КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА
Факультет комп'ютерних наук та кібернетики
Кафедра інтелектуальних програмних систем

Кваліфікаційна робота

на здобуття освітнього рівня бакалавра
за спеціальністю 121 Інженерія програмного забезпечення
на тему:

ВПРОВАДЖЕННЯ АЛГОРИТМІВ ЦИФРОВОГО ПІДПИСУ В СИСТЕМІ ЕЛЕКТРОННОГО ГОЛОСУВАННЯ НА БАЗІ ТЕХНОЛОГІЇ БЛОКЧЕЙН

Виконав студент 4-го курсу
Руслан ВОЛЧЕЦЬКИЙ _____
(підпис)

Науковий керівник:
доцент, кандидат фіз.-мат. наук
Максим ВЕРЕС _____
(підпис)

Засвідчую, що в цій роботі немає запозичень з праць інших авторів без відповідних посилань.

Студент _____
(підпис)

Роботу розглянуто й допущено до захисту на засіданні кафедри інтелектуальних програмних систем
« 29 » травня 2023 р.,
протокол № 11
Завідувач кафедри
Олександр ПРОВОТАР _____
(підпис)

Київ – 2023

РЕФЕРАТ

Обсяг роботи 45 сторінок, 6 ілюстрацій, 1 таблиця, 30 джерел посилань.

АЛГОРИТМ ЦИФРОВОГО ПІДПИСУ, БЕЗПЕКА ЦИФРОВИХ ПІДПИСІВ, БЛОКЧЕЙН, ЕЛІПТИЧНА КРИВА, ІНФРАСТРУКТУРА ВІДКРИТИХ КЛЮЧІВ, КІЛЬЦЕВИЙ ПІДПИС, ЦИФРОВИЙ ПІДПИС.

Об’єктом роботи є компонент системи анонімного голосування, який відповідає за генерацію та верифікацію цифрових підписів для транзакцій та блоків.

Метою роботи є ознайомлення з сучасними алгоритмами цифрових підписів, їх безпекою та ефективністю, аналіз вимог та використання цифрових підписів в системі анонімного голосування, технічна реалізація програмного модуля для генерації та верифікації цифрових підписів для транзакцій та блоків у системі анонімного голосування.

Методи та інструменти розроблення: мова програмування Go, інтегроване середовище розробки InteliJ IDEA, API-платформа Postman, платформа для управління контейнерами Docker, вебсервіс для спільної розробки програмного забезпечення GitHub.

Результати роботи: розглянуто найвідоміші та сучасні алгоритми цифрового підпису, їх безпека, ефективність та вразливості, проаналізовано переваги та недоліки їх використання, проведено їх порівняння за основними параметрами, реалізовано програмний модуль для генерації та верифікації цифрових підписів у системі анонімного голосуванні на базі технології блокчейн.

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАЧЕННЯ

API – Application Programming Interface, прикладний програмний інтерфейс;

FIPS – Federal Information Processing Standards, Федеральні стандарти обробки інформації;

IDE – Integrated Development Environment, інтегроване середовище розробки;

NIST – National Institute of Standards and Technology, Національний інститут стандартів і технології;

ІВК – Інфраструктура відкритих ключів.

ВСТУП

Оцінка сучасного стану об’єкта розробки. Цифровізація кожного року все більше поширюється в різні сфери сучасного життя. Тому навіть звичайний підпис документу може стати проблемним завданням для людей. Разом з тим, все більш важливою місією стає покращення безпеки та конфіденційності інформації.

Таким чином почали своє існування багато сервісів, які надають можливості цифрових підписів своїм клієнтам. Тим не менш, така частина сучасних реалій як голосування залишається проблематичною для вирішення.

Станом на сьогодні, існують компанії та сервіси, що надають можливість електронного голосування своїм користувачам. Багато з них, окрім безпеки, також стверджують про високий рівень анонімності. Але, на жаль, деякі з них викривались в недостатній конфіденційності, а переважна більшість не використовують найсучасніших методів для досягнення такої цілі.

Також за декілька останніх років з’явились нові досягнення в цих сферах та теоретичні чи концептуальні схеми в забезпеченні високого рівня анонімності.

Актуальність роботи та підстави для її виконання. Цифрові підписи дуже активно вивчаються, як з точки зору вразливостей, так і створення більш безпечних та ефективних алгоритмів. Хоч і досягнуто сильного рівня безпеки в 128-біт, але потенційні загрози від нових видів атак все ще існують.

Також дуже мала кількість сучасних застосунків використовують найсучасніші алгоритми цифрових підписів, які окрім безпеки забезпечують хорошу ефективність, а деякі особливі види підписів, як кільцевий підпис, надають додаткових можливостей, як наприклад високого рівня анонімності. Це і стало підставою до виконання цієї роботи.

Мета й завдання роботи. Метою кваліфікаційної роботи є технічна реалізація програмного модуля для генерації та верифікації цифрових підписів для транзакцій та блоків у системі анонімного голосування. Для досягнення цієї мети поставлено такі завдання.

- Дослідити існуючі алгоритми цифрових підписів.

- Провести порівняння існуючих алгоритмів цифрових підписів за основними параметрами.
- Провести аналіз вимог до цифрових підписів у системі анонімного голосування.
- Розробити програмний модуль для генерації та верифікації цифрових підписів.

Об'єкт, методи й засоби розроблення. Об’єктом розроблення є компонент системи анонімного голосування, який відповідає за генерацію та верифікацію цифрових підписів для транзакцій та блоків.

Розробці програмного засобу передував аналіз сучасних алгоритмів, вивчення їх теоретичної складової, порівняння проаналізованих алгоритмів за безпекою та ефективністю.

В якості інструменту створення програмного засобу було обрано IntelliJ IDEA – сучасна та багатофункціональна IDE, мовою програмування стала Go, компільована мова з відкритим вихідним кодом, що надає гарний інструментарій для створення паралельного та масштабованого програмного забезпечення.

Також для запуску і тестування системи було використано API-платформу Postman та платформу для управління контейнерами Docker.

Можливі сфери застосування. Цей програмний модуль може застосовуватись у будь-яких системах, де необхідно генерувати та верифікувати цифрові підписи, а також що потребують підписів з забезпеченням анонімності, зокрема у анонімній системі електронного голосування.

РОЗДІЛ 1 ОСНОВИ ЦИФРОВИХ ПІДПИСІВ

1.1 Цифровий підпис

У цифрову епоху, коли електронне спілкування та транзакції стали звичним явищем, забезпечення цілісності та автентичності цифрових документів зайняло одне з провідних значень сьогодення. Цифровий підпис – це криптографічний метод, який слугує в якості еквіваленту звичайного власноручного підпису, забезпечуючи при цьому засіб перевірки автентичності та цілісності електронних повідомлень, документів чи будь-яких інших даних.

Цифрові підписи працюють як відбитки пальців. Вони зберігаються у вигляді закодованого повідомлення та надійно пов’язують підписувача з відповідним документом у записаній транзакції. Цифрові підписи покладаються на загальноприйнятий формат – інфраструктуру відкритих ключів, або ІВК, для забезпечення посиленої безпеки, про яку ми ще згадаємо детальніше. Вони є частиною технології електронного підпису [1].

Електронний підпис – це ширше поняття, яке охоплює будь-яке електронне підтвердження згоди чи схвалення особою змісту документа, транзакції чи інших даних. Ним може бути як звичайне відскановане зображення власноручного підпису, так і надруковане ім'я наприкінці документа чи значок у полі. Зазвичай електронні підписи набагато простіше впровадити та використовувати порівняно з цифровими підписами, оскільки останні вимагають набагато більшого рівня криптографічного захисту [2].

По суті, цифровий підпис – це математичний алгоритм, який застосовується до цифрового документа з використанням приватного (закритого) ключа, який надійно зберігається у самого підписувача. Після завершення роботи алгоритму утворюється унікальний цифровий підпис, який є специфічним для документа та підписувача. Підпис прикріплюється до документа і може бути перевірений будь-ким, хто має публічний (відкритий) ключ підписувача (рисунок 1.1) [3].

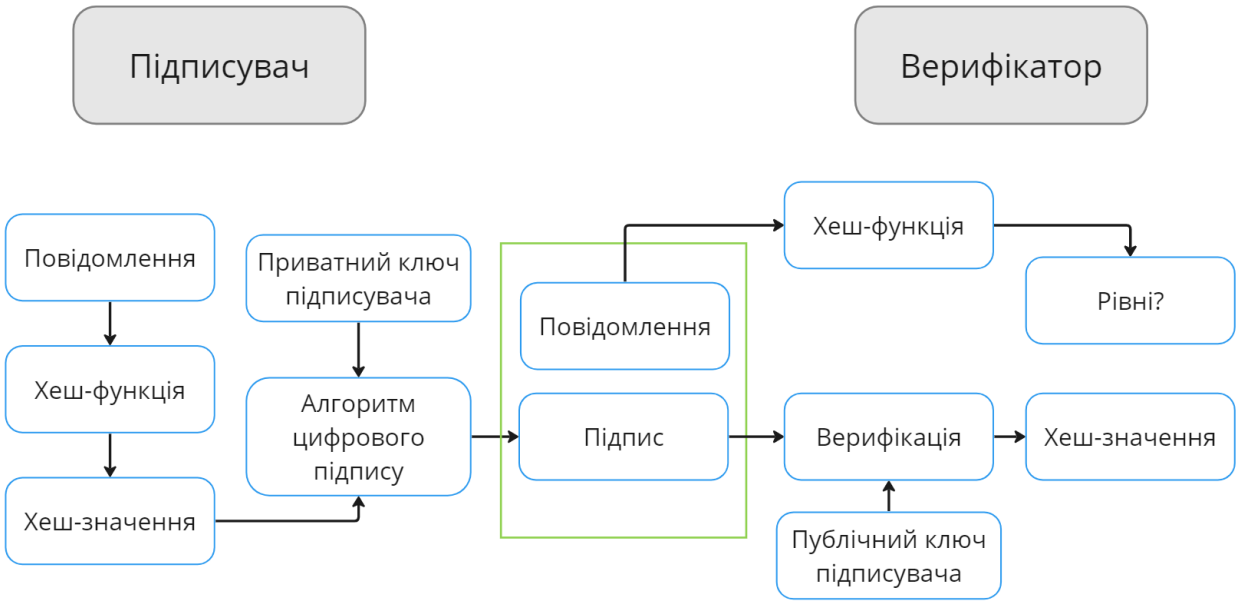

Рисунок 1.1 – Схематичний процес підпису та верифікації [4]

Процес генерації та верифікації цифрового підпису складається з кількох етапів. Спочатку приватним ключем підписувача генерується хеш-значення – унікальний рядок символів фіксованої довжини, який представляє вміст документа. Після цього хеш-значення шифрується за допомогою цього ж приватного ключа підписувача, в результаті чого отримується цифровий підпис. Тепер документ разом з цифровим підписом можна відправляти адресату.

