Міністерство освіти і науки України

Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)

Кафедра Програмної інженерії
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти другий (магістерський)

Дослідження методів сучасних технологій шифрування для захисту клієнтно-орієнтованих додатків
(тема)

Виконав:
Здобувач 2 року навчання
групи ІПЗм-23-4
Максим КУДЛАЄНКО
(Власне ім'я, ПРІЗВИЩЕ)

Спеціальність 121 – Інженерія програмного забезпечення.
(код і повна назва спеціальності)
Тип програми освітньо-наукова

Керівник доц. Ірина ЛЕЩИНСЬКА
(посада, Власне ім'я, ПРІЗВИЩЕ)

Допускається до захисту
Зав. кафедри _____ Кирило СМЕЛЯКОВ
(підпис) (Власне ім'я, ПРІЗВИЩЕ)

2025 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук

Кафедра Програмної інженерії

Рівень вищої освіти другий (магістерський)

Спеціальність 121– Інженерія програмного забезпечення
(код і повна назва)

Тип програми освітньо-наукова програма

Освітня програма Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
«___»_____ 2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві Кудлаєнко Максиму Юрійовичу
(Прізвище, ім'я, по батькові)

1. Тема роботи Дослідження методів сучасних технологій шифрування для захисту клієнтно- орієнтованих додатків .

затверджена наказом по університету №290 Ст від 15.04.2025

2. Термін подання студентом роботи до екзаменаційної комісії 16.06.2025
3. Вихідні дані до проекту: Дослідити методи сучасних технологій шифрування, порівняти їх, підвести висновки
4. Перелік питань, які потрібно опрацювати в роботі: Аналіз предметної галузі, аналіз літературних та наукових джерел, аналіз методів вирішення поставленної задачі

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Отримання завдання	15.04.2025	*виконано*
2	Аналіз предметної галузі і постановка задачі	16.04.2025	*виконано*
3	Огляд й аналіз літературних, наукових джерел	27.04.2025	*виконано*
4	Огляд й аналіз існуючих методів шифрування для клієнтно-орієнтованих додатків	01.05.2025	*виконано*
5	Проектування графічного інтерфейсу застосунка	10.05.2025	*виконано*
6	Розробка архітектури застосунку	20.05.2025	*виконано*
7	Розробка програмного забезпечення для проведення дослідження	28.05.2025	*виконано*
8	Проведення експериментального дослідження	29.05.2025	*виконано*
9	Підготовка до апробації результатів дослідження. Публікація матеріалів	30.05.2025	*виконано*
10	Програмна реалізація	01.06.2025	*виконано*
11	Підготовка пояснювальної записки	02.06.2025	*виконано*
12	Підготовка презентації та доповіді	03.06.2025	*виконано*
13	Перевірка на плагіат	04.06.2025	*виконано*
14	Нормоконтроль	05.06.2025	*виконано*
15	Рецензування	05.06.2025	*виконано*
16	Попередній захист	06.06.2025	*виконано*
17	Занесення диплома в електронний архів	09.06.2025	*виконано*
18	Допуск до захисту у зав. кафедри	10.06.2025	*виконано*

Дата видачі завдання 15.04.2025р.

Студент _____ Максим КУДЛАЄНКО
(підпис)

Керівник роботи _____ доц. Ірина ЛЕЩИНСЬКА
(підпис) (посада, Власне ім’я, прізвище)

РЕФЕРАТ / ABSTRACT

Пояснювальна записка містить: 78 с., 8 рис., 20 джерел

КІБЕРБЕЗПЕКА, МЕТОДИ ШИФРУВАННЯ, ШИФРУВАННЯ, AES, KRYSTALS-Cyber, PRESENT, RSA.

Об’єктом дослідження є процеси забезпечення безпеки клієнтно-орієнтованих додатків за допомогою сучасних методів та технологій шифрування.

Предмет дослідження: сучасні методи та підходи до реалізації гібридних технологій шифрування для захисту клієнтно-орієнтованих додатків.

Методи дослідження та аналізу: теоретичний аналіз наукової літератури та сучасних стандартів у галузі шифрування, практичне тестування алгоритмів гібридного шифрування за допомогою програмних інструментів, порівняльний аналіз ефективності алгоритмів для різних сценаріїв використання у клієнтно-орієнтованих додатках.

Результатом роботи стане створення рекомендацій щодо використання гібридних алгоритмів шифрування для клієнтно- орієнтованих додатків, що дозволить підвищити їх рівень інформаційної безпеки. Практичне застосування цих рекомендацій сприятиме адаптації існуючих методів шифрування до потреб сучасних клієнтських систем.

ENCRYPTION, ENCRYPTION METHODS, CYBERSECURITY, AES, RSA, PRESENT, CRYSTALS-Kyber.

The object of the research is the processes of ensuring the security of client-oriented applications using modern encryption methods and technologies.

The subject of the research: modern methods and approaches to the implementation of hybrid encryption technologies for the protection of client- oriented applications.

Research and analysis methods: theoretical analysis of scientific literature and

modern standards in the field of encryption, practical testing of hybrid encryption algorithms using software tools, and comparative analysis of the effectiveness of algorithms for various use cases in client-oriented applications.

The result of the work will be the creation of recommendations for the use of hybrid encryption algorithms for client-oriented applications, which will enhance their level of information security. The practical application of these recommendations will facilitate the adaptation of existing encryption methods to the needs of modern client systems.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

SSL – Secure Sockets Layer, криптографічний протокол, який забезпечує встановлення безпечного з'єднання між клієнтом і сервером.

TLS – Transport Layer Security, криптографічний протокол, що надає можливості безпечної передачі даних в інтернеті для навігації, отримання пошти, спілкування, обміну файлами, тощо.

DES – Data Encryption Standard, симетричний алгоритм шифрування певних даних.

AES – Advanced Encryption Standard, симетричний алгоритм блочного шифрування.

RSA – абревіатура від прізвищ Rivest, Shamir та Adleman, криптографічний алгоритм з відкритим ключем, що базується на обчислювальній складності задачі факторизації великих цілих.

CRYSTALS-Kyber – механізм інкапсуляції ключів, розроблений для стійкості до криптоаналітичних атак за допомогою майбутніх потужних квантових комп'ютерів.

CRYSTALS-Dilithium – один із алгоритмів постквантової криптографії, заснований на задачах теорії ґрат.

PRESENT – алгоритм, сфера використання якого лежить в спеціальних приладах, на зразок RFID міток або мереж сенсорів.

Завідувачу кафедри
ПІ_____
(скорочена назва кафедри)
проф. Кирилу СМЕЛЯКОВУ
(вчене звання, власне ім'я, прізвище)

ЗАЯВА

щодо самостійності виконання кваліфікаційної роботи та можливості її публікації (та/або публікації анотації кваліфікаційної роботи) в електронному архіві відкритого доступу ElAr KhNURE

Я,

_____Кудлаєнко Максим Юрійович_____
(прізвище, ім'я, по батькові)

здобувачка вищої освіти на другому (магістерському) рівні вищої освіти академічної групи _____ІПЗм-23-4_____
кафедра _____програмної інженерії_____
(повна назва кафедри)

заявляю: моя кваліфікаційна робота на тему

Дослідження методів сучасних технологій шифрування для захисту клієнтно-орієнтованих додатків._____,
(назва роботи)

що буде представлена в екзаменаційну комісію для публічного захисту, виконана самостійно, в ній не містяться елементи плагіату і вона може бути опублікована в репозиторії "ElArKhNURE". Погоджуюся з авторським договором, відповідно до Положення про репозиторій ХНУРЕ "ElArKhNURE". Всі запозичення з друкованих та електронних джерел мають відповідні посилання.

Я ознайомлений (а) з вимогами академічної доброчесності, згідно з якими виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування дисциплінарних заходів.

Дата Підпис

ВСТУП

В сучасному світі, де інформація є ключовим ресурсом, питання забезпечення її конфіденційності та безпеки набувають особливого значення. Зростання обсягів даних, розширення використання хмарних технологій та розвиток інтернету створюють нові виклики для захисту інформації від несанкціонованого доступу. Технології шифрування є одним із найважливіших інструментів у вирішенні цих завдань, забезпечуючи захист даних шляхом їх перетворення у вигляд, непридатний для прочитання без спеціального ключа.

Актуальність теми зумовлена швидким розвитком сучасних технологій, появою нових видів загроз, таких як атаки квантових комп'ютерів, та потребою у підвищенні стійкості шифрувальних алгоритмів. У цьому контексті важливо дослідити сучасні підходи до шифрування даних, їх сильні та слабкі сторони, а також можливості їх адаптації до нових викликів. Крім того, зростає необхідність розробки ефективних методів шифрування для специфічних галузей, таких як клієнтно-орієнтовані додатки.

Метою роботи є дослідження методів сучасних технологій шифрування для забезпечення захисту клієнтно-орієнтованих додатків, оцінка їх ефективності у різних сценаріях застосування та розробка рекомендацій щодо їх використання для підвищення інформаційної безпеки. Огляд і класифікація сучасних алгоритмів шифрування (симетричних, асиметричних, квантових), дослідження ключових критеріїв оцінки алгоритмів, практичне тестування обраних алгоритмів у контексті клієнтно-орієнтованих додатків, а також розробка рекомендацій щодо вибору технологій шифрування для таких додатків.

Об’єктом дослідження є процеси забезпечення безпеки клієнтно-орієнтованих додатків за допомогою технологій шифрування.

Предмет дослідження: методи сучасних технологій шифрування для захисту клієнтно-орієнтованих додатків.

Методи дослідження та аналізу: теоретичний аналіз наукової літератури та стандартів у галузі шифрування, практичне тестування алгоритмів за допомогою програмних інструментів у контексті клієнтно-орієнтованих додатків,

порівняльний аналіз ефективності алгоритмів за ключовими критеріями.

Результатом роботи стане створення рекомендацій щодо вибору алгоритмів шифрування для клієнтно-орієнтованих додатків, що сприятиме підвищенню рівня їх інформаційної безпеки. Практичне застосування цих рекомендацій дозволить адаптувати існуючі методи шифрування до потреб таких додатків.

1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ

1.1 Огляд предметної галузі

В умовах стрімкого зростання обсягів даних, поширення хмарних обчислень і масового впровадження Інтернету речей, ефективні алгоритми шифрування стають незамінними для захисту від зловмисників. Для клієнтно-орієнтованих додатків, таких як мобільні додатки чи веб-сервіси, загроза втрати даних або їх компрометація може мати катастрофічні наслідки для кінцевих користувачів та компаній-розробників. Від банківських транзакцій до захисту персональних даних – шифрування є базовою технологією у кожній сфері.

Дослідження предметної галузі починається з вивчення основних технологій і підходів, які використовуються для забезпечення безпеки даних шляхом їх шифрування. У сучасному світі алгоритми шифрування є ключовим інструментом у забезпеченні інформаційної безпеки. Вони застосовуються у різноманітних галузях, включаючи банківські системи, мобільні додатки, хмарні сервіси, веб та багато інших.

Незважаючи на ефективність сучасних алгоритмів шифрування, вони мають низку обмежень. Однією з основних проблем є пошук оптимального балансу між безпекою, продуктивністю та енергоефективністю. Наприклад, алгоритми, які забезпечують високий рівень захисту, можуть бути менш придатними для пристроїв з обмеженими ресурсами, таких як мобільні телефони.

