Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя
(повне найменування вищого навчального закладу)
Факультет комп'ютерно-інформаційних систем і програмної інженерії
(назва факультету)
Кафедра кібербезпеки
(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

магістр
(освітній рівень)

на тему: Застосування цифрового підпису у системах з наскрізним шифруванням повідомлень

Виконав: студент VI курсу, групи СБм-61
Спеціальності:
125 «Кібербезпека»
(шифр і назва напряму підготовки, спеціальності)
Пилипчук В. М.
підпис (прізвище та ініціали)

Керівник Загородна Н.В.
підпис (прізвище та ініціали)

Нормоконтроль Стадник М.А.
підпис (прізвище та ініціали)

Завідувач кафедри Загородна Н.В.
підпис (прізвище та ініціали)

Рецензент
підпис (прізвище та ініціали)

м. Тернопіль – 2025

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра кібербезпеки
(повна назва кафедри)

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ Загородна Н.В.
(підпис) (прізвище та ініціали)
«__» __грудня__ 2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня Магістр
(назва освітнього ступеня)

за спеціальністю 125 Кібербезпека та захист інформації
(шифр і назва спеціальності)

Студенту Пилипчуку Володимиру Миколайовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Застосування цифрового підпису у системах з наскрізним шифруванням повідомлень

Керівник роботи Загородна Наталія Володимирівна, завідувач кафедри, PhD, доц.
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від « 24 » листопада 2025 року № 4/7-1024

2. Термін подання студентом завершеної роботи

3. Вихідні дані до роботи Веб додаток , вимоги впровадження цифрового підпису

4. Зміст роботи (перелік питань, які потрібно розробити)
Вступ. Розділ 1. Аналітична частина. 1.1 Аналітичний огляд. 1.2 Аналіз існуючих підходів до захисту повідомлень у системах з наскрізним шифруванням. 1.3 Обґрунтування вибору криптографічних алгоритмів та засобів реалізації. 1.4 Модель загроз та сценарії атак для системи. 1.5 Висновки до розділу 1. Розділ 2. Основна частина. 2.1 Загальна архітектура рішення. 2.2 Проєктування механізму цифрового підпису в додатку. 2.3 Проєктування зберігання та захисту криптографічних ключів. 2.4 Реалізація механізму цифрового підпису та взаємодія модулів. Розділ 3. Науково-дослідна частина. 3.1 Дослідження стійкості додатку до атак на автентичність повідомлень. 3.2 Експериментальна перевірка механізму цифрового підпису. 3.3 Аналіз результатів експериментів та оцінка ефективності застосування цифрового підпису. 3.4 Технічна реалізація та інтеграція цифрового підпису. 3. 5 Висновки до розділу 3. Розділ 4. Безпека життєдіяльності, основи охорони праці. 4.1 Охорона праці. 4.2 Захист людини від іонізуючого випромінювання. Висновки. Список використаних джерел.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)
Тема, Мета, Вступ. Аналіз існуючих підходів до захисту повідомлень у системах з наскрізним шифруванням. Обґрунтування вибору алгоритму Ed25519. Модель загроз та механізми захисту. Загальна архітектура рішення з інтеграцією цифрового підпису. Структура підписаного повідомлення та процес верифікації. Зовнішній вигляд додатку з індикацією статусу підпису. Висновки.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Основи охорони праці	Осухівська Г.М., зав. каф. КС		
Безпека в надзвичайних ситуаціях	Теслюк В.М., проректор з адміністративно-господарської роботи та будівництва		

7. Дата видачі завдання 18.09.2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	Ознайомлення з завданням до кваліфікаційної роботи	28.09.2025	*Виконано*
2.	Підбір та опрацювання джерел про цифровий підпис	30.09.2025–07.10.2025	*Виконано*
	у системах з наскрізним шифруванням		
3.	Оформлення розділу «Аналітична частина» (аналіз	09.10.2025 – 15.10.2025	*Виконано*
	існуючих підходів, обґрунтування алгоритмів, модель		
	загроз)		
4.	Проєктування архітектури системи та механізму	17.10.2025 – 21.10.2025	*Виконано*
	цифрового підпису		
5.	Розробка та впровадження модулів цифрового	24.10.2025 – 15.11.2025	*Виконано*
	підпису (генерація ключів, підпис, верифікація)		
6.	Впровадження механізму TOFU та виявлення Key	16.11.2025 – 23.11.2025	*Виконано*
	Substitution атак		
7.	Оформлення розділу «Основна частина»	24.11.2025 – 30.11.2025	*Виконано*
	(архітектура, проєктування, реалізація механізму		
	підпису)		
8.	Експериментальна перевірка стійкості системи до	01.12.2025 – 03.12.2025	*Виконано*
	атак на автентичність повідомлень		
9.	Оформлення розділу «Науково-дослідна частина»	04.12.2025 – 07.12.2025	*Виконано*
	(дослідження стійкості, експерименти, технічна		
	реалізація)		
10.	Виконання завдання до підрозділу «Безпека	08.12.2025 – 10.12.2025	*Виконано*
	життєдіяльності, основи охорони праці»		
11.	Оформлення кваліфікаційної роботи	11.12.2025 – 14.12.2025	*Виконано*
12.	Нормоконтроль	15.12.2025 – 18.12.2025	*Виконано*
13.	Перевірка на плагіат	19.12.2025 – 21.12.2025	*Виконано*
14.	Попередній захист кваліфікаційної роботи	22.12.2025 – 23.12.2025	*Виконано*
15.	Захист кваліфікаційної роботи	23.12.2025	

Студент _____ Пилипчук В.М.
(підпис) (прізвище та ініціали)

Керівник роботи _____ Загородна Н. В.
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Застосування цифрового підпису у системах з наскрізним шифруванням повідомлень // Кваліфікаційна робота ОР «Магістр» // Пилипчук Володимир Миколайович // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра кібербезпеки, група СБм-61 // Тернопіль, 2025 // С. 73, рис. – 6, табл. – 5, кресл. – - додат. – 1.

КЛЮЧОВІ СЛОВА: цифровий підпис, наскрізне шифрування, ED25519, WEB CRYPTO API, TOFU, автентичність, кібербезпека, веб-додаток.

Кваліфікаційна робота присвячена проєктуванню та реалізації механізму цифрового підпису у системах з наскрізним шифруванням повідомлень.

Перший розділ закладає теоретичні основи для розуміння проблематики забезпечення безпеки систем обміну повідомленнями. Проаналізовано сучасні підходи до захисту повідомлень у системах з наскрізним шифруванням та їх переваги та недоліки та сформульовано необхідність додаткових засобів захисту, а саме – автентифікації відправника. Обґрунтовано вибір алгоритму Ed25519 для реалізації цифрового підпису.

Другий розділ присвячено удосконаленню авторської системи з наскрізним шифруванням шляхом реалізації механізму цифрового підпису. Спроєктовано та представлено реалізацію модулів, процеси відправлення та перевірки підписаних повідомлень, а також інтеграцію з механізмом TOFU для захисту від підміни ключів.

Третій розділ зосереджено на експериментальній перевірці ефективності впровадженого механізму. У ньому проведено дослідження стійкості системи до атак на автентичність повідомлень, виконано моделювання типових сценаріїв атак та проаналізовано результати експериментів. Продемонстровано практичну цінність поєднання наскрізного шифрування з цифровим підписом у веб-орієнтованих системах обміну повідомленнями.

ABSTRACT

Implementation of a web application for confidential messaging // Thesis of educational level "Master"// Pylypchuk Volodymyr Mykolayovych // Ternopil Ivan Puluj National Technical University, Faculty of Computer Information Systems and Software Engineering, Department of Cybersecurity, group SBm-61 // Ternopil, 2025 // p. 73, figs. 6, tbls. 5, drws. - , apps. 1.

Keywords: digital signature, end-to-end encryption, ED25519, WEB CRYPTO API, TOFU, authenticity, cybersecurity, web application.

The thesis is devoted to the design and implementation of a digital signature mechanism in systems with end-to-end message encryption.

The first section lays the theoretical foundations for understanding the issues of ensuring the security of messaging systems. It analyzes modern approaches to message protection in end-to-end encryption systems, their advantages and disadvantages, and formulates the need for additional protection measures, namely sender authentication. The choice of the Ed25519 algorithm for digital signature implementation is justified.

The second section is devoted to improving the author's end-to-end encryption system by implementing a digital signature mechanism. The implementation of modules, the processes of sending and verifying signed messages, as well as integration with the TOFU mechanism for protection against key substitution are designed and presented.

The third section focuses on experimental verification of the effectiveness of the implemented mechanism. It studies the system's resistance to attacks on message authenticity, simulates typical attack scenarios, and analyzes the results of the experiments. The practical value of combining end-to-end encryption with digital signatures in web-based messaging systems is demonstrated.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

AES	— Advanced Encryption Standard
API	— Application Programming Interface
Base64	— Base64 encoding scheme
Curve25519	— Elliptic curve used in Ed25519
E2EE	— End-to-End Encryption
Ed25519	— Edwards-curve Digital Signature Algorithm
EdDSA	— Edwards-Curve Digital Signature Algorithm
GCM	— Galois/Counter Mode
HTTPS	— HyperText Transfer Protocol Secure
IndexedDB	— Indexed Database API
JSON	— JavaScript Object Notation
MITM	— Man-in-the-Middle
Node.js	— Node.js runtime environment
PKCS#8	— Public-Key Cryptography Standards #8
TLS	— Transport Layer Security
TOFU	— Trust On First Use
UUID	— Universally Unique Identifier
WebSocket	— WebSocket communication protocol

ВСТУП

Актуальність теми. Сучасний світ переживає стрімкий розвиток цифрових технологій, що супроводжується зростанням обсягу конфіденційної інформації, що обмінюється через мережу Інтернет. Події останніх років свідчать про те, що безпека даних стала однією з найбільш актуальних проблем сучасної інформаційної епохи. Чимало випадків порушення конфіденційності та витоку даних викликали серйозні обурення як серед користувачів, так і серед організацій, що збирають та обробляють особисті дані.

У відповідь на зростання кількості атак та витоків даних сучасні системи обміну повідомленнями дедалі частіше використовують механізми наскрізного шифрування (E2EE), які забезпечують конфіденційність переданих повідомлень. Однак аналіз показує, що конфіденційність сама по собі не розв'язує проблему автентичності та довіри до автора повідомлення. У більшості базових реалізацій E2EE система не має криптографічного механізму, який би однозначно підтверджував, що зашифроване повідомлення дійсно було створене конкретним користувачем, а не підмінене зловмисником у процесі передачі або під час атаки типу «людина посередині» (MITM).

Відсутність цифрового підпису в системах з наскрізним шифруванням створює низку потенційних вразливостей. Користувачі змушені покладатися на початкову довіру до механізму обміну ключами, не маючи змоги криптографічно перевірити незмінність особи співрозмовника у часі. У разі підміни публічного ключа зловмисник може встановити новий зашифрований канал, залишаючись непоміченим, що знижує загальний рівень довіри до системи. Окрему небезпеку становить тенденція до зростання кількості активних атак, спрямованих на підміну повідомлень та модифікацію даних у процесі передачі, що робить перевірену автентичність джерела критично важливою для забезпечення безпеки комунікацій.

Таким чином, задача створення ефективного механізму цифрового підпису, інтегрованого в системи з наскрізним шифруванням, здатна не лише суттєво підвищити рівень безпеки обміну повідомленнями, а й забезпечити

комплексний захист, що включає конфіденційність, автентичність, цілісність та незаперечність повідомлень.

Мета та задачі дослідження. Підвищення рівня безпеки авторської системи обміну повідомленнями з наскрізним шифруванням шляхом інтеграції механізму цифрового підпису, що забезпечує автентифікацію автора повідомлення, цілісність даних та стійкість до активних атак без порушення існуючої моделі конфіденційності.

Для досягнення поставленої мети потрібно виконати наступні завдання:

- Проаналізувати сучасні підходи до забезпечення безпеки систем обміну повідомленнями, обмеження базових реалізацій E2EE та обґрунтувати вибір криптографічних алгоритмів для реалізації цифрового підпису.
- Спроєктувати архітектуру модуля цифрового підпису в існуючу систему з наскрізним шифруванням, визначити структуру підписаних повідомлень та механізми зберігання криптографічних ключів.
- Реалізувати модульну систему цифрового підпису з використанням алгоритму Ed25519 та Web Crypto API, інтегрувати механізм TOFU для захисту від підміни публічних ключів.
- Реалізувати процеси відправлення та перевірки підписаних повідомлень, забезпечити обробку помилок та виявлення аномалій для підвищення стійкості системи до атак.
- Провести експериментальне дослідження стійкості системи до атак на цілісність повідомлень та оцінити ефективність впровадженого механізму цифрового підпису.

Об'єкт дослідження. Системи обміну повідомленнями з наскрізним шифруванням, механізми забезпечення автентичності та цілісності повідомлень, криптографічні алгоритми цифрового підпису та методи захисту від активних атак у веб-орієнтованих застосунках.

Предмет дослідження. Методи та механізми інтеграції цифрового підпису в системи з наскрізним шифруванням повідомлень для забезпечення автентичності, цілісності та незаперечності переданих даних.

Наукова новизна одержаних результатів кваліфікаційної роботи полягає в розробці комплексного рішення, яке поєднує наскрізне шифрування з цифровим підписом у веб-орієнтованій системі обміну повідомленнями. Система реалізує підхід sign-then-encrypt, при якому підписується зашифрований пакет повідомлення разом з метаданими, що забезпечує захист від модифікації як вмісту, так і службових даних. Впроваджено механізм TOFU з автоматичним виявленням Key Substitution атак та комплексним аналізом змін публічних ключів, що дозволяє ефективно протидіяти атакам типу MITM у веб-середовищі з обмеженими можливостями зберігання ключів.

Практичне значення одержаних результатів. Розроблена система може бути використана для забезпечення безпечного обміну повідомленнями з гарантією автентичності та цілісності даних. Система дозволяє користувачам перевіряти автентичність отриманих повідомлень та виявляти спроби підміни або модифікації даних, що суттєво підвищує рівень безпеки комунікацій. Реалізація демонструє практичну цінність поєднання наскрізного шифрування з цифровим підписом у веб-орієнтованих системах обміну повідомленнями та може служити основою для подальших досліджень та вдосконалення подібних рішень.

Апробація результатів магістерської роботи. Основні результати проведених досліджень обговорювались на XIII науково-технічній конференції «Інформаційні моделі, системи та технології» (м.Тернопіль), 17-18 грудня 2025 р.

Публікації. Основні результати кваліфікаційної роботи опубліковано у тезах конференції: В. Пилипчук, Н. Загородна Інтеграція цифрового підпису у системах з наскрізним шифруванням повідомлень // XIII науково-технічна конференція «Інформаційні моделі, системи та технології» (м.Тернопіль), 17-18 грудня 2025 р. (див. Додаток А).

РОЗДІЛ 1 АНАЛІТИЧНА ЧАСТИНА

1.1 Аналітичний огляд

Стрімкий розвиток цифрових комунікацій призвів до зростання ролі систем обміну повідомленнями у повсякденному житті користувачів. Такі системи обробляють значні обсяги персональних та конфіденційних даних, що робить питання інформаційної безпеки одним із ключових аспектів їх проєктування. У відповідь на зростання кількості атак, витоків даних та масового збору інформації, сучасні месенджери дедалі частіше використовують механізми наскрізного шифрування (End-to-End Encryption, E2EE).

Наскрізне шифрування забезпечує конфіденційність переданих повідомлень, гарантує, що їхній вміст доступний виключно кінцевим учасникам обміну, а сервер виконує лише роль транспортного посередника. Найпоширеніші реалізації E2EE базуються на асиметричних алгоритмах обміну ключами, зокрема Diffie–Hellman, у поєднанні з симетричними алгоритмами шифрування, такими як AES [1]. Такий підхід ефективно захищає дані від несанкціонованого доступу та пасивного перехоплення.

Однак аналіз сучасних систем безпечного обміну повідомленнями показує, що конфіденційність сама по собі не розв’язує проблему автентичності та довіри до автора повідомлення. У більшості базових реалізацій E2EE система не має криптографічного механізму, який би однозначно підтверджував, що зашифроване повідомлення дійсно було створене конкретним користувачем, а не підмінене зловмисником у процесі передачі або під час атаки типу «людина посередині» (MITM).

