КІБЕРБЕЗПЕКА: освіта, наука, техніка
CYBERSECURITY: EDUCATION, SCIENCE, TECHNIQUE
№ 4 (28), 2025
ISSN 2663 - 4023
DOI 10.28925/2663-4023.2025.28.786
УДК 004.056.5

Гулак Наталія Костянтинівна
кандидат технічних наук, доцент кафедри Кібербезпеки
Державний університет «Київський авіаційний інститут», Київ, Україна
ORCID ID: 0009-0000-9584-7113
nataliia.hulak@npp.kai.edu.ua

Ільєнко Анна Вадимівна
кандидат технічних наук, доцент, завідувач кафедри Кібербезпеки
Державний університет «Київський авіаційний інститут», Київ, Україна
ORCID ID: 0000-0001-8565-1117
anna.ilienko@npp.nau.edu.ua

Дубчак Олена Вікторівна
старший викладач кафедри Кібербезпеки
Державний університет «Київський авіаційний інститут», Київ, Україна
ORCID ID: 0000-0001-9739-3960
3915922@npp.kai.edu.ua

ПРАКТИЧНІ АСПЕКТИ ВИКОРИСТАННЯ КРИПТОГРАФІЧНИХ МЕТОДІВ ДЛЯ ЗАХИСТУ БАЗ ДАНИХ ВІД НЕСАНКЦІОНОВАНОГО ДОСТУПУ

Анотація. У статті проаналізовано сучасні кіберзагрози, спрямовані проти України в умовах гібридної війни, зокрема атаки, що здійснюються з боку рф. Аналітичну базу дослідження склали офіційні звіти Ради національної безпеки і оборони України, загальний звіт НАТО, а також звіт Державної служби спеціального зв'язку та захисту інформації України за 2024 рік. Здійснено систематизацію основних типів атак, виявлених у критичній інформаційній інфраструктурі, з акцентом на їхню спрямованість проти державних органів, об'єктів енергетики, транспорту та зв'язку. На основі виявлених загроз обґрунтовано необхідність посилення захисту інформаційних систем від несанкціонованого доступу шляхом впровадження криптографічних методів. Визначено доцільність застосування криптографічних механізмів для забезпечення конфіденційності, цілісності та автентичності даних у цифровому середовищі.У рамках дослідження розроблено концепцію програмного модуля, який використовує реляційну базу даних SQL для зберігання криптографічних ключів, сертифікатів та зашифрованих повідомлень. Модуль дозволяє здійснювати контрольоване управління доступом до даних, забезпечуючи багаторівневу авторизацію користувачів та захист від несанкціонованих змін.Проведене експериментальне тестування підтвердило ефективність обраних криптографічних методів та доцільність їх інтеграції в інформаційні системи. Отримані результати демонструють практичну цінність розробленого підходу для підвищення рівня кіберзахисту в умовах сучасних загроз.

Ключові слова: кібербезпека; криптографія; SQL; база даних; конфіденційність; інформаційна безпека; гібридна війна; хакерські атаки.

ВСТУП

Постановка проблеми. З розвитком цифрових технологій, захист інформації від несанкціонованого доступу стає критично важливим завданням під час повномасштабного вторгнення рф в Україну. Повномасштабне вторгнення рф в Україну стало потужним каталізатором для розвитку кіберзагроз. Війна в кіберпросторі — це поле протистояння, де тактики і технології змінюються щодня. Російські хакерські

угруповання як ті, що працюють у складі спецслужб рф, так і кіберзлочинці та хактивісти, постійно шукають слабкі місця в інформаційних системах України та розробляють нові методи атак як проти нас, так і проти всього цивілізованого світу [1].

У 2022 році ворог робив акцент на операціях знищення ІТ-інфраструктур організацій сектору критичної інфраструктури, а також отримання баз даних, списків. Ворожі хакери йшли туди, де були очевидні недоліки, вразливості і можливості, якими вони могли легко скористатися. У 2023 році їх стратегія поступово змістилася до закріплення і прихованого отримання інформації та використання кіберкомпоненту для отримання зворотного зв'язку про результати їхніх кінетичних уражень. У 2024 році спостерігається зміщення фокуса атак на все, що безпосередньо пов'язане з театром бойових дій та атаками на постачальників послуг, з метою якомога довше залишатися непомітними, утримувати присутність в системах, які мають зв'язок з війною та політикою. В табл. 1 наведено тенденція розподілення загальної кількості інциндентів за рівнем критичності [1].

Таблиця 1

Інциденти за рівнем критичності (2023–2024)

Інциденти за рівнем критичності	2023	2024	Зміна за період
Критичні	31	3	-90%
Високі	156	45	-71%
Середні	1264	1670	32%
Низькі	12	21	75%
Разом	1463	1739	19%

Представлена тенденція показує необхідність підвищення рівня захищеності інформаційно-комунікаційних систем та мереж. Криптографічні методи є одним з інструментів вирішення цього завдання.

