Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра захисту інформації

ЗВІТ
З ПЕРЕДДИПЛОМНОЇ ПРАКТИКИ

Тема кваліфікаційної роботи: «ПРОГРАМНИЙ ЗАСІБ ЗАХИЩЕНОГО ПЕРЕДАВАННЯ ІНФОРМАЦІЇ. ЧАСТИНА 2. ПРОТОКОЛ ОБМІНУ ІНФОРМАЦІЄЮ»

Студента групи 1БС-23м
спеціальності 125 «Кібербезпека та захист інформації»
ОПП Безпека інформаційних і комунікаційних систем

_____ Володимир КОЗИРА
(підпис) (ім'я прізвище)

Керівник практики від бази практики

_____ Олександр БАБАЧУК
(підпис) (ім'я прізвище)

Керівник кваліфікаційної роботи

_____ Володимир ЛУЖЕЦЬКИЙ
(підпис) (ім'я прізвище)

Керівник практики від кафедри

_____ Леонід КУПЕРШТЕЙН
(підпис)

Рекомендована оцінка:

кількість балів _____ за шкалою ECTS ____

Вінниця 2025

Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра захисту інформації
Рівень вищої освіти II (магістерський)
Галузь знань – 12 Інформаційні технології
Спеціальність – 125 Кібербезпека та захист інформації
Освітньо-професійна програма – Безпека інформаційних і комунікаційних систем

ЗАТВЕРДЖУЮ
Завідувач кафедри ЗІ,
д. т. н., проф.
_____Володимир ЛУЖЕЦЬКИЙ
«___» _____ 2025 року

З А В Д А Н Н Я
НА КОМПЛЕКСНУ МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ
СТУДЕНТУ
Володимиру КОЗИРІ

1. Тема роботи: «Програмний засіб захищеного передавання інформації. Частина 2. Протокол обміну інформацією»
   керівник роботи: Володимир ЛУЖЕЦЬКИЙ, д. т. н., проф., завідувач кафедри ЗІ,
   затверджені наказом ректора ВНТУ від 2 вересня 2025 року № 2.
2. Строк подання студентом роботи 19 грудня 2025 р.
3. Вихідні дані до роботи:
   - Протокол захищеного обміну інформацією – клієнт-клієнт.
   - Протокол угоди про ключі – протокол Діфі-Гелмана.
   - Метод ущільнення ключа – на основі кватерніонів.
   - Метод нормування кватерніона.
   - Метод побудови поворотної матриці – на основі кватерніонів.
   - Розрядність елементів поворотної матриці – 16 біт.
4. Зміст текстової частини: Вступ. 1. Аналіз інформаційних джерел. 2. Розробка протоколу обміну інформацією 3. Програмна реалізація протоколу обміну інформацією. 4. Економічна частина. Висновки. Список використаних джерел. Додатки.
5. Перелік ілюстративного матеріалу: алгоритм симетричного шифрування (плакат А4), переваги та недоліки протоколів розподілу симетричних ключів (плакат А4), Алгоритм асиметричного шифрування (плакат А4), переваги та недоліки протоколів розподілу асиметричних ключів (плакат А4), послідовність інформаційних кроків протоколу (плакат А4), архітектура

програмного засобу для безпечного обміну повідомленнями (плакат А4), алгоритм узгодження параметрів та перевірки ортогональності (плакат А4), формування поворотної матриці та встановлення стану готовності (плаката А4).

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв
1	В. ЛУЖЕЦЬКИЙ, д. т. н., проф., завідувач кафедри ЗІ		
2	В. ЛУЖЕЦЬКИЙ, д. т. н., проф., завідувач кафедри ЗІ		
3	В. ЛУЖЕЦЬКИЙ, д. т. н., проф., завідувач кафедри ЗІ		
4	В. ЛУЖЕЦЬКИЙ, д. т. н., проф., завідувач кафедри ЗІ		

7. Дата видачі завдання 2 вересня 2025 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів магістерської кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз завдання. Вступ	01.09.2025 – 04.09.2025	
2	Аналіз інформаційних джерел за напрямком магістерської кваліфікаційної роботи	05.09.2025 – 15.09.2025	
3	Науково-технічне обґрунтування	16.09.2025 – 22.09.2025	
4	Розробка технічного завдання	23.09.2025 – 04.10.2025	
5	Розробка протоколу захищеного обміну інформацією	05.10.2025 – 08.10.2025	
6	Розробка алгоритму ущільнення ключа	09.10.2025 – 16.10.2025	
7	Розробка алгоритму побудови поворотної матриці	17.10.2025 – 14.11.2025	
8	Тестування програмного засобу захищеного передавання інформації	15.11.2025 – 17.11.2025	

9	Розробка розділу економічного обґрунтування доцільності розробки	18.11.2025 – 21.11.2025	
10	Аналіз виконання ТЗ, висновки	22.11.2025 – 24.11.2025	
11	Оформлення пояснювальної записки	25.11.2025 – 29.11.2025	
12	Перевірка магістерської роботи на наявність текстових запозичень	30.11.2025 – 02.12.2025	
13	Попередній захист та доопрацювання МКР	03.12.2025 – 14.12.2025	
14	Представлення МКР до захисту, рецензування	15.12.2025 – 18.12.2025	
15	Захист МКР	19.12.2025 – 23.12.2025	

Студент _____ Володимир КОЗИРА

Керівник комплексної магістерської кваліфікаційної роботи _____Володимир ЛУЖЕЦЬКИЙ

ЗМІСТ

1 АНАЛІЗ ІНФОРМАЦІЙНИХ ДЖЕРЕЛ 7

1.1 Протоколи обміну інформацією в месенджерах 7

1.2 Протоколи розподілу ключів 10

1.3 Протокол угоди про ключі Діфі–Гелмана 18

1.4 Висновки з розділу 22

2 РОЗРОБКА ПРОТОКОЛУ ОБМІНУ ІНФОРМАЦІЄЮ 24

2.1 Узагальнений протокол обміну інформацією 24

2.2 Реалізація протоколу Діфі-Гелмана 27

2.3 Процедура генерування секретного ключа 29

2.4 Обчислення поворотної матриці 31

2.5 Стандарти генерування випадкових чисел 34

2.6 Тест простоти Мілера-Рабіна 37

2.7 Висновки з розділу 40

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 42

1 АНАЛІЗ ІНФОРМАЦІЙНИХ ДЖЕРЕЛ

1.1 Протоколи обміну інформацією в месенджерах

У сучасному цифровому середовищі месенджери стали одним із основних засобів комунікації – як для особистого, так і для професійного спілкування. Вони дозволяють миттєво обмінюватись текстовими повідомленнями, голосовими й відеодзвінками, файлами, фото, відео, геоданими тощо.

Серед месенджерів, що користуються популярністю серед людей особливе місце займають наступні:

- Telegram;
- WhatsApp;
- Signal;
- Viber;
- Facebook Messenger.

У різних сучасних месенджерах реалізовано власні протоколи обміну інформацією, які відрізняються між собою структурою узгодження параметрів, способом формування спільного секрету, механізмами побудови сеансових ключів та порядком автентифікації даних.

Хоча всі системи мають спільну мету – встановити захищений канал обміну даними між двома учасниками, порядок виконання дій, залучені проміжні ключі та механізми оновлення криптографічного стану відрізняються залежно від архітектури конкретного месенджера.

Ці відмінності є критичними, оскільки саме протокол обміну інформацією визначає рівень стійкості системи до атак типу MITM та здатність забезпечувати стійкість ключів при компрометації одного з сеансів.

В Telegram шифрування реалізовано на основі MTProto 2.0 (cloud chats). У такій моделі встановлення секрету виконується між клієнтом і сервером, а не напряму між двома абонентами. Сервер виконує роль активного елемента протоколу, забезпечуючи розподіл AuthKey, який обчислюється через процедуру

обміну параметрами з використанням елементів Діфі-Гелмана. Після цього весь трафік між клієнтом та сервером шифрується symm-key на основі AuthKey. Прямого end-to-end A→B ключового матеріалу Telegram у звичайних чатах не формує.

Послідовність інформаційних кроків у процесі встановлення захищеної cloud-сесії в Telegram представлено в табл. 1.1 [1].

