Хмельницький національний університет
Факультет інформаційних технологій
Кафедра комп'ютерної інженерії та інформаційних систем

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

Галузь знань 12 – Інформаційні технології

Спеціальність 123 –Комп'ютерна інженерія

на тему «Метод та система розподілених обчислень із використанням асиметричних алгоритмів шифрування»

КвРКІ. 013042.17.01.01 ПЗ

Виконав: студент 2 курсу, група КІ2м-20-1 _____ Симак Д.О.
Підпис Ініціали, прізвище

Керівник кандидат техн. наук, доцент _____ Каштальян А.С.
Науковий ступінь, вчене звання Підпис Ініціали, прізвище

До захисту допускаю:
Зав. кафедри КІІС, д.т.н., проф. Т.О. Говорущенко _____ 2022 р.

Хмельницький, 2022

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
Кафедра КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ
Освітній рівень МАГІСТР
Галузь знань 12 ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ
Спеціальність 123 КОМП'ЮТЕРНА ІНЖЕНЕРІЯ
Освітня програма ОСВІТНЬО-НАУКОВА ПРОГРАМА «КОМП'ЮТЕРНА ІНЖЕНЕРІЯ ТА ПРОГРАМУВАННЯ»

ЗАТВЕРДЖУЮ
Зав. кафедри Т.О.Говорущенко
"01" 09 2021 р.

ЗАВДАННЯ
НА ДИПЛОМНИЙ ПРОЕКТ (РОБОТУ)

Симаку Денису Олександровичу
Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) Метод та система розподілених обчислень із використанням асиметричних алгоритмів шифрування
Керівник проекту (роботи) Каштальян А.С., к.т.н., професор
Прізвище, ім'я, по батькові, науковий ступінь, вчене звання

Затверджена наказом ректора університету від 06.01.2022 р. № 1
2. Строк подання студентом проекту (роботи) на кафедру 03.05.2022 р.
3. Вихідні дані до проекту (роботи) Завдання на дипломне проектування
4. Зміст пояснювальної записки (перелік питань, які потрібно розробити)
проведення аналізу систем розподілених обчислень;
дослідиження існуючих методів шифрування інформації;
аналіз та дослідження відомих методів розподіленого машиного навчання;
проведення аналізу способів витоку конфіденційної інформації при машиному навчанні;
застосування реалізованої системи розподілених обчислень із застосуванням алгоритмів асиметричного шифрування
5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень)

6. Консультанти розділів дипломного проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Лисенко С.М., професор кафедри КІІС		
Антиплагіат	Нічепорук А.О., доцент кафедри КІІС		

7. Дата видачі завдання «_06_»_____09_____2021р.

КАЛЕНДАРНИЙ ПЛАН

№з/п	Назва етапів (розділів) дипломного проекту (роботи)	Термін виконання етапів проекту (роботи)	Примітка
1	Вибір напряму дослідження та узгодження тематики ДРМ з керівником	05.09.2021	виконано
2	Ознайомлення з предметною областю; формулювання мети та задач дослідження; визначення об'єкта та предмета дослідження	05.10.2021	виконано
3	Робота над розділом 1 – аналіз відомих моделей, методів за темою; постановка задачі	05.11.2021	виконано
4	Робота над розділом 2 – розробка моделей для вирішення поставленої задачі	05.12.2021	виконано
5	Робота над науковою статтею	05.01.2022	виконано
6	Робота над розділом 3 – розробка методів для вирішення поставленої задачі	15.02.2022	виконано
7	Робота над розділом 4 – проектування та розробка ПЗ для вирішення поставленої задачі, експериментальна частина	05.04.2022	виконано
8	Оформлення пояснювальної записки згідно вимог	15.04.2022	виконано
9	Попередній захист ДРМ	18.04.2022	виконано
10	Захист ДРМ на засіданні ЕК	До 10.05.2022	

Студент _____ Підпис Д.О. Симак Ініціали, прізвище

Керівник проекту (роботи) _____ Підпис А.С. Каштальян Ініціали, прізвище

РЕФЕРАТ

Тема дипломної роботи: Метод та система розподілених обчислень із використанням асиметричних алгоритмів шифрування

Автор роботи: Симак Д.О.

Керівник роботи: Каштальян А.С

Пояснювальна записка: 85 с., 19 рис., 3 табл., 4 дод., 40 джерел.

СИСТЕМА РОЗПОДІЛЕНИХ ОБЧИСЛЕНЬ, АСИМЕТРИЧНІ АЛГОРИТМИ ШИФРУВАННЯ, МАШИННЕ НАВЧАННЯ, КОНФІДЕНЦІЙНІСТЬ НАВЧАЛЬНИХ ДАНИХ

Об’єктом дослідження є розподілені обчислення із використанням алгоритмів асиметричного шифрування та їх методи реалізації у сучасних системах.

Предметом дослідження є моделі розподілених обчислень із використанням алгоритмів асиметричного шифрування.

Метою дипломної роботи є аналіз та реалізація рішень застосування методів та систем розподілених обчислень із використанням асиметричних алгоритмів шифрування.

Для розв’язання поставлених задач використовувалися основні положення системного аналізу, методи аналізу даних, методи машинного навчання, методи систем розподіленого обчислення, методи асиметричного шифрування даних.

Наукова новизна отриманих результатів:

- удосконалено метод системи розподілених обчислень за допомогою асиметричних алгоритмів шифрування, який на відміну від інших, проводить додатковий аудит для перевірки цілісності даних.

На основі проведених досліджень розроблена архітектура і компоненти програмного забезпечення, які можуть використовуватись в комерційних цілях.

Практична значимість отриманих результатів полягає у вдосконалених моделях систем розподілених обчислень для використання їх для машинного навчання зі збереженням цілісності даних

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ.....6
ВСТУП.....7
1 АНАЛІЗ ВІДОМИХ МЕТОДІВ СИСТЕМ РОЗПОДІЛЕНИХ ОБЧИСЛЕНЬ…10
1.1 Огляд та поняття системи розподілених обчислень.....10
1.2 Дослідження поняття хмарних обчислень як системи розподілених обчислень.....15
1.3 Проблеми безпеки систем розподіленого обчислення та їх способи вирішення.....21
1.4 Постановка задачі.....27
1.5 Висновок.....27
2 МЕТОДИ ЗАСТОСУВАННЯ РОЗПОДІЛЕНОГО МАШИННОГО НАВЧАННЯ В СИСТЕМАХ РОЗПОДІЛЕНОГО ОБЧИСЛЕННЯ З ВИКОРИСТАННЯМ АСИМЕТРИЧНИХ АЛГОРИТМІВ ШИФРУВАННЯ.....28
2.1 Огляд та поняття розподіленого машинного навчання.....28
2.2 Аналіз відомих методів та засобів забезпечення захисту витоку конфіденційних даних.....32
2.3 Дослідження засобів забезпечення захисту витоку конфіденційних даних 37
2.4 Аналіз асиметричного алгоритму шифрування.....39
2.4.1 RSA алгоритм.....42
2.4.2 Deffie-Hellman алгоритм.....43
2.5 Опис вибраного метода захисту цілісності навчальних даних в розподіленій системі МН.....45
2.6 Основні математичні алгоритми, що застосовуються у даному дослідженні.....46
2.6.1 Білінійне відображення.....46
2.6.2 Проблема дискретного логарифму (ПДЛ).....46

2.6.3 Проблема обчислювального алгоритму Діффі-Хеллмана (ОАДХ)….47
2.6.4 Проблема спільно-обчислювального білінійного алгоритму Діффі-Хеллмана (СОАДХ).....47
2.6.5 Модель випадкового прогнозування.....47
2.7 Висновок.....48
3 МОДЕЛЬ СИСТЕМИ РОЗПОДІЛЕНОГО МАШИННОГО НАВЧАННЯ НА ОСНОВІ СХЕМИ ПЕРЕВІРКИ ЦІЛІСНОСТІ ДАНИХ З ВИКОРИСТАННЯМ АСИМЕТРИЧНИХ АЛГОРИТМІВ ШИФРУВАННЯ.....50
3.1 Модель системи РМН-ПЦД.....50
3.2 Приклад застосування даної системної моделі.....51
3.3 Детальна побудова схеми РМН-ПЦД.....52
3.3.1 Налаштування.....53
3.3.2 Створення додаткового ключа.....53
3.3.3 Генерація тегів.....54
3.3.4 Запит СА.....56
3.3.5 Генерація відповіді підтвердження цілісності.....56
3.3.6 Перевірка відповіді від СД.....57
3.4 Перевірка правильності побудови схеми РМН-ПЦД.....57
3.5 Аналіз забезпечення безпеки схемою РМН-ПЦД.....59
3.5.1 Формальні доведення.....59
3.5.2 Порівняння безпеки схеми РМН-ПЦД з іншими схемами.....66
3.6 Висновок.....66
4 РЕАЛІЗАЦІЯ ТА АНАЛІЗ ПРОДУКТИВНОСТІ СИСТЕМИ РОЗПОДІЛЕНОГО МАШИННОГО НАВЧАННЯ НА ОСНОВІ СХЕМИ ПЕРЕВІРКИ ЦІЛІСНОСТІ ДАНИХ З ВИКОРИСТАННЯМ АСИМЕТРИЧНИХ АЛГОРИТМІВ ШИФРУВАННЯ.....67
4.1 Аналіз продуктивності.....67
4.2 Обрахунок обчислень.....67
4.2.1 Витрати на обчислення для СД.....68

4.2.2 Витрати на обчислення для СА.....69
4.2.3 Витрати на обчислення для ВНД.....70
4.3 Комунікаційні витрати.....71
4.3.1 Накладні витрати на зв’язок для запиту.....72
4.3.2 Накладні витрати на зв’язок для доказу.....72
4.4 Оцінка схеми РМН-ПЦД.....74
4.5 Основні компоненти для створення програмної реалізації.....75
4.6 Розробка структури веб-додатку для прогнозування цін на авто за допомогою МН.....80
4.6.1 Лінійна регресія.....80
4.6.2 Регресія з використанням дерева рішень.....82
4.6.3 Регресія з використанням Random forest.....83
4.6.4 Регресія із використанням дерева рішень під впливом градієнта……85
4.6.5 Результати дослідження методів регресії.....85
4.6 Висновок.....89
ВИСНОВКИ.....90
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ.....92
Додаток А.....98
Додаток Б.....99
Додаток В.....113
Додаток Г.....116

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

СРО — система розподілених обчислень

ПК — персональний комп’ютер

ААШ — асиметричні алгоритми шифрування

МН — машинне навчання

РМН — розподілене машинне навчання

ШІ — штучний інтелект

КД — конфіденційні дані

ГШ — гомоморфне шифрування

ПЗ — програмне забезпечення

ОС — операційна система

НД — набір даних

ДК — диференціальна конфіденційність

СА — стороній аудитор

ВНД — власник набору даних

ЦГК — центр генерування ключів

СД — сервер даних

РМН-ПЦД — розподілене машинне навчання на основі схеми перевірки цілісності даних

ВСТУП

На теперішній час розвиток інформаційних та телекомунікаційних технологій знаходиться на тій стадії, коли в розподіленому інформаційно-телекомунікаційному середовищі все більшого значення набуває не тільки необхідність доступу та обміну інформацією, але й виконання різноманітних видів аналізу та обробки цієї інформації.

В сьогоденні комп’ютери впроваджуються в усі види діяльності, і через їх постійне застосування приходиться нарощувати їх обчислювальну потужність, тому що використання комп’ютерних мереж різного масштабу потребує задіяння значного обсягу НД, що в свою чергу призводить до дефіциту обчислювальних ресурсів при виконанні різноманітних обчислювальних процесів. Основним та раціональним шляхом для вирішення цих проблем являється застосування паралельних та розподілених обчислень.

Не малу роль при обміні інформацією відіграє приватність, а точніше конфіденційність, даних. Тобто, та інформація, якою ми ділимось повинна бути зашифрованою, і для цього використовують різні алгоритми синхронного та асинхронного шифрування, але в загальному користуються гібридними системами де використовуються обидва метода шифрування.

Машинне навчання дозволяє створювати системи, які навчаються, або удосконалюють продуктивність, за допомогою аналізу даних. Тобто МН допомагають зробити взаємодію з користувачем зручнішою, ефективнішою та безпечною. На даний момент можливості МН є тільки вершиною айсбергу, тієї кількості можливостей які з’являться у майбутньому.

Актуальність роботи полягає в аналізі вже існуючих методів та розробці нових рішень застосування методів та систем розподілених обчислень із використанням асиметричних алгоритмів шифрування.

Об’єктом дослідження є розподілені обчислення із використанням алгоритмів асиметричного шифрування та їх методи реалізації у сучасних системах.

Предметом дослідження є математичні моделі розподілених обчислень із використанням алгоритмів асиметричного шифрування.

Метою дипломної роботи є аналіз та реалізація рішень застосування методів та систем розподілених обчислень із використанням асиметричних алгоритмів шифрування.

Завданнями роботи є:

1) провести аналіз систем розподілених обчислень;
2) дослідити існуючі методи шифрування інформації;
3) провести аналіз та дослідити відомі методи розподіленого машиного навчання;
4) провести аналіз способів витоку конфіденційної інформації при машиному навчанні;
5) дослідити існуючі методи запобігання та захисту конфіденційної інформації;
6) підвести підсумки по розглянутій інформації та про необхідність створення та розробки нової системи;
7) дослідити застосування алгоритмів асиметричного шифрування для захисту потоків даних;
8) застосувати реалізовану систему розподілених обчислень із застосуванням алгоритмів асиметричного шифрування.

Наукова новизна отриманих результатів:

- удосконалено метод системи розподілених обчислень за допомогою асиметричних алгоритмів шифрування, який на відміну від інших, проводить додатковий аудит для перевірки цілісності даних.

На основі проведених досліджень розроблена архітектура і компоненти програмного забезпечення, які можуть використовуватись в комерційних цілях.

Практична значимість отриманих результатів полягає у вдосконалених моделях систем розподілених обчислень для використання їх для машинного навчання зі збереженням цілісності даних

1 АНАЛІЗ ВІДОМИХ МЕТОДІВ СИСТЕМ РОЗПОДІЛЕНИХ ОБЧИСЛЕНЬ

1.1 Огляд та поняття системи розподілених обчислень

У сьогоденні існує дуже велика кількість визначень СРО, й самим точним з них є те, що СРО це система пов’язаних між собою комп’ютерів, тобто мережа, в якій обчислювальні ресурси кожного ПК об’єднанні з ресурсами інших ПК цієї системи і є загальними та відкритими для використання [15], яка зображена на рисунку 1.1.

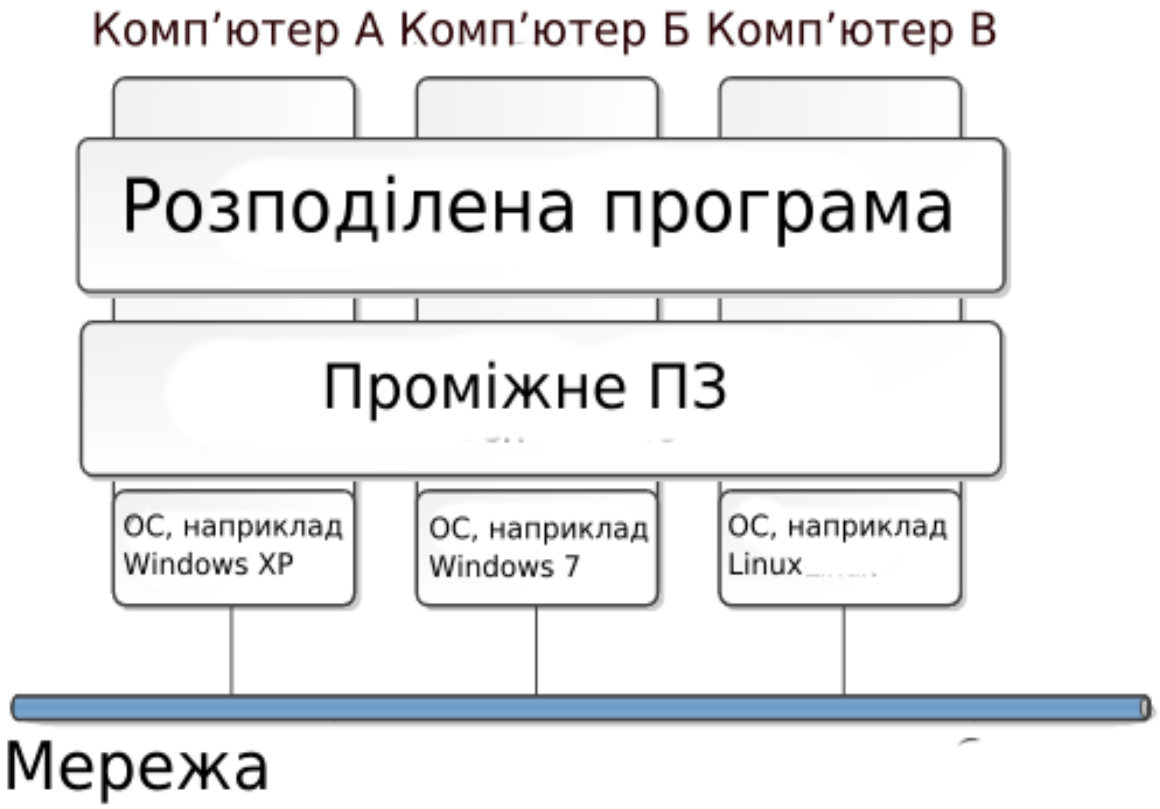


Рисунок 1.1 — Схема розподіленої системи

До прикладу, обчислювальними ресурсами ПК можна віднести такі його характеристики як продуктивність, оперативну пам’ять, БД, тактову частоту процесора, тощо [4].

Кожен користувач цієї системи, може будь-яким способом використовувати ці ресурси для вирішення тих, чи інших, математичних чи інших задач, а також він повинен надавати доступ зі свого ПК до великих обчислювальних потужностей та пристроям зберігання даних достатньо великого об’єму.

Для того щоб створити СРО, потрібно мати хоча б один ПК, який буде виконувати адміністративні функції в системі [28].

Також, ПК в СРО працюють зі спеціальним ПЗ для СРО, при цьому вони можуть працювати як на різних платформах та ОС, так і на однакових.

Принцип роботи СРО полягає в тому, що якщо на якомусь ПК не вистачає тих чи інших ресурсів для різного роду обчислень, то використовуються резерв потужностей іншого ПК, який в той час не виконує певну роботу, простішими словами “простоює”, для того щоб значно знизити час рішення задач великої складності [34].

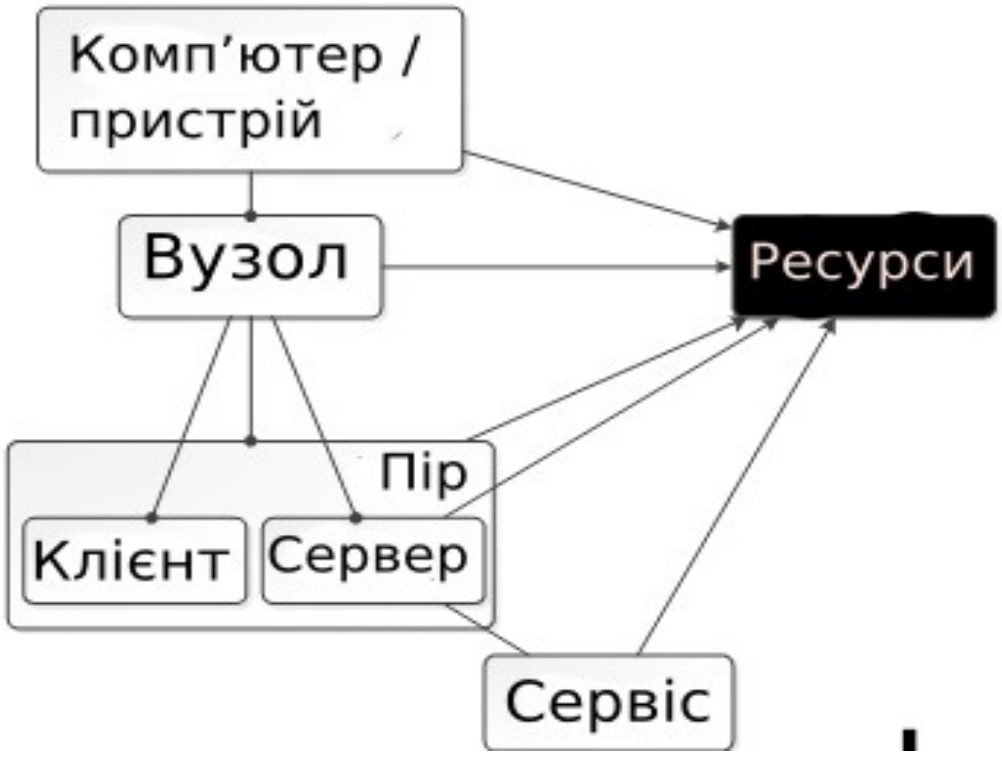


Рисунок 1.2 — Схема взаємодії в СРО

На рисунку 1.2 зображена схема взаємодії між даними в СРО, праналізувавши її, можна сказати, що кожен ПК чи пристрій являє собою сутність СРО у вигляді вузла, при цьому кожен вузол має якусь кількість користувачів, серверів та сервісів [38]. В цьому випадку, сервіс отримує запит на надання якихось даних і повертає відповідь.

Класифікувати СРО можливо за багатьма ознаками:

1) по кількості елементів в системі;
2) по рівню організації СРО;
3) по типу представлених ресурсів;
4) та іншими.

Основними вимогами до СРО [32] є: масштабованість системи, її прозорість, безпека, відкритість та надійність. Прозорість полягає в тому, що СРО повинні бути однорідним об’єктом для користувачів системи, а не просто набором автономних ПК, які між собою взаємодіють. Масштабування є однією з найважливіших властивостей СРО, через те, що в ПЗ, яке керує всією взаємодією компонетів, в основному є обмеження на загальну кількість обчислювальних вузлів системи. В свою чергу, відкритість системи [12] дає змогу взаємодіяти з іншими відкритими системами. Надійність системи, заключається в такому показнику як відмовостійкість, який визначає можливість роботи над певними задачами, яка задала програма, після виникнення якоїсь несправності.

Не дивлячись на велику кількість переваг СРО, має достатню кількість проблем:

1) проблеми адміністрування системи (балансування навантаження на вузли, відновлення даних при виникнені помилок);
2) проблеми обмеженності масштабування СРО (збільшення кількості вузлів, обмеженності можливостей сервера, обмеженності мереж при передачі даних, обмеженності алгоритмів обробки даних);

3) проблеми переносу ПЗ (забезпечення кросплатформенності).

Але основними проблемами, які виникають при задіянні та підтримці [23] СРО, є розробка і використання єдиних стандартів і протоколів та безпека системи. Для використання СРО стандарти та протоколи повинні бути чіткими та однозначними, які встановлюють передачу даних всередині самої системи. Більш цікавою є проблема безпеки системи [32], тому що сюди відносяться такі поняття, як ідентифікація та аутентифікація користувачів [35], конфіденційність та захист їх даних, також їх цілісність, обмеження дозволу на доступ до обчислювальних ресурсів [13], які надає СРО, тим чи іншим групам користувачів. Наприклад, частину питань безпеки можливо вирішити на рівні якихось окремих вузлів СРО за допомогою втановлення антивірусного ПЗ чи фаєрволів, аутентифікацією [40], тощо.

Але завдяки архітектурі СРО [45], підхід такого роду не є панацеєю для вирішення цих проблем, тому що дане ПЗ не завжди може гарантувати необхідну конфіденційність.

На сьогоднішній день, існує велика кількість концепцій та технологій СРО, напрклад, таких як однорангові мережі (peer-to-peer, P2P), сервіс-орієнтована архітектура, агентно-орієнтована парадигма та хмарні обчислення.

Ці технології є витком розвитку розподілених обчислень, які охоплюють поняття віртуальної співпраці та віртуальних організацій. Тобто, віртуальною організацією називають кількість організацій, що об’єднані якимись загальними правилами колективного доступу до обчислювальних ресурсів.

Популярна P2P, не мають центрального сервера [7], а обмінюються ресурсами відповідно між собою, що зображена на рисунку 1.3.

Основними перевагами даних систем, є те що збільшується відказостійкість, тому що функціонування системи не залежить від збою будь-якого вузла, анонімність, а також спрощується підтримка масштабування при збільшені великої кількості вузлів.

Недоліками, є низька ступінь захищенності пристроїв [11] (вони дають відкритий доступ до своїх ресурсів, і збільшується ризик взлому або зараження), збільшення потреби в потужностях кожного пристрою (обчислювальний вузол бере на себе функції, і клієнта, і сервера), можлива гетерогенність апаратного та програмного забезпечення, пошук доступних ресурсів [30] (кожному вузлу потрібно самому собі шукати ресурси для обчислень).

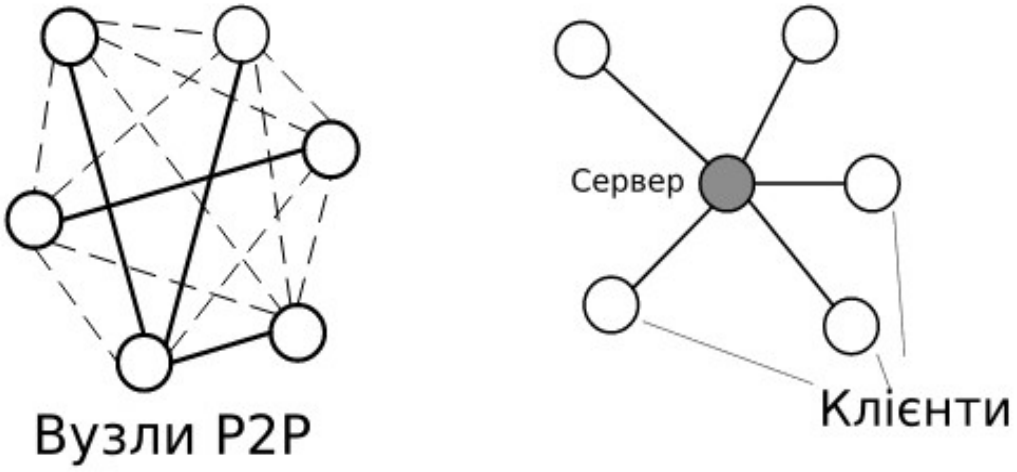


Рисунок 1.3 — Порівняння зв’язків P2P і централізованої архітектури (клієнт-сервер)

Принципи, які визначає дана структура, шукають та замінюють вузли, які вийшли з ладу, на нові. Є два види структур P2P мереж:

1) централізовані — характеризується наявністю трекера, що збирає інформацію про вузли, знаходить та надає необхідні сервіси одних вузлів іншим;

2) децентралізована — характеризується відсутністю виділеного сервера, а усілякий пошук та надання сервісів реалізується за допомогою крокового пошуку, в якому можуть бути задіяні всі вузли.

Найбільше розпоширені P2P, в системах, які обробляють великі об’єми даних [41] та забезпечують міжособний обмін даними між користувачами [37].

Далі більш докладно розгляну поняття хмарних обчислень [17] як СРО, для подальшого аналізу безпеки витоку даних.

1.2 Дослідження поняття хмарних обчислень як системи розподілених обчислень

Хмарні обчислення [27], то парадигма розподіленої та великомасштабної обробки даних, в рамках якої ряд абстрактних, віртуальних та динамічно-масштабованих ресурсів, ресурсів збереження чи сервісів надається користувачу як Інтернет-сервіс. Також, надання даних послуг здійснюватись і через звичайну локальну мережу з використанням веб-технологій.

На сьогоднішній день, хмарні обчислення є однією з найпопулярніших технологій, і привертають до себе багато уваги. Тому що по деяким оцінкам, дана технологія може в декілька разів [21] знизити вартість різних додатків, як і для бізнесу, так і для простого кінцевого користувача.

ХО мають два важливих аспекта, тобто віртуалізацію та масштабованість. Надаються віртуалізовані ресурси за допомогою певних абстрактних інтерфейсів (API) [22], й за допомогою такої архітектури для кінцевого користувача без яких-небудь наслідків забезпечується масштабованість та віртуалізація, зображено на рисунку 1.4. Масштабованість являє собою динамічне налаштування інфо-ресурсів при зміні певного навантаження, тобто потрібно змінити обчислювальну потужність або збільшити ємність для зберігання даних, коли додається/зменшується кількість користувачів. Віртуалізація використовується для забезпечення інкапсуляції та абстракції. Абстракція [25] слугує для того щоб перетворити в уніфікований пул ресурсів необроблені обчислювальні ресурси, а також налаштувати уніфікований шар ресурсів в асбстрагованому вигляді, тобто це віртуалізовані сервери, файлові системи та СУБД. Інкапсуляція дає змогу забезпечити

підвищену безпеку, ізольованість та керуючість додатків. Також не мало важливою, являється особливість хмарних платформ для кінцевого користувача, інтегрованість [9] системного ПЗ та апаратних ресурсів з додатками, яка має вигляд сервісів.

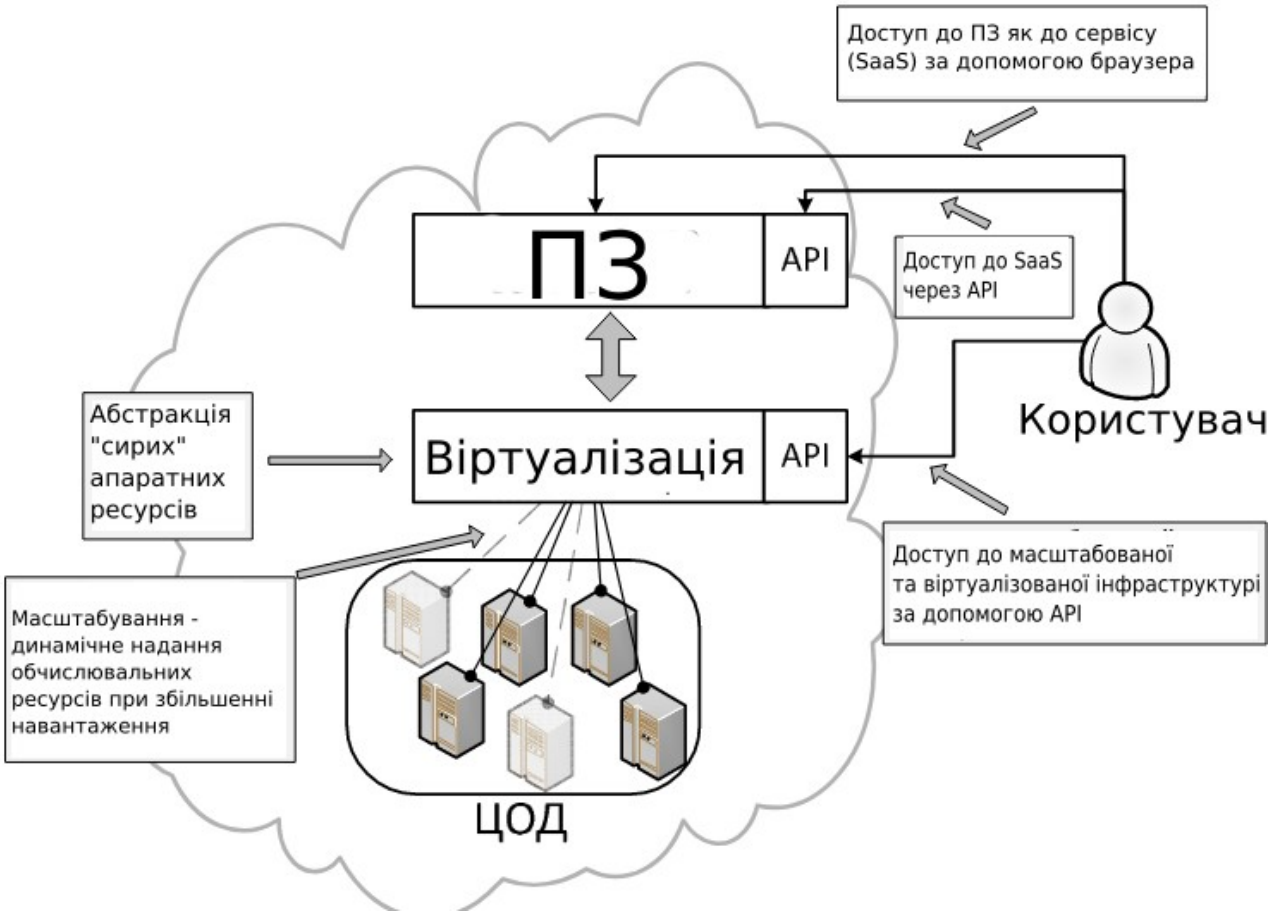


Рисунок 1.4 — Основні характеристики ХО

Архітектура хмарних систем поділяється на три рівня [26], які зображені на рисунку 1.5:

1) IaaS (інфраструктура як сервіс);
2) PaaS (платформа як сервіс);
3) SaaS (ПЗ як сервіс).

