МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Західноукраїнський національний університет
Факультет комп'ютерних інформаційних технологій
Кафедра кібербезпеки

ШЕСТЕРИНА Сергій Вадимович

Захищена система зберігання даних на основі надлишкової системи залишкових класів та хмарних сервісів / Secure Data Storage System Based on Redundant Residue Class and Cloud Services

спеціальність: 125 – Кібербезпека

освітньо-професійна програма – Кібербезпека

Кваліфікаційна робота

Виконав студент групи
КБм -21
С.В. Шестерина

Науковий керівник
к.т.н., доцент Н.Г.Яцків

Кваліфікаційну роботу
допущено до захисту:

«____» _____ ___2023 р.

Завідувач кафедри

_____ **В.В.Яцків**

ТЕРНОПІЛЬ – 2023

Факультет комп'ютерних інформаційних технологій
Кафедра кібербезпеки
Освітній ступінь «магістр»
спеціальність: 125 – Кібербезпека
освітньо-професійна програма – Кібербезпека

ЗАТВЕРДЖУЮ
Завідувач кафедри

_____ В.В.Яцків
«____» _____ 2022 року

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

ШЕСТЕРИНА СЕРГІЙ ВАДИМОВИЧ
(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи:
Захищена система зберігання даних на основі надлишкової системи залишкових класів та хмарних сервісів / Secure Data Storage System Based on Redundant Residue Class and Cloud Services
керівник роботи к.т.н., доцент Н.Г. Яцків

затверджені наказом по університету від 1 грудня 2022 року № 491

2. Строк подання студентом закінченої кваліфікаційної роботи 1 грудня 2023 р.

3. Вихідні дані до кваліфікаційної роботи: завдання на кваліфікаційну роботу студента, наукові статті, технічна література.

4. Основні питання, які потрібно розробити:
– визначити типи хмарних сервісів;
– провести аналіз хмарних провайдерів;
– дослідити поняття надлишкової системи зберігання даних;
– розробити структуру захищеної системи зберігання даних;
– розглянути поняття шифрування даних;
– реалізація захищеної системи зберігання даних на основі надлишкової системи залишкових класів та хмарних сервісів.

5. Перелік графічного матеріалу у роботі:
– типи хмарних сервісів;
– провайдери хмарних сервісів їх частка на ринку;
– структура системи надійного зберігання даних;
– представлення різних компонентів системи та їх взаємодії;
– візуалізація роботи системи.

6. Консультанти розділів кваліфікаційної роботи

	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання 8 грудня 2022 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строки виконання етапів кваліфікаційної роботи	Примітка
1	Аналіз хмарних сервісів	12.2022 р. – 03.2023 р.	
2	Структура захищеної системи зберігання даних	03.2023 р. – 05.2023 р.	
3	Розробка та тестування системи надійного зберігання даних	05.2023 р. – 11.2023 р.	

Студент _____ Шестерина С.В.
(підпис)

Керівник роботи _____ к.т.н., доцент Н.Г. Яцків
(підпис)

АНОТАЦІЯ

Магістерська робота на тему “Захищена система зберігання даних на основі надлишкової системи залишкових класів та хмарних сервісів” зі спеціальності 125 – кібербезпека написана обсягом 86 сторінок і містить 19 ілюстрацій, 1 додаток та 27 джерел за переліком посилань.

Робота присвячена розробці та аналізу інноваційної інфраструктури для забезпечення безпеки, доступності та надійності зберігання важливої інформації.

Дана робота меті розробку та вивчення інноваційного підходу до забезпечення безпеки, надійності та конфіденційності зберігання важливої інформації. Робота передбачає створення системи, яка використовує концепцію надлишковості залишкових класів для забезпечення надійності та конфіденційності інформації та інтеграція хмарних сервісів для резервного копіювання та додаткового шару безпеки.

У магістерській роботі використана комплексна методологія досліджень, охоплююча різноманітні наукові та технічні підходи для вивчення, розробки та оцінки захищеної системи зберігання даних.

Результати роботи можуть бути корисними для організацій, які працюють з важливою та конфіденційною інформацією, а також для дослідників, що цікавляться вдосконаленням сучасних методів захисту даних. Запропонована система може слугувати як базис для розробки та впровадження високопродуктивних та безпечних засобів зберігання інформації в сучасному цифровому середовищі.

Ключові слова: КРИПТОГРАФІЯ, КРИПТОАНАЛІЗ, ПРОСТІ ЧИСЛА, НАДЛИШКОВА СИСТЕМА ЗАЛИШКОВИХ КЛАСІВ, ШИФРУВАННЯ.

ABSTRACT

The master's thesis on "Secure data storage system based on the redundant residual class system and cloud services" in the specialty 125 - Cybersecurity is written on 86 pages and contains 19 illustrations, 1 appendices, and 27 sources in the list of references.

The work is devoted to the development and analysis of innovative infrastructure to ensure the security, availability, and reliability of important information storage.

This work is aimed at developing and studying an innovative approach to ensuring the security, reliability, and confidentiality of important information storage. The work involves the creation of a system that uses the concept of redundancy of residual classes to ensure the reliability and confidentiality of information and the integration of cloud services for backup and an additional layer of security.

The master's thesis uses a comprehensive research methodology that covers a variety of scientific and technical approaches to study, develop, and evaluate a secure data storage system.

The results of the work can be useful for organizations that work with important and confidential information, as well as for researchers interested in improving modern data protection methods. The proposed system can serve as a basis for the development and implementation of high-performance and secure information storage in the modern digital environment.

Keywords: CRYPTOGRAPHY, CRYPTANALYSIS, PRIMES, REDUNDANT SYSTEM OF RESIDUAL CLASSES, ENCRYPTION.

ЗМІСТ

ВСТУП

Актуальність роботи магістерської роботи. В умовах стрімкого росту обсягу цифрових даних та поширення кіберзагроз, важливість розробки та впровадження новаторських систем стає надзвичайно актуальною.

Однією з ключових особливостей даної роботи є використання підходу, який об'єднує надлишковість систем залишкових класів із перевагами хмарних сервісів. Це створює унікальну можливість підвищення надійності та доступності зберігання даних, забезпечуючи при цьому високий рівень конфіденційності та ефективність в управлінні інформацією.

У сучасному світі, де атаки на цифрові ресурси стають все більш виразними та досконалими, забезпечення безпеки даних стає завданням вищого порядку. Застосування шифрування та інших технік забезпечення конфіденційності дозволяє роботі виходити за межі звичайних технологічних рішень і висвітлювати проблематику кіберзахисту в контексті зберігання даних.

Використання хмарних сервісів розширює можливості системи, забезпечуючи гнучкість та масштабованість. Інтеграція з хмарними платформами дозволяє впроваджувати сучасні рішення для резервного копіювання, аналізу даних та ефективного управління інформаційними потоками.

Загальний контекст актуальності роботи також визначається стрімким розвитком обчислювальних технологій та постійним підвищенням вимог до зберігання та обробки даних. У цьому світлі, магістерська робота є необхідною та перспективною, вносячи свій вклад у подолання викликів інформаційної безпеки та оптимізації систем зберігання даних у сучасному цифровому середовищі.

Метою є розробка та підвищення ефективності системи зберігання даних, за рахунок використання надлишкової систем залишкових класів та хмарних сервісів.

Визначені наступні завдання, які потрібні для досягнення поставленої мети:

- проаналізувати хмарні сервіси для визначення можливості їх використання в захищеній системі зберігання даних;
- розробити систему зберігання даних, яка забезпечує надійність і конфіденційність даних;
- дослідити ефективність системи зберігання даних на основі НСЗК та хмарних сервісів;
- проведення емпіричних експериментів для тестування функціональності та ефективності розробленої систем;
- розробити рекомендації щодо впровадження системи зберігання даних в реальних умовах;

Об’єкт дослідження – процеси кодування та зберігання даних в хмарних сервісах.

Предметом дослідження – методи та алгоритми кодування та зберігання даних в хмарних сервісах, які базуються на надлишковій системі залишкових класів та використанні хмарних сервісів.

Наукова новизна отриманих результатів виявляється у наступних ключових аспектах:

- розроблена система використовує унікальний підхід до поєднання надлишковості залишкових класів із хмарними сервісами. Це відкриває нові можливості для підвищення надійності та доступності даних;
- використання надлишкової системи залишкових класів, як засобу шифрування конфіденційної інформації;

Практична цінність одержаних результатів виявляється у ряді практичних застосувань та можливостей для індустрії та організацій:

- розроблені та впроваджені методи шифрування дозволяють підвищити рівень безпеки та конфіденційності даних, що має важливе значення для організацій, які обробляють чутливу інформацію;
- інтеграція з хмарними сервісами дозволяє використовувати резервне копіювання, масштабування ресурсів та гнучке управління даними, що покращує продуктивність та доступність системи;

- адаптивність системи дозволяє легко впроваджувати нові технології та методи безпеки, що забезпечує готовність до змін та масштабування;
- реалізовано мобільний додаток на мові Kotlin для операційної системи Android шифрування та дешифрування файлів з подальшим зберіганням в хмарному сервісі Google Cloud за допомогою платформи Firebase.

Публікації та апробація КР.

1. Шестерина С.В. Аналіз хмарних сервісів. Матеріали науково-практична конференція молодих вчених, аспірантів та студентів «Кібербезпека та комп'ютерно-інтегровані технології» (КБКІТ - 2023), Тернопіль, 2023. – С.96-100.
2. Шестерина С.В. Структура захищеної системи зберігання даних. Матеріали науково-практичного симпозіуму «Захист інформації», Тернопіль, 2023. – С. 177-181.

1 АНАЛІЗ ХМАРНИХ СЕРВІСІВ

1.1 Що таке хмарні сервіси

Хмарні сервіси - це послуги, які надаються через Інтернет. Вони можуть використовуватися для зберігання, обробки та аналізу даних. Хмарні сервіси пропонують ряд переваг, зокрема масштабованість, доступність та економічність.

Хмарні служби полегшують передачу даних користувачів від зовнішніх клієнтів (наприклад, серверів користувачів, планшетів, настільних комп’ютерів, ноутбуків — будь-що на стороні користувачів) через Інтернет до систем провайдера та назад. Хмарні сервіси сприяють створенню хмарних додатків і гнучкості роботи в хмарі. Користувачі можуть отримати доступ до хмарних служб лише за допомогою комп’ютера, операційної системи та підключення до Інтернету.

Уся інфраструктура, платформи, програмне забезпечення або технології, до яких користувачі отримують доступ через Інтернет без необхідності завантажувати додаткове програмне забезпечення, можна вважати службами хмарних обчислень [1].

Хмарними службами повністю керують постачальники послуг хмарних обчислень . Вони надаються клієнтам із серверів постачальників, тому компанії не потрібно розміщувати програми на власних локальних серверах.

Послуги, які провайдер робить доступними для багатьох клієнтів через Інтернет, називаються публічними хмарними службами. Найбільшою перевагою використання загальнодоступних хмарних служб є можливість спільного використання ресурсів у масштабі, що дозволяє організаціям пропонувати співробітникам більше можливостей, ніж це було б можливо поодинці.

Послуги, які постачальник не робить загальнодоступними для корпоративних користувачів або абонентів, називаються приватними хмарними службами. Завдяки моделі приватних хмарних служб програми та дані стають доступними через власну внутрішню інфраструктуру організації. Платформа та програмне забезпечення обслуговують лише одну компанію та не надаються

зовнішнім користувачам. Компанії, які працюють з дуже конфіденційними даними, як-от у сфері охорони здоров'я та банківській сфері, часто використовують приватні хмари для використання розширених протоколів безпеки та розширення ресурсів у віртуалізованому середовищі за потреби.

У гібридному хмарному середовищі приватне хмарне рішення поєднується з публічними хмарними службами. Цей механізм часто використовується, коли організації потрібно зберігати конфіденційні дані в приватній хмарі, але хоче, щоб співробітники мали доступ до програм і ресурсів у загальнодоступній хмарі для щоденного спілкування та співпраці. Власне програмне забезпечення використовується для забезпечення зв'язку між хмарними службами, часто через єдину консоль керування IT [2].

Найчастіше хмарні провайдери надають три типи послуг, які на рисунку 1.1 представлені у вигляді піраміди.

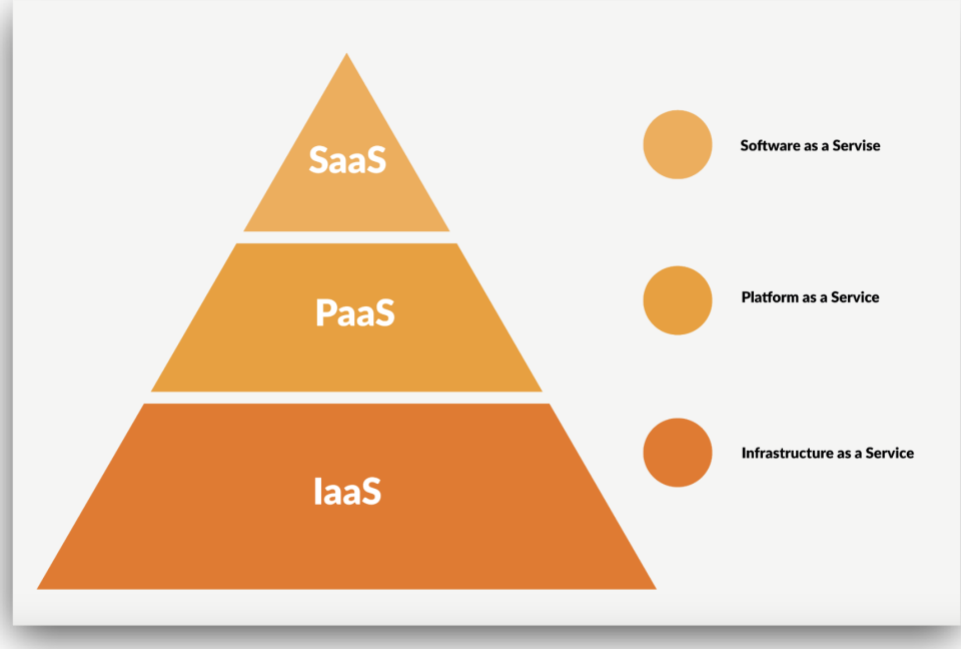


Рисунок 1.1 – Піраміда моделей хмарних сервісів

1. Інфраструктура як послуга (IaaS). Використовуючи IaaS, абоненти можуть спроектувати ціле середовище, налаштувавши віртуальну мережу, сегментовану від інших мереж. У цьому середовищі користувачі запускають

операційну систему та забезпечують обробку, зберігання, мережі та інші фундаментальні обчислювальні ресурси, необхідні для запуску програмного забезпечення в хмарній інфраструктурі. За допомогою IaaS абонент також може мати обмежений контроль над вибраними мережевими компонентами (наприклад, міжмережевими екранами хостів). Деякі постачальники також пропонуватимуть такі послуги, як моніторинг, автоматизація, безпека, балансування навантаження та стійкість зберігання.

2. Платформа як послуга (PaaS). Користувачі отримують доступ до інфраструктури з операційної системи. PaaS дозволяє користувачам розміщувати власні програми в хмарній інфраструктурі з мовами програмування, бібліотеками, службами та інструментами, які підтримуються постачальником. Абонент має контроль над розгорнутими програмами, даними та, можливо, налаштуваннями конфігурації для середовища розміщення програм. Але мережею, серверами, операційними системами та сховищем керує та контролює провайдер.

3. Програмне забезпечення як послуга (SaaS). Провайдери пропонують передплатникам використання свого програмного забезпечення, що працює в хмарній інфраструктурі, що означає, що програма може бути широко розповсюдженою та доступною. Поширені типи бізнес-технологій, розміщених постачальником SaaS, включають пакети продуктивності, програмне забезпечення для керування взаємовідносинами з клієнтами (CRM), програмне забезпечення для управління людськими ресурсами (HRM) і програмне забезпечення для керування даними. Користувачі мають можливість отримати доступ до програми(ів) через інтерфейс програми або інтерфейс тонкого клієнта, наприклад веб-браузер. За допомогою цієї послуги передплатники мають лише доступ до програмного забезпечення та його використання. Провайдер займається всім іншим: керуванням і контролем мережі, серверів, операційних систем, сховищ, віртуалізації, даних, проміжного програмного забезпечення та навіть можливостей окремих програм. Програми SaaS зазвичай розроблені таким чином, щоб бути простими у використанні для широкої аудиторії [3].

Хмарні сервіси можна поділити на категорії за формою подання на більш деталізовані типи в межах кожної категорії. Ось розширений огляд категорій хмарних сервісів:

- як сервіс зберігання даних (Storage-as-a-Service), сервіс зберігання даних, є моделлю обчислення в хмарному середовищі, яка дозволяє користувачам зберігати та управляти своїми даними через Інтернет, замість того, щоб обслуговувати власне обладнання та інфраструктуру;

- сервіс баз даних (Database-as-a-Service), модель хмарних обчислень, яка надає доступ до бази даних як до послуги. Це означає, що користувачі не повинні встановлювати, адмініструвати або обслуговувати власне обладнання або програмне забезпечення бази даних. Замість цього, вони просто оплачують використання бази даних за потреби;

- інформаційний сервіс (Information-as-a-Service), є концепцією, яка вказує на надання доступу до інформації через хмарні технології чи мережі Інтернету. Це може включати в себе різні види інформації, такі як дані, аналітику, новини, документацію та інше. Основною ідеєю є те, що користувачі можуть отримувати доступ до потрібної їм інформації в режимі онлайн, без необхідності власної установки та обслуговування інфраструктури;

- сервіс управління процесами (Process-as-a-Service) відноситься до хмарних послуг, які надають можливість організаціям автоматизувати та оптимізувати свої бізнес-процеси через Інтернет. Цей тип послуги дозволяє клієнтам використовувати та управляти бізнес-процесами без необхідності власного розроблення та підтримки програмного забезпечення для автоматизації;

- додаток як сервіс (Application-as-a-Service) концепція, що вказує на надання доступу до програмного забезпечення через хмарні технології чи мережі Інтернету. Цей тип сервісу дозволяє користувачам використовувати програми чи додатки без необхідності їх встановлення та підтримки на власних пристроях чи серверах;

- сервіс-платформа (Platform-as-a-Service), є моделлю хмарного обчислення, яка надає платформу для розробки, тестування та впровадження

програмного забезпечення без необхідності власної установки та управління інфраструктурою;

- сервіс-інтеграція програм (Integration-as-a-Service) – це модель хмарних обчислень, яка надає послуги інтеграції програм як послуги. Це означає, що клієнти не повинні розробляти, розгортати та підтримувати власні рішення для інтеграції програм, а просто оплачують використання цих послуг за потреби;

- сервіс-безпека (Security-as-a-Service) – це модель хмарних обчислень, яка надає послуги безпеки як послуги. Це означає, що клієнти не повинні розробляти, розгортати та підтримувати власні рішення для безпеки, а просто оплачують використання цих послуг за потреби;

- сервіс адміністрування та управління (Management/Governace-as-a-Service) – модель хмарних обчислень, яка надає послуги адміністрування та управління як послуги. Це означає, що клієнти не повинні інвестувати в власний персонал або ресурси для адміністрування та управління своїми хмарами, а просто оплачують використання цих послуг за потреби;

- сервіс інфраструктур (Infrastructure-as-a-Service) є однією з моделей хмарного обчислення, яка надає користувачам доступ до віртуальних обчислювальних ресурсів через Інтернет. IaaS дозволяє орендувати віртуальне обладнання, таке як сервери, мережеві ресурси, зберігання та інші компоненти інфраструктури, без необхідності власної фізичної інфраструктури;

- сервіс-дані (Desktopas a Service) − це хмарний сервіс, який надає віртуальні робочі столи користувачам через Інтернет. У цьому контексті "робочий стіл" означає віртуальне робоче середовище, яке включає в себе операційну систему, додатки та інші ресурси, доступні з будь-якого пристрою, підключеного до Інтернету;

- сервіс робоче місце (Workspace-as-a-Service) – визначає хмарний сервіс, який надає користувачам доступ до повного робочого оточення через Інтернет. Це включає в себе різні ресурси, такі як віртуальні робочі столи, додатки, файли, комунікації та інші елементи, необхідні для виконання роботи.

WaaS забезпечує інтегроване та однорідне робоче середовище, доступне з різних пристроїв та місць. Користувачі можуть отримувати доступ до свого

робочого місця, незалежно від фізичного розташування, що дозволяє робити роботу більш мобільною та гнучкою [4].

1.2 Аналіз найбільших хмарних провайдерів: плюси та мінуси

У другому кварталі 2023 року глобальні витрати на послуги хмарної інфраструктури зросли на $10 млрд порівняно з другим кварталом 2022 року, в результаті чого загальні витрати за три місяці, що закінчилися 30 червня, склали $64,8 млрд. Якщо поглянути на останні дванадцять місяців, то ринок хмарних технологій - це ринок вартістю 247 мільярдів доларів, що пояснює, чому за нього точиться така запекла боротьба. Як показано на рисунку 1.2, на Amazon, Microsoft і Google припадає майже дві третини доходів від хмарної інфраструктури в минулому кварталі, а вісім найбільших постачальників контролюють майже 80 відсотків ринку [5].