Таблиця 1.1 – Основні етапи формування AuthKey у Telegram

Етап	Напрямок	Опис дії
1	A→Server	Надсилання запиту PQ
2	Server→A	Передача p,g та public параметрів
3	A→Server	Передача g^a mod p
4	Server→A	Передача g^b mod p
5	A,Server	Обчислення спільного AuthKey
6	A↔Server	Шифрування MTProto-повідомлень симетричним ключем

WhatsApp застосовує Signal Protocol, що включає використання pre-keys, які завчасно зберігаються на сервері. Завдяки цьому відправник може ініціювати встановлення спільного секрету навіть коли одержувач офлайн. Первинна фаза відповідає X3DH, після чого протокол переходить до Double Ratchet з постійною ротацією ключів для кожного повідомлення.

Послідовність інформаційних кроків для встановлення сеансового стану WhatsApp представлено в табл. 1.2 [2].

Таблиця 1.2 – Основні етапи X3DH/Double Ratchet у WhatsApp

Етап	Напрямок	Опис дії
1	B→Server	Публікація pre-keys
2	A→Server	Запит набору pre-keys
3	A↔B	Обчислення X3DH (identity + signed + one-time)
4	A,B	Формування Root Key
5	A,B	Старт Double Ratchet
6	A→B	Передача AEAD повідомлення
7	B	Перевірка MAC та оновлення Ratchet-стану

Signal визначає еталонну модель, яка складається з X3DH та Double Ratchet. X3DH забезпечує встановлення початкового спільного секрету навіть без синхронного контакту, а Double Ratchet забезпечує forward secrecy та post-compromise security шляхом ротації ключів для кожного повідомлення.

Основні етапи встановлення сеансу у Signal наведено у табл. 1.3 [3].

Таблиця 1.3 – Основні етапи встановлення сеансу у Signal

Етап	Напрямок	Опис дії
1	B→Server	Публікація pre-key bundle
2	A→Server	Отримання pre-key bundle
3	A↔B	Обчислення X3DH секрету
4	A,B	Формування Root Key
5	A,B	Ініціалізація Double Ratchet
6	A↔B	AEAD-передача з оновленням ключів

Viber використовує схему E2E, проте без окремої моделі розширених pre-keys як у Signal. Документ протоколу Viber описує формування гібридного криптографічного стану на основі DH-обміну з подальшим формуванням трьох ключів (message key, MAC key, IV). Передача даних виконується у зашифрованому вигляді, а одержувач виконує перевірку цілісності та автентичності кожного отриманого повідомлення.

Основні етапи встановлення сесії у Viber наведено у табл. 1.4 [4].

Таблиця 1.4 – Основні етапи встановлення сесії у Viber

Етап	Напрямок	Опис дії
1	A→B	Надсилання ініціаційного повідомлення
2	A↔B	DH обмін відкритими значеннями
3	A,B	Формування сесійних ключів (3-key set)
4	A→B	Передача зашифрованих даних
5	B	Перевірка цілісності та автентичності

Оновлена версія Facebook Messenger впроваджує повний режим end-to-end encryption для приватних чатів. Ключовий матеріал формується на основі протоколу X3DH-типу (аналогічна модель до Signal), після чого застосовується

механізм ротації ключів для подальшого обміну. Повідомлення супроводжуються перевірочними MAC значеннями.

Основні етапи обміну у Facebook Messenger представлено у табл. 1.5 [5].

Таблиця 1.5 – Основні етапи обміну у Facebook Messenger

Етап	Напрямок	Опис дії
1	B→Server	Публікація pre-keys
2	A→Server	Отримання pre-keys
3	A↔B	Встановлення спільного секрету
4	A,B	Формування сесійних ключів
5	A→B	Передача зашифрованих повідомлень
6	A↔B	Подальший обмін з ротацією ключів

Отже, сучасні месенджери реалізують різні моделі встановлення спільного секрету. Telegram застосовує модель «клієнт–сервер», де AuthKey формується між клієнтом і сервером, а не між двома користувачами, тому cloud-чати не є повноцінними end-to-end.

Натомість WhatsApp, Signal та оновлений Messenger використовують Signal-підхід: формування початкового секрету через X3DH із застосуванням pre-keys та подальшу ротацію ключів за Double Ratchet, що забезпечує forward secrecy. Viber реалізує E2E-шифрування без розширеного pre-key механізму, однак також формує сесійні ключі через DH-обмін.

Таким чином, еволюція йде у напрямку асинхронності та частішого оновлення ключів, що підвищує стійкість систем до ретроспективного криптоаналізу та компрометації секретного матеріалу.

1.2 Протоколи розподілу ключів

У криптографії розподіл відкритих і закритих ключів між відправником і одержувачем є дуже монотонним завданням.

Розповсюдження ключів має вирішальне значення, оскільки безпека всієї системи залежить від того, наскільки добре ключі спільно використовуються та захищені.

Розподіл ключів – це процес безпечного обміну криптографічними ключами між сторонами, що беруть участь у зв’язку, що гарантує збереження ключів у таємниці та неможливість їх перехоплення або зміни неавторизованими третіми особами [6].

Розподіл ключів відіграє значну роль як у симетричній, так і в асиметричній криптографії та забезпечує доступність ключів для зашифрування та розшифрування, зберігаючи при цьому цілісність та конфіденційність зв’язку.

У симетричній криптографії і відправник, і одержувач використовують один і той самий ключ як для зашифрування, так і для розшифрування.

Це означає, що обидві сторони повинні мати доступ до одного й того ж секретного ключа, що створює проблеми із забезпеченням безпечного розповсюдження ключа.

Схему алгоритму симетричного шифрування зображено на рис. 1.1.

Рисунок 1.1 – Алгоритм симетричного шифрування

Розподіл симетричних ключів є критичною фазою встановлення захищеного зв’язку – саме на цьому етапі забезпечується, щоб тільки довірені

сторони мали доступ до ключа. Через це при розподілі ключів часто виникають низка практичних і організаційних проблем:

– Спільне використання секрету. Ключ має бути безпечно переданий відправнику та одержувачу, перш ніж може відбутися будь-яке зашифроване спілкування. Якщо зловмисник перехопить ключ під час розповсюдження, весь зв’язок може бути скомпрометований.

– Керування ключами. У системах з багатьма учасниками розподіл унікального ключа для кожної пари взаємодіючих сторін може стати складним.

Для забезпечення безпеки симетричного шифрування ключ повинен бути переданий сторонам надійним способом. Існує кілька протоколів розподілу симетричних ключів:

– Ручний розподіл;

– Інфраструктура відкритих ключів (PKI, Public Key Infrastructure);

– Обмін ключами за Діфі-Гелманом;

– Квантовий розподіл ключів (QKD, Quantum Key Distribution) [6].

Ручний розподіл полягає у тому, що ключ доставляється фізично або вручну обмінюється між сторонами (наприклад, через довіреного кур’єра, особисту зустріч або захищений офлайн-канал).

Хоча PKI переважно використовується в асиметричній криптографії, вона також може використовуватися для розповсюдження симетричних ключів. Система використовує пари публічних та закритих ключів , які дозволяють сторонам безпечно обмінюватися симетричними ключами.

Наприклад, симетричний ключ може бути зашифрований за допомогою відкритого ключа одержувача та безпечно надісланий [6].

Процес PKI заключається у наступному:

1) Сторона А шифрує симетричний ключ, використовуючи відкритий ключ Сторони Б.

2) Сторона В розшифровує симетричний ключ, використовуючи свій закритий ключ.

Діфі-Гелман – це протокол обміну ключами, який дозволяє двом сторонам безпечно використовувати симетричний ключ через незахищений канал без необхідності попередніх спільних секретів.

Процес розповсюдження ключів Діфі-Гелмана складається з таких етапів:

1) Обидві сторони погоджуються щодо публічної бази та модуля.

2) Кожна сторона генерує закритий ключ та обчислює відповідний відкритий ключ.

3) Відкриті ключі обмінюються, і кожна сторона поєднує свій закритий ключ з відкритим ключем іншої сторони для обчислення спільного симетричного ключа.

QKD – це передовий метод, який використовує квантову механіку для безпечного розподілу симетричних ключів. Безпека QKD базується на принципах квантової суперпозиції та заплутаності, що гарантує виявлення будь-якої спроби підслуховування обміну ключами [6].

Кожний з вищезазначених методів розподілу симетричних ключів має свої переваги та недоліки. Основні переваги та недоліки протоколів розподілу симетричних ключів представлено у табл. 1.6.