Крім того, розвиток квантових обчислень ставить під загрозу традиційні криптографічні методи. Квантові комп'ютери, за прогнозами, матимуть здатність розв'язувати математичні задачі, які лежать в основі сучасних алгоритмів, значно швидше, що може зробити деякі класичні алгоритми уразливими.

Сучасна криптографія активно розвивається у кількох напрямах. Перш за все, це розробка квантостійких алгоритмів, які є відповіддю на загрози з боку квантових обчислень. Наприклад, алгоритм CRYSTALS- Kyber [1] стає одним із перспективних рішень у цій сфері. Водночас, для пристроїв з обмеженими ресурсами розробляються легковагові алгоритми шифрування, що поєднують високу ефективність із низькими витратами енергії.

Іншим важливим трендом є інтеграція криптографії з клієнтно-орієнтованими сервісами, такими як мобільні застосунки або веб-сайти. Це забезпечує додатковий рівень безпеки при обробці та зберіганні великих обсягів даних.

Одним із ключових викликів є необхідність адаптації алгоритмів до різних сценаріїв застосування. Для клієнтно-орієнтованих додатків важливою є енергоефективність, тоді як для банківських систем на першому місці безпека. Крім того, постійне оновлення стандартів та нормативних вимог ускладнює впровадження нових рішень.

Сучасна криптографія є полем для впровадження численних інновацій. Розвиток квантового шифрування, який ґрунтується на принципах квантової фізики, відкриває можливості для створення принципово нових методів забезпечення безпеки. Також активно розробляються гібридні підходи, які поєднують сильні сторони як симетричних, так і асиметричних алгоритмів, забезпечуючи при цьому адаптивність до специфічних сценаріїв, зокрема для клієнтно-орієнтованих додатків.

1.2 Існуючі підходи шифрування

Сучасні технології шифрування можна поділити на дві основні категорії [2]: симетричне та асиметричне шифрування. Симетричні алгоритми характеризуються використанням одного і того ж ключа для шифрування і дешифрування даних. Ці алгоритми забезпечують високу швидкість обробки даних та ефективність, що робить їх ідеальними для задач реального часу, таких як передача потокових даних або зберігання великих обсягів інформації. Основним викликом для симетричних алгоритмів є забезпечення безпечного обміну ключами між сторонами. Втрата або компрометація ключа може призвести до втрати безпеки всієї системи.

Асиметричні алгоритми використовують пару ключів – відкритий та закритий. Вони забезпечують високий рівень безпеки завдяки складним математичним операціям, що ускладнюють злому. Асиметричні методи ідеально

підходять для шифрування електронних підписів, автентифікації і безпечного обміну ключами. Однак, такі алгоритми є обчислювально складними і вимагають значних ресурсів, що може бути обмеженням у певних сценаріях застосування, наприклад, для пристроїв з низькою потужністю.

Переваги симетричних алгоритмів:

– висока швидкість роботи;
– низькі вимоги до обчислювальних ресурсів;
– придатність для роботи з великими обсягами даних.

Недоліки симетричних алгоритмів:

– потреба в безпечному обміні ключами;
– ризик компрометації ключа.

Переваги асиметричних алгоритмів:

– високий рівень безпеки;
– використання пари ключів спрощує обмін інформацією між сторонами;
– ідеальне рішення для електронних підписів та автентифікації.

Недоліки асиметричних алгоритмів:

– значні обчислювальні витрати;
– менша швидкість у порівнянні з симетричними методами.

Можемо зробити висновок, що вибір між симетричними та асиметричними алгоритмами залежить від специфіки задачі. Для клієнтно-орієнтованих додатків симетричні алгоритми підходять для швидкого шифрування великих обсягів даних у реальному часі, тоді як асиметричні забезпечують надійний обмін ключами та високий рівень безпеки для критичних додатків, таких як електронна комерція чи захист конфіденційних комунікацій.

1.3 Гібридні методи

Гібридні підходи до шифрування поєднують сильні сторони симетричних та асиметричних алгоритмів, забезпечуючи ефективність і високий рівень безпеки [3]. Основна ідея полягає в тому, що асиметричне шифрування використовується для безпечного обміну ключами, а симетричне – для подальшого шифрування

основних даних. Наприклад, у протоколах SSL / TLS асиметричні алгоритми відповідають за встановлення з'єднання, після чого передається симетричний ключ для шифрування даних сесії.

Переваги гібридних методів:

- висока ефективність завдяки використанню симетричного шифрування для великих обсягів даних;
- безпечний обмін ключами завдяки асиметричним алгоритмам;
- гнучкість у застосуванні для різних сценаріїв.

Недоліки гібридних методів:

- складність реалізації, що вимагає синхронізації двох типів алгоритмів;
- потенційна залежність від безпеки обох підходів, що може збільшити вразливість системи.

Гібридні методи є стандартом у багатьох сучасних системах, включаючи хмарні сервіси та клієнтно-орієнтовані додатки, оскільки вони дозволяють досягти оптимального балансу між продуктивністю та безпекою.

2 ОГЛЯД Й АНАЛІЗ ЛІТЕРАТУРНИХ, НАУКОВИХ ДЖЕРЕЛ

2.1 Базові теоретичні джерела

Класичні роботи з теорії криптографії закладають фундаментальні основи шифрування, які й сьогодні використовуються при розробці сучасних алгоритмів і протоколів захисту інформації. Одним із найвідоміших криптографів є Клод Шеннон, якого часто називають «батьком сучасної криптографії» та теорії інформації [4]. Його новаторські ідеї, зокрема введення поняття ентропії як міри невизначеності або випадковості, стали базисом для оцінки стійкості криптографічних систем. Ентропія ключа визначає його якість, адже чим більша випадковість і непередбачуваність ключа, тим важче зловмиснику підібрати його навіть із використанням сучасних обчислювальних ресурсів.

У своїх роботах Шеннон заклав концепції, без яких неможливо уявити жодну криптографічну систему. Він ввів поняття ідеального шифру, при якому ентропія ключа має бути не меншою за ентропію самого повідомлення. Це означає, що для абсолютної безпеки (так званого одноразового блокнота) кожен біт ключа повинен нести максимум інформації і використовуватися лише один раз. Такий підхід на практиці рідко застосовується через складність генерації та зберігання ключів необхідної довжини, проте він залишається еталоном криптографічної безпеки

Важливими є також поняття плутанини та дифузії, запропоновані Шенноном. Плутанина передбачає, що зв'язок між відкритим текстом, ключем і шифротекстом має бути максимально складним і заплутаним. Це значно ускладнює для зловмисника завдання визначити ключ або знайти закономірності в шифротексті. Дифузія означає, що зміна одного біта відкритого тексту або ключа повинна призводити до значних змін у шифротексті, розсіюючи вплив окремих бітів по всьому результату. Ці принципи стали основою для побудови практично всіх сучасних блочних алгоритмів шифрування, таких як DES, AES, PRESENT та інших.

Крім робіт Шеннона, значний внесок у розвиток криптографії зробили й інші вчені й інженери. У 1970-х роках сталася справжня революція у цій галузі

завдяки появі асиметричної криптографії. Одним із ключових досягнень став алгоритм RSA [5], що заснований на складності факторизації великих чисел. Ця система дала змогу створювати пари відкритих і закритих ключів, тим самим значно спростивши обмін зашифрованими повідомленнями й аутентифікацію користувачів. Важливо, що RSA дозволяє не лише шифрувати дані, а й створювати цифрові підписи, що забезпечують цілісність і автентичність повідомлень.

Близько в той самий час був розроблений протокол Діффі – Геллмана, який уперше дозволив двом сторонам безпечно обмінюватися ключами навіть через повністю відкритий канал зв'язку. Його стійкість ґрунтується на складності обчислення дискретного логарифма, що стало значним кроком уперед у розвитку теорії криптографії й практичного захисту інформації.

У подальшому з розвитком обчислювальної техніки та зростанням загроз з'явилися нові напрямки криптографічних досліджень. Зокрема, останні десятиліття привели до активного розвитку постквантової криптографії, яка орієнтована на захист даних від потенційних атак квантових комп'ютерів. Такі алгоритми базуються на складності задач, пов'язаних із решітками, багаточленами та іншими складними математичними структурами, для яких поки не знайдено ефективних квантових алгоритмів злому. Їхня розробка стала можливою завдяки фундаментальним знанням, закладеним попередніми поколіннями вчених.

Окремо варто зазначити роль математичних основ криптографії, зокрема теорії чисел, лінійної алгебри та абстрактної алгебри, які лежать в основі сучасних криптографічних систем. Наприклад, для алгоритмів DES [6] і AES [7], що стали першими стандартизованими симетричними методами шифрування, активно використовуються операції над скінченними полями, побудови на основі матриць і багаточленів. Ці алгоритми досі є основою захисту даних у багатьох сферах: від банківських транзакцій до захисту передавання даних у клієнтно-орієнтованих додатках, де особливо важливі швидкодія та енергоефективність.

Сучасна криптографія також розширюється завдяки впровадженню

гомоморфного шифрування, яке дозволяє обчислювати над зашифрованими даними без їхнього розшифрування. Цей напрям має великий потенціал для хмарних обчислень і розподілених систем, де важливо забезпечити конфіденційність навіть під час обробки даних.

Таким чином, класичні праці, починаючи з робіт Клода Шеннона й до винаходу RSA та інших систем, заклали міцний науковий фундамент, на якому базується сучасна криптографія. Вони забезпечили підґрунтя для появи нових технологій захисту інформації, що відповідають сучасним викликам цифрової безпеки.

2.2 Сучасні дослідження

Сучасна наукова література акцентує увагу на вирішенні актуальних викликів у криптографії, особливо у контексті захисту даних у клієнтно-орієнтованих додатках. Ці додатки охоплюють мобільні застосунки, веб-сервіси, системи електронної комерції, фінансові платформи, а також пристрої Інтернету речей.

Основні напрями включають:

– квантостійка криптографія: у зв’язку з активним розвитком квантових обчислень традиційні криптографічні схеми стають потенційно вразливими, оскільки квантові комп’ютери здатні ефективно розв’язувати завдання факторизації великих чисел і обчислення дискретного логарифма. На цьому тлі особливої уваги набувають квантостійкі алгоритми, які ґрунтуються на складних для квантових атак математичних проблемах, таких як задачі на решітках, кодах або ізогенії еліптичних кривих. Наприклад, алгоритми CRYSTALS-Kyber і CRYSTALS-Dilithium активно розглядаються міжнародними організаціями, зокрема NIST, у процесі стандартизації постквантових криптографічних рішень [8]. Для клієнтно-орієнтованих додатків ці алгоритми є перспективними для забезпечення довготривалого захисту конфіденційних даних, адже вони враховують ймовірність появи

квантових атак у майбутньому;

– легковагові алгоритми: у сфері мобільних технологій, сенсорних мереж та Інтернету речей надзвичайно важливо забезпечити належний рівень безпеки без значного навантаження на обчислювальні та енергетичні ресурси пристрою. Для таких сценаріїв ведуться активні дослідження та розробка легковагових криптографічних алгоритмів. Наприклад, алгоритми типу PRESENT створені спеціально для мікроконтролерів, носимих пристроїв і сенсорних вузлів. Вони характеризуються компактністю реалізації, низьким енергоспоживанням і можливістю швидкої обробки даних у реальному часі. Це дозволяє використовувати їх у розумних будинках, промисловому обладнанні та медичних пристроях без втрати рівня захист;

– інтеграція з хмарними сервісами: зі зростанням популярності хмарних технологій усе більш актуальним стає завдання забезпечення конфіденційності даних, які передаються і обробляються у хмарних середовищах. Одним із найперспективніших напрямів у цьому контексті є застосування гомоморфного шифрування, яке дозволяє виконувати обчислення безпосередньо над зашифрованими даними, не розкриваючи їх змісту. Це відкриває нові можливості для клієнтно-орієнтованих додатків, оскільки забезпечує додатковий рівень захисту навіть у разі компрометації хмарної інфраструктури. Гомоморфне шифрування використовується, зокрема, у фінансових і медичних сервісах для збереження конфіденційності під час аналітики великих масивів чутливих даних.