Щоб перевірити справжність отриманого цифрового підпису, одержувач виконує декілька дій. Спочатку, використовуючи публічний ключ підписувача, цифровий підпис розшифровується та отримується оригінальне хеш-значення. Далі одержувач самостійно обчислює хеш-значення отриманого документа. Якщо розраховане та розшифроване хеш-значення підпису збігаються, тоді він вважається дійсним, що свідчить про те, що документ чи повідомлення не було підроблене і дійсно підписане заявленим підписувачем.

1.2 Переваги та недоліки цифрових підписів

Цифрові підписи мають низку переваг, які роблять їх цінним інструментом у сфері електронного спілкування та транзакцій. Розберемо деякі з них.

- Автентифікація та цілісність: цифрові підписи забезпечують надійний механізм для перевірки особи підписанта та забезпечення цілісності підписаного документа. Використання криптографічних алгоритмів робить надзвичайно складним для будь-кого залишити підробку документа непоміченою.
- Невідворотність: завдяки прив'язці цифрового підпису до особистого ключа підписувача, цифрові підписи є вагомим доказом причетності та намірів підписувача. Ця функція не дозволяє підписувачам згодом заперечувати свою участь у транзакції, що підвищує підзвітність і довіру [5].
- Ефективність та економія коштів: цифрові підписи спрощують і автоматизують процес підписання, усуваючи необхідність у фізичних підписах, друкуванні, скануванні та пересиланні документів. Це призводить до значної економії часу та коштів, що робить цифрові підписи привабливим рішенням для бізнесу.
- Безпека: цифрові підписи покладаються на передові алгоритми шифрування, що забезпечують конфіденційність підписаного документа під час передачі. Використання інфраструктури відкритих ключів надає додаткового рівня безпеки, забезпечуючи надійну основу для управління ключами та сертифікатами.

Незважаючи на свої численні переваги, цифрові підписи також мають певні проблеми та виклики.

- Управління ключами: безпечне зберігання приватних ключів та управління ними мають вирішальне значення для збереження цілісності цифрових підписів. Втрата, крадіжка або компрометація приватних ключів може підірвати довіру до системи цифрового підпису.

- Правові та регуляторні міркування: різні країни та юрисдикції можуть мати різну правову базу щодо визнання та примусового виконання цифрових підписів. Організації повинні розуміти та дотримуватися відповідних законів і нормативних актів, щоб забезпечити дійсність цифрових підписів у відповідних контекстах.
- Прийняття користувачами: широке впровадження цифрових підписів вимагає від користувачів прийняття та знайомства з технологією. Для подолання опору змінам і сприяння використанню цифрових підписів у різних галузях необхідні освітні та просвітницькі ініціативи.
- Розвиток технологій: з розвитком технологій з'являються нові криптографічні алгоритми та стандарти. Важливо бути в курсі останніх досягнень і забезпечувати сумісність зі стандартами цифрового підпису, що розвиваються, щоб підтримувати довгострокову цілісність та інтероперабельність підписаних документів.

1.3 Вразливості цифрових підписів

Хоча цифрові підписи забезпечують надійні гарантії безпеки та припускаючи некомпрометацію приватного ключа, все ж з роками було виявлено деякі вразливості.

Якщо алгоритм, який використовується для генерації ключів, є недосконалим або якщо обрано слабкі параметри, це може призвести до вразливостей у схемі цифрового підпису. Слабкі ключі можуть бути чутливими до атак грубої сили або методів факторизації, що дозволяє зловмиснику підробляти цифрові підписи.

В алгоритмах цифрового підпису, заснованих на хеш-функціях, таких як RSA або DSA, атаки на колізії можуть бути використані для створення фальшивих цифрових підписів. Колізія виникає, коли два різних входи дають однакове хеш-значення. У 2005 році було доведено, що алгоритм SHA-1 вразливий до атак на

зіткнення, що призвело до його відмови від використання в багатьох додатках, чутливих до безпеки [6].

Атаки побічних каналів спрямовані на фізичну реалізацію системи цифрового підпису, а не на математичні алгоритми, що лежать в її основі. Аналізуючи певні побічні ефекти обчислень, такі як споживання енергії або інформація про час чи кеш, зловмисник може отримати конфіденційну інформацію і скомпрометувати приватний ключ.

Криптографічні алгоритми, що використовуються для цифрових підписів, можуть містити вразливості, які виявляються з часом. Наприклад, якщо в алгоритмі, якому раніше довіряли, виявляються вразливості або якщо розвиток обчислювальних потужностей робить певні атаки можливими, безпека цифрових підписів, які використовують ці алгоритми, може бути порушена.

Можна побачити, що більшість з вразливостей частіше пов’язані з окремими компонентами, що застосовуються цифровим підписом, наприклад хеш-функціями, або ж з самою системою. Тому щоб запобігти цим вразливостям, важливо застосовувати надійні методи управління ключами, регулярно оновлювати криптографічні алгоритми та протоколи, використовувати безпечні методи впровадження та забезпечувати цілісність всієї інфраструктури, де застосовується цифровий підпис. Для усунення інших потенційних вразливостей і підвищення безпеки цифрових підписів важливо бути в курсі останніх досліджень і рекомендацій у сфері безпеки.

1.4 Інфраструктура відкритих ключів

Інфраструктура відкритих ключів (ІВК) – це фундаментальна технологія для встановлення безпечного зв'язку та забезпечення автентичності й цілісності цифрових транзакцій. Вона забезпечує надійну основу для управління цифровими сертифікатами, криптографічними ключами та пов'язаними з ними процесами. ІВК відіграє життєво важливу роль у різних сферах застосування, включаючи

безпечний перегляд веб-сторінок, електронну комерцію, шифрування електронної пошти, віртуальні приватні мережі та безпечне підписання документів. Створення ефективної ІВК вимагає дотримання багатьох принципів, підтримку важливих процесів та реалізацію якісних і безпечних сервісів роботи (рисунок 1.2) [7][8].

Рисунок 1.2 – Компоненти ефективної інфраструктури відкритих ключів

Основний принцип ІВК полягає у використанні асиметричного шифрування, де застосовуються два математично пов'язані ключі: відкритий і закритий. Відкритий ключ є загальнодоступним і використовується для шифрування повідомлень, тоді як закритий ключ є конфіденційним і використовується для розшифрування зашифрованих даних. ІВК забезпечує безпечне розповсюдження та перевірку цих ключів за допомогою цифрових сертифікатів, які видаються довіреними центрами сертифікації.

Центри сертифікації відповідають за перевірку ідентичності суб'єктів (наприклад, фізичних осіб, організацій або пристроїв) та видачу цифрових сертифікатів, які прив'язують їхні відкриті ключі до їхніх ідентифікаційних даних.

Ці центри підписують сертифікати за допомогою власних закритих ключів, таким чином встановлюючи довіру до автентичності сертифікатів. Ця ієрархічна модель довіри гарантує, що суб'єкти можуть перевіряти цілісність і легітимність відкритих ключів і сертифікатів, запобігаючи несанкціонованому доступу та підробці.

Стандартні системи ІВК складаються з декількох ключових компонентів та структур.

- Центри сертифікації: довірені організації, які видають і підписують цифрові сертифікати. Вони відіграють центральну роль у створенні та підтримці ієрархії довіри в рамках ІВК.
- Реєстраційні органи: організації, відповідальні за перевірку особи того, хто хоче отримати цифровий сертифікат. Реєстраційні органи діють як посередники між суб'єктами та центрами сертифікації.
- Сховища сертифікатів: централізовані або розподілені бази даних, які зберігають і поширюють видані цифрові сертифікати. Вони дозволяють користувачам отримувати та перевіряти сертифікати за потреби.
- Списки відкликання сертифікатів: списки, які ведуться центрами сертифікації і містять інформацію про відкликані або прострочені сертифікати. Ці списки дозволяють користувачам перевіряти дійсність сертифікатів, перш ніж довіряти їм.

Загалом, ІВК є основою, яка уможливлює використання деяких технологій, наприклад цифровий підпис і шифрування, великими групами користувачів, забезпечуючи підґрунтя для встановлення довіри та захисту комунікації [9].

1.5 Застосування цифрових підписів

Цифрові підписи можуть та уже відіграють важливу роль у різних сферах, включаючи кібербезпеку, фінанси, юридичні процеси та документообіг.

Цифрові підписи все частіше використовуються урядами. Урядові установи можуть набагато швидше впроваджувати нові законопроєкти та податкові

декларації, управляти контрактами, керувати посвідченнями особи та виконувати багато інших завдань. Урядові організації оперують великою кількістю конфіденційної інформації, а посилена безпека цифрових підписів зменшує ймовірність її потрапляння до чужих рук. Це прискорює весь процес і зменшує ризик небажаного витоку інформації.

