Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерної інженерії та управління
(повна назва)
Кафедра Безпеки інформаційних технологій
(повна назва)

**АТЕСТАЦІЙНА РОБОТА**
**Пояснювальна записка**

рівень вищої освіти другий (магістерський)

Аналіз безпеки та ефективності зберігання даних за допомогою хмарних сервісів
(тема)

Виконав:
студент 2 курсу, групи БІКСм-19-1
Ахтирцев І.І.
(прізвище, ініціали)
Спеціальність 125 Кібербезпека
(код і повна назва спеціальності)
Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)
Освітня програма «Безпека інформаційних і комунікаційних систем»
(повна назва освітньої програми)
Керівник Федюшин О. І.
(прізвище, ініціали)

Допускається до захисту
Зав. кафедри _____ Халімов Г.З.
(підпис) (прізвище, ініціали)

Харків 2020
Харківський національний університет радіоелектроніки

Факультет_____Комп'ютерної інженерії та управління_____

Кафедра_____Безпеки інформаційних технологій _____

Рівень вищої освіти_____другий (магістерський)_____

Спеціальність _____125 Кібербезпека_____

Тип програми _____освітньо-професійна_____
(освітньо-професійна, або освітньо-наукова)

Освітня програма_____«Безпека інформаційних і комунікаційних систем»___
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
«____»_____ 20 ___ р.

ЗАВДАННЯ
НА АТЕСТАЦІЙНУ РОБОТУ

студентові Ахтирцеву Іллі Івановичу_____
(прізвище, ім'я, по батькові)

1. Тема роботи: Аналіз безпеки та ефективності зберігання даних за допомогою хмарних сервісів.
затверджена наказом по університету від "22" жовтня 2020 р. № 1412Ст
2. Термін подання студентом роботи 14.12.2020
3. Вихідні дані до роботи літературні джерела та статті з технологій захисту баз даних; методи криптографічного захисту інформації на прикладному рівні;
_____
_____
_____
_____
_____
4. Зміст пояснювальної записки (перелік питань, що потрібно розробити)
- загальні відомості про основні проблеми захисту баз даних;
- загальні відомості про існуючі засоби забезпечення конфіденційності даних;
- аналіз можливих рішень для даних в хмарних сховищах;
- знаходження оптимального рішення для захисту баз даних;
- проведення аналізу та оцінка ефективності існучих рішень;
5. Перелік графічного матеріалу (з точним зазначенням обов'язкових кресл плакатів)
Презентаційні матеріали
_____
_____

6. Консультанти розділів роботи

| Найменування розділу | Консультант (посада, прізвище, ім'я, по батькові) | Позначка консультанта про виконання розділу | |
|---|---|---|---|
| | | Підпис | дата |
| Огляд основних проблем при захисті даних | доц. Федюшин О. І. | | 25.10.2020 |
| Огляд методів організації захисту даних в хмарних сховищах | доц. Федюшин О. І. | | 01.11.2020 |
| Огляд власного методу хмарних обчислень | доц. Федюшин О. І. | | 10.11.2020 |
| Оцінка ефективності методу | доц. Федюшин О. І. | | 01.12.2020 |

КАЛЕНДАРНИЙ ПЛАН

| № | Назва етапів роботи (проекту) | Термін виконання етапів проекту (роботи) | Примітка |
|---|---|---|---|
| 1 | Отримання завдання | 04.09.20 | виконано |
| 2 | Робота з джерелами за тематикою роботи | 04.09.20 - 01.10.20 | виконано |
| 3 | Огляд основних проблем при захисті даних | 02.10.20 - 15.10.20 | виконано |
| 4 | Огляд сучасних хмарних технологій забезпечення конфіденційності баз даних | 16.10.20 - 01.11.20 | виконано |
| 5 | Алгоритм Віктора Телло | 02.11.20 - 15.11.20 | виконано |
| 7 | Оцінка ефективності алгоритму Віктора Телло | 15.11.20 - 25.11.20 | виконано |
| 8 | Оформлення пояснювальної записки | 26.11.20 - 01.12.20 | виконано |
| 9 | Представлення роботи на здачу | 10.12.2020 | виконано |

Дата видачі завдання 4 вересня 2020 року

Студент _____
(підпис)

Керівник роботи (проекту)___ __ доц. Федюшин О. І.__
(підпис) (посада, прізвище, ініціали)

# РЕФЕРАТ

Робота включає 76 сторінок, 10 рисунків, 9 таблиць, 13 посилань.