У результаті можна стверджувати, що сучасна криптографія динамічно розвивається у відповідь на нові виклики і потреби цифрового суспільства. Інтеграція новітніх криптографічних рішень у клієнтно-орієнтовані додатки є важливою умовою забезпечення безпеки, конфіденційності та надійності інформаційних систем.

2.3 Практичні дослідження

Практична сфера досліджень охоплює впровадження криптографічних рішень у клієнтно-орієнтованих додатках, зокрема мобільних застосунках, веб-сервісах, фінансових платформах, а також у рішеннях для Інтернету речей. Сучасні інформаційно-комунікаційні технології вимагають інтеграції ефективних механізмів шифрування для забезпечення конфіденційності, цілісності та доступності даних. У цій сфері можна виділити кілька ключових напрямів:

- у мобільних додатках активно використовуються симетричні та асиметричні алгоритми для захисту конфіденційної інформації під час мобільних платежів, а також для автентифікації користувачів і запобігання підробці транзакцій. Застосування AES дозволяє ефективно шифрувати великі обсяги даних, тоді як RSA використовується для безпечного обміну ключами та цифрових підписів;
- для захисту інформації під час її передавання у веб-сервісах застосовуються протоколи SSL / TLS [9], які базуються на комбінації симетричної та асиметричної криптографії. Вони забезпечують захищене з'єднання між клієнтом і сервером, що є обов'язковою вимогою для додатків електронної комерції, онлайн-банкінгу, хмарних сервісів і корпоративних порталів. Використання таких протоколів запобігає атакам типу «людина посередині» та забезпечує високий рівень довіри до сервісу;
- у сенсорних мережах інтернету досліджуються й впроваджуються легковагові криптографічні алгоритми, що дозволяють зберігати баланс між рівнем безпеки та обмеженими обчислювальними і енергетичними ресурсами пристроїв. Це особливо важливо для смарт-датчиків, медичних імплантів, трекерів та інших пристроїв, де обсяг пам'яті та обчислювальна потужність є мінімальними.

Практичні дослідження також підкреслюють важливість оптимізації криптографічних алгоритмів з метою забезпечення швидкої обробки даних у

реальному часі. Це критично важливо для інтерактивних клієнтських додатків, таких як системи миттєвих повідомлень, відеоконференцій, ігрових платформ, де навіть затримки в мілісекунди можуть впливати на якість обслуговування користувачів. У цьому контексті важливою є інтеграція апаратних прискорювачів шифрування та використання оптимізованих бібліотек криптографічних примітивів.

Крім того, сучасні практичні дослідження охоплюють інтеграцію криптографічних рішень у хмарні обчислювальні середовища, де значну увагу приділяють питанням безпеки даних при зберіганні та обробці у віддалених дата-центрах. Такі підходи часто включають використання гомоморфного шифрування та криптографії на основі решіток як перспективних методів для постквантового захисту інформації.

2.4 Висновок з огляду джерел

Аналіз літератури показує, що сучасні дослідження у сфері криптографії зосереджені на адаптації алгоритмів до нових глобальних викликів, серед яких особливе значення мають поява квантових обчислень, розширення мережі Інтернету речей (IoT), активний розвиток хмарних технологій, а також стрімке збільшення обсягів передавання, обробки та зберігання даних. Ці фактори суттєво змінюють вимоги до систем захисту інформації, оскільки традиційні алгоритми й підходи вже не завжди здатні забезпечити належний рівень безпеки й ефективності в умовах сучасної кіберінфраструктури. Поява квантових комп’ютерів, зокрема, ставить під загрозу стійкість класичних криптографічних систем, що базуються на складності факторизації великих чисел чи обчислення дискретного логарифма. Це спонукає до активного розвитку постквантових алгоритмів шифрування, які спираються на математичні задачі, для яких поки не відомі ефективні квантові алгоритми розв’язання.

Сучасна криптографія активно розвивається у відповідь на нові загрози й технологічні виклики, а її досягнення відіграють ключову роль у створенні захищених клієнтно-орієнтованих додатків нового покоління. Ці додатки мають

відповідати не лише вимогам безпеки, а й очікуванням користувачів щодо зручності, швидкості роботи й сумісності з різними платформами. Відтак, криптографічні технології стають основою цифрової довіри в сучасному суспільстві й важливим чинником розвитку економіки на основі даних.

3 ПОСТАНОВКА ЗАДАЧІ

Потрібно реалізувати і порівняти продуктивність різних алгоритмів шифрування для клієнтно-орієнтованих додатків.

Вибір методів дослідження ґрунтується на необхідності комплексного аналізу продуктивності, адаптивності та безпеки сучасних криптографічних алгоритмів у реальних умовах використання. Це є критично важливим у світлі зростаючих вимог до інформаційної безпеки, які постійно висуваються у зв'язку зі збільшенням кількості кіберзагроз та розвитком нових технологій, таких як квантові обчислення. Основними методами будуть теоретичний аналіз літературних джерел (включаючи наукові статті, стандарти, патенти, аналітичні звіти та офіційну документацію), експериментальне тестування алгоритмів у контексті клієнтно-орієнтованих додатків і порівняльний аналіз отриманих результатів.

У дослідженні використовуватимуться як симетричні, так і асиметричні алгоритми що є найбільш поширеними у сучасних системах безпеки, а також легковагові методи, оптимізовані для пристроїв з обмеженими ресурсами.

Програмні рішення включатимуть мову програмування Python [10] для реалізації алгоритмів, бібліотека PyCryptodome [11] для тестування криптографічних операцій.

Проект враховує певні обмеження. Наприклад, дослідження зосереджується на алгоритмах, що мають найбільше практичне значення для клієнтно-орієнтованих додатків, таких як AES, RSA, CRYSTALS- Kyber, а також легковагових методах. Результати дослідження матимуть практичне значення для таких областей, як мобільні додатки, веб-сервіси сенсорні мережі та інші клієнтно-орієнтовані системи, де критично важливі швидкість, безпека та зручність використання.

Для виконання дослідження необхідно залучити такі ресурси:

- комп'ютерне обладнання з достатньою обчислювальною потужністю для моделювання алгоритмів та аналізу великих обсягів даних;

- наукові джерела для обґрунтування вибору алгоритмів, а також офіційна документація бібліотек і платформ;
- інструменти для моделювання та аналізу, такі як Python з бібліотекою PyCryptodome, а також засоби візуалізації даних для створення звітів.

Кінцевим результатом стане розробка практичних рекомендацій для інтеграції сучасних криптографічних методів у клієнтно-орієнтовані додатки. Основні напрями цих рекомендацій включатимуть:

- вибір оптимальних алгоритмів залежно від конкретного сценарію використання;
- забезпечення стійкості систем до сучасних загроз, зокрема квантових атак;
- оптимізація існуючих методів з урахуванням обмежених ресурсів пристроїв.

Дослідження сприятиме підвищенню рівня інформаційної безпеки, захисту конфіденційності даних користувачів та забезпеченню довготривалої стійкості криптографічних рішень у мінливому технологічному середовищі.

4 ТЕОРЕТИЧНЕ ДОСЛІДЖЕННЯ

4.1 Огляд існуючих методів і технологій шифрування

4.1.1 Симетричні алгоритми

Симетричне шифрування є одним із найбільш поширених способів захисту даних у сучасних інформаційних системах. Його суть полягає в тому, що для процесів шифрування та дешифрування використовується один і той самий секретний ключ, яким мають володіти як відправник, так і одержувач інформації. Найбільш відомим прикладом симетричного алгоритму є AES (Advanced Encryption Standard), який на сьогоднішній день вважається стандартом у багатьох країнах і використовується в різноманітних сферах – від захисту передавання даних у мережах до шифрування файлів і дисків.

Основною перевагою симетричних алгоритмів є їхня висока продуктивність. Завдяки низьким обчислювальним витратам вони дозволяють шифрувати та дешифрувати великі обсяги даних із мінімальними затримками, що робить їх особливо ефективними для використання в реальному часі. Наприклад, такі алгоритми часто застосовуються для захисту потокового відео чи аудіо, у хмарних сервісах, а також у мобільних додатках, де важливо забезпечити швидку роботу при обмежених апаратних ресурсах.

Проте, попри всі переваги симетричного шифрування, існує й один важливий недолік – це проблема безпечної передачі ключа. Оскільки і відправник, і одержувач повинні мати однаковий ключ, необхідно якось передати цей ключ без ризику його перехоплення зловмисником. Якщо ключ буде скомпрометовано, це поставить під загрозу всю систему захисту, оскільки зловмисник отримає можливість розшифрувати всі зашифровані дані. Саме тому на практиці симетричне шифрування часто комбінують з асиметричними алгоритмами (наприклад, RSA чи Kyber), які використовуються для безпечного обміну ключами. Такий підхід дозволяє поєднати переваги обох типів шифрування й забезпечити як високу швидкість обробки даних, так і надійний захист ключів від несанкціонованого доступу.

Симетричні алгоритми розділяються на блочні та потокові методи шифрування.

4.1.1.1 Блочний метод шифрування

AES – приклад алгоритма з блочним методом шифрування, який працює з блоками фіксованого розміру. Щоб забезпечити обробку повідомлень будь-якої довжини та підвищити стійкість до атак, використовуються різні режими роботи. Вони визначають, як саме обробляються блоки даних і як вони взаємодіють під час шифрування й дешифрування.

Найпростішим режимом є режим незалежного шифрування блоків, який називається ECB (Electronic Codebook). У цьому режимі кожен блок повідомлення шифрується окремо і незалежно від інших блоків. Такий підхід забезпечує простоту та швидкість виконання операцій, проте він є вразливим з точки зору безпеки. Якщо в повідомленні зустрічаються однакові блоки даних, вони зашифровуються однаково. Це дозволяє зловмиснику побачити повторювані структури у зашифрованому тексті, що значно спрощує спроби злому. Через ці недоліки режим ECB не рекомендується для використання у захищених системах.

Більш надійним є режим ланцюгового зв’язку блоків, який відомий як CBC (Cipher Block Chaining). У цьому режимі перед шифруванням кожного блоку його поєднують із результатом шифрування попереднього блоку. Для першого блоку застосовується випадкове початкове значення, яке називається вектором ініціалізації. Завдяки такому підходу навіть однакові блоки вихідного повідомлення після шифрування виглядають по-різному, що ускладнює аналіз шифротексту. Водночас режим CBC потребує обережного використання вектора ініціалізації: він повинен бути унікальним для кожної операції шифрування, щоб не знизити рівень захисту даних.

