рис. 2.1.

а)

в)

Рисунок 2.1 - Варіанти сканерів відбитків пальців: а - стаціонарний для обліку робочого часу співробітників, б - портативний для ПК, в - у мобільному телефоні, г - комбінований з атрибутною СКД (по електронній карті).

У деяких електронних цифрових пристроях виробники вводять сканер відбитків пальців як додаткову опцію, яка підвищує конкурентоспроможність продукту на ринку - рис. 2.1., в. Однак, якщо потрібно впровадити такий спосіб аутентифікації на основі вже існуючого апаратного забезпечення (а не такого,

що тільки буде придбане в майбутньому), то ймовірність відсутності таких датчиків в наявній (старій) техніці буде близька до 100%. Відповідно, необхідні витрати на їх придбання і впровадження, навчання персоналу, обслуговування, супровід, і т.п., тобто тут повною мірою виявляються недоліки статичних біометричних систем, а саме їх вартісні характеристики.

Ще сильніше ситуація з фінансами ускладнюється при бажанні використовувати сканери райдужної оболонки ока, які є значно більш дорогими пристроями, ніж сканери відбитків пальців. Цей спосіб аутентифікації заснований на унікальності картини райдужних оболонок ока у кожної людини. В цілому, багато «місць» на тілі людини мають унікальну картину тих, чи інших елементів, які можна використовувати для аутентифікації: це і особливості шкірних покривів, розподіл жирової тканини, параметри кісткового скелета - все, що повільно змінюється, відносно легко детектується, має задовільну стабільність на малих часових інтервалах для однієї конкретної людини, може бути використано в якості контрольованої характеристики в біометричних СКД. Звичайно, в залежності від складності процесу контролю таких різноманітних характеристик буде змінюватися і вартість необхідного апаратного забезпечення.

Так, досить дорогі сканери форми руки, голови, обличчя, тому що для адекватної роботи вони повинні містити кілька камер і складне математичне забезпечення для обробки (суміщення) просторово рознесених зображень.

Висока вартість (особливо при великій кількості абонентських пунктів СКД або, наприклад, необхідності забезпечення їх мобільності) не є єдиним недоліком статичних систем. У деяких випадках існують принципові перешкоди організаційного характеру для впровадження статичних біометричних систем, в той час, як динамічні системи можуть бути легко впроваджені. Така ситуація якраз реалізується, наприклад, при необхідності введення контролю доступу до ресурсів інформаційно-комунікаційної мережі, що має загальновідомий вихід у всесвітнє павутиння Інтернет.

У таких випадках традиційним підходом є використання парольного захисту, однак, при вході в систему, на додаток до парольного захисту або замість нього, легко можна ініціювати виконання користувачем деяких дій, і за особливостями самого процесу їх виконання людиною проводити процес аутентифікації (тобто оцінювати поведінку користувача, манеру його дій, також і числові характеристики, що характерно для поведінкових біометричних СКД). Таким чином, при наявності великої необхідності, біометричні системи можуть впроваджуватися без будь-якої попередньої підготовки, причому мова йде саме про поведінкові (динамічні) СКД. Ця ситуація є надзвичайно зручною для організації протоколу автентифікації в ІКМ, що і буде виконано у подальшому, (але спочатку треба буде докладно розглянути принципи дії поведінкових біометричних систем).

Отже, як висновок можна сказати, що фізіологічні біометричні СКД мають наступні недоліки:

- висока вартість початкових одноразових витрат на реалізацію системи;

- висока вартість супроводу системи;

- слабка гнучкість, неможливість віддаленої аутентифікації без попереднього створення апаратного абонентського пункту.

З огляду на ці недоліки, однозначно можна сказати, що в даному дослідженні використовувати статичні біометричні СКД недоцільно, тому розглянемо другий клас таких систем - побудовані на контролюванні поведінки людини.

Загальновідомо, що кожна людина в процесі життєдіяльності набуває все більшої кількості звичок. Фактично, кожен новий вид діяльності, призначений для виконання людиною, перший раз проводиться якимсь певним способом і, якщо індивід в цілому задоволений результатом, далі цей спосіб або манера виконання закріплюється і переходить в звичний стиль поведінки. У той же час, переважна більшість дій (крім, можливо, найпростіших) мають різні варіанти або способи виконання та характеризуються різними значеннями своїх

числових параметрів. Це стосується і складових комплексних процесів, зокрема, пов'язаних з вищої нервової діяльністю (як-то звичка мислити певним чином, наприклад, песимістично або оптимістично; намагатися знайти рішення проблеми самостійно або відразу звертатися за допомогою до кого-небудь, і т.д.), і навіть простих моторних дій, пов'язаних зі звичайною життєдіяльністю. Так, наприклад, ніхто не стане заперечувати, що кожна людина має свої особливості ходьби, які простими словами називають «хода», оригінальними особливостями написання тексту (свій «почерк»), і т.д.

В цілому, можна сказати, що будь-який вид діяльності людини (від досить простих до найскладніших), який раніше багато разів нею виконувався, має свої стійкі особливості, які можуть бути використані для контролю в біометричних СКД.

Слід зазначити деякі особливості самих поведінкових СКД, пов'язані з варіативністю виконання тієї чи іншої контрольованої дії при різних умовах, а саме, необхідно:

- проаналізувати особливості варіативності виконання потенційно контрольованої дії однією людиною в різні моменти часу (тобто у випадку, що найбільш часто зустрічається, для контролю якоїсь числової характеристики обраної дії, потрібно якісно або кількісно оцінити дисперсію, що отримується на основі обробки значень цієї характеристики, виміряних в різних експериментах з одним і тим же користувачем);

- розібрати особливості варіативності ознаки в межах групи людей (тобто проаналізувати варіативність виконання потенційно контрольованої дії різними людьми). Для цього можна оцінити внутрішньогрупову дисперсію, що отримується на основі обробки середніх значень цієї характеристики, що приписуються кожному користувачеві окремо, і отриманих для кожного користувача в результаті окремої серії експериментів.

Отже, в першу чергу, розглянемо варіативність для однієї людини, і ця індивідуальна варіативність, в свою чергу, також має кілька аспектів.

По-перше, деякі дії, які в рамках даної роботи доречно буде назвати надзвичайно простими, можуть у зв'язку з самою своєю природою мати малу варіативність (причому сюди ж слід віднести і групову варіативність). Наприклад, така дія як плескання в долоні, хоча і може виконуватися досить різноманітними способами з точки зору взаємного просторового положення долонь, сили удару, швидкості руху рук, і т.д., проте така його характеристика, як тривалість самого звуку (інтервал часу в мілісекундах від моменту реєстрації звукового сигналу мікрофоном і до його падіння нижче певного процентного значення, припустимо, 1%) не буде мати значної варіативності в зв'язку з самою природою процесу.

По-друге, крім самої природи процесу, варіативність також може бути обмежена просторовими, тимчасовими або іншими рамками. Наприклад, в деяких автоматизованих системах час введення інформації обмежений і по його закінченні нормальна робота користувача переривається (наприклад, задається питання, чи тут знаходиться людина). Така поведінка системи не дозволяє використовувати таку характеристику, як час введення даних в систему, для ефективної поведінкової ідентифікації окремих категорій громадян, таких як глибокі пенсіонери, яким часто потрібно для роботи в системі потрібний значний час, що суттєво перевищує середні значення. Ще одним прикладом обмежувальних рамок може служити використання незвичайних засобів введення-виведення для контролю, наприклад, почерку людини. Так, пропозиція написати текст спеціальною ручкою на скляному сенсорному екрані викличе у багатьох користувачів уповільнення всього процесу письма, і, як наслідок, більш старанне виведення окремих букв, що в цілому знижує варіативність (внутрішньогрупову - у більшій мірі, але також і індивідуальну).

По-третє, багато (якщо не всі) поведінкові характеристики можуть залежати від настрою (в меншій мірі), а також, що більш істотно - від втоми (найбільш важливий аспект), нездужання, хвороби, травм. Принципових перешкод до впровадження поведінкових систем в реальних СКД зазначені

зауваження не вносять, тому що втома може бути врахована в роботі системи якимось окремим показником (коефіцієнтом), який може змінюватися протягом робочого дня.

По-четверте, деякі процеси мають дуже погану повторюваність, наприклад, не можна контролювати силу удару м'яча по невеликій мішені, так як по ній ще потрібно потрапити, а при попаданні удар може бути не центральним (як необхідно для вимірювання сили удару), а дотичним (і тоді, при малій зміні траєкторії, зафіксована датчиком сила може бути дуже і дуже мала, що абсолютно не відповідатиме реальній силі, з якою був кинутий м'яч, тобто поведінці користувача). Однак, такі характеристики все одно можна успішно використовувати для контролю доступу, якщо тільки в якості контрольованої величини вибрати не миттєве (одноразово зняте) значення такої характеристики, а середнє значення на великому числі випробувань. Практично це буде виглядати наступним чином:

- людина кидає м'яч по мішені 10 разів і кожного разу датчиком фіксується сила удару (в деяких випадках навіть і дотичного, навіть і невдалого, тобто без попадання, коли сила в результаті дорівнює нулю);

- середнє значення сили, що спостерігається за цими 10 ударами записується в файл-еталон;

- потім, при необхідності авторизуватися в системі, людині знову пропонується N разів виконати вказану дію;

- розраховується середнє значення по числу всіх спроб;

- виконується безпосередньо порівняння двох середніх і робиться висновок про успішну авторизацію або відмову в доступі.

Порівнянню у даному випадку підлягають значення, усереднені по N спробам одного користувача.

Ще одне, надзвичайно важливе зауваження, що стосується поведінкових систем, полягає в тому, що, звичайно ж, кожна людина є живою системою, що пристосовується, тому, якщо змінюються зовнішні умови, то і поведінка також

підлягає зміні. Однак, ключовим моментом тут є те, що ці зміни все одно мають еволюційний, а не вибуховий характер.

Наприклад, якщо людині навіть наказати найсуворішим чином почати писати іншим почерком, повністю задовольнити таку вимогу миттєво вона не зможе. Можливо, відразу можна буде поміняти величину букв, або навіть їх нахил, однак більш глибинні параметри написання окремих символів (як-то різноманітні завитки, початки і закінчення при написанні літер) все одно будуть вказувати на авторство рукописного тексту. Також іноді обов'язково будуть проскакувати і старі варіанти написання тексту, що буде також свідчити про його первісне авторство. Таким чином, навіть при цілеспрямованій примусовій зміні, поведінка людини не може змінитися відразу, тим більше вона не може занадто змінитися у користувача, який навпаки бажає авторизуватися в системі і проходить процес аутентифікації.

Отже, єдиною важливою умовою успішної роботи СКД, заснованої на контролі поведінки, пов'язаною з мінливістю цієї самої поведінки, є необхідність малості проміжку часу між двома послідовними аутентифікації (не більше кількох місяців, а краще - кілька тижнів) і періодичне оновлення еталонних характеристик (наприклад, раз на місяць - при інтенсивній роботі в системі). В ідеалі необхідно, щоб таке оновлення відбувалося прозоро для користувача, тобто шляхом пасивного спостереження якоюсь автоматизованою частиною біометричної СКД, відповідальною за оновлення еталонної інформації, за його рутинними робочими процесами. Якщо таку прозору поведінку СКД реалізувати неможливо, то шляхом впровадження організаційних заходів щодо захисту інформації (тобто, наприклад, за розкладом, закріпленим спеціальною інструкцією по роботі в системі), слід періодично «знімати» з користувача системи поточні значення його контрольованих поведінкових характеристик.

Другий аспект роботи поведінкових систем заснований на потенційній варіативності обраної характеристики всередині групи, тобто для різних

особистостей. Як уже зазначалося вище, деякі характеристики не можуть сильно варіюватися в зв'язку з самою своєю природою, що стосується і індивідуальної, і внутрішньогрупової варіативності. Наприклад, час виконання стрибка на місці (від моменту відриву ніг до моменту торкання з поверхнею) не може збільшуватися для різних людей у великій мірі, тому що ніхто з людей не вміє літати. Це ще один приклад характеристики, яка має обмежену самою своєю суттю варіативність.

З огляду на вищесказане, можна стверджувати, що завдання фахівця із захисту інформації при зведенні протоколу роботи динамічної біометричної СКД полягає у відборі комплексу таких поведінкових характеристик і їх параметрів, які найкращим чином підходять для цілей захисту інформації, а точніше мають наступні властивості:

- володіють достатньою стабільністю своїх числових оцінок (оцінки) для даного індивідуума (іншими словами у таких характеристик повинна бути мала індивідуальна варіативність);

- велика групова варіативність (тобто іншими словами, значення даної характеристики для різних людей повинні бути такими, щоб мати можливість відрізнятися один від одного, причому чим у більш значній мірі, тим краще);

- не дуже швидка мінливість середнього значення характеристики в часі (інтервал часу, за який відбуваються значні зміни середнього значення, повинен бути хоча б в рази більше інтервалу між двома послідовними авторизаціями);

- принципово хороша повторюваність результатів вимірювань характеристики у одного індивідуума (скупченість результатів біля середнього значення, дисперсія мала), через що контролю підлягає саме виміряне значення;

- принципово погана повторюваність результатів вимірювань характеристики у одній групі (великий розкид значень навколо середнього, велика дисперсія).

