НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ОДЕСЬКА ЮРИДИЧНА АКАДЕМІЯ»
ФАКУЛЬТЕТ КІБЕРБЕЗПЕКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра кібербезпеки

КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«МЕТОДИ ЗАХИСТУ ІНФОРМАЦІЇ КОРИСТУВАЧІВ У ЦИФРОВОМУ ПРОСТОРІ»

Виконав: здобувач 4 курсу
Рівень бакалавр
Галузь знань 12 «Інформаційні технології»,
спеціальність 125 «Кібербезпека»
Мельников Костянтин Віталійович

Керівник: к.ф.-м.н., доцент
Чепурна Олена Євгенівна

Рецензент: к.ф.-м.н., доцент
Ахмаметьева Ганна Валеріївна

Одеса – 2025

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ОДЕСЬКА ЮРИДИЧНА АКАДЕМІЯ»

Факультет кібербезпеки та інформаційних технологій
Кафедра кібербезпеки
Галузь знань: 12 Інформаційні технології
Спеціальність: 125 Кібербезпека
Назва освітньої програми: Кібербезпека

ЗАТВЕРДЖУЮ
Завідувач кафедри кібербезпеки
доцент С.А. Горбаченко
_____ «____»_____ 20__ року

ЗАВДАННЯ
на кваліфікаційну (бакалаврську) роботу
здобувачу Мельникова Костянтина Віталійовича

1. Тема роботи: «Методи захисту інформації користувачів у цифровому просторі»
Керівник роботи: к.ф-м.н, доцент Чепурна Олена Євгенівна
затверджені наказом НУ ОЮА від 18 березня 2025 року № 548-6
2. Строк подання здобувачем роботи 19 травня 2025 року
3. Дата видачі завдання 16 вересня 2025 року

КАЛЕНДАРНИЙ ПЛАН

з/п	Назва етапів бакалаврської роботи	Строк виконання етапів роботи	Примітка

Здобувач_____

(підпис) (прізвище та ініціали)

Керівник роботи _____

(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Кваліфікаційна робота на тему: «Методи захисту інформації користувачів у цифровому просторі» на здобуття першого (бакалаврського) рівня вищої освіти за спеціальністю 125 «Кібербезпека», освітньою програмою: Кібербезпека, містить 6 рисунків, 3 таблиці, 28 використаних джерела за переліком посилань. Робота виконана на 61 сторінках загального тексту, з яких 53 сторінок становить основний текст.

Метою роботи є комплексне дослідження та аналіз сучасних методів захисту інформації користувачів у цифровому просторі, з акцентом на виявлення ефективних підходів до автентифікації, авторизації, шифрування та хешування, а також розробка комбінованої моделі безпеки, яка забезпечує надійний захист персональних даних в умовах зростаючих кіберзагроз і цифровізації суспільства.

Результати роботи можуть бути використані при розробці та вдосконаленні інформаційно-телекомунікаційних систем, що потребують підвищеного рівня захисту користувацьких даних, зокрема в сферах електронного урядування, інтернет-банкінгу, медичних інформаційних платформ, освітніх сервісів, а також у корпоративних середовищах для побудови внутрішніх систем кібербезпеки.

Можливими напрямками подальших досліджень є удосконалення алгоритмів багатофакторної автентифікації з використанням біометричних параметрів, дослідження стійкості криптографічних методів до атак із застосуванням квантових обчислень, а також розробка адаптивних систем виявлення аномалій у поведінці користувачів для своєчасного реагування на потенційні загрози.

ЗАХИСТ ІНФОРМАЦІЇ, ЦИФРОВИЙ ПРОСТІР, КІБЕРБЕЗПЕКА, ПЕРСОНАЛЬНІ ДАНІ.

ABSTRACT

The qualification thesis on the topic: "Methods of User Information Protection in the Digital Space", submitted for the attainment of the first (bachelor's) level of higher education in the specialty 125 Cybersecurity, educational program: Cybersecurity, includes 6 figures, 3 tables, and 28 sources listed in the bibliography. The work comprises 61 pages of total text, with 53 pages forming the main body of the research.

The purpose of the thesis is a comprehensive study and analysis of modern methods of protecting user information in the digital space, with a focus on identifying effective approaches to authentication, authorization, encryption, and hashing, as well as the development of a combined security model that ensures reliable protection of personal data under conditions of increasing cyber threats and the digitalization of society.

The results of the thesis can be applied in the development and enhancement of information and telecommunication systems that require an increased level of user data protection – particularly in the fields of e-government, internet banking, medical information platforms, educational services, as well as in corporate environments for building internal cybersecurity systems.

Possible directions for future research include the improvement of multi-factor authentication algorithms using biometric parameters, investigation of the resistance of cryptographic methods to attacks involving quantum computing, and the development of adaptive anomaly detection systems in user behavior to ensure timely response to potential threats.

INFORMATION PROTECTION, DIGITAL SPACE, CYBERSECURITY, PERSONAL DATA.

ЗМІСТ

ВСТУП

У XXI столітті цифрові технології стали невід'ємною частиною повсякденного життя людини. Стрімкий розвиток інтернету, мобільних пристроїв, хмарних обчислень та соціальних мереж призвів до того, що більшість особистої, професійної та конфіденційної інформації зберігається, обробляється і передається в електронному вигляді. У зв'язку з цим питання захисту даних користувачів набуває особливої актуальності та складності.

Цифровий простір – це інтегрована інфраструктура, що включає інформаційні технології, програмне забезпечення, мережі зв'язку, користувацькі пристрої та дані, які циркулюють між ними. Однак, з розширенням можливостей цифрового середовища зростає і спектр загроз, які супроводжують користувача. До найбільш поширених належать крадіжка персональних даних, фішинг, розповсюдження шкідливого програмного забезпечення, вторгнення до хмарних облікових записів, соціальна інженерія тощо.

Одночасно з розвитком цифрових технологій постає питання про ефективні методи захисту користувача від кіберзагроз. Застосування виключно технічних рішень (антивірусів, фаєрволів, VPN, шифрування тощо) вже не гарантує повної безпеки. Надзвичайно важливими є організаційні та поведінкові підходи – від налаштувань приватності до формування кібергігієни й цифрової культури користувачів.

Метою даної кваліфікаційної роботи є дослідження та систематизація сучасних методів захисту інформації користувачів у цифровому просторі, аналіз їх ефективності та практичне моделювання застосування захисних стратегій у реальних умовах. Особлива увага приділяється саме комплексному підходу до безпеки – технічному, організаційному та поведінковому.

Для досягнення поставленої мети в роботі були сформульовані такі основні завдання:

- проаналізувати сучасні загрози в цифровому середовищі;

- класифікувати та охарактеризувати існуючі методи захисту інформації користувачів;
- порівняти ефективність популярних інструментів та засобів безпеки;
- дослідити вплив цифрової гігієни на безпеку користувача;
- розробити практичні рекомендації щодо забезпечення особистої інформаційної безпеки в повсякденному користуванні цифровими сервісами.

Об’єктом дослідження є процес забезпечення безпеки користувацької інформації в цифровому середовищі.

Предметом дослідження є методи, засоби та практичні підходи до захисту персональних даних та інформаційної активності користувача у взаємодії з цифровими технологіями.

Актуальність дослідження обумовлена зростаючою кількістю кіберзлочинів, зниженням рівня довіри до цифрових сервісів, потребою в підвищенні обізнаності користувачів про ризики та необхідність формування нової культури безпечної поведінки в цифровому просторі.

Результати дослідження можуть бути використані для підвищення ефективності освітніх програм з кібергігієни, удосконалення політик захисту інформації в організаціях, а також слугувати базою для подальших наукових розвідок у сфері інформаційної безпеки.

1 АНАЛІЗ СУЧАСНИХ МЕТОДІВ ЗАХИСТУ ІНФОРМАЦІЇ

1.1 Основні загрози безпеці: фішинг, перехоплення трафіку (MITM), шкідливе ПЗ

У сучасному цифровому світі інформація стала одним із найцінніших ресурсів. Від неї залежать не лише особисте життя користувача, а й стабільність державних структур, фінансових систем, підприємств і соціальних інституцій. З розвитком інформаційних технологій та глобальної мережі Інтернет паралельно зростає і кількість загроз, що ставлять під сумнів конфіденційність, цілісність і доступність цифрових даних. Основні загрози безпеці в цифровому просторі мають різноманітну природу, вони постійно еволюціонують, адаптуючись до нових технологій та поведінки користувачів.

Фішинг є однією з найпоширеніших форм соціальної інженерії в цифровому просторі, що становить серйозну загрозу для інформаційної безпеки користувачів. Цей вид атаки базується не стільки на технічних вразливостях системи, скільки на психологічній вразливості людини. Головна мета фішингу – змусити жертву добровільно надати конфіденційну інформацію: облікові дані, фінансову інформацію, особисті документи, коди доступу тощо.

Термін "фішинг" (англ. phishing) походить від слова fishing (рибальство) і символізує «вудіння» жертв зловмисниками. Класичний фішингатакант створює повідомлення або вебсайт, який імітує справжній ресурс – наприклад, інтернетбанкінг, державний портал, соціальну мережу або службу технічної підтримки. У повідомленні користувача закликають до термінової дії: оновити пароль, підтвердити особу, отримати приз чи уникнути блокування акаунта. Зазвичай у таких повідомленнях застосовується мова тривоги, загроз або обіцянки вигоди – усе це для того, щоб спонукати людину діяти необдумано.

Фішингові атаки поділяються на кілька типів залежно від цільової аудиторії та способу реалізації:

– Класичний фішинг масове розсилання підроблених листів великій кількості користувачів без конкретного таргетування. Часто в таких листах є посилання на підроблені вебсайти, схожі за дизайном і URLадресою на справжні.

– Spear phishing (цільовий фішинг) більш персоналізований вид атаки, спрямований на конкретну особу або організацію. У цьому випадку зловмисник попередньо збирає інформацію про жертву (наприклад, через соціальні мережі) й формує повідомлення, яке максимально відповідає її контексту.

– Whaling фішинг, націлений на керівників компаній або осіб, які мають доступ до критично важливої інформації. Атаки часто маскуються під офіційне ділове листування, з використанням термінології, характерної для конкретної сфери.

– Smishing та vishing фішинг через SMS (smishing) або голосові дзвінки (vishing). Жертві надходить повідомлення або дзвінок від «служби безпеки банку», «державного органу» чи навіть «слідчого», з вимогою верифікувати дані.

Поширення фішингу зумовлене декількома факторами. Поперше, це відносно простий і дешевий метод – для його реалізації не потрібно мати глибокі технічні знання. Подруге, фішинг часто є ефективним, оскільки базується на природній довірі людини до авторитетів, офіційних структур або знайомих. Потретє, користувачі часто не володіють достатнім рівнем цифрової обізнаності, що робить їх легкою мішенню.

Фішинг не лише шкодить окремому користувачеві, але й створює загрозу для безпеки організацій. Втрата корпоративних облікових даних може призвести до витоку конфіденційної інформації, фінансових збитків, зупинки роботи бізнеспроцесів і шкоди репутації компанії.

Захист від фішингу потребує комплексного підходу. Найважливішими є:

– Освітні заходи навчання користувачів основам цифрової гігієни, навичкам виявлення фальшивих повідомлень і сайтів;

– Технічні засоби фільтри електронної пошти, антивірусне ПЗ, розширення для браузерів, що попереджають про підозрілі сайти;

– Багатофакторна аутентифікація навіть якщо зловмисник отримає пароль, йому буде складно увійти в акаунт без додаткового коду;

– Розсудливість користувача не відкривати посилання з незнайомих джерел, перевіряти URLадреси сайтів, не вводити паролі на сумнівних сторінках.

У цифровому середовищі, де основна маса даних передається мережею, важливою загрозою для конфіденційності та цілісності інформації є атаки типу MITM (ManintheMiddle) – в перекладі з англійської «людина посередині». Суть такої атаки полягає в тому, що зловмисник втручається в канал зв'язку між двома сторонами – наприклад, між користувачем і вебсайтом – і може непомітно читати, змінювати або підміняти передану інформацію.

MITMатаки не потребують порушення безпеки кінцевих пристроїв – досить контролювати трафік або точку доступу. Це робить їх особливо небезпечними, оскільки навіть при використанні на перший погляд безпечного підключення (наприклад, WiFi) користувач може стати жертвою несанкціонованого перехоплення.