Відсутність цифрового підпису в системах з наскрізним шифруванням створює низку потенційних вразливостей. Зокрема, користувачі змушені покладатися на початкову довіру до механізму обміну ключами, не маючи змоги криптографічно перевірити незмінність особи співрозмовника у часі. У разі підміни публічного ключа зловмисник може встановити новий

зашифрований канал, залишаючись непоміченим, що знижує загальний рівень довіри до системи.

У сучасних захищених месенджерах, таких як Signal [4], WhatsApp або Matrix, ця проблема вирішується шляхом поєднання наскрізного шифрування з механізмами цифрового підпису. Цифровий підпис дозволяє забезпечити автентичність автора повідомлення, цілісність переданих даних та незаперечність факту відправлення. Таким чином, цифровий підпис доповнює E2EE, формуючи повноцінну модель безпечного обміну повідомленнями.

Аналітичний огляд показує, що ефективна система захищеного обміну повідомленнями повинна поєднувати щонайменше дві незалежні криптографічні властивості: конфіденційність даних і перевірену автентичність джерела. Реалізація лише одного з цих аспектів є недостатньою в умовах сучасних загроз. Саме тому впровадження цифрового підпису у системи з наскрізним шифруванням є логічним і необхідним етапом еволюції таких рішень.

У контексті даної магістерської роботи особливу увагу приділено аналізу архітектурних та криптографічних аспектів інтеграції цифрового підпису в уже існуючу систему з наскрізним шифруванням. Це дозволяє не лише підвищити рівень безпеки обміну повідомленнями, але й продемонструвати практичне застосування теоретичних принципів цифрового підпису в реальній веб-системі.

1.2 Аналіз існуючих підходів до захисту повідомлень у системах з наскрізним шифруванням

Сучасні системи обміну повідомленнями з наскрізним шифруванням використовують різні підходи до забезпечення безпеки передачі даних. Основна увага в таких системах традиційно зосереджена на забезпеченні конфіденційності повідомлень, тоді як питання автентичності автора та довіри до ключів часто реалізуються опосередковано або покладаються на початкову ініціалізацію каналу зв’язку.

Базовий підхід до реалізації E2EE полягає у використанні асиметричного обміну ключами для створення спільного секретного ключа з подальшим шифруванням повідомлень симетричними алгоритмами. У таких системах кожен користувач володіє парою ключів для шифрування, а сервер виступає лише як посередник для передачі зашифрованих даних. Даний підхід ефективно захищає вміст повідомлень від стороннього спостерігача, включно з серверною стороною.

Водночас аналіз подібних рішень показує, що відсутність цифрового підпису створює низку обмежень. Зокрема, користувач не має криптографічної гарантії того, що отримане повідомлення було створене саме тим користувачем, за якого він себе видає. У разі компрометації каналу обміну ключами або реалізації атаки типу MITM можливе встановлення нового зашифрованого з'єднання без відома кінцевих учасників.

Альтернативний підхід, який застосовується у більш зрілих системах захищеного обміну повідомленнями, полягає у поєднанні наскрізного шифрування з цифровим підписом. У такій архітектурі кожне повідомлення додатково підписується приватним ключем відправника, а отримувач має змогу перевірити підпис за допомогою відповідного публічного ключа. Це забезпечує цілісність повідомлення та підтверджує автентичність його джерела.

Важливою складовою таких систем є механізм управління публічними ключами підпису. Найпоширенішим підходом є модель TOFU (Trust On First Use), за якої перший отриманий публічний ключ вважається довіреним, а всі наступні його зміни фіксуються та потребують підтвердження користувача. Даний механізм дозволяє виявляти спроби підміни ключів без необхідності використання централізованих сертифікаційних центрів.

Окрему увагу в сучасних реалізаціях приділяють захисту від повторної передачі повідомлень (replay-атак). Для цього у структуру повідомлення додаються унікальні ідентифікатори та часові мітки, які також включаються в процес підпису. Це унеможливлює повторне використання раніше перехоплених повідомлень навіть за умови їхньої криптографічної коректності.

Порівняльний аналіз підходів показує, що системи, які використовують виключно наскрізне шифрування без цифрового підпису, забезпечують лише часткову модель безпеки. Вони захищають вміст повідомлень, але не гарантують стабільної автентичності співрозмовника у часі. Натомість поєднання E2EE з цифровим підписом дозволяє досягти комплексного захисту, що включає конфіденційність, цілісність, автентичність та стійкість до активних атак.

З огляду на це, доцільним є впровадження механізмів цифрового підпису в системи з наскрізним шифруванням, особливо у веб-орієнтованих застосунках, де ризики підміни ключів та активних атак є підвищеними. Саме такий підхід обрано у межах даної магістерської роботи для подальшого розвитку наявної системи.

1.3 Обґрунтування вибору криптографічних алгоритмів та засобів реалізації

Під час проєктування системи цифрового підпису для веб-орієнтованої системи обміну повідомленнями з наскрізним шифруванням ключовим завданням є вибір криптографічних алгоритмів та інструментів, які забезпечують високий рівень безпеки, продуктивності та сумісності з клієнтським середовищем. Веб-орієнтована система для обміну повідомленнями з наскрізним шифруванням була розроблена автором під час виконання кваліфікаційної роботи бакалавра. У межах даної магістерської роботи з метою забезпечення автентичності та цілісності повідомлень для реалізації механізму цифрового підпису було обрано алгоритм Ed25519 у поєднанні з використанням Web Crypto API.

Ed25519 є сучасним алгоритмом цифрового підпису, що базується на еліптичній кривій Curve25519 та схемі EdDSA [2]. Однією з ключових переваг Ed25519 є висока криптографічна стійкість за відносно невеликих розмірів ключів, що є критично важливим для клієнтських веб-застосунків. На відміну

від класичних алгоритмів, таких як RSA, Ed25519 забезпечує той самий або вищий рівень безпеки при значно меншому обсязі обчислень [8].

Важливим фактором вибору Ed25519 є його стійкість до типових помилок реалізації. Алгоритм не потребує генерації випадкових чисел для кожного підпису, що зменшує ризик компрометації приватного ключа через недостатню ентропію. Це є особливо актуальним для браузерного середовища, де якість джерел випадковості може відрізнятися залежно від платформи та пристрою.

Крім того, Ed25519 широко використовується у сучасних системах захищеного обміну повідомленнями та криптографічних протоколах, що підтверджує його практичну надійність та зрілість. Його застосування дозволяє досягти балансу між безпекою, швидкодією та простотою інтеграції у клієнтську архітектуру.

Для реалізації криптографічних операцій у браузері було обрано Web Crypto API [3] — стандартний інтерфейс, підтримуваний сучасними веб-браузерами. Використання Web Crypto API дозволяє виконувати криптографічні операції безпосередньо у клієнтському середовищі, не передаючи приватні ключі на сервер та не використовуючи сторонні JavaScript-бібліотеки з невідомим рівнем безпеки.

Перевагою Web Crypto API є те, що реалізація криптографічних алгоритмів виконується на рівні браузера, часто з використанням оптимізованих та апаратно прискорених механізмів. Це підвищує як продуктивність, так і загальний рівень безпеки, зменшуючи ризики, пов’язані з помилками ручної реалізації криптографії на рівні прикладного коду.

Обрання зв’язки Ed25519 та Web Crypto API також відповідає архітектурним вимогам системи . Оскільки система реалізована як веб-застосунок із клієнтською обробкою шифрування та підпису, використання стандартних браузерних засобів забезпечує кросплатформеність, спрощує розгортання та мінімізує залежність від серверної інфраструктури.

Таким чином, вибір алгоритму Ed25519 у поєднанні з Web Crypto API є обґрунтованим з точки зору безпеки, продуктивності та практичної реалізації. Даний підхід дозволяє реалізувати механізм цифрового підпису, який органічно

доповнює наскрізне шифрування та відповідає сучасним вимогам до захищених систем обміну повідомленнями.

В таблиці 1.1 наведено порівняння алгоритмів цифрового підпису, що дозволить наглядно побачити різницю між різними алгоритмами.

Таблиця 1.1 – Порівняння алгоритмів цифрового підпису

Характеристика	Ed25519	RSA-2048	ECDSA (P-256)
Розмір ключа	256 біт	2048 біт	256 біт
Розмір підпису	512 біт	2048 біт	512 біт
Швидкість підписання	Висока	Середня	Висока
Швидкість верифікації	Висока	Низька	Висока
Стійкість до помилок реалізації	Висока	Низька	Середня
Підтримка Web Crypto API	Так	Так	Так
Використання в E2EE системах	Signal, Matrix	Застаріле	Поширене
Криптографічна стійкість	~128 біт	~112 біт	~128 біт

З наведеного порівняння видно, що Ed25519 є найбільш збалансованим і сучасним алгоритмом цифрового підпису: він забезпечує високу швидкодію, компактні ключі та підписи, високу криптографічну стійкість і мінімальну схильність до помилок реалізації, що робить його оптимальним вибором для E2EE-систем.

1.4 Модель загроз та сценарії атак для системи

Для оцінки ефективності впровадження цифрового підпису в системі необхідно сформувати модель загроз, яка відображає потенційні сценарії атак у середовищі веб-орієнтованого застосунку з наскрізним шифруванням. Модель загроз дозволяє визначити, які саме ризики існують для системи, які з них можуть бути усунені криптографічними механізмами, а які виходять за межі технічного контролю.

У межах даної роботи розглядається як клієнт-серверна система, де сервер виконує роль транспортного посередника і не має доступу до відкритого вмісту повідомлень або приватних ключів користувачів. Основні криптографічні операції, зокрема шифрування та цифровий підпис, виконуються виключно на клієнтській стороні.

Однією з базових загроз для систем обміну повідомленнями є атака типу «людина посередині» (Man-in-the-Middle). У контексті така атака може бути реалізована шляхом підміни публічних ключів користувачів під час початкового обміну або повторного встановлення з’єднання. У разі відсутності механізму перевірки автентичності ключів зловмисник може створити окремі зашифровані канали з кожною стороною, залишаючись непоміченим.

Іншою суттєвою загрозою є підміна або модифікація повідомлень у процесі передачі. Навіть за наявності наскрізного шифрування зловмисник може намагатися вставити фальшиве повідомлення або змінити структуру переданого пакета. Без цифрового підпису отримувач не має змоги криптографічно перевірити цілісність та походження такого повідомлення.

Окрему категорію становлять атаки повторної передачі повідомлень (replay-атаки). У цьому сценарії зловмисник повторно надсилає раніше перехоплене зашифроване повідомлення, яке формально залишається криптографічно коректним. За відсутності механізмів унікалізації повідомлень це може призвести до дезінформації або некоректної обробки подій у системі.

Також слід враховувати загрозу компрометації серверної інфраструктури. Оскільки сервер не перевіряє підписи та не має доступу до приватних ключів, зловмисник, отримавши контроль над сервером, може лише впливати на доставку повідомлень або намагатися підміняти публічні ключі. Проте він не здатен розшифрувати вміст або створити валідні підписані повідомлення від імені користувачів.

У межах запропонованої архітектури цифровий підпис використовується як основний механізм захисту від активних атак. Підпис кожного повідомлення приватним ключем відправника забезпечує можливість перевірки його автентичності та цілісності на стороні отримувача. Будь-яка спроба модифікації

даних призводить до невдалої перевірки підпису та відхилення повідомлення. Модель загроз та відповідні механізми протидії наведено в таблиці 1.2, що дозволяє систематизувати можливі вектори атак і оцінити ефективність застосованих засобів захисту.

Таблиця 1.2 – Модель загроз та механізми захисту

Тип загрози	Опис атаки	Механізм захисту	Ефективність
MITM (підміна ключів)	Підміна публічного ключа під час обміну	TOFU з виявленням змін ключів	Висока
Підміна повідомлень	Вставка фальшивих повідомлень	Цифровий підпис кожного повідомлення	Висока
Модифікація даних	Зміна вмісту або метаданих	Підпис захищає весь пакет	Висока
Replay-атака	Повторна передача старих повідомлень	Timestamp + унікальний messageId	Висока
Компрометація сервера	Контроль над серверною інфраструктурою	Приватні ключі не передаються на сервер	Висока
Підробка підпису	Створення валідного підпису без приватного ключа	Криптографічна стійкість Ed25519	Висока
Витік приватних ключів	Доступ до локального сховища браузера	Зберігання в IndexedDB, обмежений доступ	Середня

Аналіз таблиці 1.2 підтверджує, що запропонована система забезпечує високий рівень захисту від основних активних загроз, характерних для E2EE-систем, включно з MITM-атаками, підміною та модифікацією повідомлень, повторною передачею даних і компрометацією серверної інфраструктури.

Для зменшення ризику MITM-атак використовується механізм TOFU, який фіксує публічний ключ підпису під час першого контакту між користувачами. У разі зміни ключа система повідомляє користувача про

потенційну загрозу, що дозволяє виявляти спроби підміни ключів навіть за наявності зашифрованого каналу.

Захист від replay-атак реалізується шляхом включення до підписаного повідомлення часової мітки та унікального ідентифікатора. Це унеможливлює повторне використання раніше отриманих повідомлень без порушення криптографічної цілісності пакета.

Слід зазначити, що запропонована модель загроз не охоплює сценарії повної компрометації клієнтського пристрою користувача. У разі отримання зловмисником доступу до приватних ключів на стороні клієнта будь-які криптографічні механізми втрачають свою ефективність. Такі сценарії виходять за межі даної роботи та потребують додаткових організаційних і апаратних заходів захисту.

Таким чином, побудована модель загроз демонструє, що впровадження цифрового підпису в системі істотно підвищує стійкість системи до активних атак, доповнюючи наскрізне шифрування механізмами автентичності, цілісності та контролю довіри до ключів.

1.5 Висновки до розділу 1

У межах аналітичної частини було розглянуто сучасні підходи до забезпечення безпеки систем обміну повідомленнями, зокрема систем з наскрізним шифруванням. Проведений аналіз показав, що наскрізне шифрування ефективно забезпечує конфіденційність переданих даних, однак саме по собі не гарантує автентичність джерела повідомлень та стійкість до активних атак.

Аналіз існуючих підходів засвідчив, що відсутність цифрового підпису в системах E2EE створює ризики підміни ключів, модифікації повідомлень та реалізації атак типу «людина посередині». Сучасні захищені месенджери вирішують цю проблему шляхом поєднання механізмів наскрізного шифрування з цифровим підписом та управлінням довірою до публічних ключів.

Обґрунтування вибору криптографічних алгоритмів підтвердило доцільність використання алгоритму Ed25519 як ефективного та безпечного рішення для реалізації цифрового підпису в клієнтському веб-середовищі. Застосування Web Crypto API дозволяє виконувати криптографічні операції безпосередньо на стороні клієнта, зберігаючи приватні ключі поза серверною інфраструктурою та відповідаючи принципам zero-knowledge архітектури.

Побудована модель загроз для системи визначила основні сценарії атак, зокрема MITM-атаки, підміну повідомлень та повторну передачу даних. Аналіз показав, що впровадження цифрового підпису у поєднанні з механізмами TOFU, унікальними ідентифікаторами повідомлень та часовими мітками істотно підвищує рівень захисту системи та дозволяє своєчасно виявляти спроби компрометації.

Отримані результати аналітичного розділу підтверджують актуальність і доцільність інтеграції цифрового підпису в систему . Вони слугують теоретичним і практичним підґрунтям для подальшого проєктування, реалізації та експериментальної перевірки запропонованого рішення у наступних розділах магістерської роботи.

РОЗДІЛ 2 ОСНОВНА ЧАСТИНА

2.1 Загальна архітектура рішення

Система спроєктована як веб-орієнтований клієнт-серверний застосунок для безпечного обміну повідомленнями з використанням наскрізного шифрування. Основною ідеєю архітектури є максимальне перенесення криптографічних операцій на клієнтську сторону з повним виключенням доступу серверної частини до відкритих даних та приватних ключів користувачів.

У межах даної магістерської роботи архітектура системи була розширена шляхом інтеграції механізму цифрового підпису, який органічно доповнює існуючу модель наскрізного шифрування та підвищує рівень автентичності й цілісності обміну повідомленнями.

Архітектурно система складається з таких основних компонентів:

- Клієнтська частина (Browser Client) – Відповідає за генерацію криптографічних ключів, шифрування та розшифрування повідомлень, створення та перевірку цифрових підписів, а також взаємодію з користувачем через веб-інтерфейс.
- Серверна частина (Message Relay Server) – Виконує роль транспортного вузла для передачі зашифрованих повідомлень і службових даних між клієнтами без доступу до їхнього вмісту.
- Клієнтське сховище ключів (Local Key Storage) Забезпечує зберігання криптографічних матеріалів на стороні клієнта з використанням механізмів браузера (IndexedDB).