Слід зазначити, що застосування аутентифікації за результатами вимірювань всього лише однієї характеристики не доцільно, тому що має малу

ефективність. Набагато кращі результати дає перевірка відразу декількох характеристик, і чим більше їх підлягає контролю, тим ефективніша така система. В цьому випадку нівелюється можливість простого збігу значень вимірювання якої-небудь однієї характеристики, тому що береться до уваги цілий їх ансамбль. У цьому випадку кожен користувач після зняття еталонної інформації буде характеризуватися для системи не одним числом, але n-мірним (по числу контрольованих величин) вектором. Наочно такий вектор може бути представлений стовпчатою діаграмою, дуже зручною для поверхневого візуального аналізу.

Розглянемо докладніше приклад, в якому для аналізу особистості користувача біометричної СКД вибрано чотири характеристики, що виражаються числовими величинами в межах (0; 200). З трьох різних людей були зняті еталонні характеристики, в результаті чого отримана табл. 2.1.

Таблиця 2.1 - Приклад еталонних даних для 3 людей по 4 характеристикам.

	Хар-ка 1	Хар-ка 2	Хар-ка 3	Хар-ка 4
Пользователь1	10	22	115	48
Пользователь2	28	23	56	17
Пользователь3	34	27	55	62

Дані із табл.2.1 можуть бути наочно представлені наборами стовпчикових діаграм - як на рис. 2.2.

Проаналізуємо рис. 2.2: на ньому крім еталонних значень всіх контрольованих характеристик (пронумеровані цифрами від 1 до 4) для трьох користувачів (Еталон1, Еталон2, Еталон3), також зображений «знімок» людини, яка бажає пройти авторизацію в даний момент. Елементарний візуальний аналіз дозволяє вирішити задачу ідентифікації, якби вона була актуальна (таке завдання могло би виникнути, якби точно було відомо, що авторизуватися в системі може тільки хтось один із тих трьох людей, що в ній зареєстровані,

тобто з яких були зняті еталонні характеристики). Дійсно очевидною є схожість людини, що авторизується, і Еталона1, інша справа, що дана схожість має отримати числове математичне підтвердження. Таким чином, задача аутентифікації не може бути вирішена приблизно, а вимагає суворого математичного розрахунку.

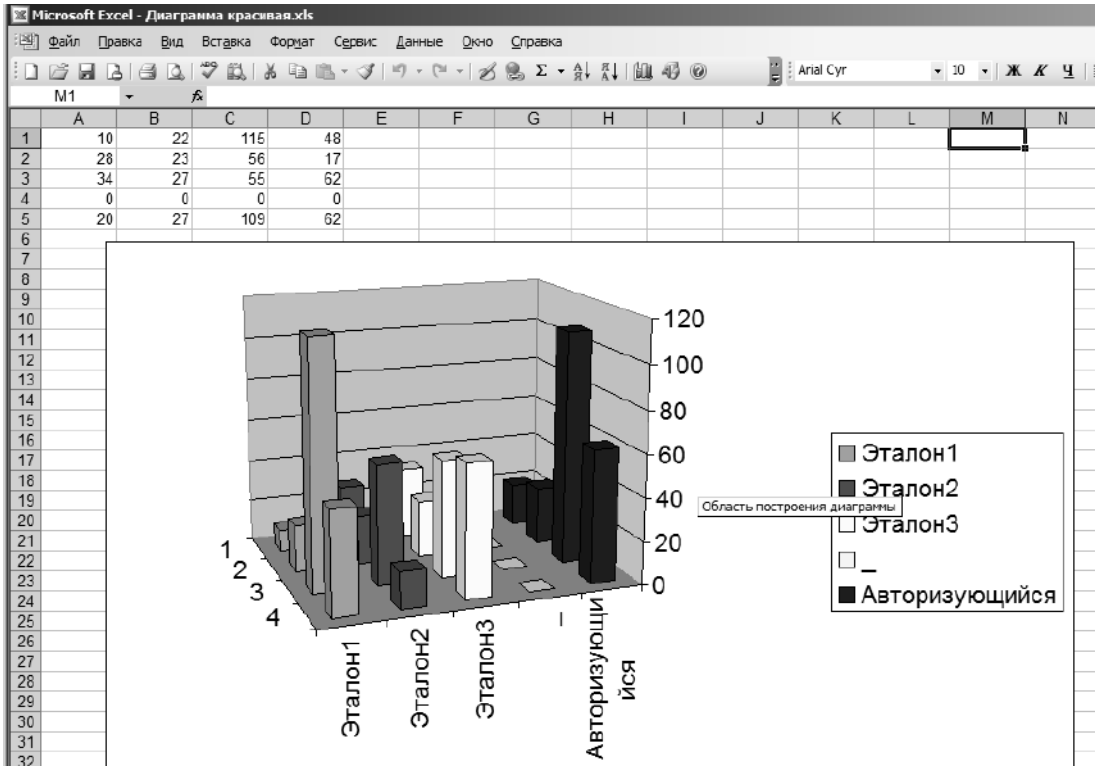


Рисунок 2.2 - Наочне представлення еталонної інформації, при якому кожна людина характеризується своїм набором чотирьох характеристик.

Беручи до уваги усю вищесказану інформацію, можна сформулювати наступне питання: які ж характеристики можуть задовольняти зазначеним теоретичним вимогам, і які реально застосовуються в біометричних СКД? В першу чергу це можуть бути характеристики роботи користувача з пристроями введення-виведення, а точніше - саме введення інформації в ПК. І найцікавішим (комплексним і складним) з таких пристроїв є клавіатура комп'ютера. Серед показників роботи з клавіатурою можна виділити наступні:

- середня швидкість набору тексту;
- відносна кількість помилок при наборі тексту;

- середній час утримання різних окремих кнопок при наборі тексту;

- середній час набору деяких поширених біграм (двосимвольних послідовностей).

Що стосується миші, як пристрою введення інформації, то тут контролю можуть підлягати:

- максимальна швидкість переміщення миші;

- середній час наведення курсору миші на невеликі об'єкти перед клацанням;

- середній інтервал часу між двома послідовними натисканнями лівої кнопки миші при подвійному натисканні;

- середній інтервал часу між викликом контекстного меню за допомогою правої кнопки миші і вибором будь-якого пункту меню;

- і т.д.

Слід сказати, що можна запропонувати і більшу кількість різних показників, однак деякі з них зручніші, інші - ні. Кожен такий показник можна характеризувати цілим комплексом властивостей, серед яких можна назвати наступні, важливі для окресленої предметної області і частково вже висвітлені вище:

- стійкість значень показника при однаковому стані користувача;

- незначний розкид значень показника при зміні стану людини;

- простота визначення показника з технічної точки зору.

Цим вимогам повністю задовольняють три показника з тільки що розглянутих: середній час утримання деяких кнопок, що часто набираються, середній час набору деяких поширених біграм, відносна кількість помилок при наборі тексту. Зазначені показники в цілому можна охарактеризувати словами «клавіатурний почерк»; розглянемо його докладніше, як показник особистості користувача, що підходить для задачі його розпізнавання.

Завдання набору тексту зустрічається при роботі на комп'ютері досить часто, незалежно від характеру програм, з якими найчастіше має справу

користувач. Найбільш повно цей вид діяльності проявляється при роботі в текстових редакторах типу Microsoft Word. Однак не слід думати, що набір тексту обмежений тільки цим додатком, адже насправді такий вид роботи зустрічається повсюдно:

- при обміні повідомленнями в соціальних мережах та Інтернет-пейджерах (типу Viber або ICQ)

- при пошуку інформації, коли спочатку завжди слід ввести її текстовий опис;

- при внесенні інформації в бази даних або електронні таблиці;

- навіть при тривіальному вході на сайт, що вимагає авторизації, необхідно набрати на клавіатурі свій логін і пароль, а це вже цілком конкретний набір символів (хоча і не дуже великий).

Грубо кажучи, можна сказати, що набір тексту - одна з основних функцій персонального комп'ютера. І для нас важливо, що кожен користувач має свій неповторний більш-менш стабільний клавіатурний почерк. Його стабільність порушується хіба що при зміні стану користувача: втоми, нервозності і т.д. Таким чином, цей показник ідеально підходить для цілей даної роботи.

Формалізуючи поняття клавіатурного почерку, можна сказати, що це набір середніх значень інтервалів часу утримань окремих клавіш, а також інтервалів між двома послідовними натисканнями на різні пари клавіш. Під клавішами маються на увазі кнопки, на які нанесені всі літери українського або англійського тексту (або того, з яким доводиться працювати людині). Такі інтервали для звичайних користувачів (що мають певний досвід роботи на клавіатурі) вимірюються в мілісекундах. Надалі будемо розглядати український алфавіт, як такий, що є досить поширеним в повсякденному застосуванні у досить великій частині Інтернету. Таким чином, клавіатурний почерк в першому наближенні є сукупністю двох комплексних математичних об'єктів:

- матриці набору біграм;

- вектора утримань окремих клавіш.

Матриця набору біграм має вигляд Т32х32 (по числу букв алфавіту), де tij - інтервал часу, який проходить з моменту набору і-тої літери алфавіту до натискання наступної за нею j-ої літери (в тому випадку, коли ці літери йдуть послідовно одна за одною - тобто утворюють біграми):

$$\boldsymbol{T}_{32x32} = \begin{pmatrix} 0 & 123 & 222 & \dots & 291 \\ 90 & 0 & 346 & \dots & 123 \\ 113 & 198 & 0 & \dots & 73 \\ \dots & \dots & \dots & \dots & \dots \\ 140 & 229 & 304 & \dots & 0 \end{pmatrix} \qquad (2.1)$$

Розглянемо матрицю (2.1), згідно з якою, наприклад, t12 = 123, і це означає, що середній час між натисканням клавіші «а» (перша буква алфавіту, тому що перший індекс дорівнює 1 і «б» (друга буква алфавіту, так як другий індекс дорівнює 2) становить 123 мс. Далі, наприклад t1 32 = 291, тобто середній час між натисканням на клавішу «а», а потім відразу на «я» становить 291 мс.

Спочатку матриця Т заповнюється в перший раз: при створенні профілю користувача і (бажано) коли він перебував в хорошому робочому настрої. При цьому йому пропонується виконати набір великого текстового фрагменту (як мінімум одна, а краще декілька сторінок). Оскільки цей процес виконується лише одноразово, то відповідні великі витрати часу на його виконання можна вважати обгрунтованими підвищеною точністю визначення почерку Т.

Далі кожен раз при необхідності оцінки стану користувача слід попросити його набрати на клавіатурі який-небудь текст поменше (припустимо з одного-двох абзаців). При цьому формується матриця Т' поточних значень часів набору біграм. Дві матриці Т' і Т порівнюються і по «відстані» між ними визначається ступінь відхилення поточного користувача від еталонного. Якщо дане відхилення перевищує певний поріг, аутентифікацію не можна вважати успішною.

Відстань між матрицями може оцінюватися різними способами, наприклад як відстань в 1024-вимірному (32x32 = 1024) евклідовому просторі (евклідова норма):

$$\Delta = \sqrt{\sum_{i=1}^{32}\sum_{j=1}^{32}\left(t'_{ij} - t_{ij}\right)^2} \qquad (2.2)$$

Основний недолік (2.2) - отримання норми в абсолютних одиницях. Це означає, що більш рідкісні біграми, на набір яких, отже, витрачається більше часу, можуть давати велику різницю і сильніше впливати на загальний результат, ніж ті, що більш часто зустрічаються при наборі (отже, які користувач набирає швидше, з меншими tij, і, отже, з меншими абсолютними відмінностями). Така ситуація звичайно ж неприпустима, і самий простий і зручний спосіб уникнути її - перейти до підсумовування відносних величин:

$$\Delta = \sqrt{\sum_{i=1}^{32}\sum_{j=1}^{32}\left(\frac{t'_{ij} - t_{ij}}{t_{ij}}\right)^2} \qquad (2.3)$$

Перевага формули (2.3) полягає в тому, що результат не залежить ані від розмірності величин, які підсумовуються (хоч це в даному прикладі і не дуже важливо, так як підсумовування підлягають тільки інтервали часу, однак при урахуванні інших за своєю природою чинників, це стане надзвичайно важливим), ані від їх співвідношення. Тому (2.3) являє більш адекватну оцінку «відстані» між двома матрицями клавіатурного почерку - поточної T' і еталонної T.

Цікавим є питання про величину тексту, що підлягає набору, при створенні еталону і при поточній перевірці. З точки зору статистики кількість текстової інформації для набору при створенні еталону T може бути точно порахована згідно з критерієм Стьюдента, однак, з огляду на те, що число біграм дуже велике (32x32 = 1024), то і обсяг тексту, який потрібно набрати при високому рівні значущості (наприклад, для перевірки гіпотези з ймовірністю 0,99) буде досить великим. В реальній практиці не вийде змусити користувача набирати

такий великий текст. Тому, в даному випадку слід керуватися критерієм розумної достатності, а не строгими математичними критеріями. Таким чином, зупиняємося на 1-2 сторінках тексту при знятті еталону та 0,5 стр. при поточній перевірці. Крім того, наведені нижче міркування дозволяють значно звузити кількість вимірюваних величин, що, загалом, піднімає точність статистичної обробки при малому обсязі вибірки.