Мета даної кваліфікаційної роботи – отримання набору технологій для гарантування безпеки в одній із послуг, які включає концепція хмарних обчислень, а саме хмарних базах даних.

Об’єкт дослідження – системи захисту інформації в хмарних базах даних.

Предметом дослідження є забезпечення належного рівня захисту інформації в хмарних базах даних.

Методи дослідження: опрацювання літератури та інших інформаційних джерел за даною темою, аналіз існуючих методів та засобів захисту інформації в хмарних базах даних та їхніх характеристик.

Результати роботи можуть бути використані для побудови системи захисту інформації, застосовної до хмарних баз даних.

Ключові слова:

інформаційна безпека, хмарні обчислення, загрози безпеки інформації, база даних, методи розбиття

# ABSTRACT

The work includes 76 pages, 10 figures, 9 tables and 13 literary references. The purpose of this qualification work is to obtain a set of technologies for guaranteeing security in one of the services that includes the concept of cloud computing, namely cloud databases.

The object of research is the security of cloud computing information.

The subject of research is to provide an adequate level of protection of information in cloud databases.

Research Methods: reviewing literature and other information sources on the topic, analyzing existing methods and tools for protecting information in cloud databases and their characteristics.

The results of the work can be used to build an information security system applicable to cloud databases.

Keywords:

information security, cloud computing, security information threats, database, partitioning method

# ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ЦОД - Центр обробки даних

ПЗ - Програмне забезпечення

IaaS - Infrastructure as a Service

PaaS - Platform as a Service

SaaS - Software as a Service

DBaaS - Database as a Service

EC2 - Elastic Compute Cloud

API - Application Programming Interface

SDL - Security Development Lifecycle

MAC - Mandatory access control

DAC - Discretionary access control

RBAC - Role based bccess control,

ABAC - Attribute based access control

ДТС - Довірена третя сторона

# ВСТУП

Нині навколо нас розвивається новий інформаційний світ - світ, який дозволяє нам переміщатися, взаємодіяти і вести бізнес набагато ефективніше, ніж будь-коли раніше. В цьому світі, в якому до Інтернету може підключитися практично будь-яка людина і безліч пристроїв, ми спостерігаємо вибухове зростання кількості технічних і інформаційних ресурсів, а також підключених до мережі об'єктів, що має значний вплив на наше буденне життя. Поряд з усіма супутніми зручностями цей «Інформаційний вибух» приносить з собою нові проблеми:

- організації стикаються з усе більш швидкими змінами бізнесу, глобальним конкурентним тиском і підвищенням вимог до соціальної відповідальності;
- надання IT-ресурсів стало більш складною справою внаслідок розвитку технологій, які підвищили обізнаність кінцевих користувачів і прискорили зростання обсягів інформації;
- користувачі, які отримують ті чи інші послуги за допомогою інтернет-технологій, розраховують на наявність безпечної та постійно діючої комп'ютерної інфраструктури, яка надавала б ці послуги так само легко і надійно, як побутова електрична мережа надає енергію. Це обумовлює необхідність фундаментального зміни характеру надання послуг.

Актуальність роботи: активний розвиток ринку хмарних послуг сприяє розгляду питань пов’язаних з безпекою інформації в даній сфері. Попередні роботи це саме роботи, в яких описуються нові технології, чи роботи, в яких розглядається більш конкретно якийсь один аспект безпеки. В даній атестаційній роботі розглядаються одразу декілька аспектів безпеки інформації та пропонується новий метод розбиття даних.

Метою роботи є отримання набору технологій для гарантування безпеки реалізації та функціонування однієї з послуг, що надає концепція хмарних обчислень, а саме використання хмарних баз даних для зберігання інформації.

Для цього були поставлені наступні завдання : аналіз основних властивостей, моделей розгортання та обслуговування хмарних технологій; визначення загроз, вразливостей хмарних обчислень та існуючих фреймворків для моделювання загроз; вибір відповідних технологій для захисту інформації в хмарних базах даних та розробка методу розбиття даних.