Таблиця 1.6 – Переваги та недоліки протоколів розподілу симетричних ключів

Назва протоколу	Переваги	Недоліки
Ручний розподіл	Безпечно, якщо ключ транспортується фізично без перехоплення	Непрактичний для великомасштабних систем та створює логістичні проблеми
PKI	Безпечний обмін ключами навіть через незахищений канал	Накладні витрати через використання асиметричного шифрування для обміну симетричними ключами
Діфі-Гелман	Не вимагає від сторін попередньої зустрічі чи обміну будь-якими секретними ключами	Вразливий до атак типу «людина посередині» (MITM, Man-In-The-Middle), якщо не проведено автентифікацію.

QKD	Теоретично захищений від прослуховування	Потребує спеціалізованого обладнання та ще не отримав широкого поширення

В асиметричній криптографії кожен учасник має пару ключів: відкритий ключ (який можна використовувати відкрито) та закритий ключ (який зберігається в таємниці). Основне використання асиметричної криптографії полягає в безпечному розподілі симетричних ключів, що дозволяє обом сторонам безпечно обмінюватися інформацією без попереднього використання секретного ключа [6].

Схему алгоритму асиметричного шифрування наведено на рис. 1.2.

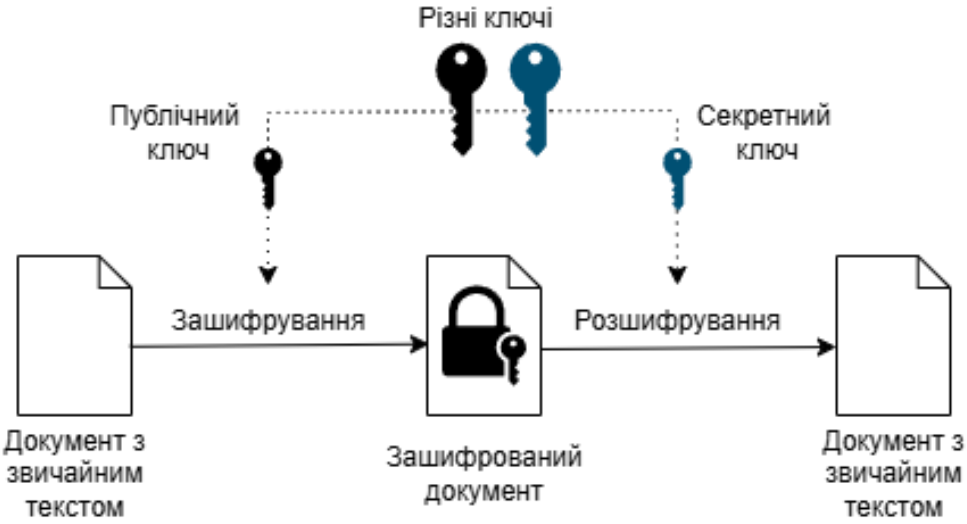

Рисунок 1.2 – Алгоритм асиметричного шифрування

Хоча асиметричний розподіл ключів значно спрощує обмін секретною інформацією, його використання також пов’язане з низкою викликів. Основні проблеми, що можуть виникати під час цього процесу, включають:

– Довіра до відкритого ключа. Одержувач повинен бути впевненим, що отриманий ним відкритий ключ є легітимним і не підроблений зловмисником. Цю проблему вирішує використання цифрових сертифікатів, які може перевірити довірений центр сертифікації (CA, Certification Authority).

– Масштабованість . У великомасштабних системах керування відкритими ключами багатьох користувачів може стати складним.

Для забезпечення надійності асиметричної криптографії необхідно використовувати спеціальні механізми обміну та перевірки відкритих ключів [6].

Існують різні протоколи асиметричного розподілу ключів:

– PKI;

– Павутиння довіри (WOT);

– Цифрові підписи.

PKI – це фреймворк, який використовує цифрові сертифікати для перевірки автентичності відкритих ключів. CA підписує цифрові сертифікати, пов’язуючи особу користувача з його відкритим ключем.

Процес PKI в асиметричній криптографії полягає у наступному:

1) Відправник отримує відкритий ключ одержувача від довіреного центру сертифікації.

2) Відправник шифрує повідомлення за допомогою відкритого ключа одержувача.

3) Одержувач розшифровує повідомлення, використовуючи свій закритий ключ.

Мережа довіри (WOT, Web Of Trust) є альтернативою PKI, де довіра до відкритих ключів встановлюється через прямі або непрямі зв’язки між користувачами. У WOT користувачі підписують відкриті ключі один одного для встановлення довіри [6].

Цифрові підписи використовують асиметричне шифрування для забезпечення автентичності повідомлень. Хоча цифрові підписи часто використовуються для перевірки цілісності даних, вони також можуть відігравати певну роль у розподілі ключів.

Процес використання цифрових підписів складається з таких етапів:

1) Відправник підписує повідомлення, використовуючи свій закритий ключ.

2) Одержувач може перевірити підпис за допомогою відкритого ключа відправника.

Кожний з вищезазначених методів розподілу асиметричних ключів має свої переваги та недоліки. Основні переваги та недоліки протоколів розподілу асиметричних ключів наведено табл. 1.7.

Таблиця 1.7 – Переваги та недоліки протоколів розподілу асиметричних ключів

Назва протоколу	Переваги	Недоліки
PKI	Безпечний розподіл ключів з перевіркою особи	Спирається на довірений центр сертифікації, а також є накладні витрати на керування сертифікатами
WOT	Децентралізований та гнучкіший, ніж PKI	Може стати громіздким зі зростанням кількості учасників, а також йому бракує централізованого органу управління, який забезпечує PKI
Цифрові підписи	Забезпечує цілісність, автентичність та неможливість заперечення	Не використовується безпосередньо для обміну ключами, але може допомогти перевірити легітимність відкритих ключів

На практиці сучасні криптографічні системи часто використовують гібридний розподіл ключів , де асиметрична криптографія використовується для безпечного обміну симетричним ключем, а потім симетричне шифрування використовується для фактичної передачі даних.

Це поєднує переваги обох типів криптографії: ефективність симетричного шифрування та можливості безпечного обміну ключами асиметричного шифрування [6].

Процес розподілу гібридних ключів складається з наступних етапів:

1) Обмін ключами . Сторони використовують асиметричну криптографічну систему (наприклад, RSA або Діфі-Гелман) для безпечного обміну симетричним ключем.

2) Зашифрування. Після обміну ключами фактичне спілкування шифрується за допомогою симетричного алгоритму шифрування, такого як AES.

3) Розшифрування. Отримувач розшифровує дані за допомогою спільного симетричного ключа.

Розподіл ключів є фундаментальним аспектом криптографічних систем, що забезпечує безпечний обмін необхідними ключами та їх використання для зашифрування та розшифрування.

Незалежно від того, чи використовується симетрична чи асиметрична криптографія, чи гібридний підхід, метод розподілу ключів повинен забезпечувати конфіденційність, автентичність та цілісність [6].

На основі проведеного аналізу протоколів розподілу ключів можна зробити висновок, що кожен із розглянутих протоколів має свої переваги, недоліки та сферу застосування.

Серед розглянутих протоколів, протокол Діфі–Гелмана займає особливе місце, оскільки поєднує у собі високу криптографічну стійкість і практичність реалізації. Він дозволяє двом сторонам безпечно сформувати спільний симетричний ключ через відкритий (незахищений) канал зв’язку, не потребуючи попереднього обміну секретами або використання централізованих довірених служб. Безпека протоколу ґрунтується на складності задачі обчислення дискретного логарифма, що робить його стійким до більшості сучасних криптографічних атак.

Завдяки своїм властивостям – простоті реалізації, високому рівню безпеки, відсутності необхідності в попередньому розподілі секретних даних та сумісності з іншими криптографічними алгоритмами — протокол Діфі–Гелмана став стандартом у багатьох сучасних системах безпечного обміну даними (зокрема, TLS, SSH, IPsec) [6].

У межах роботи для реалізації протоколу обміну інформацією було обрано саме протокол угоди про ключі Діфі–Гелмана, оскільки він забезпечує оптимальне співвідношення між рівнем безпеки, ефективністю та універсальністю застосування. Використання цього протоколу дозволяє створити надійний механізм формування спільного секретного ключа між сторонами без необхідності передачі його відкритими каналами, що повністю відповідає вимогам до розроблюваного програмного засобу для захищеного передавання інформації.

1.3 Протокол угоди про ключі Діфі–Гелмана

Протокол угоди про ключі Діфі–Гелмана – це метод обміну криптографічними ключами. Один з перших практичних прикладів узгодження ключа, що дозволяє двом учасникам, що не мають жодних попередніх даних один про одного, отримати спільний секретний ключ із використанням незахищеного каналу зв’язку. Цей ключ можна використати для шифрування наступних сеансів зв’язку, що використовують шифр з симетричним ключем.