Аналіз останніх досліджень і публікацій. Опрацювавши звіти РНБО України, Державної служби спеціального зв'язку та захисту інформації України, та загальний звіт НАТО за 2024 рік, було виявлено що найбільш поширені кіберзагрози для даних представляє несанкціонований доступ [1] – [3]. Ідея створення криптографічного програмного модуля який виконує створення підписаного довіреною особою або центром сертифікації сертифікат, що використовує структуру контейнера x.509 та зберігає це в форматі PKCS#12, а пароль до цього електронного документа зберігається в базі даних SQL у зашифрованому виді за допомогою симетричного алгоритму шифрування AES, була апробована на VII Міжнародній науково-практичній конференції «World educational trends: lifelong learning in the information society» [4].

Метою статті є практичне доведення доцільності використання криптографічних методів для додаткового рівня захисту баз даних від несанкціонованого доступу в інформаційних системах на базі розробленого криптографічного програмного модуля.

ТЕОРЕТИЧНІ ОСНОВИ ДОСЛІДЖЕННЯ

В даному розділі надано теоретичне обгрунтування використання криптографічних методів для захисту баз даних інформаційних систем від несанкціонованого доступу.

Криптографічні методи захисту є одними із головних інструментів для забезпечення конфіденційності, цілісності та аутентичності даних у сучасних

інформаційних системах. Вони використовуються для шифрування, цифрового підпису та інших механізмів захисту даних від несанкціонованого доступу. Буде розглянуто їхні особливості та підходи до забезпечення безпеки в інформаційних системах та мережах. До методів криптографії відносяться: симетричне шифрування; асиметричне шифрування; цифрові підписи та хешування; гібридні системи в криптографії.

1. Симетричне шифрування. Симетричне шифрування є одним з найпоширених методів криптографічного захисту даних. Прикладами симетричних алгоритмів шифрування є AES (Advanced Encryption Standard), 3DES, ChaCha, Salsa, Twofish, Blowfish і Serpent. Алгоритм AES буде використано у криптографічному модулі. У цьому типі шифрування один і той же ключ використовується для шифрування та розшифровки повідомлень (тобто, якщо хтось може надсилати зашифроване повідомлення, він також може розшифрувати), тобто у симетричному шифруванні відправник і отримувач повинні мати спільний секретний ключ, який використовується для шифрування вихідного повідомлення (plaintext) в зашифроване повідомлення (ciphertext) і для подальшого дешифрування його [5].

Формалізація роботи алгоритму AES.

У нас є дві сторони, які хочуть безпечно спілкуватися, і ми називаємо їх Аліса та Боб (для А та Б відповідно). Аліса хоче надіслати Бобу відкритий текст- P таким чином, щоб лише Боб міг прочитати його зміст. Вони домовилися про загальний секретний ключ, k, і вони використовують певний алгоритм, наприклад AES. Алгоритм шифрування — це певна функція E, приймаючи відкритий текст і ключ і виводячи зашифрований текст C [5]:

$$E(P, k) = C \quad (1)$$

де E — функція шифрування, P — відкритий текст, k — загальний секретний ключ, C — вихідной зашифрований текст повідомлення.

Алгоритм дешифрування, D бере зашифрований текст і ключ та повертає відкритий текст [5]:

$$D(C, k) = P \quad (2)$$

де D — функція дешифрування, C — вихідной зашифрований текст повідомлення, k— загальний секретний ключ, P — відкритий текст.

Використання одного ключа для шифрування і дешифрування робить симетричне шифрування швидким та ефективним для захисту великих обсягів даних, хоча існують певні виклики, пов’язані з управлінням ключами.

2. Асиметричне шифрування. Асиметричне шифрування, або криптографія з відкритим ключем є методом захисту інформації. Для шифрування і дешифрування використовуються різні ключі, такий підхід на відміну від симетричного шифрування базується на двох ключах: відкритий ключ (public key) використовується для шифрування даних; закритий ключ (private key) використовується для дешифрування. Розглянемо принцип асиметричного алгоритму шифрування на прикладі алгоритму RSA, що використовується в криптографічному модулі.

Робота алгоритму складається з чотирьох етапів: 1) генерація ключа; 2) розподіл ключа; 3) шифрування; 4) дешифрування.

Перший етап «генерація ключа» розбито на 5 кроків [6]:

Крок 1. Вибір двох простих чисел. Оберемо два простих великих числа p та q (просте число — це таке, що може ділитись без залишку лише на 1 та на себе). Числа p і q повинні триматися в таємниці, оскільки саме на них базується процес створення приватного та публічного ключа. Для більшої надійності p та q повинні бути: випадковими числами; великими; мати велику різницю.

Крок 2. Визначення модуля. Визначається число *n*, що використовується як модуль для відкритого і закритого ключа [6]:

$$n = p * q \quad (3)$$

де n — довжина ключа в бітах, p — просте число, q — просте число.

Довжина *n*, що виражена в бітах, і є довжиною ключа.

Крок 3. Застосування функції Ейлера. Далі необхідно визначити значення функції Ейлера від числа *n*, що має наступний вид:

$$\varphi = (p-1) \cdot (q-1) \quad (4)$$

φ — функція Ейлера, p — просте число, q — просте число.