Зловмисник може реалізувати MITMатаку кількома способами. Найпоширенішим є підміна мережі WiFi: користувач підключається до відкритої або неправильно ідентифікованої мережі, яка насправді належить атакуючому. Зловмисник при цьому виконує роль «проміжного вузла» – всі дані, що передаються від користувача до інтернету, проходять через його пристрій.

Іншим поширеним методом є ARPspoofing (підміна адрес ARP) – техніка, що дозволяє зловмиснику змінити таблиці маршрутизації в локальній мережі, щоб комп'ютери надсилали трафік саме йому. Після цього трафік може бути або збережений для подальшого аналізу, або підмінений «на льоту».

Також існує DNSspoofing, коли користувач вводить у браузері адресу легітимного сайту, але через змінені відповіді DNSсервера потрапляє на підроблену сторінку. Навіть якщо така сторінка виглядає як справжня, вона може бути призначена для збору логінів, паролів, фінансових даних тощо.

Інша поширена форма атаки – SSLstripping, яка знижує рівень шифрування з HTTPS до звичайного HTTP. У цьому випадку користувач може думати, що перебуває на захищеному сайті, хоча насправді вся його активність доступна для зловмисника в незашифрованому вигляді.

Основною метою MITMатак є перехоплення конфіденційної інформації: паролів, платіжних даних, особистого листування, облікових даних. Вони також можуть використовуватись для шпигунства, збору даних про поведінку користувача, викрадення особистості або підготовки до масштабніших атак (наприклад, через соціальну інженерію).

Наслідки таких атак можуть бути серйозними:

- фінансові втрати (через доступ до банківських акаунтів);
- компрометація корпоративної інформації;
- викрадення або підміна особистих даних;
- поширення шкідливого ПЗ;
- втручання у листування або зміну переданої інформації.

Особливо вразливими до MITM є користувачі відкритих або публічних WiFiмереж, а також ті, хто не перевіряє безпеку з’єднання або не використовує додаткові засоби захисту.

Найефективнішим засобом захисту від MITMатак є використання шифрування:

- HTTPS (TLS) усі сучасні вебсайти повинні мати сертифікати безпеки (SSL/TLS). Користувач повинен звертати увагу на наявність замка в адресному рядку та уникати сайтів без захищеного з’єднання.
- VPN (Virtual Private Network) шифрує весь трафік між користувачем і сервером, навіть у відкритих мережах. Це значно ускладнює або унеможливлює втручання MITMатакуючого.
- Двофакторна аутентифікація (2FA) навіть якщо зловмисник отримає облікові дані, він не зможе отримати доступ до облікового запису без другого фактору – наприклад, одноразового коду на телефоні.

– Використання DNSSEC та захищених DNScерверів – зменшує ймовірність DNSпідміни.

– Необхідність оновлення програмного забезпечення багато вразливостей, які використовуються для MITM, пов’язані з застарілими версіями браузерів, ОС або протоколів зв’язку.

– Освіченість користувачів уникнення підключення до невідомих WiFiмереж, перевірка сертифікатів сайтів, уважне ставлення до повідомлень браузера про проблеми з безпекою.

Атаки типу «людина посередині» є однією з найскладніших для виявлення форм кіберзагроз. Вони використовують не лише технічні прогалини, але й легковажність або необізнаність користувача. Надійний захист від MITMатак можливий лише за умови поєднання технічних рішень і свідомого користувацького підходу до цифрової безпеки. Ураховуючи постійний розвиток технологій та збільшення обсягу особистої інформації, яку ми передаємо в мережі, проблема MITMатак не втрачає актуальності й потребує постійної уваги з боку як фахівців, так і звичайних користувачів.

Таблиця 1.1 – Способи реалізації MITMатак та засоби захисту

Спосіб MITMатаки	Опис методу атаки	Засіб захисту
Підміна WiFi (Fake AP)	Створення фальшивої точки доступу WiFi, до якої підключається жертва	Використання VPN; уникнення відкритих мереж; перевірка SSID; вимкнення автопідключення до мереж
ARPspoofing	Маніпуляція адресами ARP для перенаправлення трафіку через пристрій зловмисника	Вмикання фільтрації ARP; використання статичних ARPзаписів; захищені мережеві середовища

Продовження таблиці 1.1

DNSspoofing	Видача фальшивої IPадреси у відповідь на DNSзапит користувача	Використання DNSSEC; зміна DNScерверів на безпечні (Google DNS, Cloudflare); використання HTTPS
SSLstripping	Перетворення захищеного HTTPSз'єднання на незахищене HTTP	Перевірка URL на наявність HTTPS; браузерні розширення (наприклад, HTTPS Everywhere); HSTS
Проксіперехоплення	Перехоплення трафіку через шкідливий або підконтрольний проксісервер	VPN або TOR; використання лише перевірених проксісерверів; налаштування браузера на відмову від HTTPз'єднань
Перехоплення незашифрованого трафіку	Аналіз переданої інформації у відкритому вигляді	Використання протоколів з шифруванням (HTTPS, FTPS, SFTP); уникнення публікації конфіденційних даних

Шкідливе програмне забезпечення, або malware (від англ. malicious software), є однією з найбільш серйозних та постійно еволюціонуючих загроз у сфері інформаційної безпеки. Воно охоплює широкий спектр програм, призначених для несанкціонованого доступу до даних, порушення функціонування систем,

викрадення інформації, шпигунства або нанесення шкоди пристроям, мережам і користувачам.

Malware може діяти як самостійно, так і бути частиною комплексної атаки, поєднуючи технічні та соціотехнічні методи впливу. Середовище цифрової взаємодії, де відбувається передача, зберігання та обробка великого обсягу конфіденційної інформації, створює сприятливі умови для його поширення.

Існує багато різновидів шкідливого програмного забезпечення, кожен з яких виконує специфічні функції:

- Віруси один з найдавніших типів malware. Вони прикріплюються до легітимних файлів і розмножуються при відкритті або запуску цих файлів. Метою вірусу може бути пошкодження даних, перевантаження системи або знищення інформації.
- Черв'яки (worms) саморозповсюджувальні програми, які проникають у комп'ютерну мережу, не потребуючи участі користувача. Вони можуть спричинити перевантаження мережі, витік інформації або стати «транспортом» для інших шкідливих компонентів.
- Троянські програми (трояни) маскуються під корисне або законне програмне забезпечення, однак після інсталяції відкривають доступ до системи зловмиснику. Трояни можуть викрадати паролі, контролювати комп'ютер, встановлювати інше malware.
- Spyware (шпигунське ПЗ) таємно збирає інформацію про дії користувача: відвідувані сайти, натискання клавіш (кейлогери), вміст екрану, логіни та паролі. Spyware зазвичай працює у фоновому режимі, що ускладнює його виявлення.
- Adware (рекламне ПЗ) показує нав'язливу рекламу або перенаправляє користувача на підозрілі сайти. Хоча adware не завжди є небезпечним у класичному розумінні, воно може значно знижувати продуктивність системи та виступати «воротами» для складнішого malware.
- Ransomware (програмивимагачі) шифрують файли на пристрої жертви та вимагають викуп за їх розшифрування. Це одна з найнебезпечніших форм malware,

що спричиняє серйозні фінансові та іміджеві втрати як для приватних осіб, так і для компаній. Відомі випадки, коли ці атаки паралізували роботу цілих лікарень, муніципалітетів чи інфраструктурних об’єктів.

- Rootkits і Backdoors дають зловмиснику повний контроль над системою, приховуючи свою присутність. Вони дозволяють обходити традиційні методи захисту та можуть залишатися активними в системі протягом тривалого часу.
- Botnets групи пристроїв, інфікованих шкідливим ПЗ і об’єднаних у мережу, яку контролює зловмисник. Такі мережі використовуються для здійснення DDoSатак, масового спамрозсилання, шахрайства з кліками або майнінгу криптовалют.

Malware може проникати до пристрою користувача різними шляхами:

- через вкладення в електронних листах (часто маскуються під рахунки, резюме або документи Microsoft Office із макросами);
- при завантаженні нелегального програмного забезпечення, ігор або медіаконтенту;
- через фішингові вебсайти;
- за допомогою інфікованих флешок чи зовнішніх дисків;
- внаслідок використання незахищених або застарілих протоколів;
- через автоматичні оновлення програм з недостовірних джерел.

В останні роки спостерігається зростання так званих supply chain атак – коли зловмисники вбудовують шкідливий код у легітимні оновлення ПЗ або бібліотеки, що поширюються через офіційні канали.

Наслідки можуть бути як миттєвими (втрата доступу до файлів, порушення роботи системи), так і довготривалими (витік персональних даних, репутаційні та фінансові збитки). Часто malware створює "вразливе місце" для наступних атак, використовуючи інфікований пристрій як точку входу для розширення впливу в мережі.

Для протидії malware необхідно застосовувати багаторівневий підхід, що включає:

- Антивірусне програмне забезпечення з актуальними базами сигнатур;
- Фаєрволи, які обмежують небажані з'єднання;
- Системи виявлення вторгнень (IDS) і запобігання вторгненням (IPS);
- Оновлення операційної системи та програмного забезпечення, щоб закрити відомі вразливості;
- Використання лише перевірених джерел для завантаження ПЗ;
- Використання віртуальних машин для тестування підозрілих файлів;
- Резервне копіювання даних, що дозволяє уникнути втрати інформації при атаці ransomware;
- Обізнаність користувачів найважливіша складова. Саме людська неуважність або необережність найчастіше стають «точкою входу» для malware.

Malware залишається однією з найнебезпечніших форм загроз у цифровому середовищі. Високий рівень автоматизації атак, інтелектуальне маскування, використання нових каналів розповсюдження робить шкідливе ПЗ дедалі важчим для виявлення. Ефективна боротьба з ним потребує не лише використання сучасних технологічних засобів, але й формування в користувачів стійких навичок інформаційної гігієни, критичного мислення та усвідомлення ризиків цифрового середовища.

Дуже серйозною і водночас недооціненою загрозою є людський фактор, що проявляється в нехтуванні основними правилами цифрової гігієни. Сюди належить використання слабких або однакових паролів для різних сервісів, ігнорування двофакторної аутентифікації, відкриття підозрілих листів або посилань, недбале ставлення до оновлень програмного забезпечення. Через неуважність або низький рівень обізнаності користувачі самостійно створюють «лазівки» для зловмисників, навіть не усвідомлюючи цього.

Небезпеку також становлять уразливості самих програм або сервісів. Навіть найкраще захищені системи можуть містити програмні помилки або недопрацювання, які можуть бути використані хакерами для проникнення. Такі атаки часто називаються zeroday – це уразливості, про які розробники ще не знають

або не встигли випустити оновлення безпеки. Уразливості можуть стосуватися як операційних систем, так і браузерів, мобільних додатків, драйверів, мережевого обладнання тощо.

Загрози цифрового простору мають ще одну важливу рису – вони стають дедалі менш помітними та більш персоналізованими. Завдяки аналітиці великих даних та штучному інтелекту зловмисники можуть створювати цілеспрямовані атаки на конкретного користувача або організацію, враховуючи його поведінку, переваги, соціальні зв’язки. Це підвищує ефективність атак і ускладнює їхнє виявлення традиційними методами.

1.2 Класифікація методів: автентифікація, авторизація, шифрування, хешування

У сучасному цифровому середовищі, яке характеризується високим рівнем взаємодії користувачів із інформаційними системами, питання безпеки стає стратегічно важливим. Щоб забезпечити надійний захист даних у цифровому просторі, використовуються спеціальні методи та механізми, що мають різну функціональну спрямованість. Загалом ці методи поділяються на процедурні, технічні та криптографічні, кожен із яких виконує власну роль у загальній системі інформаційної безпеки.

Основними цілями застосування методів захисту є:

- забезпечення конфіденційності тобто унеможливлення доступу до даних сторонніми особами;
- цілісність інформації гарантування того, що дані не були змінені або знищені ненавмисно чи навмисно;
- доступність забезпечення постійного, надійного доступу уповноважених осіб до ресурсів;
- автентичність підтвердження того, що інформація або доступ належать саме тій особі чи об'єкту, за яких вони себе видають;

– невідмовність унеможливлення заперечення факту здійснення певної дії користувачем.