Об`єктом дослідження є системи захисту інформації в хмарних базах даних.

Предметом дослідження є забезпечення  належного рівня захисту в хмарних базах даних

Методологія дослідження: опрацювання літератури та інших інформаційних джерел за даною темою, аналіз існуючих методів та засобів захисту інформації в хмарних базах даних та їхніх характеристик.

Наукова новизна даної роботи полягає в отриманні нового набору технологій для гарантування безпеки в хмарних базах даних, та розробці методу розбиття даних.

Практичне значення отриманих результатів : результати роботи можуть бути використані для побудови системи захисту інформації, застосовної до хмарних баз даних.

# 1 ХМАРНІ СЕРВІСИ

## 1.1 Загальний огляд

Під хмарними обчисленнями (від англ. Cloud computing, також використовується термін «хмарна (розсіяна) обробка даних») зазвичай мається на увазі надання користувачу комп'ютерних ресурсів і потужностей у вигляді інтернет-сервісу. Таким чином, обчислювальні ресурси надаються користувачеві в «чистому» вигляді, і користувач може не знати, які комп'ютери обробляють його запити, під керуванням якої операційної системи це відбувається і т.д. Під хмарними технологіями будемо розуміти модель системи процедур, що забезпечують направлену зміну матеріальних і віртуальних об'єктів.

Часто хмари порівнюють з мейнфреймами (mainframe), знаходячи між ними багато спільного. Принципова відмінність хмари від мейнфреймів в тому, що його обчислювальна потужність теоретично безмежна. Друга принципова відмінність в тому, що, простіше кажучи, термінали для мейнфреймів служили тільки для інтерактивної взаємодії користувача з запущеної на обробку завданням. В хмарі ж термінал сам є потужним обчислювальним пристроєм, здатним не тільки накопичувати проміжну інформацію, але і безпосередньо керувати глобальною системою обчислювальних ресурсів.

Серед раніше відомих (в 1990-х рр.) технологій обробки даних деяке поширення набули так звані grid-обчислення. Цей напрямок спочатку розглядався як можливість використання вільних ресурсів процесорів і розвитку системи добровільної оренди обчислювальних потужностей. Ряд проектів (GIMPS, distributed.net, SETI@home) довели, що така модель обчислень досить ефективна. Ця технологія застосовується для вирішення наукових, математичних задач, де потрібні значні обчислювальні ресурси.

Відомо, що grid-обчислення також застосовуються для комерційних цілей. Наприклад, з їх допомогою виконуються деякі трудомісткі завдання, пов'язані з економічним прогнозуванням, аналізом сейсмічних даних, розробкою і дослідженням властивостей вакцин і нових ліків. Дійсно, grid-обчислення і хмари мають багато схожих рис в архітектурі і застосовуваних принципах. Проте, модель хмарних обчислень вважається сьогодні більш перспективною, завдяки значно більш гнучкої платформі для роботи з віддаленими обчислювальними ресурсами.

Нині великі обчислювальні хмари складаються з тисяч серверів, розміщених в центрах обробки даних (ЦОД). Вони забезпечують ресурсами десятки тисяч додатків, яки користуються мільйони користувачів [1]. Хмарні технології є зручним інструментом для підприємств, яким занадто дорого утримувати власні ERP, CRM або інші сервери, що вимагають придбання і налаштування додаткового обладнання.

Серед приватних користувачів широке поширення поступово отримують завдяки своїй зручності хмарні послуги, які, наприклад, надаються компанією Google («Документи», «Календар» і ін.).

Причини зростаючої популярності хмарних технологій зрозумілі: можливості їх застосування дуже різноманітні і дозволяють економити як на обслуговуванні і персоналі, так і на інфраструктурі. Апаратне забезпечення може бути сильно спрощено при обробці даних і зберіганні інформації у віддалених центрах даних. Всі ці проблеми майже повністю перекладаються на провайдера послуг. Крім цього такий підхід дозволяє стандартизувати програмне забезпечення (ПЗ), навіть якщо на комп'ютерах підприємства встановлені різні операційні системи (Windows, Linux, MacOS і т.п.). Хмарні технології полегшують забезпечення доступу до даних компанії як для клієнтів, так і для власних співробітників, що перебувають поза офісу, але мають можливість підключитися через Інтернет.