Крок 4. Вибір відкритої експоненти**.** Після визначення значення функції Ейлера необхідно випадково обрати ціле число *e* таке, щоб $1 < e < \varphi(n)$. Число *e* (відкрита експонента) повинно бути взаємно простим до значення $\varphi(n)$. Занадто малі значення відкритої експоненти можуть послабити алгоритм RSA [6].

Крок 5. Находження приватної експоненти. Значення оберненого за модулем числа (приватної компоненти) можна визначити з допомогою розширеного алгоритму Евкліда.

Етап 2. Розподіл ключа. Після визначення всіх необхідних чисел, можна сформувати приватний та публічний ключ. Приватний ключ складається з пари *d* та *n*, а публічний з пари *e* та *n*. Число *d* повинно триматись у таємниці, оскільки воно використовується для дешифрування повідомлення. Числа *p, q* та $\varphi(n)$ також повинні триматись у таємниці, адже за їх допомогою можна визначити приватну експоненту *d*. Після визначення експоненти *d* ці числа можна одразу відкинути.

Публічний ключ пересилається надійним, але не обов’язково зашифрованим каналом зв’язку.

Етап 3. Шифрування здійснюється відкритим (публічним) ключем, який можна передавати по відкритим каналом зв’язку.

Етап 4. Дешифрування. Зашифрована інформація дешифрується за допомогою закритого ключа.

Асиметричне шифрування, або криптографія з відкритим ключем, пропонує значні переваги у порівнянні із симетричним шифруванням. Головною перевагою асиметричних алгоритмів є те, що для шифрування і дешифрування використовуються різні ключі — відкритий і закритий. Це дозволяє уникнути необхідності безпечної передачі приватного ключа, що суттєво знижує ризики компрометації [7].

Проте асиметричне шифрування має і свої недоліки, зокрема вищу обчислювальну складність і повільнішу швидкість операцій у порівнянні з симетричними методами. Симетричне шифрування, з іншого боку, є більш швидким і ефективним для обробки великих обсягів даних, оскільки використовує один і той же ключ для шифрування та дешифрування, що робить його менш ресурсоємним [7].

У результаті, асиметричні шифри зазвичай використовуються для захисту невеликих обсягів даних, таких як передача ключів або цифрові підписи, в той час як симетричні шифри більш ефективні для шифрування великих даних. Залежно від конкретних вимог до безпеки, швидкості і ресурсів, обирається той чи інший метод шифрування.

Однією з головних переваг криптографічних алгоритмів — це можливість створення гібридних систем для комплексної системи захисту інформації.

Гібридні системи поєднують переваги симетричного та асиметричного шифрування, а також електронного цифрового підпису (ЕЦП) для забезпечення високого рівня безпеки та продуктивності в обробці даних. Основна мета таких систем — використання сильних сторін кожного з методів для створення ефективного та безпечного механізму шифрування [8].

3. Криптографічні бібліотеки в Python надають розробникам потужні інструменти для забезпечення конфіденційності, цілісності та аутентичності даних у програмних системах. Вони використовуються для реалізації алгоритмів шифрування, генерації цифрових підписів, хешування даних та обміну ключами між користувачами або системами [9].

Python надає велику кількість криптографічних бібліотек, тому для виконання операцій шифрування, електронно-цифрового підпису, які використовуються для практичного здійснення мети статті, було проаналізовано криптографічні бібліотеки PyCryptodome, cryptography, PyCrypto, Fernet за наступними критеріями: підтримка криптографічних алгоритмів, інтерфейс, шифрування за замовчуванням, підтримка роботи з сертифікатами та ключами, підтримка ЕЦП, безпека, активна підтримка, можливість інтеграції з іншими бібліотеками, простота використання. На основі порівняльного аналізу було обрано бібліотеку cryptography.

Безпека та актуальність стандартів cryptography активно підтримується і регулярно оновлюється, що забезпечує використання новітніх і безпечних криптографічних стандартів. Модуль відповідає найвищим вимогам безпеки і активно інтегрує сучасні криптографічні алгоритми, які враховують нові уразливості та потреби захисту, також він активно використовує бібліотеку OpenSSL, що дозволяж йому забезпечити високий рівень безпеки та ефективності при виконанні криптографічних операцій [10].

4. Бази даних MySQL в криптографічних системах. У криптографічних системах бази даних є незамінним компонентом для організованого зберігання та доступу до сертифікатів і ключів. Вони забезпечують ефективний та структурований спосіб зберігати великі обсяги даних, інформацію і підтримують зручний та швидкий доступ, що є критичним для криптографічних модулів, де час доступу та безпека мають вирішальне значення.

MySQL — це потужна, відкрита реляційна система керування базами даних, що використовується для зберігання, обробки та доступу до великих обсягів структурованих даних. Вона побудована на мові SQL (Structured Query Language), що дозволяє виконувати складні запити для роботи з даними в базі. Завдяки своїй популярності, надійності та масштабованості MySQL широко використовується в різних типах додатків, від невеликих сайтів до великих корпоративних систем і хмарних додатків [11].