При IaaS, інформаційні ресурси, надаються в вигляді сервісу, тобто замість доступу до необроблених обчислювальних пристроїв та систем зберігання, IaaS надає віртуалізовану інфраструктуру у вигляді сервісу.

Необроблені, тобто “сирі”, ресурси базуються на базовому рівні, поверх якого налаштовуються шари сервісів за допомогою віртуалізації, які в кінцевому результаті і отримують користувачі в вигляді IaaS.

Найпопулярнішим прикладом такого підходу являється Amazon Web Services.

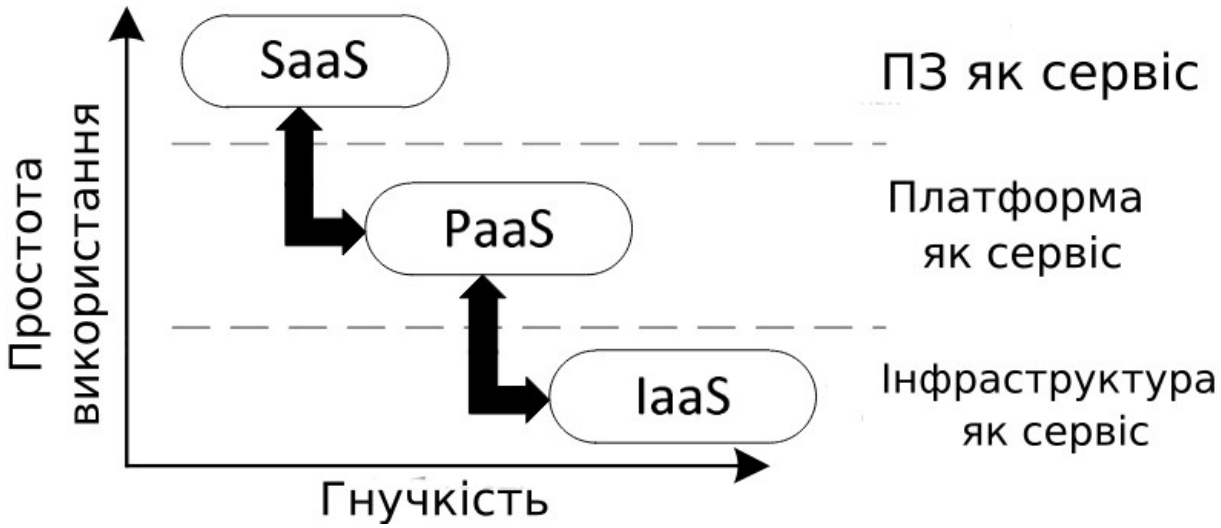


Рисунок 1.5 — Три рівня ХО

Шар PaaS має в собі основу стандартизованого інтерфейсу, що віртуалізує обчислювальні ресурси, який надає IaaS, та стандартний інтерфейс для розробки додатків, працюючих на SaaS, тобто це шар абстракції між віртуалізованою інфраструктурою IaaS та програмними додатками SaaS. Найпопулярнішим прикладом такого підходу являється Google App Enginge.

SaaS є ПЗ, яке надається по принципу “оплата в міру використання” і керується віддаленно, для кінцевих користувачів воно надає реальну цінність і забезпечує вирішення його задач. SaaS-додаток може бути розробленим на базі однієї платформи і виконуватись на інфраструктурі іншої. Найпопулярнішим прикладом є комплекс Google Apps.

Основними компонентами хмарних додатків [19] можна виділити, вони графічно зображені також на рисунку 1.6:

1) платформа — середовище та набор утиліт, які забезпечують розробку, надання та інтеграцію хмарних сервісів;

2) інформація — сховища даних, які забезпечують розподілене зберігання структурованих чи неструктурованих, статичних чи динамічно-змінних даних;

3) представлення — інтерфейс для взаємодії з хмарою користувачем;

4) ідентифікація — дані про користувачів хмарних ресурсів, які використовуються для оптимізації та налаштуваннь хмарного середовища під їх задачі;

5) інтеграція — інфраструктура, що спрощує обмін інформацією і виконання задач в розподіленно обчислювальному середовищі;

6) масштабування — забезпечує виділення додаткових обчислювальних ресурсів при збільшені навантаження на додаток;

7) інтеграція — процес розробки нового хмарного додатку, в котрий входять розробка, тестування та інтеграція в експлуатацію;

8) монетизація — облік та моніторинг ресурсів, використаних для виконання користувацьких задач;

9) функціонування — підтримка та моніторинг додатків, які є в експлуатації.

Головною перевагою [20] ХО — це є модель оплати по мірі їх використання, тобто немає необхідності передчасних інвестицій як в апаратну інфраструктуру так і в ліцензоване ПЗ. Тому користувачі використовують тільки той об’єм обчислювальних ресурсів, які їм потрібен, і оплачують тільки за те, що вони використали. Не забуваємо й про те, що ХО мають змогу легко та швидко надати необхідні обчислювальні ресурси, завдяки своїй гнучкості та масштабованості.

Але недоліки для ХО не стали виключеням, тому деякі витікають з однієї особливості, що вони обслуговують одразу велику кількість клієнтів, і тому

користувач не знає, обчислювальні задачі яких користувачів будуть виконуватись на тому чи іншому сервері [16] хмарної інфраструктури.

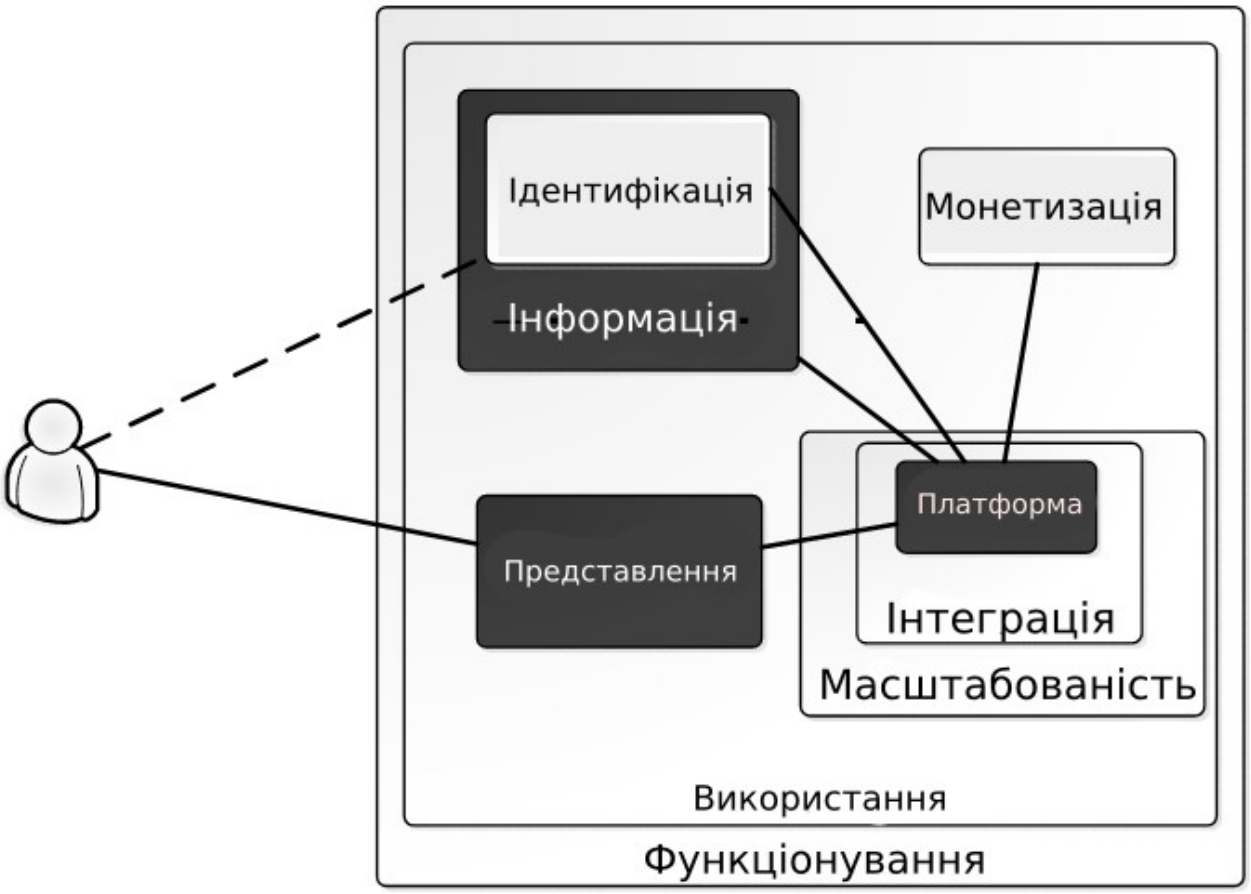


Рисунок 1.6 — Модель компонентів хмарних додатків

Тому можна бути точно впевненим, в тому, що відповідальності за безпеку майже ніхто не несе.

Користувачу потрібно довіритись в питаннях безпеки [36], продуктивності та якості обслуговування на звичайні обіцянки від постачальника даних послуг.

Також, зрозуміло, що використання ХО пов’язано з високими ризиками безпеки та конфіденційності даних, і це пов’язано з необхідністю завантаження і отримання даних з хмари та зберіганням даних на віддалених сховищ, тому що немає гарантії, що не виникне спроба несанкціонованого доступу до даних.

1.3 Проблеми безпеки систем розподіленого обчислення та їх способи вирішення

Не дивлячись на багато користі, яку можливо отримати від використання моделі ХО, у неї є ще достатньо відкритих проблем, які впливають на її розпоширеність. До прикладу, блокування постачальника інформації [24], ізоляція та багатокористування, керування даними, управління про рівень обслуговування та основна безпека конфіденційності даних.

Чому основною проблемою являється все-таки безпека конфіденційності даних:

1) управління безпекою передається третій стороні на аутсорсинг, яка надає свої обчислювальні ресурси;

2) перебування ресурсів різних орендодавців в одному й тому ж самому місці і використання того ж екземпляра послуг, без розуміння контролю безпеки, який використовується;

3) відсутність гарантій безпеки про рівень обслуговування між постачальниками цих послуг та їх користувачами;

4) розміщення набору конфіденційних даних в загальному доступі, де збільшуться ризик виникнення атак.

З точки зору постачальників послуг, безпека потребує великих витрат, ресурсів та являється досить складною проблемою для освоєння. Тому постачальники послуг повинні розуміти проблеми користувачів та шукати нові рішення забезпечення безпеки конфіденційності, котрі вирішують дані проблеми.

В моделі ХО основними характеристиками є багатокористуваність та гнучкість системи, тобто масштабованість, але ж вони і мають серйозний вплив на безпеку моделі. Наприклад, масштабування ресурсів користувача дає змогу іншим користувачам задіювати раніше виділені клієнту ресурси, що може викликати проблеми конфіденційності. Або ж, наприклад, масштабованість має

механізм розміщення послуг, який веде список доступних ресурсів з запропонованого пулу ресурсів, що теж повинен включати у собі засоби безпеки конфіденційності користувачів, такі як запобігання розміщення послуг на одному сервері і дані повинні зберігатись у межах користувацьких границь.

Модель ХО залежить від глибокого стеку взаємопов’язаних шарів об’єктів, де забезпечення безпеки та функціонування більш високого рівня залежить від нижніх, чим і ускладнює проблему забезпечення безпеки. Крім того, будь-яке порушення даних об’єктів може вплинути на безпеку усієї платформи. Тому кожен рівень має свій набір вимог для забезпечення безпеки, які повинні супроводжуватись набором засобів контролю безпеки, й це призводить до того, що стає надто велика кількість засобів контролю, якими необхідно керувати. В свою чергу, це призводить управління великої кількості різнородних засобів контролю, яке стає надто складним, через конфлікти між вимогами безпеки на кожному рівні.

Через те що постачальники послуг не знають про архітектуру розміщених сервісів, а також через зіштовхуються з великою кількістю змін в вимогах безпеки в процесі експлуатації різних засобів контролю, вони не можуть забезпечити ефективний та дійсний контроль для забезпечення безпеки. Тому прозорість, безпека, порушення та ризики, які можуть виникнути, повинні існувати серед користувачів та постачальниками послуг, тобто користувачі довіряють постачальникам, а інші в свою чергу роблять все для того, щоб користувачі мали змогу перевіряти та контролювати дотримання вимог безпеки.

Проблеми IaaS полягають у безпеці віртуальних машин, захисті репозиторія образів віртуальних машин, безпеці віртуальної мережі та безпеці гіпервізора.

Під безпекою віртуальних машин, мається на увазі, захист ОС і робочих навантажень від розпоширення загроз безпеки, такі як віруси чи шкідливі програми, за допомогою традиційних чи хмарних засобів безпеки.

Образи віртуальних машин можуть бути скомпрометовані шляхом введення шкідливих кодів в файл віртуальної машини, тому що віртуальні машини залишаються вразливими, навіть коли вони знаходяться в автономному режимі.

Через спільне користування мережевою інфраструктурою різними клієнтами на одному сервері чи в фізичних мережах може збільшити можливість використання вразливостей в DNS-серверах, DHCP чи IP-протокола, що приводять до мережевих атак на віртуальні машини.

Так як гіпервізор, це основний контролер будь-якого доступу віртуальних машин до ресурсів фізичного сервера, то будь-яка компрометація гіпервізора порушує безпеку віртуальних машин, тому що будь-яка операція віртуальних машин є незашифрованими.

Проблеми з PaaS виникають через використання моделі, яка створена на базі сервісно-орієнтованої архітектури (SOA), тому що призводить до унаслідування тих самих проблем безпеки, які існують в SOA, тобто Ddos-атаки, атаки “людина по середині”, атаки-ін’єкції, пов’язані з XML та пов’язані з перевіркою аутентифікації. Також виникають проблеми з API, тому що вони повинні бути оснащені засобами контролю безпеки, для забезпечення правильної аутентифікації та авторизації при викликах API.

Проблеми SaaS полягають у тому ж самому, що і в IaaS та PaaS, тому що вона побудована поверх них, включаючи керування безпекою даних, тобто цілісність, сегрегацію, доступність, конфіденційність, резервне копіювання, та мережеву безпеку. Ще існують проблеми в скануванні вразливостей веб-додатків, які розміщені в хмарній інфраструктурі, і повинні бути перевірені на наявність вразливостей за допомогою сканерів веб-додатків, а також помилка конфігурації безпеки веб-додатків, тому що кожен користувач має свої власні конфігурації безпеки, які можуть конфліктувати один з одним, що спричинюють проблеми з безпекою.

Проблеми безпеки пов’язані з керуванням хмари, через важливість даного рівня, тому що будь-яка вразливість чи порушення цього рівня приведе до того, що зловмисник матиме контроль над усією платформою як адміністратор.

Проблеми безпеки методів хмарного доступу з’являються через те, що ХО основані на поширені ресурсів через Інтернет, тобто доступ до даних ресурсів можна отримати через веб-браузери (HTTP/HTTPS), у випадку веб-додатків на рівні SaaS, через протоколи REST, RPC та SOAP, у випадку веб-сервісів та API на рівні PaaS, віддалені підключення, VPN та FTP для віртуальних машин і сервісів зберігання інформації на рівні IaaS.

Існують різні способи забезпечення безпеки для ХО:

1) керування ключами;
2) керування ідентифікацією та доступом;
3) керування безпекою;
4) забезпечення безпечного життєвого циклу розробки ПЗ;
5) забезпечення оптимізації компромісу між безпекою та продуктивністю;
6) об’єднання безпеки серед мультихмарних платформ.

Ідентифікація є одним з найважливіших компонент будь-якої системи з забезпеченням безпеки, це дозволяє системам розпізнавати користувачів, сервісів, серверів, тощо. Даний компонент складається з набору якихось даних, які пов’язані з певною сутністю, вона не повинна розкривати особисту інформацію користувача, тобто конфіденційну інформацію. Хмарні платформи повинні надавати чи підтримувати надійну систему управління ідентифікацією, котра повинна покривати всіх користувачів та усі об’єкти, які відповідають ідентифікаційно контекстній інформації.

Життєвий цикл безпечної розробки ПЗ включає в собі усунення вимог безпеки, моделювання ризиків, доповнення вимог безпеки до системних моделей і також до згенерованого коду. PaaS надає набір багаторазових

компонентів безпеки, які допомагають розробляти захищені хмарні додатки. Ці додатки повинні підтримувати адаптивну безпеку, щоб відповідати великому спектру вимог безпеки користувачів. Дана адаптивна безпека створена на делегуванні забезпечення безпеки та керування безпекою додатків для керування хмарною безпекою.

Модель ХО заснована на наданні послуг з використанням угод про рівень обслуговування, вони повинні захоплювати всі цілі пов’язані з продуктивністю, надійності та безпекою. Керування повинно враховувати компроміс між безпекою та продуктивністю за допомогою утиліти з функцією безпеки та продуктивності, крім того потрібно зосередитись на забезпечені адаптивної безпеки, де конфігурація контроля безпеки основані на поточному та очікуємому рівні ризику.

Об’єднання безпеки серед мультихмарних платформ має на увазі, підтримку безпеки як в платформах, так і між ними, через те що користувач використовує додатки, які залежать від послуг з різних платформ. До прикладу, коли декілька платформ інтегруються між собою заради надання більшого пулу ресурсів чи послуг, то вимоги до безпеки повинні бути об’єднані і застосовані у різних платформах.

Управління безпекою полягає в тому, що грунтуючись на великій кількості зацікавлених сторін, воно повинно включати вимоги та специфікації політики конфіденційності, конфігурації засобів контроля безпеки в залежності від цієї політики конфіденційності.

Конфіденційність є однією з ключових цілей з безпеки ХО і управління ключами дозволяє вирішити її. Тому що завдяки шифруванню, яке являється основним вирішенням проблеми забезпечення конфіденційності даних та процесів. Алгоритми симетричного та асиметричного шифрування основані на ключі, ці обидва підходи до шифрування мають проблему, пов’язану з керуванням ключами шифрування, тобто як їх правильно генерувати, зберігати, отримувати доступ чи обмінюватись приватними ключами. Крім того, PaaS

потребує ключі для викликів сервісів з інших додатків та API, дані ключі повинні зберігатись безпечно разом зі всіма іншими обліковими даними, які так необхідні додаткам для доступу з API.

1.4 Постановка задачі

Проаналізувати та реалізувати рішення застосування методів та систем розподілених обчислень із використанням асиметричних методів шифрування. Тобто провести аналіз СРО та методів РМН, дослідити відомі існуючі способи запобіганню витоку конфіденційної інформації за допомогою шифрування, застосувати реалізовану СРО.

Проаналізувавши відомі математичні моделі СРО, більш поглибленно парадигму ХО, з якою в подальшому і буду працювати, та виявивши основні переваги та недоліки, а саме проблеми забезпечення безпеки конфіденційності даних в ХО, і їх відомі методи для вирішення данної проблеми, потрібно за допомогою отриманої інформації дослідити та застосувати обрану СРО із застосуванням арифметичних алгоритмів шифрування, для забезпечення перешкод для витоку конфіденційних даних користувачів.

1.5 Висновок

Підсумовуючи, можна сказати що модель ХО є досить однією з найперспективніших обчислювальних моделей, але для того щоб застосовувати ХО, потрібно розібратись з проблемами забезпечення безпеки. Маючи на увазі ті деталі, які були описані в даному розділі, потрібно завжди враховувати ризики, які пов’язані з використанням ХО. А також, потрібно думати про те, які дані використовувати на базі ХО, а які дані краще ніколи не виводити за рамки локальної мережі. Тому як я вважаю, для забезпечення гарантій безпеки та конфіденційності даних у своїй роботі, я буду намагатись пов’язати СРО на базі ХО з асиметричними алгоритмами шифрування.

2 МЕТОДИ ЗАСТОСУВАННЯ РОЗПОДІЛЕНОГО МАШИННОГО НАВЧАННЯ В СИСТЕМАХ РОЗПОДІЛЕНОГО ОБЧИСЛЕННЯ З ВИКОРИСТАННЯМ АСИМЕТРИЧНИХ АЛГОРИТМІВ ШИФРУВАННЯ

2.1 Огляд та поняття розподіленого машинного навчання

В науковому колі, в основному в ІТ-індустрії, ШІ давно вже став одною з найпопулярніших тем для обговорення. В реальному житті людей ШІ допомагає вирішити, загалом, купу різних проблем, до прикладу таких як розпізнавання обличчя, сканер відбитку пальців, навігація, керування беспілотниками. Тому виходячи з цих проблем, дослідження та поглиблинне вивчення області ШІ мають не тільки достатню теоритичну цінність, а й велике практичне застосування у реальному житті.

Візьмемо, до прикладу, МН [1], яка є однією з основних та фундаментальних технологій ШІ, та створена для того щоб забезпечити благополуччя людей в новому интелектуально розвинутому світі, через удосконалення комп’ютерів та мереж. Перелічені вище проблеми вирішуються також за допомогою МН.

На жаль традиційна технологія МН є недостатньо ефективною через те, що вона не здатна добре працювати з великою кількістю великих та різноманітних даних [2], але для того щоб вирішити дану проблему, компаніями було створено різні науково-дослідницькі інститути МН, які спеціалізуються на великих даних та ШІ, для проведення в подальшому досліджень в області РМН [3]. В багатьох областях, до прикладу медицина та фінанси, збір великої кільскості даних являється дуже тяжким та завжди потребує обміну даними між декількома постачальниками даних. Отже, потребується збір даних між багатьма компаніями (організаціями), які можуть знаходитись у різних точках світу, і коли дані є конфіденційними чи приватними, вони стають надто вразливими для викрадення зловмисниками.

Технології РМН в основному мають у своєму розпорядженні систему на основі MapReduce, серверну систему параметрів, а також систему абстракції на основі графової моделі. Основними системами на основі MapReduce являються Hdoop [8] та Spark [10], але ці системи є достатньо повільними, на відміну від розподілених алгоритмів які були спеціально розроблені для МН. Ось, наприклад, Graphal та Pregel, які входять до систем абстракції на основі графових моделей, не тільки застосовують найкращі алгоритми МН, а й мають гнучке планування обчислень, але сама система є куди складнішою, ніж інші. Переходячи до систем з архітектурою серверної системи параметрів [16], можна сказати, що вони є кращими в плані адаптованності та ефективності щодо паралельної стратегії декількох алгоритмів МН.

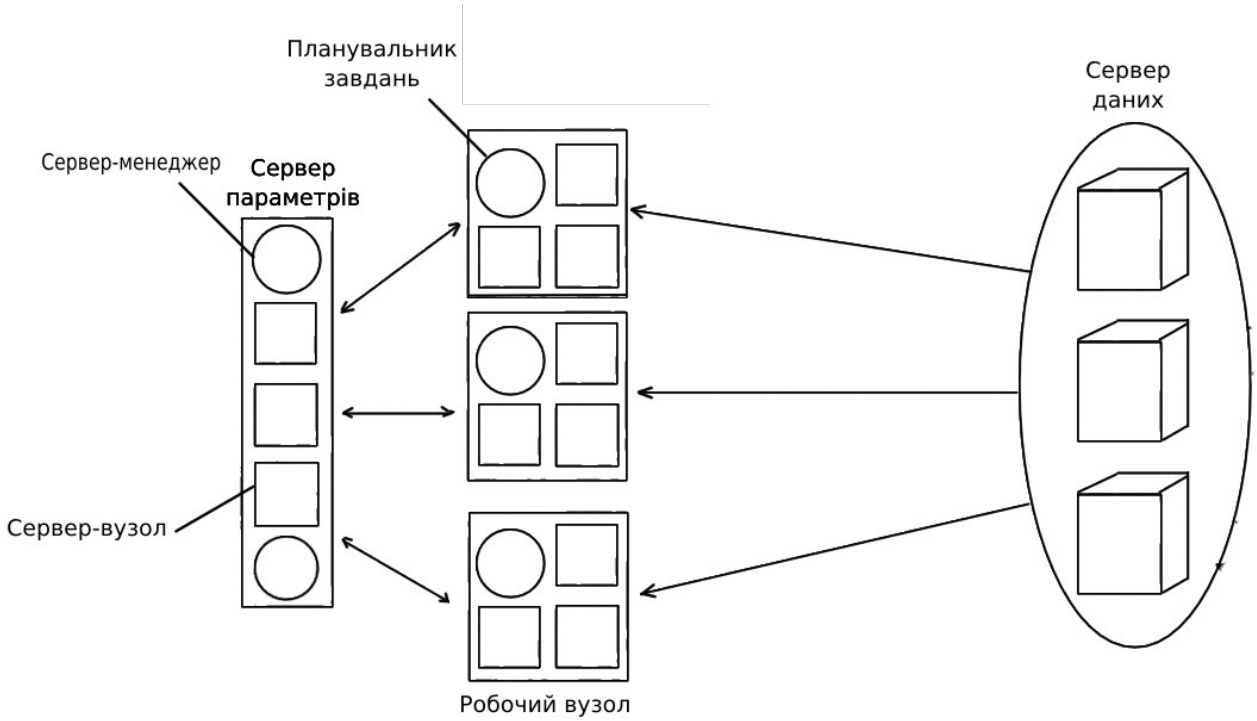


Рисунок 2.1 — Системна модель сервера параметрів

Основною для РМН являється система параметрів серверу [7], яка зображена на рисунку 2.1, через те що :

1) декілька робочих вузлів виконують обрахунки паралельно, тобто одночасно, задля того щоб забезпечити високу паралельну ефективність системи;

2) навчальні дані зберігаються на сервері даних, задля того щоб зменшити витрати робочих вузлів на локальне середовище.

Але як кажуть, не все є вічним, й не все є еталонном, тому і тут не обійшлось без недоліків. Головним є те, що в даній системі не враховується цілісніть даних на сервері даних. Тобто якщо якісь зловмисники забажають змінити, підмінити, або ж знищити дані, то це призведе певної зміни в моделі навчання, що в подальшому вплине на кінцевий результат навчання і він буде неправильним. То проблема не тільки даної системи, а й багатьох систем, особливо якщо вони працюють з хмарним середовищем, тому потрібно обов’язково захищати цілісність та конфіденційність даних, що в подальшому я буду описувати.

РМН як децентралізована теорія МН [16] дозволяє проводити розподілене навчання в великих масштабах НД на периферійних пристроях, де ні один вузол не здатний отримати повноцінне інтелектуальне рішення з масиву НД протягом певного проміжку часу. На жаль, через зростання власників розподілених даних гарантія безпеки НД від окремих власників даних стає достатньо низькою. Саме така проблема відсутності безпеки [42] збільшує ризик того, що зловмисники проведуть маніпуляції з НД, від якого залежить проміжні результати навчання. Тому, таким чином, це впливає й на цілісність даних, які є ключивим компонентом в навчанні моделей МН. Атаки на дані є одними з найрозповсюдженіших та ефективних, способів пошкодження моделей МН на етапі навчання.

2.2 Аналіз відомих методів та засобів забезпечення захисту витоку конфіденційних даних

Так як мобільні пристрої крок за кроком займають ключову нішу в розпізнавані різної активності за допомогою датчиків, це означає що ще більше КД стають доступними. Але великомасштабний збір КД тягне за собою і великі ризики.

Звісно ж, на даний момент існує чимало усіляких різних алгоритмів шифрування та стратегій захисту для вирішення цієї проблеми, які запевняють про гарантію цілісності та конфіденційності даних, але й в той самий час не заперечують факт, що при наявності у когось секретного ключа, він зможе отримати доступ до цих самих даних.

При широкому використані МН, а особливо централізованого МН, для того щоб навчити модель дані повинні збиратись та відправлятись у центральну репозиторій, який може стати точкою відмови системи, тому ці дані неминуче зіткнуться із ризиком витоку даних. Дивлячись на це все, виконання МН на НД без ризику їх витоку являється ключовим питанням для обміну інформацією.

МН [5] з багатостороннім захистом конфіденційності, яке засноване на захисті конфіденційності, може надати допомогу користувачам навчатися разом за допомогою даних які надають один одному гарантом якої виступає забезпечення безпеки своїх власних даних. Наприклад, проблему конфіденційності в багатосторонніх обчисленнях може допомогти вирішити федеративне навчання.

В епоху великих даних конфіденційність стала однією із найважливіших фундаментальних проблем для масивних структурованих і неструктурованих даних, що генеруються у різних інтелектуальних додатках, які надають велике значення агрегації даних і комбінування їх на різних вузлах.

Існують підходи які використовують різноманітні методи для вирішення цих проблем, основними методами [29] з них являються безпечні багатосторонні обчислення (MPC) та дифференціальна конфіденційність.