У сфері охорони здоров’я та лікування цифрові підписи можуть значно покращити адміністративний процес. Вони спрощують прийом пацієнтів і скорочують час обслуговування в лікарнях і клініках. Також вони прискорюють процеси у сфері виробництва, де це дозволяє покращити розробку продукції та збільшити обсяги виробництва і продажів.

Також однією з найбільш розповсюджених сфер для застосування цифрових підписів є банкінг та фінанси, де вони роблять можливим безпаперовий банкінг, легкі кредити, контракти, а також є невід’ємною частиною роботи криптовалют та автентифікації на блокчейні [10].

Спробуємо також розглянути додаткові можливості, які надає використання цифрових підписів.

Як уже згадувалось вище, цифрові підписи широко використовуються для автентифікації документів як в особистому, так і в професійному середовищі. Додаючи цифровий підпис до документа, підписувач надає унікальний ідентифікатор, який може бути перевірений одержувачем або будь-якою зацікавленою стороною. Цей процес перевірки гарантує, що документ не був підроблений з моменту його підписання, і підтверджує особу підписувача, а також унеможливлює заперечення своєї причетності до підписання документа. Ця властивість має вирішальне значення в юридичних і договірних сценаріях, де автентичність підписів є надзвичайно важливою для забезпечення виконання зобов'язань.

У сфері електронної комерції та онлайн-транзакцій цифрові підписи використовуються для забезпечення безпечної та надійної взаємодії. Вони допомагають встановити довіру між сторонами, що беруть участь у транзакції,

перевіряючи цілісність електронних повідомлень, замовлень на купівлю та інших документів. Використовуючи цифрові підписи, компанії та приватні особи можуть мінімізувати ризик шахрайства та забезпечити більш безпечне онлайн-середовище.

Цифрові підписи також цінні для довгострокового зберігання документів. Підписуючи документ цифровим підписом, підписувач створює запис із позначкою часу, який можна перевірити навіть через роки. Ця можливість може мати важливе значення для таких галузей, як архівна справа, де цілісність та автентичність історичних документів потрібно зберігати протягом тривалого часу.

Зважаючи на зростаючу тенденцію до віддаленої роботи та віртуальної співпраці, цифрові підписи сприяють ефективному та безпечному документообігу. Вони дозволяють декільком сторонам підписувати і затверджувати документи віддалено, усуваючи необхідність фізичної присутності або клопоти, пов'язані з друком, підписанням, скануванням і пересиланням документів. Цифрові підписи спрощують такі процеси, заощаджують час і забезпечують безперешкодну віддалену співпрацю.

Якщо ми підписуємо документ офлайн і передаємо його комусь фізично, буде важко відстежити походження документа. Якщо в якийсь момент виникнуть перешкоди, відстеження буде ускладнене. Цифрові підписи тут дуже корисні, оскільки вони дозволяють відстежити походження документа і простежити передачу документів з однієї системи в іншу. Також це сприяє зменшенню використання паперу, адже ми можемо шифрувати документи і зберігати їх в Інтернеті. Нам не потрібно перевозити паперові документи, що призводить до зниження витрат на транспортування і пов'язаних з цим питань безпеки [11].

Загалом, цифрові підписи знаходять застосування в різних сферах, включаючи електронні контракти, фінансові транзакції, безпечне спілкування електронною поштою, розповсюдження програмного забезпечення та державну документацію. Вони пропонують безпечний і надійний метод забезпечення автентичності, цілісності та неспростування цифрової інформації, зміцнюючи довіру та впевненість у цифровій сфері.

РОЗДІЛ 2 АЛГОРИТМИ ЦИФРОВИХ ПІДПИСІВ

Спробуємо навести найвідоміші алгоритми цифрових підписів, розібратись з основними принципами їх роботи, перевагами та недоліками, а також певними особливостями.

2.1 RSA алгоритм

RSA (Rivest-Shamir-Adleman) – широко використовуваний алгоритм шифрування з відкритим ключем, названий на честь його винахідників: Рональда Рівеста, Аді Шаміра та Леонарда Адлемана. Це асиметричний алгоритм шифрування, тобто він використовує пару ключів, відкритий і закритий, для шифрування і розшифрування. RSA відомий своїм відносно високим ступенем захисту і широко використовується для безпечного спілкування, цифрових підписів та обміну ключами.

Алгоритм RSA ґрунтується на математичних властивостях великих простих чисел і модульній арифметиці. Давайте розберемося в деталях RSA та його роботи [12][13].

Розглянемо процес генерації ключа.

- Вибирається два різних простих числа, p і q.
- Обчислюється їх добуток:

$$n = p \cdot q.$$

  Це значення, n, використовується як модуль як для публічного, так і для приватного ключів.

- Обчислюється функція Ейлера від n:

$$\varphi(n) = (p - 1) \cdot (q - 1).$$

  Ця функція являє собою кількість натуральних чисел, менших за n, які є взаємно простими з n.

– Обирається публічний показник степеня, e (публічна експонента), який, як правило, є невеликим простим числом і взаємно простим з φ(n), тобто:

$$\text{НСД}(e, \varphi(n)) = 1.$$

– Обчислюється приватний показник степеня, d (приватна експонента), який є оберненим до e за модулем φ(n), тобто:

$$d \equiv e^{-1} \pmod{\varphi(n)}.$$

Це можна зробити за допомогою розширеного алгоритму Евкліда.

В результаті цього відкритий ключ буде складатись з пари (e, n), а закритий ключ з пари (d, n). В сучасних застосуваннях RSA частіше використовують функцію Кармайкла (λ(n)) замість функції Ейлера. Це можливо тому, що значення λ(n) ділиться націло на φ(n). Функція Кармайкла є більш доцільною, тому що будь-яка можлива пара ключів RSA системи, заснованої на функції Ейлера, є можливою парою ключів для функції Кармайкла, тоді як зворотне не є загальним правилом, а також обчислення d за модулем φ(n) іноді дає результат, який є більшим, ніж потрібно [14].

Розглянемо процес шифрування повідомлення.

– Для шифрування повідомлення, представленого у вигляді цілого числа m, відправник використовує відкритий ключ одержувача (e, n).
– Відправник обчислює зашифрований текст, c, за формулою:

$$c \equiv m^e \pmod{n}.$$

Отриманий зашифрований текст c надсилається одержувачу.

Тепер розглянемо процес розшифрування повідомлення.

– Одержувач використовує свій закритий ключ (d, n) для розшифрування зашифрованого тексту c.
– Одержувач обчислює оригінальне повідомлення m за наступною формулою:

$$m = c^d \pmod{n}.$$

Отримане розшифроване повідомлення m і є вихідним відкритим текстом.

Безпека RSA базується на складності розкладання великих складених чисел на прості множники. Маючи відкритий ключ (e, n), обчислювально дуже складно визначити закритий ключ (d, n), не знаючи простих множників n. Також безпека залежить від вибору великих простих чисел p і q, щоб зробити факторизацію модуля n надзвичайно складною.

Шифрування RSA підходить лише для безпосереднього шифрування відносно коротких повідомлень, оскільки відкритий і зашифрований текст обмежені розміром за модулем n. Для шифрування довших повідомлень часто використовується гібридна схема шифрування, де RSA використовується для шифрування симетричного ключа, а симетричний ключ потім використовується для шифрування власне повідомлення.

RSA може використовуватися для цифрових підписів, коли відправник підписує повідомлення за допомогою свого приватного ключа, а одержувач перевіряє підпис за допомогою відкритого ключа відправника. Розглянемо конкретніший приклад роботи такого цифрового підпису.

Нехай Аліса хоче підписати та відправити повідомлення Бобу. Вона обчислює хеш-значення повідомлення, далі, використовуючи свій приватний ключ, підносить його до степеня d за модулем n (таким же чином, як при розшифровці повідомлення) і отримує цифровий підпис до повідомлення. Боб, отримавши підписане повідомлення та використовуючи відкритий ключ Аліси, а також той самий алгоритм хешування, підносить значення підпису до степеня e за модулем n. Після цього Боб порівнює отримане хеш-значення з хеш-значенням повідомлення, якщо ж вони збігаються, то підпис вважається дійсним і повідомлення прийшло від Аліси, адже тільки вона володіє власним приватним ключем.

2.2 DSA алгоритм

DSA (Digital Signature Algorithm) – це широко використовуваний криптографічний алгоритм, який надає безпечний метод генерації та перевірки цифрових підписів. Він був розроблений Національним інститутом стандартів і технологій (NIST) у США і базується на математичних принципах піднесення до степеня за модулем та задачі дискретного логарифмування. Розглянемо детальніше кроки для створення та перевірки цифрового підпису за цим алгоритмом [15].

Генерація ключів відбувається в два етапи.

Першим етапом є генерація параметрів.

– Обирається криптографічна хеш-функція H з вихідною довжиною |H| біт, число N, яке використовується коли довжина вихідного значення |H| більша ніж N, а також довжина ключа L. N обирається таким, що воно задовольняє умовам:

$$N < L,$$
$$N \leq |H|,$$

а L повинен бути кратним числу 64. За стандартом FIPS 186-4 значення L та N обираються серед таких пар: (1024, 60), (2048, 224), (2048, 256), (3072, 256). Більша довжина ключа призводить до довших обрахунків та одночасно до збільшення рівня безпеки.

– Обирається просте N-бітове число q і просте L-бітове число p таким чином, щоб значення (p - 1) було кратним q.

– Обирається ціле число h з діапазону (1, p - 1) не включаючи кінці.

– Обчислюється значення g:

$$g = h^{\frac{p-1}{q}} \pmod{p}.$$

Якщо g дорівнює одиниці, потрібно змінити число h, поки не отримаємо результат відмінний від одиниці.

В результаті першого етапу p, q і g – це параметри алгоритму, які є спільними для всіх користувачів.