Зрозуміло, що використання хмарних обчислень набагато зручніше. Найголовнішим недоліком, який можна відразу помітити, є повна залежність

від постачальника цих послуг. фактично підприємство (користувач) виявляється заручником провайдера сервісів і провайдера доступу в мережу Інтернет. Хоча надійність постачальників хмарних обчислень зростає, для забезпечення надійності і безпеки даних необхідно докласти чимало зусиль, наприклад, мати дублюючі канали зв'язку, що забезпечують потужності для можливості перемикання на них і, звичайно ж, подумати про доступність інформації і безпеки. Крім цього, хмарні обчислення абсолютно не підходять для підприємств, що мають відношення до державної та військової таємниці. Жодна комісія не видасть сертифікат на таку систему при роботі з інформацією, яка не підлягає розголошенню.

## 1.2 Моделі розгортання хмарних технологій

За моделлю розгортання хмари поділяють на приватні, загальнодоступні (публічні) та гібридні.

Приватні хмари- це внутрішні хмарні інфраструктура і служби підприємства. Ці хмари знаходяться в межах корпоративної мережі. Організація може керувати приватною хмарою самостійно або доручити це завдання зовнішньому підряднику. Інфраструктура може розміщуватися або в приміщеннях замовника, або у зовнішнього оператора, або частково у замовника частково у оператора. Ідеальний варіант приватної хмари - хмара, розгорнута на території організації, що обслуговує і контролюється її співробітниками. Приватні хмари володіють тими ж привілеями, що і загальнодоступні, але з однією важливою особливістю: підприємство саме займається установкою і підтримкою хмари. Складність і вартість створення внутрішньої хмари можуть бути дуже високі, а витрати на її експлуатацію можуть перевищувати вартість використання загальнодоступних хмар.

Слід зазначити, що у приватних хмар є переваги перед загальнодоступними: більш детальний контроль над різними ресурсами хмари забезпечить компанії будь-які доступні варіанти конфігурації. Крім того,

приватні хмари ідеальні, коли потрібно виконувати роботи, які не можна довірити загальнодоступній хмарі з міркувань безпеки.

Загальнодоступні (публічні) хмари - це хмарні послуги, що надаються постачальником. Вони знаходяться за межами корпоративної мережі. Користувачі даних хмар не мають можливості управляти даними хмари або обслуговувати її, вся відповідальність покладена на власника цієї хмари. Постачальник хмарних послуг приймає на себе обов'язки по установці, керуванню, реалізує надання та обслуговування програмного забезпечення, інфраструктури застосунків або фізичної інфраструктури. Клієнти платять тільки за ресурси, які вони використовують. Абонентом пропонованих сервісів може стати будь-яка компанія або індивідуальний користувач. Вони пропонують легкий і доступний за ціною спосіб розгортання веб-сайтів або бізнес-систем з великими можливостями масштабування, які в інших рішеннях були б недоступні. Приклади: онлайн-сервіси Amazon EC2 і Amazon Simple Storage Service (S3), Google Apps / Docs, Salesforce.com, Microsoft Office Web. Разом з тим послуги публічних хмар в основному надаються в вигляді стандартних конфігурацій, тобто виходячи з умов найбільш поширених випадків використання. Це означає, що у користувача залишається менше можливостей по вибору конфігурації в порівнянні з системами, в яких ресурсами керує сам споживач. Слід також мати на увазі, що, оскільки споживачі слабо контролюють інфраструктуру, процеси, що вимагають суворих заходів безпеки та відповідності нормативним вимогам, не завжди підходять для реалізації в загальнодоступному хмарі.