Алгоритм обміну ключами необхідний у комунікації та криптографії з кількох причин:

1) Це дозволяє двом або більше сторонам узгодити секретний ключ, не розкриваючи його потенційним зловмисникам , який потім використовується для шифрування та дешифрування, що життєво важливо для збереження конфіденційності комунікації.

2) Збереження цілісності даних також було серйозною проблемою в цифровому зв’язку, де дані завжди вразливі до спотворення під час передачі. Алгоритм обміну ключами допомагає зберегти цілісність переданих даних, запобігаючи несанкціонованій зміні або підробці даних під час передачі.

3) Алгоритм обміну ключами сприяє автентифікації сторін, що взаємодіють, перевіряє, ким вони себе видають, тим самим підвищуючи ризик атаки MITM [7].

Таким чином, поряд із шифруванням даних для збереження конфіденційності зв'язку, також був потрібен алгоритм обміну ключами для підтримки цілісності та авторизованого доступу до інформації.

Безпека обміну ключами Діфі-Гелмана спирається на математичні властивості модульного піднесення до степеня та задачі дискретного логарифмування.

Модульне піднесення до степеня – це процес піднесення числа до степеня та отримання лишку від ділення на модуль. В алгоритмі Діфі-Гелмана модульне піднесення до степеня використовується для обчислення A та B, якими потім обмінюються сторони.

З іншого боку, задача дискретного логарифмування – це завдання знаходження степеня за основою, модулем та результатом модульного піднесення до степеня. Безпека обміну ключами Діфі-Гелмана базується на припущенні, що задачу дискретного логарифмування неможливо вирішити обчислювально, що ускладнює для зловмисника обчислення спільного секретного ключа [7].

Протокол неавтентифікаційного розподілу ключів Діфі-Гелмана полягає у наступному.

Відкриті параметри: просте p і генератор a групи $\mathbf{Z}_p (2 \leq a \leq p-2)$.

1) Учасник A вибирає випадкове число $x (1 < x \leq p-2)$, що зберігається в секреті, і посилає учаснику B таке повідомлення:

$$A \to B: a^x \ mod \ p \tag{1.1}$$

2) Учасник B вибирає випадкове число $(1 < y \leq p-2)$, що зберігається в секреті, і посилає учаснику A таке повідомлення:

$$A \leftarrow B: a^y \ mod \ p \tag{1.2}$$

3) Учасник А одержує $a^y \ mod \ p$ й обчислює сеансовий ключ:

$$K = (a^y \ mod \ p)^x \ mod \ p = a^{xy} \ mod \ p \tag{1.3}$$

Учасник В одержує $a^x \ mod \ p$ й обчислює сеансовий ключ:

$$K = (a^x \ mod \ p)^y \ mod \ p = a^{xy} \ mod \ p \tag{1.4}$$

Переваги обміну ключами Діфі-Гелмана включають:

1) Пряма секретність – протокол дозволяє сторонам генерувати новий спільний секретний ключ для кожного сеансу зв’язку, гарантуючи, що компрометація одного ключа не вплине на безпеку минулих або майбутніх сеансів.

2) Масштабованість – обмін ключами Діфі-Гелмана добре масштабується залежно від кількості учасників, оскільки кожній стороні потрібно виконати лише невелику кількість степенів для обчислення спільного секретного ключа.

3) Відсутність попереднього спілкування – протокол не вимагає жодного попереднього спілкування чи обміну інформацією між сторонами, що робить його придатним для використання в ситуаціях, коли встановлення попередньої довіри є складним [7].

Обмеження обміну ключами Діфі-Гелмана включають:

1) Вразливість до атак MITM – протокол не забезпечує автентифікацію сторін, що робить його вразливим до атак «людина посередині», коли зловмисник може видати себе за одну або обидві сторони та перехопити або змінити зв’язок. Щоб зменшити цей ризик, обмін ключами Діфі-Гелмана часто поєднується з цифровими підписами або іншими механізмами автентифікації.

2) Обчислювальні витрати – обмін ключами Діфі-Гелмана передбачає модульне піднесення до степеня, що може бути обчислювально ресурсоємним, особливо для великих простих чисел. Однак це обмеження можна вирішити, використовуючи ефективні алгоритми для модульного піднесення до степеня або реалізуючи протокол з криптографією еліптичних кривих , яка вимагає менших розмірів ключів для еквівалентної безпеки.

3) Відсутність шифрування або захисту цілісності даних – протокол надає лише метод для встановлення спільного секретного ключа; він не пропонує шифрування даних або захисту цілісності. Для захисту зв'язку спільний секретний ключ має використовуватися разом із симетричним алгоритмом шифрування та кодом автентифікації повідомлення (MAC, Message Authentication Code) або автентифікованим шифруванням [7].

Алгоритм Діфі-Гелмана широко використовується в різних реальних програмах для встановлення безпечних каналів зв'язку між сторонами. Деякі поширені застосування включають:

1) Безпека транспортного рівня (TLS, Transport Layer Security) – як ключовий компонент протоколу TLS, обмін ключами Діфі-Гелмана використовується для встановлення спільного секретного ключа для безпечного зв'язку між веб-браузерами та серверами, захищаючи конфіденційні дані, такі як облікові дані для входу, платіжна інформація та особисті дані.

2) Безпечна оболонка (SSH, Secure Shell) – обмін ключами Діфі-Гелмана використовується в протоколі SSH для забезпечення безпечного віддаленого доступу та керування комп'ютерними системами через незахищену мережу.

3) Віртуальні приватні мережі (VPN, Virtual Private Network) – у VPN, що використовують протокол IPsec, обмін ключами Діфі-Гелмана використовується під час процесу обміну ключами Інтернету (IKE, Internet Key Exchange) для встановлення спільного секретного ключа для захисту передачі даних між кінцевими точками VPN.

4) Програми миттєвого обміну повідомленнями та голосового зв'язку через IP (VoIP, Voice over Internet Protocol) – обмін ключами Діфі-Гелмана

використовується в різних програмах миттєвого обміну повідомленнями та VoIP, таких як Signal та WhatsApp, для встановлення наскрізного шифрування, захищаючи конфіденційність повідомлень та дзвінків.

5) Шифрування електронної пошти – такі протоколи, як Pretty Good Privacy (PGP) та Secure/Multipurpose Internet Mail Extensions (S/MIME), можуть використовувати обмін ключами Діфі-Гелмана для безпечного обміну симетричними ключами для шифрування та дешифрування повідомлень електронної пошти.

Існують різні варіації алгоритму Діфі-Гелмана, зокрема:

1) Еліптична крива Діфі-Гелмана (ECDH). Цей варіант використовує криптографію еліптичної кривої, яка пропонує еквівалентну безпеку з меншими розмірами ключів, зменшуючи обчислювальні вимоги та покращуючи продуктивність.

2) Анонімний протокол Діфі-Гелмана. Цей варіант не забезпечує автентифікацію, що робить протокол вразливим до атак MITM.

3) Статичний алгоритм Діфі-Гелмана. У цьому варіанті принаймні одна сторона використовує фіксований відкритий ключ, який не забезпечує прямої секретності.

4) Ефемерний Діфі-Гелмана полягає у тому, що обидві сторони генерують тимчасові відкриті ключі для кожного сеансу, забезпечуючи пряму секретність, яка гарантує, що скомпрометований довгостроковий ключ не вплине на ключі попередніх сеансів.

5) Потрійний ключ Діфі-Гелмана. Цей протокол поєднує ефемерний ключ Діфі-Гелмана з додатковою парою ключів для забезпечення взаємної автентифікації та прямої секретності.

6) ElGamal. Це схема шифрування з відкритим ключем, заснована на обміні ключами Діфі-Гелмана, що дозволяє безпечне шифрування та дешифрування повідомлень [7].

1.4 Висновки з розділу

У результаті проведеного аналізу інформаційних джерел встановлено, що сучасні системи захищеного цифрового обміну повідомленнями базуються на механізмах узгодження та розподілу ключів, від яких безпосередньо залежить рівень криптографічної безпеки кінцевої системи.