Відповідно до функціонального призначення, методи захисту інформації можна класифікувати наступним чином:

1. Методи ідентифікації, автентифікації та авторизації – використовуються для розпізнавання суб’єкта доступу та перевірки його прав. Вони дозволяють гарантувати, що до інформації мають доступ лише уповноважені особи.

2. Криптографічні методи – забезпечують збереження конфіденційності й цілісності даних у процесі їх передавання або зберігання. До них належать шифрування, хешування, цифровий підпис та інші механізми.

3. Методи контролю доступу – реалізують політику безпеки, яка обмежує або дозволяє виконання певних дій у системі.

4. Методи моніторингу та виявлення вторгнень – слідкують за активністю в системі та виявляють аномальну або потенційно небезпечну поведінку.

5. Методи резервного копіювання та відновлення – дозволяють зберегти дані в разі збоїв або шкідливого впливу, відновивши систему до стабільного стану.

У межах цього підрозділу акцент буде зроблено на поглибленому аналізі основоположних методів забезпечення інформаційної безпеки, серед яких особливо важливими є автентифікація, авторизація, шифрування та хешування. Ці методи становлять собою базис для формування ефективної та стійкої до зовнішніх і внутрішніх загроз інфраструктури цифрової безпеки, що є актуальною вимогою для будьякої сучасної інформаційної системи. Незалежно від галузі застосування – чи то сфера електронної комерції, інтернетбанкінгу, охорони здоров’я, державного управління або соціальних медіа – використання зазначених механізмів є не лише

доцільним, а й критично необхідним для збереження конфіденційності, цілісності та доступності даних користувачів.

Розгляд цих методів дозволить не лише виявити їхню функціональну роль у контексті загальної моделі захисту інформації, але й з'ясувати особливості їхньої реалізації у сучасних цифрових системах. У процесі аналізу буде охоплено як теоретичні засади роботи кожного з підходів, так і їхні переваги, недоліки, типові вразливості та технічні обмеження. Окрема увага буде приділена взаємозв'язку між цими методами, а також способам їх інтеграції у єдину захисну модель, що дозволяє досягти максимальної ефективності у протидії кіберзагрозам. Такий підхід сприятиме більш глибокому розумінню архітектури безпеки цифрових систем та забезпечить теоретичне підґрунтя для розробки власних або удосконалення наявних рішень у сфері кіберзахисту.

Таблиця 1.2 – Класифікація основних методів захисту інформації

Метод	Призначення	Основні приклади / підходи
Автентифікація	Підтвердження особи користувача перед наданням доступу	Паролі, біометрія, OTPкоди, 2FA (двофакторна автентифікація)
Авторизація	Надання або обмеження доступу до ресурсів згідно з правами	RBAC (ролі), ABAC (атрибути), токени доступу, дозволи ОС
Шифрування	Захист конфіденційності даних під час зберігання та передавання	AES, RSA, TLS/SSL, VPN, PGP
Хешування	Перевірка цілісності даних, захист паролів, створення цифрового «відбитка»	SHA256, SHA3, хеші паролів, цифрові підписи, перевірка цілісності файлів

Автентифікація (від англ. authentication) – це процес перевірки справжності особи або об'єкта, який намагається отримати доступ до інформаційної системи,

ресурсу чи послуги. Інакше кажучи, автентифікація – це відповідь на запитання: «Хто ви?»

Автентифікація є одним із ключових елементів безпеки в цифровому просторі, оскільки саме вона дозволяє системі ідентифікувати користувача та переконатися, що він є тим, за кого себе видає. Без надійної автентифікації неможливо забезпечити ні конфіденційність, ні захист від несанкціонованого доступу, ні правову відповідальність за дії в системі.

Процес автентифікації базується на використанні одного або кількох із трьох типів автентифікаційної інформації:

1. Що користувач знає (knowledgebased) – наприклад, пароль, PINкод, відповідь на секретне запитання.
2. Що користувач має (possessionbased) – смарткарта, токен, мобільний пристрій, одноразовий код.
3. Ким користувач є (inherencebased) – біометричні характеристики: відбиток пальця, райдужка ока, розпізнавання обличчя, голос тощо.

Найбільш надійною є автентифікація, яка використовує два або більше факторів (двофакторна або багатофакторна автентифікація). Наприклад, комбінація «пароль + код із SMS» або «відбиток пальця + PIN».

Види автентифікації:

1. Однофакторна автентифікація

Найпростіший і найпоширеніший тип. Зазвичай реалізується за допомогою логіна й пароля. Хоча така схема зручна, вона є найменш безпечною через вразливість до атак типу bruteforce, фішинг, keylogging, та витоків баз даних із паролями.

2. Двофакторна автентифікація (2FA)

Поєднує два різні способи підтвердження особи, наприклад, пароль + код підтвердження на телефон. Це значно підвищує рівень безпеки, оскільки навіть при компрометації пароля зловмисник не зможе отримати доступ без другого фактора.

3. Багатофакторна автентифікація (MFA)

Використання трьох і більше факторів. Наприклад: пароль + токен + біометрія. Такий підхід найчастіше застосовується в критичних системах (державні структури, військові, фінансові установи).

4. Біометрична автентифікація

Підтвердження особи за допомогою фізіологічних або поведінкових характеристик користувача. Це можуть бути відбитки пальців, геометрія обличчя, голос, ритм набору тексту, динаміка миші. Перевага – складність підробки. Недоліки – можливість помилок (false positive/false negative) і складність змін у разі витоку.

5. Одноразові паролі (OTP, TOTP)

Генеруються щоразу нові коди, які дійсні лише один раз або протягом обмеженого часу (наприклад, 30 секунд). OTP можуть надсилатися через SMS, email або генеруватися мобільним застосунком (Google Authenticator, Authy).

Попри важливість автентифікації, вона також є вразливою ланкою, зокрема:

- Фішинг коли користувач сам вводить свої облікові дані на підробленому сайті.
- Keylogging перехоплення введення з клавіатури, часто через шкідливе ПЗ.
- Bruteforce підбір паролів методом перебору.
- Credential stuffing використання вкрадених облікових даних, які були злиті на інших сайтах.
- Соціальна інженерія маніпуляції, щоб змусити людину розкрити логін/пароль.

Ці загрози вимагають застосування комбінованих методів захисту, моніторингу поведінки користувачів, шифрування каналів передачі даних і навчання персоналу.

Автентифікація є першим і базовим етапом контролю доступу. Від її надійності залежить загальний рівень захищеності системи. У сучасному інформаційному середовищі зростає популярність концепцій безпарольної автентифікації (passwordless authentication) – наприклад, вхід за допомогою

біометрії або мобільного застосунку. Це дозволяє уникнути основних вразливостей, пов’язаних із паролями.

Крім того, автентифікація лежить в основі цифрової ідентичності, яка все частіше використовується в електронному врядуванні, банківських системах, медичних платформах, тощо.

Автентифікація є фундаментальним компонентом системи інформаційної безпеки, що забезпечує перевірку правомірності доступу до цифрових ресурсів. У сучасних умовах ефективна автентифікація повинна бути багатофакторною, адаптивною до ризиків і інтегрованою в загальну стратегію кіберзахисту. Її розвиток та вдосконалення є критично важливими для побудови безпечного цифрового суспільства.

Авторизація – це процес надання або обмеження прав доступу користувачеві до певних ресурсів, сервісів або функцій у межах інформаційної системи після проходження ним автентифікації. Якщо автентифікація відповідає на запитання «Хто ви?», то авторизація визначає «Що вам дозволено робити?». Це один із ключових етапів у побудові політик інформаційної безпеки, оскільки навіть після успішної ідентифікації не кожен користувач повинен мати повний доступ до всіх елементів системи.

Авторизація базується на принципах розподілу повноважень, що дозволяє обмежити ризики, пов’язані з несанкціонованими або помилковими діями в системі. Вона реалізується через контроль доступу, який визначає, які саме операції може виконувати користувач: читати, змінювати, видаляти, створювати об’єкти, використовувати певні інструменти або функції.

Найчастіше механізм авторизації базується на ролях або групах користувачів. Це так званий підхід RBAC (RoleBased Access Control) – контроль доступу на основі ролей. У цьому випадку кожному користувачеві призначається певна роль, яка має фіксований набір дозволених дій. Наприклад, у корпоративній системі адміністратор може мати повні права, звичайний працівник – обмежений доступ до окремих файлів, а зовнішній користувач – лише функцію перегляду.

Існують і більш гнучкі моделі, зокрема ABAC (AttributeBased Access Control), які враховують не лише роль, а й атрибути користувача (місце перебування, час доступу, тип пристрою тощо), а також DAC (Discretionary Access Control) – систему, у якій власник ресурсу сам вирішує, кому і який доступ надати. Такі моделі дозволяють точніше адаптувати авторизацію до контексту користування та безпекових вимог.

Реалізація авторизації відбувається на рівні операційних систем, баз даних, вебзастосунків, мережевих сервісів тощо. Наприклад, у файловій системі Windows або Linux можна вручну призначити права доступу до окремих папок чи файлів. У вебзастосунках авторизація часто реалізується через токени доступу, cookie, сесії або JSON Web Tokens (JWT), які зберігають інформацію про дозволи користувача.

Порушення або неправильна реалізація авторизації є серйозною загрозою. Якщо система неправильно перевіряє права доступу, зловмисник може отримати доступ до конфіденційних даних або критичних функцій, навіть не маючи відповідного рівня привілеїв. Такі помилки часто фіксуються в OWASP Top 10 – списку найнебезпечніших вразливостей вебзастосунків.

Також важливо розрізняти авторизацію та аутентифікацію: авторизація відбувається після автентифікації й не може існувати без неї. Спочатку система повинна впевнитися, що користувач – це справді та особа, за яку себе видає, а вже потім вирішувати, які саме дії цій особі дозволено виконувати. Наприклад, в електронному банкінгу клієнт спочатку вводить логін і пароль (автентифікація), а вже після цього йому відкривається доступ лише до власного рахунку, а не до рахунків інших користувачів (авторизація).

Таким чином, авторизація є важливою складовою системи інформаційної безпеки, яка дозволяє впровадити принцип найменших привілеїв (least privilege) – користувач має тільки ті права, які необхідні йому для виконання поточних задач, і не більше. Це мінімізує ризик зловживань, випадкових помилок і цілеспрямованих атак. Правильно побудована система авторизації є не лише захисним механізмом,

а й інструментом управління доступом, який забезпечує ефективність, гнучкість і безпеку всієї інформаційної архітектури.

Шифрування є одним із фундаментальних методів забезпечення інформаційної безпеки в цифровому середовищі. Це процес перетворення інформації з відкритого (читабельного) вигляду у зашифрований (недоступний для стороннього прочитання) за допомогою спеціального алгоритму та ключа. Головна мета шифрування – забезпечення конфіденційності даних, тобто унеможливлення їх прочитання будьким, крім уповноваженого адресата.

У процесі шифрування застосовуються два основні компоненти: криптографічний алгоритм і криптографічний ключ. Алгоритм визначає правила перетворення даних, тоді як ключ – це параметр, що впливає на результат шифрування і є унікальним для кожної сесії чи користувача. Без правильного ключа зашифрована інформація залишається недоступною навіть при повному доступі до її копії. У сучасній практиці використовуються різні види шифрування залежно від мети, структури системи, швидкодії та рівня захищеності, який необхідно забезпечити.

Умовно всі криптографічні методи шифрування поділяються на два основні типи: симетричне та асиметричне шифрування. Симетричне шифрування передбачає використання одного й того самого ключа як для шифрування, так і для дешифрування повідомлення. До таких алгоритмів належать AES (Advanced Encryption Standard), DES (Data Encryption Standard), RC4 тощо. Основною перевагою симетричного шифрування є його висока швидкість, що дозволяє ефективно використовувати його для захисту великих обсягів даних. Однак основним недоліком є проблема безпечної передачі ключа між відправником і отримувачем, оскільки перехоплення ключа зводить нанівець весь захист.