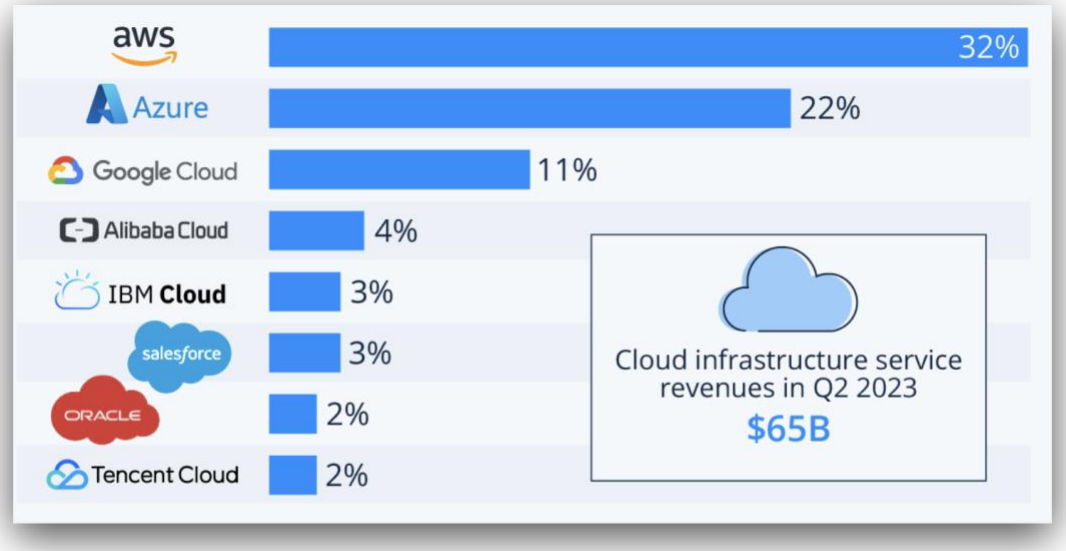


Рисунок 1.2 – Діаграма часток світового ринку хмарних провайдерів в першому кварталі 2023 року

Розгляньмо 5 найпопулярніших хмарних провайдерів: що вони пропонують, що роблять краще за інших та що може зіграти не на їхню користь.

Amazon Web Services. Те, що це найпопулярніший хмарний провайдер підтверджує не лише частка ринку, а і клієнти. До прикладу, сервісами AWS користується понад 7500 державних служб США. Amazon Web Services пропонують понад 200 послуг та покривають 31 регіон, кожен з яких із кількома зонами доступності. Серед послуг: обчислення, зберігання, бази даних, мережі, аналітика, машинне навчання, безпека тощо.

До переваг даного провайдера можна віднести:

1. Широкий вибір послуг та інструментів. Завдяки цьому компанії можуть повністю закрити свої хмарні потреби та управляти всіма сервісами в межах однієї платформи. Це може з економити час та ресурси, що дуже важливо для бізнесу.

2. Безпека. Amazon турбується про свою інфраструктуру та дані користувачів, а також пропонує сервіси захисту, шифрування та сертифікати відповідності.

3. Глобальна інфраструктура. AWS має дата центри у різних регіонах, тож компанії можуть розгортати свої програми та сервіси якнайближче до своїх користувачів. Це зменшує затримку та підвищує продуктивність.

4. Гнучкість і доступність. Можна вибрати бажану ОС, мову програмування, платформу веб застосунків та інше. Це полегшує міграцію в хмару. Також AWS пропонує інфраструктуру на вимогу, тож немає обмежень щодо ресурсів, які можна використати. Користувач платить лише за використані ресурси.

Навіть у такого популярного провайдера присутній ряд недоліків, такі як:

- Складність. Щоб ефективно працювати з AWS, необхідно мати відповідні навички. Велика кількість послуг та конфігурацій перетворить керування AWS на справжній виклик.
- Ціноутворення. Хоч вартість є однією з переваг AWS, ціноутворення може бути складним через велику кількість послуг, ціни на які формуються за різним принципом. Щоб уникнути зайвих витрат, необхідно приділяти більше часу для їхнього планування та моніторингу.

- Обмеження Amazon EC2. Твій регіон може вплинути на кількість доступних ресурсів. Також AWS не дає новим користувачам використовувати багато ресурсів. Це необхідно для безпеки, щоб зловмисники не могли використати його ресурси для хакерських атак.

Microsoft Azure. Хмарні сервіси від Microsoft. Багато бізнесів працюють з продуктами компанії, тож обирають цього провайдера для своїх потреб у клауді. Azure теж пропонує понад 200 хмарних сервісів, які дозволяють створювати, запускати та керувати програмами в різних хмарах.

Перевагами даного провайдера є:

1. Послуги на різний смак. Пропозиція Azure включає обчислення, сховище, бази даних, мережу, аналітику, AI, IoT тощо. Тож компанії точно знайдуть послугу, яка задовольнить їхні унікальні потреби.

2. Можливості гібридної хмари. Azure фокусується на гібридних хмарних рішеннях, а це означає гарну інтеграцію між локальною інфраструктурою та хмарною. Завдяки цьому компанії можуть мігрувати в хмару поступово.

3. Інтеграція з екосистемою Microsoft. Хмарні сервіси компанії добре інтегруються з її іншими продуктами та сервісами, наприклад, Windows Server, Active Directory та Office 365. Це спрощує керування, а також пропонує користувачам знайоме середовище.

4. Глобальний дата-центр. Як і AWS, Azure має мережу дата-центрів по всьому світу. Це забезпечує надійність, масштабованість і доступність послуг.

5. Безпека та відповідність. Azure надає надійні функції безпеки, зокрема шифрування даних, контроль доступу, розвідку загроз безпеки та сертифікати відповідності.

Недоліки теж присутні:

1. Відсутність доступу в разі збою. Якщо в основній системі станеться збій, користувачі не матимуть офлайн-доступу до неї.

2. Незрозумілий інтерфейс користувача. Це ускладнює користування платформою для тих, хто немає серйозного технічного досвіду. Також потрібні глибокі знання для того, щоб виконувати складні завдання.

3. Azure може бути дорогим, але в цьому випадку є вихід. Компанія пропонує інструменти для керування білінгом.

Google Cloud Platform. Хмарні сервіси від Google. Вони працюють на тій самій інфраструктурі, яку компанія використовує для своїх продуктів. GCP надає понад 200 послуг, серед яких обчислення, сховище, бази даних, мережа, сервіси для роботи з великими даними, машинне навчання тощо.

До переваг провайдера відносяться:

1. Ціна. Один із головних бенефітів GCP. Компанія пропонує місячний тарифний план, що залежить від використаних ресурсів. Також є знижки на обов'язкове використання (committed use discounts) для передбачуваних робочих навантажень, що скоротить витрати на потрібні замовнику ресурси. До прикладу, компанія купує певну кількість vCPU та пам'яті на один або три роки та може отримати знижку на ці ресурси.

2. Швидкість мережі, яку пропонує Google своїм клієнтам, складає до 10 Тбіт. Мережа компанії працює по всьому світу, тож має низьку затримку.

3. Робота з великими даними. Google має інструменти по типу BigQuery — сховища даних, що забезпечує швидку обробку великих даних, а також Cloud Dataflow для обробки у реальному часі. Також тут знайдеш інструменти машинного навчання та ШІ.

4. Масштабованість. Google надає можливість зменшувати або збільшувати ресурси залежно від потреб. Це стосується обчислювальних ресурсів, сховища та можливостей мережі.

5. Автоматизація. GCP має інструменти для автоматизації, які дозволяють автоматизувати процеси збирання, тестування та розгортання.

6. Безпека. GCP захищає програми, інфраструктуру та дані. Для цього є вбудовані функції безпеки, шифрування, ідентифікація та керування доступом, а ще сертифікати відповідності.

Навіть при такій кількості переваг присутні і деякі недоліки:

1. Вибір послуг. У Google їх багато, але на фоні AWS та Azure його пропозиція не така широка.

2. Вартість має також і негативну сторону. Google пропонує гарні ціни, проте передбачити витрати складно через різні моделі ціноутворення та потенційні додаткові витрати. Ще компанія може змінювати логіку ціноутворення та не повідомляти про це, що може призвести до неприємних сюрпризів.

3. Проблеми з конфіденційністю. Google зберігає дані користувачів на своїх серверах, має доступ до них та може використовувати ці дані у своїх цілях.

Alibaba Cloud. Публічна хмарна платформа від китайської компанії, відомої у сфері електронної комерції. Alibaba Cloud, так само як і її більші конкуренти, пропонує хмарні рішення, що включають сховище, обчислення, мережу, безпеку, аналітику, штучний інтелект і машинне навчання. Є також сервіси для девелоперів, що дозволяють розробляти застосунки, втілювати DevOps та керувати контейнерами. Загалом компанія надає понад 100 хмарних послуг.

Перевагами провайдера є:

- Більша кількість процесорів віртуальних машин (ВМ). В цьому аспекті Alibaba випереджає «велику трійку». Цей момент важливий для компаній з великим навантаженням на сервер: додаткові процесори допоможуть підвищити продуктивність програми та ефективніше використовувати ресурси.
- Вартість. Хоч ціна залежатиме від специфічних потреб конкретної компанії, проте сервіси Alibaba Cloud, наприклад, інстанси ВМ загалом коштують дешевше, ніж у провайдерів вище. Ще Alibaba пропонує велику кількість варіантів співпраці та оплати, що робить їх сервіси доступними й для малих бізнесів.
- Великий вибір хмарного сховища. Можна обрати сховище об’єктів, спільне сховище файлів, сховище для архівування, гібридне сховище, резервне копіювання, а також сховище для аварійного відновлення.
- Сервіси реляційних баз даних. Якщо компанії потрібна хмара для реляційної бази даних, то варто розглянути Alibaba Cloud. Сервіс має найбільшу кількість опцій в цьому контексті. До прикладу, ApsaraDB для MySQL, ApsaraDB RDS для SQL Server, ApsaraDB RDS для PostgreSQL тощо.

- Глобальна присутність. Компанія має центри обробки даних в багатьох країнах і регіонах, а це означає низьку затримку та високу доступність.

Присутній ряд недоліків:

- Географічна прив’язка, адже деякі сервіси доступні лише в Китаї. Також компанія сильно залежить від китайського ринку, на неї можуть вплинути урядові постанови та політична ситуація.

- Бізнеси, що співпрацюють з Alibaba Cloud іноді зазначають, що рахунки, які компанія їм виставляє, не завжди містять детальну інформацію про використані ресурси. А це викликає питання.

- Також Alibaba Cloud має менше сертифікатів відповідності, ніж інші великі хмарні провайдери.

- Сервіс має й менше інтеграцій. Не всі основні інструменти підтримують роботу з Alibaba Cloud. Більшість працюють лише з AWS, Azure та GCP, що може бути незручно.

- Знайомих із цим провайдером фахівців не так багато, тож знайти інженерів для роботи з ним може бути складно.

- Проблеми з підтримкою. Компанія надає різні ресурси, що допоможуть розібратися з роботою сервісу, наприклад, туторіали, відповіді на часті запитання та форуми користувачів. Також можна зв’язатися з підтримкою телефоном, електронною поштою або у чаті. Проте тут є нюанс: здебільшого ці ресурси пропонуються китайською мовою, що ускладнює процес.

IBM Cloud. IBM або «блакитний гігант» також пропонує хмарні сервіси, що включають PaaS, IaaS та SaaS. Компанія обіцяє рішення, які забезпечують вищий рівень відповідності, безпеки та керування, з перевіреними моделями архітектури та методами для швидкої доставки та виконання критично важливих робочих навантажень. Пропозиція IBM включає обчислення, контейнери, безпеку, штучний інтелект, сховище, аналітику, віртуалізацію та багато іншого.

Перевагами даного провайдера є:

- Можливості гібридної хмари. IBM зосереджується на гібридних хмарних рішеннях, що дозволяє бізнесам інтегрувати свою локальну інфраструктуру з хмарою. Такий підхід пропонує гнучкість та масштабованість.

- Безпека та відповідність. Хмарні сервіси IBM мають надійні заходи безпеки, шифрування, засоби контролю доступу та сертифікати відповідності.

- Штучний інтелект та аналітика. У провайдера багато послуг у цих напрямках, наприклад, Watson AI, IBM Cloud Pak for Data та IBM Cognos Analytics. Вони надають користувачам розширену аналітику, машинне навчання, обробку природної мови та можливості керування даними.

- Спеціальні рішення. У IBM є хмарні рішення для різних галузей. До прикладу, охорони здоров'я та фінансів. Вони допоможуть ефективно розв'язувати конкретні завдання потрібної галузі.

- Недоліки в даного провайдера теж присутні:

- Складність. Великий вибір послуг та фокус на гібридній хмарі може ускладнити користувачам роботу із сервісом. Особливо для новачків. Тож може знадобитися певний час, щоб розібратися та навчитися працювати з сервісом.

- Ціна. IBM Cloud не завжди може бути вигідною, особливо для невеликих компаній чи стартапів. Тож радимо звернути на це увагу, перш ніж почати роботу з провайдером.

- Менше дата-центрів. Це може бути суттєвим недоліком для компаній, яким потрібне територіально розподілене розгортання або необхідно задовольнити певні нормативні вимоги в конкретних регіонах.

- Взаємодія з користувачем. Інтерфейс IBM CLoud менш інтуїтивно зрозумілий та зручний у порівнянні з іншими провайдерами [6].

1.3 Поняття хмарного сховища (Cloud Storage)

Хмарне сховище — це режим зберігання комп'ютерних даних, у якому цифрові дані зберігаються на серверах за межами сайту. Сервери обслуговуються стороннім постачальником, який відповідає за розміщення, керування та захист даних, що зберігаються в його інфраструктурі. Провайдер гарантує, що дані на його серверах завжди доступні через загальнодоступні чи приватні підключення до Інтернету.

Хмарне сховище дозволяє організаціям зберігати, отримувати доступ і обслуговувати дані, щоб їм не потрібно було володіти та керувати власними центрами обробки даних, переводячи витрати з моделі капітальних витрат на операційну. Хмарне сховище є масштабованим, що дозволяє організаціям розширювати або зменшувати об'єм даних залежно від потреб.

Хмарне сховище використовує віддалені сервери для збереження даних, наприклад файлів, бізнес-даних, відео або зображень. Користувачі завантажують дані на сервери через підключення до Інтернету, де вони зберігаються на віртуальній машині на фізичному сервері. Щоб зберегти доступність і забезпечити резервування, хмарні постачальники часто поширюють дані на кілька віртуальних машин у центрах обробки даних, розташованих по всьому світу. Якщо потреби в сховищі збільшаться, хмарний постачальник запустить більше віртуальних машин, щоб впоратися з навантаженням. Користувачі можуть отримати доступ до даних у хмарному сховищі через підключення до Інтернету та програмне забезпечення, таке як веб-портал, браузер або мобільний додаток, через інтерфейс програмування додатків (API).

Хмарне сховище доступне в чотирьох різних моделях:

- публічне хмарне сховище (Public Cloud Storage) – це модель, у якій організація зберігає дані в центрах обробки даних постачальника послуг, які також використовуються іншими компаніями. Дані в загальнодоступному хмарному сховищі розповсюджені в кількох регіонах і часто пропонуються на основі передплати або оплати за використання. Публічне хмарне сховище вважається «еластичним», що означає, що дані, що зберігаються, можна збільшити або зменшити залежно від потреб організації. Постачальники загальнодоступних хмар зазвичай роблять дані доступними з будь-якого пристрою, наприклад смартфона або веб-порталу;

- приватне хмарне сховище (Private Cloud Storage) – це модель, у якій організація використовує власні сервери та центри обробки даних для зберігання даних у власній мережі. Крім того, організації можуть мати справу з постачальниками хмарних послуг, щоб надати виділені сервери та приватні з'єднання, які не використовуються жодною іншою організацією. Приватні

хмари зазвичай використовуються організаціями, які потребують більшого контролю над своїми даними та мають суворі вимоги до відповідності та безпеки;

- модель гібридної хмари (Hybrid cloud) – це суміш приватних і публічних моделей хмарного сховища. Модель гібридного хмарного сховища дозволяє організаціям вирішувати, які дані вони хочуть зберігати в якій хмарі. Конфіденційні дані та дані, які мають відповідати суворим вимогам відповідності, можуть зберігатися в приватній хмарі, тоді як менш конфіденційні дані зберігаються в публічній хмарі. Модель гібридного хмарного сховища зазвичай має рівень оркестровки для інтеграції між двома хмарами. Гібридна хмара забезпечує гнучкість і дозволяє організаціям все одно розширювати масштаб за допомогою загальнодоступної хмари, якщо виникне потреба;

- мультихмарна модель зберігання (Multicloud storage) – це коли організація встановлює більше ніж одну хмарну модель від кількох постачальників хмарних послуг (загальнодоступних чи приватних). Організації можуть вибрати багатохмарну модель, якщо один постачальник хмарних технологій пропонує певні пропрієтарні програми, організація вимагає, щоб дані зберігалися в певній країні, різні команди навчаються на різних хмарах або організації потрібно задовольняти різні вимоги, які не вказані в сервісах. Угоди про рівень обслуговування. Мультихмарна модель пропонує організаціям гнучкість і резервування.

Хмарне сховище пропонує кілька варіантів використання, які можуть принести користь окремим особам і організаціям. Незалежно від того, чи людина зберігає свій сімейний бюджет в електронній таблиці, чи велика організація зберігає багаторічні фінансові дані в надійно захищеній базі даних, хмарне сховище можна використовувати для збереження будь-яких цифрових даних стільки, скільки потрібно.

Резервне копіювання даних є одним із найпростіших і найпопулярніших способів використання Cloud Storage. Виробничі дані можна відокремити від даних резервного копіювання, створюючи проміжок між ними, який захищає організації у разі кіберзагроз, таких як програми-вимагачі. Резервне копіювання

даних через хмарне сховище може бути таким же простим, як збереження файлів у цифровій папці або використання блокового сховища для зберігання гігабайт або більше важливих бізнес-даних.

Можливість архівувати старі дані стала важливим аспектом хмарного сховища, оскільки організації переходять до оцифровки десятиліть старих записів, а також зберігають записи для цілей управління та відповідності.

Катастрофа — природна чи будь-яка — яка знищує центр обробки даних або старі фізичні записи, не обов'язково має бути подією, що руйнує бізнес, як це було в минулому. Хмарне сховище забезпечує аварійне відновлення, щоб організації могли продовжувати свій бізнес навіть у важкі часи.

Оскільки Cloud Storage робить цифрові дані доступними негайно, дані стають набагато кориснішими на постійній основі. Обробка даних, наприклад аналіз даних для бізнес-аналітики або застосування машинного навчання та штучного інтелекту до великих наборів даних, можлива завдяки Cloud Storage.

Завдяки можливості зберігати копії медіа-даних, таких як великі аудіо- та відеофайли, на серверах, розкиданих по всьому світу, медіа-компанії та компанії розваг можуть надавати своїй аудиторії завжди доступний вміст з низькою затримкою, де б вони не перебували.

Хмарне сховище буває трьох різних типів: об'єкт , файл і блок.

Об'єктне зберігання – це архітектура зберігання даних для великих сховищ неструктурованих даних. Він позначає кожну частину даних як об'єкт, зберігає її в окремому сховищі та об'єднує метаданими та унікальним ідентифікатором для легкого доступу та пошуку.

Файлове сховище організовує дані в ієрархічному форматі файлів і папок. Зберігання файлів є поширеним у персональних комп'ютерах, де дані зберігаються у вигляді файлів, а ці файли організовуються в папки. Зберігання файлів дозволяє легко знаходити та отримувати окремі елементи даних, коли вони потрібні. Зберігання файлів найчастіше використовується в каталогах і сховищах даних.

Блокове сховище розбиває дані на блоки, кожен з яких має унікальний ідентифікатор, а потім зберігає ці блоки як окремі частини на сервері. Хмарна

мережа зберігає ці блоки там, де це найбільш ефективно для системи. Блокове сховище найкраще використовувати для великих обсягів даних, які потребують низької затримки, наприклад для робочих навантажень, які вимагають високої продуктивності, або баз даних.

Розглянемо переваги та недоліки хмарних сервісів. До переваг можна віднести такі пункти:

- хмарне сховище дозволяє організаціям переходити від моделі капітальних витрат до моделі операційних витрат, дозволяючи їм швидко коригувати бюджети та ресурси;

- хмарне сховище є еластичним і масштабованим, тобто його можна збільшити (додати більше пам'яті) або зменшити (потрібно менше пам'яті) залежно від потреб організації;

- хмарне сховище пропонує організаціям гнучкість у тому, як зберігати та отримувати доступ до даних, розгортати та бюджетувати ресурси, а також створювати свою IT-інфраструктуру;

- більшість хмарних провайдерів пропонують надійну безпеку, включаючи фізичну безпеку в центрах обробки даних і передову безпеку на рівнях програмного забезпечення та програм. Найкращі хмарні постачальники пропонують архітектуру з нульовою довірою, керування ідентифікацією та доступом , а також шифрування;

- однією з найбільших витрат при експлуатації локальних центрів обробки даних є накладні витрати на споживання енергії. Найкращі хмарні постачальники працюють на основі сталої енергії за рахунок відновлюваних ресурсів;

- резервування (тиражування даних на кількох серверах у різних місцях) є невід'ємною рисою загальнодоступних хмар, що дозволяє організаціям відновлюватися після аварій, зберігаючи безперервність бізнесу;

До недоліків використання хмарних сервісів відносяться такі пункти:

- певні галузі, як-от фінанси та охорона здоров'я, мають суворі вимоги до того, як дані зберігаються та мають доступ до них. Деякі постачальники

публічних хмар пропонують інструменти для підтримки відповідності чинним правилам і нормам;

- трафік до хмари та з хмари може бути затриманий через перевантаження мережевого трафіку або повільне підключення до Інтернету;

- зберігання даних у загальнодоступних хмарах позбавляє певного контролю над доступом і керуванням цими даними, довіряючи, що постачальник хмарних послуг завжди зможе зробити ці дані доступними та підтримувати свої системи та безпеку;

- хоча постачальники публічних хмар прагнуть забезпечити постійну доступність, іноді трапляються збої, через що збережені дані стають недоступними [7, 8].