Месенджери реалізують різні моделі встановлення спільного секрету: Telegram використовує серверно-орієнтовану архітектуру, де AuthKey формується між клієнтом і сервером, тоді як WhatsApp, Signal та нова версія Facebook Messenger застосовують Signal-підхід з використанням pre-keys, X3DH та Double Ratchet – що забезпечує асинхронність, forward secrecy та стійкість до компрометації минулих і майбутніх ключів. Viber також використовує E2E-модель, але без розширеної pre-key інфраструктури. Таким чином, сучасна тенденція еволюції месенджерів полягає у переході від статичних сесій до безперервної ротації ключів з кожним повідомленням, що суттєво підвищує криптостійкість.

Аналіз протоколів розподілу ключів показав, що попри наявність різних підходів (ручний розподіл, PKI, Діфі-Гелман, QKD), жоден із них не є універсально застосовним для всіх сценаріїв. На практиці найпоширенішою стала гібридна модель, де асиметрична криптографія використовується для початкового встановлення секрету, а симетрична – для подальшого шифрування даних. Це дозволяє поєднати високу продуктивність симетричних шифрів із можливістю безпечного встановлення ключів без попередньої довіри.

Протокол узгодження ключів Діфі–Гелмана забезпечує можливість сторін сформувати спільний секрет без попередньо спільних даних та без передачі секрету каналом зв’язку. Незважаючи на відсутність вбудованої автентифікації, Діфі-Гелмана залишається базовим фундаментом для більшості сучасних криптоархітектур: TLS, SSH, IPsec, а також для E2E-месенджерів, де він використовується як складова X3DH. Пряма секретність, масштабованість, універсальність застосування та сумісність з іншими криптоалгоритмами роблять Діфі-Гелмана практичним і надійним рішенням на рівні індустріальних стандартів.

Таким чином, обрана в межах даної роботи модель побудови захищеного каналу на основі протоколу угоди про ключі Діфі–Гелмана є обґрунтованою, оскільки цей протокол оптимально поєднує математичну стійкість, відсутність залежності від попередньої довіри та можливість інтеграції з сучасними механізмами автентифікації, що повністю відповідає вимогам до проектованого програмного засобу для захищеного обміну інформацією.

2 РОЗРОБКА ПРОТОКОЛУ ОБМІНУ ІНФОРМАЦІЄЮ

2.1 Узагальнений протокол обміну інформацією

Узагальнений протокол обміну інформацією є центральним компонентом розробленої криптографічної системи, що забезпечує безпечну взаємодію двох сторін – користувача A та користувача B – у процесі встановлення спільного секретного сеансу. Його основна мета полягає у створенні захищеного каналу зв’язку, у якому забезпечується автентичність переданих даних.

Узагальнений протокол базується на комбінації кількох криптографічних механізмів, що реалізують різні етапи обміну інформацією:

1) Автентифікація сторін – сторони обміну повинні мати унікальні параметри, які дозволяють підтвердити їхню автентичність.

2) Узгодження параметрів обміну – визначаються глобальні параметри системи: велике просте число p, генератор g групи $\mathbf{Z}_p$, формат повідомлень, довжини ключів та інші технічні константи [8].

3) Встановлення спільного секрету – сторони незалежно обчислюють однакове секретне значення, не розкриваючи його в процесі обміну.

4) Генерування робочого секретного ключа – спільний секрет трансформується у внутрішній секретний ключ, який використовується для зашифрування і розшифрування, формування матриці обертання.

5) Передача зашифрованого повідомлення з кодом автентифікації.

6) Після розшифрування виконується порівняння переданого коду автентифікації з сформованим.

Таким чином, протокол поєднує елементи обміну ключами, симетричного шифрування та автентифікації повідомлень у єдину логічну структуру [8].

Комунікаційна модель передбачає обмін між двома учасниками:

1) Користувач А (ініціатор) – починає встановлення з'єднання.

2) Користувач В (одержувач) – приймає запит та відповідає.

Процес описується такою послідовністю кроків:

1) Ініціалізація сеансу. Сторона А ініціює обмін, надсилаючи запит на встановлення з'єднання. В запиті вказуються загальні параметри, необхідні для початку протоколу, зокрема ідентифікатор користувача, часові мітки та відкриті системні константи. Відбувається автентифікація сторін.

2) Узгодження відкритих параметрів. Сторони узгоджують спільні відкриті параметри (p,g), що визначають поле $\mathbf{Z}_p$ для подальших обчислень.

3) Генерування спільного секретного ключа. Цей етап базується на протоколі Діфі-Гелмана, який виконує узгодження спільного секрету без його передачі по мережі. У результаті обидві сторони отримують ідентичне значення спільного секретного параметра K, який стає базою для подальшої криптографічної взаємодії [8].

4) Генерування сеансового ключа. Отримане секретне значення K передається в модуль генерування, де воно проходить процедуру ущільнення – багаторівневе кватерніонне перетворення, що зменшує розрядність до 64 біт і створює робочий ключ K_0. Цей ключ використовується для побудови поворотної матриці, яка використовується для зашифрування/розшифрування першого блоку повідомлення [9,10].

5) Передача автентифікованого повідомлення. Після встановлення спільного ключа обидві сторони можуть безпечно обмінюватися повідомленнями. Кожне повідомлення зашифровується та доповнюється MAC. Процес передавання представлено узагальнено:

$$A \to B: E_{K_0}(\mathbf{M}) || \, MAC, \qquad (2.)$$

де $E_{K_0}(\mathbf{M})$ – результат зашифрування повідомлення **M** з використанням ключа K_0.

6) Перевірка автентичності. Сторона B після отримання зашифрованого повідомлення, розшифровує його, обчислює MAC* і порівнює з отриманим MAC. Якщо MAC*=MAC, то порушення цілісності повідомлення немає і воно приймається до обробки [11].

Представлення протоколу у вигляді послідовності інформаційних кроків наведено у табл. 2.1.

Таблиця 2.1 – Послідовність інформаційних кроків протоколу

Етап	Напрямок	Опис дії
1	$A \to B$	Надсилання запиту на ініціацію обміну
2	$B \to A$	Підтвердження прийняття та передача узгоджених параметрів
3	$A \leftrightarrow B$	Обмін даними згідно за протоколом Діфі-Гелмана
4	A, B	Обчислення спільного секрету K
5	A, B	Формування робочого 64-бітного ключа K_0
6	A, B	Формування поворотної матриці
7	$A \to B$	Надсилання зашифрованого повідомленнями з MAC-кодом
8	B	Перевірка автентичності та підтвердження цілісності повідомлення

Узагальнений протокол обміну інформацією є симетричним, оскільки обидві сторони незалежно обчислюють однаковий спільний секретний ключ, не передаючи його через канал зв’язку. Стійкість цього процесу базується на проблемі дискретного логарифма.

Автентичність повідомлень забезпечується використанням MAC-коду, який підтверджує їх цілісність. Висока криптографічна стійкість досягається поєднанням класичних математичних принципів і кватерніонної алгебри при формуванні ключів.

Таким чином, узагальнений протокол обміну інформацією у розробленій системі забезпечує повний цикл захищеної взаємодії – від ініціації з'єднання та узгодження параметрів до передачі автентифікованих зашифрованих даних.

Протокол інтегрує в собі механізм узгодження ключа на основі протоколу Діфі–Гелмана, модуль генерування 64-бітного кватерніонного ключа та схему автентифікованого шифрування, що разом утворюють єдиний комплекс безпечного обміну інформацією.

2.2 Реалізація протоколу Діфі-Гелмана

Реалізація протоколу Діфі–Гелмана у розробленій системі забезпечує узгодження спільного секретного ключа між двома сторонами – користувачами A та B через незахищений канал зв'язку без попереднього обміну секретною інформацією. Основу протоколу становлять операції модульного піднесення до степеня в мультиплікативній групі залишків за простим модулем, що гарантує криптографічну стійкість на основі складності задачі дискретного логарифмування.

Обидві сторони домовляються про використання спільних відкритих параметрів:

$$p - \text{велике просте число}, g - \text{первісний корінь (генератор)у } Z_p^*, \qquad (2.1)$$

де $2 \leq g \leq p - 2$.

Вибір параметра p здійснюється з використанням тесту простоти Мілера–Рабіна, що гарантує з високою ймовірністю його простоту [7,8]. Для цього p формується як непарне випадкове число заданої довжини (наприклад, 1024 біт) і перевіряється за умовами:

$$p - 1 = 2^s t, \;\; t - \text{непарне}, s \geq 1, \tag{2.1}$$

після чого перевіряється виконання критерію:

$$a^t \equiv \pm 1 \; (mod \; p) \text{ або } a^{2^r t} \equiv -1 \; (mod \; p) \tag{2.1}$$

для кількох баз $a \in [2, p-2]$. Якщо хоча б один тест не виконується, число відкидається.