Запропонована архітектура відповідає принципам zero-knowledge, згідно з якими сервер не володіє жодною інформацією, що дозволяє прочитати зміст повідомлень або підробити їх від імені користувачів.

При проєктуванні архітектури з підтримкою цифрового підпису були дотримані такі ключові принципи:

- Клієнтська криптографія.

- Розділення ролей ключів.
- Мінімальна довіра до сервера.
- Масштабованість і розширюваність.

Клієнтська криптографія – усі криптографічні операції, включно з генерацією ключів, шифруванням та підписом повідомлень, виконуються виключно на стороні клієнта.

Розділення ролей ключів – для шифрування повідомлень і для цифрового підпису використовуються окремі криптографічні пари ключів, що зменшує ризики компрометації та спрощує управління безпекою.

Мінімальна довіра до сервера – сервер не здійснює перевірку підписів і не бере участі у процесах автентифікації повідомлень, що виключає можливість підміни або модифікації даних на серверному рівні.

Масштабованість і розширюваність – архітектура системи дозволяє додавати нові криптографічні механізми або сценарії без суттєвих змін базової структури.

Цифровий підпис у системі розглядається як незалежний криптографічний шар, що накладається поверх механізму наскрізного шифрування. Такий підхід дозволяє: підтвердити автентичність відправника кожного повідомлення; гарантувати цілісність зашифрованого пакета даних; забезпечити захист від активних атак, зокрема підміни повідомлень та MITM-сценаріїв.

Важливою особливістю є те, що підписується не відкритий текст повідомлення, а сформований зашифрований пакет разом зі службовими метаданими, що виключає можливість вибіркової модифікації даних без порушення підпису.

Таким чином, загальна архітектура рішення із підтримкою цифрового підпису базується на клієнт-орієнтованій моделі безпеки, де сервер виконує суто допоміжну функцію доставки повідомлень. Запропонована структура дозволяє поєднати конфіденційність, автентичність та цілісність повідомлень у межах єдиного веб-застосунку без використання централізованих довірчих третіх сторін.

Клієнтська частина системи є ключовим елементом архітектури, оскільки саме на ній виконуються всі криптографічні операції, пов’язані з забезпеченням конфіденційності, автентичності та цілісності повідомлень. Реалізація клієнта у вигляді веб-застосунку дозволяє забезпечити кросплатформеність і доступність системи без встановлення додаткового програмного забезпечення.

Клієнтська частина системи побудована за модульним принципом і складається з логічно відокремлених компонентів, кожен з яких відповідає за конкретний аспект функціонування системи.

Основними модулями клієнта є:

– модуль управління користувачем;
– модуль обробки повідомлень;
– модуль цифрового підпису;
– модуль зберігання криптографічних ключів;
– модуль взаємодії з користувацьким інтерфейсом.

Такий підхід спрощує підтримку, тестування та подальше розширення функціональності системи. Архітектуру можна побачити на рисунку 2.1.

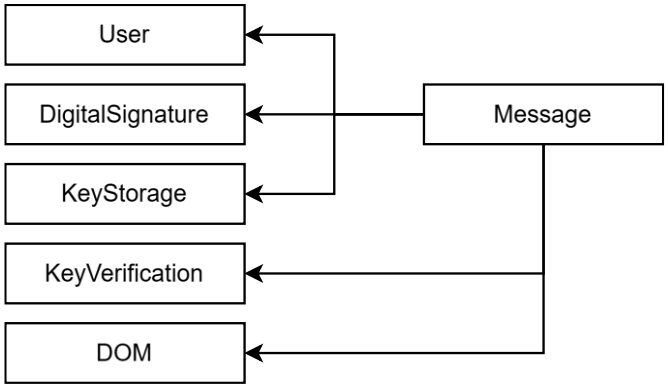

Рисунок 2.1 – Модульна архітектура клієнтської частини та взаємозв'язки між компонентами

Модуль управління користувачем (User) відповідає за ініціалізацію та управління криптографічними ключами користувача. Для виконання своїх функцій він використовує модуль цифрового підпису (DigitalSignature) для генерації пари ключів Ed25519 та модуль зберігання ключів (KeyStorage) для збереження приватних ключів у локальному сховищі. При ініціалізації

користувача модуль User викликає методи генерації ключів через DigitalSignature, після чого зберігає їх через KeyStorage. Модуль KeyStorage, у свою чергу, використовує функціональність DigitalSignature для експорту та імпорту ключів у форматі Base64, що необхідно для серіалізації криптографічних об'єктів перед збереженням у IndexedDB. Такий підхід забезпечує безпечне зберігання ключів у форматі, придатному для персистентного сховища браузера.

Модуль обробки повідомлень (Message) є центральним координатором системи та інтегрує функціональність всіх інших модулів для забезпечення повного циклу обробки повідомлень. При відправленні повідомлення модуль Message отримує приватний ключ користувача через модуль User, створює цифровий підпис через модуль DigitalSignature, перевіряє ключ отримувача через модуль KeyVerification та відображає повідомлення в інтерфейсі через модуль DOM. При отриманні повідомлення модуль Message отримує публічний ключ відправника через модуль Recipient, виконує верифікацію підпису через DigitalSignature, виявляє спроби підміни ключів через KeyVerification та оновлює інтерфейс через DOM. Така централізація логіки обробки повідомлень дозволяє забезпечити послідовність виконання всіх необхідних операцій та спрощує обробку помилок та аномальних ситуацій.

Модуль верифікації ключів (KeyVerification) реалізує механізм TOFU та виявлення Key Substitution атак, використовуючи функціональність інших модулів для отримання необхідної інформації. Для визначення необхідності ручної верифікації ключа модуль KeyVerification перевіряє історію відправлених повідомлень через модуль Message, що дозволяє автоматично верифікувати ключі співрозмовників, з якими вже був встановлений контакт. Для відображення діалогів TOFU та попереджень про зміну ключів модуль KeyVerification використовує модуль DOM, що забезпечує інтеграцію механізмів безпеки з користувацьким інтерфейсом. Модуль Recipient, який управляє публічними ключами співрозмовників, використовує KeyStorage для збереження ключів та DigitalSignature для імпорту публічних ключів з Base64 у формат CryptoKey, необхідний для верифікації підписів.

Модуль цифрового підпису (DigitalSignature) є незалежним криптографічним модулем, який інкапсулює всі операції з алгоритмом Ed25519 та не залежить від бізнес-логіки інших модулів системи. Єдина залежність модуля DigitalSignature полягає у використанні допоміжного модуля Tools для конвертації даних між форматами ArrayBuffer та Base64, що необхідно для роботи з Web Crypto API. Така ізоляція криптографічної логіки забезпечує високий рівень безпеки, спрощує тестування та аудит криптографічних операцій, а також дозволяє незалежно вдосконалювати або замінювати реалізацію криптографічних алгоритмів без впливу на інші компоненти системи.

Модуль взаємодії з користувацьким інтерфейсом (DOM) виконує роль представницького шару системи та використовує функціональність всіх основних модулів для відображення інформації користувачу. Модуль DOM отримує дані від модуля User для відображення інформації про поточного користувача, від модуля Message для відображення повідомлень та їх статусів верифікації, від модуля Recipient для відображення списку співрозмовників та їх статусів ключів, а також від модуля KeyVerification для відображення діалогів TOFU та попереджень про зміну ключів. Така архітектура забезпечує централізацію логіки відображення в одному модулі, що спрощує підтримку інтерфейсу та забезпечує консистентність відображення інформації користувачу.

Запропонована модульна архітектура забезпечує чітке розділення відповідальності між компонентами, де кожен модуль виконує конкретну функцію та використовує інші модулі виключно для отримання необхідної функціональності. Такий підхід спрощує підтримку коду, дозволяє незалежно тестувати кожен модуль, забезпечує можливість розширення функціональності без зміни базової архітектури та підвищує загальну надійність та безпеку системи за рахунок чіткого визначення меж відповідальності кожного компонента.

На стороні отримувача модуль повідомлень виконує перевірку цифрового підпису з використанням публічного ключа відправника. У разі невдалої

перевірки повідомлення відхиляється, а користувач отримує відповідне попередження.

Модуль цифрового підпису є окремим компонентом клієнтської частини, відповідальним за виконання криптографічних операцій підпису та перевірки. Він використовує можливості Web Crypto API для роботи з алгоритмом Ed25519.

До основних функцій модуля належать:

- генерація пари ключів цифрового підпису;
- створення цифрового підпису для переданих даних;
- перевірка коректності підпису отриманих повідомлень;
- експорт та імпорт публічних ключів у безпечному форматі.

Відокремлення даного модуля дозволяє мінімізувати ризик помилок у реалізації та спрощує повторне використання коду в інших компонентах системи.

Для збереження криптографічних матеріалів на клієнтській стороні використовується спеціалізований модуль локального зберігання ключів. Він реалізує взаємодію з механізмами браузера, зокрема IndexedDB, що дозволяє безпечно зберігати дані між сесіями користувача.

Приватні ключі цифрового підпису зберігаються виключно в зашифрованому вигляді або у форматі, недоступному для прямого читання прикладним кодом [5]. Це знижує ризики їх компрометації у випадку атак на рівні браузера.

Клієнтська частина системи тісно інтегрована з користувацьким інтерфейсом, який інформує користувача про стан безпеки з’єднання. Зокрема, інтерфейс відображає статус перевірки ключів співрозмовників, попередження про зміну публічних ключів та результати перевірки цифрових підписів.

Серверна частина системи виконує допоміжну роль у процесі обміну повідомленнями та не бере участі у виконанні криптографічних операцій. Такий підхід відповідає концепції мінімальної довіри до серверної інфраструктури та дозволяє знизити ризики, пов’язані з її компрометацією.

Основними завданнями серверної частини є:

– передача зашифрованих повідомлень між клієнтами;

– обробка службових типів повідомлень, зокрема обміну публічними ключами;

– маршрутизація повідомлень між активними користувачами;

– підтримка сесій з’єднання.

Сервер не здійснює розшифрування повідомлень і не має доступу до приватних ключів користувачів. Усі дані, що проходять через сервер, розглядаються як непрозорі зашифровані пакети.

У межах запропонованої архітектури сервер отримує вже сформований і підписаний пакет повідомлення. Він не перевіряє цифровий підпис і не модифікує вміст пакета, а лише передає його адресату у первісному вигляді.

Такий підхід дозволяє уникнути централізованої точки довіри; виключити можливість підміни або фальсифікації повідомлень на серверному рівні; зберегти принцип zero-knowledge для всієї системи.

У разі компрометації сервера зловмисник може впливати лише на доставку повідомлень або намагатися повторно надсилати зашифровані пакети, проте не здатен створити валідний цифровий підпис від імені користувачів.

Сервер також використовується як канал для обміну публічними ключами цифрового підпису між клієнтами. Для цього застосовується спеціальний тип службових повідомлень, який передається у зашифрованому вигляді.

Сервер не перевіряє достовірність переданих публічних ключів і не приймає рішень щодо їх довіри. Відповідальність за перевірку, збереження та валідацію ключів повністю покладається на клієнтську частину системи.

Взаємодія між клієнтом і сервером побудована за асинхронною моделлю, що дозволяє ефективно обробляти передачу повідомлень між багатьма користувачами. Клієнт ініціює всі криптографічні операції, тоді як сервер виконує лише функції пересилання та комунікації.

Така модель забезпечує масштабованість системи та дозволяє додавати нові функціональні можливості без зміни базових принципів безпеки.

Серверна частина не відповідає за:

– автентичність користувачів на криптографічному рівні;

– перевірку цифрових підписів;

– зберігання або управління приватними ключами;

– прийняття рішень щодо довіри до публічних ключів.

Ці обмеження є свідомими та обґрунтованими з точки зору побудови безпечної архітектури.

Функціонування системи з підтримкою цифрового підпису базується на чітко визначених потоках даних між клієнтами та сервером. Кожен етап обміну повідомленнями має власну зону відповідальності та супроводжується відповідними криптографічними операціями.

Процес відправлення повідомлення в системі починається на клієнтській стороні та включає такі послідовні етапи:

– Користувач створює повідомлення у клієнтському інтерфейсі.

– Повідомлення шифрується з використанням механізму наскрізного шифрування та ключів отримувача.

– До зашифрованого повідомлення додаються службові метадані, зокрема часову мітку та унікальний ідентифікатор повідомлення.

– Сформований пакет даних підписується приватним ключем цифрового підпису відправника.

– Підписаний пакет передається на сервер для подальшої доставки отримувачу.

На цьому етапі сервер отримує повністю сформований і криптографічно захищений пакет, не маючи можливості змінити його вміст без порушення підпису.

Після отримання пакета даних клієнт отримувача виконує такі дії:

– Приймає зашифрований пакет від сервера.

– Витягує публічний ключ цифрового підпису відправника або перевіряє його наявність у локальному сховищі.

– Перевіряє цифровий підпис повідомлення з використанням публічного ключа.

– У разі успішної перевірки виконує розшифрування вмісту повідомлення.

– Відображає повідомлення користувачу з відповідним статусом перевірки.

У разі невдалої перевірки підпису повідомлення відхиляється, а користувач отримує попередження про потенційну загрозу безпеці.

Обмін публічними ключами цифрового підпису здійснюється через спеціальні службові повідомлення. Під час першого контакту між користувачами публічний ключ підпису передається та зберігається в локальному сховищі отримувача відповідно до моделі TOFU.

У разі виявлення зміни публічного ключа система фіксує дану подію та інформує користувача, дозволяючи йому прийняти обґрунтоване рішення щодо довіри до нового ключа.

Інтеграція цифрового підпису безпосередньо у потоки даних дозволяє ефективно протидіяти активним атакам. Будь-яка спроба модифікації зашифрованого повідомлення, підміни службових метаданих чи повторної передачі повідомлення призводить до порушення цифрового підпису та виявляється на стороні клієнта.

Підпис у системі не є допоміжним або другорядним механізмом, а інтегрований у базові потоки даних системи. Він працює у тісному поєднанні з наскрізним шифруванням, утворюючи цілісну модель безпеки, яка забезпечує конфіденційність, автентичність і цілісність повідомлень.

Розглянута загальна архітектура рішення демонструє, що система з підтримкою цифрового підпису відповідає сучасним вимогам до безпечних веб-орієнтованих систем обміну повідомленнями. Чіткий розподіл відповідальності між клієнтом і сервером, використання клієнтської криптографії та інтеграція цифрового підпису у всі ключові потоки даних створюють надійне підґрунтя для подальшої реалізації та експериментальної перевірки системи.

2.2 Проєктування механізму цифрового підпису в додатку

Проєктування механізму цифрового підпису в додатку здійснюється з урахуванням уже наявної архітектури наскрізного шифрування та принципу

клієнтської криптографії. Основною метою впровадження цифрового підпису є забезпечення автентичності автора повідомлення, цілісності переданих даних та стійкості до активних атак без порушення існуючої моделі конфіденційності.

Цифровий підпис розглядається як незалежний криптографічний механізм, який працює паралельно з наскрізним шифруванням, але не замінює його. Шифрування відповідає за приховування вмісту повідомлення, тоді як цифровий підпис підтверджує його походження та незмінність.

У додатку цифровий підпис застосовується на рівні сформованого повідомлення, вже після його шифрування. Такий підхід дозволяє підписувати не лише вміст, а й службові метадані; унеможливити вибіркову модифікацію даних; забезпечити єдину точку перевірки цілісності повідомлення.

При проєктуванні механізму цифрового підпису були сформульовані такі ключові вимоги:

– Клієнтська генерація та зберігання ключів.
– Використання сучасних криптографічних алгоритмів.
– Підпис усього пакета повідомлення.
– Можливість перевірки без участі сервера.
– Підтримка виявлення підміни ключів.

Клієнтська генерація та зберігання ключів відповідає за приватні ключі цифрового підпису повинні створюватися та зберігатися виключно на стороні клієнта без передачі на сервер.

Використання сучасних криптографічних алгоритмів означає, що алгоритм підпису має забезпечувати високий рівень безпеки, ефективність у браузерному середовищі та стійкість до типових помилок реалізації.

Підпис усього пакета повідомлення означає що механізм має дозволяти фіксувати зміну публічних ключів та інформувати користувача про потенційні загрози.