Режим зворотного зв’язку за шифротекстом відомий як CFB (Cipher Feedback). Його особливістю є можливість обробляти дані не лише блоками, а й меншими частинами, наприклад, байтами або окремими бітами. У цьому режимі попередній зашифрований блок використовується для створення даних, які

поєднуються з початковим текстом. Це дозволяє застосовувати шифрування для потокових даних і забезпечує гнучкість під час передавання інформації. Недоліком є те, що помилка під час передачі одного символу може призвести до неправильного дешифрування наступного символу.

Існує також режим зворотного зв’язку за вихідним потоком, відомий як OFB (Output Feedback). У цьому випадку генерується псевдовипадкова послідовність, яка поєднується з початковими даними. Головною перевагою цього режиму є стійкість до помилок під час передачі, оскільки пошкодження одного блока не впливає на інші частини дешифрованого повідомлення. Однак необхідно уважно стежити за тим, щоб початкове значення не повторювалося при різних операціях шифрування, інакше це може призвести до розкриття даних.

Сучасним та високопродуктивним варіантом є режим лічильника, який називається CTR (Counter). У цьому режимі для кожного блоку даних формується спеціальна послідовність чисел (лічильник), яка шифрується, а результат поєднується з початковими даними. Особливістю режиму CTR є можливість паралельної обробки блоків, що значно підвищує швидкість шифрування і дешифрування. Важливою умовою безпеки є унікальність лічильника для кожного блоку, адже повторення значень у поєднанні з одним і тим самим ключем створює загрозу розкриття інформації.

Таким чином, кожен із розглянутих режимів має свої переваги й недоліки, а вибір конкретного варіанту залежить від завдань системи, в якій застосовується шифрування. Режими CBC та CTR є найпоширенішими в сучасних системах, оскільки вони поєднують високу стійкість до атак і достатню швидкодію. Режим ECB практично не використовується для захисту важливої інформації через свої суттєві обмеження.

4.1.1.2 Потокові методи шифрування

Основною ідеєю потокового шифру є генерування псевдовипадкової послідовності бітів, яка використовується для шифрування повідомлення. Ця послідовність, що залежить від ключа шифрування, об'єднується з відкритим

текстом за допомогою побітової операції додавання за модулем два (XOR). У результаті формується шифротекст, який складно розшифрувати без знання ключа й точного стану генератора псевдовипадкових даних.

Потокові методи шифрування мають ряд переваг, що робить їх привабливими для багатьох практичних застосувань. По-перше, вони забезпечують високу швидкість обробки даних, оскільки не потребують накопичення блоків перед шифруванням. По-друге, потокові шифри добре підходять для захисту даних, що передаються в реальному часі, таких як голос, відео або телеметрія. По-третє, вони дозволяють ефективно працювати в умовах обмежених обчислювальних ресурсів, що особливо важливо для вбудованих систем і пристроїв Інтернету речей.

Водночас потокові методи шифрування потребують обережного застосування. Використання однакової послідовності псевдовипадкових бітів для різних повідомлень може призвести до компрометації даних. Тому важливо забезпечувати унікальність початкових параметрів генератора для кожної сесії шифрування. Крім того, потокові шифри чутливі до втрати синхронізації під час передавання даних, оскільки будь-яка помилка у потоці може призвести до неправильного дешифрування подальшої частини повідомлення.

4.1.1.3 Алгоритм шифрування AES

Шифрування починається з того, що повідомлення розбивається на блоки однакового розміру. Якщо останній блок виявляється коротшим, ніж потрібно, він доповнюється службовими байтами для досягнення потрібної довжини. Кожен блок проходить через кілька раундів перетворень, кількість яких залежить від довжини ключа (для AES-128 це 10 раундів, для AES-192 – 12, для AES-256 – 14). Усі раунди, крім останнього, мають однакову структуру.

У кожному раунді блок даних піддається серії математичних операцій, які змінюють його вміст так, щоб отримати зашифрований вигляд. Спочатку виконується операція під назвою SubBytes (див. рисунок 4.1), де кожен байт блоку замінюється іншим за допомогою спеціальної таблиці замін (це таблиця, яка

задається стандартом і відома всім, але її використання ускладнює обчислення для зловмисників).

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
1	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0
2	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	D8	31	15
3	04	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75
4	09	83	2C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	84
5	53	D1	00	ED	20	FC	B1	5B	6A	CB	BE	39	4A	4C	58	CF
6	D0	EF	AA	FB	43	4D	33	85	45	F9	02	7F	50	3C	9F	A8
7	51	A3	40	8F	92	9D	38	F5	BC	B6	DA	21	10	FF	F3	D2
8	CD	0C	13	EC	5F	97	44	17	C4	A7	7E	3D	64	5D	19	73
9	60	81	4F	DC	22	2A	90	88	46	EE	B8	14	DE	5E	0B	DB
A	E0	32	3A	0A	49	06	24	5C	C2	D3	AC	62	91	95	E4	79
B	E7	C8	37	6D	8D	D5	4E	A9	6C	56	F4	EA	65	7A	AE	08
C	BA	78	25	2E	1C	A6	B4	C6	E8	DD	74	1F	4B	BD	8B	8A
D	70	3E	B5	66	48	03	F6	0E	61	35	57	B9	86	C1	1D	9E
E	E1	F8	98	11	69	D9	8E	94	9B	1E	87	E9	CE	55	28	DF
F	8C	A1	89	0D	BF	E6	42	68	41	99	2D	0F	B0	54	BB	16

Рисунок 4.1 – Таблиця SubBytes для алгоритму AES (рисунок створено самостійно)

Далі проводиться байти в рядках блоку зсуваються на певну кількість позицій. Потім стовпці блоку перемішуються за допомогою математичних операцій, щоб змішати інформацію між різними частинами блоку. В кінці до блоку додається раундовий ключ, який обчислюється з головного секретного ключа за допомогою спеціального алгоритму розширення ключа.

Після завершення всіх раундів утворюється зашифрований блок. Так обробляється кожен блок даних повідомлення. У підсумку всі зашифровані блоки об’єднуються в зашифрований текст, який можна передавати або зберігати. Розшифрування виконується у зворотному порядку з використанням тих самих ключів і зворотних операцій.

4.1.2 Асиметричні алгоритми

Асиметричне шифрування є важливою складовою сучасних систем захисту даних. Його основна особливість полягає в тому, що замість одного спільного

ключа використовуються два різних ключі: відкритий та закритий. Відкритий ключ призначений для шифрування даних і може бути вільно доступним, тоді як закритий зберігається в секреті й використовується для дешифрування. Такий підхід дозволяє значно спростити процес обміну даними між сторонами, адже немає потреби заздалегідь домовлятися про спільний секретний ключ.

Найбільш відомим і широко використовуваним прикладом асиметричного алгоритму є RSA. Він забезпечує високий рівень безпеки завдяки складності розкладання великих чисел на прості множники, що лежить в основі його математичної моделі. Ще одним прикладом є протокол обміну ключами Diffie-Hellman [12], який використовується для безпечного створення спільного секрету між сторонами в незахищеному середовищі. Ці алгоритми відіграють важливу роль у багатьох сферах, зокрема для реалізації електронних підписів, автентифікації користувачів та захисту симетричних ключів у гібридних криптографічних системах.

Основною перевагою асиметричних методів є високий рівень безпеки й зручність під час обміну ключами. Вони дозволяють ефективно організувати захист даних навіть у відкритих мережах, де є ризик перехоплення інформації. Завдяки цьому асиметричне шифрування часто використовується в інтернет-протоколах, електронних платіжних системах, сервісах електронної пошти та багатьох інших застосуваннях.

Проте асиметричні алгоритми мають і свої недоліки. Головний із них – це висока обчислювальна складність. Операції шифрування й дешифрування в таких алгоритмах потребують більше ресурсів і часу, ніж у симетричних. Це обмежує їх використання в деяких ситуаціях, наприклад у мобільних пристроях чи додатках з обмеженою продуктивністю, де надто велике навантаження на процесор або затримки можуть бути критичними. Саме тому в реальних системах асиметричне шифрування зазвичай поєднується з симетричним для досягнення оптимального балансу між безпекою та продуктивністю.

4.1.2.1 Алгоритм шифрування RSA

Процес шифрування RSA починається з того, що отримувач генерує пару ключів. Для цього він обирає два дуже великі прості числа. Ці числа перемножуються між собою, утворюючи модуль – велике число, яке входить до складу як відкритого, так і закритого ключа. Потім обчислюється функція Ейлера, що використовується для подальших обчислень. Отримувач обирає показник шифрування – це число, яке входить у відкритий ключ. Воно підбирається так, щоб бути взаємно простим із функцією Ейлера. Далі обчислюється показник розшифрування, який є частиною закритого ключа. Його обчислюють так, щоб він дозволяв легко відновлювати вихідні дані за зашифрованими.

Коли відправник хоче зашифрувати повідомлення, він перетворює його у велике число, це може бути текст у вигляді числового представлення, або хеш, або інші дані. Це число підноситься до степеня, який дорівнює показнику шифрування, а результат ділиться на модуль і береться залишок від ділення. Таким чином утворюється зашифроване повідомлення.

Отримувач, маючи закритий ключ, виконує подібну операцію: він підносить зашифроване число до показника розшифрування й бере залишок від ділення на той самий модуль. Це дозволяє відновити початкове повідомлення.

Стійкість RSA базується на тому, що дуже важко знайти вихідні прості числа, якщо відомий лише модуль, бо розкласти велике число на множники – це завдання, яке потребує надзвичайно великих обчислювальних ресурсів.

4.1.3 Легковагові алгоритми

Зростання популярності мережі Інтернет та активне використання мобільних додатків призвели до того, що з’явилася потреба у створенні та впровадженні алгоритмів шифрування, які б відповідали вимогам пристроїв із низьким енергоспоживанням та обмеженими обчислювальними можливостями. У таких умовах традиційні алгоритми, як-от AES чи RSA, не завжди є оптимальними, оскільки вони можуть створювати надмірне навантаження на процесор і споживати більше енергії. Саме тому було розроблено так звані

легковагові криптографічні алгоритми, серед яких варто відзначити Speck та Simon. Ці алгоритми призначені для забезпечення шифрування при мінімальному використанні ресурсів і мають простішу структуру, що дозволяє застосовувати їх у пристроях із обмеженими обчислювальними потужностями.

Ще одним прикладом легковагового алгоритму є PRESENT. Його спеціально створили для використання в системах, що залежать від ресурсів, таких як сенсорні мережі, пристрої Інтернету речей (IoT) або різні мобільні застосунки, де важливо забезпечити базовий рівень безпеки без суттєвого впливу на енергоспоживання чи швидкодію пристрою [13]. У таких середовищах легковагові алгоритми демонструють високу ефективність і дозволяють підтримувати прийнятний рівень захисту даних.