Дійсно, виходячи з логічного аналізу тексту українською мовою, слід зазначити, що деякі біграми не можуть зустрічатися в ньому в принципі (наприклад, «УУ», «ЦЦ» і багато інших), а ще більше число біграм зустрічається досить рідко. Біграм, що зустрічаються більш-менш часто, насправді не так вже й багато. Це важливо з тих міркувань, що показує інформаційну надмірність матеріалу матриці T32x32.

Для аналізу стану користувача за допомогою клавіатурного почерку слід попросити його набрати на клавіатурі деякий текст. Для набору хоч якої-небудь правдоподібної статистики цей текст не повинен бути занадто малий, але, навіть якщо це буде великий абзац, біграми, які рідко зустрічаються, навряд чи взагалі там будуть присутні. Це означає, що без втрати точності оцінки клавіатурного почерку можна замість заповнення і порівняння всієї повної матриці T, оперувати з деяким вектором, який буде складатися з N часів набору деяких фіксованих біграм. Логічно вибрати N біграм, які найбільш часто зустрічаються в текстах українською мовою.

Слід відмітити, що така інформація зустрічається дуже рідко, тому для визначення кількості і конкретного типу біграм, які найбільш часто зустрічаються в українських текстах, реалізуємо невелику допоміжну програму, яка аналізує текстові файли. Програму напишемо в поширеному середовищі програмування Delphi. Єдиною вхідною інформацією для програми буде файл з текстом українською мовою обсягом не менше 100 Кб. Результатом роботи програми буде таблиця частоти біграм в цьому тексті – рис. 2.3.

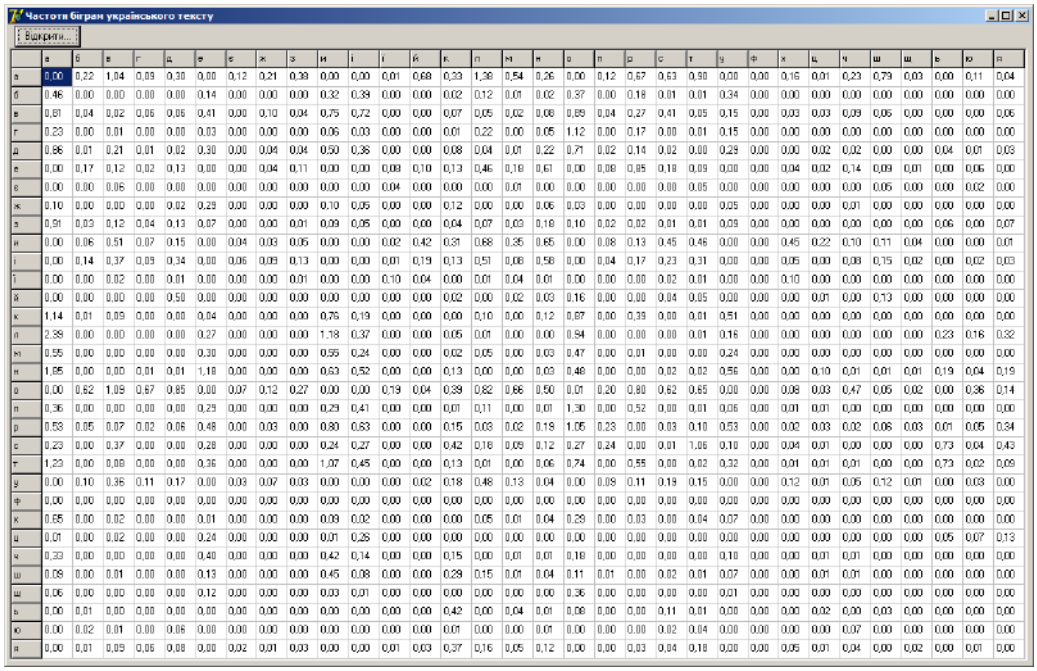


Рисунок 2.3 - Вікно допоміжної програми, написаної для аналізу частоти біграм, що зустрічаються в українському тексті.

Висхідний текст програми наведено в лістингу 2.1.

Лістинг 2.1.

```
unit Unit1;
interface
uses
  Windows,    Messages,    SysUtils,    Variants,    Classes,
Graphics, Controls, Forms,
  Dialogs, StdCtrls, Grids, StrUtils;
type
  TForm1 = class(TForm)
    OpenDialog1: TOpenDialog;
    StringGrid1: TStringGrid;
    Button1: TButton;
    procedure Button1Click(Sender: TObject);
    procedure FormCreate(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
const N=32;
var
  Form1: TForm1;
alf:string;

implementation
{$R *.dfm}
```

```
procedure TForm1.Button1Click(Sender: TObject);
var f:TextFile;
bi:array[1..N+1,1..N+1]of real;
prevs,nexts:char;
i,j,all:integer;
begin
  all:=0;
  if(OpenDialog1.Execute)then
    begin
      for i:=0 to N do
        for j:=0 to N do
          bi[i,j]:=0;
      AssignFile(f,OpenDialog1.FileName);
      Reset(f);
      Read(f,prevs);
      while(not Eof(f))do
        begin
          Read(f,nexts);
          i:=Pos(prevs,alf);
          j:=Pos(nexts,alf);
          if(i<>0)and(j<>0)then
            begin
              bi[i,j]:=bi[i,j]+1;
              all:=all+1;
            end;
          prevs:=nexts;
        end;
      CloseFile(f);
      for i:=1 to N do
        for j:=1 to N do

StringGrid1.Cells[j,i]:=FloatToStrF(bi[i,j]/all*100,ffFixed,5,2
);
      end;
end;

procedure TForm1.FormCreate(Sender: TObject);
var i:integer;
begin
  alf:='абвгдеєжзиіїйклмнопрстуфхцчшщьюя';
  for i:=1 to N do
    begin
      StringGrid1.Cells[0,i]:=alf[i];
      StringGrid1.Cells[i,0]:=alf[i];
    end;
end;

end.
```

В отриманій матриці частот виберемо 10 біграм, які зустрічаються в тексті найбільш часто – табл. 2.2 (аналіз виконано на прикладі твору «Кайдашева сім’я» загальним обсягом близько 270 Кб).

Таблиця 2.2 - Частоти 10 самих поширених біграм українського тексту.

Біграма	ЛА	НА	АЛ	ПО	ТА	НЕ	ЛИ	ОВ	ТИ	СТ
Відносна частота, %	2,39	1,85	1,38	1,30	1,23	1,18	1,18	1,09	1,07	1,06

Аналізувати будемо середній час набору цих N = 10 біграм за допомогою вектора:

$$\vec{t} = \{t_{ЛА}, t_{НА}, t_{АЛ}, t_{ПО}, t_{ТА}, t_{НЕ}, t_{ЛИ}, t_{ОВ}, t_{ТИ}, t_{СТ}\}$$

За методикою зазначеної вище, спочатку необхідно один раз сформувати еталонний вектор, а потім при кожній необхідності перевірки стану користувача слід формувати поточний вектор. Відстань між векторами будемо шукати за допомогою міри Евкліда, згідно з формулою, аналогічною (2.3):

$$\Delta = \sqrt{\sum_{i=1}^{N}\left(\frac{t'_i - t_i}{t_i}\right)^2} \qquad (2.4)$$

Величина Δ визначає розбіжність між еталоном, знятим в безпечному середовищі, і поточним користувачем.

Для практичної реалізації можна запропонувати формулу, по суті подібну (2.4), однак більш просту для програмування. Для спрощення в чисельнику (2.4) можна складати НЕ квадрати з подальшим знаходженням кореня, а модулі різниць ti' – ti. Ділити модулі таких різниць також слід не на квадрати, а саму величину ti:

$$\Delta = \sum_{i=1}^{N}\frac{\left|t'_i - t_i\right|}{t_i} \qquad (2.5)$$

Також, можна шукати не суму, а середнє арифметичне відносних нев’язок, позбавляючись, таким чином, від залежності величини Δ від кількості аналізованих величин N:

$$\Delta = \frac{1}{N}\sum_{i=1}^{N}\frac{\left|t'_i - t_i\right|}{t_i} \qquad (2.6)$$

В результаті обчислень по (2.6) отримаємо число у відносних одиницях (якщо помножити на 100% - то у процентах), яке показує, на скільки відсотків в середньому відрізняється еталонний користувач від поточного, що бажає пройти авторизацію (по 10 показникам - часам набору обраних біграм).

Крім розглянутих характеристик клавіатурного почерку, пов'язаних з набором біграм, також будемо враховувати час набору однієї літери δt_i, тобто різницю між моментом відпускання кнопки і моментом її натискання:

$$\delta t_i = t_{\text{відп } i} - t_{\text{нат } i}, \qquad (2.7)$$

где $t_{\text{відп } i}$ – момент часу, коли клавіша поточної i-тої літери була відпущена;

$t_{\text{нат } i}$ – момент часу, коли клавіша поточної i-тої літери була натиснута.

Усі такі часи δt_i утримання i-тої літери усереднюються за час набору авторизаційного тексту і потім порівнюються зі значеннями δt_i', які зберігаються в еталонному файлі відповідного користувача.

Тут, аналогічно попередньому комплексу характеристик, пов'язаних з набором біграм, виникає питання контролю всіх або тільки деяких з усіх доступних елементів такого типу. Іншими словами: чи контролювати утримання всіх без винятку букв, або вибрати тільки деякі, які найбільш часто зустрічаються. З огляду на те, що деякі букви типу "ф", "щ", "ш", "ц", "х" зустрічаються досить рідко, недоцільно контролювати будь-які характеристики, з ними пов'язані, з огляду на велику імовірність повної відсутності таких букв у тексті для набору. У якості найбільш статистично значущих можна вибрати літери, що утворюють вищенаведені біграми (табл. 2.2), що програмно реалізується у вигляді змінної, що містить весь «алфавіт», з якого скомбіновані ці біграми:

alf="авеилностп";

Оскільки букв в цьому «алфавіті» досить багато (також 10), то з метою спрощення програмної реалізації ці ж букви і можна контролювати на предмет часу утримання окремих клавіш.

На підставі часів (2.7) будемо формувати вираз виду (2.6), але не для часів набору біграм, а для часів утримання обраних букв:

$$\delta\Delta = \frac{1}{N}\sum_{i=1}^{N}\frac{\left|\delta t'_i - \delta t_i\right|}{\delta t_i} \qquad (2.8)$$

Таким чином, на підставі часів утримання обраних 10 букв формується ще одне середнє арифметичне нев'язок (2.8).

І ще однією досить важливою характеристикою клавіатурного почерку є інтенсивність здійснення помилок, друкарських помилок, тобто невірного набору. Так, загальновідомо, що будь-який оператор технічного пристрою (в т.ч. і звичайний пересічний користувач ПК) в процесі роботи може робити помилки, що, зокрема, залежить від його ступеня володіння ПК в цілому. Втім іноді люди, що дуже повільно набирають текст, роблять це занадто ретельно і практично не роблять помилок, правда, набираючи при цьому всього декілька букв за хвилину. Але і в цьому випадку інтенсивність виникнення помилок можна використовувати як елемент аутентифікації, важливо, щоб цей показник контролювався в комплексі з раніше розглянутими, і тоді він може дати досить корисний вклад до всієї методики оцінки згідно біометричного протоколу.

При комп'ютерному наборі кількість помилок можна оцінювати за двома напрямками:

- помилки, які виникли в процесі набору, але були помічені і виправлені;

- помилки, які виникли в процесі набору і не були помічені або були помічені, але з якоїсь причини не були виправлені.

Помилки другого типу контролювати автоматично досить складно з алгоритмічної точки зору, тому в подальшому будемо говорити тільки про помилки першого типу, тобто такі, які виправлялися відразу ж під час набору тексту користувачем. Виправляти помилки можна різними способами, однак

найбільш традиційним при послідовному наборі тексту є натискання клавіші BackSpace кілька разів, щоб стерти останні, набрані невірно, символи і дістатися до помилки.

Отже, кількість виправлених помилок може бути оцінений по натиснень на кнопку Backspace. В цьому випадку потрібно враховувати тільки одне з декількох поспіль натискань кнопки повернення, що випливає з таких міркувань. Середній користувач набирає текст не по одному символу, а групами. Якщо в групі символів є хоча б одна помилка, користувач знайде її не відразу, а тільки після набору всієї групи, коли подивиться на екран. Як вже зазначалося раніше, зазвичай помилка виправляється відразу, шляхом натискання кілька разів клавіші Backspace для того, щоб повернутися до символу, який містить помилку. Таким чином, всі ці численні натискання викликані однією помилкою, і тому натискання Backspace, які йдуть поспіль, повинні асоціюватися з однією помилкою. Ще один варіант виправлення помилки - стрілками «вліво» - «вправо» на клавіатурі встановити курсор прямо перед неправильним символом і натиснути Backspace один раз. При цьому одній помилці відповідає одне натискання кнопки повернення.

Таким чином, без особливого збитку для точності можна вважати, що для однієї помилки, допущеної при комп'ютерному наборі тексту, відповідає від 1 до *M* поспіль натискань на кнопку Backspace (при такому підході ми нехтуємо поправками помилок за допомогою клавіші Delete, яка, з огляду на її відокремлене становище на клавіатурі - на відміну від кнопки BackSpace, яка інтегрована в блок буквених клавіш, використовується зазвичай для редагування вже набраного тексту, але не для виправлення помилок при безпосередньому наборі).