Для інтеграції з базою даних MySQL використовується бібліотека mysql-connector-python. Ця бібліотека розроблена для прямого підключення до MySQL з Python, що забезпечує простий і ефективний спосіб взаємодії з базою даних і дозволяє виконувати основні операції з базами даних, такі як встановлення з’єднання, виконання SQL-запитів, оновлення даних та закриття з’єднання, що робить її універсальним і гнучким інструментом для розробки додатків, яким потрібен доступ до даних. Mysql-connector-python забезпечує високу сумісність з Python та MySQL, підтримуючи ключові можливості для створення надійних і продуктивних додатків. Для розробника ця бібліотека особливо зручна завдяки зрозумілому синтаксису, який дозволяє швидко налаштувати підключення та працювати з даними без зайвої складності.

Однією з переваг Mysql-connector-python є можливість безпечного виконання запитів до бази даних за допомогою параметризованих SQL-запитів, що дає можливість уникнути SQL-ін’єкцій — однієї з найпоширеніших загроз для безпеки даних. Завдяки підтримці функцій MySQL та можливості масштабування, mysql-connector-python є надійним вибором для зберігання та обробки сертифікатів.

МЕТОДИКА ДОСЛІДЖЕННЯ

Для практичного доведення досягнення стабільного рівня захищеності баз даних сучасними методами криптографії було розроблено криптографічний модуль, який реалізує механізми шифрування, підпису даних, а також забезпечення аутентифікації користувачів через сертифікати, що дозволяє створити багаторівневу систему захисту інформації. Для досягнення цієї мети використовуються два основні криптографічні методи: симетричне шифрування (AES) для захисту даних та асиметричне шифрування (RSA) для забезпечення безпеки передачі сесійних ключів і підпису даних. Модуль також інтегрується з SQL базою даних для зберігання шифрованих сертифікатів, ключів і паролів, що дає змогу централізовано керувати доступом і запобігати несанкціонованим діям.

1. Процес створення сертифікатів є важливою складовою системи забезпечення безпеки, оскільки сертифікати гарантують аутентичність та захист даних при обміні ними між користувачами. Розглянемо етапи створення сертифікатів криптографічного модуля [12].

1.1. На першому етапі створення сертифіката генерується пара ключів — приватний і публічний. Це здійснюється за допомогою криптографічного алгоритму RSA. Приватний ключ використовується для підписання даних і має бути збережений в безпечному місці, а публічний ключ поширюється разом з сертифікатом і використовується для перевірки підписів та шифрування даних.

1.2. Створення самопідписаного сертифіката, який підписується власником сертифіката, а центром сертифікації (CA). Для створення самопідписаного сертифіката в системі використовується наступна процедура: формується структура сертифіката, в якій вказуються дані про власника сертифіката (наприклад, країна, організація, місце проживання), після цього цей сертифікат підписується приватним ключем власника. Самопідписаний сертифікат буде діяти до тих пір, поки не буде відомий і не підтверджений стороннім CA.

1.3. Створення сертифіката, підписаного CA. Центр сертифікації підписує цей сертифікат, використовуючи свій приватний ключ, що підтверджує його аутентичність. Це дозволяє забезпечити вищий рівень довіри, оскільки сертифікат підписаний авторитетним джерелом.

1.4. Форматування і збереження сертифікатів: після створення сертифікатів вони можуть бути збережені у форматі PKCS#12, що забезпечує захищене зберігання як сертифіката, так і приватного ключа. Формат PKCS#12 дозволяє зберігати сертифікати та ключі в одному файлі, якій зашифрований паролем.

1.5. Зберігання сертифікатів в базі даних: у розробленому модулі для захисту від несанкціонованого доступу сертифікати зберігаються в базі даних MySQL. База даних використовується для централізованого зберігання сертифікатів, що дозволяє забезпечити доступ до них з різних частин програми або з інших систем, централізовано керувати політиками доступу, швидко виконувати запити до сертифікатів.

Сертифікати відповідають міжнародним стандартам, таким як X.509, є сумісними з різними системами та протоколами. Це дозволяє легко інтегруватися з іншими безпечними мережами та забезпечити високу інтероперабельність, а також автоматизувати багато аспектів роботи, що знижує витрати часу і ресурси на обробку запитів безпеки. Для організації надійного захисту інформацйії у базах даних SQL, що дозволить створити безпечну інформаційну систему, модулю необхідно вирішити наступні завдання: забезпечити аутентифікацію користувачів за допомогою паролів, які зберігаються у шифрованому вигляді, та сертифікатів, що підтверджують особу

користувача; шифрування даних алгоритмом AES для запобігання доступу до приватної інформації; підпис даних приватним ключем для забезпечення їх цілісності та автентичності; захист сесійних ключів за алгоритмом RSA для безпечної передачі ключів між користувачами; інтеграція з SQL базою даних, що забезпечує централізоване зберігання сертифікатів, ключів і паролів, а також дозволяє здійснювати перевірку сертифікатів та доступу.

Розглянемо вирішення цих завдань за допомогою методів криптографії. Основні функції розробленого модуля можна розділити на наступні категорії за використанням криптографічних методів:

1. Шифрування даних: шифрування є функцією модуля, яка дозволяє забезпечити конфіденційність даних, зокрема файлів, що зберігаються в базі даних, або повідомлень, які передаються між користувачами. Шифрування виконується алгоритмом симетричного шифрування AES.