Можливість перевірки без участі сервера означає що перевірка підпису здійснюється отримувачем локально, без звернення до серверної частини.

Підтримка виявлення підміни ключів означає що механізм має дозволяти фіксувати зміну публічних ключів та інформувати користувача про потенційні загрози.

Для цифрового підпису використовується окрема пара асиметричних ключів, яка не пов’язана з ключами наскрізного шифрування. Такий поділ дозволяє ізолювати криптографічні ролі ключів; спростити ротацію ключів підпису; зменшити наслідки можливої компрометації.

Публічний ключ цифрового підпису використовується для перевірки повідомлень іншими користувачами та передається через зашифрований канал у межах додатку.

Проєктування механізму цифрового підпису виконується з орієнтацією на практичну реалізацію в веб-середовищі, мінімізацію складності для користувача та прозору інтеграцію у вже існуючі потоки обміну повідомленнями. Усі криптографічні операції інкапсульовані в окремі логічні модулі, що підвищує читабельність, безпеку та розширюваність реалізації.

Важливим етапом проєктування механізму цифрового підпису є визначення структури повідомлення, яке підлягає підпису та передачі між клієнтами. У додатку підпис розглядається не як окрема сутність, а як невід’ємна частина повідомлення, що передається через комунікаційний канал.

Повідомлення в додатку формується у вигляді структурованого пакета даних, який включає як зашифрований вміст, так і службову інформацію, необхідну для забезпечення безпеки та коректної обробки. У практичній реалізації до складу такого пакета входять зашифрований вміст повідомлення, унікальний ідентифікатор повідомлення (crypto.randomUUID()), часову мітку створення (Date.now()), хеш логіну відправника (для забезпечення приватності), публічний ключ цифрового підпису відправника та цифровий підпис сформованого пакета даних.

Підпис формується на основі всіх полів пакета, за винятком самого поля підпису та поля публічного ключа. Це гарантує, що будь-яка зміна навіть одного байта даних призведе до порушення цілісності та виявляється на стороні отримувача. Використання хешу логіну замість реального логіну в підписі

забезпечує додатковий рівень приватності, оскільки сервер не має доступу до реальних ідентифікаторів користувачів. Реалізацію можна побачити на лістингу 2.1.

Лістинг 2.1 – Підпис пакета повідомлення

```
// Формування пакета для підпису після шифрування
const packetToSign = {
  timestamp: messageObj.timestamp,
  messageId: messageObj.id,
  senderLogin: User.hashName, // хеш логіну для приватності
  encryptedPayload: base64EncryptedMessage
};
// Створення цифрового підпису
const privateKey = User.getSigningPrivateKey();
const signature = await DigitalSignature.signMessage(
  privateKey,
  packetToSign
);
// Формування фінального підписаного пакета
const signedPacket = {
  ...packetToSign,
  signature: signature,
  senderPublicKey: await User.getSigningPublicKey()
};
```

Процес створення цифрового підпису виконується повністю на клієнтській стороні та включає етапи формування зашифрованого повідомлення, додавання службових метаданих до пакета, серіалізація пакета даних у детермінованому форматі, створення цифрового підпису з використанням приватного ключа та додавання підпису до пакета та передача його на сервер.

Особливу увагу приділено детермінованості серіалізації даних, оскільки будь-які відмінності у форматуванні можуть призвести до некоректної перевірки підпису.

На стороні отримувача перевірка цифрового підпису виконується до розшифрування вмісту повідомлення. Такий порядок дій дозволяє відхиляти потенційно небезпечні або підроблені повідомлення ще до їхньої обробки.

Процес перевірки включає:

– витяг публічного ключа цифрового підпису;

– перевірку його відповідності збереженому ключу співрозмовника;

– валідацію цифрового підпису отриманого пакета;

– перевірку актуальності часової мітки та унікальності ідентифікатора.

Лише після успішного проходження всіх етапів повідомлення вважається валідним і допускається до розшифрування.

Для забезпечення довіри до публічних ключів цифрового підпису використовується модель TOFU. Під час першого контакту між користувачами отриманий публічний ключ зберігається локально та вважається довіреним. Усі наступні повідомлення перевіряються з використанням цього ключа.

У разі виявлення зміни публічного ключа система фіксує дану подію та повідомляє користувача про потенційну загрозу. Користувач має змогу підтвердити або відхилити новий ключ, що дозволяє зберегти баланс між зручністю та безпекою.

Проєктування механізму цифрового підпису передбачає можливість ротації ключів у майбутньому. Для цього в локальному сховищі зберігається історія публічних ключів співрозмовників, що дозволяє виявляти підозрілі зміни ключів; аналізувати історію довіри та мінімізувати ризики підміни ключів у часі.

Механізм цифрового підпису тісно інтегрований з інтерфейсом додатку. Користувач отримує візуальні індикатори, що відображають статус перевірки ключа співрозмовника та коректність підпису кожного повідомлення.

Візуальні індикатори статусу підпису зображені на рисунку 2.2.

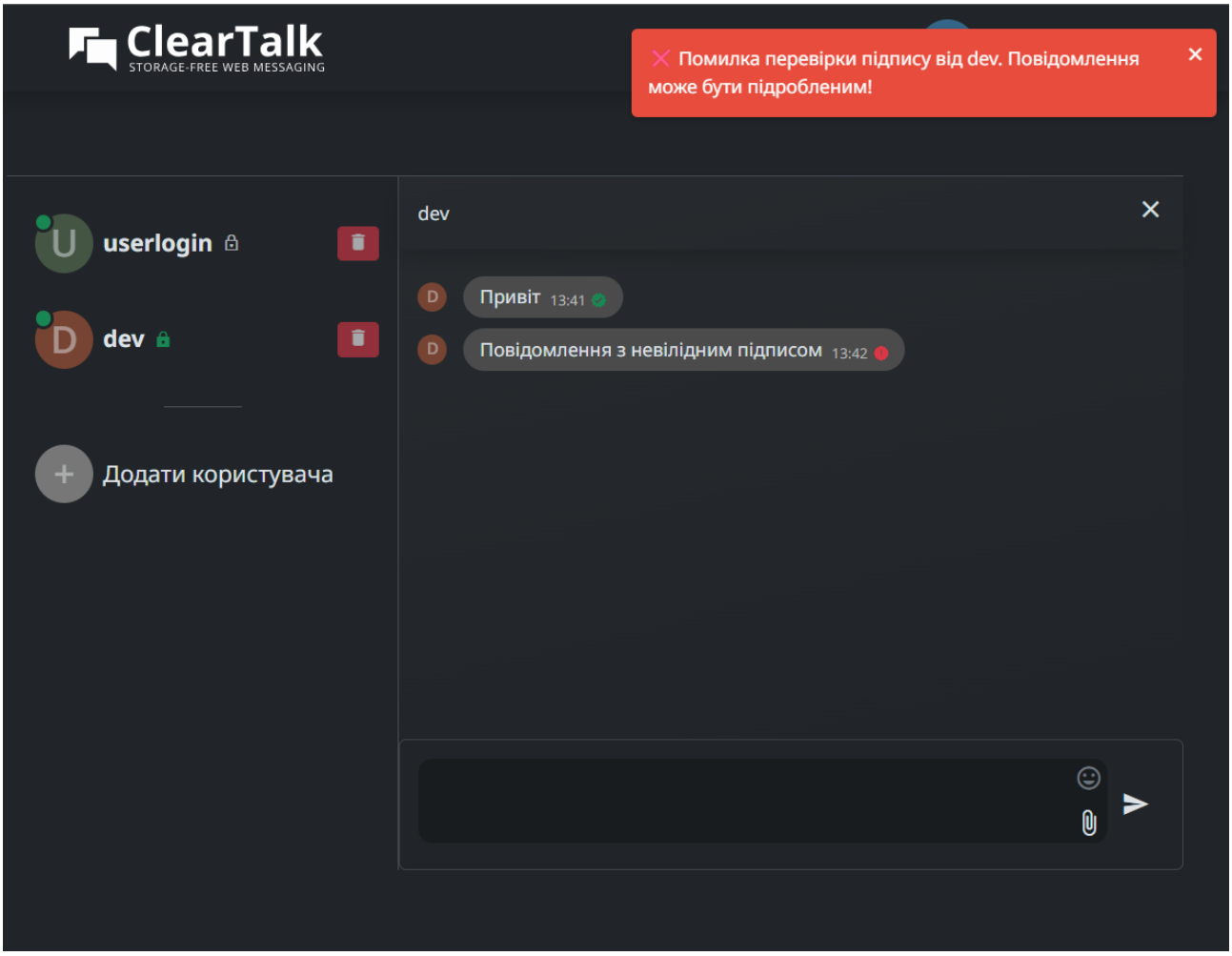

Рисунок 2.2 – Візуальні індикатори статусу підпису в повідомленнях (валідний та невалідний підпис)

Такий підхід дозволяє зробити криптографічні механізми прозорими для користувача, не перевантажуючи його технічними деталями, але водночас зберігаючи можливість усвідомленого контролю довіри [13].

Таким чином, проєктування механізму цифрового підпису в додатку ґрунтується на принципах клієнтської криптографії, мінімальної довіри до серверної частини та інтеграції безпеки у базові потоки обміну повідомленнями. Запропонований підхід забезпечує автентичність, цілісність та стійкість до активних атак без порушення властивостей наскрізного шифрування.

2.3 Проєктування зберігання та захисту криптографічних ключів

Захист криптографічних ключів є одним із найкритичніших аспектів безпеки будь-якої системи обміну повідомленнями. Навіть найстійкіші алгоритми шифрування та цифрового підпису втрачають свою ефективність у разі компрометації приватних ключів. Тому проєктування механізму зберігання та захисту ключів у додатку розглядається як окреме та принципово важливе завдання.

У межах додатку використовуються декілька незалежних типів криптографічних ключів, кожен з яких має власне призначення та рівень чутливості:

- приватні ключі цифрового підпису;
- публічні ключі цифрового підпису інших користувачів;
- ключі наскрізного шифрування повідомлень;
- допоміжні сесійні ключі.

Найвищий рівень захисту потребують саме приватні ключі цифрового підпису, оскільки їх компрометація дозволяє зловмиснику створювати валідні підписані повідомлення від імені користувача.

Класифікацію криптографічних ключів, їх призначення та рівні захисту наведено в таблиці 2.1, де окремо виділено приватні ключі цифрового підпису як критичний елемент безпеки системи, а також показано відмінності у підходах до зберігання та захисту інших типів ключового матеріалу.

Таблиця 2.1 – Типи криптографічних ключів та рівні захисту

Тип ключа	Призначення	Рівень чутливості	Місце зберігання	Механізми захисту
1	2	3	4	5
Приватний ключ підпису	Створення підписів	Критичний	IndexedDB (PKCS#8)	Локальне зберігання
Публічний ключ цифрового підпису	Верифікація підписів	Низький	IndexedDB з метаданими	TOFU, історія змін, виявлення підміни

Продовження таблиці 2.1

1	2	3	4	5
Ключі наскрізного шифрування	Шифрування повідомлень	Високий	Пам'ять (сесійні)	Diffie-Hellman, не зберігаються постійно
Допоміжні сесійні ключі	Тимчасові операції	Середній	Пам'ять	Автоматичне знищення після сесії

Аналіз таблиці 2.1 демонструє, що архітектура системи застосовує диференційований підхід до захисту криптографічних ключів залежно від їх чутливості та функціонального призначення. Приватні ключі цифрового підпису ізольовані від серверної інфраструктури та зберігаються виключно на стороні клієнта, що мінімізує ризик масової компрометації.

При проєктуванні механізму зберігання ключів були враховані такі основні загрози:

- несанкціонований доступ до локального сховища браузера;
- виконання шкідливого JavaScript-коду у контексті додатку;
- атаки через сторонні бібліотеки або залежності;
- витік даних у разі компрометації серверної частини;
- фізичний доступ до пристрою користувача.

З огляду на веб-орієнтовану природу додатку, особлива увага приділяється загрозам, пов’язаним із виконанням коду у браузері.

Для мінімізації зазначених ризиків у додатку застосовуються такі принципи:

- Локальність зберігання;
- мінімальна доступність;
- розділення ролей;
- стійкість до компрометації сервера.

Локальність зберігання – приватні ключі ніколи не передаються на сервер і не покидають межі клієнтського середовища.

Мінімальна доступність – ключі не повинні бути доступні у відкритому вигляді прикладному коду без необхідності.

Розділення ролей – ключі цифрового підпису та ключі шифрування зберігаються та використовуються незалежно один від одного.

Стійкість до компрометації сервера – навіть у разі повного контролю над сервером зловмисник не повинен отримати доступ до приватних ключів.

У веб-застосунках традиційні підходи до зберігання криптографічних ключів, характерні для десктопних або мобільних рішень, не можуть бути застосовані безпосередньо. Це зумовлює необхідність використання вбудованих механізмів браузера, які забезпечують базовий рівень ізоляції та безпеки [1].

З урахуванням цих обмежень у додатку обрано підхід до зберігання ключів, що поєднує локальне сховище браузера та використання стандартних криптографічних API для роботи з ключами у захищеному контексті.

Зберігання криптографічних ключів у веб-застосунках має низку принципових обмежень, пов’язаних з особливостями браузерного середовища. На відміну від нативних застосунків, веб-додаток не має прямого доступу до захищених апаратних сховищ або файлової системи користувача, що суттєво впливає на підходи до управління ключами.

Браузер забезпечує виконання коду у пісочниці, що з одного боку обмежує доступ до системних ресурсів, а з іншого — створює додатковий рівень ізоляції. Криптографічні операції виконуються через стандартизовані API, а прямий доступ до пам’яті та файлової системи відсутній.

Ці особливості визначають як переваги, так і обмеження з точки зору безпеки знижується ризик прямого доступу до даних сторонніми процесами, водночас зростає залежність від безпеки самого браузера, ключі потенційно можуть бути доступні у контексті виконання JavaScript-коду.

Для довготривалого зберігання криптографічних ключів у додатку використовується локальне сховище браузера, яке забезпечує збереження даних між сесіями користувача. Серед доступних механізмів було обрано IndexedDB, оскільки воно надає такі переваги: можливість зберігання складних об’єктів і

великих обсягів даних; асинхронний доступ без блокування основного потоку; кращий контроль над структурою та життєвим циклом даних порівняно з простими сховищами та підтримка всіма сучасними браузерами.

IndexedDB розглядається не як повноцінне захищене сховище, а як компромісне рішення, що відповідає вимогам веб-середовища та дозволяє реалізувати клієнтську криптографію без передачі ключів на сервер.

Порівняння основних підходів до зберігання криптографічних ключів у веб-застосунках наведено в таблиці 2.2, де проаналізовано їх переваги, обмеження та доцільність використання з точки зору безпеки та практичної реалізації E2EE.

Таблиця 2.2 – Порівняння підходів до зберігання ключів у веб-середовищі

Підхід	Переваги	Недоліки	Застосування
IndexedDB	Великі обсяги даних, асинхронний доступ, структуровані дані	Немає вбудованого шифрування	Зберігання приватних та публічних ключів
localStorage	Простота використання, синхронний доступ	Обмежений обсяг (~5-10 МБ), тільки рядки	Не використовуєтьс я
sessionStorag e	Автоматичне очищення після сесії	Втрата даних при закритті вкладки	Не використовуєтьс я
Web Crypto API (CryptoKey)	Захищений контекст, апаратне прискорення	Не зберігається між сесіями	Робота з ключами в пам'яті
Cookies	Автоматична передача на сервер	Обмежений розмір, передача на сервер	Не використовуєтьс я

Аналіз таблиці 2.2 показує, що жоден зі стандартних механізмів зберігання в браузері не забезпечує повноцінного захисту приватних ключів на рівні спеціалізованих апаратних сховищ. Водночас IndexedDB є найбільш

придатним варіантом для зберігання ключів у веб-середовищі завдяки підтримці великих обсягів даних, асинхронному доступу та відсутності автоматичної передачі інформації на сервер. Поєднання IndexedDB зі зберіганням ключів у форматі PKCS#8 та використанням Web Crypto API для криптографічних операцій дозволяє досягти прийнятного балансу між безпекою, функціональністю та обмеженнями браузерної платформи.

Приватні ключі цифрового підпису зберігаються у форматі, який мінімізує можливість їх випадкового або навмисного витоку. У процесі проєктування враховується можливість додаткового захисту приватних ключів, зокрема шляхом їх шифрування перед збереженням у локальному сховищі.