При реалізації алгоритму будемо збільшувати лічильник виправлених помилок *E* на одиницю в тому випадку, якщо натиснута кнопка Backspace, і безпосередньо перед нею була натиснута інша кнопка. Якщо натиснута кнопка Backspace, але попередня натиснута клавіша також Backspace, то лічильник

помилок збільшувати не потрібно, тому що, ймовірно, йде повернення до одного конкретного неправильно введеному символу. Абсолютне число помилок *E* потрібно поділити на загальне число набраних символів *All*, щоб перейти до відносного числа помилок, яке і можна порівнювати з еталоном:

$$R = \frac{E}{All}.$$

В результаті обчислення кількості помилок отримаємо число *R*, яке також знімається під час первинного набору тексту (тоді ж, коли формується і еталон набору біграм і утримань окремих клавіш). Як результат порівняння будемо приймати відносне збільшення числа помилок за вказаним показником:

$$\Delta_R = \frac{|R' - R|}{R}. \qquad (2.9)$$

Формулу (2.9) слід об'єднати з формулами (2.6) і (2.8) так, щоб інтегрувати всі фактори, які враховуються (час набору біграм, час утримання окремих клавіш, відносне число помилок - все разом це і буде формувати клавіатурний почерк в рамках даної роботи):

$$\Delta_{res} = \frac{\Delta + \delta\Delta + \Delta_R}{3} = \frac{1}{3}\left(\frac{1}{N}\sum_{i=1}^{N}\frac{|t'_i - t_i|}{t_i} + \frac{1}{N}\sum_{i=1}^{N}\frac{|\delta t'_i - \delta t_i|}{\delta t_i} + \frac{|R' - R|}{R}\right). \qquad (2.10)$$

У (2.10) три різних за характером нев’язки усереднюються (щоб не відхилятися від середніх, об'єктивних величин), хоча теоретично ці три групи показників можуть бути по-різному важливі для виконання процесу аутентифікації. Таким чином, замість формули (2.10) в перспективі можна використовувати формулу виду:

$$\Delta_{res} = a_1\Delta + a_2\delta\Delta + a_3\Delta_R, \qquad (2.11)$$

де a_i - вагові коефіцієнти розглянутих ознак, які можуть бути використані для аутентифікації по клавіатурному почерку. Для визначення цих вагових коефіцієнтів потрібне окрема об'ємна робота, тому в даній роботі обмежимося залежністю (2.10), хоча перспективною є і формула (2.11).

Отже, візьмемо за основу (2.10) при реалізації протоколу аутентифікації в ІКМ біометричним методом, а саме, на базі аналізу клавіатурного почерку користувача.

2.2. Можливості удосконалення обраного протоколу автентифікації в ІКМ

Інфраструктура є базовою основою для створення середовища розробки і виконання.

Ще одною перевагою мови C# є можливість розробки на основі платформи .Net Core. Дана платформа є розробленою на основі класичного .Net Framework, про на відміно від нього є модульною платформою для розробки із відкритим кодом. Дана платформа є сумісною з білістю операційних систем і є кроссплатформенною.

Розглянутий у попередньому розділі спосіб біометричної аутентифікації працює на локальному комп'ютері та потребує удосконалення для впровадження у розподілених системах типу «клієнт-сервер» - рис. 2.4.

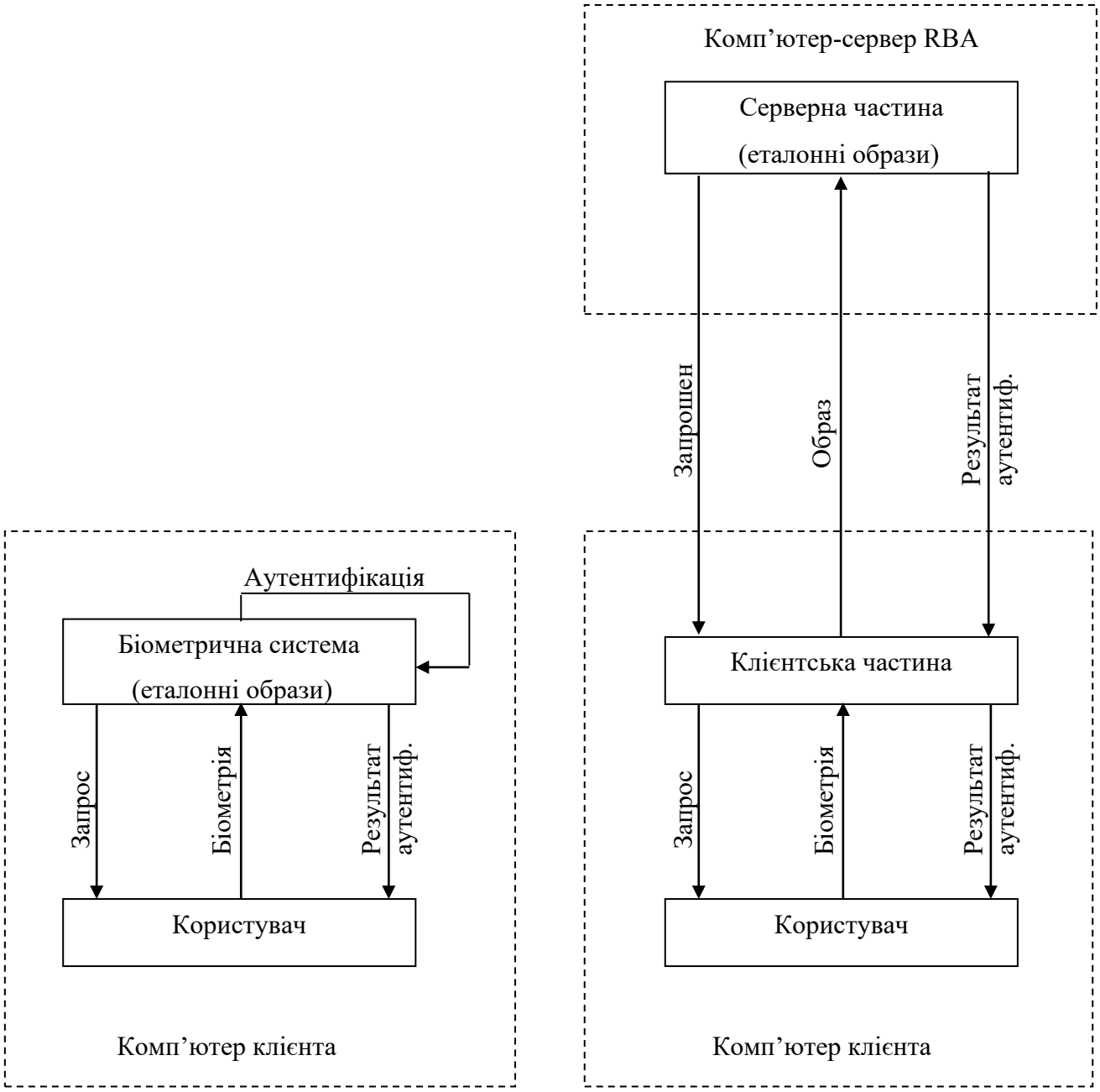


Рисунок 2.4 - Впровадження розподіленого режиму у протоколі біометричної аутентифікації RBA.

2.3 Обґрунтування вибору засобів розробки для реалізації удосконаленого протоколу автентифікації в ІКМ

Перед виконанням будь-якої програмної реалізації слід визначитися із кількома концептуальними питаннями, пов’язаними із нею, а саме:

- яку технологію програмування найкраще застосувати для реалізації даного алгоритму при даному комплексі умов;
- яку мову програмування, що підтримує обрану технологію програмування, доцільніше всього застосувати для програмної реалізації у наявних умовах;
- яке середовище розробки (або комплекс простих окремих засобів розробки) краще всього застосувати для даної програмної реалізації.

Тільки отримавши обґрунтовані відповіді на ці запитання, можна переходити безпосередньо до етапу програмування.

2.4 Висновок

В другому розділі пройшов аналіз існуючих протоколів автентифікації для можливості удосконалення обраного протоколу в ІКМ. Обгрунтований вибір засобів розробки для реалізації удосконаленого протокому ІКМ.

Також пройшов підбір можливих пакетних програм для придбання щоб за допомогою цих систем додатково забезпечити додатковий захис ІКМ.

3 ПРОЕКТНІ РІШЕННЯ ЩОДО РЕАЛІЗАЦІЇ УДОСКОНАЛЕНОГО ПРОТОКОЛУ АВТЕНТИФІКАЦІЇ В ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНІЙ МЕРЕЖІ

3.1 Вибір технології розробки з урахуванням особливостей предметної галузі.

Отже, першочерговим питанням, яке постає перед розробниками практично будь-якого програмного забезпечення, є вибір моделі або технології його розробки. Такими, що реально широко використовуються на сьогоднішній день у виробничій практиці, є технології структурного (процедурного) та об'єктно-орієнтованого програмування. Кожна з них має свої особливості, переваги і недоліки, які розглянемо докладніше.

Структурне, або як його ще називають, процедурне програмування засноване на використанні окремих структурних блоків - в першу чергу, підпрограм (процедур і функцій).

Історично перші комп'ютерні програми були відносно простими і мали пакетний режим роботи: отримуючи на вхід якусь інформацію (можливо, навіть на перфокарті) вони виконували певний обсяг операцій з обробки цих даних і видавали результат. При цьому існувала сувора функціональна залежність виходу від входу – рис. 3.1

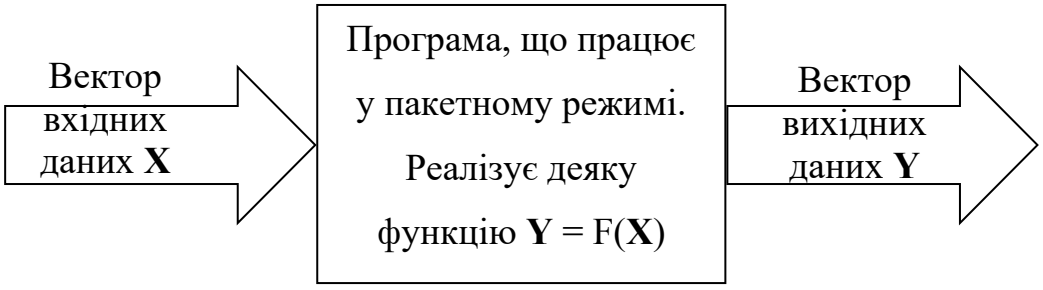


Рисунок 3.1 - Схема роботи пакетної програми.

Незважаючи на відсутність явного зв'язку функціональності і структури, пакетні програми в основному мали просту лінійну послідовність виконання.

Тобто в них були практично відсутні будь-які підпрограми, принаймні, використання підпрограм не було важливим елементом самої методики програмування.

В міру ускладнення функціональності програмного забезпечення, змінювалася і його внутрішня структура: поступово розвинулася інтерактивна модель взаємодії користувача і програми – рис. 3.2 Програми стали запитувати інформацію і активно реагувати на дії людини. Ускладнення функціональності привело до відповідного ускладнення програмного коду, який, в першу чергу, став досить обширним у розмірах. Обширним для того, щоби середньостатистична людина-програміст без всяких спеціальних хитрувань швидко і легко розібралася з незнайомим кодом. Розміщувати код у простій лінійній послідовності без виділення великих блоків стало незручно, в першу чергу, для розуміння цього коду.

Тут слід зазначити психологічні особливості сприйняття людиною складних «великих» завдань. Неструктуроване «велике» завдання (наприклад, написання дипломної роботи) зазвичай викликає певний психологічний ступор і, як результат, повну неможливість поступово розібратися з ним. Людині зручно розбити проблему на не надто велику (зазвичай до десятка, а краще 3-4) кількість завдань (наприклад, розділів у дипломній роботі), не замислюючись про реалізацію кожного з них. Коли є ясність і розуміння проблеми на найвищому рівні абстракції, слід приступати до деталізації підзадач, кожну з яких слід розбити на окремі «підпідзадачі» тобто підзадачі нижчого рівня, більш дрібні. Уже після такого розбиття слід аналізувати всі перераховані підзадачі. На певному етапі зупиняються і виконують не розбиття чергової підзадачі на більш дрібні, а безпосередню її реалізацію в програмних кодах.