Гібридні хмари представляють собою поєднання загальнодоступних і приватних хмар. Зазвичай вони створюються підприємством, а обов'язки з управління ними розподіляються між підприємством і постачальником загальнодоступного хмари. Гібридна хмара надає послуги, частина яких відноситься до загальнодоступних, а частина - до приватних. Зазвичай такий тип хмар використовується, коли організація має сезонні періоди активності. Іншими словами, як тільки внутрішня IT-інфраструктура не може виконати

поточні завдання, частина потужностей перекидається на публічну хмару (наприклад, великі обсяги статистичної інформації, які в необробленому вигляді не мають цінності для підприємства), а також для надання доступу користувачам до ресурсів підприємства (до приватної хмари) через публічну хмару. Добре продумана гібридна хмара може обслуговувати, як вимагають правила безпеки, як критично важливі процеси, такі як отримання платежів від клієнтів, або більш другорядні. Основним недоліком цього типу хмари є складність ефективного створення подібних рішень і управління ними. Необхідно отримувати послуги з різних джерел і організувати їх так, якщо б це було єдине джерело. Взаємодія між приватним і загальнодоступним компонентами може ще більше ускладнити рішення. Оскільки це відносно нова архітектурна концепція в сфері хмарних обчислень, для цієї моделі з'являються все нові і нові практичні рекомендації та інструменти, і її широке поширення може затягнутися до тих пір, поки вона не буде краще вивчена. На думку Тома Біттман, віце-президента і провідного аналітика американської дослідницької і консалтингової компанії "Gartner", серед перерахованих вище трьох моделей розгортання хмар найбільш актуальною для бізнесу в даний момент є приватні хмари. Біттман виділив п'ять основних моментів, які допомагають отримати більш точне уявлення про будову приватної хмари.

Залежно від виду хмарного сервісу, ним можуть володіти і розпоряджатися як провайдер, так і користувач, або і той і інший. Також можуть відрізнятися права доступу до ресурсів (див. таблицю 1.1).

Таблиця 1.1 - Обслуговування та управління різними видами хмарних ресурсів [2]

| Вид хмари | Ким обслуговується інфраструктура | Хто є власником інфраструктури | Де знаходиться інфраструктура | У кого є доступ |
|---|---|---|---|---|
| Публічне | Зовнішнім провайдером | Зовнішній провайдер | У зовнішнього провайдера | У будь-якого користувача |
| Приватне/ Суспільне | Користувачем або зовнішнім провайдером | Користувач або зовнішній провайдер | У зовнішнього провайдера або у користувача | У авторизованого користувача |
| Гібридне | Користувачем і зовнішнім провайдером | Користувач і зовнішній провайдер | У зовнішнього провайдера і у користувача | У авторизованих і у будь-яких зовнішніх користувачів |

Хмара - це не тільки віртуалізація. Хоча віртуалізація серверів та інфраструктури становить важливий фундамент приватних хмарних обчислень, самі по собі віртуалізація і управління віртуалізованим середовищем ще не є приватною хмарою. Віртуалізація дозволяє краще структурувати, об'єднувати в пул і динамічно надавати ресурси інфраструктури: сервери, десктопи, ємності для зберігання, мережеве обладнання, сполучне ПЗ і т.д. Але, щоб середовище технічно могло вважатися хмарним, потрібні ще й інші складові, такі як віртуальні машини, операційні системи або контейнери сполучного ПЗ, високостійкі операційні системи, ПЗ grid-обчислень, ПЗ для абстрагування ресурсів зберігання, засоби масштабування і кластеризації.

Термін «приватна хмара» на відміну від загальнодоступної або гібридної відноситься до ресурсів, що використовуються єдиною організацією, або означає, що хмарні ресурси організації повністю ізольовані в хмарі від інших. Хмара - необов'язково джерело економії. Одна з головних помилок полягає в тому, що хмара буде економити гроші. Економія можлива, але не є обов'язковим атрибутом. Приватна хмара дозволяє більш ефективно перерозподіляти ресурси, щоб задовольнити корпоративні вимоги, і здатна зменшити капітальні витрати на обладнання. Але приватна хмара вимагає інвестицій в автоматизацію, і одна лише економія може не окупати всієї вартості. Так що, зниження витрат не є головною перевагою цієї моделі. З цієї точки зору, головним стимулом до впровадження хмарної моделі повинна бути не економія, а швидкість виходу на ринок, можливість швидкої адаптації і динамічного масштабування відповідно до попиту, які дозволяють підвищити швидкість впровадження нових сервісів. Приватна хмара не завжди впроваджена у замовника. Приватна хмара означає конфіденційність, а не конкретне місце розташування, володіння ресурсами або самостійне управління. Багато постачальників пропонують нелокальні приватні хмари, тобто виділяють ресурси єдиному замовнику, виключаючи спільне використання одного пулу декількома клієнтами. «Хмара іменується приватною за її приватність, а не за те, де вона розгорнута, хто нею володіє і несе відповідальність за управління », - підкреслює Біттман. Деякі, наприклад, можуть свої ЦОД розміщувати у хостинг-провайдерів або об'єднувати в пул ресурси різних замовників, але ізолювати їх один від одного за допомогою віртуальної приватної мережі (Virtual Private Network - VPN) та інших подібних технологій. Приватна хмара (як і публічна хмара) - це не тільки інфраструктурні сервіси. Серверна віртуалізація - велика тенденція і тому потужний двигун приватних хмарних обчислень. Але приватна хмара не зводиться тільки до інфраструктури як послуги (IaaS). Наприклад, для розробки і тестування нового ПЗ PaaS має більше сенсу, ніж просто надання віртуальних машин.