Проте, попри свої переваги у швидкості роботи й низькому споживанні ресурсів, ці алгоритми все ж поступаються традиційним рішенням, таким як AES чи RSA, за рівнем крипостійкості. Це означає, що для систем із підвищеними вимогами до безпеки легковагові алгоритми можуть бути не найкращим вибором. Водночас вони залишаються актуальними й корисними там, де обмеження за енергоспоживанням і ресурсами пристрою є пріоритетом, а загальний ризик атак вважається відносно невисоким [14]. Таким чином, вибір на користь легковагових алгоритмів завжди повинен враховувати особливості конкретної системи та рівень загроз, із якими вона стикається.

4.1.3.1 Алгоритм шифрування PRESENT

PRESENT працює з блоками даних фіксованого розміру 64 біти. Повідомлення для шифрування розбивається на шматки по 64 біти, і кожен такий шматок обробляється окремо. Ключ, який використовується для шифрування, може мати довжину 80 або 128 бітів залежно від вибраної конфігурації.

Шифрування в PRESENT виконується у вигляді послідовності раундів – їх завжди 31. У кожному раунді дані блоку послідовно проходять кілька простих і ефективних операцій, що забезпечують високу стійкість до атак навіть за мінімальних витрат ресурсів. На початку кожного раунду блок даних поєднується

з частиною ключа, що змінюється від раунду до раунду. Це називається операцією додавання ключа. Потім застосовується так звана заміна: кожні 4 біти блоку замінюються іншими відповідно до спеціальної таблиці замін. Ця таблиця однакова для всіх пристроїв і задається стандартом. Вона потрібна для того, щоб навіть дуже схожі вхідні дані після заміни виглядали зовсім по-іншому, підвищуючи стійкість шифру.

Після заміни бітів застосовується перестановка бітів у блоці. Це означає, що біти міняються місцями за заздалегідь визначеною схемою. Таке перемішування забезпечує те, що зміни в одній частині блоку впливають на весь блок у подальших раундах, і створюється сильне змішування даних. Операції повторюються для кожного раунду. Після останнього раунду відбувається ще одне додавання ключа. У результаті отримується зашифрований блок довжиною 64 біти, який уже можна передавати чи зберігати.

4.1.4 Квантостійкі методи

Поява квантових комп'ютерів справді стає серйозним викликом для сучасної криптографії, адже ті алгоритми, що тривалий час вважалися надійними, наприклад RSA чи AES, можуть бути зламані за допомогою квантових обчислень у відносно короткий час. Це змушує дослідників і розробників зосереджувати увагу на створенні нових рішень, здатних протистояти загрозам з боку квантових технологій. Серед таких рішень особливе місце посідають алгоритми CRYSTALS-Kyber і CRYSTALS-Dilithium, які нині активно проходять етапи стандартизації й вже демонструють хороші результати у тестуваннях.

CRYSTALS-Kyber є алгоритмом для шифрування та обміну ключами, який забезпечує високий рівень стійкості проти квантових атак завдяки використанню математичних задач із ґратками. Його головною перевагою є те, що він дозволяє безпечно передавати симетричні ключі навіть у середовищах, де потенційно можуть бути присутні квантові обчислювальні потужності. CRYSTALS-Dilithium, у свою чергу, орієнтований на створення електронних підписів і також ґрунтується на завданнях із ґратками. Він дозволяє забезпечувати автентичність

даних та їхню цілісність у квантобезпечному середовищі. Обидва алгоритми доповнюють один одного й можуть використовуватися разом у комплексних системах захисту.

Водночас варто зазначити, що впровадження постквантових алгоритмів не позбавлене складнощів. Основною проблемою є їхня підвищена обчислювальна складність порівняно з традиційними методами. Це може стати суттєвим обмеженням для пристроїв із невеликим обсягом пам'яті чи обмеженими обчислювальними можливостями, таких як смарт-карти, сенсори або деякі мобільні пристрої. Тому сьогодні активно ведуться дослідження, спрямовані на оптимізацію реалізацій цих алгоритмів і пошук рішень для їх адаптації до різних платформ.

Окрім розробки постквантових алгоритмів, великі надії покладаються й на квантову криптографію, яка базується на законах квантової фізики й теоретично гарантує абсолютну безпеку передавання даних. Проте через високу вартість обладнання та технічну складність такі технології поки що залишаються здебільшого експериментальними й потребують подальших досліджень.

У підсумку можна сказати, що квантова загроза сьогодні виступає не лише викликом для фахівців із кібербезпеки, але й потужним стимулом до інновацій. Розробка і впровадження таких рішень, як CRYSTALS-Kyber та CRYSTALS-Dilithium, є важливим кроком до створення стійких систем захисту, здатних відповідати новим викликам епохи квантових обчислень. Попереду ще чимало завдань із вдосконалення цих алгоритмів та інтеграції їх у сучасну інфраструктуру, але вже зараз зрозуміло, що вони відіграватимуть ключову роль у забезпеченні безпеки даних у найближчому майбутньому.

4.1.4.1 Алгоритм шифрування CRYSTALS-Kyber

Шифрування в CRYSTALS-Kyber починається з того, що відправник генерує випадкову послідовність бітів – це фактично той сеансовий ключ або коротке повідомлення, яке потрібно зашифрувати. Ця послідовність перетворюється у вигляді вектора чисел – багаточлена для подальшої роботи. Далі

генеруються випадкові «шумові» дані: це теж вектори чисел, які додаються до результатів обчислень, щоб ніхто не міг легко вгадати або розшифрувати повідомлення навіть із частковим знанням даних.

На основі відкритого ключа одержувача відправник виконує операції множення цієї матриці на шумовий вектор. До результату додається ще один шумовий вектор. Це дає першу частину шифротексту. Друга частина шифротексту утворюється, коли відкритий ключ перемножається з шумовим вектором і до цього додається ще один шумовий вектор і закодоване повідомлення. Усі ці обчислення виконуються по модулю простого числа, тобто після кожної операції береться залишок від ділення, щоб числа не ставали надто великими.

У результаті формується пара векторів чисел – це і є шифротекст. Його можна безпечно передавати одержувачу. Лише той, хто володіє секретним ключем, зможе обчислити правильний сеансовий ключ, використовуючи ці два вектори та знаючи власний секретний вектор чисел. Цей процес захищений завдяки тому, що знайти правильне рішення без знання секретного ключа настільки складно, що це не під силу навіть квантовим комп’ютерам.

4.2 Інтеграція шифрування в архітектуру клієнтно-орієнтованих додатків

Шифрування відіграє ключову роль у захисті даних клієнтських додатків від несанкціонованого доступу, крадіжки інформації та кібератак. Використання сучасних криптографічних методів дозволяє гарантувати конфіденційність, цілісність і автентичність переданих та збережених даних.

У клієнт-серверній архітектурі шифрування забезпечує безпечну передачу даних між клієнтським пристроєм і сервером. Наприклад, протоколи SSL / TLS шифрують мережевий трафік, запобігаючи атакам типу «людина посередині». У мобільних додатках шифрування використовується для захисту конфіденційної інформації, такої як облікові дані користувачів, банківські реквізити та приватні повідомлення.

Шифрування також допомагає дотримуватися міжнародних стандартів і

нормативних вимог щодо захисту даних. Таким чином, інтеграція криптографічних методів стає обов'язковою умовою для розробки безпечних клієнтно-орієнтованих додатків.

У сучасних клієнтно-орієнтованих додатках шифрування є важливим елементом архітектури. Наприклад, мобільні додатки для онлайн-банкінгу активно використовують симетричні алгоритми AES для шифрування даних, що зберігаються, і асиметричні алгоритми RSA для захищеного обміну ключами між клієнтом і сервером. Веб-додатки, такі як платформи електронної комерції, застосовують протоколи SSL / TLS для захисту даних під час передачі.

Використання шифрування в сучасних системах значно знижує ризики витоків даних, захищає комунікації та дозволяє користувачам довіряти свої дані додаткам і сервісам. Інтеграція цих рішень є важливим кроком для підвищення надійності та безпеки клієнтно-орієнтованих додатків.

Сучасні криптографічні методи, зокрема легковагові та квантостійкі алгоритми, все частіше інтегруються у клієнтно-орієнтовані додатки, щоб забезпечити баланс між безпекою, продуктивністю та енергоефективністю. Легковагові алгоритми, такі як PRESENT, ідеально підходять для мобільних пристроїв та веб додатків, де ресурси обмежені. Вони забезпечують достатній рівень безпеки при мінімальному споживанні енергії, що є критичним для сенсорних мереж та портативних пристроїв.

Квантостійкі алгоритми, такі як CRYSTALS-Kyber та Dilithium, знаходять своє застосування у захищених комунікаціях та хмарних обчисленнях, адже їхній дизайн враховує загрози з боку квантових комп'ютерів. Використання таких методів дозволяє зберігати конфіденційність даних навіть у середовищі з підвищеними вимогами до безпеки.

Гібридні методи шифрування поєднують переваги симетричних та асиметричних алгоритмів, забезпечуючи ефективний обмін ключами та шифрування великих обсягів даних. Такі методи широко застосовуються в протоколах SSL / TLS та в інфраструктурах відкритих ключів, де важливим є як швидкодія, так і висока стійкість до атак.

Інтеграція цих методів у клієнтно-орієнтовані додатки дозволяє адаптувати криптографічні рішення до конкретних сценаріїв використання, забезпечуючи високий рівень захисту даних навіть у найскладніших умовах.

4.3 Довжина ключів алгоритмів шифрування

Алгоритми шифрування мають різну довжину ключів (див. рисунок 4.2), що безпосередньо впливає на рівень безпеки та продуктивність їх роботи. Вибір довжини ключа є важливим аспектом при проектуванні систем захисту інформації, адже він визначає стійкість алгоритму до атак, а також обчислювальні витрати на шифрування та дешифрування даних.

Алгоритм	Тип шифрування	Довжина ключа	Рівень безпеки
AES	Симетричне	128 / 192 / 256 біт	Високий
RSA	Асиметричне	2048 / 3072 / 4096 біт	Високий – дуже високий
CRYSTALS-Kyber	Постквантове	8000 / 12000 / 16000 біт	Дуже високий
PRESENT	Симетричне, легковагове	80 / 128 біт	Середній – прийнятний

Рисунок 4.2 – Довжина ключів у алгоритмах шифрування (рисунок виконано самостійно)

AES підтримує кілька варіантів довжини ключів: 128 біт, 192 біт та 256 біт. Найбільш поширеними на практиці є AES-128 та AES-256, оскільки вони забезпечують оптимальний баланс між рівнем безпеки та швидкістю обробки даних. Чим більший розмір ключа, тим складніше виконати перебір ключів або інші види атак, проте збільшується й обчислювальне навантаження на систему.

RSA, як алгоритм асиметричного шифрування, потребує значно більшої довжини ключа для досягнення належного рівня захисту. Стандартною довжиною ключа для RSA вважається 2048 біт, проте для більш високого рівня безпеки, особливо з урахуванням сучасних загроз, застосовуються ключі довжиною 3072 біт або навіть 4096 біт. Причиною цього є те, що стійкість RSA базується на складності

розкладання великих чисел на множники, а отже, для підвищення крипостійкості доводиться збільшувати розмір ключа.

CRYSTALS-Kyber, який належить до постквантових алгоритмів, представлений у кількох варіантах – Kyber512, Kyber768 і Kyber1024. Ці позначення відображають рівень параметрів безпеки. Довжина публічного ключа в таких алгоритмах є значно більшою, ніж у традиційних алгоритмах: для Kyber512 вона становить близько 8000 біт, для Kyber768 – понад 12 000 біт, а для Kyber1024 – ще більше. Це зумовлено тим, що для протидії атакам квантових комп'ютерів потрібні значно більші обсяги даних для генерації ключів.