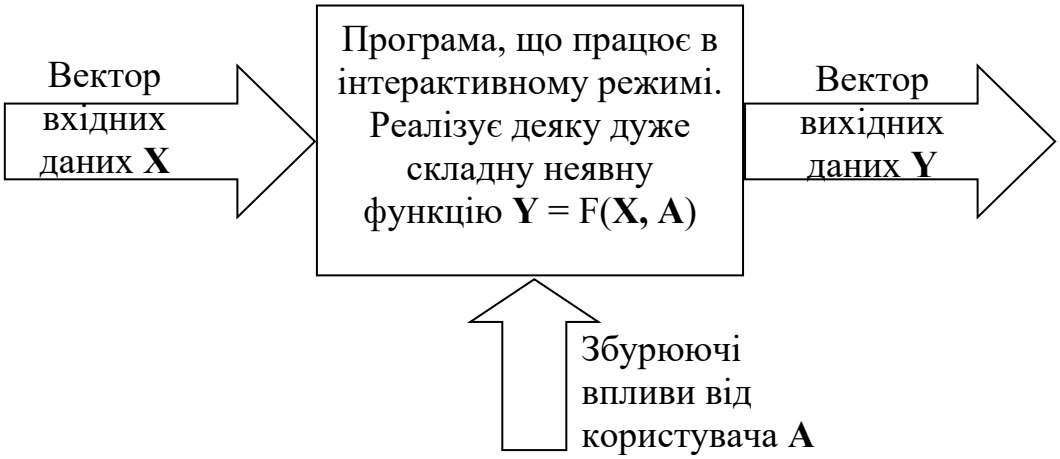


Рисунок 3.2 - Схема роботи програми, що активно взаємодіє з користувачем

Отже, можна сказати, що розвиток методик програмування відбувався паралельно з розвитком призначеного для користувача інтерфейсу: і грубо кажучи, розвиненому інтерфейсу командного рядка відповідає парадигма структурного програмування.

Говорячи більш формалізовано, структурне програмування має на увазі побудову програми відповідно до трьох основних принципів: слідування, розгалуження, повторення.

Слідування означає, що оператори і блоки програмного коду слідують і виконуються один за іншим. Розгалуження реалізується різними умовними операторами типу if, і дозволяє вибирати один з декількох подальших варіантів виконання програми. Повторення зазвичай відносять на рахунок циклів (багаторазових повторів однієї і тієї ж ділянки коду, що йдуть підряд), хоча цей же принцип можна віднести і до підпрограм.

Взагалі ж, структурне програмування іноді називають застарілою методикою програмування, на зміну якій разом з віконним інтерфейсом прийшло об’єктно-орієнтоване програмування (деякі сучасні мови програмування загального призначення навіть не дозволяють створити структурну програму, тільки об’єктно-орієнтовану – як, наприклад, Visual C# чи Java). Проте, при створенні невеликих програм (наприклад, до 10000 рядків коду і без передбачуваного розширення) застосування цієї методики

програмування однозначно більш виправдано і відповідний код набагато краще сприймається, ніж його об’єктно-орієнтований варіант.

Суть же методології об’єктно-орієнтованого програмування полягає в тому, що система розглядається, як сукупність окремих сутностей - об’єктів, які мають набір якихось своїх внутрішніх параметрів - властивостей, а також можуть взаємодіяти між собою за допомогою деяких дій - викликів методів (або трохи більше непрямим чином - шляхом надсилання повідомлень, оброблюваних методами об’єктів; для цього необхідна присутність активної сутності, яка роздає повідомлення адресатам, як, наприклад, менеджер вікон в ОС Windows).

Якщо говорити про програмний код, то для того, щоб оперувати об’єктом, його спочатку потрібно створити. Об’єкти створюються як змінні, у яких типом виступає клас об’єкта. Клас - це просто опис, які властивості можуть мати об’єкти такого типу (тобто яку інформацію вони можуть зберігати), і які у них є методи (тобто які дії вони можуть виконувати). Об’єкт - це набір значень, чому саме рівні властивості даного об’єкта (свої методи кожен об’єкт отримує від свого класу, тобто методи однакові у всіх об’єктів, що належать даному класу).

Для чого потрібен цей специфічний підхід, адже самі по собі об’єкти не додають нічого корисного (навпаки, введення об’єктів ускладнює програму, вносить в неї нові сутності)? Виявляється, реалізовуючи всі сутності, необхідні, згідно з алгоритмом, для роботи програми, у вигляді класів і об’єктів, ми спрощуємо її розуміння для самих себе. Саме тому ОО-підхід рекомендується до застосування для великих проектів (більше десятків тисяч рядків коду), коли утримувати «в голові» всю систему цілком стає важко. Можна сказати, що розбиття програми на об’єкти і проектування їх класів наближає розуміння предметної області до звичного людського образу мислення (в разі «великих» проектів). Людина мислить класами, об’єктами і зв’язками між ними.

Порівняльна складність проектів, що мають однакову функціональність, але побудованих по-різному (згідно структурному і об’єктно-орієнтованого

підходів до програмування), як функція їх обсягу показана на рис. 2.7. З графіка рис. 3.3 слід, що, якщо потрібно реалізувати продукт з невеликою функціональністю (тобто кількість рядків коду, що її реалізують буде очевидно невеликою, порядку кількох тисяч рядків), то це краще робити без застосування класів, так як вони будуть тільки ускладнювати всю справу. Якщо ж програма має більш-менш значну функціональність, а значить, реалізується хоча б кількома тисячами рядків коду, то вже є сенс замислюватися про застосування об'єктно-орієнтованого підходу. Однозначно будуватися за принципами ООП повинні програми, які мають 10000 рядків коду і більш.

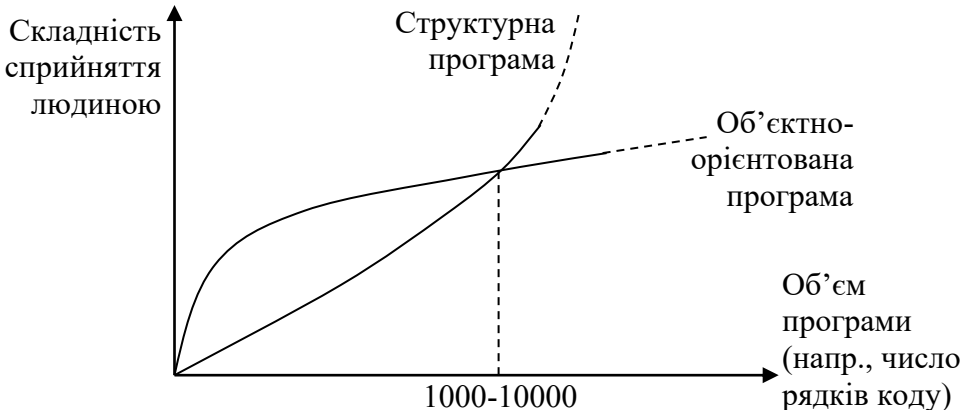


Рисунок 3.3 - Порівняльна складність висхідного тексту двох програм, що мають однакову функціональність, але реалізованих по-різному: згідно об'єктно-орієнтованому та структурному підходам.

Відзначимо, що часто крім розглянутих міркувань, також на вибір методики програмування впливають інші чинники, наприклад, можливість майбутнього розширення функціональності, створення якомога більш зрозумілого коду (для роботи над проектом цілої команди, а не одного програміста), або просто побажання замовника застосувати найбільш сучасний підхід до програмування (чи навпаки, дозвіл на використання вікових, перевірених часом технологій).

Крім розбиття (декомпозиції) всієї предметної області на об’єкти (класи) і співвідношення між ними, також ОО-підхід має на увазі дотримання трьох основних його принципів: інкапсуляція, наслідування, поліморфізм.

Під інкапсуляцією мається на увазі об’єднання даних (значення властивостей класу у деякого конкретного об’єкта) та засобів їх обробки (методи класу). Це знову ж таки зручно психологічно, так як дозволяє реалізовувати окремі завершені сутності - класи, які самі обробляють свої дані. Звернення до об’єктів цих класів відбувається за допомогою методів, що утворюють інтерфейс класу.

Наслідування дуже корисно, тому що дозволяє сильно скоротити обсяги повторюваного коду (до чого потрібно завжди прагнути при розробці будь-якого програмного забезпечення). Згідно з цим принципом виділяється клас, який має загальний набір властивостей і методів для декількох більш розширених класів. Цей клас оголошується батьком, базовим класом для декількох похідних від нього (нащадків, спадкоємців). Всі класи-нащадки успадковують від базового всі його властивості та методи, але до цього ще мають свої власні оригінальні властивості і / або методи.

Наприклад, клас Студент є похідним від класу Людина, тому що кожна людина має властивість Ім’я, Прізвище, метод Відпочити(). Однак у Студента є свої специфічні властивості і / або методи, які як раз і відрізняють його від просто Людини: СереднійБал, НомерЗаліковки, ЗдатиЕкзамен(), і т.д.

При наслідуванні іноді методи батьківського і похідного класу мають однакове призначення, але реалізовуються по-різному. Такі методи називаються перевантаженими. Наприклад, метод Відпочити() у класу Людина реалізується як відпочинок на дивані, а у класу Студент - як похід в клуб. При цьому ще раз підкреслимо, що призначення методу в обох випадках одне і той саме.

Поліморфізм є можливістю деякої функції приймати об’єкти як батьківського, так і похідних класів, і вміти викликати перевантажені методи

саме того класу, об’єкт якого був переданий в функцію. Слід сказати, що це досить специфічна можливість і в загальному багато програмістів використовують ОО-підхід і без звернення до поліморфізму.

Нехай, наприклад, у програмі є функція ПровестиВихідні(), припустимо яка не належить якомусь класу (хоча це не принципово). Нехай аргументом цієї функції є об’єкт класу Людина. Тоді в неї можна передавати об’єкти всіх похідних від Людини класів: Студент, Службовець, Пенсіонер, і т.д., тому що всі вони є Людиною (спадкоємці цього класу). Ясно, що ця функція повинна включати різні дії: ПрибратиКвартиру(), ПітиНаРинок(), і в тому числі Відпочити(). Так ось поліморфізм дозволяє всередині цієї функції просто вказати назву методу Відпочити(), не вказуючи якого саме класу він повинен бути викликаний, а вже в процесі виконання програми, якщо в функцію переданий об’єкт класу Студент, то викликається саме його метод Відпочити(), а якщо переданий об’єкт класу Пенсіонер, то автоматично викликається саме його метод Відпочити(), і т.д. Кажуть, що функція ПровестиВихідні() - поліморфна, і вона є такою завдяки тому, що реалізує принцип поліморфізму.

Важливими поняттями в ООП також є: статичні члени класу, абстрактні методи і класи, дружба функцій і класів, і т.д.

Грунтуючись на перерахованих особливостях двох існуючих методів програмування, вибираємо об’єктно-орієнтований підхід, як більш сучасний, що відповідає середнім масштабам проектованого ПЗ, а також вимогам до складності.

3.2 Вибір мови програмування

Перейдемо до розгляду програмних засобів, і основним із них, що дозволяє реалізовувати ОО-проекти, можна вважати середовище розробки Microsoft Visual Studio (у ньому нас цікавитиме можливість створення C++ проекту). Microsoft Visual Studio - це серія продуктів фірми Майкрософт, які включають

інтегроване середовище розробки програмного забезпечення та ряд інших інструментальних засобів. Ці продукти дозволяють розробляти як консольні програми, так і програми з графічним інтерфейсом, в тому числі з підтримкою технології Windows Forms, а також веб-сайти, веб-застосунки, веб-служби як в рідному, так і в керованому кодах для всіх платформ, що підтримуються Microsoft Windows, Windows Mobile, Windows Phone, Windows CE, .NET Framework, .NET Compact Framework та Microsoft Silverlight.

Якщо ж коротко сказати про характеристики мови програмування, то C++, на якій була розроблена програмна система, являється високорівневою компільованою мовою, загального призначення зі строгою типізацією, яка підходить для створення самих різних додатків. На сьогоднішній день C++ є однією із найпопулярніших і найпоширеніших мов.

3.3 Особливості програмної реалізації окремих складових удосконаленого протоколу автентифікації в ІКМ

Як було розроблено у попередньому розділі, програмне забезпечення складається із двох складових серверної – рис. 3.4 та клієнтської – рис. 3.5

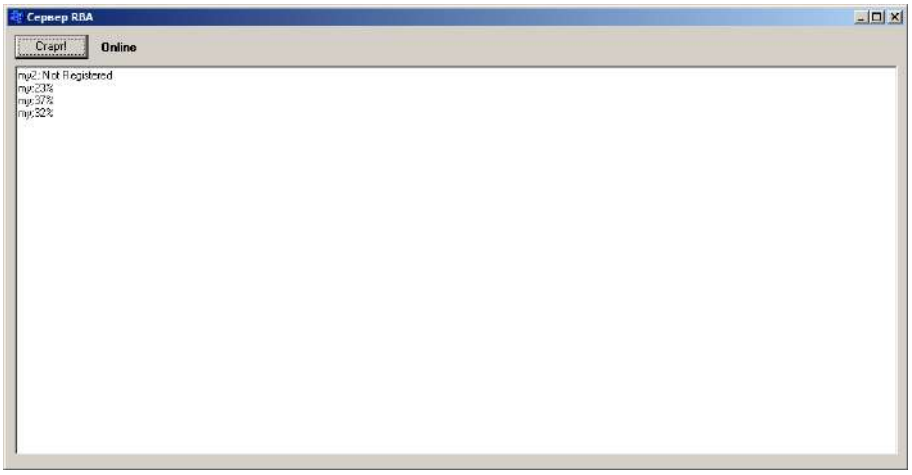


Рисунок 3.4 - Загальний вигляд вікна серверної частини розробленого програмного комплексу, що реалізує протокол RBA.