1.4 Аналіз популярних хмарних сервісів для збереження даних

Сьогодні існує чимало сервісів, які надають послуги хмарного сховища. Більшість безкоштовно надають достатньо обсягу простору для особистого користування. Зазвичай його вистачає пересічному користувачеві. При перевищенні обсягу можна змінити тариф на платний та отримати більше об’єму. Розглянемо найпопулярніші хмарні сховища.

GOOGLE DRIVE. Сьогодні такі хмарні сервіси для збереження даних як Google Drive – мають найзручніший функціонал з тих, які існують. Їх головні переваги:

- високий рівень захисту за допомогою шифрування згідно протоколу HTTPS та PFS (Perfect Forward Secrecy);

- робота з файлами різних форматів. Наприклад, у просторі можна зберігати фото, текстові документи, відео та аудіо файли. При придбанні платного пакета можна завантажувати важкі файли до 5 Тб один;

- для роботи можна використовувати різні інструменти – таблиці, презентації, малюнки, Google-форми;

- хмару дозволяється встановити на будь-яку операційну систему чи смартфон, синхронізувати дані та підключатися з будь-якого пристрою у зручний час;

- доступ до файлів може бути спільним для редагування або вивчення документа. Можна надіслати дозвіл за посиланням або вказавши електронну пошту користувача.

Безкоштовно надається 15 Гб простору. Вибираючи платний пакет, користувач може отримати до 30 Тб.

MICROSOFT ONEDRIVE. Одне з найперших віртуальних хмарних сховищ, яке з'явилось у 2007 році. Має такі особливості:

- зручне візуальне оформлення та проста ієрархія тек, які можна налаштувати на власний розсуд;

- вбудований Office 365, що дозволяє працювати з такими документами як PowerPoint, Excel, OneNote, Word;

- організовано можливість спільного доступу до документів та тек. Це дозволяє ділитися з опонентом їх вмістом;

- завантажити файли та виконувати з ними певну роботу можна навіть за відсутності інтернету. З появою останнього всі дані підтягуються до загального простору.

Користувач може використовувати 5 Гб пам'яті безкоштовно. При одноразовій сплаті тарифу можна отримати в особисте користування нескінченний простір.

DROPBOX. Сервіс має високу популярність завдяки наявності численних функцій та простому управлінню. Пропонує користувачу такі можливості:

- Безпека. Для збереження інформації використовується 256-бітове SSL та AES шифрування. Останній протокол визнається Агентством національної безпеки США одним із максимально захищених та можливий для використання з документами під грифом «цілком таємно». Кожне посилання для отримання доступу має пароль та обмеження часу дії;

- Відновлення. При випадковому видаленні документів з облікового запису протягом 30 днів вони підлягають відновленню;

- Встановлення. Сервіс може працювати на Windows, MacOS, Android, iOS. Це означає, що всі файли доступні на смартфоні або планшеті та на комп'ютері;

- USB-ключ. Кожен користувач задля підвищення безпеки при взаємодії зі своїм обліковим записом може використовувати спеціальний USB-ключ;

- Резервне копіювання в хмарі. Цей процес відбувається автоматично, тому не варто турбуватися під час ламання комп'ютера або смартфона. Всі дані будуть збережені у хмарі;

- Відсутність обмежень у відновленні журналу та даних;

- Сервіс чудово працює з 365 офісом Microsoft завдяки якісній інтеграції;

- Синхронізація між пристроями. Навіть у безкоштовній версії робота над одним документом відбувається без перешкод.

Якщо говорити щодо тарифів, то до 2 Гб об'єму можна отримати абсолютно безкоштовно. Щоб отримати більше обсягу, необхідно вибрати відповідний тариф і оплатити його. Для компаній хмара пропонує окремі тарифи, які потрібно попередньо узгоджувати з менеджером. Такі хмарні сервіси для збереження даних дуже гнучкі та лояльні до своїх клієнтів.

APPLE ICLOUD. Хмарне сховище спочатку було створено для власників гаджетів Apple з оболонками mac OS та iOS. Функціонал iCloud:

- використання на різних пристроях яблучної продукції та загальна синхронізація. Це дозволяє користувачеві зайти в обліковий запис з будь-якого гаджета;

- у хмарному сховищі постійно відбувається резервне копіювання даних, що унеможливлює їх втрату у разі ламання гаджета;

- у просторі є підтримка PDF-файлів.

- iCloud пропонує користувачам використовувати 5 Гб у рамках безкоштовного тарифу та до 2 Тб за підпискою.

MEGA. Є відносно новим, але перспективним хмарним сховищем. Користувачам пропонується:

- можливість встановлення на комп'ютері з будь-якою операційною системою або на смартфоні. При цьому працює синхронізація, тому після завантаження дані можна переглядати з будь-якого пристрою;

- високий рівень безпеки за допомогою алгоритму шифрування AES. Доступ до файлів заборонено для третіх осіб, якщо власник не погодився на цю дію;

- зручний та зрозумілий інтерфейс. Підтримка Drag&Drop дозволяє переміщувати всі файли за допомогою миші;

- перегляд та завантаження файлів доступні навіть користувачам, які не мають облікового запису в системі.

Хмарні сервіси для збереження даних мають також і недоліки – а саме для Mega це відсутність резервного копіювання. Також краще одразу придбати платний пакет, щоб гарантовано зберегти дані. Безкоштовно користувач може зайняти до 20 Гб, платно – до 16 Тб.

IDRIVE. Сервіс відрізняється високим рівнем конфіденційності, тому ідеально підходить для зберігання корпоративних документів. Також пропонує:

- можливість встановлення на комп'ютері та телефоні з синхронізацією. З одного облікового запису користувач може працювати на різних пристроях;

- копіювання видалених даних з пристрою у хмару. Це дозволяє заощаджувати місце, а також за необхідності відновлювати дані;

- захист облікового запису за допомогою 256-бітного AES шифрування. Отримати доступ можна за допомогою спеціального ключа. Останній не зберігається на серверах хмари – лише у користувача;

- можливість пересилання даних за допомогою соцмереж Twitter, Facebook, а також через електронну пошту.

Хмарні сервіси для збереження даних iDrive пропонують безкоштовне використання 10 Гб простору. Усього можна купити до 50 Тб для персонального застосування або 35 Тб корпоративного.

AMAZON DRIVE. Хмарні сервіси для збереження даних від Amazon пропонують безліч можливостей як особистого, так і корпоративного користування. Їх переваги:

- надійне зберігання архівів фотографій та керування ними. Функціонал розділу досить різноманітний: розумний пошук, розпізнавання обличчя, редагування фото;

- налаштування двофакторної автентифікації, що унеможливлює несанкціонований доступ третіх осіб;

- передплатники Amazon Prime можуть безкоштовно зберігати свої фотографії та відео у просторі на 5 Гб. Якщо потрібно збереження файлів, необхідна передплата Amazon Drive.

PCLOUD. Швейцарський сервіс, який пропонує користувачам максимальний рівень конфіденційності та захисту. Ключові аспекти хмарного функціонала:

- кожна операційна система має своє програмне забезпечення від сервісу pCloud;

- можливе завантаження будь-якого розміру файлів, найголовніше, щоб вони помістилися в доступний обсяг;

- відсутнє обмеження на завантаження та розвантаження файлів для платних та безкоштовних облікових записів. Таким чином, всі користувачі мають рівні права та можливості;

- декілька рівнів безпеки. Сервіс використовує 256-бітове шифрування, яке зламати досить важко. Існує складний захист: усі дані надходять через протокол TLS/SSL на сервери хмари, де відбувається копіювання на три та більше хмарних серверів;

- хмарні сервіси для збереження даних pCloud – пропонують зручні функції для користувачів, наприклад, спільний доступ до документів або запрошення для перегляду теки.

Зареєстрованим користувачам дають 10 Гб простору безкоштовно. За передплату можна отримати до 2 Тб.

MEDIAFIRE. За допомогою цього сервісу користувач може:

- зберегти аудіофайли, тексти, фото та відео;

- створити резервні копії файлів та поділитись ними з колегами або друзями.

- отримати 10 Гб безкоштовного простору. Єдиний мінус сервісу під час неоплаченого користування – реклама. Після покупки тарифу її можна позбутися та отримати до 2 Тб обсягу.

BOX. Хмарні сервіси для збереження даних Box – зручні та прості сервіси, що пропонують користувачам такі можливості:

- робота на будь-якому пристрої – користувач може встановити програму на комп'ютері або смартфоні;

- синхронізація – працює без помилок та підвисань, що дуже зручно при одночасному встановленні на кілька пристроїв;

- працює з 365 Office та Google документами.

Зареєстрованим абонентам надається 30 Гб безкоштовно. З використанням платного тарифу простір необмежено.

SPIDEROAK. Сервіс хмарного зберігання пропонує користувачам посилений захист, який не дає шахраям жодного шансу для крадіжки даних. Головні особливості SpiderOak:

- Безпека. Користувач отримує при авторизації ключ доступу, який не зберігається на жодному хмарному сервері компанії;

- Універсальність. Сервіс встановлюється на різні пристосування, ноутбуки, комп'ютери, лептопи;

- Шифрування. Для максимальної конфіденційності використовується протокол із нульовим знанням, а для шифрування – 256-бітна система AES.

Щоб почати використовувати такі хмарні сервіси для збереження даних як SpiderOak, необхідно сплатити мінімальну передплату і отримати 150 Гб простору. Максимальний обсяг – 4 Тб.

UCLOUD. Сервіс UCloud пропонує максимально високий рівень захисту даних користувача. Головні переваги:

- можливість протестувати сервіс у безкоштовному режимі протягом 30 днів. Надалі – підключення найвигіднішого для користувача тарифного плану;

- допомога технічної підтримки у будь-який час доби без перерв та вихідних;

- велика кількість послуг: підтримка Microsoft 365, можливість обладнання приватної хмари або гібридної хмари, використання хмарної технології Microsoft Azure;

- швидке розгортання додаткових сервісів, якщо це необхідно [9].

Висновок. У проведеному дослідженні щодо хмарних сервісів важливо підкреслити кілька ключових висновків та підсумків, які виникають із проведеного аналізу.

Перш за все, слід відзначити, що хмарні сервіси виявляються ефективним засобом оптимізації ресурсів для підприємств, дозволяючи їм зменшити витрати на IT-інфраструктуру та отримувати доступ до необхідних обчислень та ресурсів. Це стає суттєвим фактором у змаганні за підвищення конкурентоспроможності на ринку.

Однак із зростанням популярності хмарних сервісів виникають нові виклики, зокрема у сфері безпеки та конфіденційності даних. Сприйняття цих сервісів як надійного засобу збереження інформації вимагає постійного вдосконалення заходів безпеки для захисту від потенційних кіберзагроз.

Важливо також визнати, що хмарні сервіси є спонуканням до інновацій та розвитку галузі. Компанії, які успішно впроваджують ці технології, отримують можливість більш гнучко та оперативно реагувати на зміни в бізнес-середовищі, що сприяє їхньому стійкому розвитку.

Не менш важливою є увага до етичних аспектів використання хмарних сервісів. Забезпечення відповідального використання технологій та дотримання стандартів конфіденційності стає необхідністю в умовах зростаючої кількості оброблюваних даних.

Нарешті, дослідження вказує на необхідність подальших наукових досліджень у галузі хмарних сервісів. Розвиток більш продуктивних алгоритмів, посилення заходів безпеки та вивчення впливу хмарних технологій на соціально-економічний розвиток можуть стати ключовими напрямками майбутніх досліджень в цій важливій галузі інформаційних технологій. У цілому, хмарні сервіси сьогодні вже є необхідним інструментом для багатьох сучасних підприємств, проте їхнє використання потребує уважного підходу та постійного вдосконалення для максимізації переваг і зниження можливих ризиків.

2. СТРУКТУРА ЗАХИЩЕНОЇ СИСТЕМИ ЗБЕРІГАННЯ ДАНИХ

2.1 Надлишкова система залишкових класів

Системою числення, або нумерацією, називається набір правил і символів, які дозволяють представити (кодувати) будь-яке невід'ємне число. Використання систем числення пов'язане з певними вимогами, включаючи однозначне кодування невід'ємних чисел 0, 1, ... з певного обмеженого діапазону P. Це означає, що кожне число повинно бути представлене унікальним способом за допомогою обмеженої кількості символів. Також важливо, щоб системи числення дозволяли виконувати арифметичні та логічні операції з числами. Крім того, системи числення вирішують задачу нумерації, тобто ефективного перетворення чисел у відповідні номери, забезпечуючи, що ці номери мають мінімальну кількість цифр. Вибір правильної системи числення є важливим для ефективного вирішення різних завдань та її практичного використання.

Історично першими були непозиційні системи числення, які ґрунтувалися на кількісному підході до представлення чисел за допомогою спеціальних символів. Кожен з цих символів мав свій числовий еквівалент. Наприклад, у римській нумерації символ "X" відповідав числовий еквівалент 10.

Такі символи-числа використовувалися для отримання інших чисел. Наприклад, додавання вертикальної риски перед "X" давало символ "IX," що позначало віднімання одиниці від десяти, і результат становив 9. Такі символи, подібні "X," називалися вузловими та широко використовувалися в непозиційних системах числення. Важливо відзначити, що в цих системах не було символу, що відповідав би нулю, оскільки поняття нуля як числа ще не існувало.

Непозиційні системи числення були складними та обмеженими в можливостях, оскільки вимагали великої кількості символів для кодування чисел та ускладнювали виконання арифметичних та логічних операцій. Це призвело до розвитку позиційних систем числення, які мали прості правила кодування чисел і дозволяли легко виконувати операції з числами.

Позиційна система числення - це система числення, в якій значення кожної цифри в запису числа залежить від її позиції або розряду. У цій системі вага кожної позиції кратна певному натуральному числу, яке називається основою системи числення і зазвичай позначається як "b," де b > 1.

Винахід позиційної системи числення, що ґрунтується на значенні цифр залежно від їх позиції, приписують давнім цивілізаціям, таким як шумери і вавилонці. Індуси також розвинули цю систему числення і вона має величезне значення для історії людської цивілізації.

До числа таких систем відноситься сучасна Десяткова система числення (з основою b = 10), яку пов'язують з лічбою на пальцях. У середньовічній Європі ця система була вперше використана завдяки італійським купцям, які, з свого боку, запозичили її від мусульман.

В свою чергу непозиційна система числення - це системи, в яких значення кожної цифри не залежить від її позиції у числі. У таких системах цифри можуть представляти певні величини, але це представлення не залежить від того, де ця цифра розташована у числі. Зазвичай, непозиційні системи можуть мати обмеження на розташування цифр, такі як спадна або групована організація, але це не впливає на значення цифр.

Прикладом непозиційної системи числення є римська система числення, в якій латинські букви використовуються для позначення чисел. Наприклад, у римській системі "V" означає 5, "I" - 1, і навіть якщо вони розміщені в числі як "VII," значення лишається незмінним - 7. У цій системі цифри мають фіксовані значення і не залежать від їхньої позиції.

Прикладом непозиційної системи числення є числова система залишків. Де представлення чисел в системі, яка базується на китайській теоремі про залишки, включає в себе використання модульної арифметики для операцій з числами. Ця система часто використовується для представлення великих цілих чисел у вигляді набору менших цілих чисел. Вона дозволяє ефективно виконувати операції з великими цілими числами та оптимізувати їх обробку.

За допомогою китайської теореми про залишки, велике ціле число розбивається на декілька менших чисел, і операції проводяться над кожним з цих

менших чисел. Після цього застосовується теорема про залишки, щоб отримати результат для початкового великого числа.

Ця система дозволяє розподілити обчислення на менші операції та дозволяє швидше виконувати арифметичні операції з великими цілими числами. Вона знаходить застосування у криптографії, математичних обчисленнях та інших галузях, де потрібно оптимізувати роботу з великими цілими числами [10].

Через китайську теорему про залишки (ЧСЗ) визначається представлення цілого числа від 0 до (M-1), де M - добуток чисел базису (m_1, m_2, ..., m_n), які є взаємно простими. Ці числа базису використовуються для розбиття великого діапазону чисел на менші піддіапазони, щоб виконувати операції з числами більш ефективно.

Кожному цілому числу x з діапазону [0, M-1] ставиться у відповідність набір залишків (x_1, x_2, ..., x_n), де x_i є залишком від ділення числа x на m_i, і цей залишок еквівалентний x за модулем m_i.

Ця система гарантує однозначність представлення для чисел з діапазону [0, M-1], що означає, що кожне число в цьому діапазоні має єдиний представник в системі ЧСЗ залишків. Це дозволяє ефективно виконувати операції з числами та робити математичні обчислення в межах великого діапазону значень чисел.

Якщо базис системи складається з чисел, які не є взаємно простими (тобто не мають спільних дільників, крім одиниці), то їх можна використовувати для представлення чисел з діапазону [0, M-1], де M - найменше спільне кратне (НСК) чисел базису.

Наприклад, якщо базис складається з чисел $m_1 = 2$ і $m_2 = 4$, то числа 3 і 7 можуть бути представлені однаково, так як їх залишки від ділення на ці числа збігаються:

$$3_{10} = (x_1 = 1, x_2 = 3) = (1, 3)$$

$$7_{10} = (x_1 = 1, x_2 = 3) = (1, 3)$$

Це сталося через те, що найбільше число, яке можна представити в цьому базисі, дорівнює найменшому спільному кратному чисел 2 і 4, яке дорівнює 4. Тому числа 3 і 7, які дорівнюють 3 за модулем 4, мають однакове представлення.

У випадку, коли базис складається з чисел, які мають спільний дільник, важливо враховувати обмеження в межах діапазону [0, M-1], оскільки представлення може стати неоднозначним, і деякі числа можуть мати однаковий залишок при діленні на базисні числа.

У китайській теоремі про залишки, арифметичні операції, такі як додавання, віднімання, множення і ділення, можуть бути виконані поелементно для кожного з базових чисел (m_1, m_2, ..., m_n), якщо відомо, що результат є цілим числом і також належить до діапазону [0, M-1].

Це означає, що для виконання арифметичних операцій в системі ЧСЗ ви можете виконувати операції окремо для кожного базового числа m_i і потім складати результати, взявши залишок від ділення на m_i. Це дозволяє вам ефективно виконувати арифметичні операції в системі ЧСЗ [11].

Наприклад, для додавання чисел у системі ЧСЗ, ви додаєте відповідні залишки для кожного базового числа і взятий залишок від ділення на m_i. Аналогічно, для віднімання, множення і ділення ви виконуєте відповідні операції для кожного базового числа і знову берете залишок від ділення на m_i. Результат буде в межах [0, M-1], і він може бути однозначно представлений в системі ЧСЗ залишків.

Так, в системі китайської теореми про залишки (ЧСЗ) арифметичні операції додавання і віднімання можуть бути виконані поелементно для компонент чисел X і Y і потім взято залишок від ділення на відповідні числа базису m_1, m_2, ..., m_n.

Якщо задані числа X і Y, їх компоненти записані як (x_1, x_2, ..., x_n) і (y_1, y_2, ..., y_n) відповідно, то для обчислення Z = X ± Y можна виконати наступні дії:

Виконати окреме додавання або віднімання для кожної компоненти:

$$z_1 \equiv (x_1 + y_1) \, (mod \, m_1)$$
$$z_2 \equiv (x_2 + y_2) \, (mod \, m_2)$$
$$\dots$$
$$z_n \equiv (x_n + y_n) \, (mod \, m_n)$$

Отримати результат Z як набір компонент $z_1, z_2, ..., z_n$.

Результат Z теж буде представлений в системі ЧСЗ залишків, і він буде в межах [0, M-1], де M - добуток чисел базису $m_1, m_2, ..., m_n$. Таким чином, арифметичні операції додавання і віднімання можуть бути виконані в цій системі залишків, зберігаючи однозначність результату.

В ЧСЗ ділення можливе, але з певними обмеженнями. Для виконання ділення Z = X / Y в системі ЧСЗ, де X і Y є наборами компонент, спершу потрібно переконатися, що результат буде цілим числом. Для цього необхідно, щоб всі компоненти Y ($y_1, y_2, ..., y_n$) були ненульовими.