MPC переважно являється основним варіантом оптимізації обчислювальної функції над спільно розподіленими ресурсами даних з використанням криптографічних примітивів:

1) забудькувата передача — протокол передачі даних, в якому передавач передає по одній можливій частині інформації отримувачу, але не запам’ятовує , що було передано, а що ні;
2) гомоморфне шифрування;
3) сумісного використання секретів

Більшість примітивів MPC для задовільнення потреб безпеки створені, по своїй суті, на прогресі в області повністю ГШ (FHE). Данний момент дозволяє постачальникам даних шифрувати окремі НД за допомогою відкритого ключа і передавати обчислення на аутсорсинг якомусь хмарному сервісу. Очевидним фактом є те, що хмарні технології мають перевагу у реагуванні змін обчислювальних потреб та скороченні витрат на IT-інфраструктуру. За такого розкладу, повинна бути можливість для виконання обчислень над зашифрованими даними без їх дешифрування, і якраз такою характеристикою володіє ГШ, тобто вона є моделю шифрування, що може виконувати деякі математичні дії з зашифрованими НД і отримувати такий зашифрований результат, який відповідає результатам схожої операції, що проводиться з відкритими даними. Тому, якщо підсумувати, хмарні сервери використовують також технологію обчислення зашифрованого НД, але при відсутності секретного ключа хмарний сервер виконує здебільшого функцію обчислювальної платформи, яка на жаль не може отримати доступ ні до одного із окремих записів.

На сьогоднішній день ставиться акцент [45] на досягненні ефективного та реалістичного РМН, яке включає в себе застосування примітивів MPC, і

навіть в деяких областях де застосовуються вони продемонстровані масштабуванням кількома сотень мільйонів записів в навчальних завданнях. Навіть при тому що МРС має достатньо переваг, питання повноцінного захисту конфіденційної інформації залишається відкритим.

В деяких випадках хмарні сервери можуть бути як не зовсім прозорими, так і навіть інколи шкідливим, коли через нього може поширюватись якесь шкідливе ПЗ, або “злити” конфіденційну інформацію людини без її згоди на те, з цілю отриманння якогось прибутку, переваги, тощо. Для зовнішніх безпечних багатосторонніх обчислень дана проблема стає нетривіальною проблемою.

ДК застосовується для введення шуму в проміжні результати, щоб уникнути витоку КД про якийсь конкретний запис. Але для того щоб збалансувати такі характеристики моделі як її зручність та конфіденційність, потрібно пожертвувати часом, через ретельну та достатньо точну калібровку статистичного шуму. Окремою проблемою в використанні підходів пов’язаних з ДК на класифікаторі, є те, що набір навчальних даних має захист тільки в процесі навчання. ДК в своїй архітектурі містить емпіричну мінімізацію ризиків (ERM), яка відіграє одну з головних ролей, тому що вона може охоплювати більшість з проблем МН. Легко отримати диференціальні алгоритми можливо, якщо відомо як реалізовувати ERM приватному порядку (DP-ERM), при реалізації задач МН, тобто при класифікації та регресії. В даному підході DP-ERM повинен отримати якісь проміжні результати, при незначній зміні вхідного НД. Існують три методи диференціально приватної оптимізації DP-ERM:

1) Objective Perturbation (OP);
2) Output Perturbation (OuP);
3) Gradient Perturbation (GP).

Дані підходи можуть забезпечити збереження конфіденційності в централізованій області з окремим суб’єктом, який володіє всім НД.

Ще одним із методів забезпечення захисту витоку КД є схема перевірки цілісності Provable Data Possesion (PDP), яка досить успішно застосовується в

хмарних середовищах. Вона використовує аудит вибірки, при якому не потрібно завантажувати всі дані при перевірці на цілісність даних, тому витрати на зв’язок значно зменшуються. Першими хто запропонував PDP-схему з гомоморфним тегом на основі RSA були компанія Atenies, а також потім схему динамічного аудиту, але вона була лише частково динамічною і не забезпечувала захист КД. Після чого наступною компанією Erway була створена повна динамічна схема PDP.

PDP, як і інші методи, має проблему з захистом КД, яку на протязі довгого часу намагились вирішити багато людей. Було запропоновано безліч схем публічного аудиту, які зберігають конфіденційність, на основі PDP. Схеми, які були запропоновані протягом усього часу:

1) із застосуванням гомоморфного аутентифікатора з випадковою технікою маскування;

2) державного аудиту, зберігаючу конфіденційність, для підтримки динамічної роботи з даними;

3) державного аудиту, зберігаючу конфіденційність, з методом випадкового маскування, щоб приховати дані відповіді;

4) із застосуванням ідеального збереження КД для віддаленого аудиту і новою конструкцією протокола на основі ідентифікаторів з гомоморфним криптографічним примітивом;

5) з віддаленою перевіркою володіння даними (RDPC), яка могла б витримати підробку та повторну атаку, але вона не підтримувала захист конфіденційності;

6) з структурою даних аутентифікації за допомогою D&CT (в ній були введені логічний нижній індекс (LI) та номер версії (VN) блоку даних для професійної підтримки динамічних даних для нормальних розмірів файлів, а також вони застосували повномасштабне сховище даних, для того щоб зменшити витрати на обчислення та хмарні сервери);

7) публічного аудиту без громадянства з аутентифікованим списком пропусків на основі рангу;

8) державного аудиту з призначеним верифікатором (власник даних може призначити одну довірену особу для перевірки даних, але вона також не підтримувала захист конфіденційності).

Більшість схем PDP основані на інфраструктурі відкритих ключів (PKI), в якій є центр створення ключів (KGC) необхідний для управління та збереження всіх користувацьких пар відкритий/приватний ключ, тобто KGC може контролювати всі ці пари, що може викликати проблему умовного депонування ключів. Щоб вирішити проблему умовного депонування ключів та зменшити витрати на управління сертифікатами було знову запропоновано безліч схем:

1) розоділену схему володіння даними на основі ідентифікації в багатохмарному сховищі для реалізації перевірки цілісності даних;

2) проксі-орієнтоване завантаження даних та модель перевірки цілісності даних з криптографією відкритого ключа на основі ідентифікації;

3) публічного аудиту на основі ідентифікаторів в багатокористувацькому сценарії (цій схемі не вдалось вирішити проблему умовного депонування ключів та забезпечити безпеку конфіденційності порта);

4) віддаленого аудиту цілісності даних з ідеальним збереженням конфіденційності даних за допомогою криптосистем на основі ідентифікації;

5) безсертифікатна схема перевірки публічної доброчесності (була випущена схема управління сертифікатами та вирішена проблема умовного депонування ключів, прийнявши безсертифікаційний підпис та закритий ключ із двох частин, але вартість обчислень була відносно великою);

6) схема рішення проблеми обміну конфіденційною інформацією в процесі аудита цілісності даних, яка могла забезпечити безпечний обмін КД;

7) схему збереження даних з декількома копіями на основі ідентифікації, для того щоб знизити обчислювальні та комунікаційні витрати, викликані за рахунок PKI.

Також без уваги не залишимо, федеративне навчання, яке є некриптографічним підходом для навчанні моделей МН для забезпечення збереження конфіденційності, тому що останнім часом воно викликало великий інтерес. Дані залишаються під контролем своїх власників, а координує навчання сервер, відправляючи модель безпосередньо власникам тих даних, які в свій час оновлюють модель підставляючи власні дані. Оновлені моделі від декількох учасників усереднюються для отримання глобальної мооделі. Для забезпечення конфіденційності в даному випадку, дехто покладається на диференціально приватний механізм для запутування проміжних значень. Але таке запутування приховує корисність даних та моделей, в той час коли навчання точних моделей потребує високих бюджетів конфіденційності, а сам рівень конфіденційності — невідомим.

2.3 Дослідження засобів забезпечення захисту витоку конфіденційних даних

Першими хто застосував метод ДК, який проводить агрегацію локально навчених класифікаторів з застосуванням OuP для ДК, і інтегрували у свої дослідження, були Pathak. Оскільки шум в моделі обернено пропорційний найменьшому НД з p-сторонами, вони покращили шкалу шуму в$\sqrt{p}$- разів з прийнятям безпечної агрегації моделі локально навченної класифікації, а потім виконати агрегацію моделі локально навченних моделей, забезпечуючи при цьому значну гарантію конфіденційності. Даний OuP-алгоритм здатний здійснити покращення протоколу Pathak, завдяки p-компоненту, після введення статичного шуму в захищену область МРС на основі потрібної шкали шуму, яка обернено пропорційна об’єму всього НД.

Ось, до прикладу, на рисунку 2.2 зображено базовий конвеєр для безпечного навчання розподілених моделей МН, де безпечний багатопартійний фреймворк МН представляє собою кількість постачальників даних p+1 та хмарного серверу $\{\varnothing_1, \varnothing_2, \ldots, \varnothing_p\}$. В такій архітектурі, кожен постачальник даних буде повинен підвищити точність глобально навченої моделі без витоку якої-небудь конфіденційоної інформації про локально конфіденційний НД.

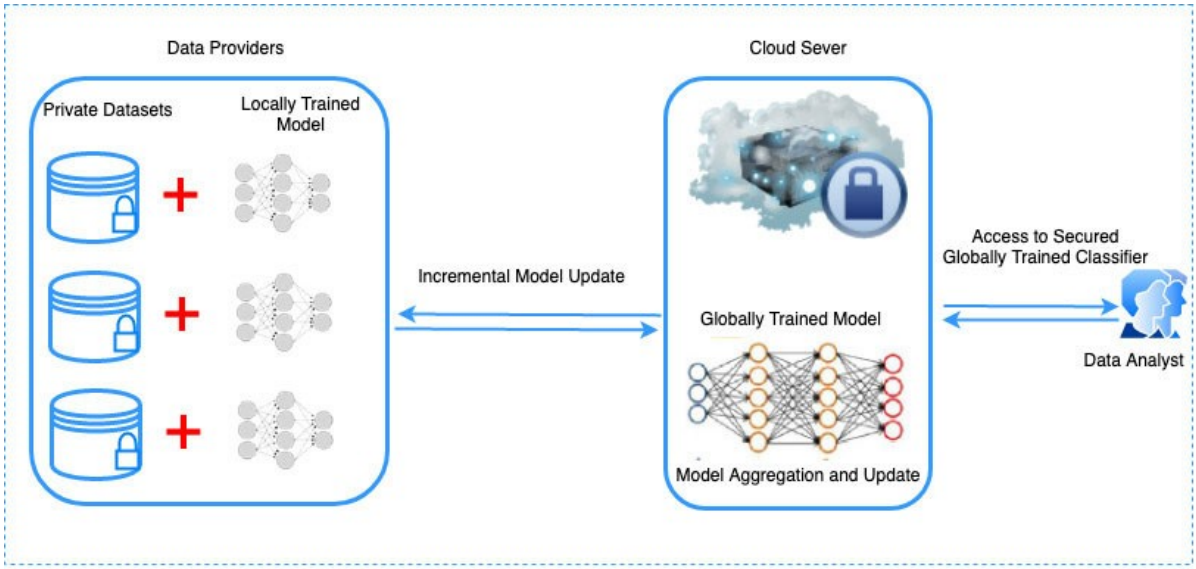


Рисунок 2.2 — Базовий конфеєр для безпечного розподіленого МН [28]

Для того щоб вирішити три фундаментально важливі проблеми в даному протоколі, потрібно ввести диференціально розподілені алгоритми МН. Тобто, буде застосовано як і OuP, так і GP з ін’єкцією статистичного шуму в захищену багатопартійну обчислювальну область.

Стратегія полягає в тому, щоб продемонструвати OuP-алгоритм, який надійно проводить агрегацію локально навченої моделі, яка шифрується за допомогою різноманітних примітивів шифрування, чи відмінними ключами для досягнення ε-ДК з додаванням статистичного шуму Лапласа до параметрів агрегованих класифікатором. За допомогою цієї схеми GP, окремі власники даних разом виконують адаптивно ітеративний протокол на основі градієнта, де вони надійно проводять агрегацію локальних градієнтів на кожній ітерації з

долею ε_t від загальної конфіденційності ε, тобто забезпечення (ε,δ)-ДК з застосуванням адаптивного градієнтного спуску для нульового концентрованого ДК (zCDP).

Даний протокол дає гарантію конфіденційності в більш чесному середовищі загроз безпеки, базованому на криптосистемі та примітивах, зберігаючих конфіденційність. В цьому середовищі зловмисники не можуть отримати нормальної можливості для доступу до конфіденційних даних, оскільки окремі моделі шифруються по усьому світі. Отже, він забезпечує більш практичний та ефективний примітив, що зберігає конфіденційність, в сфері МН.

Існуючі криптографічні розподілені рішення практичні тільки з невеликою кількістю сторін, і більшість раніше поданих рішень зосереджені або на навчанні, або на прогнозуванні. Вони не враховують повний робочий процес РМН і не дозволяють навчати модель, яка залишається секретною та дає змогу забувати передбачати КД. В багатьох випадках, навченна модель настільки ж конфіденційна, як і дані, по яким вона навчена, и застосування моделі після навчання повинно бути жорстко контрольованим.

2.4 Аналіз асиметричного алгоритму шифрування

Криптографія - це метод перетворення даних у зашифрований формат, щоб лише авторизовані користувачі мали доступ до інформації. Існує два основних типи шифрування: симетричне та асиметричне [6].

Метод симетричного шифрування містить один криптографічний ключ для шифрування та дешифрування даних, відомого як "симетричний ключ", яким володіють обидві сторони. Цей ключ застосовується для кодування та декодування інформації. Відправник використовує цей ключ перед надсиланням повідомлення, а приймач використовує його для розшифрування кодованого повідомлення. Використання одного ключа для обох операцій робить процес

простим. Коли справа доходить до передачі величезних даних, симетричні ключі є кращими. Найпопулярнішим прикладом симетричного шифрування є «шифр Цезаря».

Сучасні методи шифрування засновані на дуже складних математичних функціях, які майже неможливо розкрити. Існують сотні симетричних алгоритмів, але AES, DES і 3DES є найпоширенішими з них.

Асиметричне шифрування [33] — це відносно новий і складний режим шифрування. Асиметричне шифрування включає кілька ключів математично пов’язаних один з одним для шифрування та дешифрування даних. Один з ключів відомий як «відкритий ключ», а інший – «приватний ключ». У цьому методі для шифрування даних використовується відкритий ключ, який є загальнодоступним, а розшифровка даних виконується за допомогою приватного ключа, що забезпечує захист даних від атак. При цьому використовується певний алгоритм. Оскільки, приватний ключ, який знаходиться у розпорядженні одержувача, розшифровує його. Один і той же алгоритм стоїть за обома цими процесами. Залучення двох ключів робить асиметричне шифрування складною технікою. Таким чином, воно виявляється масовим вигідним з точки зору безпеки даних. Існує декілька основних типів асиметричного шифрування: алгоритми RSA, алгоритм Deffie-Hellman і ECC.

У 1977 році асиметричний алгоритм шифрування RSA був винайдений трьома вченими з Массачусетського технологічного інституту: Роном Рівестом, Аді Шаміром і Леонардом Адлеманом. Цей метод ефективний, оскільки два різні випадкові прості числа заданого розміру вибираються і перемножуються, щоб створити ще одне гігантське число. Завдання — визначити вихідні прості числа з помноженого гіганта. Виявляється, цю головоломку практично неможливо розгадати для сучасних суперкомп’ютерів, не кажучи вже про людей.

У 1985 році два математики Ніл Кобліц і Віктор Міллер запропонували використання еліптичних кривих у криптографії. Майже через два десятиліття

їхня ідея втілилася в життя і в 2004-2005 роках почав використовуватися алгоритм ECC (Elliptic Curve Cryptography). У процесі шифрування ECC еліптична крива представляє набір точок, які представляють математичне рівняння $y^2=x^3+ax+b$. Число, яке символізує точку на кривій, множиться на інше число і дає іншу точку на кривій, так що вам потрібно знайти нову точку на кривій, щоб розірвати цю загадку. Він побудований таким чином, що знайти нову точку практично неможливо, навіть якщо ви знаєте початкову. В основному застосовується в малих системах, вона є більш компактною та вимагає менше обчислювальних ресурсів.

Асиметричний алгоритм є більш безпечним завдяки створенню пари ключів, але він також має вразливості. Одним із способів усунути ці вразливості є використання динамічного ключа, ключа, який змінюється з кожною ітерацією. Виходячи з цього, виникає проблема пошуку безпечного алгоритму шифрування після аналізу існуючих критеріїв.

Існує також гібридний метод шифрування, який включає симетричні та асиметричні. Ідея гібридного шифрування народилася, коли стало критичним шифрувати дані на високій швидкості, забезпечуючи перевірку ідентичності. Метод гібридного шифрування використовується в сертифікатах SSL/TLS під час послідовного зв’язку між серверами та клієнтами в процесі, відомому як «TLS handshake». По-перше, ідентичність обох сторін перевіряється за допомогою приватного та відкритого ключів. Після цього дані шифруються за допомогою симетричного шифрування за допомогою ефемерного ключа. Це забезпечує швидку передачу великих обсягів даних, які ми щохвилино надсилаємо та отримуємо в Інтернеті.

Кожен з алгоритмів шифрування має плюси і мінуси, але більшість сучасних сертифікатів SSL використовує гібридний метод: асиметричне шифрування для аутентифікації та симетричне шифрування для конфіденційності.

Далі розгляну більше детально найбільш використовувані алгоритми асиметричного шифрування.

2.4.1 RSA алгоритм

Це був перший розроблений алгоритм у криптографії з відкритим ключем, і одне з перших великих досягнень у шифруванні відкритих ключів. Використовує великі прості числа при шифруванні. Також він застосовується при створенні захищеного каналу зв’язку за технологією SSL

Він включає в себе три етапи:

1) генерація ключів;
2) шифрування;
3) дешифрування.

А тепер розгляну детально ці етапи:

1. При генерації ключів RSA використовуються два ключі для процесу. Шифрування здійснюється за допомогою відкритого ключа приймача, а дешифрування здійснюється за допомогою приватного ключа отримувача. Щоб створити ключ, використовуються наступні кроки:

1) вибираються два різних і великих простих числа, кажуть, P і Q;
2) обчислюється N такі, що,$N = P * Q$;
3) обчислюється z таким, що, $z = (P-1) * (Q-1)$;
4) вибирається експонента публічного ключа: E така, що$1 < E < z$, а E та z не мають жодних спільних дільників, крім 1;
5) визначається D, який задовольняє співвідношенню $E * D = 1 (mod z)$;
6) E ділиться на найменші з серій:$z+1, 2z+1, 3z+1, 4z+1$,. тд. Тепер маємо відкритий ключ: (E, N) та приватний ключ: (D, N).

2. Шифрування - це процес перетворення звичайного тексту в шифрований. Цей процес вимагає двох речей: ключа та алгоритму

шифрування. Шифрування відбувається на стороні відправника, і використовується таке рівняння для шифрування повідомлення:

$C = M^{E} \bmod N$, де C - шифр тексту, а M - звичайний текст або повідомлення.

3. Дешифрування - це процес перетворення шифрованого тексту в звичайний текст. Цей процес вимагає двох речей: алгоритм дешифрування та ключ.

Дешифрування відбувається на стороні приймача, і для розшифрування повідомлення використовується таке рівняння: $M = C^{D} \bmod N$. Процеси шифрування та дешифрування показані на рисунку 2.3.

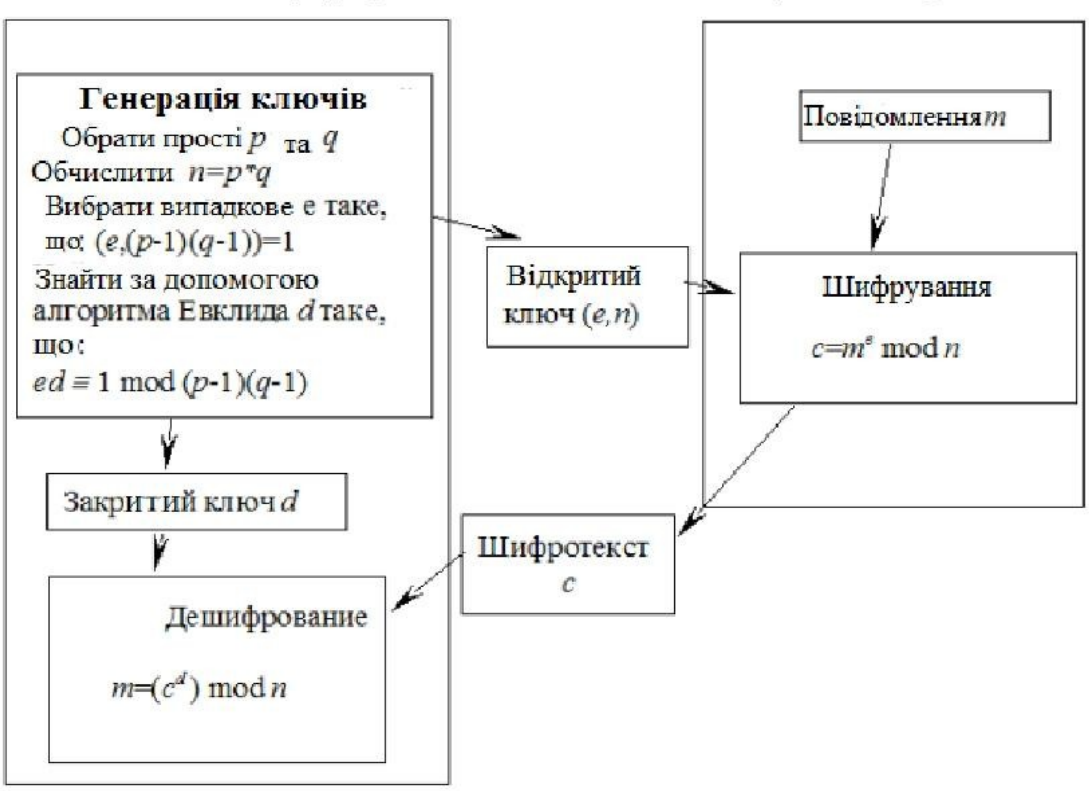


Рисунок 2.3 – Алгоритм RSA

2.4.2 Deffie-Hellman алгоритм

Алгоритм Deffie-Hellman [39] дозволяє обидвам сторонам, які не знають попередньо один про одного, взаємно встановлювати загальний секретний ключ

в небезпечному каналі. Обмін ключами Deffie-Hellman оснований на симетричній криптографії, оскільки загальний секретний ключ і ключ сеансу використовуються для шифрування та дешифрування. Особливість цього протоколу в тому, що числа A і B можна передавати по відкритому каналу. Єдиною вимогою до цього каналу є те, що зловмисник не повинен мати можливість щось міняти в каналі. Якщо ця умова виконана, то алгоритм Deffie-Hellman дозволяє встановити захищений канал зв'язку. В основному його використовують для безпечного пересилання ключів через глобальні мережі Алгоритм використовується багатьма протоколами, такими як SSL, Secure Shell та IPSec. Протокол дозволяє виробити загальний секретний ключ, не використовуючи секретний канал зв'язку.

Кроки [38] цього алгоритму такі:

1) виділяються два числа "p" (просте число) і "g";
2) вибираються два секретних числа "x" для відправника та "y" для приймача;
3) обчислюється загальнодоступне число $R_1 = g^{xmodp}$, і $R_2 = g^{ymodp}$;
4) здійснюється обмін ключами;
5) обчислюється перший сеансовий ключ як $K_s, K_s = R2^{xmodp}$;
6) обчислюється другий сеансовий ключ як K_r, $K_r - R_1^{ymodp}$;
7) тут $K_r = K_s = K$.

Відправник A і Приймач B хоче поділитися своїми секретними ключами через небезпечний канал.

Вони не діляться інформацією, вони просто діляться ключами. Основним недоліком цього алгоритму є те, що в цьому алгоритмі відбувається атака man-in-middle-attack.

Це відбувається під час обміну загальнодоступними числами, тобто R_1 і R_2.

Під час вторгнення змінюється значення R_1 і R_2 і передається нове обом сторонам. З цієї причини значення ключа сеансу стає нерівним.

2.5 Опис вибраного метода захисту цілісності навчальних даних в розподіленій системі МН

Для того щоб захистити цілісність навчальних даних в розподіленій системі МН, я вибрав за основу розроблену схему перевірки цілісності даних, яка орієнтована на МН (РМН-ПЦД), запропонованою групою науковців Південно-Східного університету Нанкіна. Дана схема являється першою в своєму роді, тобто схемою що використовує алгоритм аудиту публічної вибірки та цілісності навчальних даних в розподіленій області МН.

Основними принципами даної схеми є:

1) вона повинна забезпечити цілісність навчального НД, протидіяти підробці цих даних. Для цього ми спочатку використовуємо приватний ключ власника даних для структурування підпису цих даних, після чого застосовуємо метод вибірки та технологію PDP для створення доказів та використовуємо алгоритм білінійного відображення перевірки доказів;

2) вона може гарантувати забезпечення конфіденційнсті навчальних НД в процесі публічного аудиту вибірки. Сервер даних додає фактор спотворення до доказів аутентичності, після чого сервер даних зашифровує коефіцієнт спотворення і відпраляє докази та коефіцієнт спотворення в стороній аудитор. Завдяки, проблемі дискретного логарифма, сторонній аудитор, а також зловмисники, не можуть розшифрувати фактор спотворення, через що не можуть отримати доступ до навчальних даних;

3) вона може вирішити прооблему умовного депонування ключів та знизити витрати на керування сертифікатами безпеки, тому що в даній схемі використовується алгоритм криптографії на основі ідентифікації для створення пар відкритого/приватного ключа власника НД, для того щоб усі сутності могли перевірити відкритий ключ власника НД без сертифікатів. Для прикладу, центр генерації ключів генерує частковий довготривалий приватний ключ для власника НД, після чого він генерує повний приватний ключ на основі

отриманої частини, і центр генерації ключів вже не може визначити його приватний ключ за допомогою тієї частини яку згенерував;

4) вона може протистояти підробкам та маніпуляційним атакам спрямованим на ослаблення безпеки конфіденційності даних, може забезпечити захист конфіденційності в процесі перевірки стороннім аудитором та вирішити ключову проблему депонування ключів.

2.6 Основні математичні алгоритми, що застосовуються у даному дослідженні

2.6.1 Білінійне відображення

Нехай H_1 та H_2 — це дві мультплікативні циклічні групи великого простого порядку p, то спарювання даних груп — це їхнє білінійне відображення: $H_1 x H_1 \to H_2$. Він задовільняє наступні властивості:

Білінійність: $e(u^a, v^b) = e(u,v)^{ab}, \forall u, v \in H_1; a, b \in Z_p$.

Невиродженість: $\exists u, v = H_1$, то $e(u,v) \neq 1 \in H_1$.

Обчислювальність: $\forall u, v \in H_1$це є поліноміальний час алгоритму обчислення $e(u,v)$.

Безпека: через проблему в H_1 та H_2 складно вирахувати.

2.6.2 Проблема дискретного логарифму (ПДЛ)

Нехай $\alpha \in Z_c^{'}$, та H_1 — мультиплікативна циклічна група, відома q, $q^{\alpha} \in H_1$, для будь-якого поліноміального часу супротивника, перевега розв’язування значення α дуже незначна. Алгоритм ймовірності полономіального часу (ЙПЧ) X, який успішно вирішує ПДЛ: $Xdv^{DL} = Pr\left[X(g, g^{\alpha}) = \alpha : \alpha \in Z_c^{'}\right]$, котрий є незначним. Тому ймовірність визначається з випадкового вибору αв $Z_c^{'}$та випадкового вибору алгоритму X.

2.6.3 Проблема обчислювального алгоритму Діффі-Хеллмана (ОАДХ)

Нехай $\alpha \in Z_c'$, та H_1 — мультиплікативна циклічна група, враховуючи $C, C_a, C_ь \in H_1$, для будь-якого поліноміального часу супротивника: $C^{ab} \in H_1$. Алгоритм ЙПЧ X успішно вирішує задачу проблеми ОАДХ: $Xdv^{ОАДХ} = Pr\left[X(C, C^a, C^b) = C^{ab} : a, b \in Z_c'\right]$, котрий є незначним. Тому ймовірність визначається з випадкового вибору a, b в Z_c' та випадкового вибору алгоритму X.

2.6.4 Проблема спільно-обчислювального білінійного алгоритму Діффі-Хеллмана (СОАДХ)

Нехай $\alpha \in Z_c'$, та H_1, H_2 — дві мультиплікативні циклічні групи, враховуючи $C, C^a \in H_1, Q \in H_2$, для будь-якого поліноміального часу супротивника: $Q^a \in H_2$. Алгоритм ЙПЧ X успішно вирішує задачу проблеми СОАДХ: $Xdv^{СОАДХ} = Pr\left[X(C, C^a, Q) = Q^a : a \in Z_p'\right]$, котрий є незначним.

2.6.5 Модель випадкового прогнозування

Приклад 1:

Екзистенційна непідробність даних власника приватного ключа під час визначення безпеки виглядає наступним чином:

Налаштування: користувач (X) запускає алгоритм для входу і відправляє відкритий ключ зловмиснику (Y), й тим часом зберігає приватний ключ.

Запит: Y надається два запити до X.

1) Хеш-запит: Y відправляє запит на отримання хеш-значення на основі ідентичності не більше k_N разів на свій вибір $ID_1, \cdots, ID k_N \in \{0, 1\}'$. X вираховує I_{ID} та відправляє хеш-значення Y;

2) Запит для отримання приватного ключа: Y відправляє запит на отримання приватного ключа на основі ідентичності в більшості випадків на свій вибір $ID1, \cdots, IDk_M \in \{0, 1\}$'. X вираховує M_{ID} та відправляє значення приватного ключа Y.

Підсумок: Y виводить пару (M_{ID}, K_s, ID), де ID не є ні одним із $ID_1, \cdots, ID\,p_k$. Y перемагає, якщо (M_{ID}, Pk, ID) є валідними для входу.

Визначаю Xdv_X як ймовірність, що X виграє.

Приклад 2:

Захист конфіденційних навчальних даних в області безпеки визначається наступним чином:

Налаштування: користувач (X) запускає алгоритм налаштування для генерації публічного параметру $u \in H_1$, де H_1 — мультиплікативна циклічна група. Припущу, що $e(G_1(X), syCQ)$, є закритим ключем серверу даних, а $M = u^m$-відкритим ключем.

Запит: зловмисник (Y) робить запит на отримання відкритого ключа випадкового числа $l \in Z_c'$.

Підсумок: Y може отримати приватний ключ сервера даних m'.

Визначаю Xdv = Pr[m' = m] — 1/2. Y не може отримати дані, якщо функція Xdv є незначною для будь-якого поліноміального часу супротивника.

2.7 Висновок

В цьому розділі був виконаний теоретичний аналіз області ШІ, а саме МН, проблеми витоку КД, які передаються, при розподілених обчисленнях, розглянуті переваги та недоліки вже існуючих рішень забезпечення захисту та безпеки при передачі конфіденційних НД для навчання моделі РМН та сформована задача на їх виконання.

Відштовхуючись від існуючих рішень, зроблю висновок, що краще всього підійдуть СРО з використанням асиметричних алгоритмів шифрування

для забезпечення гарантії безпеки та захисту при передачі КД. Тому що асиметричні алгоритми шифрування дозволяють доволі безпечно обмінюватись інформацією, через застосування примітива пари відкритий/закритий (приватний) ключ, що в свою чергу робить достатньо важким отриманння будь-якої конфіденційної інформації без наявності закритого (приватного) ключа.