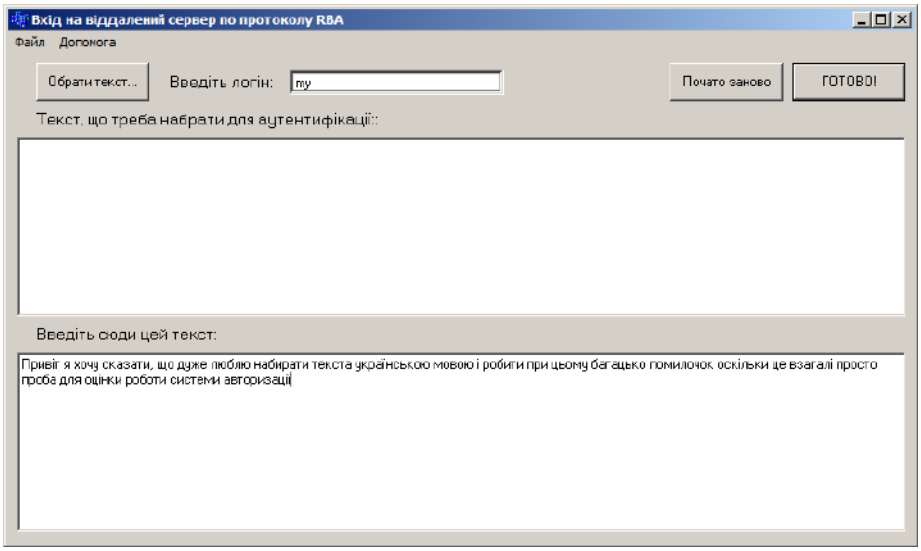


Рисунок 3.5 - Загальний вигляд вікна клієнтської частини розробленого програмного комплексу, що реалізує протокол RBA.

3.4 Загальна характеристика отриманого програмного продукту, опис інтерфейсу користувача, інструкція по експлуатації

На рис. 3.4 видно, що інтерфейс серверної частини достатньо простий і має фактично лише кнопку для запуску сервера (якщо він не запущений, а клієнт пробує під’єднатися до нього, то виникає помилка, як на рис. 3.3).

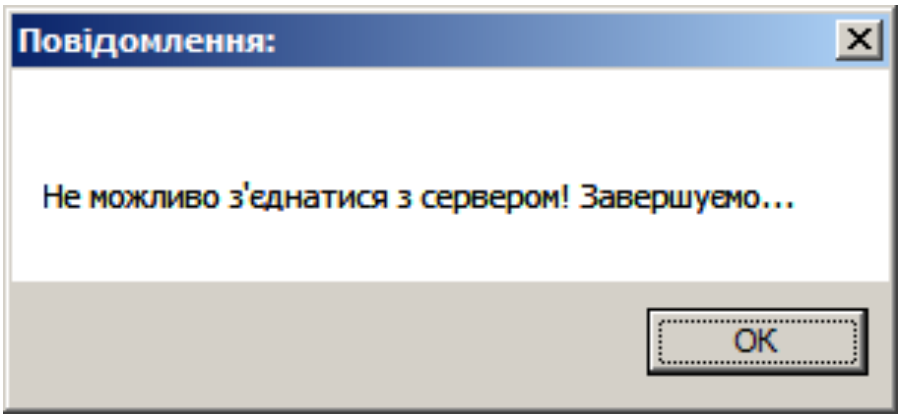


Рисунок 3.6 - Вікно про помилку, що виникає при спробі під’єднання по розробленому протоколу RBA до не активного серверу.

Коли сервер знаходиться у робочому режимі, то у великому текстовому полі відображаються результати аутентифікацій – успішні, неуспішні, спроби під’єднання користувачів із недійсними логіками – рис. 3.1.

Якщо клієнт вказав вірний логін, але не пройшлов біометричну аутентифікацію, то йому відображується вікно про провал цього процесу – рис. 3.7

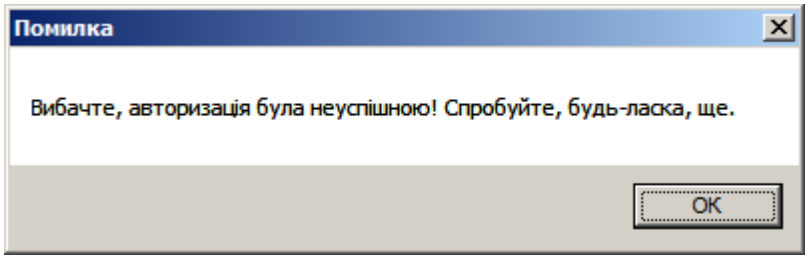


Рисунок 3.7 - Вікно про помилку, що виникає при спробі неуспішній аутентифікації по розробленому протоколу RBA існуючого клієнту.

3.5 Висновок

На етапі розділу 3 було розпочате виконання проектного рішення, що до реалізації удосконалення протоколу автентифікації в ІКМ з урахуванням предметної галузі. Також вибрана мова програмування для розробки програмного забезпечення, описана загальна характеристика отриманого програмного продукту та опис інтерфейсу по експлуатації.

РОЗДІЛ 4 ТЕСТУВАННЯ ПРОГРАМИ ЗАХИСТУ ІНФОРМАЦІЇ

4.1 Тестування системи

Для тестування системи було створено мережу із декількох віртуальних машин на яких було встановлено клієнт системи захисту від вити току даних, а також додаток який емалював діяльність роботи користувача, а саме здійснював операцію переміщення файлів та копіювання текстів і файлів. Сервером виступала віртуальна машина під керування Linux Ubuntu. На якій був розгорнутий контейнер docker в якому містився сервер керування системою захисту.

Додатки емалювання діяльності користувача мали спільне сховище із документами, які використовувалися для операцій копіювання/переміщення файлів.

На рисунку 4.1 зображений графік кількості запитів і кількість витоків які було дійсно скоєно. Я обрав початковим значенням 1000 операцій на годину і поступовим зменшенням із кроком 100. Кількість витоків задавалася випадково оператором тестування.

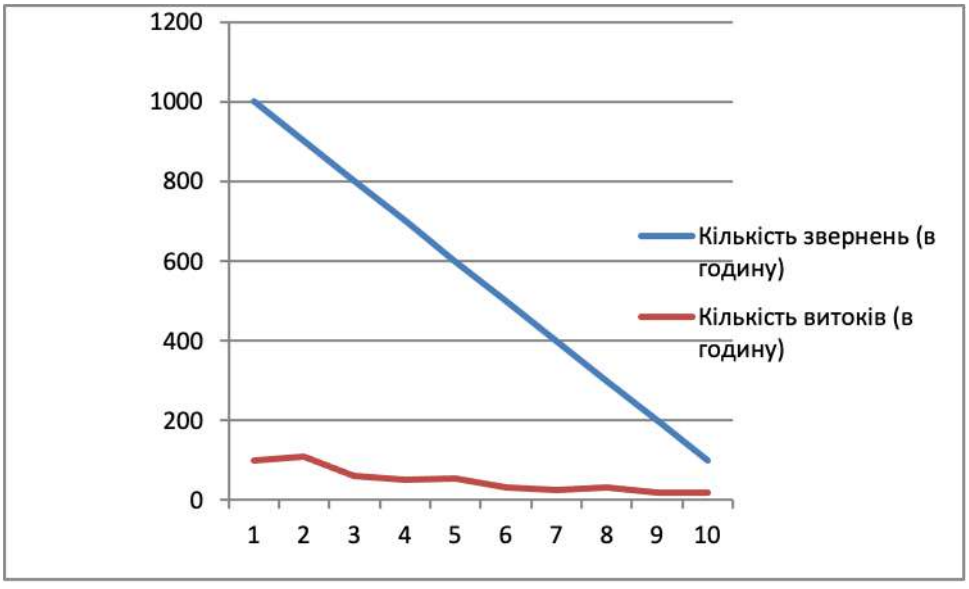


Рисунок 4.1 - Графік кількості запитів і кількості реальних витоків

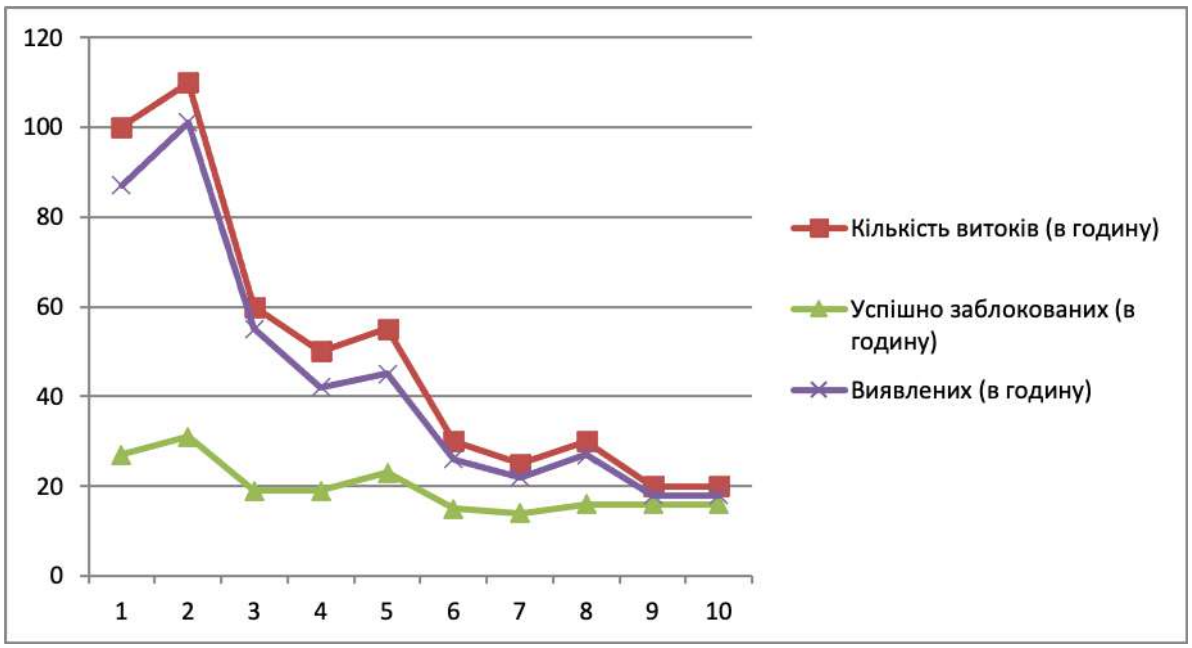


Рисунок 4.2 - Графік кількості успішно виявлених і заблокованих витоків

Як видно з рисунку 4.2 і 4.3 кількість успішних виявлень витоків доволі висока і становить близько 90%. Проте кількість успішних блокувань доволі критично залежить від кількості запитів. Причиною даної залежності є архітектура модуля лінгвістичного аналізу тексту, а саме рішення із організації аналізу тексту у вигляді черги запитів. За рахунок цього модуль не може швидко закінчити аналіз тексту до моменту здійснення операцій над файлом із конфіденційною інформацією. Також на ефективність вливає те, що тестування відбувається на одній реальній машині. Дане тестування вказало на проблеми даної архітектури. Особливої уваги потрібно надати потужностям серверу і архітектурі модуля лексичного аналізу, а саме підвищення ефективності методу оцінки і розпаралелювання процесу його здійснення.

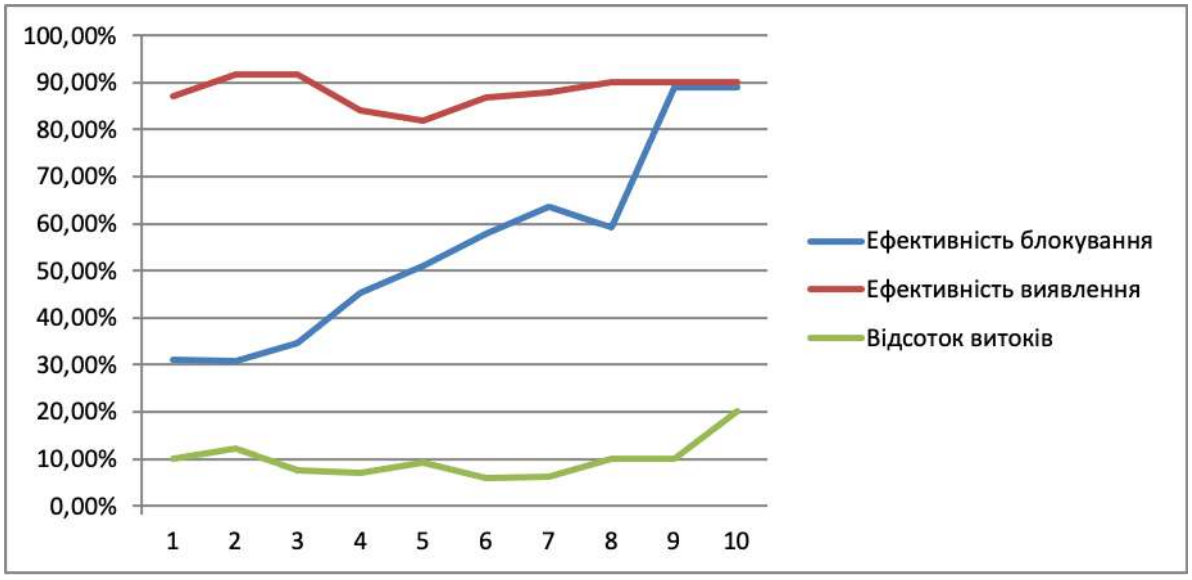


Рисунок 4.3 - Графік ефективності виявлення і блокування витоків даних

В результаті аналізу створеного програмного продукту встановлено, що при указанні порогового значення аутентифікації □ = 0,26 = 26% показник хибних відмов законним користувачам складає FRR < 5%, а імовірність успішної авторизації зловмисника складає FAR < 15 %.