Потім кожна компонента числа Z обчислюється як:

$$z_i \equiv x_i * (y_i^{-1}) \ (mod\ m_i),$$

де y_i^{-1} - обернене за модулем число до y_i, тобто число, яке задовольняє умову $y_i * (y_i^{-1}) \equiv 1 \ (\text{mod}\ m_i)$. Таким чином, компонента Z обчислюється як добуток відповідних компонент X і обернених чисел y_i^{-1} за модулем m_i.

Цей процес дозволяє виконувати ділення в системі ЧСЗ, забезпечуючи, що результат також буде представлений у вигляді набору компонент з обмеженнями [0, M-1], де M - добуток чисел базису $m_1, m_2, ..., m_n$.

ЧСЗ дійсно має важливе застосування в мікроелектроніці і спеціалізованих пристроях, таких як арифметично-логічні блоки (ALU) і центральні операційні системи (ЦОС). Це через наступні важливі переваги:

Контроль за помилками - в системах, де важлива точність обчислень, можливість контролювати помилки важлива. ЧСЗ дозволяє введення додаткових надлишкових модулів для забезпечення надійності і виявлення помилок під час операцій.

Висока швидкість роботи - ЧСЗ дозволяє виконувати паралельні реалізації базових арифметичних операцій. Це означає, що велика кількість операцій може виконуватися одночасно, що забезпечує високу швидкість роботи системи. Це особливо важливо в мікроелектроніці, де швидкість операцій має вирішальне значення.

Ефективність пам'яті: ЧСЗ дозволяє ефективно зберігати числа в системах з обмеженою пам'яттю, оскільки числа можуть бути розбиті на компоненти та обчислені над ними незалежно.

У мікроелектроніці і обчислювальних системах, де важлива швидкість, надійність і ефективність обчислень, використання ЧСЗ дозволяє оптимізувати арифметичні операції та забезпечити високу продуктивність[12].

Для прикладу візьмемо десяткове число 69 і перетворимо його в числовій системі залишків. Для потрібно мати базиси, які повинні виступати взаємно простими числами, твердження що виникло з праць видатних математиків, таких як Леонард Ейлер, Карл Фрідріх Гаусс та Пафнутий Чебишов. Чеські математики М. Валах і А. Свобода також внесли важливий внесок у розвиток ЧСЗ, особливо в представлення чисел у вигляді сукупності додатних залишків вирахуваних по групі взаємно простих основ. Ці математики допомогли сформулювати та оптимізувати концепції ЧСЗ.

Функціональні теореми Гауса стосуються поділу натуральних чисел на взаємно прості множники, і вони грають важливу роль в теорії чисел. Основні ідеї, які вони включають, виглядають так:

Перша функціональна теорема Гауса (ФТГ): Якщо a, b і c є цілими числами, і a і b взаємно прості, то якщо a ділить добуток bc, то a ділить c.
Друга функціональна теорема Гауса (ФТГ): Якщо a, b і c є цілими числами, і a і b взаємно прості, то якщо a ділить c і b ділить c, то ab ділить c.

Ці теореми важливі для розуміння взаємної простоти та дільників в цілих числах. В першій теоремі Гауса розглядається випадок поділу одним числом, а в другій - поділу добутком двох чисел. Обидві теореми вказують на важливий факт: якщо числа a і b взаємно прості, то їхні дільники ведуть себе незалежно.

Таким чином, якщо числа a_1, a_2, ..., a_n взаємно прості між собою, то їхнє подання у вигляді добутку N є єдиним, оскільки за Функціональною теоремою Гауса перше і друге функціональні правила не дозволяють жодним іншим числам ділити всі a_1, a_2, ..., a_n одночасно.

Нагадаємо, що взаємно простими числами називають числа, що мають найменший загальний дільник (НЗД), що дорівнює 1. Наприклад, 15 і 22 є взаємно прості числа (оскільки НЗД рівний 1).

Виходячи з викладеного вище використаємо для перетворення наступні базиси {5, 7, 9} , які є взаємно простими числами. Знайдемо залишки:

$$a_1 = 69 - (69 mod 5) * 5 = 69 - 65 = 4;$$
$$a_1 = 69 - (69 mod 7) * 7 = 69 - 63 = 6;$$
$$a_1 = 69 - (69 mod 9) * 9 = 69 - 63 = 6;$$

Звідси, число 69 може бути представлене в ЧСЗ у вигляді сукупності {4, 6, 6}, або в двійковій системі як 100110110.

При таких основах як {5, 7, 9} максимальне число, яке можливо представити згідно формули $R = p_1 * p_2 * p_3 * ... * p_n$ буде $R = 5 * 7 * 9 = 315$.

Для зворотнього конвертування числа з числової системи залишкових класів здійснюється за відповідною формулою:

$$N = (a_1 * B_1 + a_2 * B_2 + a_3 * B_3) - r * R;$$

де r – ранг, котрий приймає значення 0, 1, 2, … так, щоб права частина даного виразу була менша значенню R;

B_i – ортогональний базис, що визначається при виборі базису ЧСЗ.

$$B_i = k_i * \frac{R}{p_i};$$

де k_i – ціле додатне число ($k_i = 1,2, ... p_{i-1}$), при цьому k_i вибирається таким, щоб залишок від ділення B_i на p_i дорівнював одиниці (тобто вага базису k_i вибирається, виходячи з рівності) :

$$B_i = \frac{m_i * R}{p_i} = k_i * p_i + 1;$$

При базисах {5, 7, 9}, перетворимо наше число відображене у ЧСЗ як {4, 6, 6} у десяткову систему за допомогою наступних виразів:

$$B_1 = \frac{k_1 * 315}{5} = k_1 * 63 = 2 * 63 = 126;$$

для $p_1 = 5$ умова виконується при $k_1 = 2$.

$$B_2 = \frac{k_2 * 315}{7} = k_2 * 45 = 5 * 45 = 225;$$

для $p_2 = 7$ умова виконується при $k_2 = 2$.

$$B_3 = \frac{k_3 * 315}{9} = k_3 * 35 = 8 * 35 = 280;$$

для $p_3 = 9$ умова виконується при $k_3 = 8$.

Наступним кроком згідно формули:

$$N = (a_1 * B_1 + a_2 * B_2 + a_3 * B_3) - r * R;$$

Підставимо відповідні значення і отримаємо вираз:

$$N = (4 * 126 + 6 * 225 + 6 * 280) - r * 315;$$
$$N = (504 + 1350 + 1680) - r * 315;$$
$$N = 3534 - r * 315;$$

де згідно умов $r = 11$, відповідно на число в десятковій системі [13]:

$$N = 3534 - 3465 = 69;$$

Числова система залишків полягає в тому, що цифри в кожному розряді вирахувань є незалежними один від одного і не залежать від позиції, а лише від базису. Це означає, що кожна цифра коду представляє інформацію про число в

цілому, і її значення не змінюється в залежності від того, в якому розряді вона знаходиться. Ця властивість робить ЧСЗ корисною для відновлення спотвореної інформації при передачі по лінії зв'язку, оскільки дозволяє виявити та виправити помилки у прийнятому повідомленні.

Числова система залишків широко використовується в мікроелектроніці в спеціалізованих пристроях, таких як Арифметико-логічних пристроях (АЛП) і у Цифрових Обробках Сигналів (ЦОС). Це пов'язано з наступними вимогами:

- Контроль за помилками: ЧСЗ дозволяє вводити додаткові надлишкові модулі для забезпечення надійності і виявлення помилок при обробці даних.
- Висока швидкість роботи: Паралельна реалізація базових арифметичних операцій дозволяє досягти високої швидкості обчислень.
- Інформаційна безпека: ЧСЗ використовується для синтезу пристроїв високої надійності та захисту від завад.

Незважаючи на переваги, існують деякі недоліки ЧСЗ:

- Обмежена кількість чисел, яку можна представити в цій системі.
- Повільні алгоритми перетворення з позиційної системи числення в ЧСЗ і навпаки, особливо при роботі з великими числами.
- Відсутність ефективних алгоритмів для порівняння чисел, представлених у ЧСЗ.
- Порівняння зазвичай вимагає перетворення аргументів у змішану систему числення.
- Складність алгоритмів ділення, особливо у випадках, коли результат не є цілим числом.
- Труднощі у виявленні переповнення.

Незважаючи на ці недоліки, ЧСЗ ефективно використовується для вирішення конкретних завдань у мікроелектроніці, і має позитивний вплив на покращення продуктивності та надійності різних пристроїв та систем.

2.2 Захищена система зберігання даних та її характеристики

Захищена система зберігання даних - це система, що включає в себе програмне забезпечення та спеціалізоване обладнання, яке в комплексі забезпечує захист даних від несанкціонованого доступу, зміни чи знищення. Вона має такі характеристики:

1. Конфіденційність. Конфіденційність даних означає, що лише уповноважені користувачі можуть отримати доступ до даних. Для забезпечення конфіденційності даних можуть використовуватися такі технології:

- Шифрування даних перетворює їх у нерозбірливий формат, який може бути прочитаний лише за допомогою секретного ключа. Шифрування даних може бути застосоване до всього набору даних або окремих файлів або папок.
- Аутентифікація використовується для перевірки особи користувача, який намагається отримати доступ до даних. Аутентифікація може бути реалізована за допомогою таких методів, як паролі, біометрія або двофакторна аутентифікація.
- Контроль доступу використовується для обмеження доступу до даних авторизованих користувачів. Контроль доступу може бути реалізований за допомогою таких методів, як рольова модель або політика доступу на основі атрибутів.

2. Цілісність. Цілісність даних означає, що дані не були змінені чи знищені без дозволу. Для забезпечення цілісності даних можуть використовуватися такі технології:

- Резервне копіювання використовується для створення копії даних, яка може бути використана для відновлення даних у разі втрати або пошкодження. Резервне копіювання може бути виконане локально або на віддаленому сервері.
- Моніторинг використовується для виявлення змін у даних. Моніторинг може бути реалізований за допомогою таких методів, як журналювання змін або використання систем виявлення вторгнень.
- Відновлення використовується для відновлення даних у разі втрати чи пошкодження. Відновлення може бути виконане за допомогою резервної копії або за допомогою технології відновлення даних.

3. Доступність. Доступність даних означає, що дані можуть бути доступні користувачам у разі потреби. Для забезпечення доступності даних можуть використовуватися такі технології:

- Розподілені системи зберігання даних розподіляють дані між кількома серверами. Це може допомогти забезпечити доступність даних, навіть якщо один чи кілька серверів недоступні.
- Реплікація даних створює копії даних на одному чи кількох серверах. Це може допомогти забезпечити доступність даних, навіть якщо один чи кілька серверів недоступні.
- Модульні системи зберігання даних дозволяють легко додавати або видаляти сервери. Це може допомогти забезпечити доступність даних, навіть якщо зростає обсяг даних.

Вибір захищеної системи зберігання даних залежить від конкретних потреб організації. При виборі системи слід враховувати такі фактори:

- Тип даних, які потрібно зберігати: Деякі типи даних, такі як фінансові дані, вимагають більш високого рівня безпеки, ніж інші типи даних.
- Кількість даних, які потрібно зберігати: Великі обсяги даних можуть вимагати використання спеціальної системи зберігання даних.
- Сервіси, які необхідні: Деякі захищені системи зберігання даних включають додаткові послуги, такі як резервне копіювання та відновлення.

До прикладів захищених систем зберігання даних належать:

- Системи зберігання даних на основі хмарних технологій часто включають вбудовані функції безпеки, такі як шифрування та контроль доступу.
- Системи зберігання даних на базі штучного інтелекту можуть використовуватися для виявлення та запобігання вторгненням.
- Системи зберігання даних на базі блокчейну забезпечують підвищений рівень безпеки за рахунок використання децентралізованої мережі.

Захищені системи зберігання даних є важливим інструментом для захисту даних від несанкціонованого доступу, зміни чи знищення. Вони можуть допомогти організаціям захистити свою інтелектуальну власність, конфіденційну інформацію та фінансові дані.

2.3 Поняття шифрування даних

Шифрування даних – це метод безпеки, який перетворює дані в код або зашифрований текст, який можуть прочитати лише люди, які мають доступ до секретного ключа або пароля. Незашифровані дані називаються відкритим текстом. Наука про шифрування та дешифрування інформації відома як криптографія.

Шифрування даних захищає дані від викрадення, зміни чи зламу. Однак, щоб дані залишалися захищеними, ключ дешифрування має зберігатися в секреті та бути захищеним від несанкціонованого доступу.

Усі дані можна зашифрувати, включаючи дані в стані спокою (зберігаються у фіксованому місці, наприклад на жорсткому диску або на хмарних серверах) або дані в дорозі (наприклад, передаються через мережу).

Сьогодні широко використовуються два типи шифрування:

- симетричне шифрування використовує той самий ключ для шифрування та дешифрування
- асиметричне шифрування має закритий ключ, який зберігається власником даних, і відкритий ключ, виданий одержувачу даних.

Асиметричне шифрування вважається більш безпечним, оскільки воно не вимагає спільного використання закритого ключа.

Дані стали більш доступними та бажаними для зловмисників, ніж будь-коли, що збільшує потребу в захисті. Крім того, багато компаній стикаються з вимогами щодо захисту даних, багато з яких прямо вимагають використання шифрування.

Шифрування даних захищає дані від викрадення, зміни чи зламу. Однією з ключових переваг шифрування даних є те, що воно допомагає гарантувати автентичність даних. Шифруючи дані, ви можете бути впевнені, що інформація, до якої ви отримуєте доступ, не була підроблена або змінена неавторизованими особами. Шифрування даних також допомагає запобігти пошкодженню даних, яке може статися, коли дані зберігаються або передаються через різні системи.

Шифруючи дані, ви додаєте додатковий рівень захисту, який запобігає ненавмисному або зловмисному пошкодженню даних.

Багато галузей підпорядковуються суворим нормам щодо захисту конфіденційних даних. Наприклад, галузь охорони здоров'я повинна дотримуватися Закону про перенесення та підзвітність медичного страхування (HIPAA), тоді як фінансові установи мають відповідати Стандарту безпеки даних індустрії платіжних карток (PCI DSS). Впроваджуючи шифрування даних, компанії можуть переконатися, що вони відповідають цим нормативним вимогам і уникнуть потенційних штрафів або санкцій за їх невиконання.

Коли дані зберігаються у фіксованому місці, наприклад на пристрої, сервері чи базі даних, їх називають «даними в стані спокою». Неавторизовані особи можуть фізично або віддалено отримати доступ і отримати збережені дані. Завдяки шифруванню даних у стані спокою, навіть якщо зловмисники отримають носій даних, вони не зможуть інтерпретувати дані без правильного ключа дешифрування. Захищене шифрування допомагає гарантувати, що особисті дані, конфіденційна корпоративна інформація та інші конфіденційні записи залишатимуться недоступними та марними для тих, хто не авторизований.

Коли дані передаються між системами або пристроями, наприклад, через мережу, вони особливо вразливі для несанкціонованого доступу та втручання. Шифрування даних допомагає захистити дані під час передавання, забезпечуючи доступ до інформації лише авторизованим сторонам із правильними ключами розшифровки. Оскільки все більше працівників використовують мобільні пристрої для доступу до даних компанії, зростає ризик витоку даних. Шифрування даних може допомогти захистити конфіденційну інформацію, що зберігається на цих пристроях, а також дані, що передаються між мобільними пристроями та мережами компанії.

Хоча хмарне сховище пропонує численні переваги, такі як покращена доступність і зниження витрат на інфраструктуру, воно також створює унікальні проблеми безпеки. Однією з основних проблем для компаній, які використовують хмарне сховище, є безпека їхніх даних у стані спокою або даних,

що зберігаються на хмарних серверах. Шифрування даних забезпечує додатковий рівень захисту цих даних, гарантуючи, що навіть якщо неавторизовані сторони отримують доступ до хмарних серверів, вони не зможуть отримати доступ до зашифрованих даних без відповідних ключів дешифрування.

Віддалена робота стає все більш поширеною. Оскільки все більше співробітників працюють з дому чи інших віддалених місць, ризик витоку даних та інших інцидентів безпеки зріс. Шифрування даних може допомогти захистити конфіденційну інформацію, до якої мають доступ віддалені співробітники, гарантуючи, що навіть якщо їхні пристрої або з'єднання зламано, зашифровані дані залишаються в безпеці.

Інтелектуальна власність, як-от комерційні секрети, запатентовані алгоритми та дизайн продуктів, часто є джерелом життя бізнесу. Захист цієї цінної інформації має важливе значення для підтримки конкурентної переваги та запобігання корпоративному шпигунству. Шифрування даних може допомогти захистити інтелектуальну власність, гарантуючи, що навіть якщо неавторизована сторона отримає доступ до даних, вона не зможе розшифрувати зашифровану інформацію.

Симетричне шифрування використовує один закритий ключ для шифрування та дешифрування. Це швидший метод, ніж асиметричне шифрування, і його найкраще використовувати окремим особам або в закритих системах, оскільки він вважається менш безпечним. Використання симетричних методів із кількома користувачами у відкритих системах, наприклад у мережі, потребує передачі ключа та створює можливість для крадіжки. Найпоширенішим типом симетричного шифрування є AES.

Приклади симетричних алгоритмів шифрування даних:

- потрійний DES (3DES або TDES) — тричі запускає алгоритм DES, застарілий стандарт, шифруючи, дешифруючи та знову шифруючи для створення довшого ключа. Його можна запускати за допомогою одного ключа, двох або трьох різних ключів із підвищенням безпеки. 3DES використовує метод блокового шифрування, що робить його вразливим до таких атак, як зіткнення блоків;

- Twofish – один із найшвидших алгоритмів, доступний у розмірах 128, 196 і 256 біт зі складною структурою ключа для підвищення безпеки. Він безкоштовний для використання та з'являється в деяких із найкращих безкоштовних програм: VeraCrypt, PeaZip і KeePass, а також у стандарті OpenPGP;

- розширений стандарт шифрування (AES) – встановлений урядовим стандартом США для шифрування. AES – це алгоритм із симетричним ключем, який використовує методи блочного шифрування. Він доступний у розмірах 128, 192 та 256 біт із використанням зростаючої кількості раундів шифрування відповідно до розміру. Він створений для простого впровадження як в апаратному, так і в програмному забезпеченні;

- Blowfish – симетричний шифр, який має змінну довжину ключа від 32 до 448 біт. Продуктивність цього алгоритму залежить від обраної довжини ключа. Blowfish — це блоковий шифр, тому під час шифрування він розділяє дані на фіксовані блоки по 64 біти кожен;

- шифрування зі збереженням формату (FPE) – цей алгоритм шифрування також виконує анонімізацію вмісту. Він шифрує дані, зберігаючи існуючий формат. Наприклад, якщо ідентифікатор клієнта містить дві літери та десять цифр, отримана зашифрована форма матиме ту саму кількість і тип символів, але замінить їх на інші символи для захисту вихідних даних.

Асиметричне шифрування використовує два різні ключі для шифрування та дешифрування. Він має відкритий і закритий ключі, які математично пов'язані і можуть використовуватися лише разом. Для шифрування даних можна використовувати будь-який ключ, але для їх дешифрування потрібно використовувати парний ключ. Асиметричне шифрування використовується кількома користувачами та у відкритих мережах, як-от Інтернет, оскільки відкритий ключ можна вільно надавати без ризику крадіжки даних. Найпоширенішими типами асиметричного шифрування є RSA, DSA та ECC.

Приклади асиметричних алгоритмів шифрування даних:

- обмін ключами Діффі-Хельмана – метод, який дозволяє двом сторонам, кожна з яких має відкритий і закритий ключі, встановити спільний секретний

ключ через незахищений канал. Його безпека залежить від складності проблеми дискретного логарифмування. Це був один із перших алгоритмів криптографії з відкритим ключем;

- алгоритм цифрового підпису (DSA) – метод асиметричного шифрування, який використовується переважно для перевірки цифрових підписів, а не для шифрування даних. За допомогою DSA власник закритого ключа може створити підпис для повідомлення. Потім цей підпис може бути перевірений будь-ким, хто має доступ до відкритого ключа, гарантуючи автентичність повідомлення та те, що воно не було підроблено;

- RSA – один із перших алгоритмів із відкритим ключем, він використовує одностороннє асиметричне шифрування. RSA популярний завдяки великій довжині ключа і широко використовується в Інтернеті. Він є частиною багатьох протоколів безпеки, як-от SSH, OpenPGP, S/MIME та SSL/TLS, і використовується браузерами для створення безпечних з'єднань через незахищені мережі;

- криптографія з еліптичною кривою (ECC) – розроблена як удосконалення RSA, забезпечує кращий захист із значно меншою довжиною ключа. ECC – це асиметричний метод, який використовується в протоколі SSL/TLS.

Дані є цінними незалежно від того, чи передаються вони між користувачами чи зберігаються на сервері, і повинні бути захищені в будь-який час. Те, як цей захист здійснюється, залежить від стану даних.

Дані вважаються транзитними, коли вони переміщуються між пристроями, наприклад у приватних мережах, через Інтернет або з ноутбука на флешку. Дані піддаються більшому ризику під час передачі через необхідність дешифрування перед передачею та вразливість самого методу передачі. Шифрування даних під час передачі, яке називається наскрізним шифруванням, гарантує захист конфіденційності навіть у разі перехоплення даних.