На противагу цьому, асиметричне шифрування використовує пару ключів – відкритий (public key) та закритий (private key). Відкритий ключ доступний усім і використовується для шифрування, тоді як закритий ключ зберігається в таємниці та застосовується для дешифрування. Найвідомішими асиметричними

алгоритмами є RSA, DSA, ECC (еліптична криптографія). Асиметричне шифрування дозволяє уникнути необхідності передавати секретний ключ відкритим каналом, що значно підвищує рівень безпеки при обміні даними в мережах. Проте цей метод має вищу обчислювальну складність, тому зазвичай застосовується лише для шифрування коротких повідомлень, обміну ключами або цифрових підписів.

На практиці найчастіше використовується комбінований підхід, коли для початкового встановлення безпечного з'єднання застосовується асиметричне шифрування, а для подальшої передачі даних – симетричне. Така схема використовується, зокрема, у протоколі TLS/SSL, який забезпечує захищений обмін у браузерах і є основою безпечного з'єднання HTTPS.

Крім забезпечення конфіденційності, шифрування також відіграє ключову роль у досягненні таких цілей, як цілісність, автентичність і невідмовність. Так, наприклад, цифрові підписи засновані на асиметричному шифруванні та хешуванні: відправник створює унікальний підпис, який можна перевірити за допомогою відкритого ключа. Це дозволяє гарантувати, що повідомлення справді надійшло від заявленого відправника й не було змінене під час передачі.

Особливе значення шифрування має в умовах хмарних технологій, віддаленої роботи, електронного документообігу, мобільних комунікацій та електронного врядування. В умовах масового зберігання й передавання персональних даних, фінансової інформації, медичних записів, державних та комерційних секретів без шифрування неможливо уявити належний рівень цифрової безпеки.

Водночас необхідно враховувати, що ефективність шифрування залежить не лише від використаного алгоритму, а й від належного управління ключами. Збереження, оновлення, ротація, передача та знищення ключів – усе це має здійснюватися в межах чітко визначеної політики безпеки. Порушення цих процедур може призвести до компрометації навіть найсучасніших криптографічних систем.

Також слід зазначити, що розвиток квантових обчислень становить потенційний виклик для класичних алгоритмів шифрування, особливо RSA та ECC. У зв’язку з цим сьогодні активно ведуться дослідження в галузі постквантової криптографії, яка покликана забезпечити стійкість криптографічних алгоритмів у нових обчислювальних умовах.

Узагальнюючи, шифрування є основоположним методом забезпечення цифрової безпеки. Воно виконує функцію «зашивання» інформації від сторонніх очей та дає змогу безпечно передавати, зберігати й обробляти дані в агресивному інформаційному середовищі. Його ефективне впровадження є ключовим елементом будьякої стратегії захисту інформації, як на особистому, так і на корпоративному чи державному рівнях.

Хешування є одним із ключових механізмів забезпечення інформаційної безпеки, який використовується переважно для перевірки цілісності даних та зберігання паролів. Це процес одностороннього криптографічного перетворення довільного за розміром вхідного повідомлення у фіксовану послідовність бітів – так звану хешсукупність або хешзначення. Хешфункції широко застосовуються у комп’ютерних системах, цифрових підписах, автентифікації, блокчейнах, криптовалюті, а також при створенні цифрових відбитків даних.

На відміну від шифрування, хешування не передбачає зворотного перетворення: воно є незворотним (oneway). Тобто, знаючи лише хешзначення, неможливо відновити оригінальні дані – принаймні, це має бути обчислювально неможливо за умов правильної реалізації алгоритму. Саме це забезпечує надійність і практичну цінність хешування для сфери кібербезпеки.

Ефективна хешфункція повинна мати низку властивостей:

- Детермінованість одне й те саме повідомлення завжди дає однаковий хеш.
- Швидкість обчислення функція має бути швидкою навіть для великих обсягів даних.
- Невідворотність неможливість відновлення початкових даних із хешу.

– Стійкість до колізій надзвичайно мала ймовірність того, що два різні повідомлення матимуть однаковий хеш.

– Стійкість до підбору складність знаходження повідомлення, яке має заданий хеш (preimage resistance).

Найвідомішими і широко використовуваними алгоритмами хешування є MD5, SHA1, SHA2 (зокрема SHA256, SHA512) та новіші, якот SHA3. MD5 і SHA1 наразі вважаються недостатньо безпечними через наявність виявлених уразливостей, зокрема зламів через колізії, тому їх використання не рекомендується в критичних системах.

Однією з основних сфер застосування хешування є зберігання паролів. Сучасні системи автентифікації зберігають не самі паролі, а їх хешзначення. Коли користувач вводить пароль, система хешує його введене значення й порівнює з тим, що зберігається в базі. Таким чином, навіть при витоку бази даних зловмисник не отримає справжні паролі – лише їх хеші. Щоб підвищити безпеку цього підходу, використовуються техніки «соління» (salting) – до пароля додається випадкова строка (сіль), яка також хешується. Це захищає навіть від атак із використанням попередньо створених таблиць хешів (так званих rainbow tables).

Хешування також широко використовується для перевірки цілісності файлів. Під час завантаження програмного забезпечення з офіційних сайтів часто публікується SHA256хешфункція інсталяційного файлу. Користувач може самостійно обчислити хеш отриманого файлу й порівняти його з офіційним значенням. Якщо вони збігаються – файл не був змінений, підроблений або пошкоджений. Якщо ж хешзначення різняться – файл міг бути змінений або заражений шкідливим ПЗ.

Ще однією важливою сферою застосування хешування є цифрові підписи. Перед створенням цифрового підпису для документа він спочатку хешується, після чого отримане хешзначення шифрується закритим ключем відправника. Це дозволяє ефективно перевіряти автентичність і цілісність документа за допомогою відкритого ключа.

У сучасних технологіях надзвичайно популярним стало хешування в блокчейнінфраструктурі. Наприклад, у криптовалютах, таких як Bitcoin, хешфункції використовуються для формування блоків, перевірки транзакцій і обчислення ProofofWork (доказу виконаної роботи), що забезпечує децентралізацію та безпеку всієї мережі.

Попри всі переваги, хешування не є абсолютним захистом. Існують методи атак, зокрема атаки грубої сили (bruteforce), атаки за словником, а також атаки на колізії. Для протидії цим загрозам необхідно використовувати сучасні стійкі алгоритми хешування, комбінувати їх із сольовими значеннями, а також впроваджувати системи виявлення несанкціонованої активності.

Узагальнюючи, хешування – це потужний інструмент захисту даних, який забезпечує швидку й ефективну перевірку достовірності та цілісності інформації. Завдяки своїй універсальності й криптографічній стійкості, хешфункції стали невід'ємною частиною сучасних систем захисту, застосовуються в ІТбезпеці, цифровій ідентифікації, блокчейні, електронному підписі та багатьох інших сферах.

1.3 Сучасні засоби: AES, SSL/TLS, 2FA, Biometric ID

В умовах стрімкого розвитку інформаційних технологій та зростання обсягів оброблюваних даних питання захисту інформації користувачів набуває особливої актуальності. Сучасні методи і засоби захисту інформації базуються на використанні криптографічних алгоритмів, протоколів безпечної передачі даних, а також багатофакторної автентифікації та біометричних технологій. Нижче розглянуто основні сучасні засоби, які широко застосовуються для забезпечення безпеки цифрової інформації.

Таблиця 1.3 – Характеристика сучасних засобів захисту інформації користувачів у цифровому просторі

Назва	Тип захисту	Переваги	Недоліки
AES	Симетричне шифрування	Висока швидкість, стійкість до злому, стандартизований алгоритм	Потребує безпечної передачі ключа
SSL/TLS	Протокол безпечної передачі	Конфіденційність, цілісність, автентичність даних, сумісність із сучасними вебтехнологіями	Потребує правильного налаштування, можливі вразливості в старих версіях
2FA	Механізм багаторівневої автентифікації	Значно підвищує захист облікових записів, знижує ризик викрадення паролів	Може бути менш зручним, можливість втрати другого фактора
Biometric ID	Біометрична автентифікація	Зручність, висока точність, немає потреби запам’ятовувати паролі	Питання конфіденційності, складність у випадку змін зовнішності або пошкодження сенсорів

Advanced Encryption Standard (AES) – це один із найпоширеніших і найнадійніших симетричних алгоритмів шифрування, який використовується для захисту інформації у цифровому просторі. Він був офіційно затверджений Національним інститутом стандартів і технологій США (NIST) у 2001 році як заміна застарілого алгоритму DES (Data Encryption Standard), який через обмежену довжину ключа та вразливості став недостатньо безпечним.

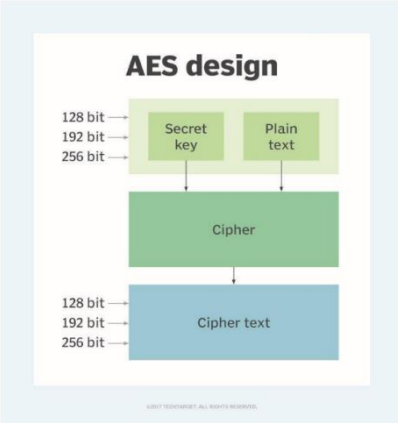

Рисунок 1.1 – Головна структура AES

AES належить до категорії блочних симетричних шифрів, тобто працює з фіксованими блоками даних по 128 біт і для шифрування, і для дешифрування використовує один і той же секретний ключ. Довжина ключа може бути 128, 192 або 256 біт, що дозволяє балансувати між рівнем безпеки та продуктивністю. Чим довший ключ, тим вищий рівень захисту, але водночас збільшується обчислювальне навантаження.

Процес шифрування AES складається із серії раундів, кількість яких залежить від довжини ключа: 10 раундів для 128бітного ключа, 12 – для 192бітного, і 14 – для 256бітного. Кожен раунд включає низку складних перетворень, серед яких підстановка байтів відповідно до таблиці Sbox для створення нелінійності, циклічний зсув рядків, перемішування стовпців за допомогою матричних операцій у кінцевому полі, а також додавання раундового ключа, що формується із основного ключа. Завдяки такій структурі AES забезпечує високу криптографічну стійкість, ускладнюючи будьякі спроби розшифрувати дані без знання ключа.

AES відрізняється не лише своєю надійністю, а й швидкістю роботи. Він оптимізований для ефективного виконання як на програмному, так і на апаратному рівні, що дозволяє використовувати його для шифрування великих обсягів інформації в режимі реального часу. Цей алгоритм широко застосовується у захисті мережевого трафіку (наприклад, у VPN, WiFi мережах WPA2, протоколах

SSL/TLS), у шифруванні файлів, жорстких дисків та мобільних пристроїв, а також у різноманітних криптографічних протоколах.

Завдяки міжнародній стандартизації (FIPS 197) і підтримці більшості сучасних операційних систем та апаратних платформ, AES став основним інструментом для забезпечення конфіденційності і цілісності даних у цифровому світі. Сьогодні він є надійним бар'єром на шляху несанкціонованого доступу та захищає користувачів від кіберзагроз, що постійно зростають у глобальній мережі.

SSL (Secure Sockets Layer) та його наступник TLS (Transport Layer Security) – це криптографічні протоколи, які забезпечують захищене з'єднання між клієнтом, наприклад, веббраузером, та сервером у мережі Інтернет. Вони є основою для безпечного обміну інформацією, гарантують конфіденційність, цілісність та автентичність даних, що передаються. Історично протокол SSL був розроблений компанією Netscape у середині 1990х років. Спочатку існувало кілька версій SSL, однак через виявлені вразливості він був замінений протоколом TLS, який став сучасним стандартом безпеки. Перша версія TLS базувалася на SSL 3.0, але включала суттєві покращення у безпеці. Сьогодні використовуються версії TLS 1.2 і TLS 1.3, які вважаються найбільш безпечними та ефективними.

Основною функцією SSL/TLS є шифрування даних, що передаються між клієнтом і сервером. Це гарантує, що навіть у випадку перехоплення інформації третіми особами, без доступу до відповідних ключів дані залишатимуться незрозумілими. Для цього застосовується симетричне шифрування (наприклад, алгоритм AES), ключі для якого обмінюються за допомогою асиметричної криптографії (RSA або ECC). Протокол також забезпечує автентифікацію сервера за допомогою цифрових сертифікатів, виданих довіреними центрами сертифікації, що дозволяє клієнту переконатися у справжності сервера та уникнути атак типу "людина посередині". Крім того, SSL/TLS гарантує цілісність переданих даних, запобігаючи їх зміні або підробці під час передачі, використовуючи механізми хешування і контрольних сум.