3 МОДЕЛЬ СИСТЕМИ РОЗПОДІЛЕНОГО МАШИННОГО НАВЧАННЯ НА ОСНОВІ СХЕМИ ПЕРЕВІРКИ ЦІЛІСНОСТІ ДАНИХ З ВИКОРИСТАННЯМ АСИМЕТРИЧНИХ АЛГОРИТМІВ ШИФРУВАННЯ

3.1 Модель системи РМН-ПЦД

Схема РМН-ПЦД складається з чотирьох компонентів: серверу даних, стороннього аудитора, власника НД та центру генерації ключів, які зображенні на рисунку 3.1.

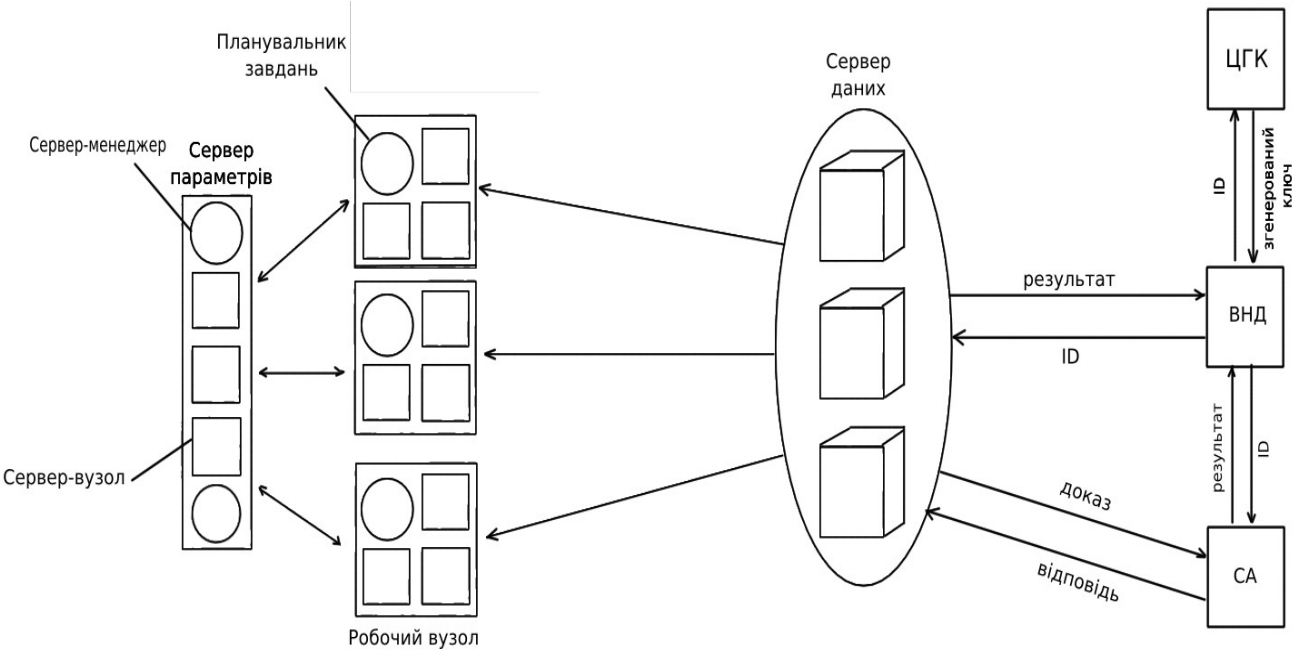


Рисунок 3.1 — Системна модель схеми РМН-ПЦД

Далі роз’ясню задачі та повноваження кожного компоненту даної системної моделі:

1) ВНД — відповідає за збір навчальних даних та їх завантаження на сервер даних, ці дані можуть приходити з різних пристроїів;

2) СД — відповідає за надання хмарних сервісів та зберігає навчальні дані, при прийомі запиту від СА, він повинен довести цілісність та збереженність НД, після чого генерує та відправляє назад відповідь;

3) СА — перевіряє цілісність навчального НД, які зберігаються на СД, як написано в характеристиці СД, за допомогою відправки запиту та прийому відповіді на рахунок цілісності цього НД, а також він може проводити публічний аудит за допомогою відкритого ключа власника НД;

4) ЦГК — генерує приватний та відкритий ключі, й після ідентифікації власника НД генерує частковий приватний ключ для нього з ідентифікатором та мастер-ключем, тобто ЦГК керує частковим ключем власника НД.

3.2 Приклад застосування даної системної моделі

Якщо брати в розрахунок дану системну модель, то дана схема РМН-ПЦД може бути застосована для прогнозування, рекомендації чи класифікації в розподілених алгоритмах МН.

Візьму до прикладу додаток, що надає рекомендації по рекламі, який прекрасно ілюструє роботу даної моделі, системна модель якого зображена на рисунку 3.2.

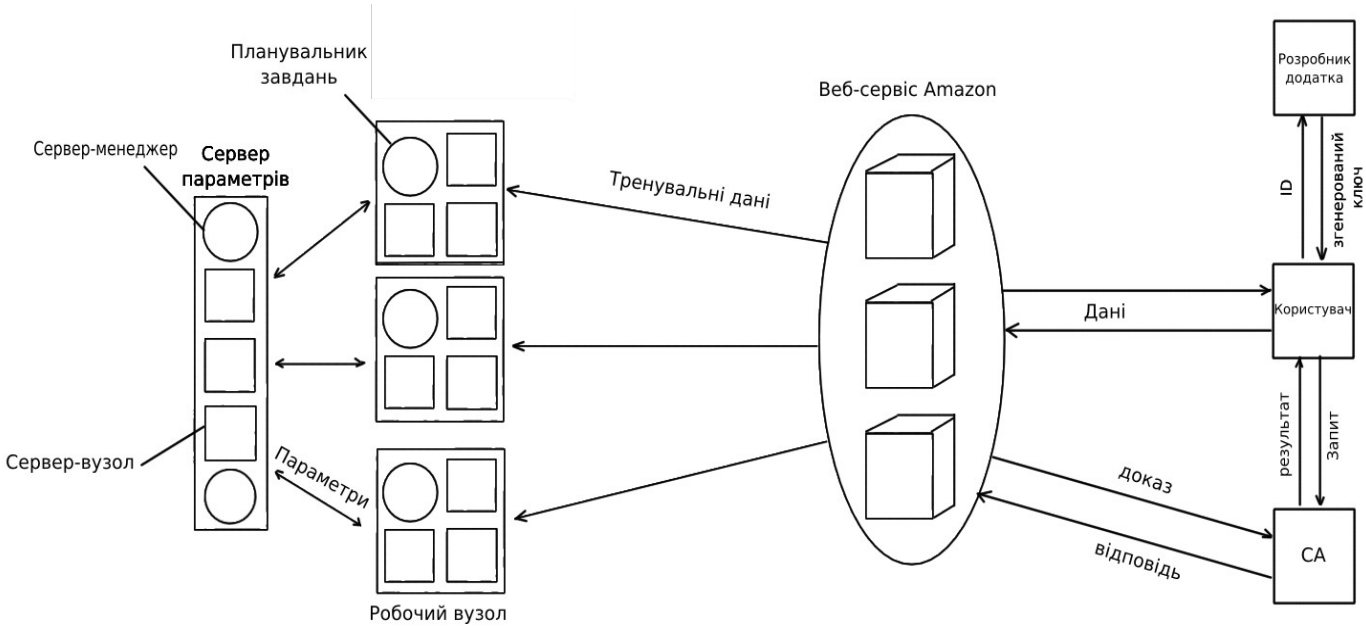


Рисунок 3.2 — Системна модель додатку для надання рекомендацій по рекламі

Допустимо, що в додатку ВНД є користувачі, котрі реєструються у веб-додатку, СД являється сервіс Amazon, котрий зберігає дані користувача, СА — якийсь довіренний аудитор, що перевіряє цілісність даних, а ЦГК — розробник додатка, котрий генерує ключ для користувача.

Алгоритм роботи даного рішення системної моделі:

1) запуск алгоритму де відбувається генерація та передача приватних/відкритих ключів, а також задання початкових параметрів;

2) збір НД про натискання на ту, чи іншу рекламу, та передача НД у СД;

3) відбувається запит від СА до СД з питанням про цілісність НД та отримує відповідь;

4) відправка з СА результатів свого запиту користувачам та робочим вузлам, якщо результат буде негативний, тобо дані будуть пошкоджені або втрачені, то робочі вузли працювати не будуть;

5) робочі вузли підтягують дані із СД та запускають алгоритм розподіленого проксимального градієнту із затримкою блоку для отримання параметрів, після чого результати відправляють на сервер параметрів. Таким чином, відбувається оновлення вхідних параметрів на сервері параметрів, параметрами робочих вузлів;

6) додаток надає рекомендації по рекламі користувачам на основі результатів навчання.

3.3 Детальна побудова схеми РМН-ПЦД

В цьому розділі розписана детальна побудова схеми РМН-ПЦД. Дана схема включає в себе шість етапів, такі як налаштування, створення додаткового ключа, генерація тегів (збір НД), запит СА, генерація відповіді підтвердження цілісності (відповіді від СД) та перевірка відповіді від СД.

3.3.1 Налаштування

Нехай H_1 та H_2 — це дві мультплікативні циклічні групи з одним і тим самим порядком — це їхнє білінійне відображення: $H_1 x H_1 \to H_2$, а u — елемент, випадково вибраним з H_1. Дві одностороні хеш-функції: $HASH_1 \odot : \{0, 1\}^* \; Z_c'$ та $HASH_2 \odot : \{0, 1\}^* \to H_1$. В алгоритмі RSA d — приватний ключ ВНД, c — відкритий ключ ВНД, n — модуль.

ЦГК вибирає довільний генератор $C \in H_1$ і випадковим чином вибирає мастер-ключ $h \in Z_c'$, та обчислюється $C_{pub} = C^h \in H_1$, що буде головним відкритим ключем. Системними параметрами є :

$$param = \{H_1, H_2, e, u, C, C_{pub}, HASH_1, HASH_2, (l, n)\}$$

3.3.2 Створення додаткового ключа

ВНД відправляє свій ідентифікаційний ідентифікатор $ID \in .\{0,1\}$ в ЦГК для створення часткового приватного ключа. По-перше, ЦГК вираховує $Q = HASH_2(ID)$, застосовує головний ключ h для обчислення часткового приватного ключа ВНД $D = Q_s$. ЦГК відправляє частковий приватний ключ D ВНД.

По-друге, ВНД може перевірити правильність частково приватного ключа від ЦГК наступним чином:

$$e(D, C) \stackrel{?}{=} e(Q, C_{pub}). \qquad (3.1)$$

Якщо наведене вище рівняння вірне, то воно вказує на правильність часткового приватного ключа, отриманого ВНД від ЦГК. В інакшому випадку ВНД завершує операцію.

По-третє, якщо частково приватний ключ правильний, то ВНД випадковим чином вибирає секретне значення $x \in Z_c'$. Після чого він обчислює $sk = HASH_1(D^x)$ в якості приватного ключа.

Нарешті, ВНД обчислює:

$$Z = C_{pub}^{x} = C^{sx}, X = C^{sk} iY = \left(\left(HASH_2(X \| Z) \right) Q \right)^x \tag{3.2}$$

ВНД встановлює $pk = (X, Y, Z)$ в якості відкритого ключа.

Всі сутності можуть використовувати публічні параметри param і ідентифікатор ВНД для перевірки відкритого ключа ВНД наступним чином:

$$e(Y, C_{pub}) \stackrel{?}{=} e(HASH_2, X \| Z * Q, Z) \tag{3.3}$$

Якщо дане рівняння виконується вірно, то $pk = (X, Y, Z)$ є правильним відкритим ключем ВНД.

3.3.3 Генерація тегів

По-перше, ВНД збирає навчальні дані та генерує відповідні підписи (теги). Навчальні дані L = {L_1, L_2, ... , L_m} та відповідний підпис для L_i:

$$\sigma_i = \left(HASH_2(V_i) * \square^L \right)^{sk}, \text{ де } V_i = title \| i \, (i = 1, 2, \ldots, m), \tag{3.4}$$

і title рівномірно випадково розподіляє з Z_c. При тому, ВНД застосовує алгоритм підписів RSA для обчислень:

$$CAP_{sk}(title \| m) = (title \| m)^d, \tag{3.5}$$

де d — приватний ключа ВНД в алгоритмі підпису RSA, а n — модуль в алгоритмі підпису RSA. ВНД обчислює як підпис навчальних даних:

$$CAP_L = title \parallel m \parallel CAP_{sk}(title \parallel m) \tag{3.6}$$

По-друге, ВНД завантажує навчальні дані та підписи {L, Ψ, CAP_L} на СД та видаляє їх в локальному сховищі, де:

$$\Psi = \{\sigma_i\},\ i = 1, 2, ..., m \tag{3.7}$$

По-третє, після отримання НД від ВНД СД генерує відповідь на основі завантажених даних та підписів та повертає відповідь ВНД, про отримання та цілісність навчального НД. Відповідь виглядає наступним чином:

$$\sigma = \prod_{i \in [1,m]} \sigma_i^{v_i}, \tag{3.8}$$

$$\mu = \sum_{i \in [1,m]} v_i L_i + rand, \tag{3.9}$$

де v_i , i = 1, 2, ... , m є випадковим значенням з Z_c. Водночас, СД обраховує $F_1 = u^{rand}$ та відправляє F_1 ВНД, де $rand \in Z_c$ випадковим числом (осліплюючий фактор). СД передасть $\{\sigma, \mu, F_1, HASH_2(V_i), v_i\}$, де i = 1, 2, ..., m, що повернуться до ВНД, і після чого він обчислює:

$$F_1 u^{r * sk} \tag{3.10}$$

В підсумку, ВНД перевіряє відповідь отриману від СД наступним чином:

$$e(\sigma * F, C) \stackrel{?}{=} e\left(\prod_{i \in [1,m]} HASH_2(V_i)^{(v_i)} * u^{\mu}, X\right) \tag{3.11}$$

Й тому, якщо дане рівняння доведене, то СД отримує та зберігає навчальні НД повністю.

3.3.4 Запит СА

Для початку, СА перевіряє цілісність підписів навчальних даних CAP_L, перевіряючи чи $CAP_{sk}(title \| m)$ є дійсно підписом з відкритим ключем pk ВНД, тобто СА визначає, чи рівняння $CAP_{sk}(title \| m)^b mod n \stackrel{?}{=} title \| m$ вірне, де b — відкритий ключ ВНД в алгоритмі підпису RSA. СА перериває повідомлення, якщо перевірка не вдалась. В інакшому випадку, СА відновлює $title \| m$ і далі генерує наступне повідомлення про аудит.

По-друге, СА випадковим чином вибирає підрядковий набір J навчального НД, де $J \subset [1, m]$. Також СА вибирає набір випадкових чисел v_i, $i \in J$, де $v_i \in Z_c$.

В підсумку, СА генерує виклик $Q = \{J, v_i, i \in J\}$ та відправляє його на СД.

3.3.5 Генерація відповіді підтвердження цілісності

На виклик СА СД генерує відповідь на основі задачі Q, щоб забезпечити правильність наступним чином:

$$\sigma = \prod_{i \in J} \sigma_i^{v_i} \tag{3.12}$$

$$\mu = \sum_{i \in J} v_i L_i + rand \tag{3.13}$$

де $J \subset [1, m]$ та v_i, $i \in J$ приходять з виклику Q, і rand є випадковим значенням з Z_c. Тоді СД вираховує $F_1 = u^{rand}$ та відправляє F_1 ВНД, де $rand \in Z_c$ випадковим числом (осліплюючий фактор). ВНД обчислює $F = F_1 u^{r * sk}$ та

відправляє на СД. На кінець, СД відправляє відповідь-доказ $\{\sigma, \mu, F, HASH_2(V_i)\}$, де $i \in J$ в СА та гарантує, що він зберігає навчальні НД цілісно.

3.3.6 Перевірка відповіді від СД

Після отримання відповіді згенерованої СД, СА перевіряє відповідь-доказ, щоб дізнатись чи не пошкоджені навчальні НД наступним чином:

$$e(\sigma * F, C) \stackrel{?}{=} e\left(\prod_{i \in J} HASH_2(V_i)^{|v_i|} * u^{\mu}, X\right) \tag{3.14}$$

Підсумую, якщо приведене істинно, то СД повністю зберігає навчальні НД. В інакшому випадку, це доказує те, що навчальні НД, які зберігаються на СД, по якійсь причині втрачені чи пошкоджені.

3.4 Перевірка правильності побудови схеми РМН-ПЦД

Щоб довести, що ВНД може перевірити правильність часткового приватного ключа, що генерується ЦГК, потрібно перевірити еквівалентне судження правильності рівняння (3.1). Відштовхуючись від властивостей білінійного відображення, правильність рівняння (3.1) може бути доведена наступним чином:

$$e(D, C) = e(Q^S, C) = e(Q, C^S) = e(Q, C_{pub}) \tag{3.15}$$

Щоб довести правильність відкритого ключа ВНД, потрібно перевірити еквівалентне судження правильності рівняння (3.3). Відштовхуючись від властивостей білінійного відображення, правильність рівняння (3.3) може бути доведена наступним чином:

$$e(Y, C_{pub}) = e(HASH_2(X \| Z) * Q, C^{xs}) = e(HASH_2(X \| Z) * Q, Z), \quad (3.16)$$

де $pk = (X, Y, Z)$ є відкритим ключем ВНД, а C_{pub}- основним відкритим ключем.

Щоб довести, що СД цілісно зберігає навчальні НД, потрібно перевірити еквівалентне судження правильності рівняння (3.14). Відштовхуючись від властивостей білінійного відображення, правильність рівняння (3.14) можна довести наступним чином:

$$\begin{gathered}
e(\sigma * F, C) = e\left(\prod_{i \in J} \sigma_i^{v_i} F_1^{sk}\right) = ¿ \\
e\left(\prod_{i \in J} \left(\left(HASH_2(V_i) * u^{(L_i)}\right)^{(sk * v_i)} * F_1^{sk}, C\right)\right) = ¿ \\
e\left(\prod_{i \in J} \left(\left(HASH_2(V_i) * u^{(L_i)}\right)^{(sk * v_i)} * u^{sk * rand}, C\right)\right) = ¿ \\
e\left(\prod_{i \in J} \left(HASH_2(V_i) * u^{L_i}\right)^{v_i} * u^{rand}, C^{sk}\right) = ¿ \\
e\left(\prod_{i \in J} HASH_2(V_i)^{v_i} * u^{\sum_{i \in J} v_i L_i} * u^{rand}, X\right) = ¿ \\
e\left(\prod_{i \in J} HASH_2(V_i)^{v_i} * u^{\sum_{i \in J} v_i L_i + rand}, X\right) = ¿ \\
e\left(\prod_{i \in J} HASH_2(V_i)^{v_i} * u^{\mu}, X\right)
\end{gathered} \quad (3.17)$$

Дані судження доводять правильність моєї схеми РМН-ПЦД. По-перше, що ВНД може перевірити правильність часткового приватного ключа, згенерований ЦГК. По-друге, що будь-яка сутність може перевірити правильність відкритого ключа ВНД. І накінець, що СА може правильно перевіряти цілісність навчальних НД, що зберігаються на СД.

3.5 Аналіз забезпечення безпеки схемою РМН-ПЦД

3.5.1 Формальні доведення

В даному розділі доводиться, що дана схема може протидіяти підробці та фальсифікації, а також вирішити ключову проблему умовного депонування ключів і запобігти розкриття СА та мережевим зловмисникам конфіденційних навчальних НД.

Наприклад, якщо який-небудь з оспорюваних навчальних блоків НД L_i чи відповідний σ_i пощкоджений або втрачений на СД, то СД не зможе пройти перевірку цілісності СА.

Коли СД захоче запустити фальсифікативний запит та намагається пройти перевірку цілісності СА, то СД може використати іншу пару навчального блоку НД та підписів (L_t, σ_t) для заміни вибраного (L_i, σ_i). Тому, доведення доказу підпису σ :

$$\sigma = \prod_{i \in J, i \notin t} \sigma_i^{v_i} \sigma_t^{v_t} \tag{3.18}$$

Доведення судження:

$$\mu' = \sum_{i \in J, i \notin t} v_i L_i + v_t L_t + rand \tag{3.19}$$

Тоді, рівняння перевірки (4) може передано даним чином:

$$e(\sigma * F, C) \stackrel{?}{=} e\left(\prod_{i \in J} HASH_2(V_i)^{v_i} * \left(\frac{HASH_2 * (V_t)^{v_t}}{HASH_2 * (V_i)^{v_i}}\right) * u^{\mu'}, X\right) \tag{3.20}$$

Через супротив колізії хеш-функції, $\left(\frac{HASH_2 * (V_t)^{\nu_t}}{HASH_2 * (V_i)^{\nu_i}}\right) \neq 1$. Таким чином, перевірка цілісності (4) не виконується, тому СД не може пройти перевірку цілісності. Тому, дана схема РМН-ПЦД може протистояти фальсифікативним атакам запущеним з СД. Точно так само моя схема РМН-ПЦД також може протистояти атакам, які створюють мережеві зловмисники.

Згідно технічної літератури можна отримати наступну форму судження. Якщо СД може пройти цю перевірку, підробив відповідь СА, то це означає, що й можна вирішити проблему СОАДХ, яка суперечить гіпотезі СОАДХ.

Враховуючи той самий виклик Q = $\{J, \nu_i, i \in J\}$ з СА, правильним доказом повино бути $C = ¿\{\sigma, \mu\}$. Однак, СД може створювати неправильний доказ$C' = ¿\{\sigma', \mu'\}$, де $\sigma \neq \sigma', \mu \neq \mu'$.Якщо доказ сфальсифікований СД, може пройти перевірку СА, у відповідності з рівнянням (4), виходить:

$$e(\sigma' * F, C) = e\left(\prod_{i \in J} HASH_2(V_i)^{\nu_i} * u^{\mu'}, X\right) \tag{3.21}$$

Оскільки, $C = ¿\{\sigma, \mu\}$ є правильним доказом, отримую:

$$e(\sigma * F, C) = e\left(\prod_{i \in J} HASH_2(V_i)^{\nu_i} * u^{\mu}, X\right) \tag{3.22}$$

Відштовхуючись від властивостей білінійної карти, взнаю, що:

$$e(\sigma'/\sigma, C) = e(u^{\mu' - \mu}, X) \tag{3.23}$$

Переписую:

$$e(\sigma'/\sigma, C) = e(u^{\nabla\mu}, C^{sk}), \text{ де } \nabla\mu = \mu' - \mu \tag{3.24}$$

Звідси можна отримати:

$$\sigma^{'}/\sigma = u^{\nabla\mu * sk} \tag{3.25}$$

В той самий час можливо переписати дану формулу:

$$u^{sk} = \left(\sigma^{'}/\sigma\right)^{1/\nabla\mu} \tag{3.26}$$

Тобто, враховуючи u, C, C^{sx}, C^{pk}, можна вирахувати u^{sk}. Отже, можна знайти алгоритм вирішення проблеми СОАДХ, яка суперечить гіпотезі СОАДХ. Таким чином, обчислювально неможливо , щоб розрив НД підроблював відповідь аудита, щоб пройти перевірку СА. Точно так само, і мережеві зловмисники не можуть підробити його відповідь для проходження перевірки СА в моїй схемі РМН-ПЦД.

Згідно з даними судженнями, отримую, що СД не може пройти перевірку цілісності СА, якщо він зберігає навчальні НД не повністю. Тобто схема РМН-ПЦД може противостояти підробці та фальсифікації зі сторони СД та мережевих зловмисників.

В цій схемі РМН-ПЦД можна вирішити проблему депонування ключів. Тобто ЦГК та мережеві зловмисники не можуть підробити приватний ключ користувача. Припущу, що (t, ε) - алгоритм може підробити ідентифікаційний номер. Після того можна знайти $(t^{'}, \varepsilon^{'})$противника А, що може вирішити проблему ОАДХ з $t^{'} < t + C_m\left(p_{HASH} + p_s + 1\right)$та $\varepsilon^{'} \geq \frac{\varepsilon}{p_{HASH} + p_s}$, де C_m- посилається на час, який необхідний для множення скалярних точок в H_1.

Нехай С — генератор H_1, даю (С, C^a, C^b), де $a, b \in Z_c^{'}$, відмітивши $(C, A, B) \in H_1$, претендент Y повинен з'ясувати C^{ab}, котрий вирішує проблему ОАДХ та суперечить його визначенню.

Нехай $C_{pub}=C^{a}$, гру між претендентом Y та супротивником A можливо описати наступним чином: спершу, претендент Y налаштовує гру, і вхід дорівнює 1^{k}. Потім зловмисник A отримує параметри та починає гру.

1) Перевірка на предмет отримання будь-якого доступу через запит відкритого ключа.

Претендент Y володіє таблицею PT, яка включає в себе $\left(ID_i, c_i, x_i, y_i, Q_{ID_i}\right)$і дозволяє зловмиснику A виконувати запити з відкритим ключем на основі ідентифікатора ID_i. По-перше, претендент Y перевіряє, чи знаходяться запрошувані ID_iв таблиці PT. Якщо, наприклад, ID_iнемає в таблиці PT, то претендент Y випадковим чином вибирає два елемента $X_i, y_i \in H_1$. По-друге, претендент Y випадковим чином розігрує “орел-решка”, тобто $c_i \in \{0,1\}$. Я припускаю$Pr\left(c_i=0\right)=\delta$, та $Pr\left(c_i=1\right)=1-\delta$. Якщо$c_i=0$, то претендент Y вираховує $C^{y_i}=HASH_1\left(ID_i\right)=Q_{ID_i}$б в іншому випадку він обчислює$B^{y_i}=HASH_1\left(ID_i\right)=Q_{ID_i}$. Після цього, претендент Y відповідає A з $HASH_1\left(ID_i\right)=Q_{ID_i}$. Накінець, претендент Y додає$\left(ID_i, c_i, x_i, y_i, Q_{ID_i}\right)$в таблицю PT. Якщо$ID_i$є в таблиці PT, претендент Y знаходить Q_{ID_i}, яке відповідає ID_iв таблиці PT, та відправляє Q_{ID_i}A.

2) Перевірка на предмет отримання будь-якого доступу через запит приватного ключа.

Претендент Y володіє таблицею ST, яка включає в себе $\left(x_i, S_{ID_i}\right)$, та дозволяє зловмисникам A виконувати запити з приватним ключом на основі ідентифікатораID_i. По-перше, претендент Y перевіряє, чи знаходиться цейID_iв таблиці ST. ЯкщоID_iнемає в таблиці ST, то гра повертається до фази запитів відкритого ключа. В інакшому випадку претендент Y знаходитьc_i, відповідний ID_iв таблиці PT. Якщо$c_i=0$претендент Y обчислює $A^{y_i}=S_{ID_i}$та додає$\left(x_i, S_{ID_i}\right)$в таблицю ST. Якщо$c_i=1$, претендент Y не зможе згенерувати частковий приватний ключ, який допоможе задовільнити рівняння перевірки.

3) Перевірка на предмет отримання будь-якого доступу за допомогою вихідних запитів.

Супротивник А надсилає запит про отримання ідентифікатора, який не запитується на етапі запитів приватного ключа. Супротивник А успішно підробляє ідентифікаційний ID та тепер може отримати запис $(ID_i, c_i, x_i, y_i, Q_{ID_i})$. Оскільки супротивник А успішно підробляє ідентифікаційний ID, $e(Y, C_{pub}) = e(HASH_2, X \| Z * Q, Z)$ є правильним. Претендент Y передивляється c_i, відповідного ID. Якщо $c_i = 1$, тоді Y зазнає невдачі. В інакшому випадку, $c_i = 0$, й таким чином $e(Y, C_{pub}) = e((HASH_2, X \| Z * Q)^{yb}, C^a)$, $e(HASH_2, X \| Z * Q) = e(HASH_2, X \| Z * Q, C^{sy})$. Тепер $e((HASH_2, X \| Z * Q)^{yb}, C^a) e((HASH_2, X \| Z * Q), C^{aby}) = e((HASH_2, X \| Z * Q), C^{sy})$, $C^{sy} = C^{aby}$ є рішенням проблеми ОАДХ, що належить претенденту Y.

Нехай кількість запитів публічного ключа буде $p_{HASH} \le p$, а кількість запитів приватного ключа буде $p_S \le p$. Зловмисник А не зможе повторно надсилати запит на отримання одного й того ж самого ID, він може запитувати тільки до $p_{HASH} + p_S$ разів. Претендент Y дає різні відповіді на запити у відповідності з різними значеннями результата c_i.

Нехай $Pr(c_i = 0) = \delta$, знаємо що число $c_i = 0$, це $(p_{HASH} + p_S)\delta$, та число $c_i = 1$, це $(p_{HASH} + p_S)(1 - \delta)$. В запитах з приватним ключем, якщо $c_i = 1$, то претендент Y не може згенерувати частковий приватний ключ, який зміг би задовільнити рівняння перевірки. Таким чином, вірогідність невдачі рівняється $(1 - \delta)$, а вірогідність успіху - δ. Як запит з публічним ключем, так і запит з приватним ключем, є успішними, так що претендент Y може успішно вирішити проблему ОАДХ. Тому, вірогідність успіху рівняється :

$$\varepsilon' > \frac{\delta}{(1-\delta)(p_{HASH} + p_s)} \varepsilon \qquad (3.27)$$

Тільки тоді, коли $\delta = \frac{1}{2}$, максимум звідси $\frac{\delta}{(1-\delta)(p_{HASH}+p_s)}\varepsilon$ є $\frac{\varepsilon}{(p_{HASH}+p_s)}$. Тобто, вірогідність успіху претендента є $\varepsilon' \geq \frac{\varepsilon}{(p_{HASH}+p_s)}$. Передбачається, що після $p_{HASH}+p_S$ запитів, зловмисник А виводить набір валідних $(ID_i, c_i, x_i, y_i, Q_{ID_i})$ та $e(Y, C_{pub}) = e(HASH_2, X \| Z * Q, Z)$ є істинним. Таким чином, претендент Y може вирішити проблему ОАДХ в групі H_1, яка противоречить його визначенню.

Дане судження доказує, що дана схема РМН-ПЦД вирішує проблему умовного депонування ключа, тобто ЦГК та зловмисники не можуть підробити приватний ключ користувача.

В цій схемі РМН-ПЦД, відповідно відповіді від СД {σ, μ,F}, СА та зловмисники не можуть розкрити навчальні блоки даних. Наприклад, що алгоритм (t, ε) може отримати навчальні блоки даних в процесі аудиту. Потім можна знайти (t', ε') зловмисника А, який може вирішити проблему ОАДХ з $t' \approx 2t$ та $\varepsilon' \geq \frac{\varepsilon^2}{p}$.

По-перше, зловмисник А ініціює заперечення передачі даних, щоб отримати доказ даних μ. Тоді, припускаю, що $r \in Z_p$- приватний ключ СД, а $F_1 = u^r$ - відповідний публічний ключ, Першим кроком претендент Y обирає випадковий $x \in Z_p$, вираховує $F' = u^x$ та відправляє F' зловмиснику А. Отримавши F', вибирає випадковий елемент $c \in Z_p$ та відправляє претенденту Y. Отримавши c, претендент Y вираховує $y = x + random * c$ та знову відпраляє його зловмиснику А. Якщо $u^y = F' F_1^c$, зловмисник приймає y, в ыншому випадку відкидає його. Таким чином, зловмисник А може отримати $random$. Крім того, згідно $\mu = \sum_{i \in J} v_i R_i + random$, зловмисник А може отримати:

$$\sum_{i \in J} v_i R_i = \mu - random \qquad (3.28)$$

Якщо зловмисник А виконує вищезазначені операції $|J|$-разів, то він може отримати лінійну систему рівнянь. Після чого зловмисник А зможе отримати навчальні блоки даних, вирішуючи лінійні рівняння.