Користувач А генерує випадкове приватне число:

$$x_A \in [2, p-2], \tag{2.1}$$

використовуючи криптографічно стійкий генератор випадкових бітів, що відповідає стандарту ISO/IEC 18031 [12], який забезпечує непередбачуваність і рівномірний розподіл значень. Відповідний відкритий ключ обчислюється за формулою:

$$y_A = g^{x_A} \; mod \; p \tag{2.1}$$

Аналогічно користувач В генерує приватний параметр

$$x_B \in [2, p-2], \tag{2.1}$$

і обчислює відкритий ключ:

$$y_B = g^{x_B} \, mod \, p \tag{2.1}$$

Користувач А передає y_A користувачу В, а користувач В надсилає y_B користувачу А через відкритий канал. Незважаючи на відкритість переданих значень, неможливо відновити приватні ключі x_A, x_B, оскільки для цього потрібно розв'язати задачу дискретного логарифмування:

$$x = \log_g y \, (mod \, p) \tag{2.1}$$

яка є обчислювально нездійсненною для великих p [9,10].

Після обміну відкритими ключами обидві сторони незалежно обчислюють спільний секретний ключ *K*:

$$K_A = (y_B)^{x_A} \, mod \, p = g^{x_A x_B} \, mod \, p \tag{2.1}$$
$$K_B = (y_A)^{x_B} \, mod \, p = g^{x_A x_B} \, mod \, p$$

Таким чином, обидві сторони отримують однакове значення:

$$K = g^{x_A x_B} \, mod \, p \tag{2.1}$$

яке ніколи не передається по каналу зв'язку [7,8].

Для подальшого використання в системі автентифікованого шифрування спільний секрет *K* виступає джерелом для формування 1024-бітного ключа K_s, який потім ущільнюється до 64-бітного значення K_0.

2.3 Процедура генерування секретного ключа

Процес формування секретного ключа є початковим етапом. Його метою є створення 64-бітного ключа K_0 з початкового 1024-бітного секретного ключа, який отримано за протоколом Діфі-Гелмана.

Початкове 1024-бітне число K ділиться на чотири рівні частини по 256 біт кожна:

$$K = (K_1, K_2, K_3, K_4), \;\; K_i \in \; [0, 2^{256} - 1]. \tag{2.1}$$

Кожна частина далі поділяється на чотири 64-бітні компоненти:

$$K_1 = (a_1, b_1, c_1, d_1), K_2 = (a_2, b_2, c_2, d_2), K_3 = (a_3, b_3, c_3, d_3), \\ K_4 = (a_4, b_4, c_4, d_4), \tag{2.2}$$

де $(a_i, b_i, c_i, d_i) \in [0, 2^{64} - 1]$.

Далі кожна частина подається як кватерніон [9]:

$$\begin{aligned} K_1 &= a_1 + b_1 i + c_1 j + d_1 k, \\ K_2 &= a_2 + b_2 i + c_2 j + d_2 k, \\ K_3 &= a_3 + b_3 i + c_3 j + d_3 k, \\ K_4 &= a_4 + b_4 i + c_4 j + d_4 k. \end{aligned} \tag{2.3}$$

Далі виконується послідовне множення кватерніонів:

$$D_1 = K_1 K_2 = \begin{cases} a_1 a_2 - b_1 b_2 - c_1 c_2 - d_1 d_2 = w_1, \\ a_1 b_2 + b_1 a_2 + c_1 d_2 - d_1 c_2 = z_1, \\ a_1 c_2 - b_1 d_2 + c_1 a_2 + d_1 b_2 = x_1, \\ a_1 d_2 + b_1 c_2 - c_1 b_2 + d_1 a_2 = k_1. \end{cases}$$

$$D_2 = K_3 K_4 = \begin{cases} a_3 a_4 - b_3 b_4 - c_3 c_4 - d_3 d_4 = w_2, \\ a_3 b_4 + b_3 a_4 + c_3 d_4 - d_3 c_4 = z_2, \\ a_3 c_4 - b_3 d_4 + c_3 a_4 + d_3 b_4 = x_2, \\ a_3 d_4 + b_3 c_4 - c_3 b_4 + d_3 a_4 = k_2. \end{cases} \tag{2.5}$$

$$Q = D_1 D_2 = \begin{cases} w_1 w_2 - z_1 z_2 - x_1 x_2 - k_1 k_2 = q_1 \\ w_1 z_2 + z_1 w_2 + x_1 k_2 - k_1 x_2 = q_2 \\ w_1 x_2 - z_1 k_2 + x_1 w_2 + k_1 z_2 = q_3 \\ w_1 k_2 + z_1 x_2 - x_1 z_2 + k_1 w_2. = q_4 \end{cases}$$

Усі операції виконуються за модулем 2^{64} [13].

$$K_0 = (q_1 + q_2 + q_3 + q_4)\ mod\ 2^{64}$$

Таким чином, початковий 1024-бітний ключ згортається до 64-бітного представлення.

Процедура ущільнення має такі властивості:

– K_0 є геш-значенням для коду K.

– Нелінійність – кватерніонне множення створює складну залежність між вхідними частинами, що унеможливлює лінійне відновлення початкових компонент.

– Незворотність – через геш-значення.

– Ентропійність – усі 1024 біти початкового ключа впливають на кінцеве значення K_0, забезпечуючи повне змішування.

2.4 Обчислення поворотної матриці

Після отримання 64-бітного ключа K_0, що є результатом багатоступеневої процедури згортання початкового 1024-бітного секретного значення, виконується його перетворення у кватерніон, на основі якого формується поворотна матриця.

Цей етап має важливе значення, оскільки саме кватерніонна форма представлення ключа забезпечує можливість побудови ортогональної поворотної матриці, що використовується для реалізації першої стадії шифрування – блочного матричного перетворення.

Отримане 64-бітне число K_0 розбивається на чотири 16-бітні складові:

$$K_0 = (w, x, y, z), \tag{2.10}$$

де $w, x, y, z \in [0, 2^{16})$.

Кожна з цих компонент визначає параметри обертання у чотиривимірному просторі, а їх поєднання утворює кватерніон:

$$q = w + xi + yj + zk, \tag{2.11}$$

що задовольняє класичні правила множення базових елементів:

$$i^2 = j^2 = k^2 = ijk = -1 \tag{2.12}$$

Таким чином, кватерніон q є узагальненням поняття комплексного числа та забезпечує компактне і зручне математичне представлення просторових обертань.

Норма кватерніона визначається як сума квадратів його компонент за модулем обраного простого числа $p = 65537$ [14]:

$$N = (w^2 + x^2 + y^2 + z^2)\ mod\ p \tag{2.13}$$

Норма характеризує «довжину» кватерніона у чотиривимірному просторі. Для побудови ортогональної поворотної матриці необхідно, щоб кватерніон був одиничним, тобто мав норму, рівну 1. Якщо ця умова не виконується, потрібно виконати нормування.

Нормування полягає у приведенні довжини кватерніона до одиниці. Для цього вводиться масштабувальний коефіцієнт t, який задовольняє рівняння:

$$t^2 \equiv N^{-1}\ (mod\ p), \tag{2.14}$$

де N^{-1} – мультиплікативно обернене до N число в полі Z_p.

Нормований кватерніон обчислюється за формулою:

$$\hat{q} = t \cdot q \ mod \ p, \tag{2.15}$$

після чого забезпечується властивість:

$$||\hat{q}||^2 \equiv 1 \ (mod \ p), \tag{2.16}$$

тобто $\hat{q}$ є одиничним нормованим кватерніоном [14].

У випадку, якщо N не є квадратичним лишком у полі Z_p, тобто рівняння для t немає розв'язку, виконується мінімальна корекція однієї з компонент (зазвичай z) на невелике значення δ до тих пір, поки оновлена норма N' не стане квадратичним лишком. Це гарантує існування квадратного кореня та можливість коректної нормалізації.

Такий підхід мінімізує спотворення початкових даних і зберігає стійкість алгоритму.