PRESENT передбачає використання ключів довжиною 80 біт або 128 біт. Незважаючи на відносно невелику довжину ключа порівняно з іншими алгоритмами, він забезпечує прийнятний рівень безпеки для тих сценаріїв, де головними є мінімізація споживання ресурсів та енергоефективність.

Таким чином, можна зазначити, що довжина ключа тісно пов'язана з цілями використання того чи іншого алгоритму. Симетричні алгоритми, як AES чи PRESENT, працюють із коротшими ключами і підходять для захисту даних у режимі реального часу, тоді як асиметричні й постквантові алгоритми потребують значно більших ключів для забезпечення високого рівня безпеки, особливо в умовах сучасних і майбутніх загроз. Вибір довжини ключа та алгоритму шифрування має базуватись на вимогах до безпеки, продуктивності системи та характеристиках середовища, де ці алгоритми застосовуються.

4.4 Гомоморфне шифрування

Гомоморфне шифрування є особливим видом криптографічного шифрування, який дозволяє виконувати обчислення безпосередньо над зашифрованими даними. Це означає, що дані можуть залишатися зашифрованими протягом усього процесу обробки, а результати таких обчислень після розшифрування будуть ідентичними тим, які б були отримані при виконанні тих самих операцій над відкритими даними. Така властивість відкриває можливості

для безпечної обробки конфіденційної інформації на сторонніх платформах, наприклад у хмарних сервісах або на серверах з обмеженим рівнем довіри.

Основні алгоритми гомоморфного шифрування можна поділити за типами підтримуваних операцій і рівнем гнучкості виконання обчислень. Одним із перших і найвідоміших прикладів повністю гомоморфного шифрування є схема, запропонована Крейгом Джентрі у 2009 році. Ця схема ґрунтується на використанні решіткових структур і забезпечує можливість виконання як додавання, так і множення над зашифрованими даними без обмежень на кількість таких операцій. Важливо зазначити, що схема Джентрі заклала фундамент для подальших досліджень у цій сфері, хоча її практичне використання було обмежене через високу обчислювальну складність і значне збільшення розміру шифротекстів.

З часом були розроблені й інші алгоритми, що оптимізують ідеї гомоморфного шифрування та роблять його більш практичним. Наприклад, алгоритми на основі решіткових проблем, такі як BGV (Brakerski-Gentry-Vaikuntanathan), BFV (Brakerski-Fan-Vercauteren), CKKS (Cheon-Kim-Kim-Song), дозволяють виконувати певний набір арифметичних операцій над шифротекстами і застосовуються в реальних прототипах захищених обчислень. BGV і BFV забезпечують точну арифметику над цілими числами, тоді як CKKS орієнтується на обчислення з числами з плаваючою комою, що робить його корисним для машинного навчання на зашифрованих даних. Ці схеми значно зменшили обчислювальні витрати у порівнянні з оригінальною схемою Джентрі й дозволили реалізовувати гомоморфні обчислення в експериментальних і навіть комерційних продуктах.

Гомоморфне шифрування продовжує залишатися активною сферою наукових досліджень і розробок, оскільки його властивості відкривають нові горизонти для захищеної обробки інформації в умовах зростаючих загроз та потреб у конфіденційності.

Переваги гомоморфного шифрування:

– дозволяє виконувати обчислення над зашифрованими даними, не

розшифровуючи їх, що мінімізує ризик розкриття приватної інформації;

- дані залишаються захищеними навіть під час обробки на сторонніх серверах, що особливо важливо для хмарних обчислень;
- повністю гомоморфні схеми дозволяють виконувати як додавання, так і множення над шифротекстами необмежену кількість разів;
- сучасні схеми підтримують точні або наближені обчислення, що робить технологію корисною для широкого спектру задач, зокрема машинного навчання.

Недоліки гомоморфного шифрування:

- початкові схеми є надзвичайно ресурсозатратними у плані процесорного часу та пам’яті;
- зашифровані дані можуть бути у десятки або сотні разів більшими за початкові, що ускладнює зберігання та передачу;
- хоча сучасні алгоритми стали більш ефективними, продуктивність все ще може бути недостатньою для деяких застосувань із великим обсягом даних або високою швидкістю обробки;
- розробка ефективних систем на основі гомоморфного шифрування вимагає глибоких знань криптографії і часто потребує тонкої оптимізації;
- деякі схеми підтримують лише певні типи операцій (наприклад, додавання або множення), що може обмежувати застосування.

Гомоморфне шифрування знаходить застосування в тих сферах, де потрібна обробка конфіденційної інформації без її розшифрування. Однією з ключових галузей є хмарні обчислення. Використовуючи гомоморфне шифрування, компанії та приватні користувачі можуть безпечно передавати дані до хмарних сервісів для обробки, не розкриваючи їхній зміст навіть хостинговому провайдеру. Це дозволяє будувати сервіси для фінансових розрахунків, обробки приватних документів та інших чутливих даних у середовищах з обмеженою довірою.

Ще одна важлива сфера застосування – медицина. У цій галузі гомоморфне шифрування забезпечує можливість аналізу даних пацієнтів для діагностики,

досліджень або статистики без загрози розкриття персональної інформації. Це особливо актуально при обміні даними між лікарнями або під час використання сторонніх аналітичних платформ.

Технологія також застосовується у фінансовому секторі, де потрібна обробка зашифрованих транзакцій або виконання розрахунків між банками й фінансовими установами без розкриття деталей операцій. Це дає змогу зберігати конфіденційність фінансової інформації навіть при проведенні складних обчислень.

У сфері штучного інтелекту та машинного навчання гомоморфне шифрування дає можливість будувати моделі, які навчаються або роблять прогнози без доступу до незашифрованих даних. Це відкриває перспективи для використання технології в розумних сервісах із підвищеними вимогами до приватності, наприклад у медичних або юридичних консультаціях на основі даних користувача.

Завдяки своїм властивостям гомоморфне шифрування стає перспективним інструментом для будь-яких систем, де потрібно поєднати обчислювальну потужність віддалених сервісів з високим рівнем захисту даних.

4.5 Аналіз алгоритмів шифрування

Для аналізу алгоритмів шифрування необхідно враховувати їх продуктивність, енергоефективність та стійкість до атак (див. рисунок 4.3).

Критерій	AES (симетричний)	RSA (асиметричний)	PRESENT (легковаговий)	CRYSTALS-Kyber (квантостійкий)
Продуктивність	Висока	Низька	Висока	Середня
Енергоефективність	Висока	Низька	Висока	Середня
Стійкість до атак	Висока (традиційні атаки)	Висока (традиційні атаки)	Середня	Висока (включно з квантовими)
Сфера застосування	Великі обсяги даних	Автентифікація, передача ключів	IoT, мобільні пристрої	Квантова безпека, довгострокові дані

Рисунок 4.3 – Порівняння алгоритмів шифрування (рисунок виконано самостійно)

Можна зробити висновок, що кожен із розглянутих алгоритмів має свої чітко окреслені сфери застосування, які залежать від вимог до безпеки, продуктивності, енергоефективності та доступних ресурсів конкретної системи. Алгоритм AES продовжує залишатися основним вибором для захисту великих обсягів інформації завдяки поєднанню високої швидкодії, гнучкості у виборі режимів роботи та перевіреної на практиці стійкості до широкого спектра криптоаналітичних атак. Його застосування є особливо доцільним у серверних рішеннях, хмарних обчислювальних платформах, корпоративних мережах і системах електронного урядування, де швидкість обробки даних і надійність захисту мають вирішальне значення. Крім того, завдяки ефективній реалізації на апаратному рівні, AES підходить навіть для пристроїв із обмеженими часовими рамками обробки даних, зокрема в системах реального часу.

Алгоритм RSA, попри свої обмеження у швидкодії та відносно високе споживання обчислювальних ресурсів, зберігає актуальність у тих завданнях, де пріоритетним є захищений обмін ключами, перевірка автентичності та забезпечення юридичної значущості електронних документів завдяки використанню цифрового підпису. Його крипостійкість базується на складності факторизації великих чисел, що дає змогу ефективно захищати конфіденційну інформацію під час її передавання через публічні канали зв’язку. Проте у випадку мобільних пристроїв, сенсорних вузлів чи інших систем із обмеженими обчислювальними можливостями застосування RSA часто є менш практичним через його ресурсомісткість.

Алгоритм PRESENT, як представник легковагових криптографічних рішень, знаходить широке застосування в таких сферах, як Інтернет речей, сенсорні мережі, вбудовані системи, медичне обладнання та інші пристрої, що працюють в умовах обмеженої енергетичної автономності або мають мінімальний обсяг вільної пам’яті та обчислювальних ресурсів. Його компактна структура й низькі апаратні вимоги роблять його ефективним інструментом для забезпечення базового рівня захисту без суттєвого навантаження на систему й збільшення витрат енергії. Це дає змогу використовувати його навіть у найменших

мікроконтролерах та інтегрованих пристроях.

CRYSTALS-Kyber демонструє значний потенціал у сфері постквантової криптографії, оскільки він створений із урахуванням нових загроз, пов’язаних із появою квантових комп’ютерів. Його використання є особливо актуальним у сценаріях, де важливо забезпечити довготривалу конфіденційність інформації, наприклад при зберіганні персональних даних, фінансової звітності або державних секретів. CRYSTALS-Kyber застосовується в експериментальних проєктах із побудови стійких до квантових атак протоколів обміну ключами та шифрування даних у хмарних і розподілених системах. Попри те, що його продуктивність у певних випадках поступається традиційним симетричним алгоритмам, його перевагою є підвищена стійкість до атак на основі квантових обчислень, що робить його стратегічно важливим для майбутніх систем кіберзахисту.

Загалом вибір алгоритму завжди має ґрунтуватися на комплексному аналізі конкретного завдання, характеристик апаратного й програмного середовища та прогнозів щодо розвитку загроз. Не існує універсального рішення, яке було б оптимальним для всіх сценаріїв – кожен алгоритм є окремим інструментом у розпорядженні розробника безпекових систем, і правильна комбінація цих інструментів дає змогу створювати сучасні системи захисту, стійкі до актуальних і майбутніх атак. Успішна побудова таких систем вимагає не лише вибору криптоалгоритмів, а й комплексного підходу до розробки архітектури безпеки, управління ключами й протоколів обміну даними..

5 ЕКСПЕРИМЕНТАЛЬНА ЧАСТИНА

5.1 Вхідні дану експерименту

Було вирішено провести два різних експеримента, що дозволить наглядно підтвердити та проаналізувати алгоритми шифрування між собою, що в свою чергу дозволить створити рекомендації для клієнтно-орієнтованих додатків.

Для порівняння алгоритмів шифрування (див. рисунок 5.1, рисунок 5.2) був написаний код на мові програмування Python, який дозволяє оцінити продуктивність різних криптографічних алгоритмів. У реалізації використовувалися такі популярні бібліотеки, як Crypto [15] для симетричних та асиметричних алгоритмів, pypresent для роботи з легковаговими алгоритмами, такими як PRESENT [16], та pqcrypto [17] для постквантових алгоритмів, зокрема CRYSTALS-Kyber. Вибір цих бібліотек зумовлений їх зручністю, широкою підтримкою та відповідністю сучасним стандартам криптографії. Повний код розробленого застосунку можна знайти на посиланням [18] у списку джерел.