За допомогою вищезазначеного процесу доказу зловмисник А може отримати вирішення задачі ОАДХ, тому що умова успіху (t, ε), зловмисник А може виконати задачу ОАДХ в проміжках $t' \approx 2t$ та з незначною вірогідністю $\varepsilon' \geq \frac{\varepsilon^2}{p}$, що суперечить з визначенням проблеми ОАДХ.

Дане судження доказує, те що дана схема РМН-ПЦД може досягнути захисту конфіденційності, в процесі публічного аудиту СА та мережевих зловмисників, тобто дані суб’єкти не можуть отримати секрет навчального НД.

3.5.2 Порівняння безпеки схеми РМН-ПЦД з іншими схемами

В даному розділі проводиться порівняння безпеки моєї схеми РМН-ПЦД з іншими схемами, для зручності найменую їх: схема 1 — схема Ванга [14], схема 2 — схема Яна [18], схема 3 — схема Чжу [22] та схема 4 — інша схема Ванга [40], що знаходяться в Таблиці 1. Порівняння відносяться до ключової проблеми умовного депонування, підробці СД, атаки за допомогою підробки та проблеми захисту конфіденційності. Згідно Таблиці 1 схема РМН-ПЦД може реалізувати всі вищевказані цілі захисту безпеки, які інші не можуть.

Таблиця 1 — Порівняння різних схем

	Схема 1	Схема 2	Схема 3	Схема 4	РМН-ПЦД
Проблема депонування ключів	Ні	Ні	Ні	Ні	Так
Захист конфіденційності	Так	Ні	Так	Ні	Так
Атака підробкою	Так	Так	Так	Так	Так
Атака з фальсифікацією	Так	Так	Так	Так	Так

3.6 Висновок

В даному розділі було виконано проектування моделі системи розподіленого МН на основі схеми перевірки та цілісності даних з використанням асиметричних алгоритмів шифрування. Був наведений приклад роботи даної моделі та проведений аналіз забезпечення безпеки даною схемою. Провів порівняння спроектованої схеми з іншими вже існуючими.

4 РЕАЛІЗАЦІЯ ТА АНАЛІЗ ПРОДУКТИВНОСТІ СИСТЕМИ РОЗПОДІЛЕНОГО МАШИННОГО НАВЧАННЯ НА ОСНОВІ СХЕМИ ПЕРЕВІРКИ ЦІЛІСНОСТІ ДАНИХ З ВИКОРИСТАННЯМ АСИМЕТРИЧНИХ АЛГОРИТМІВ ШИФРУВАННЯ

4.1 Аналіз продуктивності

В цьому розділі проаналізовано витрати на обчислення та накладні витрати на зв’язок в процесі перевірки цілісності даних, а також оціню продуктивність даної схеми РМН-ПЦД на прикладі. Порівнюється вартість обчислень на зв’язок між схемами з Таблиці 1.

4.2 Обрахунок обчислень

Проходить весь процес перевірки на персональному ноутбуці з AMD Ryzen 5 3500U та 8Гб оперативної пам’яті. Основний алгоритм агрегації виконується за допомогою бібліотеки GUN Multiple Precision Arithmetic (GMP) версії 6.1.2. Ми вибираємо криву MNT типу d з бібліотеки PBC. Встановлюємо, що довжина H1 дорівнює 175. Усі експериментальні результати представляють середнє значення 20 випробувань.

У моєму експерименті Y позначає довжину кожного блоку даних, EH позначає операцію потужності на групах H1 і H2, EZ позначає операцію потужності на числовому полі Zp, MH вказує операцію множення на групі H1 і H2, MZ вказує на множення операція над числовим полем Zp, OZ вказує операцію додавання в числовому полі Zp, та HASH позначає операцію обчислення хеш-значення.

В схемі аудиту публічної вибірки вартість обчислень в основному відбувається на СД, СА та ВНД.

4.2.1 Витрати на обчислення для СД

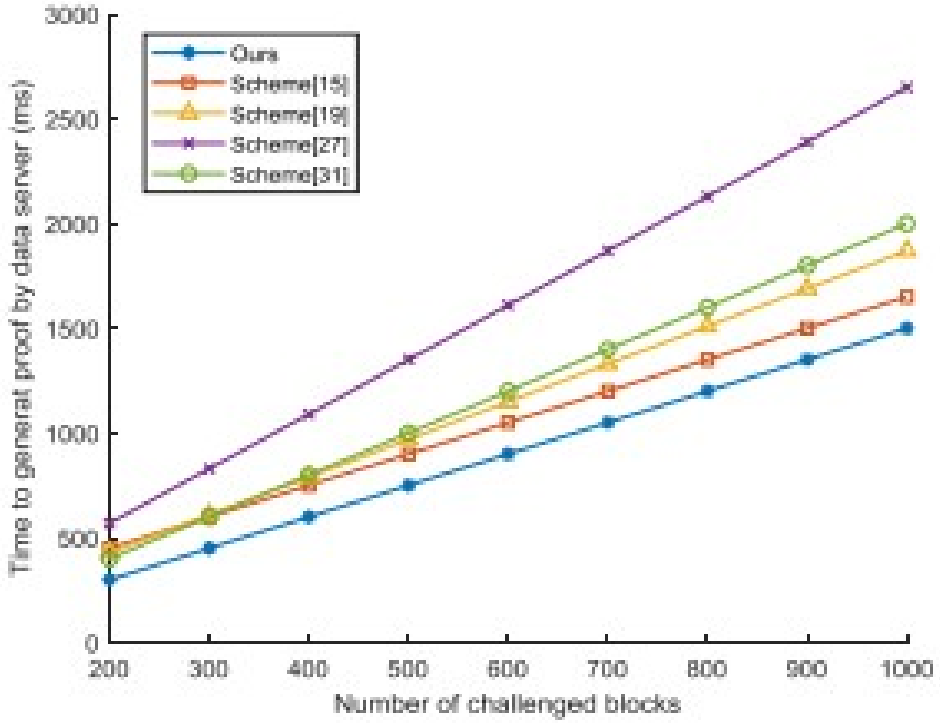


Рисунок 4.1 — Графік порівняння часу генерації доказів СД

У всьому процесі перевірки СД виконує обчислення для створення доказів. Тому імітую доказ часу генерації під час перевірки цілісності даних на СД. Припущу, що кількість блоків для файлу рівно 10000, тобто m = 10000, результат зображений на рисунку 4.1.

На даному графіку відображений час генерації доказів лінійно збільшується з збільшенням кількості блоків. Причина полягає в тому, що СД накопичує тільки блоки, що мають заперечення, та різноманітні відповідні підписи.

Як показано на графіку вартість обчислень СД в моїй схемі меньше, ніж в інших схемах. До прикладу, в схемі 1 система повинна вирахувати маскувальний коефіцієнт з операцією білінійного відображення, операцією хеш-значення та операцією потужності. В схемі 2 СД повинен зіставляти

випадкові нижні індекси та випадкові числа з інформацією про виклик з СА. А в схемі 3 СД повинен обрахувати більше доказів підпису. В схемі 4 система повинна розділити блок даних двічі та провести агрегацію даних і підписів також двічі, щоб отримати докази. В РМН-ПЦД схемі системі потрібно тільки вирахувати маскувальний фактор та провести агрегацію навчального НД та підпису для отримання доказів. Тому в процесі генерації доказу, схема РМН-ПЦД більш ефективна, ніж інші приведені схеми.

4.2.2 Витрати на обчислення для СА

Під час усього процесу перевірки СА виконує обчислення для перевірки доказу. Імітую час перевірки доказу під час перевірки цілісності даних на СА. Припущу, що кількість блоків даних для файлу рівняється 10000, тобто m = 10000, результат зображений на рисунку 4.2.

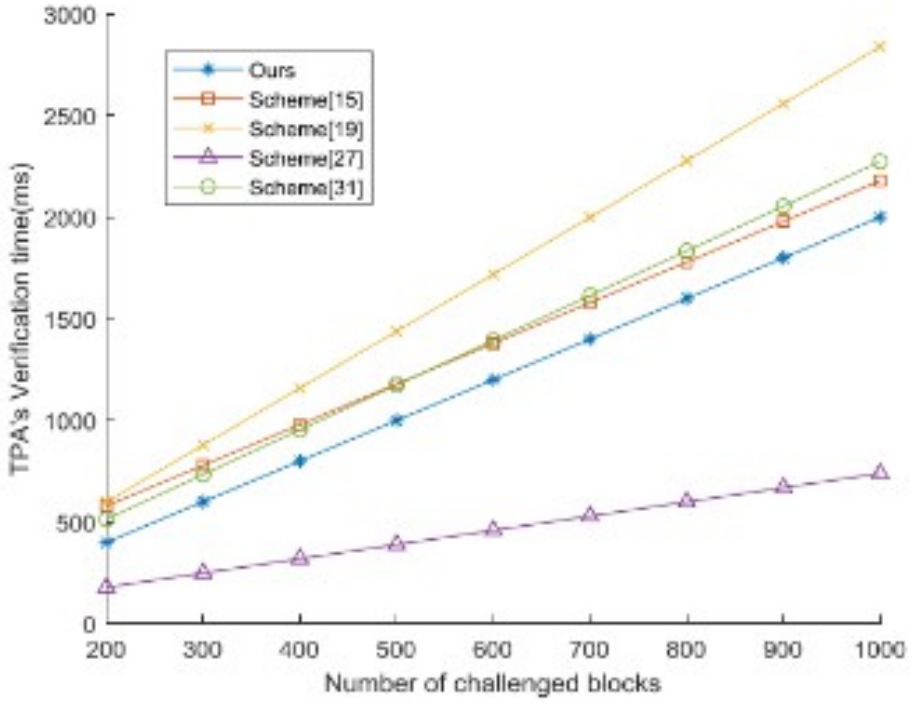


Рисунок 4.2 — Графік порівняння часу перевірки доказу СА

На даному графіку відображений час перевірки доказу лінійно збільшується з кількістю блоків. Причина в тому, що розрахунок для рівняння верифікації пов’язаний тільки з кількістю блоків, що мають заперечення.

Як показано на графіку вартість обчислень СА в схемі РМН-ПЦД меньша, ніж у інших раніше приведених схемах. Але вартість розрахунків СА в схемі РМН-ПЦД більша, ніж у схеми 3. До прикладу, в схемі 1 система повинна додати нові експоненційні операції та операції множення для реалізації захисту конфіденційності. У схемі 2 СА необхідно вираховувати випадкові нижні індекси та випадкові числа на основі двух функцій. Схема 3 має тільки дві операції поєднання, одна операція множення та одна операція додавання перевірки цілісності НД. У схемі 4 алгоритм повинен провести агрегацію доказу з декількох серверів в рівнянні перевірки. В схемі РМН-ПЦД СА потрібно тільки провести агрегацію інформації про блок НД $HASH_2(V_i)$ та вирахувати дві пари білінійного відображення. Тому в процесі перевірки цілісності НД, РМН-ПЦД більш ефективна, ніж схеми 1, 2 та 4, але гірша від схеми 3.

4.2.3 Витрати на обчислення для ВНД

Під час усього процесу перевірки ВНД виконує обчислення для створення підписів. В Таблиці 2 проаналізовано та порівняно обчислювальну вартість ВНД при обчисленні підписів між схемами 1, 2, 3, 4 і схемою РМН-ПЦД. Згідно Таблиці 2, видно, що з точки зору обчислення вартості генерації підписів схема РМН-ПЦД рівна схемам 1, 2, 4 і трохи більше, ніж схема 3.

Таблиця 2 — Порівняння обчислень ВНД

	ВНД
РМН-ПЦД	$\frac{sizeof(L)}{O}(M_H+2E_H+HASH)$
1	$\frac{sizeof(L)}{O}(M_H+2E_H+HASH)$
2	$\frac{sizeof(L)}{O}(M_H+2E_H+HASH)$
3	$\frac{sizeof(L)}{O}(M_H+M_Z+HASH+A_Z)$
4	$\frac{sizeof(L)}{O}(M_H+2E_H+HASH)$

В схемі РМН-ПЦД вартість кожного підпису рівняється $2M_H+E_H+HASH$ та кількість блоків НД є size of(L)/O (де O — кількість бітів блока НД). Таким чином, для файлу даних L, розрахункова вартість генерації підписів: $\frac{sizeof(L)}{O}(M_H+2E_H+HASH)$. Згідно Таблиці 2, обчислення ВНД в схемі РМН-ПЦД рівняються розрахункам в схемах 1, 2, 4. Причина в тому, що підписи блоків НД в схемах 1, 2, 4 такі ж самі як в схемі РМН-ПЦД. Згідно $A_Z+M_Z<M_H+E_H$, я знаю, що обчислення ВНД в схемі РМН-ПЦД трохи більше, ніж в схемі 3. Причина в тому, що в схемі 3 підписи НД мають форму додавання та множення.

4.3 Комунікаційні витрати

В цій частині порівнюю накладні витрати на зв’язок між наведеними схемами 1, 2, 3, 4 та схемою РМН-ПЦД в Таблиці 4. В даній таблиці, $|Z_p|$ відображає розмір Z_p, та $|H|$ відображає розмір групи H_1. В схемах 1, 2, 3 та схемі РМН-П

ЦД відображаю кількість блоків НД, що містять заперечення. В схемі 4 J відображає кількість блоків заперечення НД, а I - кількість блоків НД.

4.3.1 Накладні витрати на зв’язок для запиту

Порівняння комунікаційних накладних витрат для виклику між схемами 1, 2, 3, 4 та схемою РМН-ПЦД. В РМН-ПЦД, запит $Q=¿\{J, v_{i, i \in J}\}$, котрий має $2J|Z_p|$.У схемі 1, запит $Q=¿\{j, v_{ij}\}\❑_{j \in I}$, котрий має $2J|Z_p|$.У схемі 2, запит $Q=¿\{c, k_1, k_2\}$, котрий має $3|Z_p|$.У схемі 3, запит $Q=¿\{i, v_i\}\❑_{i \in J}$, котрий має $2J|Z_p|$.У схемі 4, запит $Q=¿\{j, v_{ij}\}\❑_{j \in I}$, котрий має $2JI|Z_p|$.

Комунікаційні накладні витрати для запиту в схемі 2 меньше, ніж в схемі РМН-ПЦД. Причина в тому, що у схемі 2 накладні витрати на зв’язок для запиту рівняються $3|Z_p|$., а в схемі РМН-ПЦД накладні витрати на комунікацію для запиту: $2J|Z_p|$., при чому $J>2$.

Комунікаційні накладні витрати для запиту в схемі 3 рівні зі схемою РМН-ПЦД. Причина в тому, що у схемі 3 накладні витрати на зв’язок для запиту, як і в РМН-ПЦД, рівняються $2J|Z_p|$.

Комунікаційні накладні витрати для запиту в схемі 4 більше, ніж в схемі РМН-ПЦД. Причина в тому, що у схемі 4 система повинна зберігати одні й ті самі блоки НД на декількох серверах, що може призвести до відправки проблеми на декілька серверів. В схемі РМН-ПЦД розділяються навчальні НД на n-блоків та зберігаєм блоки НД на СД, тому завдання відправляється тільки на один СД.

Тому що у схемі 1, точно так само застосована ідея PDP, тобто підтвердження володіння даними, накладні витрати на зв’язок для запиту в схемі 1 такі ж самі, як і в схемі РМН-ПЦД.

4.3.2 Накладні витрати на зв’язок для доказу

Порівняння комунікаційних накладних витрат на зв’язок для доказу між схемами 1, 2, 3, 4 та схемою РМН-ПЦД. В РМН-ПЦД, запит $Q=¿\{\sigma, \mu, F, HASH_2(m_{ij})_{i \in J}\}$, котрий коштує $(J+1)|H|+2|Z_p|$.

В схемі 1, доказ відправляється хмарним сервером, що являється $\{\sigma, \mu, F, HASH_2(m_{ij})_{i \in J}\}$, і котрий коштує $(J+1)|H|+2|Z_p|$.

В схемі 2, доказ відправляється хмарним сервером, що являється $\{\overline{T}, \overline{L}, HASH_2(a \parallel L_{id} \parallel v_i)_{i \in J}\}$, і котрий коштує $(J+1)|H|+|Z_p|$.

В схемі 3, доказ відправляється хмарним сервером, що являється $\{F, \mu, \eta, HASH_2(m_{ij})_{i \in J}\}$, і котрий коштує $(J+3)|H|$.

В схемі 4, доказ відправляється хмарним сервером, що являється $\{\sigma_i, \mu_i, h(m_{ij})_{i \in J}\}_{j \in I}$, і котрий коштує $IJ|H|+2J|Z_p|$.

Таблиця 3 — Порівняння зв'язку між схемами

	Запит	Доказ
РМН-ПЦД	$2J\|Z_p\|$	$(J+1)\|H\|+2\|Z_p\|$
1	$2J\|Z_p\|$	$(J+1)\|H\|+2\|Z_p\|$
2	$3\|Z_p\|$	$(J+1)\|H\|+\|Z_p\|$
3	$2J\|Z_p\|$	$(J+3)\|H\|$
4	$2JI\|Z_p\|$	$IJ\|H\|+2J\|Z_p\|$

Як видно з таблиці 3 накладні витрати на зв'язок для доказу у схемі 2, набагато меньше, ніж у схемі РМН-ПЦД. Причина в тому, що в схемі 2 накладні витрати на доказ є $(J+1)|H|+|Z_p|$,а в РМН-ПЦД рівняється $(J+1)|H|+2|Z_p|$.

Накладні витрати на зв'язок між схемою 3 та РМН-ПЦД приблизно рівняються. Оскільки порядок групи H_1 дорівнює порядку Z_p, можна отримати $|Z_p| \approx |H|$. Тому, $(J+3)|H| \approx (J+1)|H|+2|Z_p|$, це означає, що накладні витрати на зв'язок для доказу в схемі 3 приблизно рівні накладним витратам в схемі РМН-ПЦД.

Також накладні витрати на зв'язок для доказу у схемі 4, більше, ніж у схемі РМН-ПЦД. Причина в тому, що в схемі 4 декілька серверів повинні генерувати доказ та відправляти його в СА, щоб підтвердити, що вони зберігають НД разом. В РМН-ПЦД, розділюються навчальні НД на n-блоків та зберігаєм блоки НД на СД, таким чином, тільки одному серверу потрібно підтвердити доказ в СА.

Тому що у схемі 1, точно так само застосована ідея PDP, накладні витрати на зв’язок для доказу в схемі 1 такі ж самі, як і в схемі PMH-ПЦВ.

4.4 Оцінка схеми PMH-ПЦД

В цій частині, застосовується оцінка реклами, наведеної ранішше, для оцінки ефективності схеми PMH-ПЦД.

Збираючи НД для прогнозування кліків з 1.5 мілліардами прикладів та 0.6 мілліардами унікальних функцій. Даний НД має в собі інформації на 1.4ТБ та зберігається на Amazon. Імітуєм роботу, з допомогою фреймворку Parameter Server, на 12 машинах, кожна з яких має по 8 фізичних ядер. Дев’ять машин діють як робочі вузли, а три машини діють як сервери параметрів.

Застосовуючи алгоритм PMH для оцінки ефективності нашої схеми PMH-ПЦД. Порівняння обчислень та часу очікування між загальним сервером параметрів без захисту цілісності даних та схемою PMH-ПЦД показано на рисунку 4.1.

Згідно даному порівнянню, даний фреймворк без захисту цілісності даних, час обчислень рівняється 0.901 год, а час очікування — 0.115 год. В PMH-ПЦД, час обчислень дорівнює 0.921 год, а час очікування — 0.128 год. Тому, таким чином, схема PMH-ПЦД майже така сама, як і загальна структура фреймворку Parameter Server, в плані часу обчислень та очікування. Хоча в схемі PMH-ПЦД система повинна перевіряти цілісність тренувальних НД, додавання обчислення та час очікування для кожного робочого вузла практично незначні. Тому під час обчислення та очікування схема PMH-ПЦД майже так сама, як і загальна система Parameter Server.

4.5 Основні компоненти для створення програмної реалізації

Для створення програмної реалізації поставленої задачі, тобто створення моделі МН на основі РМН-ПЦД, я використав програмне середовище Visual Studio Code. Програмна реалізація написана на мові програмування Python на основі веб-сокетів, з додатковим використанням суміжних фреймворків та бібліотек. Фреймворки, які використовувались для створення програмної реалізації: Spark, а точніше його надбудова над Python — PySpark, Flask.

Spark Framework — це простий та виразний Java/Kotlin DSL веб-фреймворк, створений для швидкого розвитку в сфері МН. Spark надає альтернативу для розробників Java/Kotlin, що зочуть розробляти власні веб-додатки якомога виразнішими та з використанням мінімальних шаблонів. Завдяки чіткій філософії Spark створено не тільки для того, щоб зробити веб-додаток продуктивнішим, але й зробити код кращим за допомогою декларативного синтаксису Spark.

PySpark — це інтерфейс для Apache Spark для Python. Він не тільки дозволяє писати програми Spark за допомогою API Python, але також надає оболонку PySpark для інтерактивного аналізу даних у розподіленому середовищі. PySpark підтримує більшість функцій Spark, таких як Spark SQL, DataFrame, Streaming, Mllib (для МН) та Spark Core, зображено на рисунку 4.3.

Flask — це мікровеб-фреймворк, написаний на Python. Його класифікують як мікрофреймворк через те, що він не вимагає особливих інструментів та бібліотек.

У ньому немає рівня абстракції бази даних, перевірки форми чи будь-яких інших компонентів, де вже існуючі сторонні бібліотеки надають загальні функції.

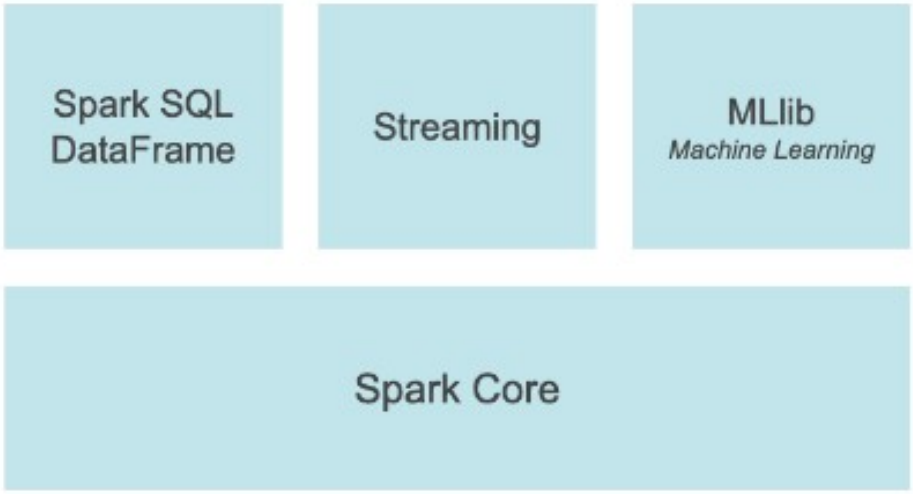


Рисунок 4.3 — Внутрішня система Spark

Натомість Flask підтримує розширення для додавання такої функціональності до вашої програми, як якщо б вона була реалізована в самому Flask. Численні розширення забезпечують інтеграцію бази даних, перевірку форм, обробку завантаження, різні технології відкритої аутентифікації тощо.

Допоміжними бібліотеками для допомоги в розробці є pandas, findspark, matplotlib.pyplot.

Через те, що PySpark по замовчуванню не містить sys.path, використовується findspark. Тобто findspark вирішує проблему, додавши символьне посилання у пакети сайту, або додавши PySpark до sys.path під час виконання.

Matplotlib — це бібліотека для створення статичних, анімованих та інтерективних візуалізацій на мові програмування Python.

Pandas — це високорівнева бібліотека на мові програмування Python, що призначена для обробки та аналізу великих об’ємів даних. Дана бібліотека працює з даними поверх бібліотеки NumPy, тому що вона являється більш низькорівневою бібліотекою. Вона надає спеціальні структури даних, щоб маніпулювати таблицями та часовими чергами.

4.6 Розробка структури веб-додатку для прогнозування цін на авто за допомогою МН

Створений веб-додаток міститиме меню для вибору метода навчання моделі, поле для зазначення кількості задіяних робочих вузлів, поле для виводу інформації про виконані дії (logs) та результати навчання вибраної моделі МН з вкзаним числом робочих вузлів. Також у root-директорії проекту будуть створюватись графічні зображення (графіки) для відображення точності моделі у зв’язку із заданими початковими даними.

Для відтворення даної концепції для початку створена та стилізована веб-сторінка з попередньо зазначеними полями, що зображена на рисунку 4.4.

Рисунок 4.4 — Веб-сторінка майбутнього веб-додатку

Для того, щоб ефективно прогнозувати ціни на авто вибрав декілька типів МН:

1) Лінійна регресія;
2) Регресія з використанням дерева рішень;
3) Регресія з використанням Random forest;
4) Регресія з використанням дерева рішень під впливом градієнта.

4.6.1 Лінійна регресія

Найпростішими словами, лінійна регресія [43-44] — це контрольована модель МН, в якій модель знаходить найбільш відповідну лінійну залежність між незалежною та залежною змінною, тобто вона знаходить лінійний зв’язок

між залежною та незалежною змінними. Приклад лінійної регресії зображений на рисунку 4.5.

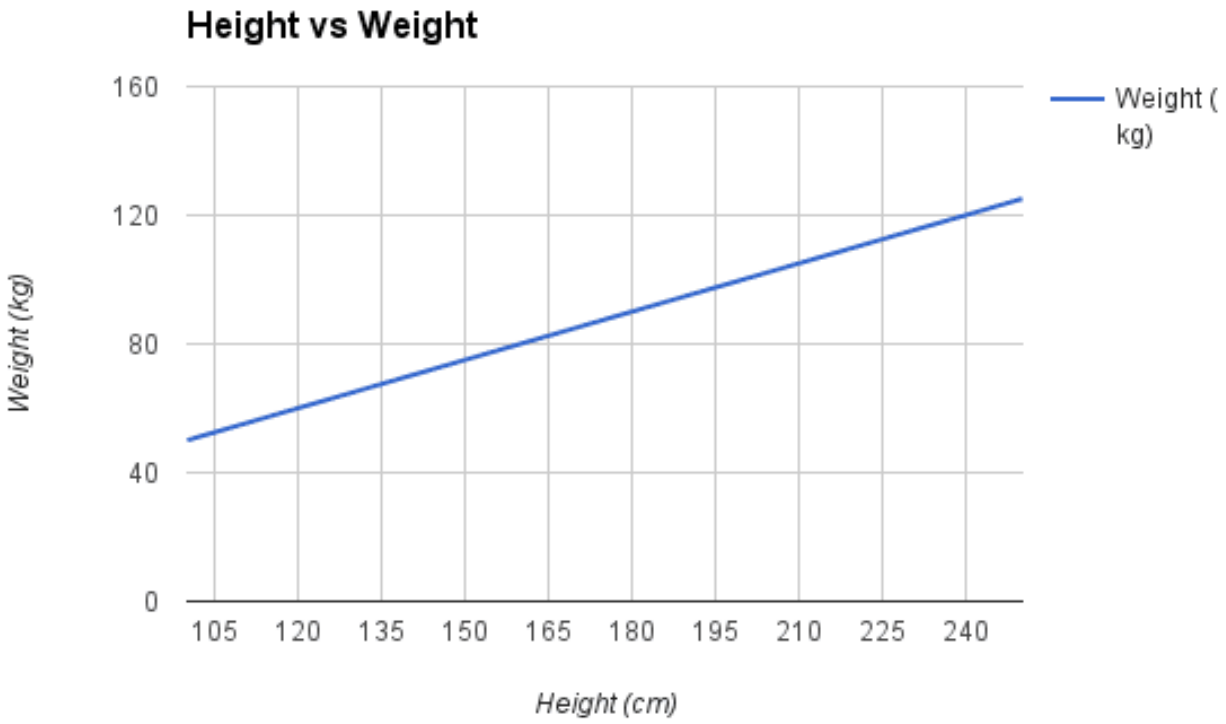


Рисунок 4.5 — Лінійна регресія

Лінійна регресія базується [48] на звичайних найменших квадратах (OLS), модель підібрана таким чином, що сума квадратів різниць спостережуваних і прогнозованих значень мінімізується.

Модель лінійної регресії заснована на кількох припущеннях (наприклад, помилки зазвичай розподіляються з нульовим середнім і постійною дисперсією). За умови виконання припущень, регресійні оцінки [57] є оптимальними в тому сенсі, що вони неупереджені, ефективні та послідовні. Незміщене означає, що очікуване значення оцінювача дорівнює справжньому значенню параметра. Ефективний означає, що оцінювач має меншу дисперсію, ніж будь-який інший оцінювач. Послідовний означає, що зсув і дисперсія

оцінювача наближаються до нуля, коли розмір вибірки наближається до нескінченності.

4.6.2 Регресія з використанням дерева рішень

Дерево рішень [49-51] будує моделі регресії або класифікації у вигляді деревоподібної структури. Він розбиває набір даних на все менші й менші підмножини, в той же час поступово розвивається пов’язане дерево рішень. Кінцевим результатом є дерево з вузлами рішень і листовими вузлами.

Вузол прийняття рішень має дві або більше гілок, кожна з яких представляє значення для перевіреного атрибута. Листовий вузол представляє рішення щодо числової цілі. Найвищий вузол прийняття рішень у дереві, який відповідає найкращому предиктору, який називається кореневим вузлом. Дерева рішень можуть обробляти як категоріальні, так і числові дані. Приклад регресії із застосуванням дерева рішень зображено на рисунку 4.6.