Після нормалізації кватерніона на його основі формується ортогональна поворотна матриця $\mathbf{\Gamma}(\hat{q})$, елементи якої визначаються за стандартною кватерніонною формулою [15]:

$$\mathbf{\Gamma}(\hat{q}) = \begin{bmatrix} 1 - 2(y^2 + z^2) & 2(xy - wz) & 2(xz + wy) \\ 2(xy + wz) & 1 - 2(x^2 + z^2) & 2(yz - wx) \\ 2(xz - wy) & 2(yz + wx) & 1 - 2(x^2 + y^2) \end{bmatrix} \tag{2.17}$$

Ця поворотна матриця описує тривимірне обертання, яке відповідає напрямку та величині, заданим кватерніоном $\hat{q}$. Її головною властивістю є ортогональність:

$$\Gamma(\hat{q})^{\mathrm{T}}\Gamma(\hat{q}) \equiv I \ (mod \ p), \tag{2.18}$$

де I – одинична матриця. Це означає, що обернена матриця для розшифрування є транспонованою для матриці обертання.

Використання поворотної матриці як блочного шифру забезпечує рівномірне розподілення впливу кожного біта ключа на елементи зашифрованого блоку. Завдяки цьому досягається висока дифузія та стійкість до статистичних атак. Крім того, ортогональність гарантує, що обернення процесу не призводить до накопичення похибок при багатократних обчисленнях у полі $\mathbf{Z}_p$.

Таким чином, поворотна матриця $\mathbf{\Gamma}(\hat{q})$ виступає ядром блочного шифру, яке зв'язує алгебраїчні властивості кватерніонів із практичними вимогами до безпеки криптографічних систем [15].

2.5 Стандарти генерування випадкових чисел

У сучасній криптографії та інформаційній безпеці генерування випадкових чисел є фундаментальною складовою захищених обчислень. Від якості та непередбачуваності випадкових чисел залежить криптостійкість алгоритмів шифрування, надійність ключів, автентичність цифрових підписів і загальна безпека інформаційного обміну.

Для забезпечення високого рівня безпеки створено низку міжнародних і національних стандартів, що регламентують методи генерування випадкових і псевдовипадкових чисел, визначають вимоги до джерел ентропії, критерії статистичної оцінки випадковості та процедури валідації генераторів. Використання таких стандартів забезпечує криптографічну стійкість систем, що реалізують цифрові підписи, шифрування даних і протоколи обміну ключами.

До найвідоміших належать:

– NIST SP 800-90A/B/C – стандарти, що описують архітектуру детермінованих генераторів випадкових чисел (DRBG) та вимоги до ентропійних джерел.

– ISO/IEC 18031 – міжнародний стандарт, який визначає принципи, вимоги та алгоритмічні засоби генерування випадкових і псевдовипадкових чисел у криптографічних додатках.

– FIPS 140-3 і FIPS 186-5 – стандарти, що визначають вимоги до криптографічних модулів і процесів генерування ключів для алгоритмів цифрового підпису.

– RFC 4086 – рекомендації IETF щодо безпечної генерування випадкових чисел у мережевих протоколах.

Для протоколу обміну ключами Діфі–Гелмана було обрано міжнародний стандарт ISO/IEC 18031, який визначає вимоги до криптографічних генераторів випадкових бітів (RBG, Random Bit Generators).

Стандарт ISO/IEC 18031 [12] описує принципи побудови, функціонування та оцінки генераторів випадкових бітів (ГВБ), що застосовуються у криптографічних системах. Його метою є забезпечення криптографічної надійності – тобто неможливості передбачити вихідні значення навіть за умови часткового розкриття внутрішнього стану генератора.

Документ визначає дві основні категорії генераторів:

1) Фізичні генератори випадкових чисел (TRNG, True Random Number Generators) – ґрунтуються на апаратних джерелах ентропії, таких як електричний шум, теплові флуктуації або квантові процеси.

2) Детерміновані генератори випадкових чисел (DRBG, Deterministic Random Bit Generators) – створюють послідовності бітів на основі криптографічних алгоритмів (наприклад, AES, SHA-2, HMAC, CTR), використовуючи початкове насіння (seed) з високою ентропією.

Генератор випадкових чисел відповідно до ISO/IEC 18031 має три основні компоненти:

1) Джерело ентропії – забезпечує отримання непередбачуваних фізичних подій для ініціалізації генератора.

2) Механізм обробки ентропії – перетворює «сирі» випадкові дані в однорідну, рівномірно розподілену послідовність бітів.

3) Детермінований криптографічний механізм – створює довгі криптографічно стійкі послідовності бітів із короткого насіння, підтримуючи властивості непередбачуваності та незалежності результатів [12].

Відповідно до стандарту ISO/IEC 18031, генератор повинен відповідати таким вимогам:

– Непередбачуваність. Знання попередніх або наступних вихідних бітів не дозволяє обчислити поточні значення.

– Стійкість до компрометації. Навіть у разі витоку внутрішнього стану генератора попередні вихідні дані не повинні бути відновлені.

– Перевірюваність. Генератор має проходити статистичні тести випадковості (наприклад, NIST SP 800-22 або Diehard tests).

– Самовідновлення. Генератор повинен мати здатність до оновлення насіння для підвищення ентропії в процесі роботи.

– Криптографічна узгодженість. Застосовувані механізми повинні відповідати алгоритмам, визнаним безпечними на міжнародному рівні (AES, HMAC, SHA-3 тощо) [12].

Згідно з ISO/IEC 18031, генератори випадкових чисел застосовуються для:

– створення ключів шифрування та ініціалізаційних векторів (IV);

– формування приватних ключів цифрових підписів;

– генерування сеансових ключів у протоколах обміну інформацією;

– створення псевдовипадкових послідовностей для автентифікаційних механізмів;

– формування одноразових паролів (OTP), PIN-кодів і токенів доступу.

Крім того, ISO/IEC 18031 передбачає, що криптографічно безпечний генератор має підтримувати періодичне оновлення внутрішнього стану для запобігання накопиченню корельованих результатів та зниженню рівня ентропії.

Переваги використання стандарту ISO/IEC 18031 полягають у наступному:

– забезпечує уніфікований підхід до побудови криптографічно стійких генераторів;

– підвищує довіру до безпеки реалізованих систем;

– дозволяє проходити міжнародну сертифікацію криптографічних рішень;

– забезпечує сумісність із іншими стандартами, такими як ISO/IEC 19790 (вимоги до криптографічних модулів) та ISO/IEC 29192 (легковагова криптографія) [12].

2.6 Тест простоти Мілера-Рабіна

Для ефективного та надійного визначення простоти чисел широко застосовується тест Мілера-Рабіна – ймовірнісний алгоритм, який дозволяє перевіряти, чи є число простим, з високою ймовірністю. Цей тест використовується не лише для загальної генерування простих чисел, а й безпосередньо в таких криптографічних протоколах, як протокол Діфі-Гелмана, де прості числа великого порядку слугують основою для обчислення ключів обміну та забезпечують стійкість шифрування до атак [16].

Мала теорема Ферма стверджує, що якщо m – просте число, то $a^{m-1} \equiv 1(mod\ m)$ для будь-якої основи a, яка є взаємно простою з р. Цю умову a^{p-1} можна використовувати як тест на простоту. Якщо ця умова виконується, то ми кажемо, що m – ймовірне просте число з основою a. На жаль, існують складені числа, m такі що $a^{p-1} \equiv 1(mod\ m)$ для всіх a чисел , які є взаємно простими з m (такі складені числа m називаються числами Кармайкла). Найменше число Кармайкла дорівнює $561 = 17 \times 33$. Застосовуючи Малу теорему Ферма до числа Кармайкла m, ми можемо сплутати його з простим числом [16].

Тест Мілера-Рабіна є набагато кращим інструментом для ідентифікації чисел Кармайкла. Він є більш ефективним ймовірнісним тестом, в якому використовується критерій, в кінцевому рахунку, оснований на факті, що для простого модуля квадратними коренями з одиниці є лише числа ± 1, а для

складеного непарного модуля $n = uv, (uv) = 1,$ число таких коренів більше двох.

Нехай *n* – непарне натуральне число. Тоді можна записати:

$$n - 1 = 2^s t, \tag{2.1}$$

де t – непарне і $s \geq 1$.

Якщо число *n* – непарне натуральне число. Тоді можна записати:

$$a^{n-1} \equiv 1(mod\ n), \tag{2.2}$$

при (*a*,*n*) = 1.

Тому квадратні корені з одиниці мають вигляд:

$$a^{(n-1)/2} = \pm 1(mod\ n), \tag{2.3}$$

де показник рівний $2^{s-1}t$ [18].