Дані вважаються неактивними, якщо вони зберігаються на пристрої зберігання й не використовуються або не передаються активно. Дані в стані спокою часто менш вразливі, ніж під час передачі, через функції безпеки

пристрою, які обмежують доступ, але це не захищено. Крім того, він часто містить більш цінну інформацію, тому є більш привабливою мішенню для злодіїв.

Шифрування даних у стані спокою зменшує можливості для крадіжки даних, спричиненої втраченими чи викраденими пристроями, ненавмисним обміном паролем або випадковим наданням дозволу, збільшуючи час, необхідний для доступу до інформації, і надаючи час, необхідний для виявлення втрати даних, атак програм-вимагачів, віддалено стертих даних або змінив облікові дані.

Зашифровані дані можна зламати. Зловмисники можуть зламати системи шифрування даних кількома способами:

- випадкове відкриття – ключ дешифрування є секретним і має бути захищений від несанкціонованого доступу. Якщо користувачі випадково розкриють ключ або не захистять його належним чином, зловмисники можуть отримати доступ до захищених даних;
- зловмисне програмне забезпечення на кінцевих пристроях – багато кінцевих пристроїв мають такі механізми шифрування, як повне шифрування диска. Зловмисники можуть скомпрометувати кінцевий пристрій за допомогою зловмисного програмного забезпечення та використати ключі на пристрої для дешифрування даних;
- атаки грубою силою – зловмисники зазвичай намагаються зламати шифрування, випадково пробуючи різні ключі. Шанси на успіх безпосередньо залежать від розміру ключа. Ось чому більшість стандартів шифрування передбачають використання 256-бітних ключів шифрування. Однак деякі системи шифрування використовують слабкі шифри, які вразливі до атак грубої сили;
- криптоаналіз – це техніка, за якої зловмисники знаходять слабке місце в самому шифрі та використовують його для отримання доступу до даних;
- атаки по бічному каналу – це передбачає пошук помилок або слабких місць у структурі системи, що дозволяє користувачам розшифровувати дані або запобігати їх шифруванню, не порушуючи сам шифр;

- атаки соціальної інженерії . Ймовірно, найпростішим способом зламати зашифровані дані є використання фішингу або інших методів соціальної інженерії, щоб обманом змусити привілейованого користувача надати ключ;
- внутрішні загрози – серйозною загрозою для зашифрованих даних є ймовірність того, що привілейована особа обернеться проти організації та зловживатиме своїми привілеями, щоб викрасти дані. Інсайдерські загрози також включають недбалих користувачів, які не дотримуються політики безпеки.

Незважаючи на всі ці ризики, шифрування є надійним і ефективним засобом безпеки. Але з огляду на ймовірність того, що шифрування буде скомпрометовано, його слід розглядати як ще один рівень захисту, а не єдиний захист, який організації використовують для захисту своїх даних.

Коли організація зберігає дані в хмарі, вона може використовувати здатність постачальника хмари шифрувати дані. Більшість постачальників хмарних послуг пропонують шифрування як послугу, вбудовану в хмарні служби або як окрему пропозицію.

Перш ніж використовувати хмарне шифрування, важливо визначити, що саме пропонує хмарний постачальник:

- яка міцність шифрування та чи відповідає воно вимогам організації;
- хто керує ключами – існує кілька моделей, включаючи повністю керовані ключі шифрування та ключі шифрування, керовані клієнтом;
- як налаштувати наскрізне шифрування, щоб дані залишалися зашифрованими під час їх переміщення з хмари до кінцевих користувачів і назад.

Шифрування в хмарі є центральним компонентом будь-якої стратегії безпеки в хмарі. Однак організації повинні знати про такі важливі проблеми:

Хмарне шифрування може вважатися складним для кінцевих користувачів, особливо коли існує повне наскрізне шифрування.

Може бути важко інтегрувати хмарне шифрування з системами, що працюють локально або на кінцевих пристроях.

Існує потреба відстежувати використання хмарного шифрування, оскільки це інтенсивний процес. Залежно від цінової моделі це також може призвести до високих витрат на хмару.

Керування ключами має здійснюватися обережно, тому що якщо ключі шифрування втрачено, дані стануть марними, а якщо ключі не захищені належним чином, шифрування не дає ніяких переваг у безпеці.

Шифрування в хмарі є центральним компонентом будь-якої стратегії безпеки в хмарі. Однак організації повинні знати про такі важливі проблеми:

- хмарне шифрування може вважатися складним для кінцевих користувачів, особливо коли існує повне наскрізне шифрування;
- може бути важко інтегрувати хмарне шифрування з системами, що працюють локально або на кінцевих пристроях;
- існує потреба відстежувати використання хмарного шифрування, оскільки це інтенсивний процес. Залежно від цінової моделі це також може призвести до високих витрат на хмару;
- керування ключами має здійснюватися обережно, тому що якщо ключі шифрування втрачено, дані стануть марними, а якщо ключі не захищені належним чином, шифрування не дає ніяких переваг у безпеці.

Ось кілька тенденцій, які, ймовірно, сприятимуть розвитку шифрування даних у майбутньому:

- BYOE – це модель безпеки хмарних обчислень, яка дозволяє клієнтам хмарних служб керувати своїми власними ключами шифрування за допомогою власного програмного забезпечення для шифрування. Він також відомий як Bring Your Own Key (BYOK). BYOE працює, дозволяючи клієнтам розгортати віртуалізовані екземпляри власного програмного забезпечення для шифрування поряд із хмарними бізнес-додатками.
- EaaS – це модель підписки, у якій хмарні провайдери пропонують шифрування на основі оплати за використання. Цей підхід усуває проблеми відповідності та надає клієнтам певні можливості для керування власним шифруванням, щоб захищати дані в середовищах із кількома клієнтами. Ці служби зазвичай пропонують повне шифрування диска (FDE), шифрування бази даних або шифрування файлів.

Служба, у якій постачальники хмарних сховищ використовують алгоритми шифрування для захисту всіх даних, збережених у хмарних сховищах. Це схоже

на шифрування, яке виконується локально, але з важливими відмінностями. Клієнтам хмарних технологій слід приділити час, щоб зрозуміти політику та процедури постачальника щодо шифрування та керування ключами, щоб відповідати рівню конфіденційності їхніх зашифрованих даних, які самостійно керують.

З контексту проаналізованої інформації видно, що поєднання підходу до шифрування інформації на основі надлишкової системи залишкових класів та використання хмарних сервісів є потужним засобом для побудови високонадійної та захищеної системи зберігання даних. Отже, це поєднання двох технологічних підходів сприяє створенню комплексної системи, що володіє високим рівнем безпеки, доступності та гнучкості. Така система є перспективною для використання в різноманітних областях, де важливо зберігати та обробляти дані з максимальною ступенем захисту [14].

3. РОЗРОБКА ТА ТЕСТУВАННЯ СИСТЕМИ НАДІЙНОГО ЗБЕРІГАННЯ ДАНИХ

3.1 Компоненти та алгоритм роботи системи

Для створення програмного комплексу системи для надійного зберігання даних користувача на основі надлишкової системи залишкових класів в хмарних сервісах було розроблено мобільний додаток на базі операційної системи Android написаний на мові програмування Kotlin. Хмарним сервісом було обрано послуги надані платформою для мобільної розробки від компанії Google під назвою Firebase. Даний інструмент створений для того, щоб легко підключати продукти Google Cloud у міру зростання ваших потреб чи інфраструктури. Firebase і Google Cloud мають спільну інфраструктуру, про переваги якої було зазначено вище. Структура системи зображена на рисунку 3.1.

Рисунок 3.1 – Структура системи надійного зберігання даних

Клієнтською стороною виступає мобільний додаток, в якому відбувається процес шифрування і дешифрування, файлу вибраного користувачем, з використанням системи надлишкової системи залишкових класів. Серверна частина складається з Cloud Storage для зберігання файлів користувача та Firebase Firestore для зберігання додаткової інформації в базу даних. Процес аутентифікації відбувається за допомогою Google Account вибраного користувачем.

Firebase Cloud Firestore – це гнучка масштабована NoSQL база даних для мобільних, веб-розробок і серверних розробок побудована на інфраструктурі Google Cloud, для зберігання та синхронізації даних для розробки на стороні клієнта та сервера.

Дотримуючись моделі даних NoSQL Cloud Firestore, дані зберігаються в документах, які містять поля, зіставлені зі значеннями. Ці документи зберігаються в колекціях, які є контейнерами для документів, які можна використовувати для впорядкування даних і створення запитів. Документи підтримують багато різних типів даних, від простих рядків і чисел до складних вкладених об'єктів. Також можна створювати підколекції в документах і створювати ієрархічні структури даних, які масштабуються в міру зростання бази даних. Модель даних Cloud Firestore підтримує будь-яку структуру даних, яка найкраще підходить для програми [15, 16].

На рисунку 3.2 відображена структура бази даних використана в системі: колекції, документи, поля та залежності.

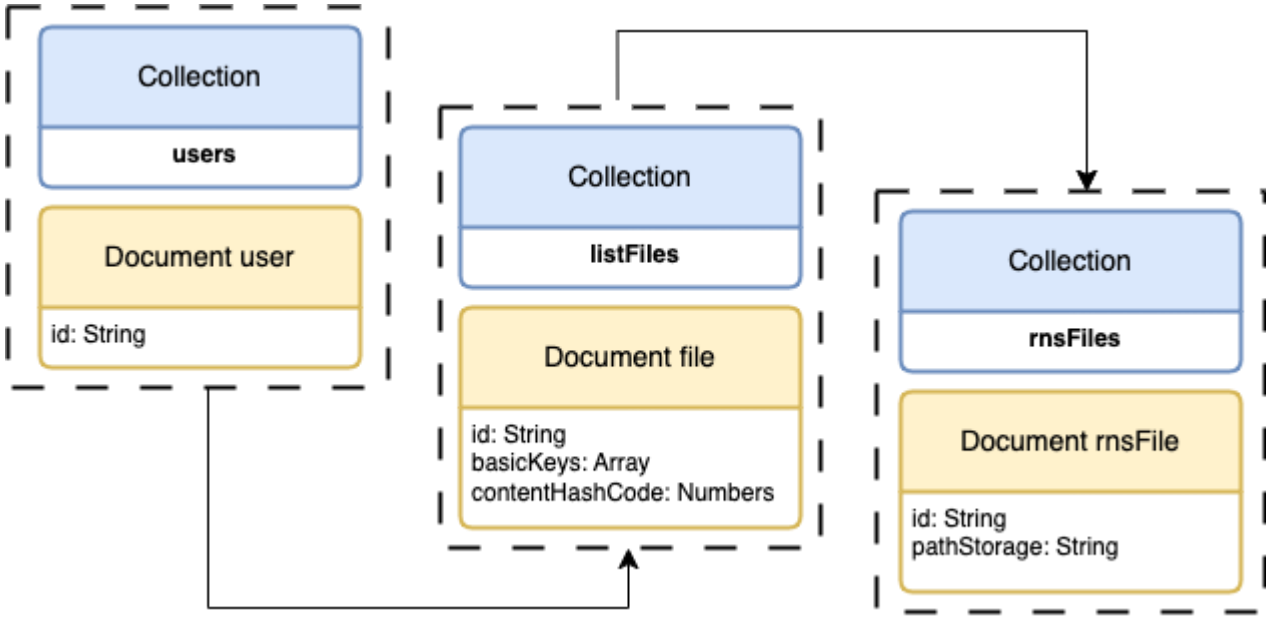


Рисунок 3.2 – Структура бази даних системи

Колекція "users" містить записи (документ) "user":

- "id" - ідентифікатор користувача;

Документу “user” підпорядкована колекція “listFiles” відображає список файлів, які користувач зашифрував і завантажив на сервер. Документ “file” містить такі поля і колекцію:

- “id” - ідентифікатор файлу користувача;
- “basicKeys” - масив базових ключів;
- “contentHashCode” - хеш-код контенту оригінального файлу;

Документу “file” підпорядкована колекція “rnsFiles” відображає інформацію про файли в яких записані залишки, присутні такі поля:

- “id” - ідентифікатор файлу-залишків;
- “pathStorege” - шлях до файлу залишків в CloudStorage.

Структура записів в базу даних може бути відображена в форматі JSON, зображене на рисунку 3.3. [17]

```
{
  "users": [
    {
      "id": "email",
      "listFiles": [
        {
          "id": "original_name.docx",
          "basicKeys": [
            "basicKey1",
            "basicKey2",
            "basicKey3"
          ],
          "contentHashCode": "content_hash_code_numbers",
          "rnsFiles": [
            {
              "id": "rnsPart_1",
              "pathStorage": "cloud_storage_path_1"
            },
            {
              "id": "rnsPart_2",
              "pathStorage": "cloud_storage_path_2"
            },
            {
              "id": "rnsPart_3",
              "pathStorage": "cloud_storage_path_3"
            }
          ]
        }
      ]
    }
  ]
}
```

Рисунок 3.3 – Структура запису даних в базі даних в форматі JSON

Процес отримання трьох файлів, які містять залишки від застосування надлишкової системи залишкових класів (НСЗК), зображено на рисунку 3.4.

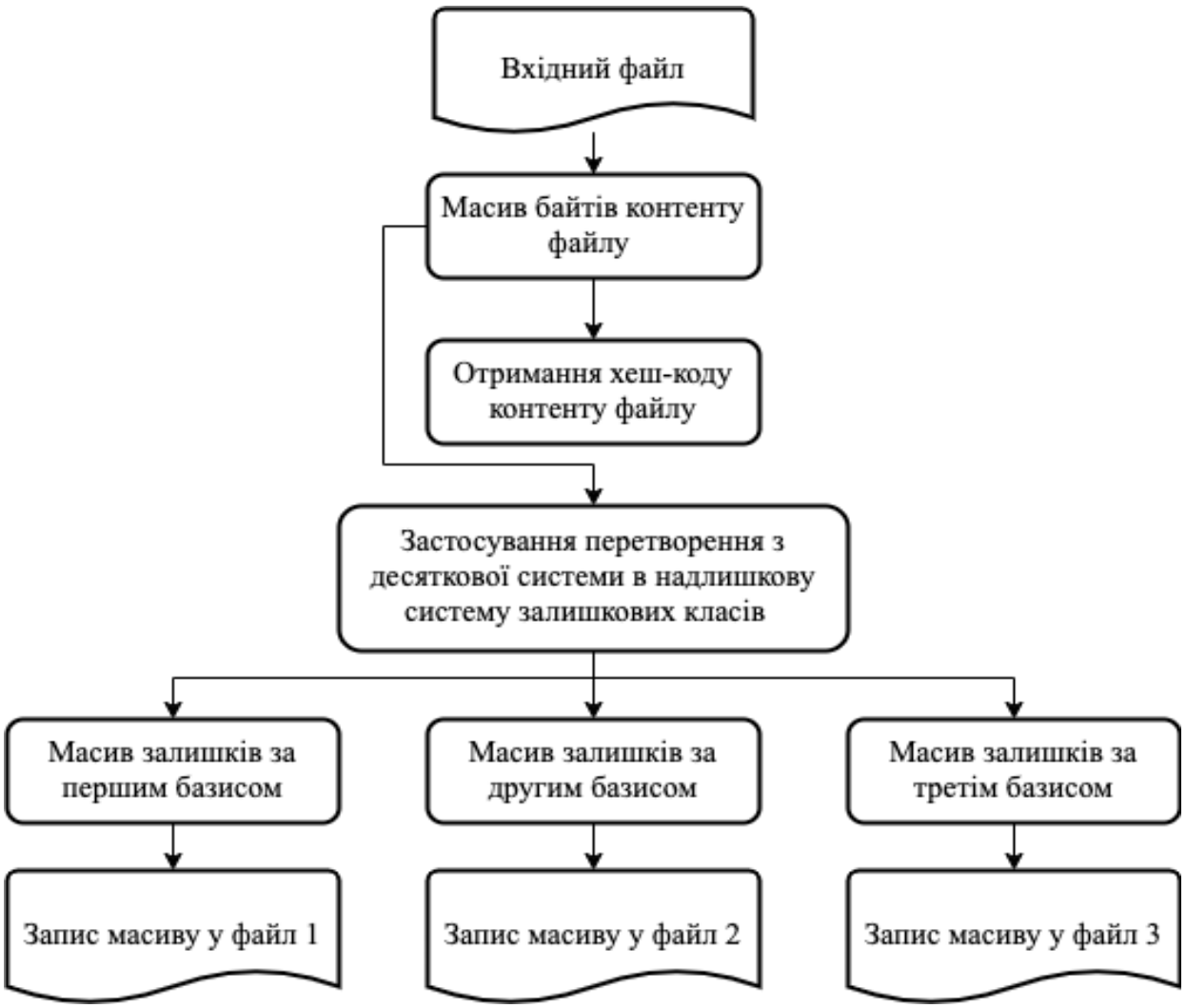


Рисунок 3.4 – Схема отримання трьох файлів з залишками від НСЗК

Оригінальний файл користувача початковим етапом отримаємо масив байтів в ASCII. Паралельно з цим отримуємо хеш-код вмісту файлу, який буде записаний у базу даних і використаний в подальшому, для перевірки правильності відновленого файлу.

Для перетворення в систему залишкових класів необхідно визначити певні числа (основи), які будуть використовуватися при обчисленні залишків. Алгоритм склеює шість чисел, і максимальне значення одного з цих чисел може бути 255, як вказано в таблиці ASCII. Тому умовою є те, що добуток трьох основ

повинен перевищувати число 255255255255255255, а також дані числа повинні бути взаємно простими.

Після отриманого отриманого масиву байтів ASCII та основ, розпочинається робота алгоритму перетворення з десяткової системи числення в систему залишкових класів. В циклі алгоритм зчитує по бітно файл, при цьому за прохід береться 6 біт і відповідно за кожною основою формується новий файл, який містить масив символів ASCII, які отримуються при склеюванні чисел після застосування алгоритму системи залишкових класів. При цьому кожне число повинно займати три символи, це вирішується додаванням необхідної кількості нулів перед числом. До прикладу, «7» буде представлене у вигляді «007» і т. д. [18].

Процес отримання початкового файлу користувача розпочинається із завантаження трьох відповідних файлів із хмарного сховища. Далі система використовує надлишкову систему залишкових класів для отримання масивів залишків. Алгоритм включає цикл, під час якого кожен елемент зчитується з кожного масиву, і застосовується обернений процес відновлення десяткового числа згідно з системою залишкових класів. Отримані результати записуються до основного масиву у десятковій системі числення.

Далі система циклічно зчитує основний масив, розбиваючи кожен елемент на три символи відповідно кодуванню в системі ASCII. З отриманого масиву байтів обчислюється хеш-код, який порівнюється із початковим. У разі співпадіння система запускає процес формування вихідного файлу користувача. У випадку, якщо хеш-коди не збігаються, користувач отримує повідомлення про помилку.

Цей алгоритм забезпечує безпеку та цілісність отриманих даних, використовуючи залишкові класи та хеш-код для перевірки вірності і цілісності інформації.

Роботу алгоритму для відновлення початкового файлу користувача зображено на рисунку 3.5.