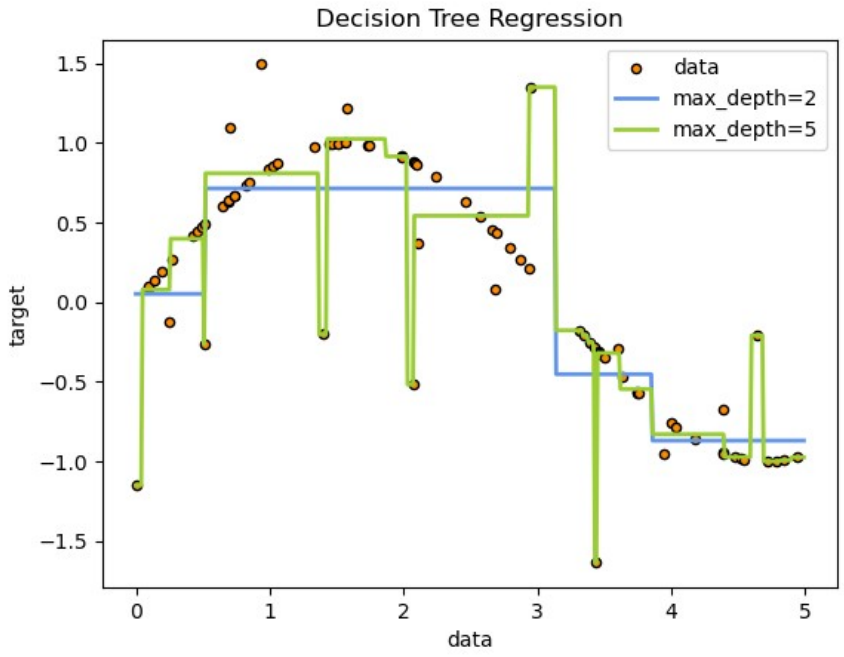


Рисунок 4.6 — Регресія із використанням дерева рішень

Основний алгоритм для побудови дерев рішень під назвою ID3 Дж. Р. Квінлана, який використовує жадібний пошук зверху вниз у просторі можливих гілок без повернення назад. Алгоритм ID3 можна використовувати для побудови дерева рішень для регресії, замінивши приріст інформації на зменшення стандартного відхилення.

Дерево рішень [54] будується зверху вниз від кореневого вузла і передбачає розбиття даних на підмножини, які містять екземпляри з подібними значеннями (однорідними). Я використовую стандартне відхилення для обчислення однорідності числової вибірки. Якщо числова вибірка повністю однорідна, її стандартне відхилення дорівнює нулю.

Зменшення стандартного відхилення засноване на зменшенні стандартного відхилення після того, як набір даних розділений на атрибут. Побудова дерева рішень полягає в тому, щоб знайти атрибут, який повертає найвище зменшення стандартного відхилення (тобто найбільш однорідні гілки).

На практиці нам потрібні деякі критерії припинення. Наприклад, коли коефіцієнт відхилення (CV) для гілки стає меншим за певний поріг (наприклад, 10%) та/або коли у гілки залишається занадто мало екземплярів (n)

Коли кількість екземплярів у листовому вузлі більше одного, ми обчислюємо середнє як кінцеве значення для цілі.

4.6.3 Регресія з використанням Random forest

Регресія з використанням Random forest [46-47] — це алгоритм навчання з наглядом, який використовує метод ансамблевого навчання для регресії. Метод ансамблевого навчання – це методика, яка поєднує передбачення з кількох алгоритмів машинного навчання, щоб зробити прогноз більш точним, ніж одна модель. Вимога до ансамблевого навчання полягає в тому, що помилки кожної моделі (у даному випадку дерева рішень) є незалежними та

відрізняються від дерева до дерева. Приклад регресії із використанням Random forest зображено на рисунку 4.7.

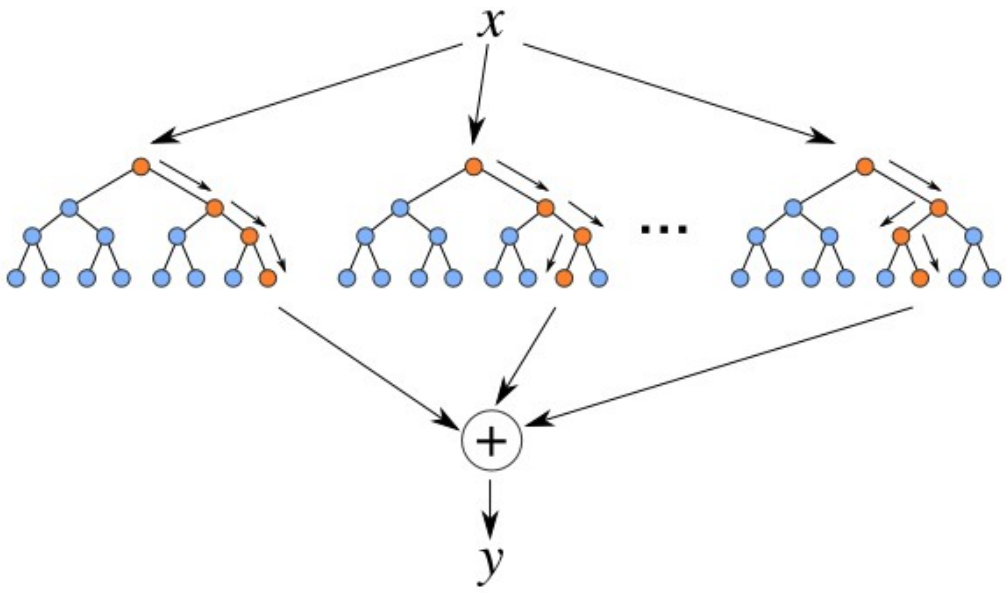


Рисунок 4.7 — Регресія з використанням Random forest

На діаграмі вище показано структуру випадкового лісу. Можна помітити, що дерева працюють паралельно без взаємодії між ними. Випадковий ліс працює шляхом побудови кількох дерев рішень під час навчання та виведення середнього значення класів як передбачення всіх дерев.

Алгоритм роботи Random forest:

1) Потрібно вибрати випадково k точок даних із навчального НД;
2) Побудувати дерево рішень, пов’язане з цими k-точками;
3) Вибрати кількість N дерев, якІ потрібно побудувати, та повторити кроки 1 і 2;

1. Для нової точки потрібно кожне з ваших N-дерев передбачити значення y для відповідної точки даних і призначити новій точці даних середнє значення для всіх передбачених значень y;

Модель регресії із використанням Random forest є потужною та точною. Зазвичай вона відмінно справляється з багатьма проблемами, включаючи

функції з нелінійними зв’язками. До недоліків, однак, можна віднести наступне: відсутність інтерпретації, може легко відбутися переобладнання, та потрібно вибрати кількість дерев для включення в модель.

4.6.4 Регресія із використанням дерева рішень під впливом градієнта

Регресія із використанням дерева рішень під впливом градієнта або коротше Gradient Boosting [52] — це гнучкий непараметричний метод статистичного навчання для регресії. Він достатньо потужний, щоб знайти будь-які нелінійні зв’язки між цільовою метою вашої моделі та функціями, і має чудову зручність використання, яка може впоратися з відсутніми значеннями, викидами та категоріальними значеннями високої потужності для ваших функцій без будь-якої спеціальної обробки [55-56]. Даний метод також вважається методом ансамблевого навчання, тобто коли потрібно створити кілька моделей, щоб отримати кращу продуктивність в загальному. Приклад даної регресії зображений на рисунку 4.8.

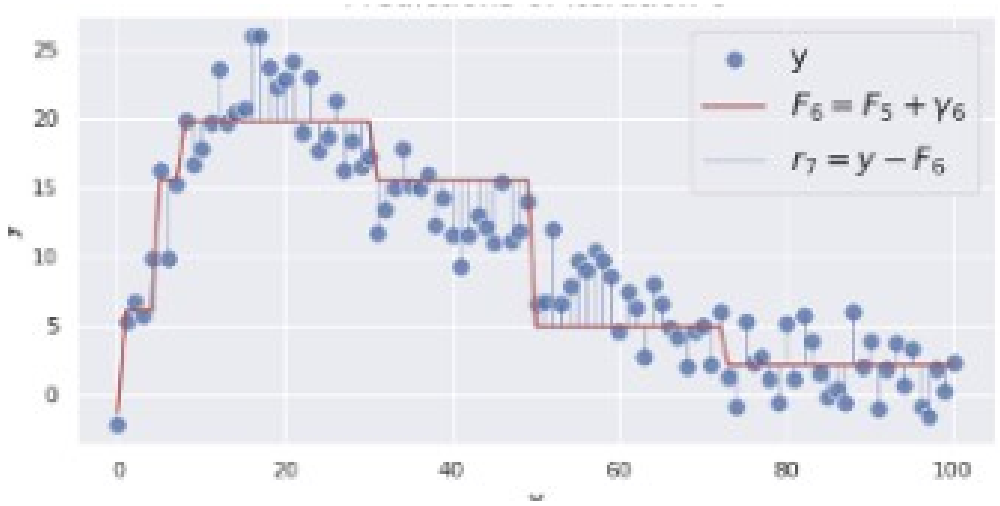


Рисунок 4.8 — Регресія із використанням дерева рішень під впливом градієнта

4.6.5 Результати дослідження методів регресії

Результати імітують реальні дані, а тому є не зовсім точними. Тому, що при використані декількох робочих вузлів та досить невеликої вибірки даних, потрібно або ресурсоємкий комп’ютер, або група комп’ютерів, для використання даних технологій.. На жаль, мій ПК не є тим типом комп’ютеру, на якому можна проводити дані дослідження. Він може дати видимий результат лише на малій кількості вузлів.

1) Лінійна регресія

При обчисленнях на 1 робочому вузлі: точність моделі 100% та час навчання 29.18 сек.

При обчисленнях на 2 робочих вузлах: точність моделі 100% та час навчання 14.72 сек.

При обчисленнях на 4 робочих вузлах: точність моделі 100% та час навчання 13.66 сек.

Дивлячись на результати можна з впевненістю сказати, що модель працює добре та при збільшені кількості робочих вузлів зменшується час навчання, що ми і планували досягти. У даному випадку при збільшені кількості робочих вузлів з 1 до 2, час навчання скоротився удвічі, а збільшивши до 4, в порівнянні з 2, не дало великої різниці. Це спричинено тим, що я описував на початку даного розділу.

2) Регресія з використанням дерева рішень

При обчисленнях на 1 робочому вузлі: точність моделі 98.84% та час навчання 19.83 сек.

При обчисленнях на 2 робочих вузлах: точність моделі 98.9% та час навчання 16.40 сек.

При обчисленнях на 5 робочих вузлах: точність моделі 99.1% та час навчання 15.12 сек.

Дивлячись на результати можна з впевненістю сказати, що модель працює добре та при збільшені кількості робочих вузлів зменшується час навчання, що ми і планували досягти. У даному випадку при збільшені кількості робочих вузлів з 1 до 2, час навчання скоротився на 19%, а збільшивши до 5, в порівнянні з 2, не дало великої різниці, усього лише 8%. Це спричинено тим, що я описував на початку даного розділу.

3) Регресія з використанням Random forest

При обчисленнях на 1 робочому вузлі: точність моделі 96.2% та час навчання 20.06 сек.

При обчисленнях на 2 робочих вузлах: точність моделі 96.1% та час навчання 17.58 сек.

При обчисленнях на 5 робочих вузлах: точність моделі 96.3% та час навчання 17.34 сек.

Дивлячись на результати можна з впевненістю сказати, що модель працює добре та при збільшені кількості робочих вузлів зменшується час навчання, що ми і планували досягти. У даному випадку при збільшені кількості робочих вузлів з 1 до 2, час навчання скоротився майже на 13%, а збільшивши до 5, в порівнянні з 2, майже не дало ніякої різниці 1.37%. Це спричинено тим, що я описував на початку даного розділу.

4) Регресія із використанням дерева рішень під впливом градієнта

При обчисленнях на 1 робочому вузлі: точність моделі 99.4% та час навчання 37.33 сек.

При обчисленнях на 2 робочих вузлах: точність моделі 99.2% та час навчання 30.62 сек.

При обчисленнях на 4 робочих вузлах: точність моделі 99.5% та час навчання 30.61 сек.

Дивлячись на результати можна з впевненістю сказати, що модель працює добре та при збільшені кількості робочих вузлів зменшується час навчання, що ми і планували досягти. У даному випадку при збільшені

кількості робочих вузлів з 1 до 2, час навчання скоротився на 18%, а збільшивши до 4, в порівнянні з 2, майже взагалі не змінився. Це спричинено тим, що я описував на початку даного розділу.

Усі зображення з результатами досліджень регресій наведено у Додатку 2.

4.6 Висновок

В даному розділі була запропонована та розглянута розподілена схема перевірки цілісності даних, яка орієнтована на МН (РМН-ПЦД) для побудови серверу параметрів. Дана схема РМН-ПЦД може забезпечити цілісність навчальних НД, що зберігаються на СД, та протистояти різним атакам, пов’язаних з підробкою інформації. Крім того, дана схема забезпечує захист конфіденційності, вирішує проблему умовного депонування ключів та знижує витрати на управління сертифікатами. Результати моделювання, показують, що схема РМН-ПЦД працює більш ефективно, ніж інші наведені для порівняння схеми.

Була створена та аналізована програмна реалізація у вигляді веб-додатку, ціллю якого стояло прогнозування цін на авто за різними моделями машинного навчання.

ВИСНОВКИ

У роботі за результатами виконаних теоритичних та практичних досліджень була змодельована схема перевірки цілісності даних, яка орієнтована на МН (РМН-ПЦД) та створене на її основі програмне рішення для МН, яке буде прогнозувати ціни на авто, того чи іншого типу, за заданими НД.

У першому розділі було аналізовано відомі методи систем розподілених обчислень, виявлено проблеми безпеки та конфіденційності СРО та способи їх вирішення. Дійшов висновку, що через неконтрольований витік НД, потрібно застосувати асиметричні алгоритми шифрування для збереження конфіденційності НД.

У другому розділі було розглянуто методи застосування розподіленого МН у СРО з використанням асиметричних алгоритмів шифрування. Тобто, в цьому розділі був виконаний теоретичний аналіз області ШІ, а саме МН, проблеми витоку КД, які передаються, при розподілених обчисленнях, розглянуті переваги та недоліки вже існуючих рішень забезпечення захисту та безпеки при передачі конфіденційних НД для навчання моделі РМН та сформована задача на їх виконання.

У третьому розділі було виконано проектування моделі системи розподіленого МН на основі схеми перевірки та цілісності даних з використанням асиметричних алгоритмів шифрування. Був наведений приклад роботи даної моделі та проведений аналіз забезпечення безпеки даною схемою. Провів порівняння спроектованої схеми з іншими вже існуючими.

У четвертому розділі була запропонована та розглянута розподілена схема перевірки цілісності даних, яка орієнтована на МН (РМН-ПЦД) для побудови серверу параметрів. А також створена та аналізована програмна реалізація у вигляді веб-додатку, ціллю якого стояло прогнозування цін на авто за різними моделями машинного навчання.

Впровадження результатів роботи дозволили створити програмну реалізацію для прогнозування цін на авто, за допомогою різних моделей МН за допомогою СРО.

За темою дипломної роботи опублікована одна стаття у фаховому науковому виданні - Збірник тез доповідей Міжнародної науково-практичної конференції “Проблеми та перспективи розвитку науки, освіти та суспільства в XXI столітті”.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Wang, H., et al.: Machine learning basics, *Deep learning*, 2016, ст. 98-164.

2. Sra, Suvrit, Sebastian Nowozin, and Stephen J. Wright, eds.: Optimization for machine learning, *Mit Press*, 2012.

3. Carleo, Giuseppe, et al.: Machine learning and the physical sciences, *Reviews of Modern Physics*, 2019.

4. Xing, Eric P., et al.: Strategies and principles of distributed machine learning on big data, *Engineering 2*, 2016, ст. 179-195.

5. Carleo, Giuseppe, et al.: Machine learning and the physical sciences, *Reviews of Modern Physics*, 2019.

6. Lai, XueJia, et al.: Asymmetric encryption and signature method with DNA technology, *Science China Information Sciences*, 2010, ст. 506-514.

7. M. Li: Scaling distributed machine learning with the parameter server, *in Proc. Int. Conf. Big Data Sci. Comput. (BigDataSci)*, 2014, ст. 583–598.

8. J. Dean and S. Ghemawat: MapReduce: A flexible data processing tool, *Communications of the ACM*, 2010, ст. 72–77.

9. K. He, C. Huang, J. Shi, and J. Wang: Public integrity auditing for dynamic regenerating code based cloud storage, *in Proc. IEEE Symp. Comput. Commun. (ISCC)*, 2016, ст. 581–588.

10. M. Zaharia, M. Chowdhury, and M. Franklin: Spark: Cluster computing with working sets, *in Proc. 2nd USENIX Conf. Hot Topics Cloud Comput.*, 2010, ст. 1–7

11. Y. Yu, M. H. Au, G. Ateniese, X. Huang, W. Susilo, Y. Dai, and G. Min: Identity-based remote data integrity checking with perfect data privacy preserving for cloud storage, *IEEE Trans. Inf. Forensics Security, vol. 12, no. 4*, 2017, ст. 767–778

12. A. Smola and S. Narayanamurthy: An architecture for parallel topic models, *Proc. VLDB Endow., vol. 3, nos. 1–2*, 2010,ст. 703–710

13. W. Shen, J. Qin, J. Yu, R. Hao, and J. Hu: Enabling identity-based integrity auditing and data sharing with sensitive information hiding for secure cloud storage, *IEEE Trans. Inf. Forensics Security, vol. 14, no. 2,* 2019, ст. 331–346

14. C. Wang, S. S. M. Chow, and Q. Wang: Privacy-preserving public auditing for secure cloud storage, *IEEE Trans. Comput., vol. 62, no. 2*, 2013, ст. 362–375.

15. Peteiro-Barral, Diego, and Bertha Guijarro-Berdiñas: A survey of methods for distributed machine learning, *Progress in Artificial Intelligence*, 2013, ст. 1-11.

16. M. Li, Z. Li, and A. Smola: Parameter server for distributed machine learning, *in Proc. Big Learn. NIPS Workshop*, 2013, ст. 1–10.

17. Q. Zheng, S. Xu, and G. Ateniese: Efficient query integrity for outsourced dynamic databases, *in Proc. ACM Workshop Cloud Comput. Secur. Workshop (CCSW)*, 2012, ст. 71–82.

18. H. Yan, J. Li, and Y. Zhang: Remote data checking with a designated verifier in cloud storage, *IEEE Syst. J.*, to be published, doi: 10.1109/jsyst.2019.2918022.

19. C. Wang, Q. Wang, K. Ren, and W. Lou: Privacy-preserving public auditing for data storage security in cloud computing, *in Proc. IEEE INFOCOM*, 2010, ст. 525–533.

20. Li, Mu, et al.: Scaling distributed machine learning with the parameter server, *11th USENIX Symposium on Operating Systems Design and Implementation (OSDI 14)*, 2014.

21. Y. Zhu, H. Hu, G.-J. Ahn, and S. S. Yau: Efficient audit service outsourcing for data integrity in clouds, *J. Syst. Softw., vol. 85, no. 5*, 2012, ст. 1083–1095.

22. H. Zhu, Y. Yuan, Y. Chen, Y. Zha, W. Xi, B. Jia, and Y. Xin,: A secure and efficient data integrity verification scheme for cloud-IoT based on short signature, *IEEE Access, vol. 7*, 2019, ст. 90036–90044.

23. M. Sookhak, F. R. Yu, and A. Y. Zomaya: Auditing big data storage in cloud computing using divide and conquer tables, *IEEE Trans. Parallel Distrib. Syst., vol. 29, no. 5*, 2018, ст. 999–1012.

24. H. Zhao, X. Yao, X. Zheng, T. Qiu, and H. Ning: User stateless privacy-preserving TPA auditing scheme for cloud storage, *J. Netw. Comput. Appl., vol. 129*, 2019, ст. 62–70.

25. H. Yan, J. Li, J. Han, and Y. Zhang: A novel efficient remote data possession checking protocol in cloud storage, *IEEE Trans. Inf. Forensics Security, vol. 12, no. 1*, 2017, ст. 78–88.

26. K. He, C. Huang, J. Shi, and J. Wang: Public integrity auditing for dynamic regenerating code based cloud storage, i*n Proc. IEEE Symp. Comput. Commun. (ISCC)*, 2016, ст. 581–588.

27. Zhao, Xiao-Ping, and Rui Jiang: Distributed machine learning oriented data integrity verification scheme in cloud computing environment, *IEEE Access* 8, 2020.

28. Owusu-Agyemang, Kwabena, et al.: Guaranteed distributed machine learning: Privacy-preserving empirical risk minimization, *Mathematical Biosciences and Engineering* 18.4, 2021, ст. 4772-4796.

29. Bonawitz, Keith, et al.: Practical secure aggregation for privacy-preserving machine learning, *proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, 2017.

30. Froelicher, David, et al.: Scalable privacy-preserving distributed learning, *Proceedings on Privacy Enhancing Technologies* , 2021, ст. 323-347.

31. Fang, Haokun, and Quan Qian: Privacy preserving machine learning with homomorphic encryption and federated learning, *Future Internet 13.4*, 2021, ст. 94.

32. Prabhu, Mukesh M., and S. V. Raghavan: Security in computer networks and distributed systems, *Computer Communications*, 1996, ст. 379-388.

33. Lai, XueJia, et al.: Asymmetric encryption and signature method with DNA technology, *Science China Information Sciences*, 2010, ст. 506-514.

34. H. Wang: Identity-based distributed provable data possession in multi-cloud storage, *IEEE Trans. Services Comput., vol. 8, no. 2*, 2015, ст. 328–340.

35. A. A. Hussain and R. Subashini: Identity-based proxy-oriented data uploading and remote data integrity checking in public cloud, *IEEE Trans.Inf. Forensics Security, vol. 11, no. 6*, 2019, ст. 1165–1176.

36. J. Zhang and Q. Dong: Efficient ID-based public auditing for the outsourced data in cloud storage, *Inf. Sci., vols. 343–344*, 2016, ст. 1–14.

37. J. Li, H. Yan, and Y. Zhang: Certificateless public integrity checking of group shared data on cloud storage, *IEEE Trans. Serv. Comput.*, to be published, doi: 10.1109/tsc.2018.2789893.

38. H. H. Krawczyk: A high-performance secure Diffie–Hellman protocol, *in Advances in Cryptology. Berlin, Germany: Springer*, 2005, ст. 1–62.

39. F. Bao, R. H. Deng, and H. Zhu: Variations of Diffie–Hellman problem, *in Proc. Int. Conf. Inf. Commun. Secur., Huhehaote, China*, 2003, ст. 301–312.

40. A. Hu, R. Jiang, and B. Bhargava: Identity-preserving public integrity checking with dynamic groups for cloud storage, *IEEE Trans. Serv. Comput.*, to be published.

41. Шилін, О. С. "Дослідження методів і технології підвищення ефективності інформаційних Cloud систем з Big Data сегментами.", 2019.

42. Bonawitz, K.; Ivanov, V.; Kreuter, B.; Marcedone, A.; McMahan, H.B.; Patel, S.; Ramage, D.; Segal, A.; Seth: K. Practical secure aggregation for privacy-preserving machine learning, *In Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, Dallas, TX, USA*, 2017; ст. 1175–1191.

43. Hall, R.; Fienberg, S.E.; Nardi, Y.: Secure multiple linear regression based on homomorphic encryption, *J. Off. Stat.*, 2011, ст. *27*, 669.

44. Yi, X.; Paulet, R.; Bertino, E.: Homomorphic encryption, *In Homomorphic Encryption and Applications*, 2014; ст. 27–46.

45. Xing, E.P.; Ho, Q.; Dai, W.; Kim, J.K.; Wei, J.; Lee, S.; Zheng, X.; Xie, P.; Kumar, A.; Yu, Y. Petuum: A new platform for distributed machine learning on big data, *IEEE Trans. Big Data*, 2015, *1*, ст. 49–67.

46. Smith, Paul F., Siva Ganesh, and Ping Liu: A comparison of random forest regression and multiple linear regression for prediction in neuroscience, *Journal of neuroscience methods* 220.1, 2013, ст. 85-91.

47. Svetnik, Vladimir, et al.: Random forest: a classification and regression tool for compound classification and QSAR modeling, *Journal of chemical information and computer sciences*, 2003, ст. 1947-1958.

48. Grömping, Ulrike: Variable importance assessment in regression: linear regression versus random forest, *The American Statistician*, 2009, ст. 308-319.

49. Xu, Min, et al.: Decision tree regression for soft classification of remote sensing data." *Remote Sensing of Environment*, 2005, ст. 322-336.

50. Rathore, Santosh Singh, and Sandeep Kumar: A decision tree regression based approach for the number of software faults prediction, *ACM SIGSOFT software engineering notes*, 2016, ст. 1-6.

51. Ibrahim, Zaidah, and Daliela Rusli: Predicting students' academic performance: comparing artificial neural network, decision tree and linear regression, *21st Annual SAS Malaysia Forum*, 2007.

52. Feng, Ji, Yang Yu, and Zhi-Hua Zhou: Multi-layered gradient boosting decision trees, *Advances in neural information processing systems* 31, 2018.

53. Friedman, Jerome H.: Stochastic gradient boosting, *Computational statistics & data analysis*, 2002, ст. 367-378.

54. Su, Jiang, and Harry Zhang: A fast decision tree learning algorithm, *Aaai*., 2006.

55. Seber, George AF, and Alan J. Lee: *Linear regression analysis*, 2012.

56. Su, Xiaogang, Xin Yan, and Chih-Ling Tsai: Linear regression, *Wiley Interdisciplinary Reviews: Computational Statistics*, 2012, ст. 275-294.

57. Montgomery, Douglas C., Elizabeth A. Peck, and G. Geoffrey Vining: *Introduction to linear regression analysis*, 2021.

Додаток А

ЛІСТИНГ КОДУ

Index.html:

```
<!doctype html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css"
rel="stylesheet"
integrity="sha384-EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuC
OmLASjC" crossorigin="anonymous">
<style>
body {
height: 100vh;
display: flex;
justify-content: center;
align-items: center;
}
form {
margin: 10px;
}
pre {
margin: 40px;
overflow: hidden;
}
pre code {
background-color: #eee;
border-radius: 10px;
padding: 10px;
display: block;
}
code > span {
text-align: left;
}
</style>
<title>Diploma work</title>
</head>
<body>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js"
integrity="sha384-MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/
tWtIaxVXM" crossorigin="anonymous"></script>
<nav class="navbar navbar-dark bg-primary">
<div class="container-fluid">
<span class="navbar-text">
```

```
Denis Symak KI-20-2m. Theme: Method and system of distributed calculations using
asymmetric encryption algorithms
</span>
</div>
</nav>
<div class="container">
<div class="row">
<div class="col">
<form id="form">
<div class="mb-3">
<label for="model" class="form-label">Model</label>
<select id="model" class="form-select" aria-label="Default select example">
<option value="LinearRegression">Linear regression</option>
<option value="DecisionTreeRegressor">Decision tree regression</option>
<option value="RandomForestRegressor">Random forest regression</option>
<option value="GBTRegressor">Gradient-boosted tree regression</option>
</select>
</div>
<div class="mb-3">
<label for="process_number" class="form-label">Number of processes</label>
<input type="text" class="form-control" id="process_number">
</div>
<button type="submit" class="btn btn-primary">Submit</button>
</form>
</div>
<div class="col">
<pre><code >Logs: <div id="log"></div>
<div id="spinner" class="spinner-border spinner-border-sm float-end" role="status"
hidden="true">
<span class="visually-hidden">Loading...</span>
</div>
</code></pre>
</div>
</div>
</div>
<script>
const log = (text) => {
if (text.includes("Time of processing with")) {
document.getElementById('spinner').hidden = true;
}
document.getElementById('log').innerHTML += `${text}`;
};

const socket = new WebSocket('ws://' + location.host + '/echo');
socket.addEventListener('message', ev => {
log(ev.data);
});
document.getElementById('form').onsubmit = ev => {
ev.preventDefault();
```

```
document.getElementById('log').innerHTML = ''
const modelField = document.getElementById('model').value;
const numberOfProcessesField = document.getElementById('process_number').value;
const values = `${modelField},${numberOfProcessesField}`
document.getElementById('spinner').hidden = false;
socket.send(values);
};
</script>
</body>
</html>
```

app.py

```
from flask import Flask, render_template
from flask_sock import Sock
from processing import Processing
import pandas as pd

app = Flask(__name__)
sock = Sock(app)

car = pd.read_csv('car_2.csv')
@app.route('/')
def index():
return render_template('index.html')


@sock.route('/echo')
def echo(sock):
while True:
data = sock.receive()
model, number_of_processes = data.split(',')
process = Processing(model, number_of_processes, sock, car)
process.fit()
```

processing.py

```
import findspark
import pyspark
from pyspark.sql import SparkSession
from pyspark.ml.feature import VectorAssembler
from pyspark.ml.regression import RandomForestRegressor, LinearRegression,
DecisionTreeRegressor, GBTRegressor
from pyspark.ml.evaluation import RegressionEvaluator
from time import time
```

```
import matplotlib.pyplot as plt


class Processing:
def __init__(self, model, number_of_process, socket, dataset):
self.model = model
self.number_of_process = number_of_process
self.socket = socket
self.dataset = dataset
def fit(self):
time0 = time()

self.socket.send("<br> >> Creating Spark context...")
conf =
pyspark.SparkConf().setAppName('Diploma_App').setMaster(f"local[{self.number_of_proce
ss}]")
sc = pyspark.SparkContext(conf=conf)
self.socket.send("<br> >> Spark context created!")

findspark.init()
spark = SparkSession.builder.getOrCreate()
self.socket.send("<br> >> Creating dataframe...")
df = spark.createDataFrame(self.dataset)
self.socket.send("<br> >> Dataframe created!")
self.socket.send("<br> >> Deleting columns...")
X = df.drop('model','type', 'gearbox' 'price', 'posting_date')
self.socket.send("<br> >> Columns deleted!")
self.socket.send("<br> >> Change categorical text data to numeric...")
vecAssembler = VectorAssembler(inputCols=X.columns, outputCol='X')
output = vecAssembler.transform(df).select('X', 'price')
self.socket.send("<br> >> Done!")

train, test = output.randomSplit([0.7, 0.3])
self.socket.send("<br> >> Create and train...")
regr = eval(self.model)(featuresCol='X', labelCol='price')

time1 = time()
model = regr.fit(train)
training_time = time() - time1
self.socket.send(f"<br> Time of training with {self.number_of_process} nodes:
{training_time}s")
self.socket.send("<br> >> Processing is done!")
result_time = "%.2f" % (time() - time0)
self.socket.send(f"<br> Time of processing with {self.number_of_process} nodes:
{result_time}s")
predict = model.transform(test)
score = RegressionEvaluator(predictionCol='prediction', labelCol='price',
metricName='r2')
self.socket.send("<br> Score on model: " + str(score.evaluate(predict)))
```

```
d = predict.toPandas()[100:350]
plt.rcParams['figure.figsize']=15,10
d[['price', 'prediction']].plot()
plt.savefig(f'{self.model}_{self.number_of_process}.png')
sc.stop()
```