Другий етап – створення самих ключів.

– Генерується випадкове число x, де

$$0 < x < q.$$

Воно буде закритим ключем.

– Обчислюється відповідний відкритий ключ, y, за формулою:

$$y = g^x \pmod{p}.$$

Таким чином ми отримуємо x – приватний ключ, та y – публічний ключ.

Розглянемо процес створення підпису.

– Генерується випадкове число k, аналогічно до числа x в другому етапі.

– Обчислюється значення r за формулою:

$$r = (g^k \pmod{p}) \pmod{q}.$$

– Обчислюється значення s за формулою:

$$s = k^{-1} \cdot (H(m) + x \cdot r) \pmod{q},$$

де H(m) – хеш-значення повідомлення m за допомогою попередньо обраної криптографічної хеш-функції H.

Отримана пара (r, s) формує цифровий підпис повідомлення m.

Розглянемо процес перевірки цифрового підпису (r, s) повідомлення m з відомим публічним ключем підписувача y.

– Обчислюється хеш-значення отриманого повідомлення, H(m).

– Обчислюється значення w за формулою:

$$w = s^{-1} \pmod{q}.$$

– Обчислюється значення u_1 та u_2 за формулами:

$$u_1 = H(m) \cdot w \pmod{q},$$
$$u_2 = r \cdot w \pmod{q}.$$

– Обчислюється v за формулою:

$$v = (g^{u_1} \cdot y^{u_2} \pmod{p}) \pmod{q}.$$

Якщо v дорівнює r, то підпис є дійсним, в іншому випадку він вважається недійсним.

З доведенням коректності даного алгоритму можна ознайомитись в відповідному розділі офіційного документу від NIST [16].

Серед переваг цього алгоритму є висока надійність з точки зору безпеки та стабільності порівняно з попередньо описаним RSA алгоритмом, генерація ключів відбувається набагато швидше. Також DSA вимагає менше місця для зберігання даних для роботи всього циклу.

У DSA випадкові значення елементу ключа підпису k є критично важливими, оскільки порушення правил генерації може розкрити всю конфіденційність зловмисникам. Іноді використання одного й того ж значення для цього елемента ключа підпису або витік його декількох бітів може бути достатньо для зловмисників, щоб розкрити приватний ключ x. Цій атаці можна запобігти лише якщо для кожного нового значення підпису обчислювати нове випадкове значення k або обраховувати його детерміновано з приватного ключа та хешу повідомлення. Таким чином, необхідно, щоб значення k було різним для кожного значення H(m) і непередбачуваним, щоб зберегти приватний ключ x в таємниці від зловмисників [17].

DSA в основному використовується для цифрових підписів і не забезпечує конфіденційності. У багатьох практичних сценаріях потрібне поєднання цифрових підписів і шифрування. Зазвичай це досягається за допомогою гібридних криптосистем, де DSA використовується для підписів, а симетричні або асиметричні алгоритми шифрування (такі як RSA) використовуються для забезпечення конфіденційності.

2.3 ECDSA алгоритм

ECDSA (Elliptic Curve Digital Signature Algorithm) – це широко використовуваний криптографічний алгоритм, який забезпечує функціональність цифрового підпису на основі криптографії еліптичних кривих. Це варіант алгоритму DSA, який пропонує менший розмір ключа та підвищену ефективність.

Криптографія еліптичних кривих служить основою для ECDSA. Вона включає в себе математичні операції над еліптичними кривими, визначеними над кінцевими полями. Рівняння еліптичної кривої в алгоритмі ECDSA має вигляд

$$y^2 = x^3 + a \cdot x + b,$$

де a і b є константами і утворюють параметри кривої (рисунок 2.1). Вибір цих констант має вирішальне значення для безпеки та продуктивності.

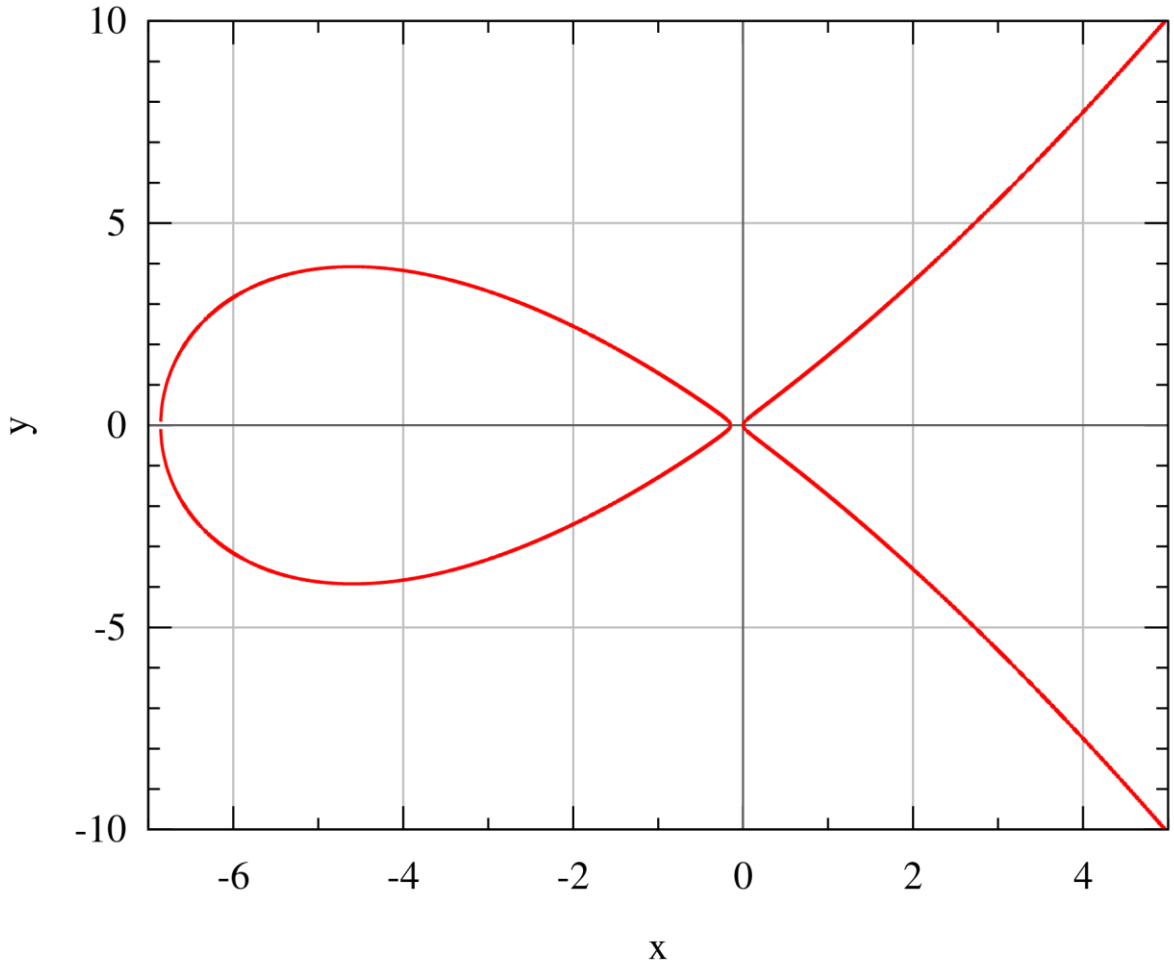

Рисунок 2.1 – Вигляд еліптичної кривої на прикладі Curve25519

Окрім самої кривої та простого порядку скінченного поля p, над яким виконуються всі обрахунки, важливими параметрами є точка генератор G та її мультиплікативний порядок n, який до того ж повинен бути простим числом. Вибір всіх цих параметрів разом з кривою є складним процесом та існує декілька можливих алгоритмів їх підбору. Всі ці деталі чудово розкриваються у іншій роботі з більш детальним описом одного з варіантів ECDSA, а ми надалі припустимо, що всі параметри відповідають необхідним умовам [18].

Першим кроком в ECDSA є генерація ключів. Користувач генерує приватний ключ d як випадкове число в певному діапазоні. Потім відповідний відкритий ключ D отримується шляхом множення точки генератора G еліптичної кривої на приватний ключ, тобто D також є точкою на кривій [19][20].

Розглянемо процес генерації підпису.

- Спочатку генерується випадкове число k у діапазоні (0, n) не включаючи кінці.
- Використовуючи це випадкове число k, обчислюється точка на еліптичній кривій:

$$K = k \cdot G,$$

з відповідними координатами x та y.

- Координата x згенерованої точки K зменшується за модулем порядку n точки-генератора:

$$r = x \pmod{n}.$$

- Обчислюється обернене число до k:

$$k_{inv} = k^{-1} \pmod{n}.$$

- Підпис s обчислюється таким чином:

$$s = k_{inv} \cdot (H(m) + r \cdot d) \pmod{n},$$

де H(m) – значення хеш-функції H від повідомлення m.

В результаті отримуємо кінцевий підпис з пари (r, s)

Тепер розглянемо процес верифікації підпису.

– Обчислюється обернена величина до s за модулем n:

$$s_{inv} = s^{-1} \pmod{n}.$$

– Обчислюється дві точки на еліптичній кривій:

$$P_1 = (H(m) \cdot s_{inv}) \cdot G,$$
$$P_2 = (r \cdot s_{inv}) \cdot D,$$

– Далі дві отримані точки P_1 і P_2 додаються, щоб отримати результуючу точку P’ з координатами x’ та y’.

Якщо виконується умова

$$r \equiv x' \pmod{n},$$

то підпис є дійсним, в іншому випадку він вважається недійсним.

Перераховуючи переваги алгоритму ECDSA необхідно згадати про високий рівень безпеки. Це особливо ефективне рішення, засноване на криптографії з відкритим ключем, що слугує основою безпеки біткоїна і широко застосовується в додатках для безпечного обміну повідомленнями. З погляду на ряд причин, менші ключі є кращими ніж більші ключі. Оскільки математика простіша з меншими ключами, самі алгоритми швидше генерують підписи. Також високою є і швидкість перевірки цифрового підпису.