Це означає, що в послідовності $a^t, a^{2t}, \dots, a^{2^{s-1}t}$ лишків за простим модулем, які є послідовними квадратами числа a^t, або з'явиться$-1(mod\ n)$, або всі ці лишки порівнянні з одиницею, тобто $a^t = 1(mod\ n)$. Зазначимо, що при простому *n* лівіше $-1(mod\ n)$ можуть розташовуватися лише лишки не рівні $\pm 1(mod\ n)$.

Якщо *n* – складене, то можливі й інші варіанти, оскільки в цьому випадку крім ± 1 існують інші корені з одиниці за модулем *n*.

Заснований на даному зауваженні тест Мілера-Рабіна полягає в тому, що:

1) псевдовипадково вибираємо залишок $a \in \{2, \dots, n-1\}$ і перевіряємо умову $(a, n) = 1$. Якщо умова не виконана, значить, n – складене і робота закінчена;

2) обчислюємо $a^t = mod\ n$. Якщо $a^t = \pm 1\ mod\ n$, то не виключено, що число n – просте і необхідно перейти на початок, щоб повторити тест для іншої основи;

3) обчислюємо послідовно лишки чисел $a^{2t}, \dots, a^{2^{s-1}t}$,за модулем n, поки не з'явиться (-1) , або не вичерпається список;

4) якщо (-1) знайдено в списку, то не виключено, що число n – просте і необхідно перейти на початок, щоб повторити тест для іншої основи;

5) якщо жодне число із списку не порівнянно з (-1), то число n – складене і необхідно закінчити роботу [16].

Як і для інших імовірнісних тестів, існують складені числа n, які, для відповідних основ a, проходять даний тест.

Назвемо число $n = 2^s t + 1$, де $s \geq 1,\ t$ – непарне, сильним псевдопростим за основою $a \neq 1\ mod\ n$, $(a,n) = 1$, якщо виконується одна з двох умов: $a^t = \pm 1\ mod\ n$, або в послідовності $a^{2t}, \dots, a^{2^{s-1}t}$ існує число, порівнянне з –1 за модулем n.

Виявляється, можна показати, що для будь-якого a, $(a,n) = 1$, існує нескінченно багато сильних псевдопростих чисел n за основою a.

Приклади: a = 7, n = 25; a = 5, n = 781.

Можна довести такі основні властивості сильних псевдопростих чисел:

1) число n, сильне псевдопросте за основою a, є ейлеровим псевдопростим за тією ж основою;

2) якщо непарне складене число n є сильним псевдопростим за основою a, то загальна кількість основ, за якою це число є сильним псевдопростим, не перевищує $(n-1)/4$.

Тому можна стверджувати, що при повторенні випробувань тесту Мілера-Рабіна k раз ймовірність невідбракування складеного числа $\leq (1/4)^k$.

Крім того, виявляється, кількість повторень тесту, достатню для практичних додатків, можна обмежити величиною $2log_2^2 n$.

Простоту невеликих простих чисел можна довести, використовуючи декілька раніше вказаних основ [16].

2.7 Висновки з розділу

У результаті проведеного аналізу розроблено узагальнений протокол обміну інформацією, який забезпечує повний цикл захищеної взаємодії між двома сторонами – від автентифікації користувачів до передачі зашифрованих і автентифікованих даних. Основною метою протоколу є формування надійного каналу зв’язку, який гарантує конфіденційність, цілісність та автентичність переданої інформації. Структура протоколу побудована на поєднанні класичних криптографічних підходів і новітніх методів кватерніонного аналізу, що підвищує рівень стійкості до атак та забезпечує високий рівень ентропії ключових параметрів.

В основі протоколу лежить процедура узгодження спільного секретного ключа за допомогою алгоритму Діфі–Гелмана. Цей механізм дозволяє двом сторонам отримати ідентичне секретне значення без його прямої передачі по мережі, що унеможливлює його перехоплення зловмисниками. Криптографічна стійкість даного процесу забезпечується складністю задачі дискретного логарифмування та додатковою перевіркою простоти числа p за тестом Мілера–Рабіна. Використання цього тесту гарантує високу ймовірність коректності обраних простих чисел, що є основою безпеки всього протоколу.

Особливою складовою системи є процедура ущільнення 1024-бітного спільного секрету до 64-бітного робочого ключа K_0 за допомогою кватерніонного множення. Такий підхід забезпечує нелінійність і незворотність перетворення, що унеможливлює відновлення початкового ключа навіть при частковому розкритті його компонент. Формування ключа відбувається через послідовне множення кватерніонів, що створює високу ступінь змішування даних і рівномірний розподіл результатів у просторі можливих значень.

Подальше використання 64-бітного ключа K_0 полягає у створенні нормованого кватерніона, на основі якого будується ортогональна матриця обертання. Ця матриця застосовується як блочний шифр для початкового етапу обробки повідомлення, забезпечуючи рівномірний вплив кожного біта ключа на елементи зашифрованого блоку. Ортогональність матриці гарантує зворотність операцій шифрування та високий рівень дифузії, що підвищує стійкість до статистичних атак.

Важливим елементом системи є використання міжнародного стандарту ISO/IEC 18031 для генерації випадкових чисел. Застосування цього стандарту гарантує високу ентропію та непередбачуваність ключових параметрів, що значно підвищує безпечність криптографічних операцій. Використання як фізичних, так і детермінованих генераторів забезпечує адаптивність системи до різних апаратних і програмних середовищ.

Таким чином, розроблений протокол обміну інформацією є комплексним і стійким рішенням, яке поєднує математичну точність, алгоритмічну ефективність і криптографічну безпеку. Його структура дозволяє забезпечити автентифікацію, конфіденційність та цілісність переданих даних, а поєднання класичних та кватерніонних методів створює основу для подальшого розвитку сучасних систем захищеного інформаційного обміну.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. MTProto Mobile Protocol. URL: https://core.telegram.org/mtproto (accessed: 08.09.2025).

2. Everything You Need to Know about WhatsApp End-to-End Encryption. URL: https://www.gupshup.ai/resources/blog/whatsapp-end-to-end-encryption/#:~:text=WhatsApp%20employs%20the%20Signal%20Protocol,to%20ensure%20confidentiality%20and%20authenticity (accessed: 12.09.2025).

3. The X3DH Key Agreement Protocol. URL: https://signal.org/docs/specifications/x3dh/ (accessed: 15.09.2025).

4. Viber Encryption Overview. URL: https://www.viber.com/app/uploads/viber-encryption-overview.pdf (accessed: 19.09.2025).

5. Messenger End-to-End Encryption Overview. URL: https://engineering.fb.com/wp-content/uploads/2023/12/MessengerEnd-to-EndEncryptionOverview_12-6-2023.pdf (accessed: 23.09.2025).

6. Key Distribution in Cryptography. URL: https://awjunaid.com/cryptography/key-distribution-in-cryptography/ (accessed: 25.09.2025).

7. Diffie-Hellman Key Exchange Algorithm. URL: https://www.1kosmos.com/security-glossary/diffie-hellman-key-exchange-algorithm/ (accessed: 27.09.2025).

8. Diffie-Hellman Key Exchange (DHKE) Algorithm. URL: https://dev.to/techlabma/diffie-hellman-key-exchange-dhke-algorithm-505b (accessed: 30.09.2025).

9. Conrad K. Quaternion Algebras. URL: https://kconrad.math.uconn.edu/blurbs/ringtheory/quaternionalg.pdf (accessed: 02.10.2025).

10. Dzwonkowski M., Rykaczewski R. Quaternion Encryption Method for Image and Video Transmission. Article. September 2013.

11. What Is A Message Authentication Code? URL: https://www.fortinet.com/uk/resources/cyberglossary/message-authentication-code (accessed: 05.10.2025).

12. ISO/IEC 18031:2025. Information technology – Security techniques – Random bit generation.

13. Viro O. Lecture 5. Quaternions. URL: https://www.math.stonybrook.edu/~oleg/courses/mat150-spr16/lecture-5.pdf (accessed: 09.10.2025).

14. Кватерніони. URL: https://stud.com.ua/156188/informatika/kvaternioni (accessed: 12.10.2025).

15. Rotation Matrices and Orthogonal Matrices. URL: https://math.libretexts.org/Bookshelves/Differential_Equations/Applied_Linear_Algebra_and_Differential_Equations_(Chasnov)/02%3A_II._Linear_Algebra/01%3A_Matrices/1.04%3A_Rotation_Matrices_and_Orthogonal_Matrices (accessed: 17.10.2025).

16. Тест Рабіна-Міллера і сильні псевдопрості числа. URL: https://studfile.net/preview/5367441/page:7/ (accessed: 20.10.2025).