2. Підпис даних: забезпечення цілісності та аутентичності даних виконується за допомогою приватного ключа відправника, що дозволяє одержувачу перевірити аутентичніть даних. Підпис даних виконується асиметричнм алгоритмом шифрування RSA.

3. Сертифікація користувачів та перевірка сертифікатів: модуль реалізовує систему сертифікації, що дозволяє здійснювати аутентифікацію користувачів за допомогою сертифікатів. Кожен користувач має особистий сертифікат, підписаний центром сертифікації. Сертифікати містять 64 публічні ключі, що використовуються для шифрування сесійних ключів і перевірки підписів.

4. Зберігання та управління ключами: ключі зберігаються в зашифрованому вигляді в базі даних для забезпечення безпеки та контролю доступу до даних. Модуль використовує методи для безпечного зберігання приватних та публічних ключів, а також сертифікатів у форматі PKCS#12. Для захисту паролів та ключів використовується алгоритм шифрування AES, що забезпечує захист від несанкціонованого доступу. Ключі та сертифікати зберігаються в базі даних SQL, а це дозволяє централізовано управляти доступом та сертифікацією. Паролі для доступу до зашифрованих сертифікатів і ключів зберігаються в зашифрованому вигляді в базі даних.

5. Управління сесійними ключами: модуль також підтримує управління сесійними ключами, що використовуються для шифрування даних під час сеансів передачі.

Вище зазначені функції взаємодіють між собою для забезпечення комплексного захисту даних та доступу в системі, інтегруючи криптографічні алгоритми та методи управління ключами для створення надійної та безпечної інфраструктури. Всі завдання для організації надійного захисту інформації у базах даних SQL, що дозволить створити безпечну інформаційну систему, було вирішено.

РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

За результатами досліджень було розроблено криптографічний модуль, який складається з кількох ключових компонентів, що взаємодіють для забезпечення захисту даних від несанкціонованого доступу. Архітектура модуля побудована таким чином, щоб забезпечити гнучкість та зручність використання. Компоненти модуля включають в себе управління сертифікатами, шифрування даних, процедури підпису, перевірку аутентичності та управління ключами.

Для основних компонентів криптографічного модуля були розроблені алгоритми, а саме:

- Алгоритм класу KeyManager відповідає за управління ключами, зокрема за їх генерацію, зберігання та шифрування. Він забезпечує безпечне зберігання паролів і ключів за допомогою алгоритму AES який використовується для шифрування паролів користувачів. Крім того, KeyManager відповідає за розшифровку паролів для доступу до зашифрованих сертифікатів і ключів, що зберігаються в базі даних.
- Алгоритм взаємодії з базою даних MySQL (Database): цей компонент відповідає за зберігання сертифікатів та інших даних, що використовуються для шифрування і підписання. База даних реалізує інтеграцію з SQL-системою (MySQL), що дозволяє зберігати зашифровані сертифікати, публічні і приватні ключі, а також паролі користувачів у зашифрованому вигляді. Database також дозволяє ефективно керувати доступом та сертифікацією користувачів, що є важливим для забезпечення захисту від несанкціонованого доступу.
- CertificateManager (Менеджер сертифікатів): цей компонент відповідає за створення, підписання, зберігання та перевірку сертифікатів. CertificateManager працює з сертифікатами у форматі PKCS#12, що забезпечує їх підписання за допомогою центра сертифікації та зберігання в базі даних. Менеджер сертифікатів також підтримує процес перевірки сертифікатів для аутентифікації користувачів та підтвердження їх аутентичності.
- MessageSecurity (Безпека повідомлень): компонент для забезпечення шифрування та підписування повідомлень. Використовує як симетричне (AES), так і асиметричне (RSA) шифрування для забезпечення конфіденційності і цілісності переданих даних. MessageSecurity відповідає за генерацію сесійних ключів, шифрування повідомленя та підписування.
- Main (Головний компонент): основний компонент модуля, який інтегрує всі функціональні можливості, що надаються іншими компонентами, і надає зручний інтерфейс для взаємодії з користувачем. Всі операції з шифруванням, підписуванням, а також управлінням користувачами та їх сертифікатами виконуються через головний компонент. Він забезпечує зручне меню для адміністраторів та користувачів для виконання відповідних операцій, таких як генерація сертифікатів, підписування файлів.

Логіка роботи головного компоненту модуля показана на рис. 1.

Розроблений криптографічний програмний модуль має наукове значення, оскільки в процесі його створення були проаналізовані й впроваджені сучасні підходи до захисту даних. У рамках дослідження проведено аналіз алгоритмів AES і RSA, що є одними із основних сертифікованих криптографічних алгоритмів для забезпечення безпеки. Зокрема, було підтверджено ефективність використання AES для швидкого і надійного симетричного шифрування великих обсягів даних та RSA для захисту ключів і цифрового підпису. Науковим результатом цього дослідження є інтеграція цих алгоритмів у єдиній програмній архітектурі для забезпечення високого рівня безпеки при оптимальній продуктивності системи.