Як висновок можна сказати, що розроблений протокол RBA може застосовуватися для захисту шляхом розподіленої біометричної (поведінкової) аутентифікації комерційної інформації середнього рівня конфіденційності (максимум – із грифом «Для службового користування»).

і

4.2 Впровадження системи в промислову експлуатацію

Так як система виконана у вигляді класичної клієнт-серверної архітектури, то для роботи системи необхідно виділений сервер із встановленою серверною частиною системи, а на робочі станції на яких потрібно відслідковувати витоки даних необхідно встановити клієнт.

Особливу увагу потрібно приділити потужностям роботи серверу.

Підтримка та адміністрування системи здійснюється за допомогою вебпанелі. Оновлення системи необхідно робити в разі потреби зміни

функціоналу системи захисту або через значне оновлення технічної бази інформаційно-комунікаційної системи.

4.3 Висновок

Чертвертий розділ являється завершальним і підсумковим розділом до кваліфікаційної роботи в ньому був проведений детальний тест програмного продукту тестування самої системи та серверної частини.

Було впроваджено систему в промислову експлуатацію.

ВИСНОВКИ

В роботі виконано розробку та реалізації удосконаленого протоколу аутентифікації для захисту інформації у інформаційно-комунікаційних мережах, тобто у розподіленому режимі. Рішення реалізовано мовою C/C++ програмно у вигляді комплексу, який працює за технологією «клієнт-сервер», тобто містить дві частини:

- клієнтська збирає поведінкові характеристики клієнта (А саме особливості його клавіатурного почерку) та надсилає на сервер;

- серверна частина порівнює отриманий образ із еталонами, що на ній зберігаються і надсилає клієнтові висновок – успішна авторизація, чи ні.

Дана робота реалізована у вигляді окремого продукту, який надає користувачеві клієнтської частини вердикт «так», чи «ні», але може бути інтегрована у будь-яку більш складну СЗІ.

В роботі проаналізовано особливості українського тексту на предмет використання його для розпізнавання клавіатурного почерку, розроблено спеціальний програмний продукт для відповідних статистичних досліджень та визначено найуживаніші біграми.

Робота являє собою завершеною і може використовуватися на практиці, а також – як основа для подальших робіт.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Грайворонський М. В., Новіков О. М. Безпека інформаційно-комунікаційних систем. К., 2009.

2. Богуш В. М., Довидьков О. А., Кривуца В. Г. Теоретичні основи захищених інформаційних технологій. К., 2010.

3. Андреєв В. І., Хорошко В. О., Чередниченко В. С., Шелест М. Є. Основи інформаційної безпеки. К., 2009.

4. Домарев В. В. Безопасность информационных технологий. Системный подход. — К.: ООО ТИД Диа Софт, 2004. — 992 с.

5. Брюс Шнайер. Прикладная криптография. Протоколы, алгоритмы, исходные тексты на языке Си = Applied Cryptography. Protocols, Algorithms and Source Code in C. — М.: Триумф, 2002. — 816 с.

6. J. G. Steiner, B. C. Neuman, and J. I. Schiller, ‘‘Kerberos: An Authentication Service for Open Network Systems,’’ pp. 191-202 in Usenix Conference Proceedings, Dallas, Texas (February, 1988).

7. John T. Kohl, B. Clifford Neuman, Theodore Y. Ts’o. The Evolution of the Kerberos Authentication Service [Електронний ресурс], Режим доступу: ftp://ftp.isi.edu/isi-pubs/rs-94-412.pdf.

8. E. Hammer-Lahav, Ed. The OAuth 1.0 Protocol // IETF. — 2010. — 63 p.

9. Ryan Boyd. Getting Started with OAuth 2.0. — Sebastopol: O’Reilly Media, Inc., 2012. — p. 67.

10. D. Hardt, Ed. The OAuth 2.0 Authorization Framework. - RFC 6749: Microsoft, 2012. – 75 p.

11. OpenID Authentication 2.0 Specification [Електронний ресурс]. – OpenID Foundation. Режим доступу: http://openid.net/ (дата звернення 5.12.2019 р.).

12. Microsoft and Google Both Ship OpenID [Електронний ресурс]. – OpenID Foundation. Режим доступу: http://openid.net/ (дата звернення 5.12.2019 р.).

13. Мельников В. П., Клейменов С. А., Петраков А. М. Информационная безопасность и защита информации 3-е изд. Учеб. Пособие для студ. высш. учеб. заведений/В. П. Мельников, С. А. Клейменов, А. М. Петраков.-М.:2008. — 336 с.

14. Ленков С. В., Перегудов Д. А., Хорошко В. А. Методы и средства защиты информации: В 2 т. К., 2008.

15. Задірака В. К., Кудін А. М., Людвиченко В. О., Олексюк О. С. Комп'ютерні технології криптографічного захисту інформації на спеціальних цифрових носіях. К.; Т., 2007.

16. Богуш В.М., Довидьков О.А. Теоретичні основи захищених інформаційних технологій - К.: ДУІКТ, 2006. - 508 с.

17. Галицький А.В., Рябко С.Д., Шаньгин В.Ф. Защита информации в сети – аналіз технологий и синтез решений . – М.: ДМК Пресс, 2004, 2004. – 616 с.

18. Глинских А. Мировой рынок систем електронного документооборота // Информационный бюллетень Jet Info. - 2002. - № 8. – 40 с.

19. Закон України "Про електронні документи та електронний документообіг" від 22 травня 2003 р. № 851-IV.

20. Закон України "Про електронний цифровий підпис" від 22 травня 2003 р. № 852-IV.

21. Информационная Безопасность открытых систем: учебник для вузов. В 2-х томах. Том 1 – Угрозы уязвимости, атаки и подходы к защите / С.В. Запечников, Н.Г. Милославская, А.И. Толстой, Д.В. Ушаков. – М.: Горячая линия-Телеком, 2006 – 536 с.

22. НД ТЗІ 2.5-004-99. Критерії оцінки захищеності інформації в комп'ютерних системах від несанкціонованого доступу. Затверджено наказом ДСТЗІ СБ України від 26.07. 99 р. № 22.

23. Саттон Майкл Дж.Д. Корпоративный документооборот: принципы, технологии, методология внедрения. СПб.: Издательство “Азбука”, 2002. – 448 с.

24. ISO/IEC 17799: Information technology - Code of practice for Information security management, 2000.

25. Вильям Столингс. Криптография и защита сетей: принципы и практика, 2-е издание: пер. с английского – М.: Издательский дом «Вильямс», 2001.

26. Зегжда Д.П., Ивашко А.М. Основы безопасности информационных систем. – М.: Горячая линия – Телеком, 2000. – 452 с.

27. Алферов А.П., Зубов А.Ю., Кузьмин А.С., Черемушкин А.В. Основы криптографии: Учебное пособие. – М.: Гелиос АРВ, 2001. – 480с, ил.

28. Разработка Web-приложений на PHP и MySQL: Пер. с англ./Лаура Томпсон, Люк Веллинг. – 2-е изд., испр. – СПб: ООО «ДиаСофтЮП», 2003. – 672 с.

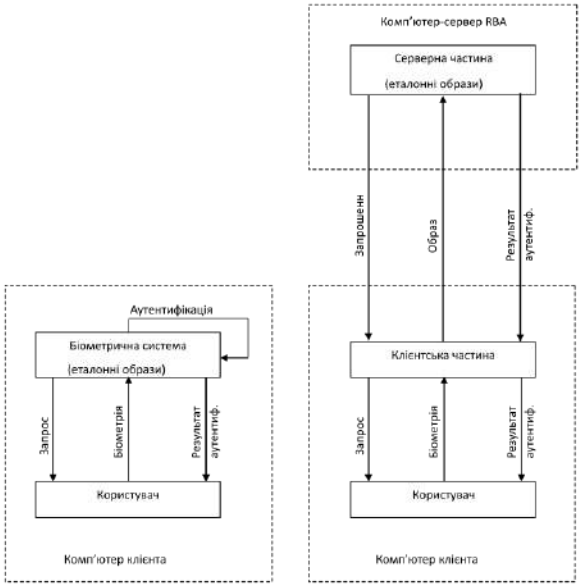
29. Александр ТАРАНОВ, Олег СЛЕПОВ. Безопасность систем электронной почты. Электронный журнал “Jet Info”, #6/2003, http://www.citforum.ru/security/internet/email/article1.6.2003.html

30. Игорь Тарасов. Аутентификация, идентификация и несанкционированный доступ, http://itsoft.ru/docs/web/c14_auth.html

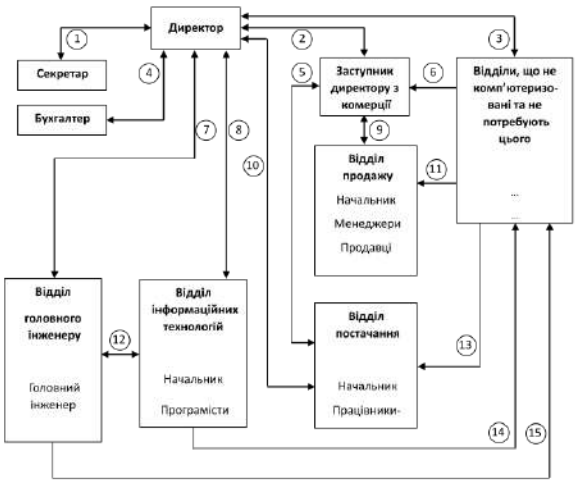
31. Роджер А. Граймз. Скрытые опасности при работе с сообщениями электронной почты и группами новостей (стаття), http://www.osp.ru/win2000/exchange/45exch10.htm

32. Атаки сети, виды и защита (стаття), http://www.dlink.ru/technology/attacks.php

33. Алексей Резниченко, Евгений Суржиков. Защита электронной почты. Журнал "Открытые системы", #07-08, 2002 год, http://www.osp.ru/os/2002/07-08/028_1.htm



Зм.	Арк.	№ докум.	Підпис	Дата	КвРКБ. 170141.17.01.02 Е8		
Розроб.		Бережуля Я.О.			Впровадження розподіленого режиму	Літера	Маса / Масштаб
Перевір.		Чекалін В.М.					
Н. Контр.						Аркуш	Аркушів
Т. Контр.						ХНУ КБ-17-1	
Затв.							



					КвРКБ. 170141.17.01.02 Е8		
Зм.	Арк.	№ докум.	Підпис	Дата			
Розроб.		Бережуля Я.О.			Інформаційна модель підприємства	Літера	Маса / Масштаб
Перевір.		Чижич В.М.					
Н. Контр.						Аркуш	Аркушів
Т. Контр.						ХНУ КБ-17-1	
Затв.							

					КвРКБ. 170141.17.01.02 Е8		
Зм.	Арк.	№ докум.	Підпис	Дата	Схема роботи програми, що взаємодіє з користувачем	Літера	Маса / Масштаб
Розроб.		Бережна Я.О.					
Перевір.		Чекан В.М.					
Н. Контр.						Аркуш	Аркушів
Т. Контр.						ХНУ КБ-17-1	
Затв.							