Для графічної візуалізації була використана бібліотека Tkinter.

5.2 Експеримент з строками

Суть експеримента полягає в тому, що на вході користувач дає строку довільного розміру. Після чого всі чотири алгоритми шифрують її та повертають час, який займає процес шифрування. Можна побачити, що для короткої строки (див. рисунок 5.1) цей час займає мілісекунди, але навіть тут вже можна побачити різницю. AES і PRESENT значно швидші за RSA та Kyber. AES показав найкращу продуктивність: він майже миттєво обробив короткий рядок, що підтверджує його ефективність для завдань, де потрібна висока швидкість. PRESENT хоч і трохи поступається AES за швидкістю, але теж продемонстрував дуже хороший результат і може бути корисним там, де важливо економити ресурси. RSA і Kyber на короткому тексті працювали значно повільніше. Це пояснюється тим, що ці алгоритми є більш складними за своєю структурою, вони призначені для інших цілей, і швидкість тут не є їх головною перевагою.

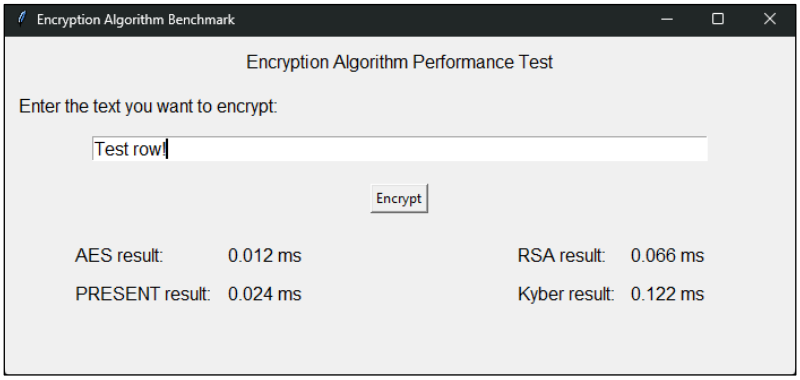

Рисунок 5.1 – Результат порівняння алгоритмів з коротким рядком (рисунок виконано самостійно)

Для довгої строки (див. рисунок 5.2) загалом результат залишився таким же, як і для короткого тексту. AES знову показав себе з найкращого боку й упорався із завданням дуже швидко навіть при збільшенні обсягу даних. PRESENT теж майже не втратив у швидкості й лишився серед лідерів за продуктивністю. RSA і Kyber при роботі з довгим текстом виявилися повільнішими, але приріст часу у порівнянні з коротким текстом був не надто великий. Це свідчить про те, що для асиметричних алгоритмів довжина повідомлення не так сильно впливає на час роботи, бо основна складність закладена в самому принципі шифрування.

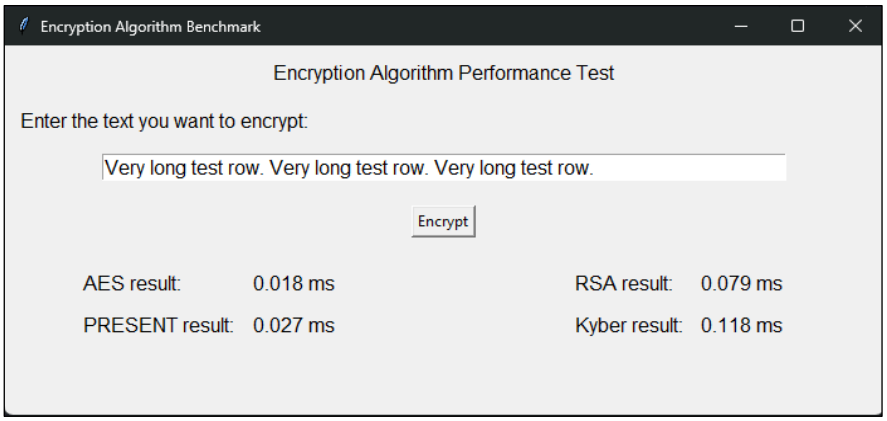

Рисунок 5.2 – Результат порівняння алгоритмів з довгою строкою (рисунок виконано самостійно)

Загалом обидва теста підтверджують, що для завдань, де потрібно швидко обробляти дані, особливо в режимі реального часу або при роботі з великими обсягами інформації, варто обирати симетричні алгоритми, такі як AES або

PRESENT. Вони добре підходять для клієнтських додатків, де важлива швидкість роботи й обмежені ресурси пристрою. RSA і Kyber більше підходять для завдань, де на першому місці стоїть безпека під час обміну ключами, створення цифрового підпису або забезпечення захисту від потенційних загроз у майбутньому, включно з квантовими атаками. Тобто ці алгоритми варто використовувати там, де час шифрування не є критичним фактором, а безпека стоїть на першому місці. У результаті видно, що вибір алгоритму залежить від конкретної ситуації та вимог до додатку: чи то швидкість, чи надійність, чи підготовка до майбутніх викликів у сфері кібербезпеки.

Можна побачити, що для короткої строки (див. рисунок 5.1) цей час займає мілісекунди, але навіть тут вже можна побачити різницю. AES і PRESENT значно швидші за RSA та Kyber. AES показав найкращу продуктивність: він майже миттєво обробив короткий рядок, що підтверджує його ефективність для завдань, де потрібна висока швидкість. PRESENT хоч і трохи поступається AES за швидкістю, але теж продемонстрував дуже хороший результат і може бути корисним там, де важливо економити ресурси. RSA і Kyber на короткому тексті працювали значно повільніше. Це пояснюється тим, що ці алгоритми є більш складними за своєю структурою, вони призначені для інших цілей, і швидкість тут не є їх головною перевагою.

Для зручності перегляду результатів був створений графік (див. рисунок 5.3)

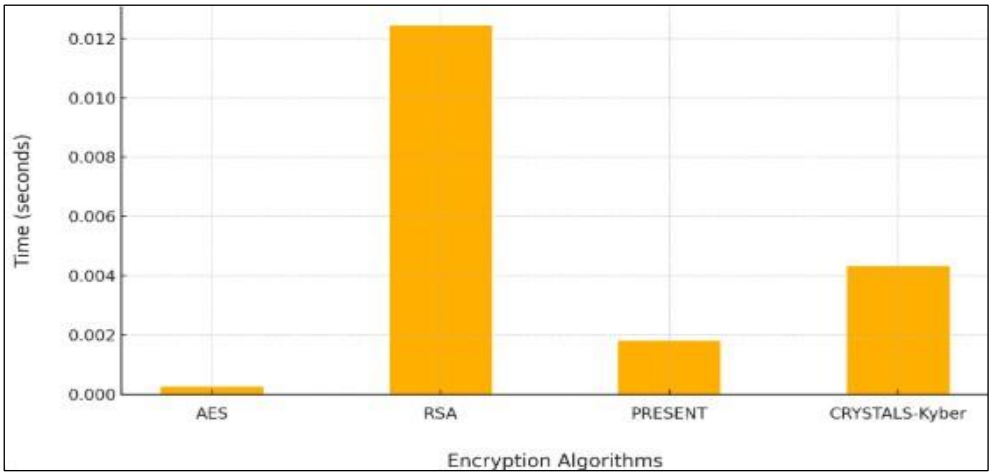

Рисунок 5.3 – Графіки порівняння алгоритмів (рисунок виконано самостійно)

5.3. Експеримент з архівами

Суть експеримента полягає в тому, що на вході користувач дає архів з текстовими файлами. Кожен наступний архів у 10 разів більший за попередній. Початковий розмір архів – 10 кб, фінальний розмір – 1 гб. Після чого всі чотири алгоритми шифрують алгоритми та повертають час, який займає процес шифрування.

На основі даних, наведених на рисунку (див. рисунок 5.4), можна зробити висновок, що алгоритм AES демонструє найкращі показники швидкості шифрування для всіх обсягів архівів. Його значення часу залишаються мінімальними навіть при збільшенні розміру даних. У той час як RSA виявився найповільнішим – при великих обсягах, таких як 100 MB або 1 GB, час шифрування становить хвилини, що робить його малопридатним для роботи з великими файлами. Алгоритм PRESENT показує середню швидкість – він працює краще за RSA, але повільніше за AES. Постквантовий алгоритм CRYSTALS-Kyber демонструє досить стабільні результати і шифрує дані швидше за RSA, але трохи поступається AES та PRESENT.

№	Розмір архіва	AES	RSA	PRESENT	CRYSTALS-Kyber
1	10 KB	0,012 мс	13 мс	0,7 мс	0,51 мс
2	100 KB	0,13 мс	0,13 с	7 мс	5,12 мс
3	1 MB	1,28 мс	1,31 с	1,01 мс	49 мс
4	10 MB	14 мс	18 с	0,7 с	0,526 с
5	100 MB	0,14 с	2 хв 23 с	6 с	4,0 с
6	1 GB	1,22 с	36 хв	54,5 с	54 с

Рисунок 5.4 – Результат порівняння шифрування архівів (рисунок виконано самостійно)

Проведене тестування показало, що:

- AES є найкращим вибором для шифрування великих обсягів даних у реальному часі завдяки високій продуктивності;
- PRESENT є оптимальним для пристроїв із обмеженими ресурсами, через низьке енергоспоживання та високу швидкодію;
- CRYSTALS-Kyber демонструє високий рівень захисту від квантових атак і є перспективним вибором для довгострокового зберігання

критично важливих даних;

– RSA забезпечує надійний обмін ключами та автентифікацію, проте його висока обчислювальна складність обмежує використання в сценаріях із низькими ресурсами.

Таким чином, вибір алгоритму залежить від специфіки використання. Для реального часу та великих даних найкращим є AES, а для веб сфери таким алгоритмом є PRESENT, для довготривалого захисту підходить CRYSTALS-Kyber, а для автентифікації та обміну ключами – RSA. Кожен із цих алгоритмів має свої сильні та слабкі сторони, і їх вибір залежить від вимог конкретної задачі.

Грамотний вибір алгоритму шифрування дозволяє не тільки підвищити ефективність і безпеку системи, але й забезпечити її відповідність сучасним і майбутнім викликам у сфері інформаційної безпеки.

5.4. Рекомендації вибору алгоритмів

На основі проведеного аналізу та практичного тестування розроблено розширені рекомендації щодо вибору криптографічних методів для захисту даних у клієнтно-орієнтованих додатках. Рекомендації були основані по прикладу робот Chalyi S., Leshchynskyi V., Leshchynska I. [19] [20]. Ці рекомендації враховують специфіку середовища застосування, вимоги до продуктивності, рівня безпеки та ресурсних обмежень (див. рисунок 5.5).