Такий підхід дозволяє знизити ризики компрометації ключів у разі доступу до локальних даних браузера, хоча й не забезпечує абсолютного захисту у випадку повного контролю над клієнтським середовищем.

Публічні ключі цифрового підпису інших користувачів зберігаються окремо від приватних ключів та супроводжуються додатковими метаданими, зокрема інформацією про час отримання та історію змін. Це дозволяє реалізувати механізми виявлення підміни ключів і підтримувати модель довіри на основі TOFU.

Попри використання стандартних механізмів браузера, клієнтське зберігання криптографічних ключів має низку обмежень. Зокрема, воно не захищає від атак у разі виконання шкідливого коду у контексті додатку або повної компрометації пристрою користувача.

Ці обмеження визнаються в межах даної роботи як такі, що не можуть бути повністю усунені без використання апаратних або платформозалежних рішень.

Проєктування механізму зберігання криптографічних ключів у додатку неможливе без чіткого визначення їх життєвого циклу. Життєвий цикл ключа охоплює всі етапи його існування — від генерації до знищення — і визначає правила, за якими ключ вважається дійсним або скомпрометованим.

Генерація ключів цифрового підпису відбувається виключно на клієнтській стороні під час першого використання додатку або у разі

відсутності збережених ключів. Процес генерації базується на стандартних криптографічних механізмах браузера, що забезпечують достатній рівень ентропії та стійкість до прогнозування.

Згенеровані ключі одразу прив’язуються до локального середовища користувача, що унеможливлює їх повторне використання на інших пристроях без явної дії з боку користувача. Генерація пари ключів показана в лістингу 2.2.

Лістинг 2.2 – Генерація пари ключів для цифрового підпису

```
// Генерація пари ключів Ed25519 через Web Crypto API
static async generateKeyPair() {
  const keyPair = await window.crypto.subtle.generateKey(
    {
      name: "Ed25519",
    },
    true, // extractable для можливості експорту
    ["sign", "verify"] // дозволені операції
  );
  return keyPair;
}

// Експорт приватного ключа для зберігання в IndexedDB
static async exportPrivateKey(privateKey) {
  const exported = await window.crypto.subtle.exportKey(
    "pkcs8", // формат для приватних ключів
    privateKey
  );
  return Tools.arrayBufferToBase64(exported);
}
```

Після генерації приватний ключ використовується виключно для створення цифрових підписів і не передається іншим компонентам системи без необхідності. Публічний ключ застосовується для перевірки підписів та передається іншим користувачам у межах захищеного каналу.

Під час нормальної роботи додатку ключі перебувають у пасивному стані та використовуються лише у момент створення або перевірки підпису, що зменшує поверхню атаки.

Архітектура додатку передбачає можливість ротації криптографічних ключів у майбутньому. Ротація може бути ініційована користувачем або вимушеною у разі підозри на компрометацію.

Для забезпечення прозорості змін у додатку зберігається історія публічних ключів співрозмовників. Це дозволяє фіксувати моменти зміни ключів; виявляти потенційно небезпечні сценарії; інформувати користувача про необхідність повторної перевірки довіри.

У разі виходу користувача з додатку, перевстановлення браузера або очищення локального сховища приватні ключі можуть бути знищені. У такому випадку нові ключі генеруються автоматично, що призводить до зміни криптографічної ідентичності користувача.

Такий сценарій розглядається як допустимий і контрольований, за умови, що інші користувачі будуть проінформовані про зміну ключа та зможуть підтвердити або відхилити нову довіру.

Важливо зазначити, що запропонована модель захисту криптографічних ключів не претендує на абсолютну безпеку. У разі повної компрометації клієнтського пристрою, виконання шкідливого коду або фізичного доступу до системи користувача будь-які програмні механізми захисту можуть бути обійдені.

У межах даної роботи такі сценарії розглядаються як такі, що виходять за межі відповідальності веб-застосунку та потребують додаткових апаратних або організаційних заходів безпеки.

Таким чином, проєктування зберігання та захисту криптографічних ключів у додатку базується на реалістичній моделі загроз, принципах клієнтської криптографії та чіткому визначенні життєвого циклу ключів. Запропонований підхід дозволяє мінімізувати ризики компрометації ключів у межах можливостей веб-середовища та створює надійне підґрунтя для подальшої реалізації механізмів цифрового підпису.

2.4 Реалізація механізму цифрового підпису та взаємодія модулів

Реалізація механізму цифрового підпису в додатку спрямована на забезпечення автентичності, цілісності та незаперечності повідомлень у системі з наскрізним шифруванням. На відміну від шифрування, яке захищає зміст повідомлення, цифровий підпис гарантує, що повідомлення було створене конкретним відправником і не було змінене під час передачі.

Механізм цифрового підпису інтегрований у клієнтську частину додатку та функціонує незалежно від серверної логіки. Сервер використовується виключно як транспортний рівень і не бере участі у створенні або перевірці підписів.

Цифровий підпис застосовується до вже зашифрованого повідомлення, що дозволяє зберегти властивості наскрізного шифрування, уникнути розкриття вмісту повідомлення на сервері та забезпечити перевірку цілісності всього пакета даних. Таким чином, цифровий підпис доповнює існуючий механізм захисту, не змінюючи його базових принципів.

Архітектура модулів цифрового підпису реалізована як набір взаємопов'язаних модулів, кожен з яких виконує чітко визначену функцію. Модуль `DigitalSignature` відповідає за криптографічні операції: генерацію пари ключів Ed25519 через Web Crypto API, створення цифрового підпису для об'єктів повідомлень, перевірку валідності підпису з використанням публічного ключа, а також експорт та імпорт ключів у форматі Base64 для зберігання. Цей модуль не має доступу до мережевої логіки або інтерфейсу користувача, що зменшує ризик витоку ключів і спрощує аудит безпеки.

Модуль `User` управляє особистими ключами користувача: ініціалізує ключі при вході в систему (генерація або завантаження з IndexedDB), зберігає приватний ключ виключно в пам'яті під час сесії та надає доступ до ключів для операцій підпису. Модуль користувача взаємодіє з модулем цифрового підпису для виконання операцій підписання, а також із модулем зберігання ключів для їх збереження та завантаження між сесіями.

Модуль `Message` інтегрує підпис у процес обміну повідомленнями: формує підписаний пакет перед відправкою, перевіряє підпис при отриманні повідомлення та відображає статус верифікації в інтерфейсі користувача. Саме на цьому рівні формується остаточна структура повідомлення, що включає метадані, зашифрований вміст та цифровий підпис.

Модуль `KeyVerification` реалізує механізм TOFU: показує діалог перевірки ключа при першому контакті, виявляє зміну публічного ключа (Key Substitution атака) та зберігає історію ключів для аналізу підозрілих змін. У разі отримання повідомлення з новим або зміненим публічним ключем цей компонент ініціює відповідні дії, зокрема відображення попереджень або блокування довіри до повідомлення.

Модуль зберігання ключів відповідає за фізичне збереження криптографічних матеріалів у локальному сховищі браузера. Він виступає як абстракція над IndexedDB і забезпечує централізований контроль доступу до ключів. Завдяки такій архітектурі прикладна логіка не має прямого доступу до механізмів зберігання, що зменшує ризик помилок реалізації та витоку даних.

Процес відправлення підписаного повідомлення включає послідовність етапів. Спочатку створюється об'єкт повідомлення з унікальним ідентифікатором (crypto.randomUUID()) та часовою міткою (Date.now()). Потім повідомлення шифрується з використанням спільного ключа Diffie-Hellman, отриманого в процесі наскрізного шифрування. Після шифрування формується пакет для підпису, який включає часову мітку, унікальний ідентифікатор повідомлення, хеш логіну відправника (для забезпечення приватності) та зашифрований вміст у форматі Base64. Важливо, що підписується саме зашифрований пакет, а не відкритий текст, що забезпечує захист від модифікації як вмісту, так і метаданих.

Цифровий підпис створюється приватним ключем користувача через метод `DigitalSignature.signMessage()`, який виконує серіалізацію об'єкта в JSON-рядок та створює підпис за алгоритмом Ed25519. Підпис та публічний ключ відправника додаються до пакета, після чого сформований підписаний

пакет передається на сервер у вигляді JSON-рядка. Метод підписання об’єкта повідомлення показано в лістингу 2.3.

Лістинг 2.3 – Метод підписання об’єкта повідомлення

```
static async signMessage(privateKey, messageObj) {
  // Виключення поля signature з об'єкта для підпису
  const { signature, ...dataToSign } = messageObj;
// Серіалізація в JSON для детермінованого підпису
  const dataString = JSON.stringify(dataToSign);

  // Створення підпису за алгоритмом Ed25519
  const signatureBuffer = await window.crypto.subtle.sign(
    "Ed25519",
    privateKey,
    new TextEncoder().encode(dataString)
  );

  // Конвертація підпису в Base64
  return Tools.arrayBufferToBase64(signatureBuffer);
}
```

При отриманні повідомлення виконується послідовність дій, спрямованих на перевірку автентичності та цілісності. Спочатку сервер передає підписаний пакет як JSON-рядок, який парситься на клієнтській стороні. Якщо в пакеті міститься новий публічний ключ відправника, система зберігає його та порівнює зі збереженим. У разі виявлення зміни ключа виконується аналіз на Key Substitution атаку: перевіряється, чи ключ був раніше верифікований, чи новий ключ був у історії (rollback атака), а також аналізується частота змін ключів. При виявленні підозрілих змін показується попередження з відповідним рівнем критичності.

Перевірка цифрового підпису виконується до розшифрування повідомлення. Система отримує збережений публічний ключ відправника та виконує верифікацію підпису через метод `DigitalSignature.verifyMessage()`. Для перевірки формується об'єкт з тих самих полів, що були підписані: часової

мітки, ідентифікатора повідомлення, хешу логіну та зашифрованого вмісту. Якщо підпис валідний, повідомлення розшифровується та відображається користувачу з візуальним індикатором "verified". Якщо підпис невалідний, система показує попередження про можливу підробку, а повідомлення може бути заблоковане залежно від налаштувань безпеки. Відображення повідомлення з валідним цифровим підписом показано на рисунку 2.3

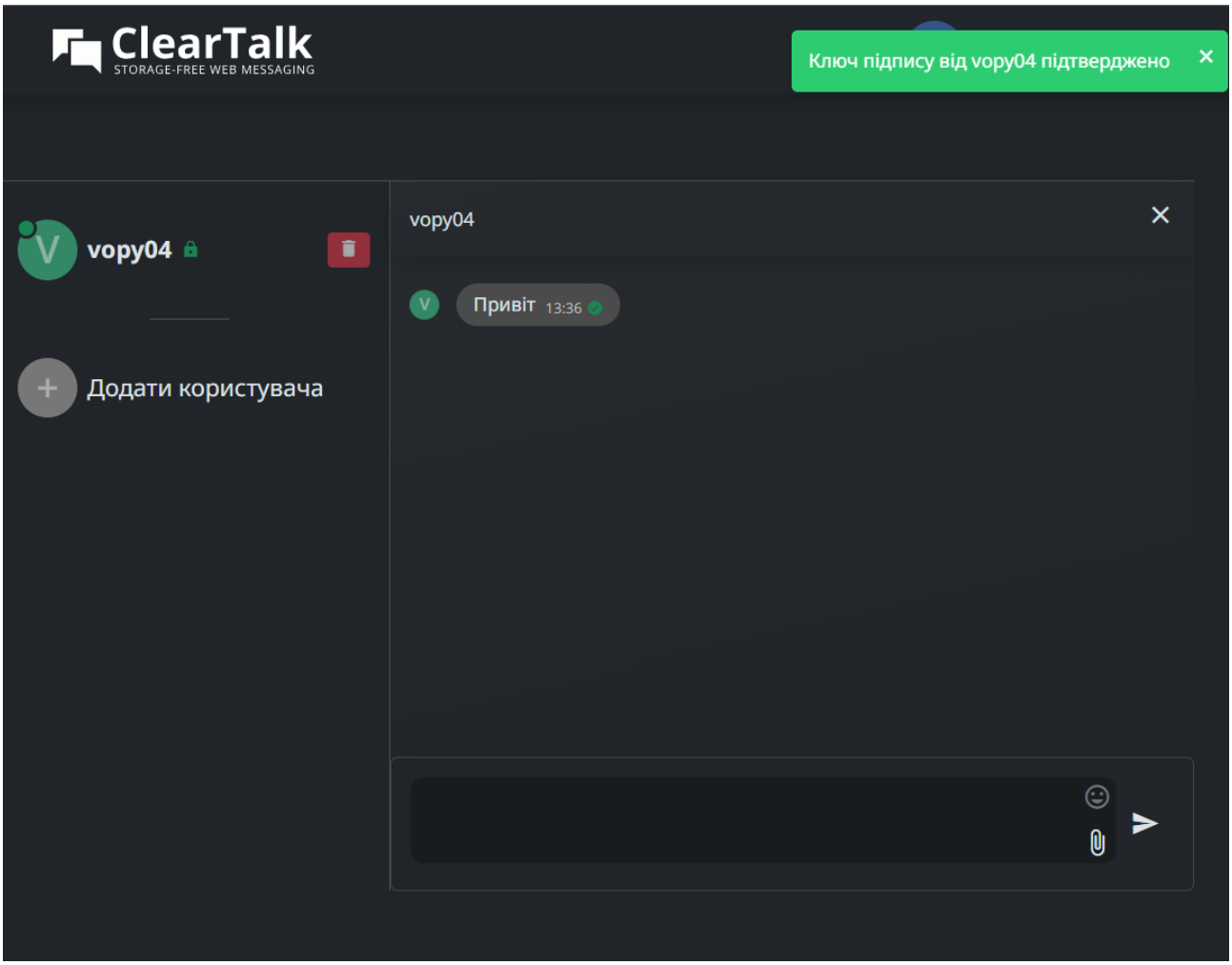

Рисунок 2.3 – Відображення повідомлення з валідним цифровим підписом

Реалізація отримання та верифікації цифрового підпису можна побачити в лістингу 2.4.

Лістинг 2.4 – Отримання та верифікація цифрового підпису

```
// Отримання публічного ключа відправника
const publicKey = await recipient.getSigningPublicKey();
// Верифікація цифрового підпису
```

```
const isValid = await DigitalSignature.verifyMessage(
publicKey,
  {
    timestamp: signedPacket.timestamp,
    messageId: signedPacket.messageId,
    senderLogin: signedPacket.senderLogin,
    encryptedPayload: signedPacket.encryptedPayload,
    signature: signedPacket.signature
  }
);
// Обробка результату верифікації
if (isValid) {
  // Підпис валідний - розшифрування та відображення
  const decryptedContent = await Message.decrypt(
    signedPacket.encryptedPayload,
    senderHashName
  );
  message.signatureValid = true;
} else {
  // Підпис невалідний - блокування повідомлення
  message.signatureValid = false;
  showSecurityWarning("Підпис повідомлення невалідний");
}
```

Інтеграція з механізмом TOFU забезпечує захист від підміни публічних ключів. При першому контакті між користувачами отримувач отримує публічний ключ відправника. Система перевіряє, чи поточний користувач вже відправляв повідомлення цьому отримувачу. Якщо так, ключ автоматично верифікується на основі принципу взаємної довіри. Якщо ні, показується діалог TOFU для ручного підтвердження ключа, де користувач може перевірити ключ через безпечний канал (наприклад, QR-код) та підтвердити його достовірність. Діалог TOFU для верифікації публічного ключа можна побачити на рисунку 2.4. При зміні публічного ключа система виконує комплексний аналіз на Key Substitution атаку. Перевіряється, чи ключ був раніше верифікований, чи новий

ключ був у історії (що може свідчити про rollback атаку), а також аналізується частота змін ключів. На основі результатів аналізу показується попередження з відповідним рівнем критичності: низький, середній, високий або критичний.

Попередження Key Substitution атаку можна побачити на рисунку 2.5.