ДОДАТОК Б
(Обов'язковий)
Програмна реалізація

```
//---------------------------------------------------------------------
---------------

#include <winsock.h>
#include <vcl.h>
#pragma hdrstop

#include "Unit1.h"
#include "Unit2.h"
//---------------------------------------------------------------------
--------------
#pragma package(smart_init)
#pragma resource "*.dfm"

#define SRV_PORT 1234
#define CLNT_PORT 1235


String alf;
String bigrami[10];
float t[10];          //массив для общего времени всех наборов i-
той биграммы
int  numbi[10];  //массив  для  количества  раз,  сколько  была
набрана i-тая биграмма
                                //потом поделим t[i] на numbi[i] и
получим среднее время набора биграммы
int      numsym[10];//коды      букв,      образующих      биграммы
а,в,е,и,к,н,о,р,с,т
int CanAnalyze;
short int prevc,curc,NumPrinted,all; //Word
float  NumErr1,  NumErr2;  //количество  ошибок  на  единицу
набранного текста
int prevt,curt,curup;
char cursym,prevsym,cursymup;
float tlet[10];     //массив для общего времени всех наборов i-
той буквы
int  numlet[10];//массив  для  количества  раз,  сколько  была
набрана i-тая буква
float rest[10],reslet[10];

void Error(int ErrNo)
{
char mess[127];
switch(ErrNo)
     {
     case 0:
          strcpy(mess,"\nНе  можливо  ініціалізувати  систему
WinSock! Завершуємо...");
```

```
            break;
        case 1:
            strcpy(mess,"\nНе та версія WinSock! Завершуємо...");
            break;
        case 2:
            strcpy(mess,"\nНе можливо створити сокет!
Завершуємо...");
            break;
        case 3:
            strcpy(mess,"\nНе можливо зв'язати сокет!
Завершуємо...");
            break;
        case 4:
            strcpy(mess,"\nНе можливо з'єднатися з сервером!
Завершуємо...");
            break;
        case 5:
            strcpy(mess,"\Завершуємо...");
            break;
        case 6:
            strcpy(mess,"\nНе можливо розпізнати ім'я хоста!
Завершуємо...");
            break;
        case 7:
            strcpy(mess,"\nНе можливо отримати дані!
Завершуємо...");
            break;
        case 8:
            strcpy(mess,"\nНе можливо надіслати дані!
Завершуємо...");
            break;
    }
MessageBox(NULL,mess,"Повідомлення:",0);
WSACleanup();
//ExitProcess(0);
}

TForm1 *Form1;
//---------------------------------------------------------------------
--------------
__fastcall TForm1::TForm1(TComponent* Owner)
    : TForm(Owner)
{
}
//---------------------------------------------------------------------
--------------

void __fastcall TForm1::N2Click(TObject *Sender)
{
    Form1->Close();
}
```

```
//---------------------------------------------------------------------
--------------
void __fastcall TForm1::N4Click(TObject *Sender)
{
    Form2->ShowModal();
}
//---------------------------------------------------------------------
--------------
void __fastcall TForm1::FormCreate(TObject *Sender)
{
  int i;
  alf="авеилностп";
  bigrami[0]="ЛА";
  bigrami[1]="НА";
  bigrami[2]="АЛ";
  bigrami[3]="ПО";
  bigrami[4]="ТА";
  bigrami[5]="НЕ";
  bigrami[6]="ЛИ";
  bigrami[7]="ОВ";
  bigrami[8]="ТИ";
  bigrami[9]="СТ";

  numsym[0]=70;
  numsym[1]=68;
  numsym[2]=84;
  numsym[3]=66;
  numsym[4]=75;
  numsym[5]=89;
  numsym[6]=74;
  numsym[7]=67;
  numsym[8]=78;
  numsym[9]=71;
  prevc=0;
  for(i=0;i<10;i++)
    t[i]=0;
}
//---------------------------------------------------------------------
--------------
void __fastcall TForm1::Button1Click(TObject *Sender)
{
    TStringList *SL=new TStringList();
    int i;
    if(OpenDialog1->Execute())
    {
        Memo1->Clear();
        //SL=TStringList->Create();
        SL->LoadFromFile(OpenDialog1->FileName);
        for(i=0;i<SL->Count;i++)
            Memo1->Lines->Add(SL->Strings[i]);
        SL->Free();
    }
```

```
}
//---------------------------------------------------------------------
--------------
void __fastcall TForm1::Memo2KeyDown(TObject *Sender, WORD
&Key,
      TShiftState Shift)
{
  int i,j;
  String bigr;
  curt=GetTickCount();
  bigr="00";
  //if(not CanAnalyze)then exit;
  for(i=0;i<10;i++)
    if(Key==numsym[i])
    {
        cursym=alf[i+1];
        if(prevsym!='0')
        {
            bigr[1]=prevsym;
            bigr[2]=cursym;
            for(j=0;j<10;j++)
              if(AnsiCompareText(bigr,bigrami[j])==0)
              {
                  t[j]=t[j]+curt-prevt;
                  numbi[j]++;
                  break;
              }
        }
        all++;
        break;
    }
    else
      cursym='0';
  if((Key==8)&&(prevc!=8))
      NumErr1++;
  prevsym=cursym;
  prevc=Key;
  prevt=curt;
}


void __fastcall TForm1::Memo2KeyPress(TObject *Sender, char
&Key)
{
  if (Memo2->Text.Length()>100)
    Button4->Enabled=true;
  else
    Button4->Enabled=false;
}
//---------------------------------------------------------------------
--------------
```

```
void __fastcall TForm1::Button2Click(TObject *Sender)
{
  int i;
  for(i=0;i<10;i++)
  {
      t[i]=0;
      numbi[i]=0;
      tlet[i]=0;
      numlet[i]=0;
  }
  Memo1->Clear();
  Memo2->Clear();
  Edit1->Clear();
}
//---------------------------------------------------------------
--------------

void __fastcall TForm1::Memo2KeyUp(TObject *Sender, WORD &Key,
      TShiftState Shift)
{
  int i;
  curup=GetTickCount();
  for(i=0;i<10;i++)
    if(Key==numsym[i])
    {
        cursymup=alf[i+1];
        if(cursym==cursymup)
        {
            tlet[i]=tlet[i]+curup-curt;
            numlet[i]=numlet[i]+1;
        }
        break;
    }
}
//---------------------------------------------------------------
--------------

void __fastcall TForm1::Button4Click(TObject *Sender)
{


int s, min=-1,max=-1,numb,i;
struct sockaddr_in clnt_sin,srv_sin;
char local[]="127.0.0.1";
char res=-1,chlogin[128];
long double sum=0;
WSADATA wsadata;

if(WSAStartup(MAKEWORD(1,1),&wsadata)) //Version == 257
     Error(0);
if(wsadata.wVersion!=MAKEWORD(1,1))
     Error(1);
```

```
if((s=socket(AF_INET,SOCK_STREAM,0))==SOCKET_ERROR) //-1
     Error(2);
memset((char*)&clnt_sin,0,sizeof(clnt_sin));
clnt_sin.sin_family=AF_INET;
clnt_sin.sin_addr.s_addr=inet_addr(local);
clnt_sin.sin_port=htons(CLNT_PORT);
if(bind(s,(struct
sockaddr*)&clnt_sin,sizeof(clnt_sin))==SOCKET_ERROR)
     Error(3);

memset((char*)&srv_sin,0,sizeof(srv_sin));

srv_sin.sin_family=AF_INET;
srv_sin.sin_addr.s_addr=inet_addr(local);
srv_sin.sin_port=SRV_PORT;

if(connect(s,(struct                sockaddr*)&srv_sin,sizeof(struct
sockaddr))==SOCKET_ERROR)
     Error(4);

for(i=0;i<10;i++)
{
    if(numbi[i])rest[i]=t[i]/numbi[i];
    if(numlet[i])reslet[i]=tlet[i]/numlet[i];
}
strcpy(chlogin,Edit1->Text.c_str());
numb=send(s,(char*)chlogin,128,0);
if(!numb)Error(8);
numb=send(s,(char*)rest,sizeof(float)*10,0);
if(!numb)Error(8);
numb=send(s,(char*)reslet,sizeof(float)*10,0);
if(!numb)Error(8);

numb=recv(s,(char*)&res,1,0);
if(!numb)Error(7);

if(res==1)
    MessageBox(NULL,"Вітаємо,   Ви   успішно   авторизувалися   в
системі!","Вітаємо",0);

if(res==0)
    MessageBox(NULL,"Вибачте,   авторизація   була   неуспішною!
Спробуйте, будь-ласка, ще.","Помилка",0);

closesocket(s);
}
//---------------------------------------------------------------
--------------
```

РІШЕННЯ ЕКСПЕРНОЇ КОМІСІЇ

КАФЕДРИ КІБЕРБЕЗПЕКИ ТА КОМП'ЮТЕРНИХ СИСТЕМ І МЕРЕЖ

ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: Система захисту інформаційно-комунікаційної мережі ДП "Новатор" Державного концерну "Укроборонпром", м.Хмельницький

Автор: Беркута Ярослав Олександрович

Спеціальність: 125 – Кібербезпека

Освітня програма: Кібербезпека

Керівник: Чешун Віктор Миколайович, к.т.н., доцент

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом (далі – зазначаються підстави віднесення запозичень до правомірних). Робота приймається до захисту.	відповідає
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи (далі – зазначаються детальні та аргументовані підстави віднесення запозичень до правомірних). Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи.	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	
5	Інше:	

Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

1) запозичення розміщені в розділах аналізу існуючих аналогів та прототипів, які не описують безпосередньо авторське дослідження і не стосуються результатів роботи;
2) усі запозичення фрагментарні, або мають належним чином оформленні посилання;
3) всі зафіксовані системою ознаки модифікації тексту відносяться до комбінування латинських символів зі україномовними скороченнями індексів в формулах, що не є модифікацією тексту.

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ ідентичності/схожості, складає 5.5% що, з урахуванням наведених обґрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.

Керівник роботи _____ В.М. Чешун

Завідувач кафедри КбКСМ, гарант ОП _____ Ю.П. Кльоц

Дата: 07.06.2021

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

освітнього ступеня «бакалавр»

Студент Беркута Ярослав Олександрович

Тема Система захисту інформаційно-комунікаційної мережі ДП "Новатор" Державного концерну "Укроборомпром", м.Хмельницький

Спеціальність 125 – Кібербезпека

Обсяг кваліфікаційної роботи освітньо-кваліфікаційного рівня «магістр»:

кількість листів креслень 3 ; кількість сторінок записки 70

1. Короткий зміст роботи та прийнятих рішень У кваліфікаційній роботі розроблено систему для захисту від витоку даних .

2. Висновок про відповідність кваліфікаційної роботи завданню Кваліфікаційна робота у повній мірі відповідає поставленому завданню як в теоретичній, так і в практичній частині роботи

3. Характеристика виконання кожного розділу роботи, ступінь використання останніх досягнень науки і техніки і передових методів роботи: У вступі подана загальна характеристика поставленої задачі, чітко визначено об'єкт, предмет та методи дослідження, сформульована актуальність. Визначені задачі, які необхідно вирішити для досягнення поставленої мети, практична цінність отриманих результатів, їхня новизна та наведені відомості про публікації. У першому розділі проведено огляд використовуваних в комп'ютерних системах методів захисту конфіденційної інформації, виконане обгрунтування актуальності теми дослідження і виконана постановка задачі. В другому розділі наведені засоби та технології використані для побудови системи захисту. В третьому розділі визначено основні положення системи та розроблено алгоритми її роботи.

4. Позитивні сторони роботи Кваліфікаційна робота має комплексну практичну цінність. Практична цінність полягає у розробці системи захисту інформаційно-комунікаційної системи. На основі даного аналізу в подальшому здійснюється керуючий вплив на витік конфіденційних даних. Практична цінність результатів кваліфікаційної роботи полягає у створені системи захисту від витоків даних, що може бути підлаштована для будь якої категорії даних, і у описі оптимальних моделей функціонування систем захисту подібного характеру.

5. Негативні сторони роботи Розроблена система захисту від витоків даних досить чутлива до навантаження. В роботі мало уваги надається технічній реалізації системи (програмній, програмно-апаратній).

6. Оцінка графічного оформлення та пояснювальної записки роботи Графічне оформлення виконане відповідно до теми кваліфікаційної роботи з дотриманням стандартів. В загальному графічне оформлення виконане якісно, пояснювальна записка відповідає нормам щодо її оформлення.

7. Відгук про роботу в цілому В загальному кваліфікаційна робота заслуговує позитивної оцінки. Весь матеріал кваліфікаційної роботи структурований, чіткий та послідовний. Усі розділи роботи послідовні та логічні, що дозволяє чітко розуміти викладений матеріал в рамках тематики кваліфікаційної роботи. Графічний матеріал дозволяє наочно побачити доцільність та ефективність рішень, які були прийняті за основу для досягнення поставленої мети.

8. Інші зауваження Окремі описи в пояснювальній записці подано занадто деталізовано, що ускладнює сприйняття матеріалу в обраній предметній галузі.

9. Оцінка кваліфікаційної роботи Враховуючи всі позитивні та негативні сторони представленої кваліфікаційної роботи, можна зробити висновок, що вона заслуговує оцінку «добре».

РЕЦЕНЗЕНТ (прізвище, ім'я, по батькові, посада, місце роботи) Лисенко С.Н

д.т.н доцент КІПН

« 07 » 06 2021. (підпис)

UNICHECK

User name:
Кафедра кибербезпеки

Check ID:
1008351682

Check date:
23.06.2021 12:10:43 EEST

Check type:
Doc vs Internet

Report date:
23.06.2021 12:13:07 EEST

User ID:
100005590

File name: **Кваліфікаційна робота Беркута**

Page count: **67** Word count: **13466** Character count: **100579** File size: **1.66 MB** File ID: **1008421414**

5.48% Matches

Highest match: **1.14%** with Internet source (**https://nauka-online.com/ua/publications/informatsionnye-tehnologii/2019/10/issled**

5.48% Internet sources **481** ..... Page 69

No Library search was conducted

0% Quotes

Exclusion of quotes is off

Exclusion of references is off

0% Exclusions

No exclusions

Modifind

Text modifications detected. Find more details in the online report.

Replaced characters **11**

Anti-Plagiarism v-15.257

Максимальное совпадение с одним документом 9.0%

Словари проверки: en_US, ru_RU, ua_UA. **Ошибок в документах: 7%**

	Документ		Суммарное совпадение по Базе Данных	
ID: 95288 Название: Система захисту інформаційно-комунікаційної мережі ДП”Новатор” Державного концерну “Укроборомпром”, м.Хмельницький Добавлено в БД: 2021-06-23 Авторы: Беркута Я.О. Руководители: Чешун В.М. Консультанты: Опоненты:	Символы	Лексемы	Символы	Лексемы
	85510	594	11517 (13%)	99 (17%)

Источник плагиата

ID	Описание	Наличие плагиата в документе	
		Символы	Лексемы