ECDSA масштабується і може підтримувати різні рівні безпеки шляхом вибору відповідних параметрів кривої. Це дозволяє знайти компроміс між безпекою та обчислювальною ефективністю, виходячи з конкретних вимог програми. Така гнучкість робить алгоритм адаптованим до широкого спектру середовищ та систем.

Серед основних викликів та проблем є безпечна реалізація ECDSA, що досить складно і проблематично, особливо для звичайних кривих. Основним недоліком використання еліптичних кривих є те, що він значно збільшує розмір зашифрованого повідомлення в порівнянні з RSA-шифруванням. Крім того,

алгоритм стає складніший і важчий для реалізації, що збільшує ризик помилок при реалізації і знижує безпеку методу.

Також варто згадати про вибір числа k, який, як і в DSA алгоритмі, може стати причиною компрометації приватного ключа через вже згадані причини.

2.4 EdDSA алгоритм

EdDSA – це сучасна схема цифрового підпису, яка пропонує високі гарантії безпеки, будучи при цьому ефективною і стійкою до різних криптографічних атак. Цей алгоритм також заснований на математиці еліптичних кривих і був розроблений для усунення деяких обмежень і вразливостей, притаманних попереднім алгоритмам підпису.

Якщо в ECDSA крива представлена за допомогою рівняння Веєрштрасса, то EdDSA використовує форму еліптичних кривих Едвардса (рисунок 2.2). Форма Едвардса має певні переваги з точки зору ефективності та стійкості до певних типів атак. Сама форма визначається рівнянням

$$x^2 + y^2 = 1 + d \cdot x^2 \cdot y^2.$$

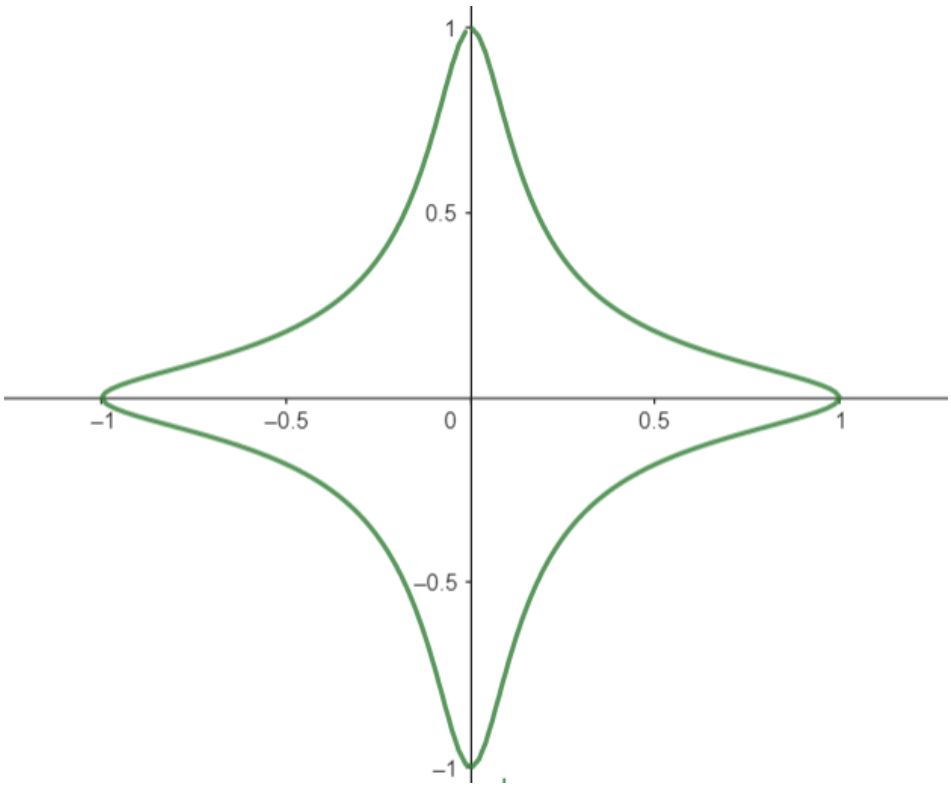

Рисунок 2.2 – Вигляд еліптичної кривої форми Едвардса

Перед початком обирається еліптична крива разом з всіма параметрами, включаючи коефіцієнти, генератор, хеш-функцію та інші. Цей процес ще складніший в порівнянні з ECDSA, тому найчастіше використовують готові параметризації кривих Ed25519 або Ed448 [21].

Розглянемо процес генерації ключів.

– Генерується випадкове число s з інтервалу (0, n), де n – порядок кривої.
– Обчислюється хеш-функція від числа s, H(s). З неї очищаються останні кілька бітів, що відповідають кофактору кривої (кофактор відомий як один з параметрів), після цього очищається найстарший біт і встановлюється попередній до нього біт. Результатом цих дій буде приватний ключ p_r.
– Обчислюється публічний ключ p_b шляхом множення приватного ключа на генератор групи G.

Всі ці перетворення гарантують, що приватний ключ завжди буде належати до однієї і тієї ж підгрупи точок на кривій, і що приватні ключі завжди будуть мати

однакову довжину (для захисту від атак побічних каналів, що базуються на синхронізації).

Розглянемо процес створення цифрового підпису.

– Генерується число r:

$$r = H(H(p_r) + m) \pmod{q},$$

де H – це обрана хеш функція;

m – повідомлення;

q – порядок генератора G.

– Використовуючи r та генератор обчислюється точка R:

$$R = r \cdot G.$$

– Обчислюється число h:

$$h = H(R + p_b + m) \pmod{q}. \quad (1)$$

– Обчислюється число s:

$$s = r + h \cdot p_r \pmod{q}.$$

В результаті підпис складається з пари (R, s).

Тепер розглянемо процес верифікації цифрового підпису.

– Обчислюється число h за формулою (1).

– Обчислюється дві точки:

$$P_1 = s \cdot G,$$
$$P_2 = R + h \cdot p_b.$$

Якщо точки P_1 та P_2 рівні, то підпис вважається дійсним.

Специфічні математичні операції, що використовуються в EdDSA, такі як додавання точок, подвоєння точок та скалярне множення, виконуються на основі

властивостей еліптичної кривої та рівняння форми Едвардса. Саме тому вони обраховуються швидше, ніж в звичайних RSA та DSA алгоритмів, та трошки швидше ніж в ECDSA (якщо брати до порівняння саме реалізації з використанням Ed25519 та Ed448).

EdDSA розроблено для захисту від різних криптографічних атак, включаючи атаки побічних каналів та колізійні атаки, чого не можна сказати про попередні алгоритми. Також він простіший в реалізації та розумінні, ніж ECDSA, та має схожий рівень безпеки при кривих з однаковими довжинами ключів [22].

2.5 Алгоритм кільцевого підпису

Кільцевий підпис – це ще один спосіб створення цифрових підписів, який дозволяє користувачеві підписувати повідомлення від імені групи, не розкриваючи свою індивідуальну ідентичність. Цей алгоритм забезпечує анонімність, неможливість відмови підписувача та цілісність повідомлення. В своїй основі він може використовувати будь-який інший алгоритм підпису з невеликими змінами, наприклад алгоритм RSA чи інші з використанням еліптичних кривих.

У схемі кільцевого підпису підписувач обирає набір відкритих ключів, відомий як кільце, який включає його власний відкритий ключ. Підписувач генерує підпис, який може перевірити лише той, хто володіє відповідними відкритими ключами, але конкретна особа підписувача залишається прихованою (рисунок 2.3).

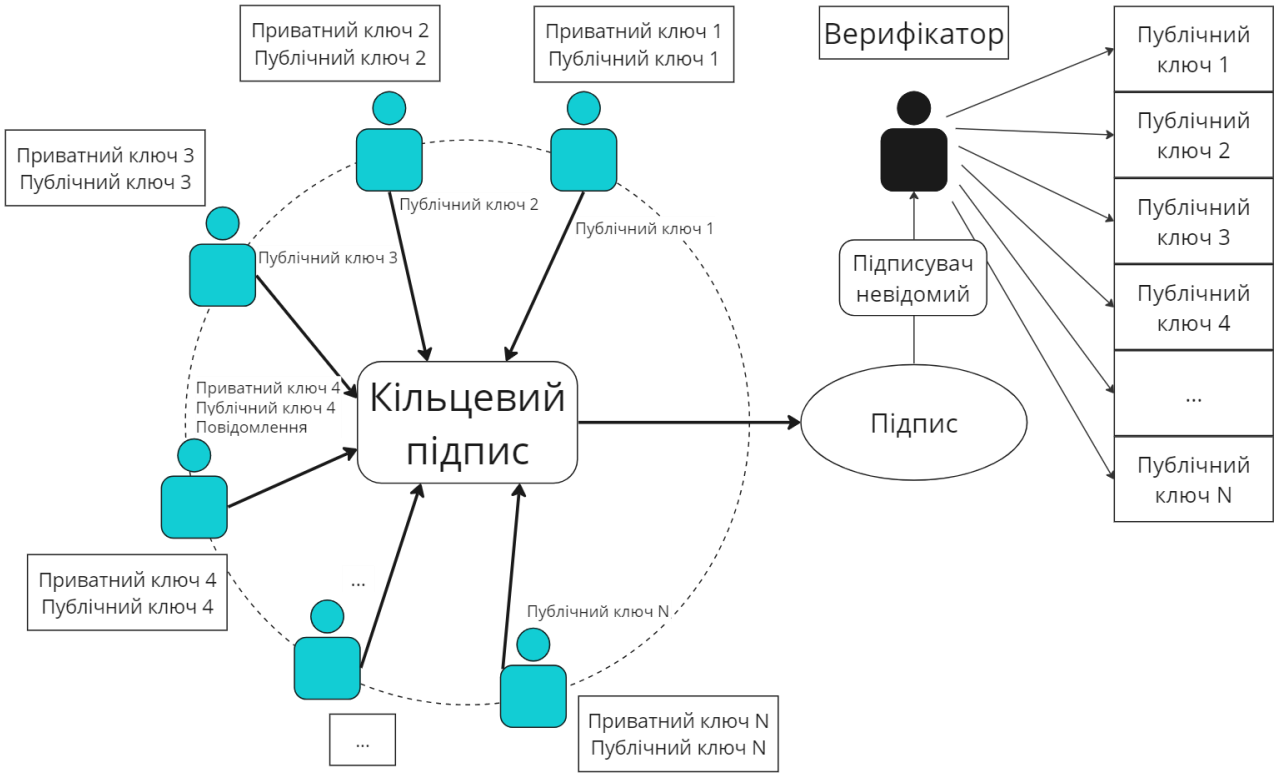

Рисунок 2.3 – Схема роботи кільцевого підпису [23]

Розглянемо один з варіантів роботи кільцевого підпису з використанням певної еліптичної кривої, параметри якої нехай нам уже відомі [24].