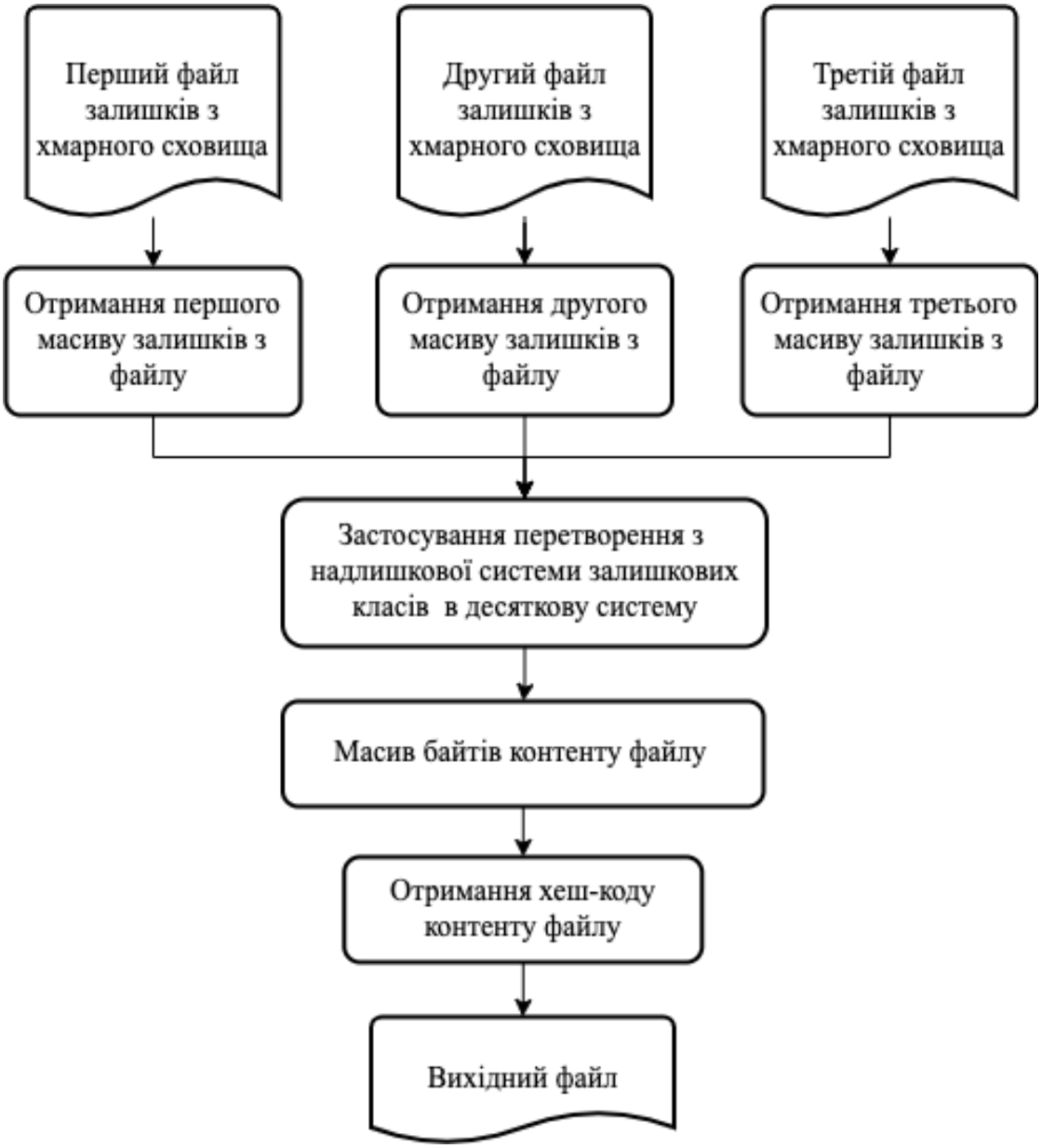


Рисунок 3.5 – Схема перетворення файлів-залишків у початковий

3.2 Реалізація системи надійного зберігання даних

Для реалізації мобільного додатку для реалізації системи надійного зберігання даних для платформи Android з використанням ряду сучасних інструментів та технологій. Вибір Android як основної платформи був обґрунтований її широким розповсюдженням та популярністю серед користувачів мобільних пристроїв. Мова програмування Kotlin використовувалась для реалізації застосунку. Kotlin надає зручний та сучасний синтаксис, що полегшує написання коду та покращував його читабельність.

Архітектурний шаблон MVVM (Model-View-ViewModel) використовується для ефективного розділення бізнес-логіки та відображення. Цей підхід сприяє збереженню чіткості та організації кодової бази, а також полегшує тестування. Для управління асинхронним кодом та фоновими задачами використовувались Coroutines. Це дозволяло ефективно уникати громіздкості коду та забезпечувало його кращу читабельність. Механізм Dependency Injection (DI) було реалізовано за допомогою бібліотеки Hilt. DI сприяє ефективному управлінню залежностями в додатку, забезпечуючи його гнучкість та зручність у розширенні. Структура проекту зображена на рисунку 3.7. [19, 20]

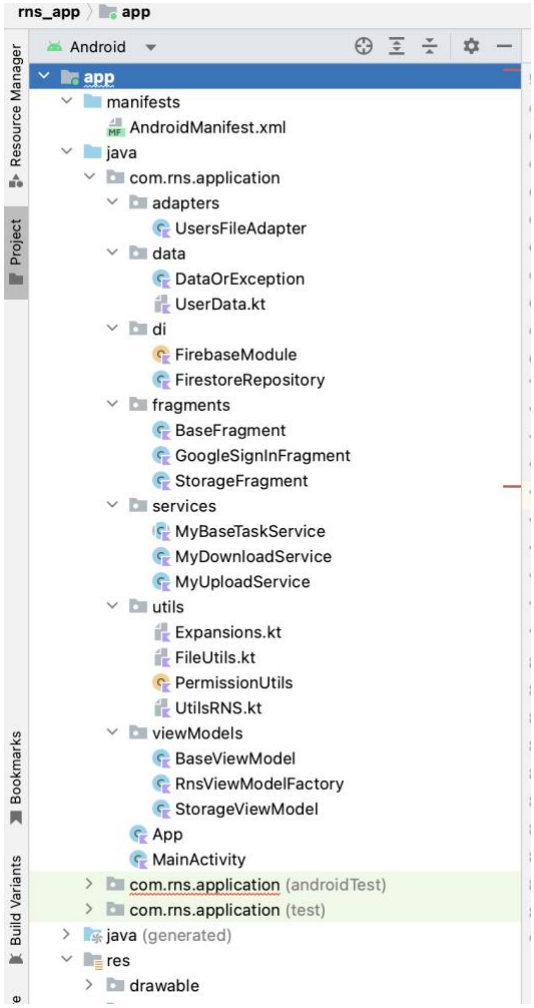


Рисунок 3.7 – Структура проекту мобільного додатку

Написання проекту відбувається в Android Studio, яка інтегрованим середовищем розробки (IDE), призначене для створення мобільних додатків на

платформі Android. Середовище включає інтелектуальний редактор коду з автозаповненням та підказками, графічний дизайнер та редактор ресурсів для зручного створення інтерфейсу користувача. Його інтеграція з Android SDK дозволяє ефективно керувати версіями та компонентами SDK, а також створювати та керувати віртуальними пристроями для тестування. Збірка та запуск відбуваються за допомогою Gradle Build System, який дозволяє налаштовувати конфігурації для різних пристроїв та версій Android. Android Studio також надає інструменти для тестування, включаючи вбудований емулятор Android та засоби для тестування та профілювання додатків. Інтеграція з системою контролю версій Git, підтримка мов програмування Kotlin та Java, а також моніторинг використання ресурсів за допомогою Android Profiler роблять Android Studio потужним інструментом для розробників Android-додатків. [21-25]

На рисунку 3.7 зображено стартові екрани мобільного додатку, екран запуску додатку та екран авторизації додатку за допомогою облікових даних Google.

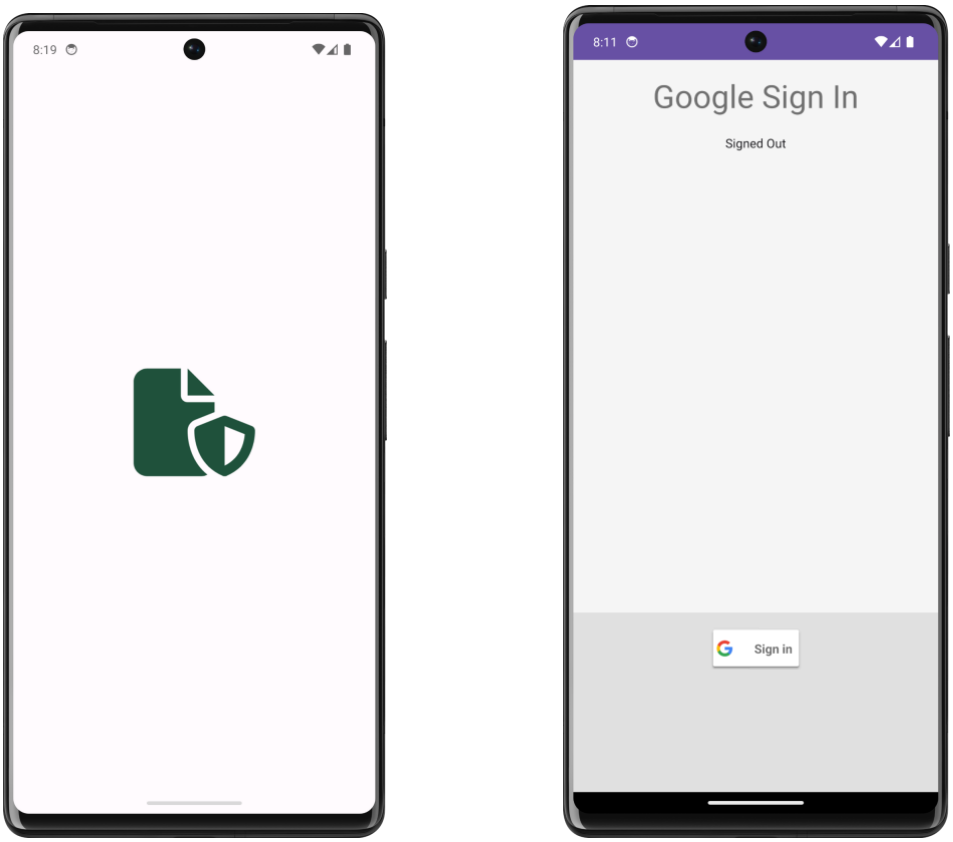


Рисунок 3.7 – Стартові екрани мобільно додатку

Аутентифікація відбувається за допомогою Firebase Authentication. Після успішного входу в додаток, користувач потрапляє на екран (Fragment) з відображенням файлів користувача,, які доступні до завантаження з хмарного сховища Google, та функціональні кнопки, що зображено на рисунку 3.8. [26]

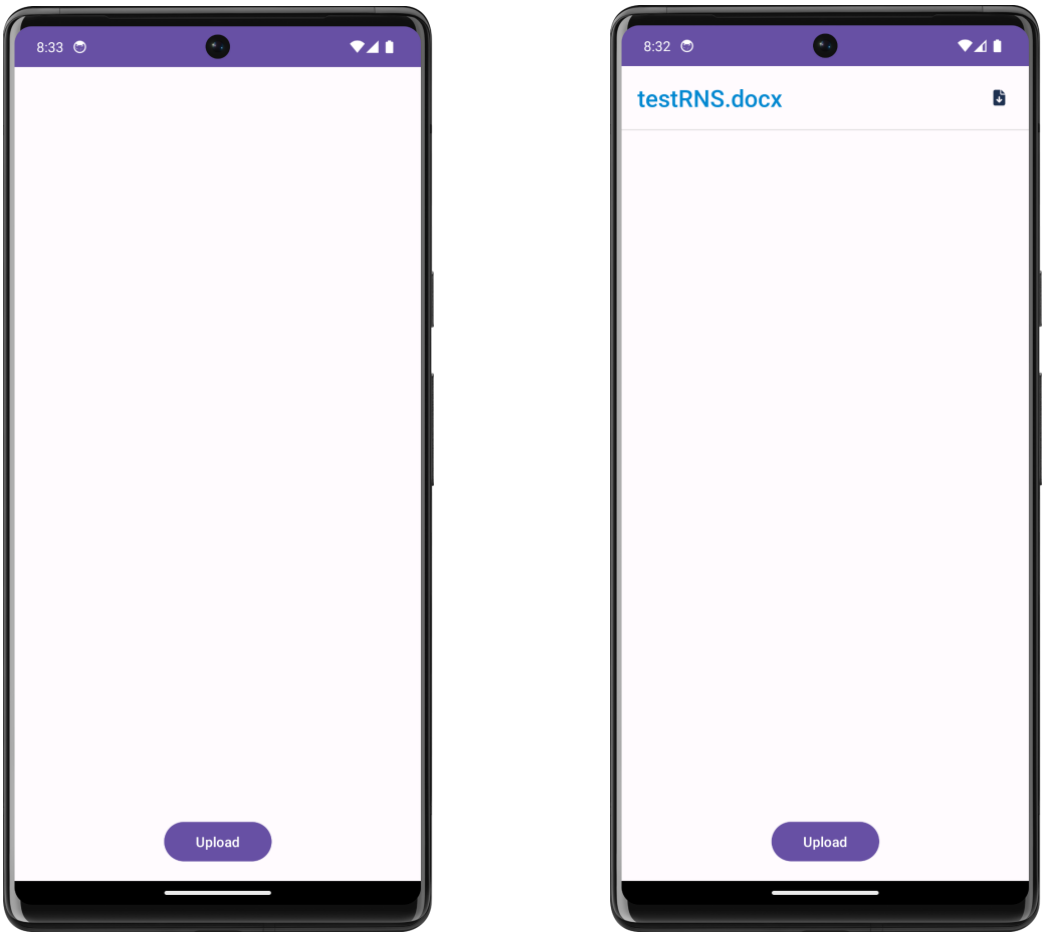


Рисунок 3.8 – Список файлів користувача та функціональні клавіші

Кнопка “Upload” відповідає за функціонал вибору файлу з пристрою користувача до якого застосується процес шифрування на основі надлишкової системи залишкових класів. Після отриманих файлів залишків запуститься процес вивантаження їх на хмарне сховище.

Натиснувши на елемент списку вибраного файлу, буде запущено обернений процес отримання дешифрованого файлу користувача. Першочергово відбудеться процес завантаження файлів з залишками, які відповідають файлу, також з бази даних буде отримана інформація про основи, які відповідають кожному з файлів залишків. Далі за методикою надлишкової системи

залишкових класів відбудеться процес дешифрування файлу і отримання першопочаткової інформації з файлу.

Для реалізації застосування надлишкої системи залишкових класів було написано функцію для отримання трьох основ, які задовільняють умовам, реалізація функції зображена на рисунку 3.9.

```
fun generateCoprimeNumbers(): List<Long> {
    val random1 : Random  = Random(System.currentTimeMillis())
    val start : Long  = (900000L ≤ .. ≤ 900999L).random(random1)

    val random2 : Random  = Random(System.currentTimeMillis())
    val end : Long  = (start+1 ≤ .. ≤ 909999L).random(random2)

    require( value: start <= end) { "The start value must be less than or equal to the end value." }

    return (start ≤ .. ≤ end)
        .filter { number -> (2 ≤ until  < number).all { number % it != 0L } }
        .take( n: 3)
}
```

Рисунок 3.9 – Функція отримання трьох взаємно простих чисел

Для відновлення першочергово файлу було розроблено функцію для розширеного алгоритма Евкліда зображено на рисунку 3.10.

```
fun extendedEuclideanAlgorithm(a: BigInteger, b: BigInteger): Triple<BigInteger, BigInteger, BigInteger> {
    if (b == BigInteger.ZERO) {
        return Triple(a, BigInteger.ONE, BigInteger.ZERO)
    }

    val (gcdPrev : BigInteger , xPrev : BigInteger , yPrev : BigInteger ) = extendedEuclideanAlgorithm(b,  b: a % b)
    val y : BigInteger  = xPrev - (a / b) * yPrev

    return Triple(gcdPrev, yPrev, y)
}

fun findModularInverse(b: BigInteger, m: BigInteger): BigInteger {
    val (gcd : BigInteger , x : BigInteger , _ : BigInteger ) = extendedEuclideanAlgorithm(b, m)
    require( value: gcd == BigInteger.ONE) { "Numbers are not coprime" }

    return (x + m) % m
}
```

Рисунок 3.10 – Функція для реалізації розширеного алгоритма Евкліда

Для реалізації захищеної системи зберігання даних на основі надлишкової системи залишкових класів було написано відповідний функціонал шифрування, реалізація функції якої зображено на рисунку 3.10.

```
fun encryptFile(uriFile: Uri) {
    saveUriFile = uriFile
    setShowProgress(true)
    viewModelScope.launch(Dispatchers.IO) { this: CoroutineScope
        val byteArray : ByteArray = getByteArrayFromDocx(context, uriFile)
        contentHashCode = byteArray.contentHashCode()

        val listBigInteger : List<BigInteger> = byteArrayToBigIntegerArray(byteArray!!)
        generatedCoprimeNumbers = generateCoprimeNumbers()

        val reminders1: ArrayList<BigInteger> = arrayListOf()
        val reminders2: ArrayList<BigInteger> = arrayListOf()
        val reminders3: ArrayList<BigInteger> = arrayListOf()

        for(i : Int in listBigInteger.indices) {
            reminders1.add(listBigInteger[i].remMod(generatedCoprimeNumbers[0].toBigInteger()))
            reminders2.add(listBigInteger[i].remMod(generatedCoprimeNumbers[1].toBigInteger()))
            reminders3.add(listBigInteger[i].remMod(generatedCoprimeNumbers[2].toBigInteger()))
        }

        saveArrayToFile(context, reminders1, reminders1.hashCode().toString())?.let { uri ->
            addRnsUriFileSend(uri)
        }
        saveArrayToFile(context, reminders2, reminders2.hashCode().toString())?.let { uri ->
            addRnsUriFileSend(uri)
        }
        saveArrayToFile(context, reminders3, reminders3.hashCode().toString())?.let { uri ->
            addRnsUriFileSend(uri)
        }
    }
}
```

Рисунок 3.10 – Реалізація функції шифрування файлу

При реалізації алгоритму дешифрування файлу на основі надлишкової системи залишкових класів у мобільному додатку, враховано кілька важливих кроків, які сприяють ефективності та безпеці операцій з розшифруванням.

Функціонал дешифрування був успішно реалізований для кожного залишкового класу, використовуючи необхідний приватний ключ та інформацію про залишок. Це дозволяє відновлювати оригінальні дані з зашифрованого файлу відповідно до вибраної надлишкової системи. Забезпечена правильна обробка та взаємодія залишкових класів гарантує коректність розшифрування, усуває

можливі конфлікти та дозволяє використовувати систему залишкових класів для максимальної точності відновлення оригінальної інформації. Алгоритм був оптимізований для підвищення продуктивності, спрощуючи операції залишку та використовуючи інші оптимізації обчислень. Це не лише забезпечує швидке розшифрування, але і дозволяє оптимально використовувати ресурси мобільного пристрою.

Такий підхід до реалізації алгоритму дешифрування в мобільному додатку гарантує не лише його ефективність, але і високий рівень безпеки при обробці конфіденційних даних. Реалізація функції процесу дешифрування зображено на рисунку 3.11.

```
fun decryptFile() {
    selectedUserFiles?.let { userFile ->
        setShowProgress(true)
        generatedCoprimeNumbers = userFile.basicKeys
        userFile.rnsFiles.let { rnsFiles ->
            val reminders1 : ArrayList<BigInteger> = readArrayFromFile(context, rnsFiles[0].getNameFile())
            val reminders2 : ArrayList<BigInteger> = readArrayFromFile(context, rnsFiles[1].getNameFile())
            val reminders3 : ArrayList<BigInteger> = readArrayFromFile(context, rnsFiles[2].getNameFile())

            val moduli : BigInteger =
                generatedCoprimeNumbers.fold( initial: 1L) { acc, base -> acc * base }.toBigInteger()
            val decryptListDecimal: ArrayList<BigInteger> = arrayListOf()
            for (rem : Int in 0 ≤ ..<reminders1.count()) {
                val x : BigInteger = generatedCoprimeNumbers.indices.sumOf { i ->
                    val mi : BigInteger = moduli / generatedCoprimeNumbers[i].toBigInteger()
                    val yi : BigInteger = findModularInverse(mi, generatedCoprimeNumbers[i].toBigInteger())
                    return@sumOf when (i) {
                        0 -> reminders1[rem] * mi * yi
                        1 -> reminders2[rem] * mi * yi
                        else -> reminders3[rem] * mi * yi
                    }
                } % moduli
                decryptListDecimal.add(x)
            }
            bigIntegerArrayToByteArray(decryptListDecimal).let { it: ByteArray
                if (it.contentHashCode() == userFile.contentHashCode) {
                    writeDocxFromByteArray(context, it, userFile.id)
                } else {
                    Log.d(StorageFragment.TAG, msg: "Хеші не співпадають")
                }
            }
        }
    }
    setShowProgress(false)
    Log.d(StorageFragment.TAG, msg: "END DECRYPT FILE")
}
```

Рисунок 3.11 – Реалізація функції дешифрування файлу

3.3 Тестування та налагодження системи

Захищена система зберігання даних на основі надлишкової системи залишкових класів та хмарних сервісів досягається за допомогою інтеграції передових методів криптографії та сучасних технологій хмарних обчислень. Цей підхід забезпечує високий рівень безпеки та надійності при зберіганні та обробці конфіденційних даних. Надлишкова система залишкових класів (RNS) в поєднанні з хмарними сервісами створює високоефективну та стійку до зломів інфраструктуру для зберігання і обробки важливої інформації.

Для проведення тестування буде використано документ "*.docx", що є форматом документів типу “Word” із включеним текстом. Вміст такого документу представлено на рисунку 3.12.

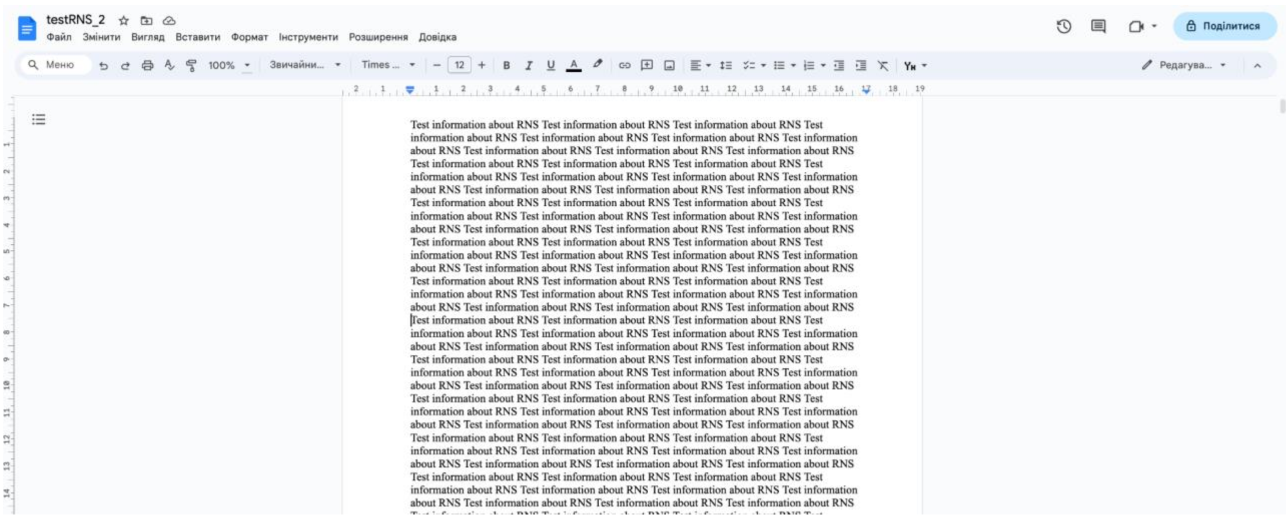


Рисунок 3.12 – Вміст текстового документа для тестування

Документ був попередньо завантажений на мобільний телефон Pixel 6 Pro з операційною системою Android 14, на даному пристрої буде відбуватися тестування роботи додатку. В залежності від моделі телефону та версії операційної системи Android, результати можуть відрізнятися.