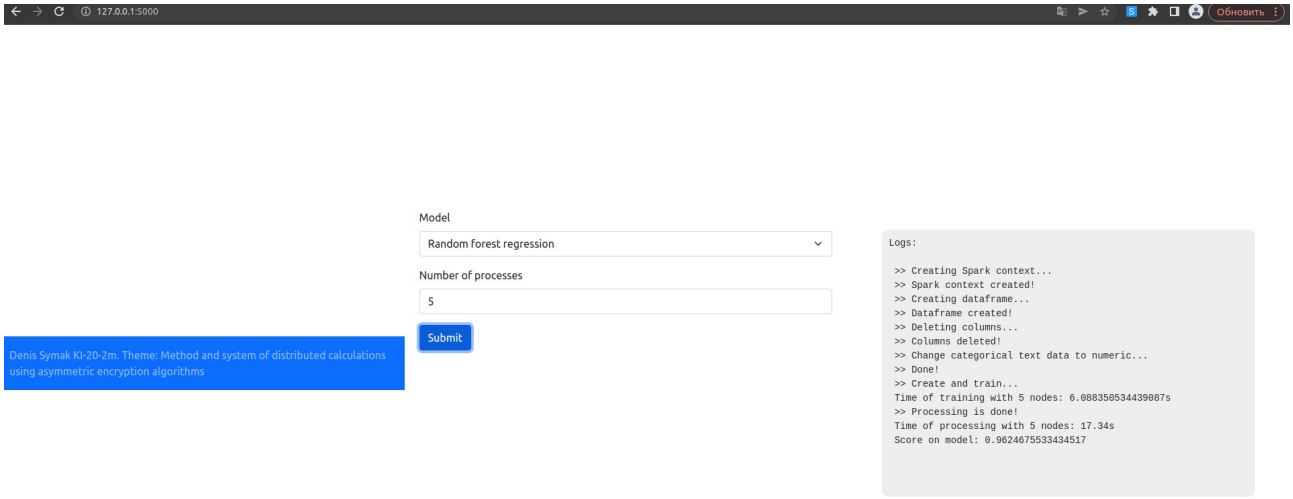
car_2.csv — набІр даних

```
brand,model,type,gearbox,year,mileage,fuel,price,condition,posting_d
17,sonata,7,0,2014,93600,4,7500,1,2020-12-02
4,x3 3.0i,2,0,2006,87046,4,4900,3,2020-12-01
39,tacoma double cab sr5,6,2,2016,33290,4,29590,3,2020-12-01
20,wrangler unlimited sport s,5,2,2017,29614,4,31990,3,2020-11-28
4,m3 coupe 2d,8,2,2013,50956,4,36990,3,2020-11-27
34,1500 crew cab slt pickup 4d,6,2,2016,57926,3,24990,3,2020-11-26
13,expedition,2,0,2003,177000,4,4900,3,2020-11-25
7,corvette grand sport,5,0,2012,49245,4,33990,3,2020-11-25
20,gladiator,6,0,2020,10500,4,47000,1,2020-11-23
39,tacoma double cab sr5,6,2,2018,17117,4,28990,3,2020-11-23
20,wrangler unlimited sport s,2,0,2020,17622,4,34990,3,2020-11-20
4,528i xdrive,7,0,2013,115372,4,11500,1,2020-11-19
4,m3 convertible 2d,5,2,2012,61881,4,27990,3,2020-11-19
13,f-150,6,0,2012,154878,4,15998,1,2020-11-18
10,grand caravan,4,0,2013,94325,4,7998,1,2020-11-18
13,flex,0,0,2009,148452,4,8998,1,2020-11-18
10,challenger,8,0,2015,91661,4,17998,1,2020-11-18
10,journey,2,0,2013,43423,4,10998,1,2020-11-18
13,mustang,8,1,2015,89469,4,13498,1,2020-11-18
13,fusion,7,0,2015,76546,4,8998,1,2020-11-18
23,nx 300 f sport,2,0,2019,18022,4,29988,1,2020-11-18
16,civic lx,7,0,2016,35860,4,15988,1,2020-11-18
26,cla 250,7,0,2014,46313,4,20488,1,2020-11-18
20,wrangler unlimited sahara,5,2,2014,39910,4,28990,3,2020-11-18
13,fusion sel,7,0,2010,82000,4,6250,3,2020-11-14
7,camaro ss coupe 2d,8,2,2018,21240,4,31990,3,2020-11-13
6,xts,7,0,2013,130527,4,12998,1,2020-11-12
14,sierra 2500hd,6,0,2004,160026,0,17998,1,2020-11-12
14,terrain,2,0,2017,74716,4,14498,1,2020-11-12
17,sonata,7,0,2018,39784,4,13998,1,2020-11-12
16,odyssey,4,0,2016,103492,4,14498,1,2020-11-12
7,silverado 1500 crew,6,2,2011,62404,4,25590,3,2020-11-12
39,camry,7,0,2017,50890,4,14498,1,2020-11-11
37,outback,0,0,2011,138247,4,8498,1,2020-11-11
39,prius,3,0,2015,137720,2,7998,1,2020-11-11
37,impreza,0,0,2014,99598,4,12998,1,2020-11-11
```

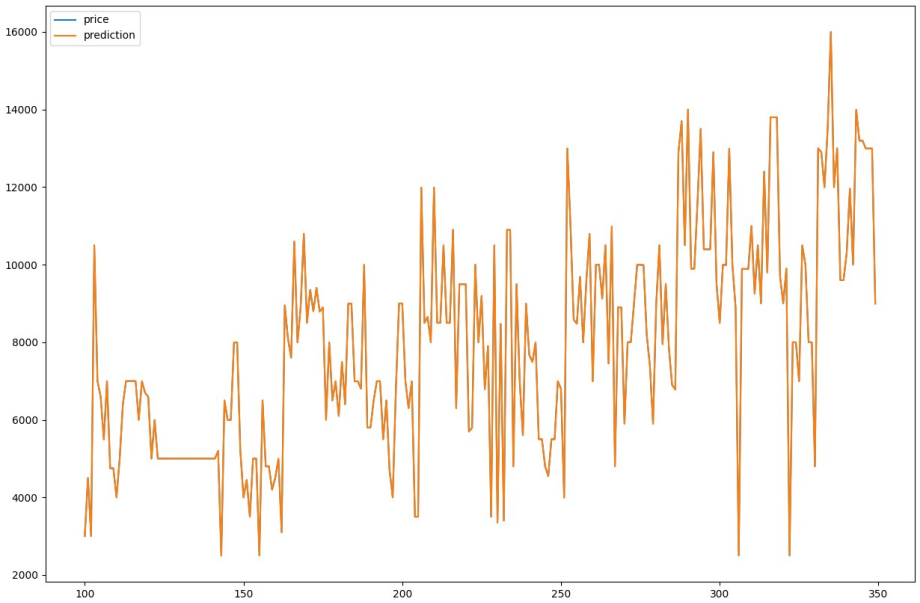
Додаток Б

Результати роботи програми:

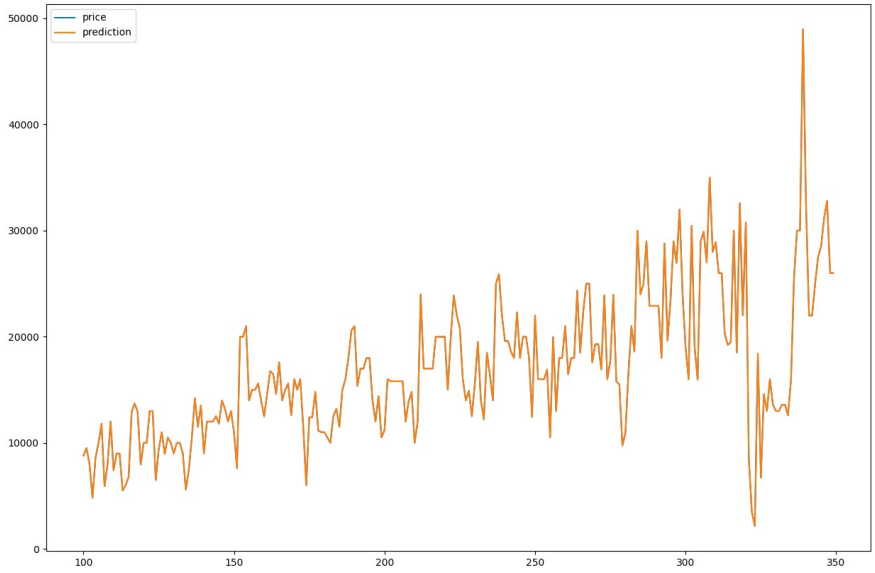
Приклад роботи програми:



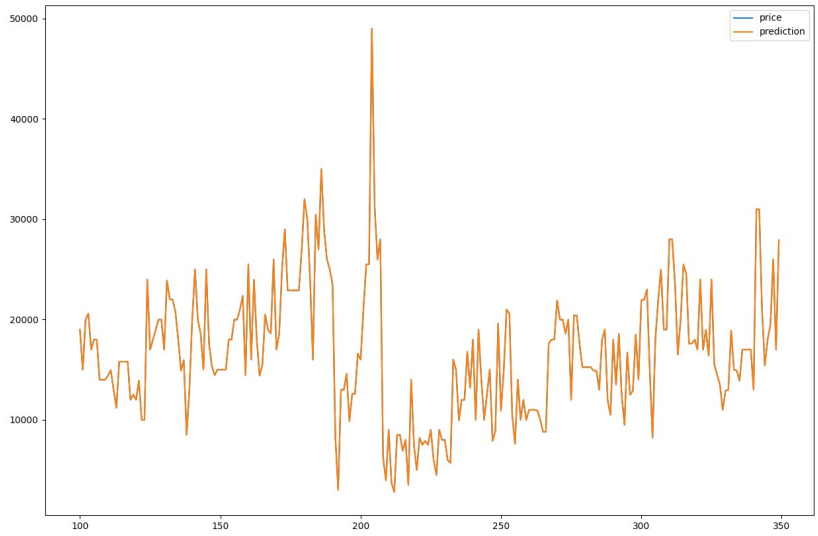
Лінійна регресія з використанням 1 робочого вузла:



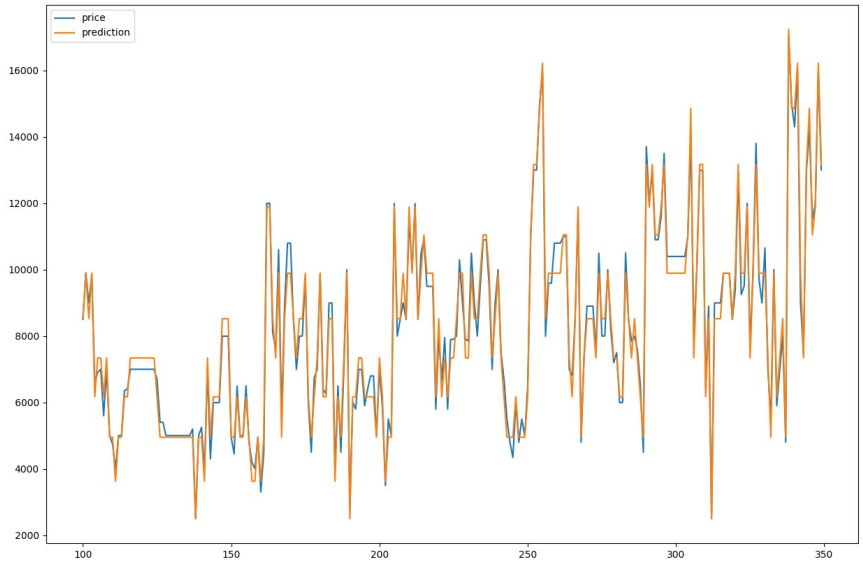
Лінійна регресія з використанням 2 робочих вузлів:



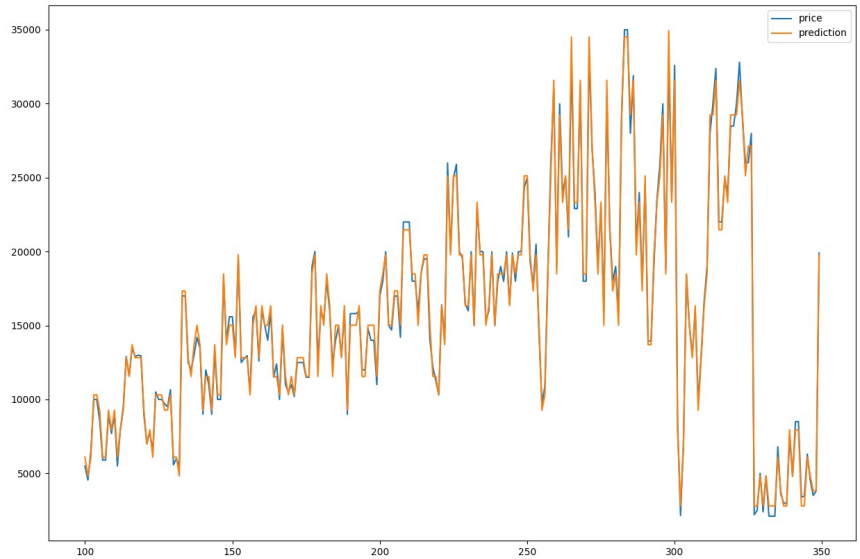
Лінійна регресія з використанням 4 робочих вузлів:



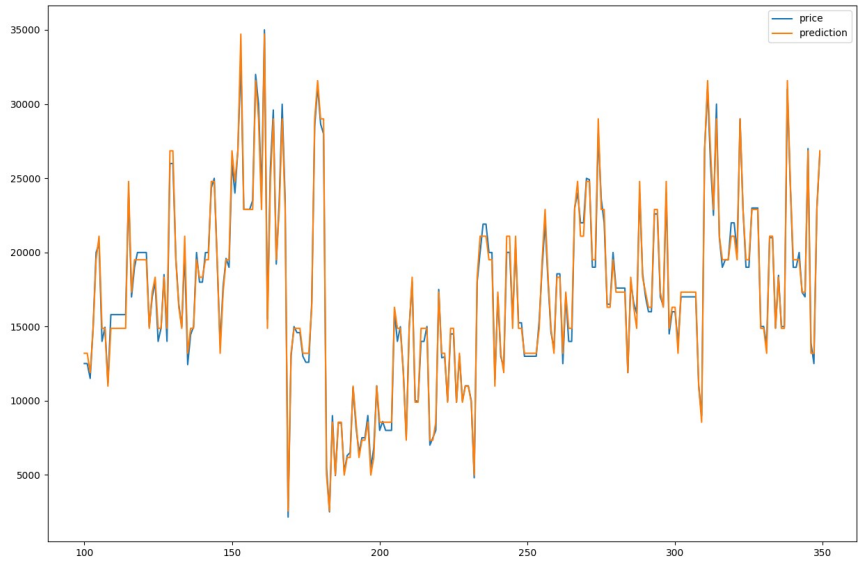
Регресія з використанням дерева рішень з використанням 1 робочого вузла:



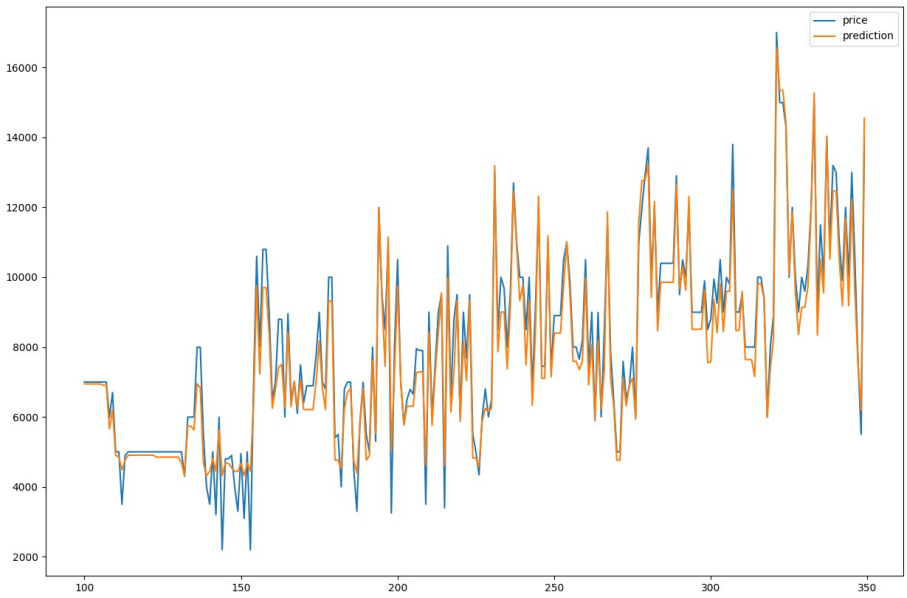
Регресія з використанням дерева рішень з використанням 2 робочих вузлів:



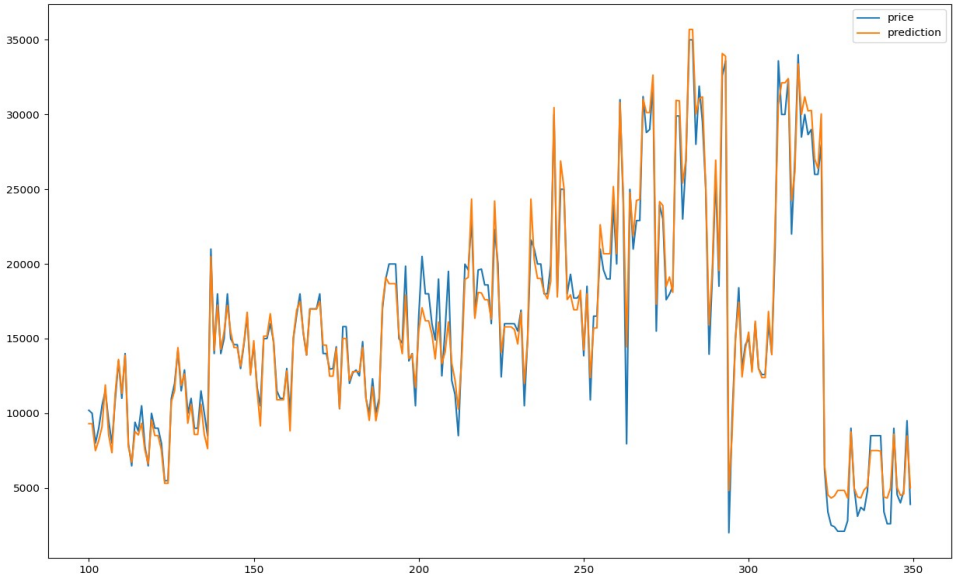
Регресія з використанням дерева рішень з використанням 5 робочих вузлів:



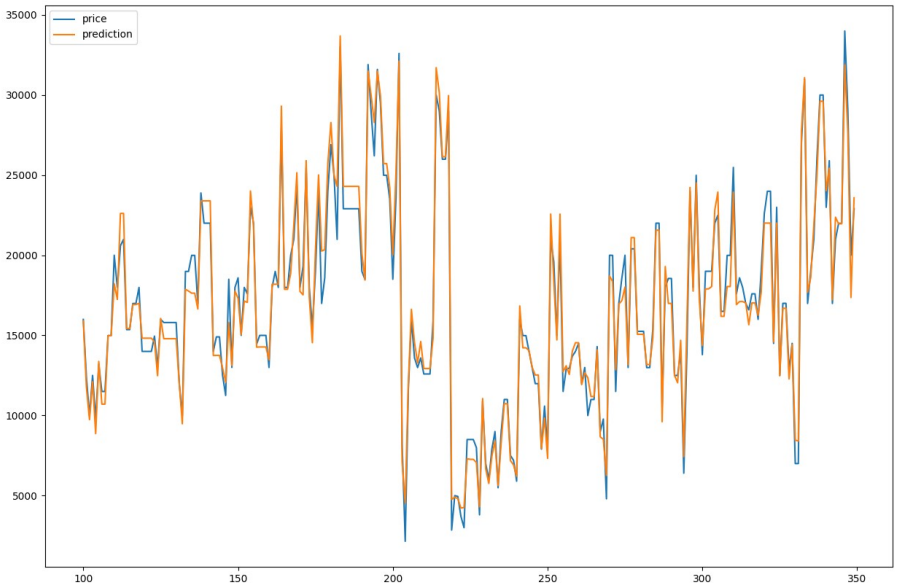
Регресія з використанням Random forest з використанням 1 робочого вузла:



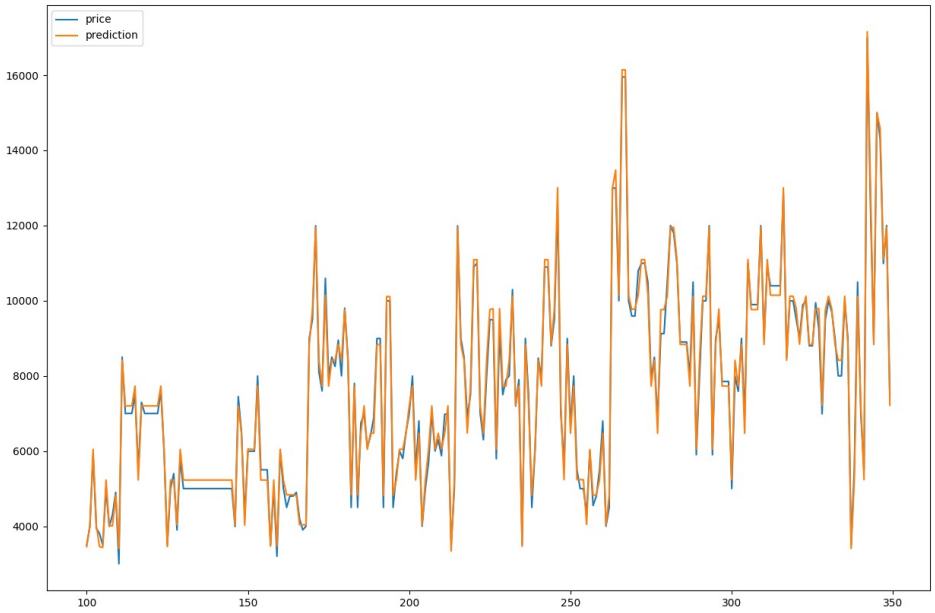
Регресія з використанням Random forest з використанням 2 робочих вузлів:



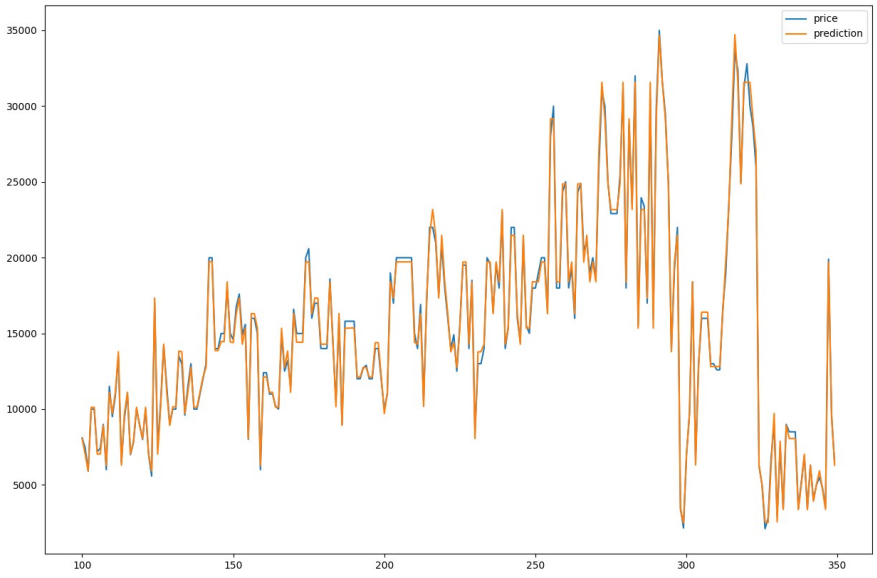
Регресія з використанням Random forest з використанням 5 робочого вузлів:



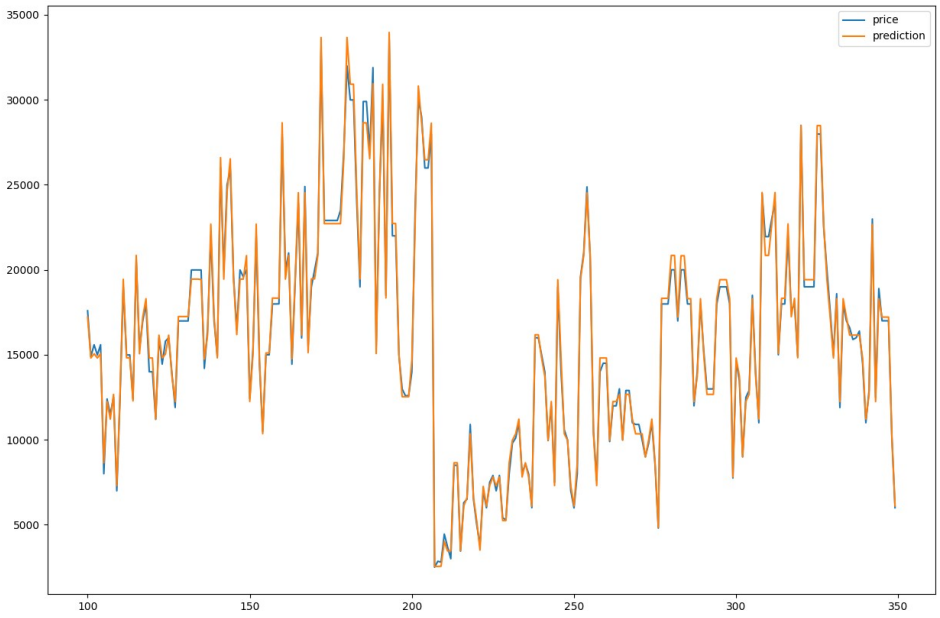
Регресія із використанням дерева рішень під впливом градієнта з використанням 1 робочого вузла:



Регресія із використанням дерева рішень під впливом градієнта з використанням 2 робочих вузлів:



Регресія із використанням дерева рішень під впливом градієнта з використанням 4 робочих вузлів:



Додаток В

(обов’язковий)

КОПІЯ ФАХОВОЇ СТАТТІ

Симак Д.О.

студент кафедри комп’ютерної інженерії та інформаційних систем

Хмельницький національний університет

МОДЕЛЬ СИСТЕМИ РОЗПОДІЛЕНОГО МАШИННОГО НАВЧАННЯ НА ОСНОВІ СХЕМИ ПЕРЕВІРКИ ЦІЛІСНОСТІ ДАНИХ З ВИКОРИСТАННЯМ АСИМЕТРИЧНИХ АЛГОРИТМІВ ШИФРУВАННЯ

Машинне навчання дозволяє створювати системи, які навчаються, або удосконалюють продуктивність, за допомогою аналізу даних. Тобто МН допомагають зробити взаємодію з користувачем зручнішою, ефективнішою та безпечною. На даний момент можливості МН є тільки вершиною айсбергу, тієї кількості можливостей які з’являться у майбутньому.

Схема РМН-ПЦД складається з чотирьох компонентів: серверу даних, стороннього аудитора [1], власника НД та центру генерації ключів, які зображенні на рисунку 1.

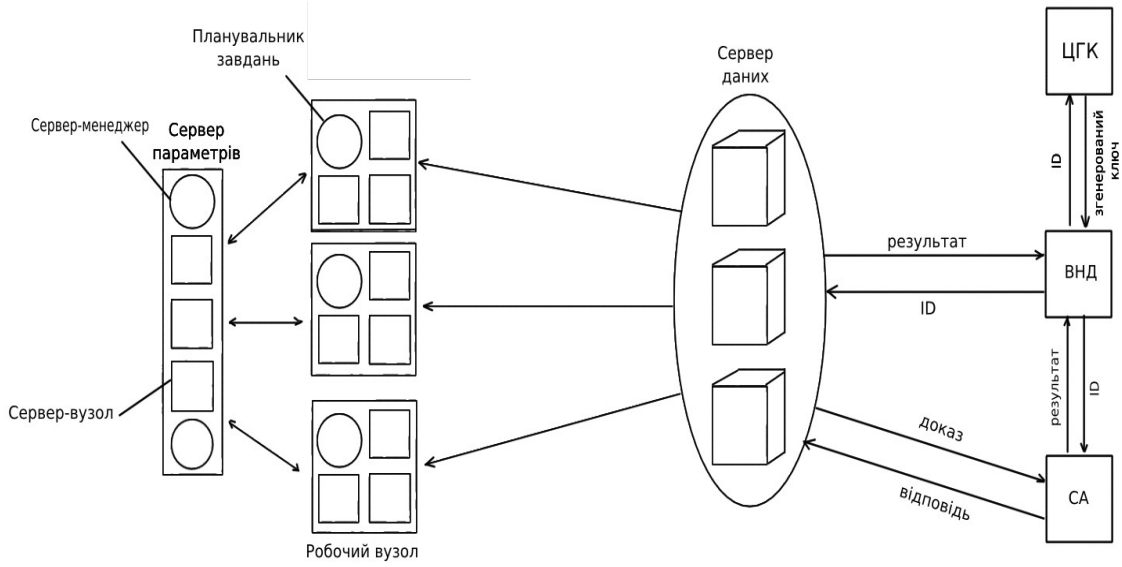


Рисунок 1 — Системна модель схеми РМН-ПЦД

Далі роз’ясню задачі та повноваження кожного компоненту даної системної моделі:

- власник НД (ВНД) — відповідає за збір навчальних даних та їх завантаження на сервер даних, ці дані можуть приходити з різних пристроїів;
- сервер даних (СД) — відповідає за надання хмарних сервісів та зберігає навчальні дані, при прийомі запиту від СА, він повинен довести цілісність та збереженність НД, після чого генерує та відправляє назад відповідь;
- стороній аудитор (СА) — перевіряє цілісність навчального НД, які зберігаються на СД, як написано в характеристиці СД, за допомогою відправки запиту та прийому відповіді на рахунок цілісності цього НД, а також він може проводити публічний аудит за допомогою відкритого ключа власника НД [2];
- центр генерування ключів (ЦГК) — генерує приватний та відкритий ключі, й після ідентифікації власника НД генерує частковий приватний ключ для нього з ідентифікатором та мастер-ключем, тобто ЦГК керує частковим ключем власника НД.

Якщо брати в розрахунок дану системну модель, то дана схема РМН-ПЦД може бути застосована для прогнозування, рекомендації чи класифікації в розподілених алгоритмах МН.

Візьму до прикладу додаток, що надає рекомендації по рекламі, який прекрасно ілюструє роботу даної моделі.

Допустимо, що в додатку ВНД є користувачі, котрі реєструються у веб-додатку, СД являється сервіс Amazon, котрий зберігає дані користувача, СА — якийсь довіренний аудитор, що перевіряє цілісність даних, а ЦГК — розробник додатка, котрий генерує ключ для користувача.

Алгоритм роботи даного рішення системної моделі:

1) Запуск алгоритму де відбувається генерація та передача приватних/відкритих ключів, а також задання початкових параметрів [3].

2) Збір НД про натискання на ту, чи іншу рекламу, та передача НД у СД.

3) Відбувається запит від СА до СД з питанням про цілісність НД та отримує відповідь.

4) Відправка з СА результатів свого запиту користувачам та робочим вузлам, якщо результат буде негативний, тобо дані будуть пошкоджені або втрачені, то робочі вузли працювати не будуть.

5) Робочі вузли підтягують дані із СД та запускають алгоритм розподіленого проксимального градієнту із затримкою блоку для отримання параметрів, після чого результати відправляють на сервер параметрів. Таким чином, відбувається оновлення вхідних параметрів на сервері параметрів, параметрами робочих вузлів.

6) Додаток надає рекомендації по рекламі користувачам на основі результатів навчання.