Спочатку відбувається генерація ключів.

– Обирається випадкове значення x з проміжку (0, l), де l – порядок точки генератора групи, над якою ми працюємо.
– Обчислюється публічний ключ P:

$$P = x \cdot G,$$

де G – генератор групи.

– Обчислюється додатковий публічний ключ, або так званий образ, I:

$$I = x \cdot P.$$

Таким чином ми маємо 3 необхідних ключі – приватний, публічний та образ x, P та I відповідно.

Розглянемо процес створення кільцевого підпису.

– Обирається набір S’ будь-якого розміру n з публічних ключів інших користувачів P_i. Підписувач також поміщає свій публічний ключ в цей набір на будь-яку позицію s,

$$0 \leq s \leq n.$$

– Генеруються два набори випадкових чисел q та w в проміжку (1, l) розмірами (n + 1) елементів.

– Створюється два набори L та R за такими правилами:

$$L_i = \begin{cases} q_i \cdot G, & \text{якщо } i = s \\ q_i \cdot G + w_i \cdot P_i, & \text{якщо } i \neq s \end{cases}$$

$$R_i = \begin{cases} q_i \cdot P_i, & \text{якщо } i = s \\ q_i \cdot P_i + w_i \cdot I, & \text{якщо } i \neq s \end{cases}$$

– Обраховується так званий “виклик” (challenge) C:

$$C = H(m, L_1, L_2, \dots, L_n, R_1, \dots, R_n).$$

Тобто в обрану попередньо криптографічну функцію H передається конкатенація повідомлення m та всіх елементів L та R.

– Наприкінці обчислюються два набори c та r за наступними формулами:

$$c_i = \begin{cases} w_i, & \text{якщо } i \neq s \\ C - \sum_{i=0}^{n} c_i \pmod{l}, & \text{якщо } i = s \end{cases}$$

$$r_i = \begin{cases} q_i, & \text{якщо } i \neq s \\ q_i + c_s \cdot x \pmod{l}, & \text{якщо } i = s \end{cases}$$

Отриманий підпис складається з образу I, та двох наборів c та r.

Тепер розглянемо процес верифікації кільцевого підпису.

– Обчислюється два набори L’ та R’ за наступними формулами:

$$\begin{cases} L_i' = r_i \cdot G + c_i \cdot P_i \\ R_i' = r_i \cdot P_i + c_i \cdot I \end{cases}$$

– Виконується наступна перевірка:

$$\sum_{i=0}^{n} c_i = H(m, L'_0, \dots, L'_n, R'_0, \dots, R'_n) \pmod{l}.$$

Якщо рівність істинна, то підпис вважається дійсним.

Методи реалізації кільцевих підписів можуть бути різними, але найпоширеніші інструменти включають саме криптографію еліптичних кривих та криптографічні хеш-функції.

Кільцеві підписи мають важливі властивості безпеки. Анонімність гарантує, що особа підписувача залишається прихованою всередині кільця, що унеможливлює віднесення підпису до конкретного учасника. Ця властивість надала кільцевим підписам практичного застосування в різних сферах. Прикладом можуть бути криптовалюти, орієнтовані на конфіденційність, такі як Monero, де кільцеві підписи використовуються для приховування особи відправника під час транзакцій, забезпечуючи конфіденційність транзакцій.

2.6 Порівняння основних алгоритмів цифрового підпису

У таблиці 2.1 проведено порівняння алгоритмів RSA, DSA, ECDSA та EdDSA за деякими основними параметрами, як популярність, продуктивність та безпека.

Таблиця 2.1 – Порівняння описаних алгоритмів цифрового підпису [25]

	Популярність використання	Продуктивність	Безпека
RSA	Широко впроваджений і підтримуваний	Більші ключі призводять до довшої генерації	Середня, сучасні алгоритми можуть швидше факторизувати числа
DSA	Рідко використовується в стандартному вигляді	Швидкий при генерації, повільна верифікація	Висока, залежить від генерації випадкових чисел та довжини ключа
ECDSA	Використовується нечасто	Значно менший розмір ключа, висока швидкість роботи	Висока, потенційно вразливий до певних видів атак
EdDSA	Новий і дуже популярний серед сучасних систем	Значно менший розмір ключа, дуже висока швидкість роботи	Дуже висока, покращений захист від вразливостей ECDSA

Порівнюючи подані алгоритми, можна сказати, що використання RSA вже є не актуальним, оскільки його рівень безпеки є набагато нижчим від конкурентів і напряму залежить від розміру ключа, що призводить до зниження ефективності його роботи.

В алгоритмі DSA суттєво збільшився рівень безпеки, але розмір ключів та відповідно ефективність роботи залишається на одному рівні з RSA. Водночас, критично важливо правильно обирати випадкові значення, щоб досягти максимальної непередбачуваності і, як наслідок, збільшити захищеність від різних атак.

Алгоритм ECDSA зробив величезний стрибок в збільшенні швидкості роботи та зменшенні довжини ключів і підписів порівнюючи з попередніми двома. 256 бітів ключа надають такий же ж рівень захисту, як і 3072 біти при генерації за допомогою RSA чи DSA. Але щодо загальної безпеки, то вона залишилась на рівні DSA схеми, вимагає ретельного вибору генератора чисел та має вразливості до певних видів атак.

EdDSA алгоритм використовує іншу форму еліптичних кривих – криві Едвардса. Їх особливості забезпечують невелику перевагу в швидкості операцій

додавання та скалярного множення над точками, і найважливіше – пропонують суттєві покращення щодо безпеки і захисту від атак, які були загрозливими для ECDSA. Замість того, щоб покладатися на випадкове число для потрібного значення, EdDSA генерує його детерміновано у вигляді хешу, що робить його стійким до колізій.

Порівняння алгоритму кільцевого підпису з рештою не має сенсу, бо він в своїй основі покладається на один із наведених попередньо алгоритмів з використанням публічних ключів інших користувачів. Тому його ефективність та безпека залежить від обраного алгоритму, а рівень анонімності особи лінійно залежить від кількості обраних публічних ключів, так само як і розмір самого підпису.

РОЗДІЛ 3 ТЕХНІЧНА РЕАЛІЗАЦІЯ ЦИФРОВОГО ПІДПИСУ В СИСТЕМІ АНОНІМНОГО ГОЛОСУВАННЯ

3.1 Вимоги та загальна структура системи анонімного голосування

Загальним технічним завданням була реалізація повноцінної системи анонімного голосування на базі блокчейну. Цей проєкт передбачає за собою досить складну архітектуру, вимоги, основні компоненти та роботу якої варто згадати.

Загальними вимогами до системи анонімного голосування стали:

- анонімність: користувачам необхідно надати можливість брати участь у голосуванні із забезпеченням анонімного віддання голосу;
- відкритість: дані у вигляді транзакцій необхідно зберігати у відкритому вигляді, тобто будь-хто може перевірити вміст блокчейну;
- децентралізованість: у системі немає центрального сховища, блокчейн повинен синхронізовано зберігатись у всіх валідаторів;
- масштабованість: система повинна мати можливість приєднувати нових валідаторів та підтримувати додавання інших типів транзакцій;
- цілісність даних: не повинно бути можливості змінити або видалити дані чи транзакції, які уже знаходяться в блокчейні.

Як уже було згадано, за основу системи взято блокчейн. Особливі сутності – валідатори – будуть запущені як окремі процеси на різних пристроях, і будуть відповідати за створення блоків, їх додавання в блокчейн, верифікацію транзакцій та наповнення блоку транзакціями.

Так як в системі є більше одного валідатора, то необхідно було обрати метод досягнення консенсусу для прийняття рішень. Таким було обрано так званий консенсус Практичної візантійської відмовостійкості (Practical Byzantine Fault Tolerance). За цим алгоритмом, один із валідаторів буде пропонувати новий блок для додавання в блокчейн решті, і шляхом отримання більше певного відсотку голосів підтвердження (наприклад більше двох третіх підтверджень зі всіх валідаторів) пропонований блок додаватиметься у блокчейн, а інакше

відкидатиметься. Таким чином кожен з валідаторів буде зберігати копію блокчейну і ми отримаємо децентралізовану систему зі зберігання транзакцій.

Сам блок складається з трьох компонент:

- заголовок – це спеціальна структура з деякими інформаційними та корисними даними як версія, хеш-значення попереднього блоку, прив'язка до часу та корінь дерева Меркла від усіх транзакцій, що використовується для перевірки цілісності даних;
- тіло блоку – це список всіх транзакцій, що входять в даний блок;
- "засвідчення" (witness) – це по суті цифрові підписи валідаторів про підтвердження даного блоку, що слугує для перевірки та доказу його додавання до блокчейну.