Спочатку користувач авторизовується в мобільному додатку за допомогою відповідних облікових даних Google. Потрапивши на екран де в нижній частині присутня функціональна кнопка “Upload”, при натисканні на яку відкриється

екран для вибору файлу “testRNS.docx”. На рисунку 3.13 відображено процес вибору файлу з мобільного пристрою.

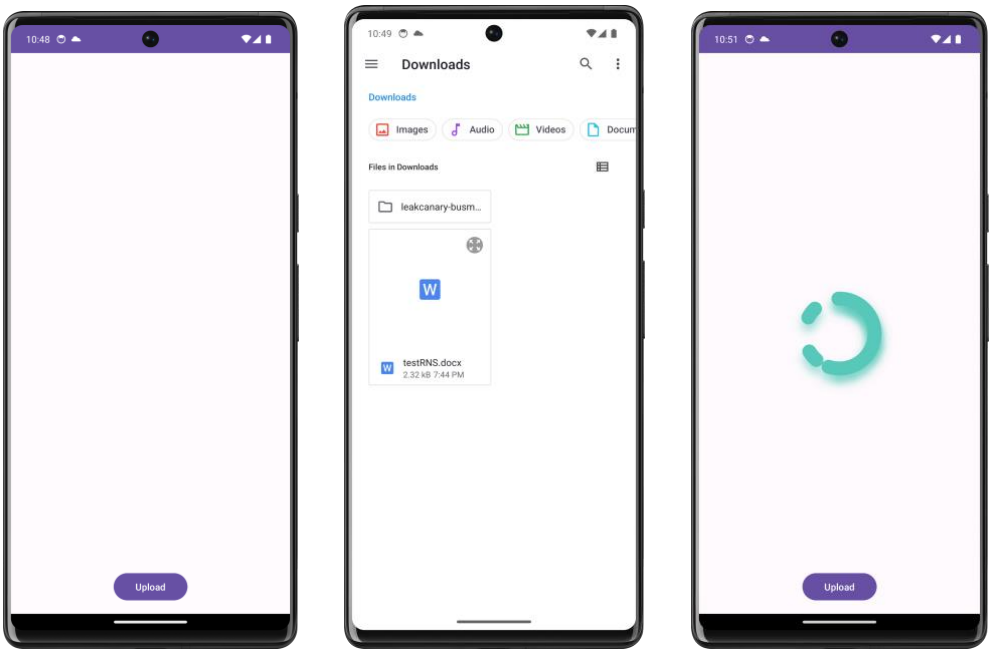


Рисунок 3.13 Процес вибору файлу з мобільного телефону

Після вибору файлу запуститься процес шифрування за допомогою алгоритму надлишкової системи залишкових класів. При успішному процесі формування трьох файлів залишків вони будуть завантажені на хмарне сховище, зайшовши на Firebase Console в пункті Storage видно дані файли на рисунку 3.14 зображено результат. [27]

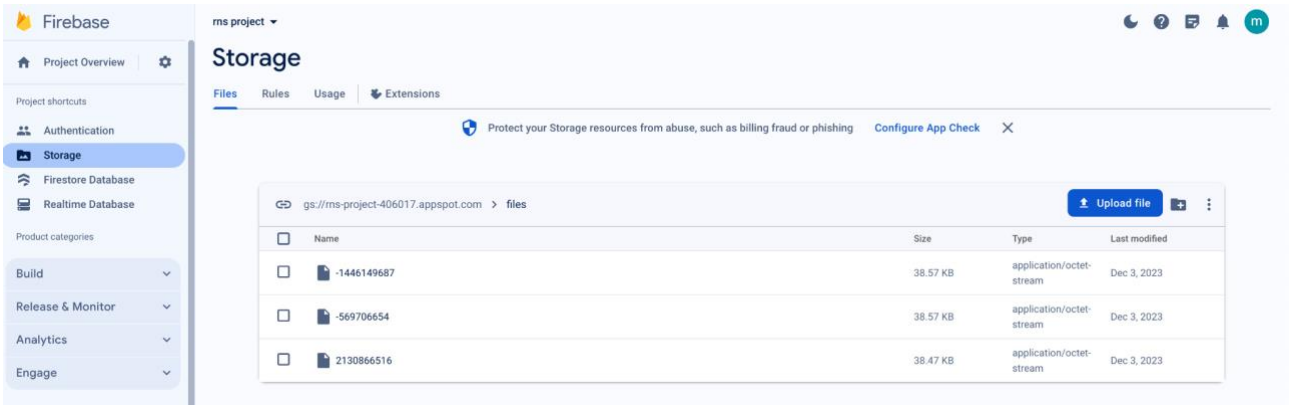


Рисунок 3.14 – Файли залишки у Firebase Storage

Паралельно з цим буде здійснено відповідний запис з додатковою інформацією у базу даних Firestore Database, який відображено на рисунку 3.15.

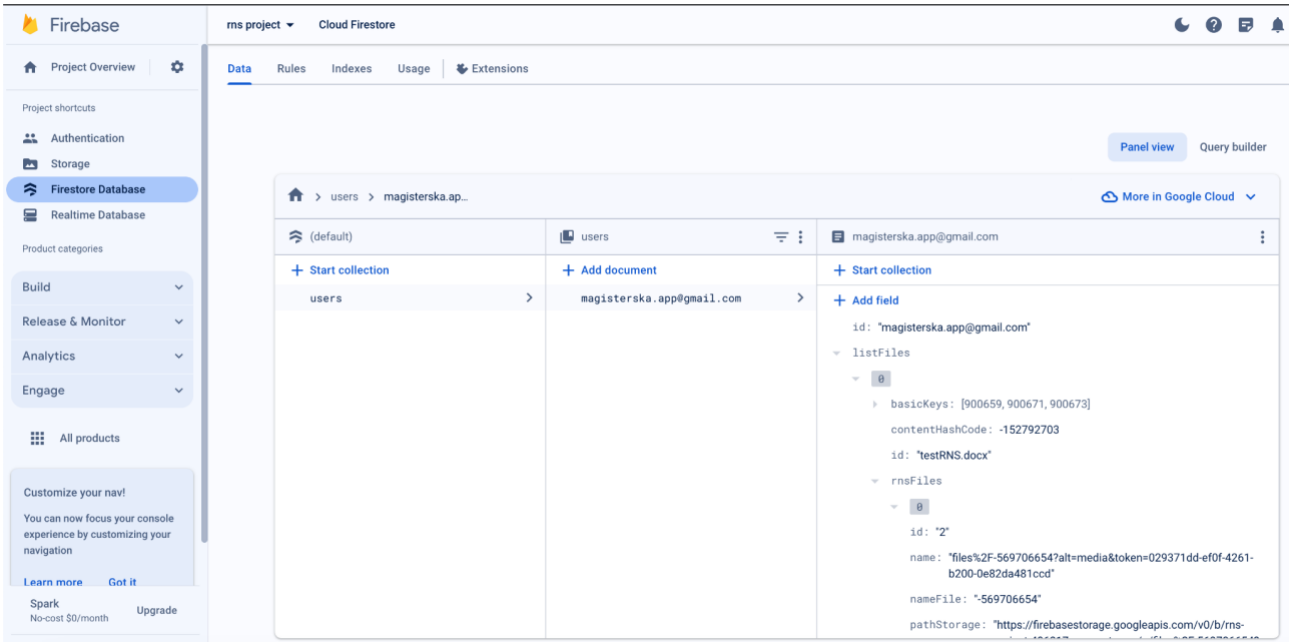


Рисунок 3.15 – Запис в базі даних Firestore Database

Як результат успішного застосування надлишкової системи залишкових класів, для користувача на екрані телефону в списку файлів з’явиться елемент з назвою файлу, який був вивантажений на хмарний сервіс, що зображено на рисунку 3.16.

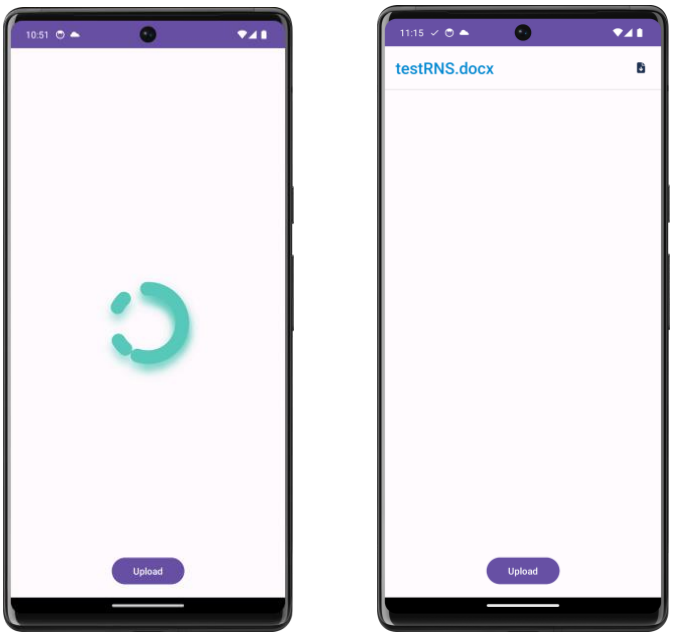


Рисунок 3.16 – Результат вивантаженого файлу

В подальшому натиснувши на відповідний елемент списку, вибраний файл, запускається процес завантаження файлів залишків і відновлення файлу на пристрої користувача.

ВИСНОВКИ

Випускна кваліфікаційна робота пройшла через комплексне дослідження та ретельний аналіз проблематики сучасних систем зберігання даних. У ході вивчення та дослідження системи зберігання даних було виявлено, що головною метою є підвищення ефективності цієї системи. В рамках виконаної роботи були визначені основні принципи та технічні аспекти реалізації захищеної системи зберігання даних на основі надлишкової системи залишкових класів та хмарних сервісів. Було визначено оптимальні стратегії для забезпечення надійності та високої продуктивності системи, а також вивчено можливості застосування шифрування та інших заходів забезпечення безпеки.

1. У ході дослідження визначено, що хмарні сервіси представляють собою інноваційний підхід до забезпечення доступу до обчислювальних ресурсів та збереження даних через Інтернет. Вони дозволяють користувачам отримувати доступ до великої кількості ресурсів без необхідності утримання власної інфраструктури.

2. Розглянуто основних постачальників хмарних послуг, виявлені їхні переваги та недоліки. Кожен з провайдерів має свої унікальні характеристики, і вибір між ними повинен враховувати конкретні потреби та вимоги користувача.

3. Проаналізовані різні популярні хмарні сервіси, приділено увагу їхнім можливостям та обмеженням. Визначено основні фактори, які слід враховувати при виборі конкретного сервісу для зберігання даних.

4. Вивчено та впроваджено надлишкову систему залишкових класів у структурі зберігання даних. Цей підхід дозволяє підвищити надійність та конфіденційність безпечної роботи системи.

5. Визначено та розглянуто характеристики захищеної системи зберігання даних. Застосування відповідних заходів безпеки, таких як ідентифікація, автентифікація та авторизація, дозволяє забезпечити конфіденційність та цілісність збережених даних. Також враховано фізичну та логічну структуру системи для оптимального захисту інформації.

6. Розглянуто та впроваджено шифрування даних як важливий аспект безпеки в системі зберігання. Шифрування дозволяє захистити інформацію від несанкціонованого доступу, забезпечуючи конфіденційність даних навіть у випадку їхньої втрати чи крадіжки.

7. Розроблено та реалізовано прототип системи, який враховує всі виявлені аспекти та вимоги. Тестування прототипу підтвердило його ефективність та високий рівень захищеності. Важливим етапом в роботі було врахування сучасних тенденцій та підходів до зберігання даних, а також врахування потреб користувачів у сфері безпеки та надійності.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. What are cloud services? [Електронний ресурс]. - Режим доступу: https://www.redhat.com/en/topics/cloud-computing/what-are-cloud-services

2. What is a Cloud Service? – Cloud Services Solutions [Електронний ресурс]. - Режим доступу: https://www.citrix.com/solutions/digital-workspace/what-is-a-cloud-service.html

3. Що таке хмарні сервіси [Електронний ресурс]. - Режим доступу: https://ucloud.ua/shho-take-hmarni-servisy/

4. Cloud Services [Електронний ресурс]. - Режим доступу: https://www.hpe.com/us/en/what-is/cloud-services.html

5. Amazon Maintains Lead in the Cloud Market. [Електронний ресурс]. - Режим доступу: https://www.statista.com/chart/18819/worldwide-market-share-of-leading-cloud-infrastructure-service-providers/

6. Топ 5 хмарних провайдерів: плюси та мінуси. [Електронний ресурс]. - Режим доступу: https://blog.iteducenter.ua/ru/sysadministration/top-5-cloud-providers-advantages-and-disadvantages/

7. Cloud Storage [Електронний ресурс]. - Режим доступу: https://cloud.google.com/learn/what-is-cloud-storage

8. Garth A. Network attached storage architecture / Garth A. Gibson and Rodney Van Meter // Communications of the ACM. – New York, 2000. – Vol. 43 (№ 11). – P. 37–45. 2

9. Хмарні сервіси для збереження даних [Електронний ресурс]. - Режим доступу: https://ucloud.ua/hmarni-servisy-dlya-zberezhennya-danyh/

10. Система числення [Електронний ресурс]. - Режим доступу: https://uk.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0_%D1%87%D0%B8%D1%81%D0%BB%D0%B5%D0%BD%D0%BD%D1%8F

11. Xiao H. New Error Control Algorithms for Residue Number System Codes / Xiao H., Garg H. K., Hu J., & Xiao G // ETRI Journal. – 2016. – Vol. 38 (№ 2). – P. 326–336.

12. Акушский И. Я. Машинная арифметика в остаточных классах / Акушский И. Я., Юдицкий Д.И. – М. : Сов. радио. 1968. – С. 460.

13. Системи залишкових класів [Електронний ресурс]. – Режим доступу: https://studfile.net/preview/5082748/page:4/

14. Data Encryption: The Ultimate Guide [Електронний ресурс]. – Режим доступу: https://cloudian.com/guides/data-protection/data-encryption-the-ultimate-guide/

15. Cloud Firestore [Електронний ресурс]. – Режим доступу: https://firebase.google.com/docs/firestore

16. What is NoSQL? [Електронний ресурс]. – Режим доступу: https://cloud.google.com/discover/what-is-nosql

17. Working with JSON [Електронний ресурс]. – Режим доступу: https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Objects/JSON

18. ASCII Table [Електронний ресурс]. – Режим доступу: https://www.ascii-code.com/

19. What is Android [Електронний ресурс]. – Режим доступу: https://www.android.com/what-is-android/

20. Kotlin for Android [Електронний ресурс]. – Режим доступу: https://kotlinlang.org/docs/android-overview.html

21. Android Studio [Електронний ресурс]. – Режим доступу: https://developer.android.com/studio

22. What is a software development kit (SDK)? [Електронний ресурс]. – Режим доступу: https://www.adjust.com/glossary/sdk/

23. Inspect your app's memory usage with Memory Profiler [Електронний ресурс]. – Режим доступу: https://developer.android.com/studio/profile/memory-profiler

24. About GIT [Електронний ресурс]. – Режим доступу: https://git-scm.com/about

25. Gradle User Manual [Електронний ресурс]. – Режим доступу: https://docs.gradle.org/current/userguide/userguide.html?_gl=1*1uiija*_ga*MTc1ND

I1NDI0OS4xNzAxMzM4MDE3*_ga_7W7NC6YNPT*MTcwMjU2MzYxNy4yLjAuMTcwMjU2MzYxNy42MC4wLjA.

26. Firebase Authentication [Електронний ресурс]. – Режим доступу: https://firebase.google.com/docs/auth

27. Make your app the best it can be [Електронний ресурс]. – Режим доступу: https://firebase.google.com/

ДОДАТОК А.

Копія публікацій

ГРОМАДСЬКЕ ОБЄДНАННЯ
«КІБЕРБЕЗПЕКА І АВТОМАТИЗАЦІЯ»

Матеріали
науково-практичного симпозіуму
«ЗАХИСТ ІНФОРМАЦІЇ»

30 листопада 2023
Тернопіль

розвиватися природним шляхом, і безпечним, щоб робити це без загроз.

Агенти. Агенти — це всі люди, чиї дії впливають на екосистему IoT, до них відносяться як інженери, що розгортали IoT та і оператори платформ. IoT - це складна екосистема створена з конкретним завданням, а саме – для підвищення ефективності та покращення якості життя. І саме агенти вирішують, як використовувати пристрої, мережі та платформи для досягнення цих результатів.

Саме на цьому етапі технології та бізнес сходяться, оскільки саме бізнес-цілі значною мірою формують екосистему IoT, проте люди також є важливою частиною цього рівняння. Вони створюють системи, керують ними, і, зрештою, саме вони несуть відповідальність за реалізацію їх повного потенціалу.

Висновки.

Пристрої збирають дані, але саме люди розуміють їх та використовують. Так само з мережами та платформами, які є необхідним компонентом системи, але не мали б великої цінності, якби не люди, які створюють і вдосконалюють їх відповідно до своїх потреб. Проте, це все фізичні складові системи IoT, окрім них чітко виділяють логічні складові, розуміння яких допомагає краще усвідомити справжнє значення та функціональність IoT.

Перелік використаних джерел

1. Upadhyay, P., & Upadhyay, D. (2021). Internet of things - A Survey. International Journal of Scientific Research in Computer Science, Engineering and Information Technology, 417–438. https://doi.org/10.32628/cseit217394

2. Komilov, D. R. (2023). Application of zigbee technology in IOT. International Journal of Advance Scientific Research, 3(09), 343-349.

3. Sadeeq, M. M., Abdulkareem, N. M., Zeebaree, S. R., Ahmed, D. M., Sami, A. S., & Zebari, R. R. (2021). IoT and Cloud computing issues, challenges and opportunities: A review. Qubahan Academic Journal, 1(2), 1-7.

УДК 004.056.5

ШЕСТЕРИНА С. В.[1]

[1] *Західноукраїнський національний університет*

СТРУКТУРА ЗАХИЩЕНОЇ СИСТЕМИ ЗБЕРІГАННЯ ДАННИХ

Вступ. У сучасному цифровому суспільстві, де обсяги інформації ростуть експоненційно, проблема зберігання та забезпечення безпеки цих даних стає все більш актуальною. Особливо важливою є задача захисту конфіденційної інформації від несанкціонованого доступу та втрати.

Мета: Аналіз та оцінка структури захищеної системи зберігання данх з метою забезпечення не тільки ефективного зберігання, але й високого рівня безпеки.

1. Захищена система зберігання даних та її характеристики

Захищена система зберігання даних (ЗСЗД) - це система, що включає в себе програмне забезпечення та спеціалізоване обладнання, яке в комплексі забезпечує захист даних від несанкціонованого доступу, зміни чи знищення. Вона має такі характеристики [1]:

1. Конфіденційність - означає, що лише уповноважені користувачі можуть отримати доступ до даних. Для забезпечення конфіденційності даних можуть використовуватися такі технології:

- Шифрування даних, які можливо прочитати з допомогою секретного ключа.
- Аутентифікація використовується для перевірки особи користувача. Реалізована за допомогою таких методів, як паролі, біометрія або двофакторна аутентифікація.
- Контроль доступу використовується для обмеження доступу до даних авторизованих користувачів. Може бути реалізований за допомогою таких методів, як рольова модель або політика доступу на основі атрибутів.

2. Цілісність - означає, що дані не були змінені чи знищені без дозволу. Для забезпечення цілісності даних можуть використовуватися такі технології:

- Резервне копіювання для створення копії даних, яка може бути використана для відновлення даних у разі втрати або пошкодження.
- Моніторинг для виявлення змін у даних, реалізований за допомогою таких методів, як журналювання змін або використання систем виявлення вторгнень.
- Відновлення у разі втрати чи пошкодження. Може бути виконане за допомогою резервної копії або за допомогою технології відновлення даних.

3. Доступність - означає, що дані можуть бути доступні користувачам у разі потреби. Можуть використовуватися такі технології:

- Розподілені системи зберігання даних розподіляють дані між кількома серверами.
- Реплікація даних створює копії даних на одному чи кількох серверах.
- Модульні системи зберігання даних дозволяють легко масштабувати сервери.