Процес встановлення захищеного з’єднання, відомий як TLS Handshake, включає кілька етапів. Спочатку клієнт надсилає серверу інформацію про підтримувані алгоритми шифрування і версію протоколу. Сервер відповідає вибором найбільш підходящих параметрів і відправляє свій цифровий сертифікат. Клієнт перевіряє сертифікат, підтверджуючи його дійсність та довіру до центру сертифікації. Після цього відбувається обмін ключами, які використовуються для симетричного шифрування подальшої сесії. Весь цей процес дозволяє встановити захищене і швидке з’єднання.

SSL/TLS широко застосовуються у багатьох інтернетсервісах: для захисту вебсайтів за допомогою протоколу HTTPS, для безпечної передачі електронної пошти, у VPNз’єднаннях, а також у різноманітних додатках, де важлива конфіденційність інформації. Завдяки SSL/TLS користувачі можуть безпечно вводити свої особисті дані, паролі та платіжну інформацію, знижуючи ризик кібершахрайства. Протокол постійно вдосконалюється, щоб відповідати новим загрозам і забезпечувати високий рівень безпеки в цифровому середовищі, що швидко змінюється. Таким чином, SSL/TLS є невід’ємною складовою сучасної інфраструктури безпеки в мережі Інтернет.

Двофакторна аутентифікація, або 2FA (TwoFactor Authentication), – це сучасний метод підвищення безпеки доступу до цифрових систем і сервісів, який базується на використанні двох незалежних факторів для підтвердження особи користувача. Ця технологія значно ускладнює несанкціонований доступ, оскільки для проходження перевірки потрібно не лише знати пароль, але й мати другий унікальний елемент, що підтверджує право користувача на доступ.

Ідея двофакторної аутентифікації полягає в поєднанні двох різних типів факторів: щось, що користувач знає (наприклад, пароль або ПІНкод), і щось, що користувач має або є. Другий фактор може бути фізичним пристроєм – апаратним токеном або смартфоном, який генерує одноразові коди, або біометричними даними, такими як відбитки пальців чи розпізнавання обличчя. Саме це поєднання

значно знижує ризик компрометації облікового запису навіть у разі викрадення або підбору пароля, адже зловмиснику знадобиться також отримати другий фактор.

Процес двофакторної аутентифікації зазвичай починається зі стандартного введення логіна і пароля користувачем. Після успішної верифікації цих даних система запитує другий фактор. Найпоширенішим варіантом є одноразовий код, який генерується спеціальним додатком (наприклад, Google Authenticator або Authy) або надсилається через SMS на мобільний телефон користувача. Іноді другий фактор може бути у вигляді апаратного ключа безпеки (наприклад, USBтокена), який потрібно фізично підключити до пристрою. Якщо другий фактор введений правильно, користувач отримує доступ до системи.

Впровадження 2FA має вирішальне значення для захисту особистих даних, банківських рахунків, корпоративних мереж та інших критично важливих інформаційних ресурсів. Навіть якщо пароль потрапляє до рук зловмисника через фішингову атаку або витік даних, наявність другого фактора значно ускладнює несанкціонований доступ. Завдяки цьому двофакторна аутентифікація стає одним із найбільш ефективних і доступних засобів підвищення кібербезпеки.

У сучасних інформаційних системах 2FA широко використовується у вебсервісах, онлайнбанкінгу, корпоративних системах управління доступом, хмарних платформах та мобільних додатках. Постійний розвиток технологій дозволяє розширювати методи двофакторної аутентифікації, включаючи біометричні засоби, апаратні ключі безпеки, а також більш зручні і безпечні способи підтвердження особи, які не потребують введення кодів вручну.

Таким чином, двофакторна аутентифікація є невід’ємною частиною сучасної системи захисту цифрової інформації, яка значно знижує ризики несанкціонованого доступу та забезпечує більш високий рівень безпеки для користувачів у цифровому просторі.

Біометрична ідентифікація (Biometric ID) – це сучасний метод автентифікації, який базується на унікальних фізіологічних або поведінкових характеристиках людини для підтвердження її особи. На відміну від традиційних

способів аутентифікації, таких як паролі або токени, біометричні дані важко підробити або вкрасти, що робить цей метод одним із найнадійніших засобів захисту інформації в цифровому просторі. Біометрія використовує природні особливості людини, які залишаються стабільними протягом життя та відрізняють її від інших користувачів.

До найбільш поширених видів біометричної ідентифікації належать відбитки пальців, розпізнавання обличчя, сканування райдужної оболонки ока, а також голосова аутентифікація і аналіз поведінкових моделей, таких як почерк або спосіб набору тексту. Кожен із цих методів має свої особливості, але всі вони зводяться до збору, збереження та порівняння біометричних шаблонів, що забезпечує ідентифікацію користувача з високою точністю. Технології біометрії постійно вдосконалюються, що дозволяє підвищувати швидкість розпізнавання і знижувати кількість помилкових спрацьовувань.

Процес біометричної аутентифікації починається зі збору біометричних даних користувача за допомогою спеціальних сенсорів, таких як сканери відбитків пальців або камери для розпізнавання обличчя. Ці дані перетворюються на цифрові шаблони, які зберігаються в захищеній базі даних або безпосередньо на пристрої користувача. При наступному вході система порівнює отриманий біометричний зразок із шаблоном у базі. Якщо збіг є достатньо точним, доступ надається. Такий підхід виключає необхідність запам’ятовування паролів і значно знижує ризик компрометації облікових записів.

Біометрична ідентифікація застосовується у різних сферах, починаючи від мобільних пристроїв, де вона використовується для розблокування смартфонів і авторизації у додатках, до великих корпоративних систем і державних програм. Вона активно використовується у контролі доступу на об’єктах з високими вимогами безпеки, у банківській сфері для підтвердження операцій, а також у системах електронного уряду та прикордонного контролю. Завдяки своїй зручності і надійності біометрія сприяє підвищенню якості сервісу і водночас посилює захист персональних даних.

Проте використання біометричних технологій викликає і певні питання, зокрема щодо приватності та збереження конфіденційності особистих даних. Оскільки біометричні характеристики є унікальними і незмінними, їх компрометація може мати серйозні наслідки, тому системи повинні відповідати високим стандартам захисту та шифрування. Водночас, розвиток технологій дозволяє реалізувати біометричні рішення, які зберігають дані локально на пристрої, мінімізуючи ризики витоку.

Таким чином, біометрична ідентифікація є одним із найбільш перспективних і ефективних методів захисту інформації у цифровому просторі. Вона поєднує в собі високу точність, зручність для користувача та надійний рівень безпеки, що робить її незамінною складовою сучасних систем аутентифікації.

1.4 Недоліки існуючих підходів: складність інтеграції, людський фактор, застарілі алгоритми

Незважаючи на постійний розвиток та вдосконалення технологій захисту інформації в цифровому просторі, сучасні підходи все ще мають низку суттєвих недоліків, які ускладнюють їх широке впровадження та ефективне функціонування. Однією з основних проблем залишається складність інтеграції новітніх засобів захисту в існуючі інформаційні системи. Багато компаній, особливо середнього та малого бізнесу, використовують застарілу інфраструктуру, яка не підтримує сучасні протоколи безпеки, або ж не має ресурсів на масштабну модернізацію. Запровадження передових методів, таких як багатофакторна автентифікація чи біометрична ідентифікація, часто вимагає суттєвих технічних змін, закупівлі додаткового обладнання та перенавчання персоналу, що потребує часу, коштів та високої кваліфікації.

Ще одним важливим обмеженням є людський фактор, який залишається одним з найслабших місць у системах кібербезпеки. Навіть найсучасніші засоби захисту не можуть гарантувати безпеку, якщо користувачі проявляють недбалість

або не мають належної кібергігієни. Частими є випадки використання простих і легко вгадуваних паролів, нехтування двофакторною автентифікацією, натискання на шкідливі посилання в фішингових листах, або передавання конфіденційних даних стороннім особам. Помилки користувачів часто спричиняють серйозні витоки інформації або відкривають доступ до систем для зловмисників, що свідчить про необхідність постійного навчання і підвищення обізнаності щодо інформаційної безпеки.

Окрему загрозу становить використання застарілих алгоритмів шифрування та автентифікації, які більше не забезпечують належного рівня захисту. З розвитком обчислювальної техніки та зростанням потужностей атакувальних інструментів деякі криптографічні стандарти втрачають свою ефективність. Алгоритми, які були надійними десять чи двадцять років тому, сьогодні можуть бути успішно зламані в розумні терміни за допомогою сучасного програмного забезпечення або хмарних сервісів. Вразливість до атак типу «brute force», «maninthemiddle», а також недоліки в реалізації протоколів можуть призводити до компрометації захищених даних. Крім того, існує проблема інерції впровадження нових стандартів: навіть після визнання певного алгоритму небезпечним, його можуть продовжувати використовувати через технічні або організаційні обмеження.

Усе це свідчить про те, що ефективний захист інформації у цифровому середовищі потребує не лише впровадження новітніх технологій, а й комплексного підходу до організації безпеки. Він має включати технічне оновлення, підвищення обізнаності користувачів, постійний аудит безпеки та гнучкість у адаптації до нових викликів і загроз.

2. РОЗРОБКА КОМБІНОВАНОЇ МОДЕЛІ ЗАХИСТУ

2.1 Формування вимог до системи захисту користувача

Проєктування ефективної системи захисту інформації користувачів у цифровому середовищі вимагає чіткого визначення функціональних і нефункціональних вимог до такої системи. Формування вимог є початковим і одним із найважливіших етапів у розробці комбінованої моделі захисту, оскільки саме на цьому етапі визначаються ключові цілі безпеки, характеристики користувачів, загрози, на які система має реагувати, а також контекст її використання.

Передусім система захисту має забезпечувати базові принципи інформаційної безпеки – конфіденційність, цілісність та доступність. Конфіденційність означає, що доступ до персональних або чутливих даних повинен мати лише уповноважений користувач. Цілісність передбачає захист даних від несанкціонованого змінення або спотворення. Доступність, у свою чергу, вимагає, щоб уповноважені користувачі могли отримати доступ до даних у потрібний час без перешкод. Окрім цих основних принципів, актуальними є також автентичність – підтвердження особи користувача, та невідмовність – забезпечення доказу факту доступу або дії.

Формуючи вимоги до майбутньої системи, необхідно враховувати рівень цифрової грамотності користувачів, технічні можливості їхніх пристроїв, а також особливості середовища, в якому відбуватиметься взаємодія. Наприклад, якщо користувачі часто працюють із мобільних пристроїв, система повинна бути адаптивною та підтримувати мобільну автентифікацію. Якщо ж ідеться про корпоративну мережу, особливу увагу варто приділити централізованому управлінню правами доступу, веденню журналів подій, а також інтеграції з існуючими корпоративними сервісами.

Окрему увагу слід приділити вимогам до гнучкості та масштабованості системи. В умовах швидкого розвитку цифрових технологій система захисту має

бути здатною адаптуватися до нових викликів, змін у структурі організації, зростання обсягу даних або кількості користувачів. Гнучкість також означає можливість комбінування різних методів автентифікації – наприклад, пароля, двофакторної перевірки, біометрії – залежно від рівня ризику, контексту доступу або політики безпеки.

Не менш важливими є вимоги до зручності використання. Надмірна складність або незручність процедур захисту може призвести до того, що користувачі почнуть їх ігнорувати або шукати способи обійти захист, що суттєво знизить загальну ефективність системи. Тому необхідно знайти баланс між надійністю безпеки та ергономікою користування. Інтерфейс має бути інтуїтивно зрозумілим, а процедури входу – мінімально обтяжливими, але при цьому надійними.

Рисунок 2.1 – Структурна схема комбінованої моделі захисту інформації користувача

Загалом формування вимог до системи захисту користувача включає ідентифікацію загроз, визначення сценаріїв використання, встановлення цілей безпеки, врахування технічних та організаційних обмежень, а також узгодження інтересів користувачів і розробників. Це дозволяє створити основу для побудови ефективної, надійної та гнучкої комбінованої моделі захисту, яка відповідатиме як сучасним викликам кіберпростору, так і реальним потребам користувачів.