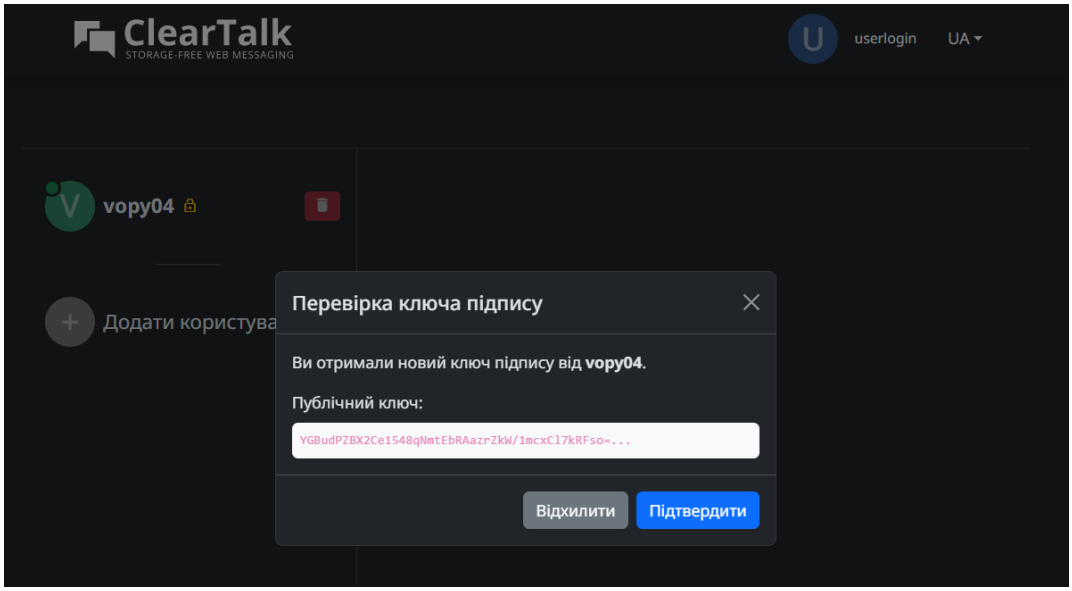

Рисунок 2.4 – Діалог TOFU для верифікації публічного ключа

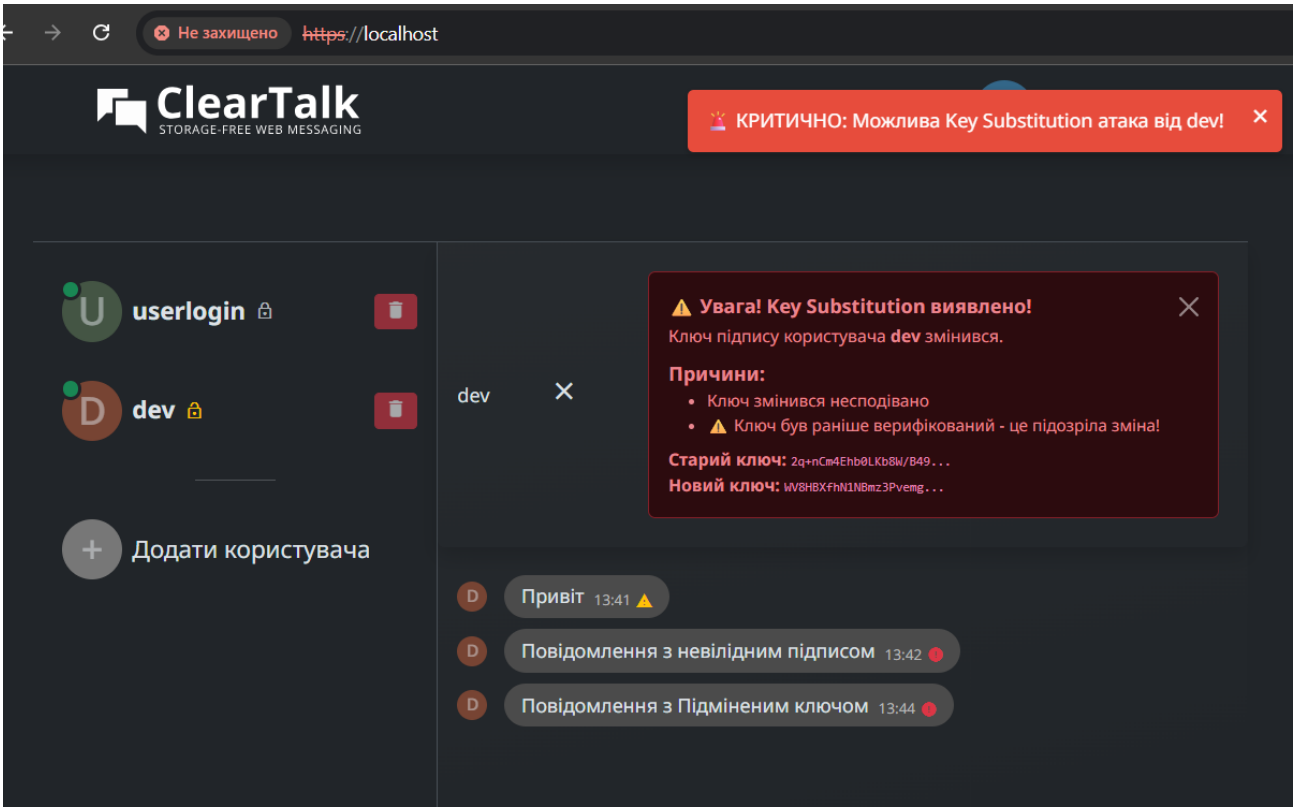

Рисунок 2.5 – Попередження про зміну публічного ключа (Key Substitution атака)

Приватні ключі цифрового підпису зберігаються в IndexedDB [9] у форматі PKCS#8 [10] (зашифрований формат) та завантажуються в пам'ять тільки під час активної сесії користувача. Приватні ключі ніколи не передаються на сервер та не покидають межі клієнтського середовища. Публічні ключі інших користувачів зберігаються в IndexedDB разом з метаданими: часом отримання, статусом верифікації (verified/unverified) та історією змін ключів. Ця інформація використовується для перевірки підписів та виявлення підозрілих змін у криптографічній ідентичності співрозмовників.

Система обробляє різні сценарії помилок та аномальних ситуацій. У разі невалідного підпису повідомлення відхиляється, користувач отримує попередження про можливу підробку, а підозріле повідомлення не розшифровується. Якщо публічний ключ відправника відсутній у локальному сховищі, система зберігає ключ, отриманий з повідомлення, ініціює процес TOFU для його верифікації та обробляє повідомлення після підтвердження ключа користувачем. При виявленні зміни публічного ключа система фіксує факт зміни, виконує аналіз на Key Substitution атаку та показує попередження з рекомендаціями щодо подальших дій. Рівень критичності попередження залежить від результатів аналізу: зміна верифікованого ключа вважається високою загрозою, а спроба повернути старий ключ з історії - критичною.

У разі виникнення помилок, що можуть свідчити про атаку або некоректну роботу, додаток дотримується принципу «fail-safe», тобто переходить у безпечний стан. Це означає, що повідомлення не відображається користувачу, дані не зберігаються у локальному сховищі, а користувач отримує повідомлення про проблему без розкриття технічних деталей. Такий підхід зменшує ризик соціальної інженерії та неправильного трактування повідомлень.

Інтеграція цифрового підпису в процес відправлення повідомлення реалізована безпосередньо в методі `Message.sendMessage()`. Після шифрування повідомлення та формування об'єкта з метаданими система автоматично створює підписаний пакет. Ключовим моментом є те, що підписується саме

зашифрований контент разом зі службовими даними, що забезпечує захист від модифікації як вмісту, так і метаданих повідомлення.

Процес інтеграції включає наступні кроки: перевірку наявності публічного ключа отримувача та відправку ключа підпису, якщо він ще не був переданий; формування пакета для підпису з включенням зашифрованого вмісту, часової мітки, унікального ідентифікатора та хешу логіну відправника; створення цифрового підпису через модуль `DigitalSignature`; додавання підпису та публічного ключа до фінального пакета; передачу підписаного пакета на сервер у вигляді JSON-рядка. Інтеграцію цифрового підпису в процес відправлення повідомлення можна побачити на лістингу 2.5.

Лістинг 2.5 – Інтеграція цифрового підпису в процес відправлення повідомлення

```
// Перевірка та відправка публічного ключа підпису
const recipientObj = Recipient.getByHashName(recipient);
if (!recipientObj.signingPublicKeyBase64) {
  const publicKeyBase64 = await User.getSigningPublicKey();
  if (publicKeyBase64) {
    socket.send(JSON.stringify({
      type: "signing_key_exchange",
      sender,
      recipient,
      key: publicKeyBase64
    }));
  }
}
// Формування пакета для підпису після шифрування
const packetToSign = {
  timestamp: messageObj.timestamp,
  messageId: messageObj.id,
  senderLogin: User.hashName,
  encryptedPayload: base64EncryptedMessage
};
// Створення цифрового підпису
const privateKey = User.getSigningPrivateKey();
```

Продовження лістингу 2.5

```
const signature = await DigitalSignature.signMessage(
  privateKey,
  packetToSign
);
// Формування фінального підписаного пакета
const signedPacket = {
  ...packetToSign,
  signature: signature,
  senderPublicKey: await User.getSigningPublicKey()
};
```

При отриманні повідомлення інтеграція цифрового підпису відбувається в методі `Message.handleMessage()`. Система спочатку визначає, чи отримане повідомлення є підписаним пакетом, перевіряючи наявність полів `signature` та `encryptedPayload`. Якщо повідомлення містить новий публічний ключ, система оновлює локальне сховище та виконує перевірку на Key Substitution атаку через модуль `KeyVerification`. Після цього виконується верифікація цифрового підпису, і лише у разі успішної перевірки повідомлення розшифровується та відображається користувачу.

РОЗДІЛ 3 НАУКОВО-ДОСЛІДНА ЧАСТИНА

3.1 Дослідження стійкості додатку до атак на автентичність повідомлень

Метою даного підрозділу є експериментальне дослідження стійкості додатку до атак, спрямованих на порушення автентичності та цілісності повідомлень у системі з наскрізним шифруванням. Особлива увага приділяється перевірці ефективності впровадженого механізму цифрового підпису в умовах типових загроз, характерних для розподілених комунікаційних систем.

У межах даного дослідження розглядається гіпотеза, що застосування цифрового підпису у поєднанні з наскрізним шифруванням дозволяє суттєво підвищити рівень захисту від атак на автентичність повідомлень, зокрема від їх підміни, модифікації та повторного відтворення.

Для перевірки цієї гіпотези було сформульовано такі дослідницькі задачі:

- визначити перелік типових атак, які можуть бути здійснені на додаток;
- змоделювати відповідні сценарії атак у контрольованому середовищі;
- оцінити поведінку додатку під час кожної з атак;
- проаналізувати здатність системи виявляти та блокувати небезпечні повідомлення.

Дослідження проводилося у середовищі клієнт–серверної веб-архітектури, де клієнтська частина додатку функціонує у браузері користувача, а сервер виконує роль транспортного вузла для передачі зашифрованих повідомлень.

Криптографічні операції, зокрема створення та перевірка цифрових підписів, виконуються виключно на клієнтській стороні. Сервер не має доступу до приватних ключів і не здійснює перевірку підписів, що відповідає моделі zero-knowledge та виключає можливість впливу сервера на результати дослідження.

У межах дослідження розглядаються атаки, які не спрямовані на злам криптографічних алгоритмів, а використовують слабкі місця в логіці обміну ключами та обробки повідомлень. До таких атак належать: атака повторного відтворення повідомлень (replay attack) [12]; атака підміни публічного ключа цифрового підпису [11]; вставка фальшивих або модифікованих повідомлень; атака типу “людина посередині” під час обміну ключами.

Кожен сценарій атаки моделюється окремо з метою оцінки реакції додатку на порушення цілісності або автентичності даних.

Для оцінки ефективності механізму цифрового підпису визначено такі критерії: здатність додатку виявляти підроблені повідомлення; неможливість прийняття повідомлення з некоректним підписом; коректне виявлення повторних або застарілих повідомлень; інформування користувача про зміну публічного ключа співрозмовника; відсутність помилкових спрацьовувань для коректних повідомлень.

Система вважається стійкою до відповідного класу атак, якщо жодне з атакованих повідомлень не було прийняте як валідне.

Методика дослідження полягає у послідовному моделюванні кожного сценарію атаки з фіксацією результатів обробки повідомлень додатком. Для кожного експерименту реєструється стан системи до та після атаки, а також реакція механізмів перевірки цифрового підпису.

Отримані результати використовуються для подальшого аналізу та порівняння з очікуваною поведінкою системи.

3.2 Експериментальна перевірка механізму цифрового підпису

У даному підрозділі наведено результати експериментальної перевірки механізму цифрового підпису, впровадженого у додатку, шляхом моделювання типових атак на автентичність та цілісність повідомлень. Метою експериментів є підтвердження здатності системи виявляти та блокувати несанкціоновані або модифіковані повідомлення без участі серверної сторони.

Кожен експеримент проводився у контрольованому середовищі з двома логічними ролями: відправник та отримувач. Повідомлення передавалися через сервер, який розглядався як потенційно недовірений елемент і не здійснював жодних криптографічних перевірок.

Для кожного сценарію атаки спочатку фіксувалася очікувана поведінка додатку, після чого виконувалося штучне порушення нормального процесу обміну повідомленнями. Результати експериментів порівнювалися з теоретично передбаченими.

У межах даного експерименту моделювалася ситуація, за якої зловмисник намагається вставити в систему повідомлення, сформоване без використання приватного ключа цифрового підпису відправника.

Таке повідомлення містило коректний формат даних, однак підпис або був відсутній, або створений стороннім ключем. У результаті експерименту додаток коректно ідентифікував повідомлення як недійсне та відхиляв його на етапі перевірки підпису. Повідомлення не передавалося до етапу розшифрування і не відображалося користувачу.

Цей результат підтверджує, що механізм цифрового підпису ефективно запобігає атакам, спрямованим на підміну джерела повідомлення.

Наступний експеримент був спрямований на перевірку стійкості системи до атак повторного відтворення повідомлень. Для цього коректне підписане повідомлення повторно надсилалося отримувачу з ідентичними метаданими.

У ході експерименту додаток виявляв повторюваний ідентифікатор повідомлення або невідповідну часову мітку та відхиляв повідомлення як потенційно небезпечне. Повторне повідомлення не проходило валідацію навіть за умови коректного цифрового підпису.

Таким чином, поєднання цифрового підпису з механізмами контролю унікальності повідомлень забезпечує ефективний захист від replay-атак.

Окрему увагу було приділено експерименту з підміною публічного ключа цифрового підпису відправника. У цьому сценарії зловмисник намагався замінити публічний ключ, що використовується для перевірки підпису, на новий.

У результаті експерименту додаток фіксував факт зміни ключа та переводив стан довіри до співрозмовника у непідтверджений. Користувач отримував попередження про зміну криптографічної ідентичності відправника, а подальша обробка повідомлень вимагала явного підтвердження.

Це свідчить про ефективність механізму контролю ключів і здатність системи виявляти атаки типу “людина посередині”.

У межах заключного експерименту було змодельовано комплексний сценарій, що поєднував підміну ключа, модифікацію повідомлення та повторне відтворення. Усі такі повідомлення були послідовно відхилені системою на різних етапах валідації.

Додаток демонстрував коректну поведінку, не допускаючи жодного некоректного повідомлення до кінцевого користувача, що підтверджує узгоджену роботу всіх захисних механізмів.

Під час експериментальної перевірки додаток демонстрував коректну та стабільну поведінку, не допускаючи жодного некоректного або скомпрометованого повідомлення до кінцевого користувача, що свідчить про узгоджену роботу всіх реалізованих захисних механізмів.

Результати тестування стійкості системи до основних типів атак наведено в таблиці 3.1, де для кожного сценарію зафіксовано етап виявлення загрози та підсумковий статус обробки повідомлення.

Таблиця 3.1 – Результати експериментальної перевірки стійкості до атак

Тип атаки	Опис сценарію	Результат	Етап	Статус
1	2	3	4	5
Підміна повідомлення	Вставка повідомлення без валідного підпису	Блоковано	Перевірка підпису	Успішно
Модифікація даних	Зміна вмісту або підписаного пакета	Блоковано	Перевірка підпису	Успішно
Replay-атака	Повторна передача старого повідомлення	Блоковано	Перевірка messageId і timestamp	Успішно

Продовження таблиці 3.1

1	2	3	4	5
Підміна публічного ключа	Заміна ключа під час обміну	Виявлено	TOFU та аналіз змін ключів	Успішно
Key Substitution	Підміна верифікованого ключа	Виявлено з попередженням	Аналіз історії ключів	Успішно
Комплексна атака	Поєднання підміни ключа та модифікації	Блоковано	Множинні етапи валідації	Успішно

Аналіз даних, наведених у таблиці 3.1, підтверджує ефективність запропонованої архітектури захисту в умовах активного противника. Усі спроби підміни, модифікації або повторної передачі повідомлень були своєчасно виявлені та заблоковані на відповідних етапах валідації. Застосування багаторівневої перевірки, що поєднує цифровий підпис, контроль унікальності повідомлень і TOFU-механізми, забезпечує стійкість системи як до окремих, так і до комплексних атак. Отримані результати експерименту підтверджують практичну придатність розробленого рішення для використання в системах з наскрізним шифруванням.