Критерії вибору алгоритмів:

– для додатків, які працюють із великими обсягами даних у реальному часі та потребують високої продуктивності, оптимальним вибором є AES. Цей алгоритм забезпечує відмінне співвідношення між швидкістю шифрування та рівнем захисту, а також ефективно використовує апаратні ресурси, що робить його ідеальним для серверних систем, вебсервісів та потокового передавання даних;

– для мобільних пристроїв, де є обмеження на обчислювальні потужності та енергоспоживання, доцільно використовувати PRESENT. Цей легковаговий алгоритм забезпечує достатній рівень безпеки при

мінімальному споживанні ресурсів, що дозволяє збільшити автономність пристроїв та зменшити навантаження на процесор;

– для довгострокового зберігання конфіденційних даних і захисту від майбутніх квантових атак рекомендовано застосовувати CRYSTALS-Kyber. Цей алгоритм вже зараз демонструє високу стійкість проти атак, що потенційно можуть виконуватися квантовими комп'ютерами. Його використання забезпечує додатковий запас міцності для інформації, яка повинна залишатися конфіденційною протягом багатьох років;

– для задач автентифікації та обміну ключами оптимальним рішенням є RSA. Цей алгоритм перевірений часом і широко підтримується як у програмному, так і в апаратному забезпеченні. RSA є надійним інструментом для створення електронних підписів, захисту електронної пошти, встановлення безпечних з'єднань (наприклад, у протоколах SSL/TLS).

Вибір алгоритму повинен базуватися на особливостях архітектури додатка, вимогах до швидкодії та рівня безпеки, а також на врахуванні можливих загроз, включаючи перспективу розвитку квантових технологій. При розробці клієнтно-орієнтованих додатків доцільно поєднувати кілька методів шифрування для досягнення оптимального балансу між продуктивністю, стійкістю та гнучкістю захисту.

Сценарій використання	Рекомендований алгоритм	Обґрунтування
Шифрування великих обсягів даних у реальному часі	AES	Висока продуктивність та енергоефективність
Мобільні пристрої та IoT-системи	PRESENT	Низьке споживання ресурсів, висока швидкодія
Довгострокове зберігання конфіденційних даних	CRYSTALS-Kyber	Стійкість до квантових атак, довготривалий рівень безпеки
Автентифікація та обмін ключами	RSA/ECC	Надійність для забезпечення автентичності та безпечного обміну ключами

Рисунок 5.5 – Рекомендації використання алгоритмів (рисунок виконано самостійно)

Оптимізація алгоритмів:

– у середовищах із обмеженими ресурсами варто впроваджувати

легковагові алгоритми з мінімальним споживанням енергії;

- для підвищення продуктивності можна використовувати апаратне прискорення, наприклад, підтримку AES у сучасних процесорах;
- у сценаріях з високими вимогами до безпеки впроваджувати мультифакторний підхід, поєднуючи кілька типів алгоритмів для різних рівнів захисту.

Перспективи використання квантової криптографії:

- зважаючи на розвиток квантових обчислень, впровадження квантостійких алгоритмів, таких як CRYSTALS-Kyber, має стати стандартом для систем із довготривалими вимогами до безпеки;
- активно розробляти та тестувати гібридні системи, що поєднують традиційні та квантові методи для поступової адаптації до нових загроз;
- вивчати інтеграцію квантових алгоритмів у хмарні сервіси для захисту великих обсягів даних.

Стандартизація та майбутні дослідження:

- сприяти впровадженню відкритих стандартів для квантостійких алгоритмів;
- проводити додаткові дослідження в галузі автоматизованого вибору алгоритмів залежно від параметрів системи, таких як продуктивність, енергоефективність та специфіка додатків.

Впровадження цих рекомендацій дозволить забезпечити максимальну безпеку клієнтно-орієнтованих додатків, враховуючи специфіку їх використання та сучасні виклики інформаційної безпеки. Це особливо актуально у зв'язку з розвитком квантових технологій, які становлять загрозу для традиційних методів шифрування. Крім того, дотримання рекомендацій дозволить оптимізувати використання ресурсів у мобільних пристроях, забезпечуючи збалансоване співвідношення між продуктивністю та захистом даних. Успішна реалізація таких підходів сприятиме впровадженню сучасних стандартів безпеки та підвищенню рівня довіри до клієнтно-орієнтованих сервісів.

ВИСНОВКИ

У результаті проведеного дослідження було детально проаналізовано сучасні методи шифрування даних, зокрема симетричні, асиметричні, легковагові та квантостійкі алгоритми. Було визначено їх переваги, недоліки та області застосування. Практична частина роботи підтвердила ефективність алгоритмів у різних сценаріях використання, а також показала, що вибір технології шифрування має базуватися на конкретних вимогах до продуктивності, енергоефективності та стійкості до атак.

Особливу увагу приділено перспективам використання різних типів алгоритмів залежно від їхніх характеристик і сфер застосування. Симетричні алгоритми, такі як AES, забезпечують високу продуктивність і енергоефективність, що робить їх ідеальними для роботи з великими обсягами даних у реальному часі. Легковагові алгоритми, наприклад, PRESENT, демонструють оптимальні результати в умовах обмежених ресурсів, таких як IoT-пристрої. Асиметричні алгоритми, такі як RSA, залишаються ефективними для автентифікації та обміну ключами.

Квантостійкі алгоритми, зокрема CRYSTALS-Kyber, виявилися перспективними для захисту даних у довгостроковій перспективі, особливо з урахуванням можливих загроз з боку квантових обчислень. Таким чином, вибір оптимального алгоритму залежить від специфіки завдань та ресурсних обмежень.

Розроблені рекомендації дозволяють інтегрувати сучасні методи шифрування у клієнтно-орієнтовані додатки, забезпечуючи високий рівень безпеки та відповідність сучасним викликам. Це особливо актуально для захисту даних у реальному часі, роботи з мобільними пристроями та IoT-системами, а також для довгострокового зберігання конфіденційної інформації.

Результати дослідження підтвердили ефективність симетричних, асиметричних, легковагових та квантостійких алгоритмів залежно від специфічних умов використання. Наприклад, AES показав високу продуктивність при роботі з великими обсягами даних, тоді як PRESENT продемонстрував ефективність у середовищах з обмеженими ресурсами. Алгоритм CRYSTALS-

Kyber виявився перспективним рішенням для майбутніх викликів, пов’язаних із розвитком квантових обчислень.

Впровадження цих рекомендацій сприятиме підвищенню надійності інформаційних систем, створенню гнучких та масштабованих рішень для різних сфер використання та адаптації до нових викликів у галузі кібербезпеки.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Udara P. CRYSTALS Kyber : The Key to Post-Quantum Encryption URL: https://medium.com/identity-beyond-borders/crystals-kyber-the-key-to-post-quantum-encryption-3154b305e7bd .

2. Honcharnko D. Шифрування: типи і алгоритми. Що це, чим відрізняються і де використовуються? URL: https://hostpro.ua/wiki/ua/security/encryption-types-algorithms/.

3. EXBASE Порівняння симетричного і асиметричного шифрування. URL: https://exbase.io/uk/wiki/simetrichne-i-asimetrichne-shifruvannya.

4. Шеннон К. А. Mathematical Theory of Communication. URL: https://uk.wikipedia.org/wiki/Математична_теорія_зв%27язку_(стаття).

5. Литвинеко В. Алгоритм шифрування RSA, види атак на нього. Реалізація мовою Python. URL: https://dou.ua/forums/topic/43026/.

6. DES: The story of the Data Encryption Standard. URL: https://coinrivet.com/uk/des-the-story-of-the-data-encryption- standard/.

7. Лазаря А. Що таке розширений стандарт шифрування (AES) і як він пов'язаний з NIST**?** URL: https://lazarusalliance.com/uk/what-is-advanced-encryption-standard-aes-and-how-is-it-related-to-nist/.

8. Huaxin Wang, Yiwen Gao, Yuejun Liu, Qian Zhang, Yongbin Zhou In- depth Correlation Power Analysis Attacks on a Hardware Implementation of CRYSTALS-Dilithium. URL: https://cybersecurity.springeropen.com/articles/10.1186/s42400- 024-00209-9.

9. What is SSL, TLS & HTTPS? How do they increase trust in websites? And how to look beyond the lock to know who's behind the website. URL: https://www.digicert.com/what-is-ssl-tls-and-https.

10. Abhay Singh Kathayat. Python: A Comprehensive Overview in One Article. URL: https://dev.to/abhay_yt_52a8e72b213be229/python-a- comprehensive-overview-in-one-article-3lmh.

11. Shimura M. RSA Encryption and Decryption with Python's pycryptodome Library. URL: https://medium.com/coinmonks/rsa-encryption-and-decryption- with-

pythons-pycryptodome-library-94f28a6a1816.

12. Quiocho C. What Is a Diffie-Hellman Key exchange? URL: https://www.ninjaone.com/it-hub/endpoint-security/what-is-a- diffie-hellman-key-exchange/.

13. Sleem L., Couturier R. Speck-R: An ultra light-weight cryptographic scheme for Internet of Things. URL: https://www.researchgate.net/publication/344539102_Speck-R_An_ultra_light-weight_cryptographic_scheme_for_Internet_of_Things .

14. Salton G., Simon D., Lin C. Exploring Simon's Algorithm with Daniel Simon. URL: https://aws.amazon.com/blogs/quantum-computing/simons- algorithm/.

15. Python Cryptography Toolkit (pycrypto). URL: https://pypi.org/project/pycrypto/.

16. Imdad M., Najwa S., Mahdin H. An Enhanced Key Schedule Algorithm of PRESENT-128 Block Cipher for Random and Non-Random Secret Keys. URL: https://www.mdpi.com/2073-8994/14/3/604.

17. Post-Quantum Cryptography (PQCrypto). URL: https://pypi.org/project/pqcrypto/.

18. GitHub - MaksymKudlaienko/Master-s-thesis. *GitHub*. URL: https://github.com/MaksymKudlaienko/Master-s-thesis (дата звернення: 12.06.2025).

19. Chalyi S., Leshchynskyi V., Leshchynska I. Detailing explanations in the recommender system based on matching temporal knowledge // *Eastern-European Journal of Enterprise Technologies*. – 2020. – Т. 4. – № 2 (106). – С. 6–13. URL: https://doi.org/10.15587/1729-4061.2020.210013. (дата звернення: 30.05.2025).

20. Chalyi S., Leshchynskyi V., Leshchynska I. Method of forming recommendations using temporal constraints in a situation of cyclic cold start of the recommender system // *EUREKA: Physics and Engineering*. – 2019. – № 4. – С. 34–40. URL: https://doi.org/10.21303/2461-4262.2019.00952. (дата звернення: 30.05.2025).

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ ЗА НАУКОВИМИ НАПРЯМАМИ КЕРІВНИКА ТА НАУКОВЦІВ КАФЕДРИ ПРОГРАМНОЇ ІНЖЕНЕРІЇ

19. Chalyi S., Leshchynskyi V., Leshchynska I. Detailing explanations in the recommender system based on matching temporal knowledge // *Eastern-European Journal of Enterprise Technologies*. – 2020. – Т. 4. – № 2 (106). – С. 6–13. URL: https://doi.org/10.15587/1729-4061.2020.210013. (дата звернення: 30.05.2025).

20. Chalyi S., Leshchynskyi V., Leshchynska I. Method of forming recommendations using temporal constraints in a situation of cyclic cold start of the recommender system // *EUREKA: Physics and Engineering*. – 2019. – № 4. – С. 34–40. URL: https://doi.org/10.21303/2461-4262.2019.00952. (дата звернення: 30.05.2025).