Транзакції у даній системі також є окремою структурою, що складається з декількох елементів:

- тип транзакції;
- тіло транзакції – залежить від типу транзакції, несе в собі відповідну основну інформацію про транзакцію;
- випадкове число – потрібне для зменшення ймовірності появи колізій при хешуванні транзакції;
- цифровий підпис – доказ створення цієї транзакції користувачем з конкретним публічним ключем;
- публічний ключ підписувача.

Щодо типів транзакції, то загалом є чотири різних типи транзакцій, та ще один окремий тип, який суттєво відрізняється від інших. До перших чотирьох типів належить такі:

- транзакція зі створення акаунту – як зрозуміло з назви, вона несе в собі інформацію про створення нового акаунту в системі;
- транзакція зі створення групи – подібно до створення акаунту, ця транзакція несе інформацію про створення групи, до якої відноситься певна кількість публічних ключів користувачів;
- транзакція створення голосування;

– транзакція віддання голосу.

Особливим типом транзакції є транзакція анонімного віддання голосу. Її відрізняє від решти факт того, що ми не можемо зберігати там публічний ключ підписувача, а отже робота з нею буде дещо відрізнятись порівняно з іншими, про що ми ще згадаємо пізніше.

Щодо акаунтів, попередньо передбачається три різних типи:

– адміністратор з реєстрації – підписує транзакції про реєстрації інших користувачів системи;
– адміністратор створення голосування – користувач, що має можливість створювати транзакції з голосуванням;
– користувач – може лише приймати участь у голосуваннях.

Зрозуміло що сама система є трохи складнішою, але описана вище інформація дозволяє зрозуміти її базову архітектуру та структуризацію і перейти до опису проблематики та реалізації можливого вирішення.

3.2 Цифрові підписи в системі та вимоги до них

Проаналізувавши попередній розділ, можна побачити що цифрові підписи відіграють одну з важливих ролей серед цієї системи. Загалом, вони використовуються в двох випадках:

– підписання валідатором нового блоку, який додається до блокчейну;
– підписання користувачем створення нової транзакції.

А так як в системі є можливість анонімного та неанонімного віддання голосу, то потрібно реалізувати підпис і з такими можливостями. Тобто можемо сформулювати наступні вимоги до цифрових підписів у системі:

– високий рівень безпеки
– висока ефективність роботи (відносно обраного рівня безпеки)
– можливість неанонімного підпису (для валідаторів та користувачів)
– можливість анонімного підпису (для користувачів)

Враховуючи всі потреби, а також розглянувши різні типи цифрових підписів в розділі 2, необхідно реалізувати два алгоритми: EdDSA та алгоритм кільцевого підпису на основі еліптичних кривих.

Алгоритм EdDSA є найсучаснішим серед розглянутих кандидатів, надає високий рівень безпеки та є найефективнішим серед них. Кільцевий підпис в свою чергу зможе гарантувати високий рівень анонімності, а, створений на базі еліптичних кривих, також матиме чудові показники захисту та ефективності.

3.3 Технічна реалізація, зміни та покращення

Для реалізації всіх необхідних алгоритмів було використано мову програмування Go, тому що вона надає чудові інструменти для створення масштабованого програмного забезпечення та роботи з потоками, що є важливою частиною цієї системи, зокрема для процесів валідатора.

Для зручної роботи з великими цілочисельними значеннями та довгої арифметики використаний підмодуль big модуля math, який надає всю необхідну функціональність. Також у якості криптографічної хеш-функції було обрано вбудований алгоритм SHA-256 модуля crypto.

Першим етапом реалізації став вибір еліптичної кривої та її параметрів, а після цього власне реалізація алгоритму EdDSA. Переглянувши аналіз ефективності математичних операцій використовуючи різні криві, було вирішено використати одну з відомих кривих форми Монтгомері Curve25519, яка є спорідненою з кривою Ed25519, яка згадувалась раніше в алгоритмі EdDSA [26][27]. Фактично, Ed25519 є поворотом кривої Curve25519, що означає що криві відображаються одна в одну. За допомогою відповідних формул можна використовувати один і той же приватний ключ для генерації відкритих ключів на обох кривих, а потім трансформувати ці відкриті ключі між собою без знання приватного ключа за потреби.

Також, так як ми використовуємо криву у формі Монтгомері, було вирішено застосувати алгоритми пришвидшеної арифметики цих кривих, які за відомостями

певних праць не поступаються в ефективності, а іноді є трохи швидшими, ніж арифметика на основі кривих Едвардса [28].

Було створено інтерфейс для майбутньої підтримки кривих, відмінних від Curve25519, та описані всі необхідні методи цих кривих для повноцінної роботи підпису.

Ще одним покращенням стало стиснене зберігання точки на кривій в розмірі 33 байт, та відповідні конвертації одна в одну. Це дозволяє зменшити загальний розмір підпису при зберіганні в транзакціях та блоці.

Спочатку було вирішено імплементувати звичайний алгоритм ECDSA та відповідні допоміжні функції, для можливих майбутніх потреб та покращень, а також як основа для реалізації інших алгоритмів.

Наступним кроком була власне реалізація алгоритму EdDSA. Відповідно до опису в розділі 2.4, були розроблені всі необхідні функції та методи та протестовано створення підпису та верифікацію.

Другий етапом стала реалізація кільцевого підпису. Схожим чином, відповідно до опису в розділі 2.5, були розроблені всі необхідні функції та методи. Для цього алгоритму також використовувалась крива Curve25519 та пришвидшена арифметика кривих Монтгомері.

Так як, відповідно до розділу 3.2, підписи використовуються в двох випадках, було створено дві сутності, які відповідають за підписання блоку та підписання транзакцій. У випадку генерації підпису для транзакцій також реалізовано два варіанти його створення: анонімний та неанонімний (рисунок 3.1).

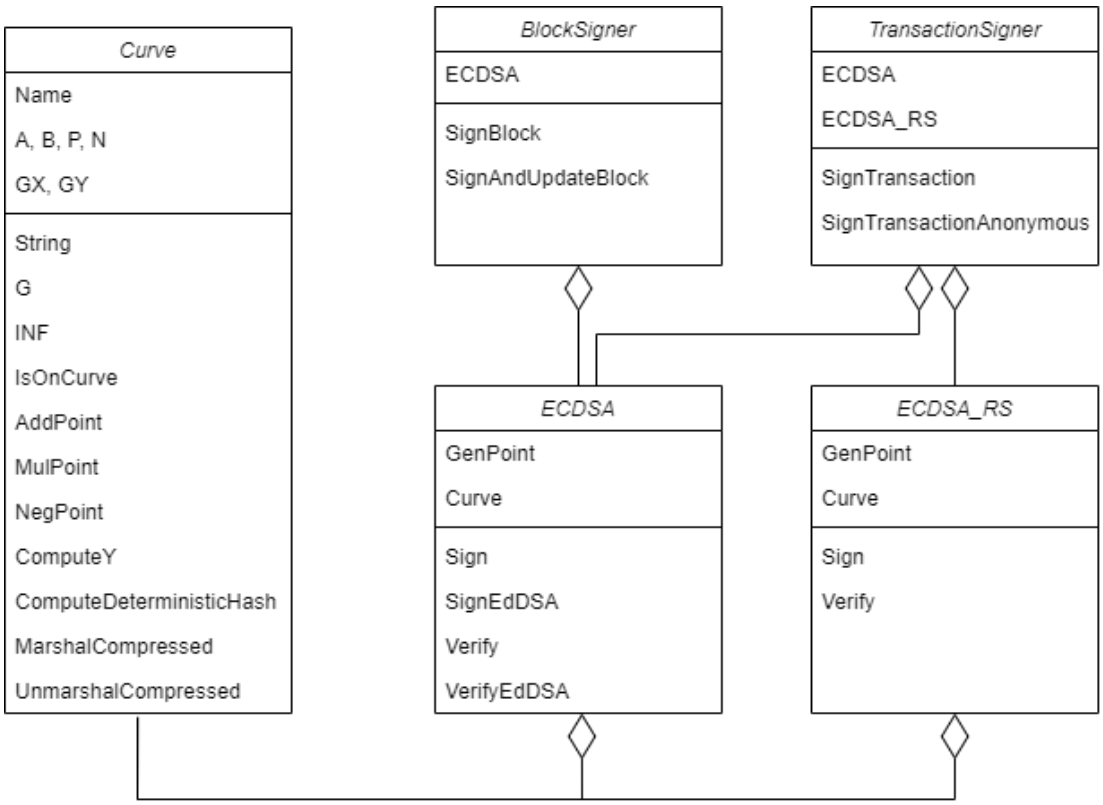

Рисунок 3.1 – Концептуальна діаграма класів для компонентів підписів

3.4 Результати роботи та аналоги системи

Було реалізовано два основних алгоритми:

- EdDSA для створення неанонімних цифрових підписів
- Кільцевий підпис для створення анонімних цифрових підписів

Також було реалізовано ECDSA алгоритм як основа для подальших покращень для реалізацій інших алгоритмів з використанням еліптичних кривих. Вони вже активно використовуються в системі для створення відповідних підписів для транзакцій та блоків.

Також було розглянуто інші системи електронного голосування на базі технології блокчейн. Серед найвідоміших представників були Agora, Horizon State та деякі інші [29]. Деякі з них стверджують, що надають високий рівень анонімності та безпеки.

Проаналізувавши відповідні аналоги, можна прийти до висновку, що більшість використовували звичайні хеш-функції для надання анонімності

віддання голосу, або ж спирались на безпеку реалізації самого блокчейну, який також проводить хешування відповідних структур.

Щодо цифрових підписів, переважна частина різних застосунків використовують звичайні ECDSA або ж EdDSA алгоритми з пришвидшеною арифметикою кривих Едвардса. Використання кільцевого підпису було знайдено лише в системі AMVchain [30].