Вибір захищеної системи зберігання даних залежить від конкретних потреб організації. При виборі системи слід враховувати такі фактори:

- Тип даних, які потрібно зберігати: Деякі типи даних, такі як фінансові дані, вимагають більш високого рівня безпеки, ніж інші типи даних.
- Кількість даних, які потрібно зберігати: Великі обсяги даних можуть вимагати використання спеціальної системи зберігання даних.
- Сервіси, які необхідні: Деякі захищені системи зберігання даних включають додаткові послуги, такі як резервне копіювання та відновлення.

До прикладів захищених систем зберігання даних належать:

- Системи зберігання даних на основі хмарних технологій часто включають вбудовані функції безпеки, такі як шифрування та контроль доступу.
- Системи зберігання даних на базі штучного інтелекту можуть

використовуватися для виявлення та запобігання вторгненням.

– Системи зберігання даних на базі блокчейну забезпечують підвищений рівень безпеки за рахунок використання децентралізованої мережі.

2. Розробка захищеної системи зберігання даних на основі надлишкової системи залишкових класів

Ідея роботи полягає у тому, що підвищення захищеності системи зберігання даних можна досягти шлях застосування надлишкової системи залишкових класів (НСЗК). Використаємо даний підхід, як засіб шифрування інформації [2, 3].

Для прикладу візьмемо десяткове число 69 і перетворимо його в систему залишків класів (СЗК). Для потрібно мати базиси, які повинні виступати взаємно простими числами, твердження що виникло з праць видатних математиків, таких як Леонард Ейлер, Карл Фрідріх Гаусс та Пафнутий Чебишов. Ці математики допомогли сформулювати та оптимізувати концепції СЗК [4].

Виходячи з викладеного вище використаємо для перетворення наступні базиси {5, 7, 9} , які є взаємно простими числами. Знайдемо залишки:

$$a_1 = 69 - (69 \bmod 5) * 5 = 69 - 65 = 4;$$
$$a_1 = 69 - (69 \bmod 7) * 7 = 69 - 63 = 6;$$
$$a_1 = 69 - (69 \bmod 9) * 9 = 69 - 63 = 6;$$

Звідси, число 69 може бути представлене в СЗК у вигляді сукупності цифр {4, 6, 6}, або в двійковій системі як 100, 110, 110.

При таких основах як {5, 7, 9} максимальне число, яке можливо представити згідно формули $R = p_1 * p_2 * p_3 * \dots * p_n$ буде $R = 5 * 7 * 9 = 315$.

Для зворотнього конвертування числа з числової системи залишкових класів здійснюється за відповідною формулою:

$$N = (a_1 * B_1 + a_2 * B_2 + a_3 * B_3) - r * R;$$

де r – ранг, котрий приймає значення 0, 1, 2, … так, щоб права частина даного виразу була менша значенню R;

B_i – ортогональний базис, що визначається при виборі базису СЗК.

$$B_i = k_i * \frac{R}{p_i};$$

де k_i – ціле додатне число ($k_i = 1,2, \dots p_{i-1}$), при цьому k_i вибирається таким, щоб залишок від ділення B_i на p_i дорівнював одиниці (тобто вага базису k_i вибирається, виходячи з рівності) [19]:

$$B_i = \frac{m_i * R}{p_i} = k_i * p_i + 1;$$

При базисах {5, 7, 9}, перетворимо наше число відображене у ЧСЗ як {4, 6, 6} у десяткову систему за допомогою наступних виразів:

$$B_1 = \frac{k_1 * 315}{5} = k_1 * 63 = 2 * 63 = 126;$$

для $p_1 = 5$ умова виконується при $k_1 = 2$.

$$B_2 = \frac{k_2 * 315}{7} = k_2 * 45 = 5 * 45 = 225;$$

для $p_2 = 7$ умова виконується при $k_2 = 2$.

$$B_3 = \frac{k_3 * 315}{9} = k_3 * 35 = 8 * 35 = 280;$$

для $p_3 = 9$ умова виконується при $k_3 = 8$.

Наступним кроком згідно формули:

$$N = (a_1 * B_1 + a_2 * B_2 + a_3 * B_3) - r * R;$$

Підставимо відповідні значення і отримаємо вираз:

$$N = (4 * 126 + 6 * 225 + 6 * 280) - r * 315;$$

$$N = (504 + 1350 + 1680) - r * 315;$$

$$N = 3534 - r * 315;$$

де згідно умов $r = 11$, відповідно на число в десятковій системі:

$$N = 3534 - 3465 = 69;$$

Система залишків класів полягає в тому, що цифри в кожному розряді числа є незалежними один від одного і не залежать від позиції, а лише від базису.

На початок 2023 року 5,44 мільярда людей користуються мобільними телефонами, що становить 68% загальної чисельності населення світу. Кількість унікальних користувачів мобільних телефонів за останній рік збільшилася більш ніж на 3% — на 168 мільйонів нових користувачів. Це показує, що користувачі все частіше використовують мобільні додатки в повсякденному житті, що зробило наше життя простішим і зручнішим. Однак зі зростанням насиченості ринку застосунків користувачі стали більш розбірливими щодо програм, які вони завантажують і використовують. Зросла потреба і в надійності зберігання конфіденційної інформації користувача.

Тому, для створення програмного комплексу для надійного зберігання даних користувача на основі НСЗК в хмарних сервісах було розроблено мобільний додаток на базі операційної системи Android та GOOGLE DRIVE.

Концепція роботи даного комплексу полягає в надійному зберіганні користувацьких файлів з мобільного пристрою на хмарному сервісі GOOGLE DRIVE. Робота мобільного додатку полягає в тому, щоб користувач міг вибрати файл який за допомогою запрограмованого алгоритму на основі надлишкової системи залишкових класів здійснить його шифрування і подальше розміщення на хмарному сервісі. Власник файлу також має можливість його завантажити та прочитати, після процесу дешифрування. Даний підхід забезпечить більшу надійність для збереження даних і їх подальше використання.

Процес шифрування полягає в отриманні трьох файлів, які в собі містять залишки СЗК. В першу чергу алгоритм перетворює файл в масив байтів ASCII. Наприклад тестовий файл в системі ASCII буде мати вигляд, як набір чисел, що дозволить застосувати надлишкову систему залишкових класів. Для того, що почати процес перетворення потрібно отримати певні числа (основи), які застосовані при обчисленні залишків. Дані числа будуть запропоновані користувачеві за принципом генерації взаємно простих чисел і їхній добуток не перевищуватиме 255255255255255.

Після отриманого масиву байтів ASCII та основ, розпочинається робота алгоритму перетворення з десяткової системи числення в систему залишкових класів. В циклі алгоритм зчитує по бітно файл, при цьому за прохід береться 8 біт

і відповідно за кожною основою формується новий файл, який містить масив символів ASCII, які отримуються при склеюванні чисел після застосування алгоритму системи залишкових класів. При цьому кожне число повинно займати три символи, це вирішується додаванням необхідної кількості нулів перед числом. До прикладу, «7» буде представлене у вигляді «007» і т. д.

Висновки.

Використання такого підходу виявляється цінним у забезпеченні високого рівня конфіденційності та надійності інформації. Зазначимо, що врахування та впровадження надлишкових систем залишкових класів у сфері зберігання даних визначає новий стандарт для надійності та стійкості цифрових інфраструктур. Такий підхід відкриває перспективи для створення систем, які можуть ефективно працювати в умовах постійно змінних викликів у сфері кібербезпеки і не тільки.

Перелік використаних джерел.

1. Xiao H. New Error Control Algorithms for Residue Number System Codes / Xiao H., Garg H. K., Hu J., & Xiao G // ETRI Journal. – 2016. – Vol. 38 (№ 2). – P. 326–336.

2. Акушский И. Я. Машинная арифметика в остаточных классах / Акушский И. Я., Юдицкий Д.И. – М. : Сов. радио. 1968. – С. 460.

3. Кулина С.В. Виявлення помилок на основі коригуючих кодів системи залишкових класів / С.В. Кулина // Фізико-технологічні проблеми передавання, оброблення та зберігання інформації в інфокомунікаційних системах : матеріали VII Міжнародної науково-практичної конференції. – Чернівці : «Місто», 2018. – С. 126–127.

4. Системи залишкових класів [Електронний ресурс]. – Режим доступу: https://studfile.net/preview/5082748/page:4/

УДК 004.056.52

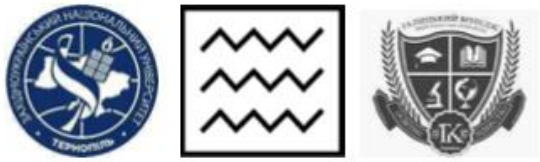
ШУМКА М.І.[1], ГОЛЕМБЙОВСЬКИЙ М.П.[1], ЧЕРНЯК В.А.[2]

[1]Західноукраїнський національний університет

[2]Рівненський фаховий коледж Національного університету біоресурсів і природокористування України

МЕТОД ПОБУДОВИ МОДЕЛІ ПОРУШНИКА ІНФОРМАЦІЙНОЇ БЕЗПЕКИ

Вступ. Як показав світовий і вітчизняний досвід, атаки є найбільш небезпечними загрозами інформаційній безпеці [1]. Можливості проведення атак обумовлені можливостями порушника. Іншими словами, конкретні можливості порушника визначають конкретні атаки, які може провести порушник. Але тоді з урахуванням визначення поняття "модель порушника" всі можливі атаки визначаються моделлю порушника. Модель порушника тісно пов'язана з моделлю загроз. У моделі загроз міститься максимально повний опис загроз безпеці



ЗАХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ВОДНОГО ГОСПОДАРСТВА ТА ПРИРОДОКОРИСТУВАННЯ
ГАЛИЦЬКИЙ ФАХОВИЙ КОЛЕДЖ ІМ. В. ЧОРНОВОЛА

КІБЕРБЕЗПЕКА ТА КОМП'ЮТЕРНО-ІНТЕГРОВАНІ ТЕХНОЛОГІЇ (КБКІТ – 2023)

науково-практична конференція
молодих вчених, аспірантів та студентів

29–31 серпня 2023
Тернопіль

СЕКЦІЯ 2
БЕЗПЕКА ІНТЕРНЕТ РЕЧЕЙ

УДК 004.056.5

Шестерина С. В.[1]

[1]*Західноукраїнський національний університет*

АНАЛІЗ ХМАРНИХ СЕРВІСІВ

Вступ. Розгортання інформаційних технологій у сучасному світі викликає суттєві трансформації у способах зберігання та обробки даних. Цей непередбачуваний розвиток став приводом для активного впровадження хмарних сервісів, які допомагають оптимізувати ресурси та полегшують доступ до обчислень через Інтернет [1].

Сучасна ділова стратегія все частіше включає в себе використання хмарних сервісів як ключового компонента, що сприяє оптимізації витрат на IT-інфраструктуру та розширенню можливостей підприємств для забезпечення якісних послуг своїм клієнтам. Проте, разом із швидким поширенням цих технологій, виникають нові виклики, пов'язані з конфіденційністю інформації, стійкістю до кіберзагроз та етичними аспектами використання хмарних ресурсів.

Мета: Дослідження хмарних сервісів, включаючи їх технічну архітектуру, рівень безпеки, аналіз призначених для них алгоритмів та вплив на ефективність бізнес-процесів.

1. Аналіз типів хмарних сервісів доступних на світовому ринку

Хмарні сервіси - це послуги, які надаються через Інтернет. Вони можуть використовуватися для зберігання, обробки та аналізу даних. Хмарні сервіси пропонують ряд переваг, зокрема масштабованість, доступність та економічність.

На рисунку 1 ключові моделі хмарних сервісів представлені у вигляді піраміди [2].

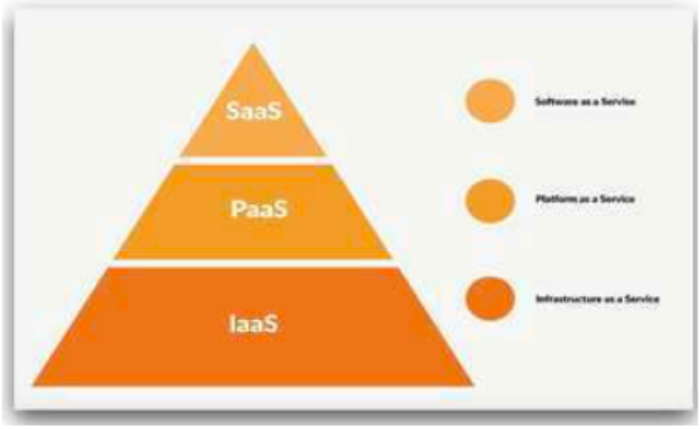


Рисунок 1 – Піраміда моделей хмарних сервісів

Інфраструктура як послуга (IaaS). Постачальник послуги надає в оренду обчислювальні ресурси. За допомогою IaaS користувач може швидко розгортати копії ОС, запускаючи віртуальні копії ряду програмних пакетів. Все необхідне

надається постачальником IaaS.

Платформа як послуга (PaaS). Це один із способів надання клієнту готового програмного середовища. Елементами PaaS є апаратне забезпечення, операційна система, СУБД, проміжне ПЗ, інструменти тестування та розробки. Зараз PaaS розглядається як один із стандартів для електронної комерції.

Програмне забезпечення як послуга (SaaS). Зазвичай означає відсутність необхідності інсталювати пакет програм. При цьому із SaaS може працювати відразу кілька користувачів. Плата зазвичай знімається у вигляді абонентської плати, або ж на основі обсягу операцій. Технічна підтримка лягає на плечі розробника SaaS-платформи. Перевагами SaaS перед стандартною моделлю роботи з ліцензійним ПЗ є відсутність необхідності разової оплати ліцензії. Витрати в цьому випадку можуть бути дуже солідними.

2. Аналіз найбільших хмарних провайдерів

У II кварталі 2023 року глобальні витрати на послуги хмарної інфраструктури зросли на $10 млрд порівняно з другим кварталом 2022 року, в результаті чого загальні витрати склали $64,8 млрд. Якщо поглянути на останні дванадцять місяців, то ринок хмарних технологій - це ринок вартістю $247 млрд доларів, що пояснює, чому за нього точиться така запекла боротьба. Як показано на рисунку 2, на Amazon, Microsoft і Google припадає майже дві третини доходів від хмарної інфраструктури в минулому кварталі, а вісім найбільших постачальників контролюють майже 80 відсотків ринку (рисунок 2) [3].

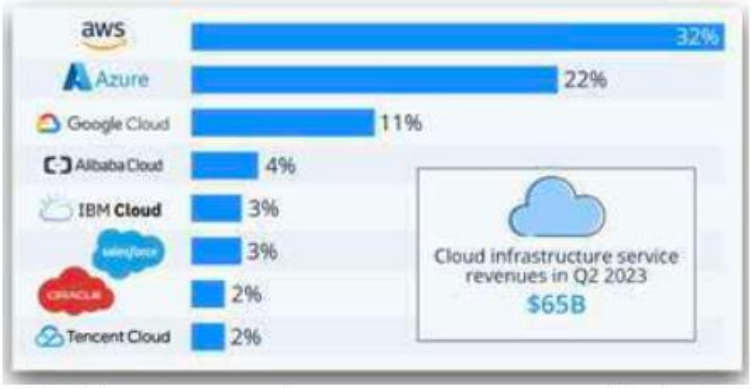


Рисунок 2 – Діаграма часток світового ринку хмарних провайдерів у I кварталі 2023 року

Розглянемо 3 найпопулярніших хмарних провайдерів: що вони пропонують, що роблять краще за інших та що може зіграти не на їхню користь.

Amazon Web Services. Те, що це найпопулярніший хмарний провайдер підтверджує не лише частка ринку, а і клієнти. До прикладу, сервісами AWS користується понад 7500 державних служб США. Amazon Web Services пропонують понад 200 послуг та покривають 31 регіон, кожен з яких із кількома зонами доступності. До переваг даного провайдера можна віднести [2]:

- Широкий вибір послуг та інструментів. Компанії управляти всіма сервісами в межах однієї платформи.
- Безпека. Сервіси захисту, шифрування та сертифікати відповідності.

- Глобальна інфраструктура. AWS має дата центри у різних регіонах.
- Гнучкість і доступність. Можна вибрати бажану ОС, мову програмування, платформу вебзастосунків та інше. Користувач платить лише за використані ресурси.

Навіть у такого популярного провайдера присутній ряд недоліків, такі як:

- Складність. Щоб працювати з AWS, необхідно мати відповідні навички.
- Ціноутворення. AWS, ціноутворення може бути складним через велику кількість послуг, ціни на які формуються за різним принципом.
- Обмеження Amazon EC2. Регіон впливає на кількість доступних ресурсів.

Microsoft Azure. Хмарні сервіси від Microsoft. Багато бізнесів працюють з продуктами компанії, тож обирають цього провайдера для своїх потреб. Azure теж пропонує понад 200 хмарних сервісів, які дозволяють створювати, запускати та керувати програмами в різних хмарах. Перевагами даного провайдера є :

- Послуги на різний смак. Обчислення, сховище, БД, мережу, аналітику, AI, IoT тощо.
- Можливості гібридної хмари. Azure фокусується на гібридних хмарних рішеннях.
- Інтеграція з екосистемою Microsoft. Добре інтегруються з її іншими продуктами та сервісами.
- Глобальний дата-центр. Має мережу дата-центрів по всьому світу.
- Безпека та відповідність. Шифрування даних, контроль доступу, розвідку загроз безпеки та сертифікати відповідності.

Недоліки теж присутні:

- Відсутність доступу в разі збою. Якщо в основній системі станеться збій, користувачі не матимуть офлайн-доступу до неї.
- Незрозумілий інтерфейс користувача. Ускладнине користування платформою для тих, хто немає серйозного технічного досвіду.
- Azure може бути дорогим, але в цьому випадку є вихід. Компанія пропонує інструменти для керування білінгом.

Google Cloud Platform. Хмарні сервіси від Google. Вони працюють на тій самій інфраструктурі, яку компанія використовує для своїх продуктів. GCP надає понад 200 послуг. До переваг провайдера відносяться:

- Ціна. Один із головних переваг GCP.
- Швидкість мережі, яку пропонує Google своїм клієнтам, складає до 10 Тбіт. Мережа компанії працює по всьому світу, тож має низьку затримку.
- Робота з великими даними. Google має інструменти по типу BigQuery, а також Cloud Dataflow для обробки у реальному часі.
- Масштабованість. Google надає можливість зменшувати або збільшувати ресурси залежно від потреб.
- Автоматизація. GCP має інструменти для автоматизації, які дозволяють автоматизувати процеси збирання, тестування та розгортання.
- Безпека. GCP захищає програми, інфраструктуру та дані.

Навіть при такій кількості переваг присутні і деякі недоліки:

- Вибір послуг. На фоні AWS та Azure його пропозиція не така широка.
- Вартість має також і негативну сторону. Передбачити витрати складно через різні моделі ціноутворення та потенційні додаткові витрати.
- Проблеми з конфіденційністю. Google зберігає дані користувачів на своїх серверах, має доступ до них та може використовувати ці дані у своїх цілях.

Висновок. У проведеному дослідженні важливо підкреслити кілька ключових висновків, які виникають із проведеного аналізу.

Перш за все, слід відзначити, що хмарні сервіси виявляються ефективним засобом оптимізації ресурсів для підприємств, дозволяючи їм зменшити витрати на IT-інфраструктуру та отримувати доступ до необхідних обчислень та ресурсів.

Однак із зростанням популярності хмарних сервісів виникають нові виклики, зокрема у сфері безпеки та конфіденційності даних. Сприйняття цих сервісів як надійного засобу збереження інформації вимагає постійного вдосконалення заходів безпеки для захисту від потенційних кіберзагроз.

Не менш важливою є увага до етичних аспектів використання хмарних сервісів. Забезпечення відповідального використання технологій та дотримання стандартів конфіденційності стає необхідністю в умовах зростаючої кількості оброблюваних даних.

Нарешті, дослідження вказує на необхідність подальших наукових досліджень у галузі хмарних сервісів. Розвиток більш продуктивних алгоритмів, посилення заходів безпеки та вивчення впливу хмарних технологій на соціально-економічний розвиток можуть стати ключовими напрямками майбутніх досліджень в цій важливій галузі інформаційних технологій. У цілому, хмарні сервіси сьогодні вже є необхідним інструментом для багатьох сучасних підприємств, проте їхнє використання потребує уважного підходу та постійного вдосконалення для максимізації переваг і зниження можливих ризиків.

Перелік використаних джерел.

1. Amazon Maintains Lead in the Cloud Market. [Електронний ресурс]. - Режим доступу:https://www.statista.com/chart/18819/worldwide-market-share-of-leading-cloud-infrastructure-service-providers/

2. Топ 5 хмарних провайдерів: плюси та мінуси. [Електронний ресурс]. - Режим доступу: https://blog.iteducenter.ua/ru/sysadministration/top-5-cloud-providers-advantages-and-disadvantages/

3. Хмарні сервіси [Електронний ресурс]. - Режим доступу: https://www.pharmencyclopedia.com.ua/article/7857/xmarni-servisi