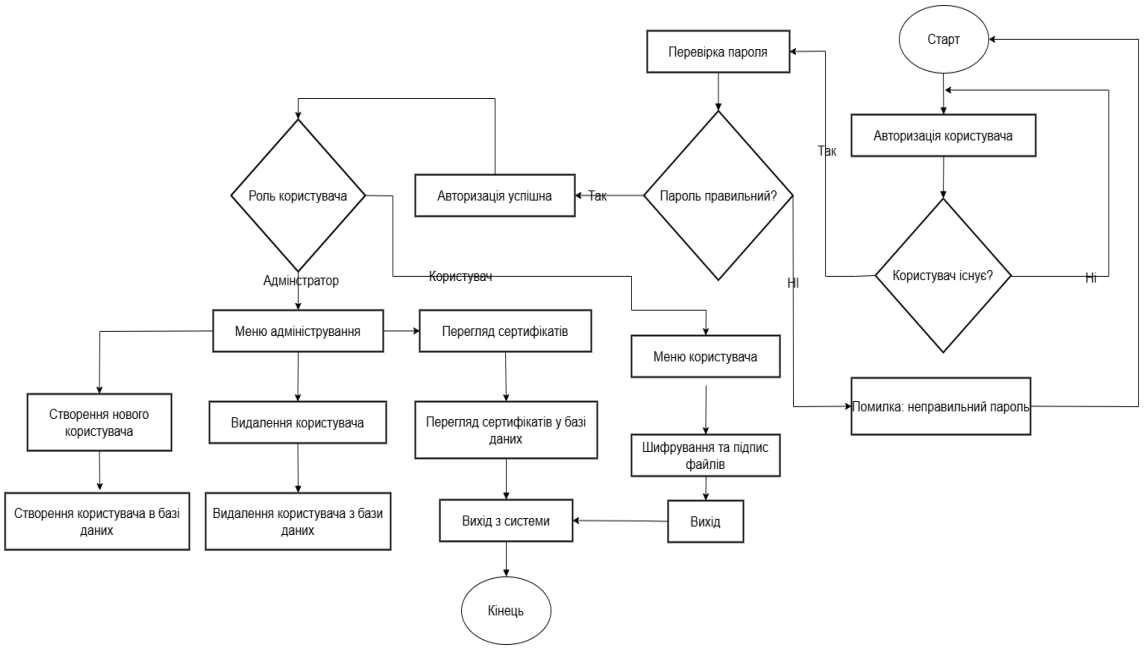

Рис. 1. Алгоритм роботи головного компоненту

Архітектура модуля побудована за принципом модульності, де кожен компонент виконує свою специфічну задачу, і всі компоненти взаємодіють між собою через чітко визначені інтерфейси. Логіка взаємодії компонентів модуля показана на рис. 2.

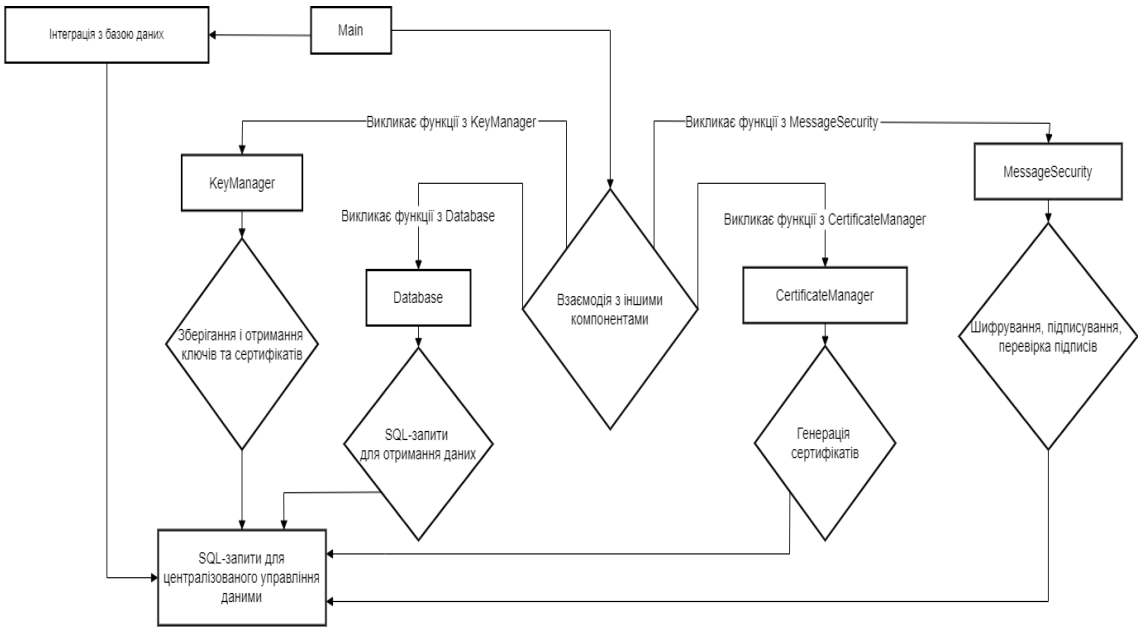

Рис. 2. Логіка взаємодії компонентів модуля

Для підтвердження результатів дослідження проведемо порівняння системи безпеки з розробленим модулем и стандартних систем безпеки по наступним критеріям (табл. 2): можливість використання симетричного і асиметричного шифрування, інтеграція з базою даних MySQL, менеджмент та зберігання сертифікатів відповідно стандартів PKCS, використання рольової моделі.