3.3 Аналіз результатів експериментів та оцінка ефективності застосування цифрового підпису

За результатами проведених експериментальних досліджень було отримано сукупність даних, що дозволяє оцінити ефективність застосування цифрового підпису в додатку з наскрізним шифруванням повідомлень. Аналіз результатів підтверджує, що впроваджений механізм істотно підвищує рівень автентичності та цілісності переданих даних.

Усі змодельовані сценарії атак, спрямовані на порушення автентичності повідомлень, були успішно виявлені та заблоковані додатком. Жодне з

атакованих повідомлень не було прийняте як валідне або відображене кінцевому користувачу.

Зокрема фальшиві або модифіковані повідомлення відхилялися на етапі перевірки цифрового підпису; повторно відтворені повідомлення блокувалися завдяки контролю ідентифікаторів та часових міток; спроби підміни публічного ключа призводили до скидання довіри та вимагали підтвердження користувача.

Отримані результати повністю відповідають теоретичним очікуванням, сформульованим у попередніх розділах роботи.

Порівняльний аналіз показує, що система з наскрізним шифруванням без механізму цифрового підпису не здатна гарантувати автентичність відправника та цілісність повідомлень у присутності активного зловмисника. У таких системах сервер або зловмисник може здійснювати підміну повідомлень або ключів без негайного виявлення.

Натомість застосування цифрового підпису дозволяє чітко відокремити шифрування як засіб конфіденційності від підпису як засобу автентичності; забезпечити криптографічне підтвердження джерела повідомлення; виявляти зміни в повідомленнях незалежно від їхнього шифрування.

Це підтверджує доцільність використання цифрового підпису як обов’язкового елемента систем захищеного обміну повідомленнями.

Запровадження цифрового підпису не призводить до критичного зниження продуктивності додатку, оскільки операції підписання та перевірки виконуються на клієнтській стороні та не потребують взаємодії із сервером.

Водночас рівень безпеки системи суттєво зростає за рахунок зменшення довіри до серверної частини; підвищення стійкості до активних атак; надання користувачу інструментів для контролю криптографічної ідентичності співрозмовників.

Отримані результати підтверджують практичну придатність поєднання цифрового підпису з наскрізним шифруванням [6] у веб-орієнтованих системах обміну повідомленнями. Дослідження демонструє, що навіть у середовищі з обмеженими можливостями зберігання ключів можливо реалізувати ефективний механізм автентичності без втрати основних властивостей системи.

3.4 Технічна реалізація та інтеграція цифрового підпису

У межах дослідження було реалізовано повнофункціональну систему цифрового підпису, інтегровану в існуючий веб-додаток для безпечного обміну повідомленнями. Технічна реалізація базується на використанні Web Crypto API для виконання криптографічних операцій безпосередньо в браузері, що забезпечує нульову довіру до серверної інфраструктури.

Система реалізована як модульна архітектура з чітким розподілом відповідальності між компонентами. Модуль `DigitalSignature` інкапсулює всі операції з Ed25519: генерацію ключів, створення та перевірку підписів, експорт та імпорт ключів. Модуль `User` управляє життєвим циклом особистих ключів користувача, забезпечуючи їх генерацію при першому використанні та збереження між сесіями через IndexedDB.

Ініціалізація ключів цифрового підпису відбувається автоматично при вході користувача в систему. Модуль `User` перевіряє наявність збережених ключів у локальному сховищі та або завантажує їх, або генерує нову пару ключів. Процес ініціалізації інтегрований в метод `Message.handleSetSender()`, який викликається після успішної авторизації користувача на сервері.

Інтеграція цифрового підпису в процес обміну повідомленнями реалізована в модулі `Message`, який формує підписаний пакет перед відправкою та виконує перевірку підпису при отриманні. Особливістю реалізації є те, що підписується зашифрований пакет (sign-then-encrypt підхід), що забезпечує захист від модифікації як вмісту, так і метаданих повідомлення.

Процес відправлення повідомлення з цифровим підписом включає наступні етапи: шифрування вмісту повідомлення з використанням спільного ключа Diffie-Hellman; формування пакета для підпису, що включає зашифрований контент, часову мітку, унікальний ідентифікатор та хеш логіну відправника; створення цифрового підпису через метод `DigitalSignature.signMessage()`, який серіалізує об'єкт у JSON-рядок та створює підпис за алгоритмом Ed25519; додавання підпису та публічного ключа до пакета; передачу підписаного пакета на сервер у вигляді JSON-рядка.

При отриманні повідомлення система автоматично визначає, чи є воно підписаним пакетом, перевіряючи наявність полів `signature` та `encryptedPayload`. Якщо повідомлення містить новий публічний ключ, система оновлює локальне сховище, виконує перевірку на Key Substitution атаку та ініціює процес TOFU [7], якщо це перший контакт. Після цього виконується верифікація цифрового підпису, і лише у разі успішної перевірки повідомлення розшифровується та відображається користувачу з відповідним індикатором статусу верифікації.

Механізм TOFU реалізований в модулі `KeyVerification`, який показує діалог перевірки ключа при першому контакті, виявляє зміну публічного ключа та зберігає історію ключів для аналізу підозрілих змін. Система автоматично виявляє Key Substitution атаки через метод `checkKeySubstitutionAttack()`, який аналізує: чи ключ був раніше верифікований, чи новий ключ був у історії (що може свідчити про rollback атаку), частоту змін ключів. На основі результатів аналізу показується попередження з відповідним рівнем критичності: низький, середній, високий або критичний.

Інтеграція з інтерфейсом користувача забезпечує візуальну індикацію статусу верифікації підписів через модуль `DOM`. Кожне повідомлення відображається з індикатором, який показує результат перевірки підпису та статус верифікації ключа співрозмовника. Додатково, в списку користувачів відображається статус верифікації ключа кожного співрозмовника, що дозволяє користувачу швидко оцінити рівень довіри до комунікації.

Результати технічної реалізації демонструють, що інтеграція цифрового підпису не призводить до критичного зниження продуктивності додатку, оскільки операції підписання та перевірки виконуються на клієнтській стороні та не потребують взаємодії з сервером. Водночас рівень безпеки системи суттєво зростає за рахунок зменшення довіри до серверної частини, підвищення стійкості до активних атак та надання користувачу інструментів для контролю криптографічної ідентичності співрозмовників.

3.5 Висновки до розділу 3

У межах науково-дослідної частини було проведено експериментальне дослідження ефективності застосування цифрового підпису в додатку з наскрізним шифруванням повідомлень. Основна увага приділялася перевірці стійкості системи до атак, спрямованих на порушення автентичності та цілісності даних.

У ході дослідження змодельовано типові сценарії атак, зокрема підміну повідомлень, повторне відтворення даних, зміну публічних ключів та комплексні атаки з активною участю зловмисника. За результатами експериментів встановлено, що впроваджений механізм цифрового підпису забезпечує своєчасне виявлення та блокування всіх досліджуваних загроз.

Отримані результати підтверджують, що поєднання наскрізного шифрування з цифровим підписом дозволяє суттєво підвищити рівень безпеки обміну повідомленнями, зокрема в частині автентичності відправника та цілісності переданих даних. При цьому серверна частина не потребує довіри і не бере участі у криптографічних перевірках, що відповідає сучасним вимогам до захищених комунікаційних систем.

Технічна реалізація системи демонструє практичну придатність поєднання цифрового підпису з наскрізним шифруванням у веб-орієнтованих системах обміну повідомленнями. Модульна архітектура забезпечує гнучкість та розширюваність рішення, а використання стандартних браузерних API гарантує кросплатформеність та сумісність з сучасними веб-технологіями.

Таким чином, проведене дослідження підтверджує доцільність і ефективність використання цифрового підпису як обов'язкового елемента систем з наскрізним шифруванням повідомлень, а отримані результати можуть бути використані як основа для подальших досліджень та вдосконалення подібних рішень.

РОЗДІЛ 4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

4.1. Охорона праці

Для забезпечення продуктивної та безпечної діяльності персоналу, залученого до розробки програмного забезпечення комп'ютерних систем, у тому числі фахівців з розробки та дослідження програмного забезпечення криптографічних систем і захищених систем обміну повідомленнями. При цьому керівництво суб'єкта господарювання несе пряму відповідальність за дотримання вимог законодавства у сфері охорони праці.

Додатково, на індивідуальних робочих місцях необхідно гарантувати відповідність вимогам НПАОП 0.00-7.15-18 «Вимоги щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями».

Згідно з вимогами НПАОП 0.00-7.15-18, приміщення, у яких розміщені робочі місця операторів, за винятком приміщень з робочими місцями операторів великих ЕОМ загального призначення (серверних), повинні бути обладнані системами автоматичної пожежної сигналізації відповідно до таких нормативних документів:

– НАПБ Б.06.004-2005 «Перелік однотипних за призначенням об'єктів, які підлягають обладнанню автоматичними установками пожежогасіння та пожежної сигналізації»;

– ДБН В.2.5-56:2014 «Інженерне обладнання будинків і споруд. Пожежна автоматика будинків і споруд», із застосуванням димових пожежних сповіщувачів та переносних вуглекислотних вогнегасників.

У приміщеннях, що не мають спеціалізованого призначення, допускається встановлення теплових пожежних сповіщувачів. Робочі місця операторів повинні бути оснащені первинними засобами пожежогасіння, кількість яких визначається відповідно до вимог ДСТУ 4297:2004 «Пожежна техніка. Технічне обслуговування вогнегасників» з урахуванням загальних технічних вимог та гранично допустимих концентрацій вогнегасних речовин згідно з НАПБ А.01.001-2014.

Приміщення, в яких розташовані сервери загального призначення, обов’язково обладнуються системами автоматичної пожежної сигналізації та засобами пожежогасіння, які мають відповідати нормам ДБН В.2.5-56:2014, НАПБ А.01.001-2014, а також технічним та експлуатаційним вимогам виробника обладнання. Усі підходи до засобів пожежогасіння мають залишатися вільними для доступу.

Електромережа, призначена для живлення комп’ютерів та периферійного обладнання, повинна виконуватися окремою трипровідною груповою лінією із використанням фазового, нульового робочого та нульового захисного провідників. Нульовий захисний провідник використовується виключно для заземлення техніки, і його заборонено застосовувати як нульовий робочий провідник. Він має бути проложений від розподільчого щита або пункту до розеток електроживлення з дотриманням стандартів. Заборонено підключати на одному контактному затискачі нульовий робочий і захисний провідники.

Площа перерізу фазового, нульового робочого та нульового захисного провідників у груповій мережі не повинна бути меншою за площу перерізу фазового провідника. Усі провідники повинні відповідати параметрам мережі, передбаченому навантаженню, умовам експлуатації, розташуванню проводів, температурним режимам та характеристикам використовуваної апаратури захисту згідно з вимогами НПАОП 40.1-1.01-97.

У приміщеннях із кількістю комп’ютерів понад п’ять одиниць на видному і доступному місці має бути встановлений аварійний резервний вимикач для повного знеструмлення приміщення, окрім освітлення. Підключення комп’ютерів до мережі припустиме виключно за допомогою справних штепсельних з'єднань та електророзеток заводського виробництва.

Щодо штепсельних з’єднань і електророзеток, окрім контактів для фазового та нульового робочого провідників, необхідно передбачити спеціалізовані контакти для підключення нульового захисного провідника. Конструкція таких елементів має забезпечувати, щоб під'єднання нульового захисного провідника відбувалося раніше, ніж з'єднання фазового та нульового робочого провідників, а при їх роз’єднанні цей порядок має бути інверсним.

Комп'ютери не дозволено підключати до стандартної двопровідної електромережі, навіть за допомогою перехідних пристроїв. Електромережі для штепсельних з’єднань та електророзеток, призначених для живлення комп’ютерної техніки, мають бути облаштовані за магістральною схемою і містити по 3–6 з’єднань чи електророзеток в одному колі. Крім того, штепсельні з’єднання та електророзетки для напруги 12 В і 42 В за конструктивними характеристиками повинні суттєво відрізнятися від тих, що призначені для напруги 127 В і 220 В. Для візуального розмежування їх слід виконувати у різних кольорах.

При підвищенні ефективності контролю доступу в приміщення, де для забезпечення безпеки мешканців, співробітників і збереження майна використовуються ДС, важливим, з точки зору охорони праці, є забезпечення достатньої величини природного та штучного освітлення, які визначені у НПАОП 0.00-7.15-18. Організація робочого місця фахівця із дослідження методів та програмно-апаратних засобів оптимізаційних процесів на основі ГА повинна забезпечувати відповідність усіх елементів робочого місця та їх розташування ергономічним вимогам ДСТУ 8604:2015 «Дизайн і ергономіка. Робоче місце для виконання робіт у положенні сидячи. Загальні ергономічні вимоги».

Таким чином, у результаті аналізу вимог з охорони праці визначено основні аспекти безпечної організації робочих місць фахівців, які здійснюють розробку та дослідження програмного забезпечення систем із наскрізним шифруванням повідомлень та використанням цифрового підпису. Дотримання вимог пожежної та електробезпеки, нормативів освітлення, а також ергономічних і санітарно-гігієнічних норм створює умови для ефективної, надійної та безпечної професійної діяльності розробників криптографічних програмно-апаратних рішень.

4.2 Захист людини від іонізуючого випромінювання

Іонізуюче випромінювання – це вид енергії, здатний творити заряджені атоми та молекули у середовищі, з яким він контактує. Воно має як природні джерела (космічні промені, земні природні радіонукліди), так і штучні: ядерні реактори, рентгенівські установки, прискорювачі частинок та радіоактивні ізотопи. До іонізуючих випромінювань відносять корпускулярні (альфа-, бета-випромінювання та нейтрони) й електромагнітні – гама- та рентгенівські промені.

Їхня головна властивість полягає у високій енергії, що може порушувати структуру біологічних клітин і навіть спричиняти їхнє знищення. На ці випромінювання не реагують органи чуття людини, що робить їх особливо небезпечними. Водночас іонізуюче випромінювання має широке практичне використання – від медицини та промисловості до сільського господарства й атомної енергетики.

Основними нормативними актами, що регулюють радіаційну безпеку в Україні, є ДНАОП 00.3-3.24-97, «Норми радіаційної безпеки України» НРБУ-97/Д-2000 та «Основні санітарні норми України» ОСПУ-2000. За допустимими основними дозовими межами встановлюються такі категорії осіб, що опромінюються:

– категорія А – персонал, що має прямий зв'язок із джерелами іонізуючого випромінювання. Сумарна доза опромінення на рік – 5 бер (50 мЗв);

– категорія Б – персонал, що безпосередньо не оперує з радіоактивними речовинами, але при розміщенні на робочих або житлових місцях може зазнати впливу опромінення. Гранична доза опромінення – 0,5 бер/рік;

– категорія В – решта населення держави. Доза не лімітується, але не має перевищувати природний фон – від 40 до 200 мбер/рік.

Для осіб категорій А і Б НРБУ-97 встановлює ліміти ефективної та еквівалентної дози за календарний рік. Обмеження опромінення категорії В

здійснюється введенням лімітів річної ефективної та еквівалентної доз для критичних груп осіб категорії Б.

Окрім лімітів ефективної та еквівалентної річних доз, НРБУ-97 визначено допустимі рівні надходження радіонуклідів в організм людини за календарний рік, потужності еквівалентної дози, концентрації радіонуклідів у повітрі, питній воді та раціоні, щільності потоку частинок, забруднення шкіри, спецодягу, робочих поверхонь тощо. Величина окремого допустимого рівня розраховується за умови, що створена ним річна доза не повинна перевищувати ліміту відповідної дози. При багаторазовому радіаційному опроміненні допустимі рівні визначаються за умови, щоб річна сумарна доза від усіх джерел випромінювання не перевищувала відповідного ліміту дози [19].

Для захисту від зовнішнього опромінювання, яке спостерігається при роботі із закритими джерелами випромінювання (такими, що унеможливлюють потрапляння радіоактивних речовин у довкілля), основні зусилля слід спрямувати на запобігання переопроміненню персоналу шляхом:

- збільшення дистанції між джерелом випромінювання і людиною;
- зменшення тривалості роботи в зоні випромінювання;
- екранування джерела випромінювання.