Тобто, порівнюючи реалізовану систему з аналогами, ми досягнули високого рівня безпеки, використовуючи одночасно архітектуру блокчейну, алгоритм EdDSA для створення цифрових підписів та хешування за допомогою алгоритму SHA-256. Також система забезпечує користувачам анонімність завдяки кільцевому підпису на базі еліптичних кривих з використанням тієї ж хеш-функції SHA-256, що набагато надійніше від аналогічних систем. Ще однією модернізацією системи в порівнянні з іншими стало використання пришвидшеної арифметики Монтгомері.

ВИСНОВКИ

Цифрові підписи набувають все більшої популярності та все частіше використовуються в системах, які намагаються надати максимальну безпеку та конфіденційність для своїх користувачів. Уже існує дуже велика кількість теоретичних праць з аналізом безпеки та ефективності багатьох алгоритмів, але ця сфера все ще далеко від повного її вивчення, а нові здобутки з'являються кожного року. Тому для розробників та користувачів систем з їх використанням дуже важливо постійно слідкувати за останніми змінами та досягненнями.

У цій роботі розглянуто різні аспекти цифрових підписів, зокрема:

- безпека цифрових підписів;
- ефективність в порівнянні з бажаним рівнем безпеки;
- вразливості деяких алгоритмів;
- практичне використання цифрових підписів.

Більш детально було розглянуто найвідоміші алгоритми цифрових підписів: RSA, DSA, ECDSA та EdDSA. Також проведено порівняння цих алгоритмів, в ході якого вияснили, що RSA алгоритм вже є застарілим та ненадійним, в той час як EdDSA перевершує решту за практично всіма параметрами.

Також розглянули особливий вид цифрового підпису, кільцевий підпис, основною відмінністю якого є забезпечення високого рівня анонімності його користувачам за рахунок формування підпису з використанням публічних ключів інших користувачів.

Під час практичної частини роботи було розглянуто вимоги та загальну структуру системи анонімного голосування на базі технології блокчейн, після чого проаналізовано завдання та вимоги до цифрового підпису в цій системі відповідно до потреб його використання. Таким чином було обрано та реалізовано два основних алгоритми цифрового підпису: EdDSA та кільцевий підпис.

Модифікаціями та покращеннями стали:

- використання кривої Curve25519 з відповідними параметрами, як однієї з найбезпечніших наразі відомих кривих (128-бітовий рівень захисту);

- застосування пришвидшеної арифметики кривих форми Монтгомері, що збільшує швидкість створення та верифікації підписів;
- стиснене зберігання точки кривої у розмірі 33 байт та відповідні конвертації одна в одну;
- створення інтерфейсу для можливості додавання реалізацій інших кривих в майбутній розробці.

З можливих покращень можна реалізувати зберігання точки у більш стисненій формі з розміром в 32 байти [21], а також створення інтерфейсу для алгоритмів родини ECDSA, що використовують еліптичні криві в своїй основі та мають схожі механізми створення підпису та верифікації, для можливості підбору параметрів, щоб знайти компроміс між безпекою та ефективністю роботи.

Загалом, реалізовані алгоритми повністю задовольняють усі потреби даної системи, до яких входять високий рівень безпеки та ефективності, а також забезпечення анонімності користувачів при голосуванні.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. What is Digital Signature: How it works, Benefits, Objectives, Concept [Електронний ресурс] : Eliza Paul – Режим доступу до ресурсу: https://www.emptrust.com/blog/benefits-of-using-digital-signatures/.
2. Electronic Signatures vs Digital Signatures [Електронний ресурс] – Режим доступу до ресурсу: https://www.gonitro.com/sign/digital-signatures-vs-electronic-signatures.
3. Digital Signatures and Certificates [Електронний ресурс] – Режим доступу до ресурсу: https://www.geeksforgeeks.org/digital-signatures-certificates/.
4. Cryptography Digital signatures [Електронний ресурс] – Режим доступу до ресурсу: https://tinyurl.com/4m3a3k3n.
5. Customer confidence in electronic signatures - What is an electronic signature? (and other definitions) [Електронний ресурс] : Cryptomathic / Francis Richards – Режим доступу до ресурсу: https://tinyurl.com/4b2uf9km.
6. Teske E. The SHA-1 Attack Further Emphasizes the Need for Crypto-Agility [Електронний ресурс] / Edlyn Teske. – 2020. – Режим доступу до ресурсу: https://tinyurl.com/476cwxf4.
7. Vacca J. Public Key Infrastructure: Building Trusted Applications and Web Services / John Vacca., 2004. – С. 7-18.
8. WHAT IS PKI? [Електронний ресурс] – Режим доступу до ресурсу: https://www.entrust.com/resources/certificate-solutions/learn/what-is-pki.
9. What is PKI? [Електронний ресурс] – Режим доступу до ресурсу: https://tinyurl.com/y3yd6wt2.
10. How different industries use digital signatures [Електронний ресурс] – Режим доступу до ресурсу: https://www.foxit.com/blog/how-different-industries-use-digital-signatures/.
11. Digital Signature Uses [Електронний ресурс] – Режим доступу до ресурсу: https://www.educba.com/digital-signature-uses/.

12. Rivest R. L. A Method for Obtaining Digital Signatures and PublicKey Cryptosystems / R. L. Rivest, A. Shamir, L. Adleman. // Communications of the ACM. – 1978. – №2. – С. 120–126.
13. Cobb M. RSA algorithm (Rivest-Shamir-Adleman) [Електронний ресурс] / Michael Cobb – Режим доступу до ресурсу: https://www.techtarget.com/searchsecurity/definition/RSA.
14. Vries A. The ray attack, an inefficient trial to break RSA cryptosystems [Електронний ресурс] / Andreas de Vries. – 2003. – Режим доступу до ресурсу: https://arxiv.org/pdf/cs/0307029.pdf.
15. Jena B. K. Digital Signature Algorithm (DSA) in Cryptography: How It Works & More [Електронний ресурс] / Baivab Kumar Jena // Simplilearn. – 2023. – Режим доступу до ресурсу: https://tinyurl.com/yztm5drv.
16. Romine C. H. Digital Signature Standard (DSS) / Charles H. Romine. // FEDERAL INFORMATION PROCESSING STANDARDS PUBLICATION. – 2023.
17. LAWSON N. DSA requirements for random k value [Електронний ресурс] / NATE LAWSON // rdist. – 2010. – Режим доступу до ресурсу: https://rdist.root.org/2010/11/19/dsa-requirements-for-random-k-value/.
18. Johnson D. The Elliptic Curve Digital Signature Algorithm (ECDSA) [Електронний ресурс] / D. Johnson, A. Menezes, S. Vanstone // Certicom Corporation. – 2001. – Режим доступу до ресурсу: https://www.cs.miami.edu/home/burt/learning/Csc609.142/ecdsa-cert.pdf.
19. Nakov S. ECDSA: Elliptic Curve Signatures [Електронний ресурс] / Svetlin Nakov. – 2018. – Режим доступу до ресурсу: https://cryptobook.nakov.com/digital-signatures/ecdsa-sign-verify-messages.
20. Blockchain – Elliptic Curve Digital Signature Algorithm (ECDSA) [Електронний ресурс] – Режим доступу до ресурсу: https://tinyurl.com/9wzh2f3d.
21. Josefsson S. Edwards-Curve Digital Signature Algorithm (EdDSA) [Електронний ресурс] / S. Josefsson. – 2017. – Режим доступу до ресурсу: https://www.rfc-editor.org/rfc/rfc8032.

22. Nakov S. EdDSA and Ed25519 [Електронний ресурс] / Svetlin Nakov. – 2018. – Режим доступу до ресурсу: https://cryptobook.nakov.com/digital-signatures/eddsa-and-ed25519.
23. Anonymous Decentralized E-Voting System [Електронний ресурс] / [O. Kurbatov, P. Kravchenko, O. Shapoval та ін.]. – 2019. – Режим доступу до ресурсу: https://ceur-ws.org/Vol-2588/paper2.pdf.
24. Saberhagen N. CryptoNote v 2.0 [Електронний ресурс] / Nicolas van Saberhagen. – 2013. – Режим доступу до ресурсу: https://bytecoin.org/old/whitepaper.pdf.
25. Kontsevoy E. Comparing SSH Keys - RSA, DSA, ECDSA, or EdDSA? [Електронний ресурс] / E. Kontsevoy // Teleport. – 2022. – Режим доступу до ресурсу: https://goteleport.com/blog/comparing-ssh-keys/.
26. Bernstein D. J. Curve25519: new Diffie-Hellman speed records [Електронний ресурс] / Daniel Julius Bernstein – Режим доступу до ресурсу: http://cr.yp.to/ecdh/curve25519-20060209.pdf.
27. Bernstein D. J. A state-of-the-art Diffie-Hellman function [Електронний ресурс] / Daniel Julius Bernstein – Режим доступу до ресурсу: http://cr.yp.to/ecdh.html.
28. Meyer M. On hybrid SIDH schemes using Edwards and Montgomery curve arithmetic [Електронний ресурс] / M. Meyer, S. Reith, F. Campos – Режим доступу до ресурсу: https://eprint.iacr.org/2017/1213.pdf.
29. Tasmia S. A. DVTChain: A blockchain-based decentralized mechanism to ensure the security of digital voting system voting system [Електронний ресурс] / Syada Alvi Tasmia. – 2022. – Режим доступу до ресурсу: https://www.sciencedirect.com/science/article/pii/S1319157822002221.
30. Chenchen L. AMVchain: authority management mechanism on blockchain-based voting systems [Електронний ресурс] / L. Chenchen, X. Jiang, D. Xiaohai. – 2021. – Режим доступу до ресурсу: https://link.springer.com/article/10.1007/s12083-021-01100-x.