Список літератури

1. K. He, C. Huang, J. Shi, and J. Wang, ‘‘Public integrity auditing for dynamic regenerating code based cloud storage,’’ *in Proc. IEEE Symp. Comput. Commun. (ISCC)*, 2016, ст. 581–588.

2. W. Shen, J. Qin, J. Yu, R. Hao, and J. Hu, ‘‘Enabling identity-based integrity auditing and data sharing with sensitive information hiding for secure cloud storage,’’ *IEEE Trans. Inf. Forensics Security, vol. 14, no. 2,* 2019, ст. 331–346

3. Y. Yu, M. H. Au, G. Ateniese, X. Huang, W. Susilo, Y. Dai, and G. Min, ‘‘Identity-based remote data integrity checking with perfect data privacy preserving for cloud storage,’’ *IEEE Trans. Inf. Forensics Security, vol. 12, no. 4,* 2017, ст. 767–778

Додаток Г

ПРЕЗЕНТАЦІЯ ДО ЗАХИСТУ КВАЛІФІКАЦІЙНОЇ РОБОТИ

Хмельницький національний університет
Факультет інформаційних технологій
Кафедра комп’ютерної інженерії та інформаційних систем

Симак Денис Олександрович

Метод та система розподілених обчислень із використанням асиметричних алгоритмів шифрування

Науковий керівник — к.т.н. Каштальян А.С.

Мета та задачі дослідження

Метою дипломної роботи є аналіз та реалізація рішень застосування методів та систем розподілених обчислень із використанням асиметричних алгоритмів шифрування.

Об’єктом дослідження є розподілені обчислення із використанням алгоритмів асиметричного шифрування та їх методи реалізації у сучасних системах.

Предметом дослідження є моделі розподілених обчислень із використанням алгоритмів асиметричного шифрування.

Мета та задачі дослідження

Завданнями роботи є:

- провести аналіз систем розподілених обчислень;
- дослідити існуючі методи шифрування інформації;
- провести аналіз та дослідити відомі методи розподіленого машиного навчання;
- провести аналіз способів витоку конфіденційної інформації при машиному навчанні;
- дослідити існуючі методи запобігання та захисту конфіденційної інформації;
- підвести підсумки по розглянутій інформації та про необхідність створення та розробки нової системи;
- дослідити застосування алгоритмів асиметричного шифрування для захисту потоків даних;
- застосувати реалізовану систему розподілених обчислень із застосуванням алгоритмів асиметричного шифрування.

Наукова новизна отриманих результатів

- удосконалено метод системи розподілених обчислень за допомогою асиметричних алгоритмів шифрування, який на відміну від інших, проводить додатковий аудит для перевірки цілісності даних.

Практичне значення отриманих результатів

Полягає у вдосконалених моделях систем розподілених обчислень для використання їх для машинного навчання зі збереженням цілісності даних

Актуальність роботи полягає в аналізі вже існуючих методів та розробці нових рішень застосування методів та систем розподілених обчислень із використанням асиметричних алгоритмів шифрування.

Основні вимоги до СРО

Прозорість полягає в тому, що СРО повинні бути однорідним об’єктом для користувачів системи, а не просто набором автономних ПК, які між собою взаємодіють.

Масштабування є однією з найважливіших властивостей СРО, через те, що в ПЗ, яке керує всією взаємодією компонетів, в основному є обмеження на загальну кількість обчислювальних вузлів системи. В свою чергу, відкритість системи дає змогу взаємодіяти з іншими відкритими системами.

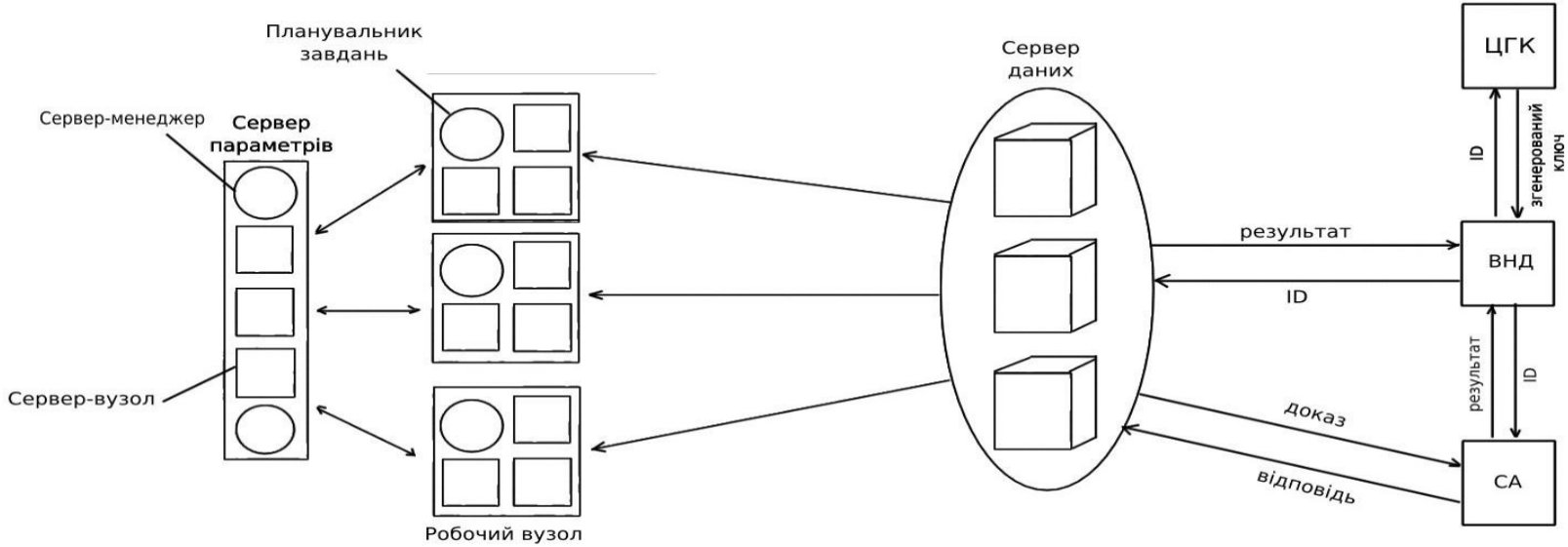
Надійність системи, заключається в такому показнику як відмовостійкість, який визначає можливість роботи над певними задачами, яка задала програма, після виникнення якоїсь несправності.

Недоліки існуючих методів

Недоліками, є низька ступінь захищенності пристроїв (вони дають відкритий доступ до своїх ресурсів, і збільшується ризик взлому або зараження), збільшення потреби в потужностях кожного пристрою (обчислювальний вузол бере на себе функції, і клієнта, і сервера), можлива гетерогенність апаратного та програмного забезпечення, пошук доступних ресурсів (кожному вузлу потрібно самому собі шукати ресурси для обчислень).

Модель системи розподіленого машинного навчання на основі схеми перевірки цілісності даних з використанням асиметричних алгоритмів шифрування

Схема РМН-ПЦД складається з чотирьох компонентів: серверу даних, стороннього аудитора, власника НД та центру генерації ключів, які зображенні на рисунку



Задачі та повноваження кожного компоненту даної системної моделі:

ВНД — відповідає за збір навчальних даних та їх завантаження на сервер даних, ці дані можуть приходити з різних пристроїів;

СД — відповідає за надання хмарних сервісів та зберігає навчальні дані, при прийомі запиту від СА, він повинен довести цілісність та збереженність НД, після чого генерує та відправляє назад відповідь;

СА — перевіряє цілісність навчального НД, які зберігаються на СД, як написано в характеристиці СД, за допомогою відправки запиту та прийому відповіді на рахунок цілісності цього НД, а також він може проводити публічний аудит за допомогою відкритого ключа власника НД;

ЦГК — генерує приватний та відкритий ключі, й після ідентифікації власника НД генерує частковий приватний ключ для нього з ідентифікатором та мастер-ключем, тобто ЦГК керує частковим ключем власника НД.

Алгоритм роботи даного рішення системної моделі:

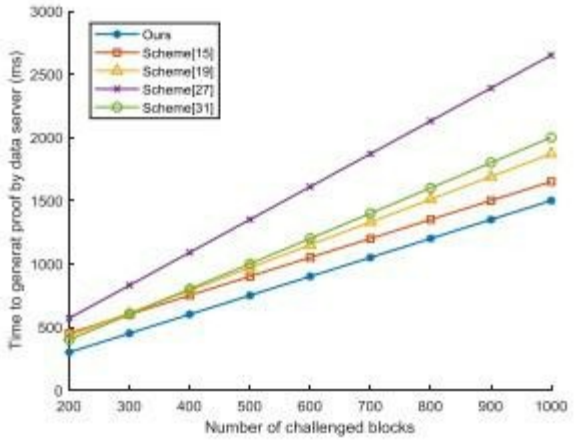
1) запуск алгоритму де відбувається генерація та передача приватних/відкритих ключів, а також задання початкових параметрів;
2) збір НД та передача НД у СД;
3) відбувається запит від СА до СД з питанням про цілісність НД та отримує відповідь;
4) відправка з СА результатів свого запиту користувачам та робочим вузлам, якщо результат буде негативний, тобо дані будуть пошкоджені або втрачені, то робочі вузли працювати не будуть;
5) робочі вузли підтягують дані із СД та запускають алгоритм розподіленого проксимального градієнту із затримкою блоку для отримання параметрів, після чого результати відправляють на сервер параметрів. Таким чином, відбувається оновлення вхідних параметрів на сервері параметрів, параметрами робочих вузлів;
6) додаток надає результати роботи користувачам на основі результатів навчання.

Проходить весь процес перевірки на персональному ноутбуці з AMD Ryzen 5 3500U та 8Гб оперативної пам’яті. Основний алгоритм агрегації виконується за допомогою бібліотеки GUN Multiple Precision Arithmetic (GMP) версії 6.1.2. Ми вибираємо криву MNT типу d з бібліотеки PBC. Встановлюємо, що довжина H1 дорівнює 175. Усі експериментальні результати представляють середнє значення 20 випробувань.

У моєму експерименті Y позначає довжину кожного блоку даних, EH позначає операцію потужності на групах H1 і H2, EZ позначає операцію потужності на числовому полі Zp, MH вказує операцію множення на групі H1 і H2, MZ вказує на множення операція над числовим полем Zp, OZ вказує операцію додавання в числовому полі Zp, та HASH позначає операцію обчислення хеш-значення.

В схемі аудиту публічної вибірки вартість обчислень в основному відбувається на СД, СА та ВНД.

Графік порівняння часу генерації доказів СД



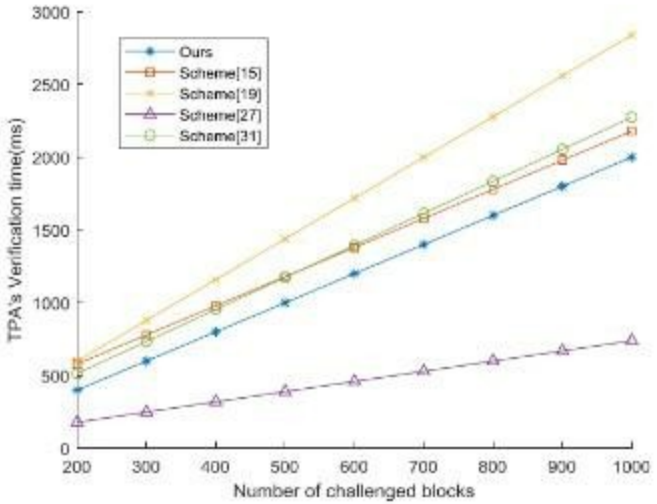
У всьому процесі перевірки СД виконує обчислення для створення доказів. Тому імітую доказ часу генерації під час перевірки цілісності даних на СД. Припущу, що кількість блоків для файлу рівно 10000, тобто m = 10000, результат зображений на рисунку.

На даному графіку відображений час генерації доказів лінійно збільшується з збільшенням кількості блоків.

Причина полягає в тому, що СД накопичує тільки блоки, що мають заперечення, та різноманітні відповідні підписи.

Як показано на графіку вартість обчислень СД в моїй схемі меньше

Графік порівняння часу перевірки доказу CA



Під час усього процесу перевірки CA виконує обчислення для перевірки доказу. Імітую час перевірки доказу під час перевірки цілісності даних на CA.

Припущу, що кількість блоків даних для файлу рівняється 10000, тобто m = 10000, результат зображений на рисунку.

На даному графіку відображений час перевірки доказу лінійно збільшується з кількістю блоків.

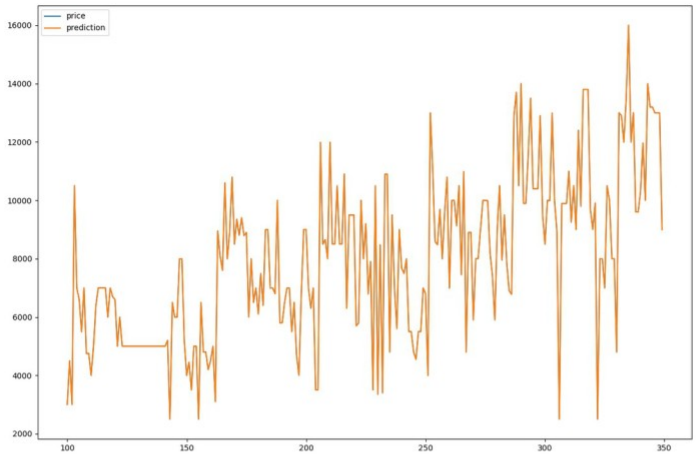
Причина в тому, що розрахунок для рівняння верифікації пов’язаний тільки з кількістю блоків, що мають заперечення.

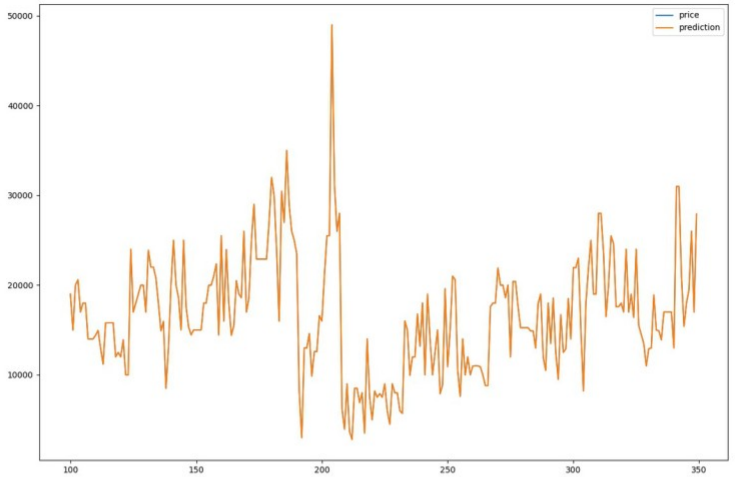
Як показано на графіку вартість обчислень CA в схемі РМН-ПЦД меньша.

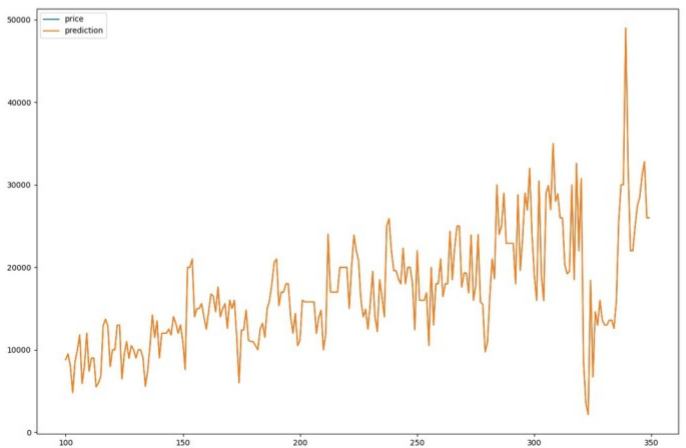
Порівняння обчислень ВНД

	ВНД
РМН-ПЦД	$\frac{sizeof(L)}{O}(M_H+2E_H+HASH)$
1	$\frac{sizeof(L)}{O}(M_H+2E_H+HASH)$
2	$\frac{sizeof(L)}{O}(M_H+2E_H+HASH)$
3	$\frac{sizeof(L)}{O}(M_H+M_Z+HASH+A_Z)$
4	$\frac{sizeof(L)}{O}(M_H+2E_H+HASH)$

Під час усього процесу перевірки ВНД виконує обчислення для створення підписів. В Таблиці 2 проаналізовано та порівняно обчислювальну вартість ВНД при обчисленні підписів між схемами 1, 2, 3, 4 і схемою РМН-ПЦД. Згідно Таблиці 2, видно, що з точки зору обчислення вартості генерації підписів схема РМН-ПЦД рівна схемам 1, 2, 4 і трохи більше, ніж схема 3.







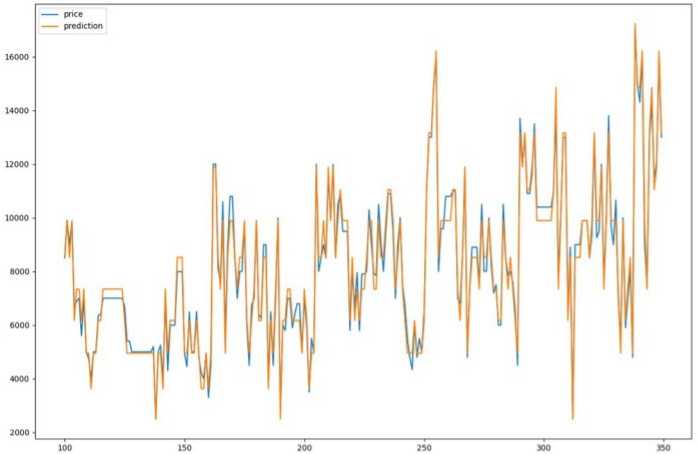
1) Лінійна регресія

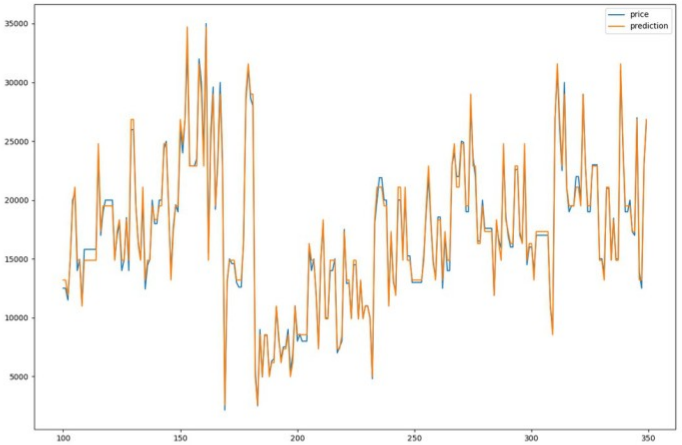
При обчисленнях на 1 робочому вузлі: точність моделі 100% та час навчання 29.18 сек.

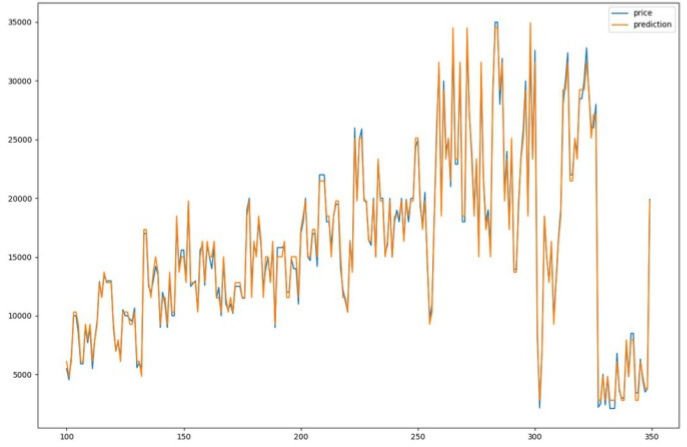
При обчисленнях на 2 робочих вузлах: точність моделі 100% та час навчання 14.72 сек.

При обчисленнях на 4 робочих вузлах: точність моделі 100% та час навчання 13.66 сек.

Дивлячись на результати можна з впевненістю сказати, що модель працює добре та при збільшені кількості робочих вузлів зменшується час навчання, що ми і планували досягти. У даному випадку при збільшені кількості робочих вузлів з 1 до 2, час навчання скоротився удвічі, а збільшивши до 4, в порівнянні з 2, не дало великої різниці.







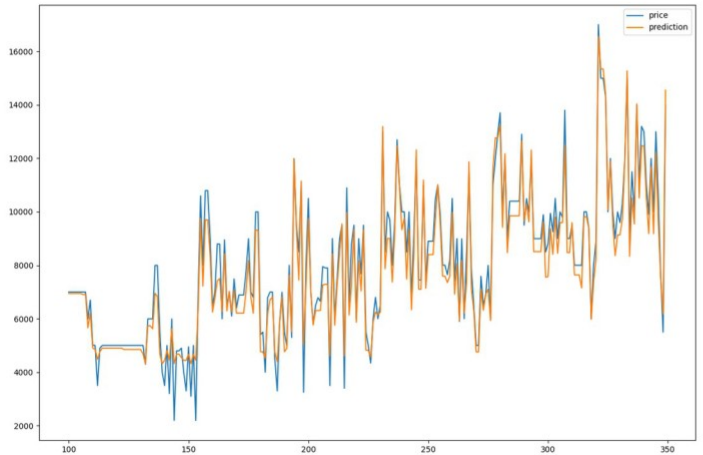
2) Регресія з використанням дерева рішень

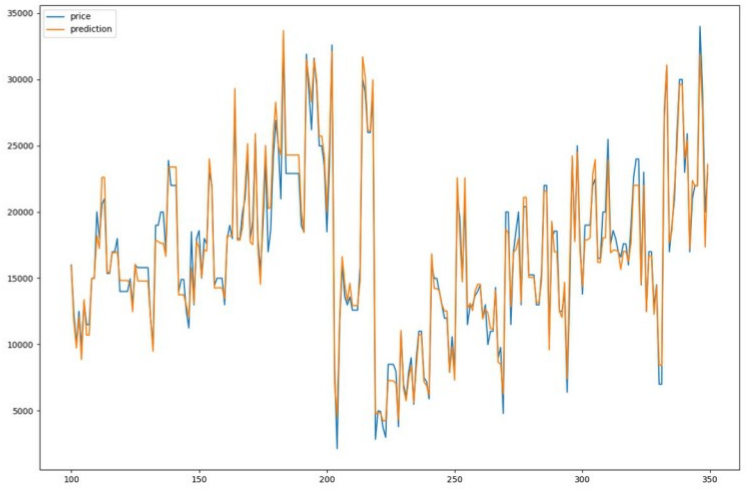
При обчисленнях на 1 робочому вузлі: точність моделі 98.84% та час навчання 19.83 сек.

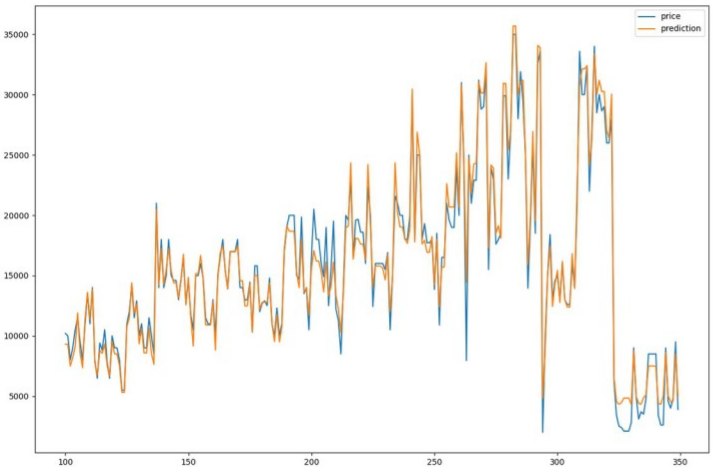
При обчисленнях на 2 робочих вузлах: точність моделі 98.9% та час навчання 16.40 сек.

При обчисленнях на 5 робочих вузлах: точність моделі 99.1% та час навчання 15.12 сек.

Дивлячись на результати можна з впевненістю сказати, що модель працює добре та при збільшені кількості робочих вузлів зменшується час навчання, що ми і планували досягти. У даному випадку при збільшені кількості робочих вузлів з 1 до 2, час навчання скоротився на 19%, а збільшивши до 5, в порівнянні з 2, не дало великої різниці, усього лише 8%.







3) Регресія з використанням Random forest

При обчисленнях на 1 робочому вузлі: точність моделі 96.2% та час навчання 20.06 сек.

При обчисленнях на 2 робочих вузлах: точність моделі 96.1% та час навчання 17.58 сек.

При обчисленнях на 5 робочих вузлах: точність моделі 96.3% та час навчання 17.34 сек.

Дивлячись на результати можна з впевненістю сказати, що модель працює добре та при збільшені кількості робочих вузлів зменшується час навчання, що ми і планували досягти. У даному випадку при збільшені кількості робочих вузлів з 1 до 2, час навчання скоротився майже на 13%, а збільшивши до 5, в порівнянні з 2, майже не дало ніякої різниці 1.37%.

Anti-Plagiarism v-15.257

Максимальное совпадение с одним документом 0.0%

Словари проверки: en_US, ru_RU, ua_UA. **Ошибок в документах: 8%**

ID: 103691 Название: Метод та система розподілених обчислень із використанням асиметричних алгоритмів шифрування Добавлено в БД: 2022-05-19 Авторы: Симак Д.О. Руководители: Каштальян А.С. Консультанты: Оппоненты:	Документ		Суммарное совпадение по Базе Данных	
	Символы	Лексемы	Символы	Лексемы
	94022	732	271 (0%)	4 (1%)

Источник плагиата

ID	Описание	Наличие плагиата в документе	
		Символы	Лексемы

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Дипломник: Симак Денис Олександрович

Тема: Метод та система розподілених обчислень із використанням асиметричних методів шифрування

Спеціальність: 123 «Комп'ютерна інженерія»

Обсяг дипломної роботи:

Кількість сторінок записки 82 с.

1. Короткий зміст роботи та прийнятих рішень: Метою кваліфікаційної роботи є розробка системи розподілених обчислень із забезпеченням захисту даних на основі асиметричних методів шифрування.

2. Висновок про відповідність роботи дипломному завданню: Робота повністю відповідає поставленому завданню.

3. Характеристика виконання кожного розділу, ступінь використання останніх досягнень науки і техніки і передових методів роботи: В першому розділі було аналізовано відомі методи систем розподілених обчислень, виявлено проблеми безпеки та конфіденційності СРО та способи їх вирішення. В другому розділі було розглянуто методи застосування розподіленого машинного навчання у системі розподілених обчислень з використанням асиметричних алгоритмів шифрування. В третьому розділі було виконано проектування моделі системи розподіленого МН на основі схеми перевірки та цілісності даних з використанням асиметричних алгоритмів шифрування. Розроблено метод розподіленого навчання із використанням асиметричних методів шифрування для захисту від несанкціонованого доступу до даних. В четвертому розділі була запропонована та розглянута розподілена схема перевірки цілісності даних, яка орієнтована на розподілене машинна навчання для побудови серверу параметрів.

4. Позитивні сторони роботи: ретельний аналіз існуючих рішень.

Негативні сторони роботи: розроблений метод не має достатнього обґрунтування, реалізація системи не розкриває розроблений метод паралельного навчання.

6. Оцінка графічного оформлення та пояснювальної записки роботи: Пояснювальна записка оформлена коректно, згідно діючих стандартів оформлення документації.

7. Відгук про роботу в цілому: В загальному робота виконана на задовільному науково-технічному рівні.

8. Інші зауваження: _____

9. Оцінка дипломної роботи: задовільно/E.

Рецензент (прізвище, ім'я, по батькові, посада, місце роботи) _____

Мартинюк Валерій Володимирович, зав. каф. АКІТ, д.т.н., проф.

"18" 05 2022 р. _____ (підпис)

РІШЕННЯ ЕКСПЕРНОЇ КОМІСІЇ

КАФЕДРИ КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОМАЦІЙНИХ СИСТЕМ

ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: Метод та система розподілених обчислень із використанням асиметричних алгоритмів шифрування

Автор: Симак Денис Олександрович

Спеціальність: 123 – Компютерна інженерія

Освітня програма: освітньо-наукова

Науковий керівник: Каштальян А.С., к.т.н, доцент

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту.	відповідає
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	

Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

1) запозичення розміщені в розділах аналізу існуючих аналогів та прототипів, які не описують безпосередньо авторське дослідження і не стосуються результатів роботи;
2) усі запозичення фрагментарні, або мають належним чином оформленні посилання;
3) окремі виявлені збіги є загальновживаними фразами або виразами, про що свідчить посилання системи на збіг з 23 джерелами з Інтернету та 71 джерелами з бібліотеки;
4) всі зафіксовані системою ознаки модифікації тексту відносяться до комбінування латинських символів зі україномовними скороченнями індексів в формулах, що не є модифікацією тексту.

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ ідентичності/схожості, складає 0.88% і адресується до 93 першоджерел, що, з урахуванням наведених обґрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.

Керівник роботи _____ А.С. Каштальян

Гарант ОП _____ О. С. Савенко

Завідувач кафедри КІПСч _____ Т. О. Говорущенко

UNICHECK

Ім'я користувача:
Кафедра КІ

ID перевірки:
1011255708

Дата перевірки:
19.05.2022 18:51:06 EEST

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
19.05.2022 18:52:15 EEST

ID користувача:
100005591

Назва документа: Денис Симак_Метод та система розподілених обчислень із використанням асиметричних а...

Кількість сторінок: 87 Кількість слів: 15534 Кількість символів: 118060 Розмір файлу: 1.27 MB ID файлу: 1011145923

0.88%
Схожість

Найбільша схожість: 0.55% з джерелом з Бібліотеки (ID файлу: 1011139258)

0.19% Джерела з Інтернету	23	Сторінка 89
0.75% Джерела з Бібліотеки	71	Сторінка 89

0% Цитат

Не знайдено жодних цитат

Не знайдено жодних посилань

0%
Вилучень

Немає вилучених джерел

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи	58

Завідувачу кафедри КІІС
д-р.техн.наук, проф. Говорущенко Т. О.

Симак Денис Олександрович
ПІБ здобувача вищої освіти

ФПКТС, 2 курсу, групи КІ2м-19-1

ЗАЯВА

З правилами чинного Положення «Про дотримання академічної доброчесності в Хмельницькому національному університеті» від 26.09.2020 (зі змінами від 26.11.2020), згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування заходів дисциплінарної та академічної відповідальності, ознайомлений (а). Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на плагіатоповіщений (а) та надаю свою згоду на обробку та збереження університетом моєї роботи в інституційному репозитарії університету.

Також надаю університету право на передачу моєї роботи для обробки та збереження в базах даних програмно-технічних засобів (Unicheck та Anti-Plagiarism) та використання роботи для виявлення плагіату в інших роботах, які перевіряються програмно-технічними засобами та користувачами, що мають доступ до цих програмно-технічних засобів, виключно в обмежених цілях для виявлення плагіату в текстах робіт.

Робота для перевірки університетом надається в друкованому та електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

10.05.2022р.
дата

підпис