Сьогодні самий швидко зростаючий сегмент хмарних обчислень - це IaaS. Вона надає самі низькорівневі ресурси ЦОД в простій для використання формі, але не змінює фундаментально принципи роботи. Щоб створити нові застосунки, спочатку призначені для хмари і надають абсолютно нові послуги, які можуть дуже відрізнятися від того, що давали колишні застосунки, розробникам зручніше використовувати PaaS. Приватна хмара може перестати бути приватною. З одного боку, приватна хмара надає переваги хмари: швидкість перебудови, масштабованість і ефективність, позбавляє від деяких загроз безпеки, потенційних і реальних, які характерні для загальнодоступних хмар. З іншого боку, з часом рівень обслуговування, безпека і контроль дотримання вимог в загальнодоступних хмарних сервісах безумовно будуть підвищуватися. Тому деякі приватні хмари, можливо, цілком перейдуть в категорію загальнодоступних. Більшість же сервісів приватної хмари, швидше за все, будуть еволюціонувати в гібридні хмарні сервіси, розширюючи доступні можливості за рахунок використання загальнодоступних хмарних послуг і інших сторонніх ресурсів.

## 1.3 Основні властивості хмарних технологій

Національний Інститут стандартів і технологій NIST (National Institute of Standards and Technology, USA) в своєму документі "The NIST Definition of Cloud Computing " дає визначення таким характеристикам хмар:

По-перше, "широкосмуговий доступ до мережі". Доступ до ресурсів в хмарі можна отримати за допомогою декількох типів пристроїв. Сюди входять не тільки найбільш поширені пристрої (ноутбуки, робочі станції і т.д.), але і мобільні телефони, тонкі клієнти і так далі. Контраст " широкосмугового доступу до мережі" з доступом до обчислювальних і мережевих ресурсів в епоху мейнфреймів. Розрахункові ресурси сорок років тому були мізерними і дорогими. Їх використання обмежене через пріоритетності та важливості

робочого навантаження з метою збереження цих ресурсів. Аналогічним чином, ресурси мережі також обмежені. Мережі на основі Інтернет-протоколу (IP) були широко поширені в той час, тому не існувало мереж з високою пропускною здатністю і низькою затримкою. Згодом витрати, пов'язані з мережею (наприклад, витрати, пов'язані з обчисленнями і зберіганням даних), скоротилися в зв'язку з масштабністю виробництва і комерціалізацією відповідних технологій. Зі збільшенням пропускної спроможності мережі відповідно збільшився доступ до неї і її масштабованість. "Широкосмуговий доступ до мережі" можна розглядати як характеристику хмарних обчислень, так і як фактор, що сприяє їх розвитку.

"Самообслуговування на вимогу" є ключовою - дехто каже, основною - характеристикою хмарних обчислень. Якщо розглядати IT як складний ланцюжок поставок з застосунком і кінцевим користувачем в кінці ланцюжка, здатність до самозабезпечення ресурсами в типових IT-середовищах фундаментально порушує робочий процес і процеси, які розвивалися як функція корпоративних IT за останні кілька десятиліть. Це включає в себе робочі процеси, пов'язані із закупівлею систем зберігання, серверів, мережевих вузлів, ліцензій на програмне забезпечення і так далі. Самозабезпечення в не хмарних або старих архітектурах змушує традиційні процеси і функції, такі як планування пропускної здатності, управління мережею (забезпечення якості обслуговування або QOS) і безпеку (створення списків контролю доступу або ACL), зупинятися або навіть повністю руйнуватися. Добре відомий ефект в управлінні ланцюжком поставок - коли неповна або неточна інформація призводить до високої мінливості виробничих витрат - відноситься не тільки до виробничому середовищі, а й безпосередньо до виділення IT ресурсів в не хмарному середовищі. Хмарні архітектури, проте, проектуються і створюються з урахуванням необхідності самозабезпечення. Ця передумова має на увазі використання досить складних програмних рамок або порталів для управління функціями ініціалізації і допоміжного офісу. Історично склалося так, що відсутність комерційного готового програмного забезпечення (COTS),