2.2 Вибір криптоалгоритму: AES256 у режимі EAX

Під час створення надійної системи захисту інформації важливим кроком є вибір ефективного криптографічного алгоритму, який би відповідав сучасним вимогам безпеки та дозволяв забезпечити як конфіденційність, так і цілісність даних. У межах розробки комбінованої моделі захисту користувачів у цифровому середовищі доцільним є застосування алгоритму симетричного шифрування AES256 у режимі роботи EAX. Такий вибір обумовлений рядом технічних, криптографічних і практичних переваг, які забезпечують високу стійкість системи до атак і гнучкість її застосування в різних цифрових середовищах.

Advanced Encryption Standard (AES) – це сучасний і стандартизований алгоритм блочного симетричного шифрування, що отримав широке визнання у сфері інформаційної безпеки. Вибір розрядності ключа на 256 біт (AES256) дозволяє досягти найвищого рівня криптографічного захисту серед доступних варіантів цього стандарту. Такий ключовий простір забезпечує надзвичайну стійкість до атак методом повного перебору (bruteforce), навіть із урахуванням стрімкого розвитку обчислювальних потужностей. AES256 активно використовується урядовими, фінансовими та військовими структурами у всьому світі, що свідчить про його ефективність і довіру до нього в критично важливих сферах.

Однак важливим аспектом безпеки є не лише сам алгоритм шифрування, але й режим його використання. У цьому контексті режим EAX (EncryptthenAuthenticate Extension) є одним із найефективніших, оскільки поєднує переваги конфіденційного шифрування даних з одночасною перевіркою їхньої цілісності та автентичності. Режим EAX реалізує принцип шифрування з автентифікацією (Authenticated Encryption), що означає захист не лише від несанкціонованого доступу, але й від підробки або модифікації зашифрованої інформації.

Режим EAX має низку переваг у порівнянні з традиційними режимами, такими як CBC (Cipher Block Chaining) або ECB (Electronic Codebook). Поперше, він не потребує складного механізму обробки помилок – у разі виявлення змін у повідомленні воно повністю відкидається. Подруге, він забезпечує детерміновану та прозору структуру повідомлення, що важливо для безпечного зберігання або передачі критичних даних. Потретє, режим EAX не має вразливостей, пов'язаних із передачею однакових блоків у різних повідомленнях, що знижує ризик витоку шаблонів і структури даних.

У практичному сенсі AES256 у режимі EAX добре підходить як для серверного, так і для клієнтського шифрування, зокрема в браузерах, мобільних додатках, хмарних сховищах і базах даних. Реалізація цього алгоритму можлива в більшості сучасних криптографічних бібліотек (наприклад, OpenSSL, PyCryptodome, Libsodium), що дозволяє легко інтегрувати його в існуючі інфраструктури. Додатково, режим EAX не вимагає складного керування ініціалізаційними векторами, що спрощує процес програмної реалізації.

Таким чином, обґрунтоване впровадження AES256 у режимі EAX як основного криптографічного інструмента у комбінованій моделі захисту користувачів дозволяє гарантувати високий рівень інформаційної безпеки, поєднуючи надійне шифрування з автентифікацією, що є критично важливим у сучасному цифровому середовищі.

2.3 Алгоритм OTP (одноразового пароля) на основі TOTP (Timebased One Time Password)

Один із ключових елементів сучасної системи автентифікації – це механізм створення одноразових паролів (OTP), який значно підвищує стійкість до несанкціонованого доступу, особливо у випадках, коли основний пароль може бути скомпрометований. У рамках розробки комбінованої моделі захисту інформації

користувачів доцільно впровадити алгоритм TOTP (Timebased OneTime Password) – тобто одноразові паролі, що генеруються на основі поточного часу.

TOTP є розширенням стандартного підходу OTP, що базується на алгоритмі HMAC (Hashbased Message Authentication Code). Основною ідеєю TOTP є створення тимчасово чинного коду, який оновлюється через заздалегідь визначений інтервал (зазвичай кожні 30 секунд). Це означає, що навіть якщо одноразовий код перехоплений, його використання зловмисником стає малоймовірним через короткий строк дії. В основі алгоритму лежить поєднання секретного ключа (shared secret), який зберігається як у клієнта (наприклад, мобільному застосунку типу Google Authenticator або Microsoft Authenticator), так і на сервері, з поточним часовим відрізком. Результатом є шістка або вісімка цифрових цифр, що і є одноразовим паролем.

Алгоритм TOTP має декілька важливих переваг. Поперше, він не потребує постійного підключення до інтернету на боці користувача: генерація одноразового пароля відбувається локально на пристрої, що суттєво підвищує автономність та стійкість до атак типу "людина посередині" (MITM). Подруге, він є відкритим стандартом, описаним у RFC 6238, що забезпечує його сумісність із багатьма платформами та інструментами. Потретє, система на основі TOTP добре масштабується: вона підходить як для автентифікації користувачів у невеликих застосунках, так і в масштабних корпоративних системах.

У практичній реалізації TOTP можна налаштувати як частину двофакторної автентифікації (2FA), де користувач вводить стандартний логін і пароль, а потім – додатковий код, згенерований мобільним пристроєм. Це значно ускладнює можливість несанкціонованого входу, адже зловмисник повинен мати не лише пароль, але й доступ до пристрою користувача або до його секретного ключа. Більше того, у разі втрати пристрою можна реалізувати процедури резервного відновлення доступу, що також враховується при створенні повноцінної системи захисту.

З технічного боку, TOTP є безпечним лише за умови належного зберігання секретного ключа, а також синхронізації годинників між клієнтом і сервером. При цьому реалізації часто враховують похибку у часі (наприклад, у межах одногодвох періодів), що зменшує кількість помилок при вході, але при цьому не знижує загальну безпеку.

Таким чином, інтеграція алгоритму TOTP як механізму генерації одноразових паролів у комбіновану модель захисту є обґрунтованим кроком, що дозволяє реалізувати ефективну, масштабовану й стійку до зловживань систему автентифікації користувача. Це сприяє значному зниженню ризику компрометації облікових даних навіть у разі витоку основного пароля.

2.4 Схема моделі: Користувач → 2FA → Шифрування → База даних

Модель, що включає послідовність етапів «Користувач → 2FA → Шифрування → База даних», є класичною схемою побудови захищених інформаційних систем, які спрямовані на забезпечення високого рівня безпеки доступу та збереження даних. На початковому етапі взаємодії з системою виступає користувач, який має отримати доступ до захищеної інформації або сервісу. Для цього він вводить свої облікові дані, як правило, логін та пароль. Однак сам по собі цей традиційний спосіб аутентифікації вже не забезпечує достатнього рівня захисту у зв’язку з поширеністю методів зломів, таких як фішинг, перебір паролів або крадіжка облікових даних (див. рисунок 2.2).

Щоб посилити безпеку та уникнути ризику несанкціонованого доступу, впроваджується двофакторна аутентифікація (2FA). Цей механізм передбачає підтвердження особи користувача за допомогою двох різних факторів, що істотно знижує ймовірність компрометації. Першим фактором, як правило, є щось, що користувач знає – наприклад, пароль. Другим фактором виступає щось, що користувач має – це може бути одноразовий код, який генерується апаратним токеном, програмоюгенератором або надсилається SMSповідомленням на

мобільний телефон. Таким чином, навіть якщо зловмисник здобуде пароль, він не зможе увійти в систему без другого фактора, який є у розпорядженні лише легітимного користувача.

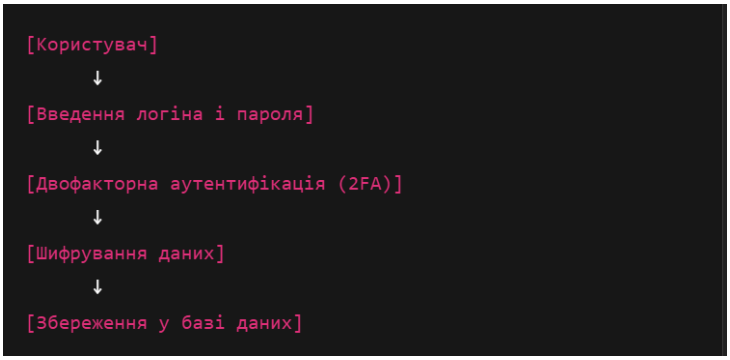

Рисунок 2.2 – Схема моделі: Користувач → 2FA → Шифрування → База даних

Щоб посилити безпеку та уникнути ризику несанкціонованого доступу, впроваджується двофакторна аутентифікація (2FA). Цей механізм передбачає підтвердження особи користувача за допомогою двох різних факторів, що істотно знижує ймовірність компрометації. Першим фактором, як правило, є щось, що користувач знає – наприклад, пароль. Другим фактором виступає щось, що користувач має – це може бути одноразовий код, який генерується апаратним токеном, програмоюгенератором або надсилається SMSповідомленням на мобільний телефон. Таким чином, навіть якщо зловмисник здобуде пароль, він не зможе увійти в систему без другого фактора, який є у розпорядженні лише легітимного користувача.

Після успішного проходження аутентифікації необхідно забезпечити безпечну обробку і зберігання даних, які вводить користувач або які генеруються системою. Для цього застосовується шифрування – процес трансформації інформації у форму, недоступну для прочитання без спеціального криптографічного ключа. Шифрування є критично важливим для захисту

конфіденційності, цілісності та автентичності даних. Воно дозволяє зменшити ризики витоку інформації під час її передачі в мережі або зберігання у базі даних. Залежно від конкретних вимог системи, можуть використовуватися різні типи шифрування – симетричне, де один ключ використовується і для шифрування, і для розшифрування, або асиметричне, що використовує пару ключів – публічний і приватний. Завдяки цьому, навіть у разі компрометації окремих компонентів системи, самі дані залишаються захищеними.

Заключним етапом у цій схемі є безпечне збереження зашифрованих даних у базі даних. Система управління базою даних (СУБД) відповідає за структурування, збереження і управління інформацією, що робить її доступною лише для авторизованих користувачів і служб. Для додаткового захисту база даних може бути обладнана механізмами контролю доступу, аудитом подій, резервним копіюванням та моніторингом активності. У поєднанні із двофакторною аутентифікацією і криптографічним захистом, це створює багаторівневий бар’єр, що ефективно перешкоджає несанкціонованому доступу і потенційному викраденню або зміні інформації.

Таким чином, модель «Користувач → 2FA → Шифрування → База даних» забезпечує комплексний підхід до інформаційної безпеки, поєднуючи підтвердження автентичності користувача, захист даних на рівні передачі і збереження, а також контроль доступу до інформації на рівні сховища. Цей підхід відповідає сучасним вимогам до захисту інформаційних систем і є фундаментом для побудови надійних, стійких до зовнішніх і внутрішніх загроз рішень.

3. ПРАКТИЧНА РЕАЛІЗАЦІЯ ТА ОЦІНКА ЕФЕКТИВНОСТІ

3.1 Реалізація програмної частини: Flask, PyCryptodome, PyOTP

Для створення надійної та безпечної вебсистеми в якості основного серверного фреймворка було обрано Flask – легкий і гнучкий мікрофреймворк на мові Python. Flask ідеально підходить для розробки вебдодатків середньої складності, адже він надає мінімальний каркас, що дозволяє гнучко будувати логіку без надлишкової абстракції. Ця простота робить Flask популярним вибором, особливо коли потрібна швидка реалізація з можливістю подальшого масштабування. Завдяки модульності Flask легко інтегрується з іншими бібліотеками, що розширюють його функціональність – у нашому випадку це PyCryptodome і PyOTP.

PyCryptodome – це сучасна криптографічна бібліотека для Python, яка пропонує широкий набір алгоритмів як симетричного, так і асиметричного шифрування, хешування, генерації випадкових чисел, цифрових підписів тощо. Вона є заміною застарілої бібліотеки PyCrypto і підтримує оновлені стандарти безпеки. Використання PyCryptodome дає змогу реалізувати важливі механізми захисту даних – наприклад, безпечне шифрування паролів, секретних ключів та інших конфіденційних відомостей. У рамках розробки системи на базі Flask, PyCryptodome застосовується для шифрування даних, що зберігаються в базі, а також для захисту інформації під час її передачі між клієнтом і сервером. Особливо важливо, що бібліотека дозволяє реалізувати такі алгоритми, як AES (Advanced Encryption Standard) у різних режимах, що забезпечує високий рівень криптозахисту. Крім того, бібліотека підтримує генерацію надійних випадкових чисел, що використовується для створення криптографічних ключів.