Таблиця 2

Порівнняння систем безпеки

Критерії	Система безпеки з розробленим модулем	Kleopatra PGP	XCA	KeyStore Explorer
Асиметричне шифрування	+	+	+	+
Симетричне шифрування	+	+	-	-
Інтеграція з базою даних MySQL	+	-	-	-
Менеджмент та зберігання сертифікатів відповідно стандартів PKCS	+	частково	+	+
Рольова модель	+	-	-	-

Окремо варто відзначити розробку механізму управління цифровими сертифікатами, що включає їх генерацію, зберігання в захищеному форматі PKCS#12, верифікацію підписів і інтеграцію з SQL-базою даних. Цей підхід дозволяє не лише забезпечити надійність криптографічного захисту, але й створює нові можливості для впровадження подібних рішень у практичній сфері, наприклад, у корпоративних системах управління доступом. Результатом роботи є вдосконалення процесів обробки сертифікатів і створення основи для подальших досліджень у сфері криптографії. Для практичного застосування доцільно впровадити модуль у системи з високими вимогами до безпеки, додати підтримку мобільних платформ і розширити функціонал за рахунок двофакторної автентифікації.

ВИСНОВКИ ТА ПЕРСПЕКТИВИ ПОДАЛЬШИХ ДОСЛІДЖЕНЬ

У статті всебічно обґрунтовано актуальність використання криптографічних методів захисту інформаційних систем, особливо в умовах гібридної війни та зростаючих кіберзагроз з боку рф. Результати аналітичного огляду загроз критичній інформаційній інфраструктурі України за останні роки вказують на необхідність посилення захисту даних, зокрема від несанкціонованого доступу, що є найбільш поширеним типом кіберінцидентів. Запропонований у роботі криптографічний програмний модуль реалізує сучасні підходи до шифрування, аутентифікації та управління ключами, поєднуючи переваги симетричних (AES) та асиметричних (RSA) алгоритмів. Його інтеграція з реляційною базою даних SQL та підтримка сертифікатів у форматі PKCS#12 дозволяє ефективно зберігати та обробляти ключі, сертифікати та інші чутливі дані. Особлива увага приділена питанням безпечного управління доступом до бази даних, багаторівневій авторизації та ролям користувачів, що є критичними аспектами інформаційної безпеки. Наукова новизна дослідження полягає у створенні архітектурно цілісного криптографічного модуля з модульною структурою, що може бути адаптований до різних інформаційних систем. Практична цінність роботи підтверджена експериментальним тестуванням і демонстрацією переваг запропонованого підходу порівняно з існуючими рішеннями. Таким чином, впровадження розробленого модуля може суттєво підвищити рівень безпеки інформаційних систем у державному та корпоративному секторах. Перспективами подальших досліджень є розширення функціоналу модуля через впровадження двофакторної автентифікації, підтримку мобільних платформ, а також використання механізмів машинного навчання для виявлення аномальної активності користувачів у системі.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Nakonechna, E. (n.d.). *Russian cyber operations*. https://docs.google.com/viewer?url=https://cip.gov.ua/services/cm/api/attachment/download?id=65897&embedded=true&a=bi
2. *National Security and Defense Council of Ukraine.* (2024). National Cyber Digest 2024. https://www.rnbo.gov.ua/files/2024/NATIONAL_CYBER_SCC/20240916/2024%2008%20Cyber%20digest_ENG.pdf
3. Davydiuk, V., & Potii, V. (2024). *National Cybersecurity Governance in Ukraine.* https://ccdcoe.org/uploads/2024/08/National-Cybersecurity-Governance_Ukraine_Davydiuk_Potii_2024.pdf
4. Rushchak, A. M., & Hulak, N. K. (2024). Cryptographic module for protection against unauthorized access. *VII International Scientific and Practical Conference "World Educational Trends: Lifelong Learning in the Information Society"*, 67–69.
5. *Symmetric encryption*. (n. d.). https://docs.google.com/viewer?url=https://cip.gov.ua/services/cm/api/attachment/download?id=65897&embedded=true&a=bi
6. Lytvynenko, V. (n .d.). *RSA encryption algorithm, types of attacks on it, and implementation in Python.* https://dou.ua/forums/topic/43026/
7. *Symmetric vs. Asymmetric Encryption: What's the Difference?* (n .d.). https://www.trentonsystems.com/en-gb/blog/symmetric-vs-asymmetric-encryption
8. *Basic Principles of Functioning of the Hybrid Cryptographic System.* (n .d.). https://ena.lpnu.ua:8443/server/api/core/bitstreams/2ef4a067-2029-4f8e-bad5-cb78c05714d0/content
9. *Python Standard Library*. (n .d.). https://docs.python.org/uk/3.13/library/index.html
10. *Cryptography Python module documentation*. (n .d.). https://cryptography.io/en/latest/
11. Oracle. (n. d.). *MySQL: Understanding What It Is and How It's Used*. https://www.oracle.com/mysql/what-is-mysql/
12. *ITU-T Recommendation X.509.* (2019). Information technology – Open Systems Interconnection – The Directory: Public-key and attribute certificate frameworks. https://www.itu.int/rec/T-REC-X.509-201910-I/en