Під внутрішнім опроміненням розуміють вплив на тіло людини випромінювань радіоактивних речовин, що проникають всередину організму. На дверях приміщень, у яких провадиться робота з відкритими джерелами радіоактивного випромінювання, має бути розміщено знак радіаційної небезпеки.

Радіоактивні речовини мають зберігатися у спеціальних приміщеннях. Щодо кожного з них необхідно вести суворий облік надходжень та витрат, щоб виключити можливість їх неконтрольованого використання. Порядок транспортування радіоактивних речовин регламентується спеціальними правилами. Радіоактивні речовини перевозять у спеціальних контейнерах та спеціально обладнаним транспортом. До організацій та установ, де постійно виконуються роботи з радіоактивними речовинами, висуваються підвищені вимоги з охорони праці. Керівництво цих організацій зобов'язане розробити

детальні інструкції, в яких викладено порядок виконання робіт, облік зберігання та використання джерел випромінювання, збір та знешкодження відходів, порядок проведення дозиметричного контролю [20].

Особливе значення при роботі з відкритими джерелами радіоактивного випромінювання має особиста гігієна та індивідуальні засоби захисту працюючого. Залежно від типу виконуваних робіт та небезпеки цих робіт застосовують спецодяг (комбінезони чи костюми), спецбілизну, шкарпетки, спецвзуття, рукавиці, респіратори. У випадках можливого забруднення повітря радіоактивним пилом, газами чи парою – додатково використовувати респіратори (ШБ-1 Пелюстка). Також застосовуються спеціальні пневматичні костюми з пластичних матеріалів (ЛГ-4), а також гумові чоботи. При використанні ЗІЗ необхідно обов'язково звертати увагу на черговість їх одягання та знімання. Після роботи з радіоактивними речовинами потрібно добре вимити руки й обличчя, а також перевірити їхню чистоту дозиметричними приладами.

Безпеку роботи з радіоактивними речовинами (РАР) та джерелами випромінювання можна забезпечити тоді, коли є організований систематичний моніторинг за рівнем зовнішнього та внутрішнього опромінення персоналу, а також за рівнем радіації довкілля. Усі, хто працює з РАР, мусять бути забезпечені індивідуальними дозиметрами для контролю дози гама-випромінювань, яку отримує кожен працівник окремо. Для виявлення та кількісного виміру радіоактивного випромінювання використовують методи:

– Іонізаційний – вимірювання рівня іонізації випромінювання;

– сцинтиляційний – вимірювання інтенсивності світлових спалахів, які виникають у речовинах, що люмінесціюють при проходженні крізь них іонізуючих випромінювань;

– фотографічний – ґрунтується на здатності фото емульсійного шару під дією РАР темніти після проявлення);

– хімічний – здатність деяких розчинів змінювати свій колір під дією іонізуючих випромінювань [21].

Результати усіх видів радіаційного контролю повинні зберігатися протягом 30-ти років. При індивідуальному контролі ведеться облік річної дози опромінення, а також сумарної дози за весь період професійної діяльності.

ВИСНОВКИ

Під час виконання кваліфікаційної роботи освітнього рівня «Магістр» було здійснено комплексне дослідження механізмів забезпечення безпеки систем обміну повідомленнями з метою розроблення та інтеграції цифрового підпису в систему з наскрізним шифруванням. Проведений аналіз дозволив обґрунтувати обмеження базових реалізацій E2EE та необхідність додаткового механізму для забезпечення автентичності та цілісності повідомлень. Дослідження сучасних підходів до захисту комунікацій дозволило виокремити ключові вимоги до архітектури системи та обґрунтувати вибір алгоритму Ed25519 для реалізації цифрового підпису.

На основі цього сформовано концепцію та архітектуру системи, функціонування якої охоплює модулі генерації та зберігання криптографічних ключів, підписання повідомлень, верифікації підписів, управління публічними ключами та виявлення атак типу Key Substitution. Реалізована система забезпечує можливість криптографічної перевірки автентичності автора повідомлення та цілісності переданих даних, надаючи користувачам гарантію того, що отримані повідомлення дійсно були створені заявленим відправником і не були модифіковані під час передачі.

Спроєктована архітектура дала змогу інтегрувати всі функціональні компоненти в єдине рішення, здатне забезпечувати комплексний захист обміну повідомленнями, що включає конфіденційність, автентичність, цілісність та незаперечність. Значну увагу приділено аспектам інформаційної безпеки: зберіганню приватних ключів виключно на клієнтській стороні з використанням IndexedDB, забезпеченню zero-trust моделі, де сервер не має доступу до криптографічних ключів або вмісту повідомлень, протидії атакам типу MITM через механізм TOFU та захисту від replay-атак за допомогою timestamp та унікальних ідентифікаторів повідомлень.

Проведене експериментальне дослідження стійкості системи до атак на автентичність повідомлень підтвердило коректність реалізованих механізмів. Усі змодельовані сценарії атак, спрямовані на порушення автентичності

повідомлень, були успішно виявлені та заблоковані додатком. Жодне з атакованих повідомлень не було прийняте як валідне або відображене кінцевому користувачу, що демонструє ефективність впровадженого механізму цифрового підпису.

Підсумовуючи, можна зробити висновок, що поставлену мету досягнуто: сформовано науково й практично обґрунтовану концепцію інтеграції цифрового підпису в системи з наскрізним шифруванням, створено архітектуру програмного забезпечення та реалізовано ключові модулі, необхідні для забезпечення автентичності та цілісності повідомлень у веб-орієнтованих системах обміну повідомленнями. Розроблений підхід може слугувати основою для побудови захищених комунікаційних систем та вдосконалення існуючих рішень з наскрізним шифруванням.

Подальші перспективи в цьому напрямку можуть зосереджуватись на кількох ключових напрямках. Наприклад, розширення функціональності механізму TOFU: впровадження системи репутації ключів, яка дозволить оцінювати надійність публічних ключів на основі історії взаємодій, а також інтеграція з децентралізованими системами верифікації ідентичності для підвищення рівня довіри. У сфері криптографії варто дослідити можливості використання квантово-стійких алгоритмів цифрового підпису для забезпечення довгострокової безпеки системи. У сфері безпеки програмного забезпечення варто продовжувати вдосконалювати захист зберігання ключів, оптимізувати процеси підписання та верифікації для підвищення продуктивності та впроваджувати системи моніторингу аномальної активності для виявлення потенційних загроз у реальному часі.

Таким чином, проведена робота створює міцну основу для подальших досліджень і практичних розробок у сфері поєднання наскрізного шифрування з цифровим підписом. Розроблена система має потенціал перетворитися на комплексне рішення для забезпечення безпеки комунікацій у веб-орієнтованих застосунках, здатне підвищувати рівень довіри користувачів та забезпечувати криптографічно перевірену автентичність та цілісність переданих даних.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Diffie W., Hellman M. New Directions in Cryptography. IEEE Transactions on Information Theory. 1976. Vol. 22, No. 6. P. 644-654. (дата звернення: 15.10.2025).
2. Josefsson S., Liusvaara I. Edwards-Curve Digital Signature Algorithm (EdDSA). RFC 8032. IETF. 2017. URL: https://www.rfc-editor.org/rfc/rfc8032 (дата звернення: 15.10.2025).
3. Web Crypto API. MDN Web Docs. URL: https://developer.mozilla.org/en-US/docs/Web/API/Web_Crypto_API (дата звернення: 20.10.2025).
4. Signal Protocol Documentation. Signal Foundation. URL: https://signal.org/docs/ (дата звернення: 20.10.2025).
5. Cryptographic Storage Cheat Sheet. OWASP Foundation. URL: https://cheatsheetseries.owasp.org/cheatsheets/Cryptographic_Storage_Cheat_Sheet.html (дата звернення: 22.10.2025).
6. End-to-End Encryption. Wikipedia. URL: https://en.wikipedia.org/wiki/End-to-end_encryption (дата звернення: 18.10.2025).
7. Trust On First Use. Wikipedia. URL: https://en.wikipedia.org/wiki/Trust_on_first_use (дата звернення: 18.10.2025).
8. Bernstein D. J., Duif N., Lange T., Schwabe P., Yang B. High-speed high-security signatures. Journal of Cryptographic Engineering. 2012. Vol. 2, No. 2. P. 77-89. doi: 10.1007/s13389-012-0027-1. (дата звернення: 15.10.2025).
9. IndexedDB API. MDN Web Docs. URL: https://developer.mozilla.org/en-US/docs/Web/API/IndexedDB_API (дата звернення: 25.10.2025).
10. PKCS #8: Private-Key Information Syntax Specification. RFC 5208. IETF. 2008. URL: https://www.rfc-editor.org/rfc/rfc5208 (дата звернення: 25.10.2025).

11. Man-in-the-Middle Attack. OWASP Foundation. URL: https://owasp.org/www-community/attacks/Man-in-the-middle_attack (дата звернення: 22.10.2025).

12. Replay Attack. OWASP Foundation. URL: https://owasp.org/www-community/attacks/Replay_attack (дата звернення: 22.10.2025).

13. Tymoshchuk D., Yasniy O., Mytnyk M., Zagorodna N., Tymoshchuk V. Detection and classification of DDoS flooding attacks by machine learning method. CEUR Workshop Proceedings. 2024. Vol. 3842. P. 184-195. (дата звернення: 10.11.2025).

14. Петрик М. Р., Петрик О. Ю. Моделювання програмного забезпечення : наук.-метод. посіб. Тернопіль: Вид-во ТНТУ імені Івана Пулюя, 2015. 200 с.

15. Lypa B., Horyn I., Zagorodna N., Tymoshchuk D., Lechachenko T. Comparison of feature extraction tools for network traffic data. CEUR Workshop Proceedings. 2024. Vol. 3896. P. 1-11. (дата звернення: 10.11.2025).

16. Zagorodna N., Stadnyk M., Lypa B., Gavrylov M., Kozak R. Network Attack Detection Using Machine Learning Methods. Challenges to national defence in contemporary geopolitical situation. 2022. No. 1. P. 55-61. (дата звернення: 10.11.2025).

17. Revniuk O.A., Zagorodna N.V., Kozak R.O., Karpinski M.P., Flud L.O. The improvement of web-application SDL process to prevent Insecure Design vulnerabilities. Applied Aspects of Information Technology. 2024. Vol. 7, No. 2. P. 162-174. doi: 10.15276/aait.07.2024.12. (дата звернення: 10.11.2025).

18. Nataliya Zagorodna, Iryna Kramar. Economics, Business and Security: Review of Relations. Business Risk in Changing Dynamics of Global Village BRCDGV-2020: Monograph / Edited by Pradeep Kumar, Mahammad Sharif. India, Patna: Novelty & Co., Ashok Rajpath, 2020. 446 p. P. 25-39.

19. M. Derkach, et al., CrypticWave: A Zero-Persistence Ephemeral Messaging System with Client-Side Encryption, in: Cyber Security and Data Protection, vol. 4042, 2025, 316–323.

20. Деркач М. В., Мишко О. Є. Використання алгоритму шифрування AES-256-CBC для зберігання даних автентифікації автономного помічника. Наукові вісті Далівського університету. 2023. №24.

21. Kuznetsov, O., Poluyanenko, N., Frontoni, E., Kandiy, S., Karpinski, M., & Shevchuk, R. (2024). Enhancing Cryptographic Primitives through Dynamic Cost Function Optimization in Heuristic Search. Electronics, 13(10), 1-52. doi: 10.3390/electronics13101825.

22. Yakymenko, I., Karpinski, M., Shevchuk, R., & Kasianchuk, M. (2024). Symmetric Encryption Algorithms in a Polynomial Residue Number System. Journal of Applied Mathematics, art. id 4894415, 1-12. doi: 10.1155/2024/4894415.

23. Zagorodna, N., Skorenkyy, Y., Kunanets, N., Baran, I., & Stadnyk, M. (2022). Augmented Reality Enhanced Learning Tools Development for Cybersecurity Major. In ITTAP (pp. 25-32).

24. Микитишин, А. Г., Митник, М. М., Голотенко, О. С., & Карташов, В. В. (2023). Комплексна безпека інформаційних мережевих систем. Навчальний посібник для студентів спеціальності 174 «Автоматизація, комп'ютерно-інтегровані технології та робототехніка».

25. Про введення в дію Державних гігієнічних нормативів «Норми радіаційної безпеки України (НРБУ-97)». Верховна Рада України. URL: https://zakon.rada.gov.ua/rada/show/v0062282-97#Text (дата звернення: 01.12.2025).

26. Дія іонізуючого випромінення на організм людини. Львівський державний університет безпеки життєдіяльності ДСНС України. URL: https://virt.ldubgd.edu.ua/mod/page/view.php?id=1577 (дата звернення: 01.12.2025).

27. Види іонізуючих випромінювань, їх фізична природа та особливості розповсюдження. ATutor. URL: https://dl.tntu.edu.ua/content.php?cid=289166 (дата звернення: 01.12.2025).

Додаток А Публікація

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ТЕРНОПІЛЬСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ імені ІВАНА ПУЛЮЯ
НАУКОВЕ ТОВАРИСТВО ім. ШЕВЧЕНКА

XIII

НАУКОВО-ТЕХНІЧНА

К О Н Ф Е Р Е Н Ц І Я

Інформаційні моделі, системи та технології

17-18 грудня 2025 року

ТЕРНОПІЛЬ – 2025

УДК 004.056

Володимир Пилипчук, студент гр. СБм-61, Н.В. Загородна, к.т.н., доц.

(Тернопільський національний технічний університет імені Івана Пулюя)

ІНТЕГРАЦІЯ ЦИФРОВОГО ПІДПИСУ У СИСТЕМАХ З НАСКРІЗНИМ ШИФРУВАННЯМ ПОВІДОМЛЕНЬ

UDC 004.056

Volodymyr Pylypchuk, student of group SBm-61, N. Zagorodna, PhD, Associate Professor

INTEGRATION OF DIGITAL SIGNATURE IN END-TO-END ENCRYPTED MESSAGING SYSTEMS

У сучасних системах захищеного обміну інформацією наскрізне шифрування є базовим механізмом забезпечення конфіденційності, оскільки лише відправник і отримувач володіють ключами для дешифрування повідомлень. Проте сам факт шифрування не усуває ризиків, пов'язаних із підміною відправника, компрометацією ключів або непомітним втручанням у процедури встановлення сеансу зв'язку. У таких сценаріях існує можливість перехоплення та модифікації даних без розкриття їх вмісту, що створює потребу в окремому механізмі підтвердження авторства й незмінності повідомлення.

Гарантування автентичності та цілісності забезпечується цифровим підписом. Його використання дозволяє підтвердити походження повідомлення, засвідчити відсутність змін під час передачі та знизити ймовірність атак «man-in-the-middle», фальсифікації змісту, повторного відправлення старих даних та прихованого втручання у процес початкового обміну ключами. Таким чином, цифровий підпис стає доповненням до шифрування, яке підвищує рівень гарантій безпеки понад ті, що надає сама криптографія шифрування.

Додавання цифрового підпису розширює функціональність E2EE-систем без зміни фундаментальної архітектури взаємодії, зберігаючи властивість незалежності від сервера та відсутності потреби в централізованих компонентах довіри. З урахуванням ротації ключів у сучасних протоколах обміну сформовано підхід, що поєднує цифровий підпис із механізмом довіри при першому використанні (TOFU), підтримкою історії змін ключів для контролю підміни та збереженням сумісності з forward secrecy. Така інтеграція дозволяє виявляти потенційні аномалії без розкриття ключової інформації та без порушення децентралізованої моделі взаємодії.

Реалізація цифрового підпису в системах із наскрізним шифруванням здатна підвищити рівень довіри між учасниками комунікації, посилити захист від сценаріїв підміни та покращити точність виявлення недостовірних повідомлень у зашифрованих каналах. Такий підхід робить цифровий підпис доцільним компонентом сучасних E2EE-рішень як у корпоративному, так і персональному обміні даними, де критичними є конфіденційність, автономність клієнта та доказовість джерела інформації.

Література

1. Menezes A., Van Oorschot P., Vanstone S. Handbook of Applied Cryptography. CRC Press, 2021.
2. Perrin T. The Signal Protocol: Technical Overview. Signal Foundation, 2022.
3. Bernstein D., Schwabe P. Ed25519: High-speed high-security signatures. Springer, 2017.