PyOTP – це спеціалізована бібліотека, що реалізує алгоритми одноразових паролів, у тому числі популярний стандарт TOTP (Timebased OneTime Password). TOTP є одним із найбільш поширених методів двофакторної аутентифікації (2FA), що базується на синхронізації часових кодів між сервером і користувачем.

Користувач, встановивши на свій смартфон додатокгенератор кодів (наприклад, Google Authenticator або Authy), отримує одноразові шестизначні коди, що змінюються кожні 30 секунд. Для реалізації такої функціональності на сервері, через Flask, створюється механізм генерації та зберігання унікального секретного ключа для кожного користувача. Цей ключ зберігається в зашифрованому вигляді за допомогою PyCryptodome для захисту від несанкціонованого доступу. Під час авторизації користувач вводить не тільки логін і пароль, а й одноразовий код, який сервер перевіряє за допомогою PyOTP. Якщо код коректний і актуальний у межах часових рамок, користувач отримує доступ до системи.

Інтеграція Flask, PyCryptodome та PyOTP побудована на принципі модульності і безпеки. Flask приймає HTTPзапити від клієнтів, проводить первинну перевірку автентичності (логін і пароль), а потім запускає другий етап – перевірку двофакторного коду. Весь процес взаємодії захищений завдяки криптографічним операціям PyCryptodome, які шифрують і розшифровують конфіденційні дані, що зберігаються у базі даних або передаються між клієнтом і сервером. Таким чином, система реалізує багаторівневу модель безпеки, яка значно ускладнює несанкціонований доступ і підвищує довіру користувачів.

Особливістю практичної реалізації є також простота розгортання і подальшої підтримки системи. Flask як легкий фреймворк легко інтегрується з базами даних, що дає змогу масштабувати систему, а Pythonбібліотеки PyCryptodome і PyOTP активно підтримуються спільнотою та мають стабільні версії, що гарантує надійність і сумісність із сучасними стандартами безпеки.

Загалом, об’єднання Flask, PyCryptodome та PyOTP у єдину систему дозволяє реалізувати захищений вебдодаток з двофакторною автентифікацією, який відповідає сучасним вимогам безпеки, є простим у використанні та масштабованим для майбутнього розвитку.

3.2 Структура додатку: модулі auth.py, crypto.py, routes.py

Для реалізації безпечного вебдодатку, який підтримує автентифікацію користувачів із двофакторною аутентифікацією (2FA), було застосовано архітектуру, розділену на три ключові модулі: crypto.py, auth.py та routes.py. Такий розподіл функціональності дозволяє підтримувати чистоту коду, забезпечує високий рівень безпеки та спрощує подальший розвиток і тестування системи.

Приклад коду «Модуль crypto.py»

```
from Crypto.Cipher import AES
from Crypto.Random import get_random_bytes
from base64 import b64encode, b64decode
import hashlib

class CryptoManager:
def init(self, key: bytes):
    self.key = hashlib.sha256(key).digest()
def encrypt(self, data: str) > str:
    data_bytes = data.encode('utf8')
    cipher = AES.new(self.key, AES.MODE_GCM)
    ciphertext, tag = cipher.encrypt_and_digest(data_bytes)
    encrypted = b64encode(cipher.nonce + tag +
ciphertext).decode('utf8')
    return encrypted

def decrypt(self, encrypted_data: str) > str:
    decoded = b64decode(encrypted_data)
    nonce = decoded[:16]
    tag = decoded[16:32]
    ciphertext = decoded[32:]
    cipher = AES.new(self.key, AES.MODE_GCM, nonce=nonce)
    data_bytes = cipher.decrypt_and_verify(ciphertext, tag)
    return data_bytes.decode('utf8')

@staticmethod
def generate_key(length=32) > bytes:
    return get_random_bytes(length)
```

Модуль crypto.py виконує всі криптографічні операції, необхідні для захисту конфіденційних даних. За допомогою бібліотеки PyCryptodome тут реалізовано

симетричне шифрування AES у режимі GCM, що забезпечує не тільки конфіденційність, а й цілісність даних.

Основні функції модуля:

- encrypt – шифрує текстові дані, повертаючи їх у форматі base64, що зручно для збереження у базі даних або передачі.
- decrypt – розшифровує зашифровані дані, перевіряючи їх цілісність.
- generate_key – створює випадковий криптографічний ключ необхідної довжини.

Цей модуль інкапсулює всі криптографічні операції, що дозволяє централізовано контролювати безпеку даних та легко оновлювати алгоритми шифрування.

Приклад коду «Модуль auth.py»

```
import pyotp
import bcrypt
from crypto import CryptoManager
class AuthManager:
def init(self, encryption_key: bytes):
    self.crypto = CryptoManager(encryption_key)
def hash_password(self, password: str) > bytes:
    return bcrypt.hashpw(password.encode('utf8'),
bcrypt.gensalt())

def verify_password(self, password: str, hashed: bytes) >
bool:
    return bcrypt.checkpw(password.encode('utf8'), hashed)

def generate_2fa_secret(self) > str:
    return pyotp.random_base32()

def encrypt_2fa_secret(self, secret: str) > str:
    return self.crypto.encrypt(secret)

def decrypt_2fa_secret(self, encrypted_secret: str) > str:
    return self.crypto.decrypt(encrypted_secret)

def verify_2fa_code(self, encrypted_secret: str, code: str)
> bool:
    secret = self.decrypt_2fa_secret(encrypted_secret)
```

```
totp = pyotp.TOTP(secret)
return totp.verify(code)
```

Модуль auth.py відповідає за логіку аутентифікації користувачів: хешування та перевірку паролів, генерацію секретних ключів для 2FA, а також перевірку одноразових кодів. Для безпечного збереження паролів застосовується алгоритм bcrypt, який стійкий до атак перебором.

Важливі аспекти цього модуля:

– Паролі користувачів хешуються за допомогою bcrypt із сольовим додаванням, що робить злом пароля надзвичайно складним.

– Секретні ключі для двофакторної аутентифікації генеруються випадковим чином через PyOTP і додатково шифруються перед збереженням, що запобігає компрометації.

– Перевірка одноразового коду відбувається шляхом розшифрування збереженого секрету та його порівняння з кодом, введеним користувачем.

Приклад коду «Модуль auth.py»

```
stored_hashed_password                                    =
b'$2b$12$examplehashedpassword...'
stored_encrypted_secret = 'exampleEncryptedSecret=='
if not auth_manager.verify_password(password,
stored_hashed_password):
    return jsonify({'message': 'Невірний логін або
пароль'}), 401
if not
auth_manager.verify_2fa_code(stored_encrypted_secret,
code):
    return jsonify({'message': 'Невірний 2FA код'}),
401
session['username'] = username
    return jsonify({'message': 'Вхід успішний'}), 200
if name == 'main':
    app.run(debug=True)
```

Модуль routes.py реалізує основні маршрути вебдодатку на Flask, які взаємодіють із користувачем. Тут визначено маршрути для реєстрації, входу та аутентифікації з двофакторним підтвердженням.

Приклад коду «Модуль routes.py»

```
from flask import Flask, request, jsonify, session
from auth import AuthManager
app = Flask(name) app.secret_key = 'your_flask_secret_key'
auth_manager =
AuthManager(encryption_key=b'my_super_secret_key_32_bytes_l
ong!')
@app.route('/register', methods=['POST'])
def register():
    data = request.json
    username = data.get('username')
    password = data.get('password')
    hashed_password = auth_manager.hash_password(password)
    secret_2fa = auth_manager.generate_2fa_secret()
    encrypted_secret =
auth_manager.encrypt_2fa_secret(secret_2fa)
return jsonify({
    'message': 'Користувач успішно зареєстрований',
    '2fa_secret': secret_2fa  # Повертаємо секрет для
налаштування 2FA
}), 201
@app.route('/login', methods=['POST'])
def login():
    data = request.json
    username = data.get('username')
    password = data.get('password')
    code = data.get('2fa_code')
```

Цей модуль забезпечує:

- Реєстрацію користувача із хешуванням пароля та створенням 2FA секрету.
- Логіку входу з перевіркою пароля та двофакторного коду.
- Управління сесією користувача після успішної аутентифікації.
- Поділ функціональності на модулі crypto.py, auth.py та routes.py дозволяє реалізувати чітку, модульну архітектуру, де:
- crypto.py централізує всі криптографічні операції, що підвищує безпеку та зручність оновлення алгоритмів шифрування;
- auth.py відповідає за безпечну обробку аутентифікації користувачів, включно з хешуванням паролів та підтримкою двофакторної аутентифікації;

– routes.py управляє маршрутизацією HTTPзапитів і взаємодією з користувачем через REST API.

Такий підхід забезпечує надійний захист даних користувачів, модульність системи та спрощує її підтримку й розвиток.

3.3 Результати тестування: коректна автентифікація, швидкість шифрування, аналіз логів

Тестування інформаційнобезпекових систем є невід'ємною частиною процесу їх розробки та впровадження. Його мета – виявлення потенційних вразливостей, підтвердження працездатності функціоналу й оцінка ефективності криптографічних механізмів. У контексті вебзастосунків, що працюють із персональними чи конфіденційними даними, особливо важливою є перевірка таких компонентів, як автентифікація користувачів, криптографічне шифрування інформації та журналювання подій системи (логування).

Одним із ключових аспектів тестування є перевірка механізмів автентифікації, зокрема – точності розпізнавання облікових даних користувача, стійкості до несанкціонованого доступу та наявності підтримки двофакторної аутентифікації (2FA). Теоретично ефективна система автентифікації має забезпечувати правильну ідентифікацію користувача за логіном і паролем, при цьому використовуючи хешфункції з сольовим значенням (наприклад, bcrypt або Argon2), що гарантує захист навіть у разі компрометації бази даних. Крім того, для підвищення рівня безпеки широко застосовується технологія одноразових тимчасових кодів (TOTP), які генеруються на основі секретного ключа та поточного часу, і вимагають незалежного підтвердження при вході до системи.

Ще одним важливим елементом тестування виступає оцінка продуктивності криптографічного шифрування. При використанні симетричних алгоритмів, таких як AES (Advanced Encryption Standard), особливу увагу приділяють не лише криптостійкості, але й швидкодії – часу, необхідному для шифрування та

дешифрування даних. Згідно з теоретичними розрахунками, сучасні алгоритми шифрування у режимі GCM (Galois/Counter Mode) здатні забезпечити як високу швидкість обробки інформації, так і додаткову автентифікацію повідомлень, що дозволяє перевірити їхню цілісність. Для вебсистем важливо, щоб криптографічні операції не створювали затримок у відповіді сервера, особливо при великій кількості паралельних запитів.

Окрему увагу у процесі теоретичного тестування системи слід приділяти аналізу логів – журналів, у яких зберігається інформація про всі значущі події в додатку: реєстрації, спроби входу, помилки автентифікації, зміни паролів, генерацію або валідацію 2FAкодів тощо. Теоретично правильна побудова логування має враховувати як повноту записів, так і їхню безпечність (наприклад, відсутність чутливих даних у відкритому вигляді). Логування слугує не лише для моніторингу роботи системи, а й є важливим інструментом розслідування інцидентів безпеки та виявлення аномальної активності, зокрема – багаторазових помилок при вході, підозрілих спроб обходу автентифікації або порушень часових обмежень при роботі з 2FA.

Усі згадані аспекти – коректна автентифікація, висока продуктивність шифрування та ефективний аналіз логів – у сукупності визначають ступінь надійності інформаційної системи. Їхнє комплексне тестування дозволяє забезпечити не лише відповідність системи стандартам безпеки, а й її стійкість до реальних загроз у цифровому середовищі. Таким чином, теоретичне обґрунтування та належна реалізація цих елементів є ключовими передумовами для створення ефективного програмного забезпечення з високим рівнем інформаційного захисту.

3.4 Висновки щодо ефективності та можливостей масштабування

Ефективність інформаційнозахищених програмних систем є одним із ключових критеріїв їх придатності до використання в умовах реального експлуатаційного середовища. У сучасних умовах зростання обсягів цифрових

даних, підвищення вимог до безпеки обробки інформації та динамічного розвитку кіберзагроз, особливого значення набуває здатність програмних систем поєднувати високу продуктивність із надійними механізмами захисту.