КІБЕРБЕЗПЕКА: освіта, наука, техніка
CYBERSECURITY: EDUCATION, SCIENCE, TECHNIQUE
№ 4 (28), 2025
ISSN 2663 - 4023

Nataliia Hulak
Candidate of Technical Sciences, Associate of the Cybersecurity Department
State University "Kyiv Aviation Institute", Kyiv, Ukraine
ORCID ID: 0009-0000-9584-7113
nataliia.hulak@npp.kai.edu.ua

Anna Ilyenko
Candidate of Technical Sciences, Associate Professor, Head of the Cybersecurity Department
State University "Kyiv Aviation Institute", Kyiv, Ukraine
ORCID ID: 0000-0001-8565-1117
anna.ilienko@npp. kai.edu.ua

Olena Dubchak
Senior Lecturer of the Cybersecurity Department
State University "Kyiv Aviation Institute",Kyiv, Ukraine
ORCID ID: 0000-0001-9739-3960
3915922@npp.kai.edu.ua

PRACTICAL ASPECTS OF USING CRYPTOGRAPHIC METHODS TO PROTECT DATABASES FROM UNAUTHORIZED ACCESS

Abstract. The article analyzes current cyber threats targeting Ukraine in the context of hybrid warfare, particularly attacks conducted by the Russian Federation. The analytical foundation of the study is based on official reports from the National Security and Defense Council of Ukraine, the general NATO report, and the 2024 report of the State Service of Special Communications and Information Protection of Ukraine. The main types of attacks identified within critical information infrastructure are systematized, with a focus on those directed at governmental agencies, energy, transport, and communication sectors. Based on the identified threats, the necessity to enhance the protection of information systems against unauthorized access is substantiated through the implementation of cryptographic methods. The study highlights the relevance of applying cryptographic mechanisms to ensure the confidentiality, integrity, and authenticity of data in digital environments. A conceptual software module was developed using a relational SQL database platform to store cryptographic keys, certificates, and encrypted user messages. The module enables controlled data management, providing multilevel user authorization and protecting information from unauthorized modifications. Experimental testing confirmed the effectiveness of the selected cryptographic methods and demonstrated their practical applicability in information security systems. The obtained results underscore the value of the proposed approach for strengthening cybersecurity in the face of modern threats.

Keywords: cybersecurity; cryptography; SQL; database; confidentiality; information security; hybrid warfare; cyberattacks.

REFERENCES (TRANSLATED AND TRANSLITERATED)

13. Nakonechna, E. (n.d.). *Russian cyber operations*. https://docs.google.com/viewer?url=https://cip.gov.ua/services/cm/api/attachment/download?id=65897&embedded=true&a=bi
14. *National Security and Defense Council of Ukraine.* (2024). National Cyber Digest 2024. https://www.rnbo.gov.ua/files/2024/NATIONAL_CYBER_SCC/20240916/2024%2008%20Cyber%20digest_ENG.pdf
15. Davydiuk, V., & Potii, V. (2024). *National Cybersecurity Governance in Ukraine.* https://ccdcoe.org/uploads/2024/08/National-Cybersecurity-Governance_Ukraine_Davydiuk_Potii_2024.pdf
16. Rushchak, A. M., & Hulak, N. K. (2024). Cryptographic module for protection against unauthorized access. *VII International Scientific and Practical Conference "World Educational Trends: Lifelong Learning in the Information Society"*, 67–69.
17. *Symmetric encryption*. (n. d.). https://docs.google.com/viewer?url=https://cip.gov.ua/services/cm/api/attachment/download?id=65897&embedded=true&a=bi

18. Lytvynenko, V. (n .d.). *RSA encryption algorithm, types of attacks on it, and implementation in Python*. https://dou.ua/forums/topic/43026/
19. *Symmetric vs. Asymmetric Encryption: What's the Difference?* (n .d.). https://www.trentonsystems.com/en-gb/blog/symmetric-vs-asymmetric-encryption
20. *Basic Principles of Functioning of the Hybrid Cryptographic System.* (n .d.). https://ena.lpnu.ua:8443/server/api/core/bitstreams/2ef4a067-2029-4f8e-bad5-cb78c05714d0/content
21. *Python Standard Library*. (n .d.). https://docs.python.org/uk/3.13/library/index.html
22. *Cryptography Python module documentation*. (n .d.). https://cryptography.io/en/latest/
23. Oracle. (n. d.). *MySQL: Understanding What It Is and How It's Used*. https://www.oracle.com/mysql/what-is-mysql/
24. *ITU-T Recommendation X.509.* (2019). Information technology – Open Systems Interconnection – The Directory: Public-key and attribute certificate frameworks. https://www.itu.int/rec/T-REC-X.509-201910-I/en