Теоретичне осмислення ефективності таких систем передбачає аналіз сукупності характеристик, зокрема: обчислювальної складності криптографічних операцій, швидкодії системи автентифікації, стабільності функціонування при підвищених навантаженнях, а також відповідності використовуваних алгоритмів сучасним стандартам інформаційної безпеки (зокрема, рекомендаціям OWASP, ISO/IEC 27001, NIST SP 80063 тощо). Ефективною вважається така система, що забезпечує мінімальні витрати ресурсів на реалізацію захисних процедур (шифрування, хешування, генерація токенів, перевірка автентичності), водночас гарантуючи стійкість до типових атак, включно з перебором паролів, підробкою автентифікаційних даних або перехопленням сесій.

З погляду криптографічної стійкості, найбільш придатними для реалізації сучасних систем захисту є алгоритми блочного симетричного шифрування (наприклад, AES у режимі GCM), алгоритми хешування з додатковим сіллю (наприклад, bcrypt або Argon2), а також механізми одноразових кодів, сформованих на основі секретного ключа та часу (TOTP). Такі методи не лише забезпечують високий рівень захисту, але й добре масштабуються завдяки своїй простоті у реалізації на стороні клієнта та сервера.

Окрему увагу в контексті ефективності варто приділити багатофакторній автентифікації (MultiFactor Authentication, MFA), яка значно знижує ризик несанкціонованого доступу навіть у разі компрометації одного з факторів. Теоретично доведено, що поєднання знання (наприклад, пароля) та володіння (наприклад, пристрою з TOTPгенератором) формує більш стійку модель безпеки, порівняно з традиційною однофакторною перевіркою.

Ще одним критичним параметром сучасного безпечного програмного забезпечення є масштабованість, тобто здатність системи зберігати функціональність і продуктивність при зростанні кількості користувачів,

транзакцій або обсягу оброблюваних даних. У контексті вебдодатків масштабованість досягається за допомогою правильної архітектури – зокрема, використанням мікросервісного підходу, модульності програмного коду, підтримки RESTful API, а також розгортанням компонентів у контейнеризованому середовищі (наприклад, Docker або Kubernetes). Такі рішення дозволяють динамічно балансувати навантаження між сервісами, проводити оновлення без простоїв і забезпечувати гнучке горизонтальне масштабування.

Важливою передумовою масштабування також є використання відкритих та перевірених бібліотек і фреймворків, що підтримуються професійною спільнотою. Наприклад, бібліотеки на кшталт PyCryptodome, PyOTP, bcrypt, у поєднанні з легкими серверними фреймворками (наприклад, Flask) дозволяють швидко створювати безпечні вебсервіси, придатні для інтеграції в корпоративну або хмарну інфраструктуру.

Підсумовуючи вищезазначене, можна зробити висновок, що ефективність та масштабованість є взаємопов’язаними характеристиками, що визначають не лише технічну життєздатність програмної системи, але й її відповідність сучасним вимогам до захисту персональних даних, збереження цілісності інформації та забезпечення доступності сервісів. Розробка таких систем повинна базуватися на принципах безпечного проектування, модульності, повторного використання коду та відповідності міжнародним стандартам інформаційної безпеки. Застосування відповідних теоретичних підходів у цьому контексті є запорукою створення масштабованих, гнучких та надійних програмних рішень, здатних забезпечити захист цифрового простору в умовах сучасного кіберсередовища.

ВИСНОВКИ

У процесі виконання кваліфікаційної роботи було здійснено комплексне дослідження проблеми захисту інформації користувачів у сучасному цифровому просторі, що в умовах стрімкої цифровізації суспільства й активного поширення онлайнсервісів набуває особливого значення. Сьогодні цифровий простір охоплює всі сфери життя людини – від особистої комунікації до електронного урядування, фінансових операцій і доступу до конфіденційних медичних чи правових даних. Водночас зростає й кількість кіберзагроз, що спрямовані на порушення конфіденційності, цілісності та доступності інформації. Це обумовлює необхідність створення надійних, адаптивних і багаторівневих систем захисту, здатних ефективно функціонувати в умовах динамічних технологічних змін і зростаючих ризиків.

У межах дослідження було підтверджено, що поодиноке використання навіть найсучасніших методів захисту – таких як шифрування чи автентифікація – не забезпечує достатнього рівня безпеки. Саме комплексний підхід, який передбачає інтеграцію декількох механізмів захисту (наприклад, двофакторної автентифікації, криптографічного шифрування, захищеної передачі даних через SSL/TLS, контроль доступу та журналювання дій користувачів), створює найбільш ефективну архітектуру безпеки для користувацьких даних. Така стратегія дозволяє не лише зменшити вірогідність успішного зламу, а й створити стійку до внутрішніх і зовнішніх загроз інформаційну систему.

Одним із важливих досягнень виконаної роботи є практичне підтвердження того, що використання алгоритму AES256 у режимі EAX є виправданим вибором для симетричного шифрування користувацької інформації. Цей алгоритм забезпечує не лише надійне шифрування, а й вбудовану перевірку цілісності за допомогою автентифікаційного тегу, що мінімізує ризик непомітного втручання у дані. Додатково, впровадження TOTPалгоритму для створення одноразових кодів підтвердження автентифікації дозволяє реалізувати сучасну модель 2FA, яка

значно підвищує захищеність облікових записів навіть у разі компрометації основного пароля. Це особливо актуально у зв’язку зі зростанням соціальної інженерії, фішингових атак та витоків даних із публічних і приватних баз.

Результати роботи свідчать про доцільність побудови систем безпеки на основі модульного програмного підходу з чітко визначеними функціями кожного компонента. Така структура полегшує супровід, тестування, масштабування та адаптацію системи під конкретні потреби проєкту. Розроблені програмні компоненти є сумісними з більшістю сучасних вебфреймворків та систем управління користувачами, що відкриває широкі можливості для інтеграції розробленої моделі в реальні проєкти – як у сфері електронної комерції, так і в медичних, освітніх або правових сервісах.

Окремої уваги заслуговує потенціал масштабування запропонованого рішення. Застосовані алгоритми не є ресурсомісткими, що дозволяє ефективно працювати як у малих, так і в розподілених середовищах із великою кількістю запитів. Реалізація захисту на стороні серверної логіки у поєднанні з клієнтськими компонентами забезпечує можливість побудови багаторівневих архітектур безпеки з підтримкою API, мобільних застосунків та вебінтерфейсів. Отже, створена модель може бути масштабована до будьяких обсягів даних і числа користувачів за умови правильного адміністрування інфраструктури.

Узагальнюючи вищенаведене, можна дійти висновку, що захист інформації користувачів у цифровому просторі має будуватися на багатокомпонентному, гнучкому, прозорому для контролю та стійкому до загроз фундаменті. Запропонований у рамках дослідження підхід відповідає сучасним вимогам до інформаційної безпеки та є практично орієнтованим рішенням, здатним адаптуватися до нових викликів. Його подальший розвиток може включати інтеграцію систем штучного інтелекту для виявлення аномальної поведінки користувачів, адаптивну політику доступу та автоматичну реакцію на потенційні загрози, що дозволить вивести захист користувачів на ще вищий рівень.

ПЕРЕЛІК ПОСИЛАНЬ

1. Поняття комп’ютерних мереж. Історія розвитку комп’ютерних мереж. URL: https://surl.li/pdyqby
2. Топологія комп'ютерних мереж. URL: https://kremenetskyy.blogspot.com/2017/10/bl ogpost.html
3. НД ТЗІ 1.100507 Захист інформації на об’єктах інформаційної діяльності. Створення комплексу технічного захисту інформації. Основні положення. 2007-12-12 .Арк 1-7
4. Сенів М. М. Безпека програм та даних: навч. посібник / М.М. Сенів,В.С. Яковина. – Львів : Видавництво Львівської політехніки, 2015. – 256 с.
5. Горбенко І. Д. Гриненко Т. О. Захист інформації в інформаційнотелекомунікаційних системах: Навч. посібник. Ч.1. Криптографічний захист інформації Харків: ХНУРЕ, 2004 368 с.
6. Symmetric Key Algorithms survey. URL: https://www.tutorialspoint.com/symmetrickeyalgorithms
7. Asymmetric Key Algorithms survey. URL: https://www.techtarget.com/searchsecurity/definition/asymmetriccryptography
8. What is VPN. URL: https://www.kaspersky.com/resourcecenter/definitions/whatisavpn
9. VPN survey. URL: https://www.techtarget.com/searchnetworking/definition/virtualprivatenetwork
10. What is SSL. URL: https://www.cloudflare.com/learning/ssl/whatisssl/
11. Paar, C., & Pelzl, J. (2010). Understanding Cryptography: A Textbook for Students and Practitioners. Springer. URL: https://gnanavelrec.wordpress.com/wp-content/uploads/2019/06/2.understanding-cryptography-by-christof-paar-.pdf
12. Tanenbaum, A. S., & Wetherall, D. J. Computer Networks / під ред. A. S. Tanenbaum, D. J. Wetherall. – 5е вид. – Нова Джерсі: Prentice Hall, 2021. – С. 45–50, 550–555.

13. DSpace. (2020). Дослідження в області безпеки мереж. URL: https://dspace.nadpsu.edu.ua/handle/123456789/2841
14. Міністерство цифрової трансформації України. (2023). *Національна стратегія у сфері кібербезпеки України на 2021–2025 роки*. URL: https://thedigital.gov.ua
15. Державна служба спеціального зв'язку та захисту інформації України. (2022). *Щорічний звіт про стан кібербезпеки в Україні*. URL: https://cip.gov.ua
16. Стрижак, О. С., & Дерев'янко, В. І. (2021). Захист інформації в комп'ютерних системах. Збірник наукових праць НУ «Львівська політехніка», №2(132), 45–54.
17. NIST (National Institute of Standards and Technology). (2020). Zero Trust Architecture (Special Publication 800207). DOI: 10.6028/NIST.SP.800207 URL: https://csrc.nist.gov/pubs/sp/800/207/final
18. European Union Agency for Cybersecurity (ENISA). (2023). Cybersecurity Threat Landscape 2022. URL: https://www.enisa.europa.eu/publications/enisathreatlandscape2022
19. OWASP Foundation. (2023). *OWASP Top 10 – 2023: The Ten Most Critical Web Application Security Risks*. URL: https://owasp.org/wwwprojecttopten/
20. Arora, A., & Singh, R. (2021). *TwoFactor Authentication Mechanisms: A Comparative Study of TOTP and HOTP Algorithms*. *Journal of Cybersecurity and Privacy*, 1(2), 134–146. URL: https://doi.org/10.3390/jcp1020009
21. Ali, M., & Khan, S. (2020). *Modern Cryptographic Techniques: A Survey on AES and RSA Algorithms*. *International Journal of Computer Science and Network Security*, 20(8), 80–86.
22. Зозуля, А. І. (2022). Проблеми захисту персональних даних у цифрову епоху: виклики та перспективи. Інформаційна безпека та кіберзахист, №4(18), 22–29.
23. Vacca, J. R. (2022). Computer and Information Security Handbook (4th ed.). Academic Press. ISBN: 9780128192054. URL: https://shop.elsevier.com/books/computer-and-information-security-handbook-2-volume-set/vacca/978-0-443-13223-0

24. Microsoft. (2023). Security Best *Practices for Identity and Access Management*. URL: https://learn.microsoft.com/enus/security/compass/identityaccessbestpractices
25. Kaspersky Lab. (2024). *The State of Cybersecurity Report*. URL: https://www.kaspersky.com/resourcecenter
26. Cisco Systems. (2023). Cisco Annual Cybersecurity Report. URL: https://www.cisco.com/c/en/us/products/security/securityreports.html
27. ISO/IEC 27001:2022. Information Security, Cybersecurity and Privacy Protection – Information Security Management Systems – Requirements. International Organization for Standardization.URL: https://www.iso.org/standard/27001
28. Гудзь, М. І. (2021). Методи криптографічного захисту в умовах хмарних обчислень. Інформаційні *технології та комп'ютерна інженерія*, №1(47), 58-63.