спеціально розробленого для автоматизації хмарних обчислень, змусило багато компаній створювати власні портали і структури для підтримки цих функцій. Пакети програмного забезпечення COTS, призначені для управління і автоматизації корпоративних робочих навантажень, в даний час складають значну частину загального потенційного ринку хмарних обчислень.

"Об'єднання ресурсів в пули" є фундаментальною передумовою масштабованості в хмарі. Без об'єднаних обчислень мереж і сховищ постачальник послуг повинен надавати послуги через кілька ізольованих або дискретних, незалежних ресурсів з невеликою кількістю з'єднань або без них. Основою загальнодоступних хмарних інфраструктур є розраховані на багато користувачів середовища, в яких кілька клієнтів спільно використовують суміжні ресурси в хмарі зі своїми колегами. При багатокористувацької оренді спостерігається неминуче збільшення експлуатаційних витрат, які можуть бути зменшені за рахунок певних конфігурацій обладнання та програмних рішень, таких як профілі застосунків і серверів.

"Вимірюваний сервіс" означає, що використання цих "об'єднаних ресурсів" контролюється і повідомляється споживачу, забезпечуючи прозорість норм споживання і витрат. Моніторинг використання для цілей повернення платежів (або просто для міжвідділової звітності та складання бюджету) вже давно є вимогою для зацікавлених сторін IT - ще одним священним Граалем, але побудова такої системи зазвичай не є основною компетенцією більшості IT-відділів.

Останньою характеристикою, виділеної у визначенні хмарних обчислень NIST, є "швидка еластичність". Еластичні обчислення мають вирішальне значення для скорочення витрат і часу виходу на ринок (TTM). Дійсно, поняття гнучких ресурсів в ланцюжку поставок IT настільки бажано, що Amazon назвала свою хмарну платформу "Elastic Compute Cloud" ( "EC2"). Що стосується витрат на еквівалент повної зайнятості операційні витрати, пов'язані з виділенням ресурсів (переміщення, додавання, зміна або MAC) в ланцюжку

поставок IT зазвичай складають найбільш значну частину витрат, пов'язаних з розгортанням застосунків.

## 1.4 Моделі обслуговування хмарних технологій

В даний час прийнято виділяти три основні моделі обслуговування хмарних технологій, які іноді називають шарами хмари. Можна сказати, що ці три шари - послуги інфраструктури, послуги платформи і послуги застосунків - відображають будову не тільки хмарних технологій, а й інформаційних технологій в цілому. Зупинимося докладніше на кожній з них.

До послуг інфраструктури (Infrastructure as a Service - IaaS) можна віднести набір фізичних ресурсів, таких як сервери, мережеве обладнання та накопичувачі, пропоновані замовникам як послуги, що надаються. Послуги інфраструктури вирішують задачу належного оснащення ЦОД, надаючи обчислювальні потужності в міру необхідності. Зазвичай ці послуги підтримують інфраструктуру і набагато більше число споживачів в порівнянні з послугами застосунків. Приватним прикладом послуг інфраструктури є апаратне забезпечення як послуга (Hardware as a Service - HaaS). Як послугу користувач отримує обладнання, на основі якого розгортає свою власну інфраструктуру з використанням найбільш підходящого ПЗ.

Споживач при цьому не керує базовою інфраструктурою хмари, але має контроль над операційними системами, системами зберігання, розгорнутими застосунками і, можливо, обмежений контроль вибору мережевих компонентів (наприклад, хост з мережевими екранами). В такому випадку захист платформ і застосунків забезпечує сам споживач, а провайдер хмари повинен організувати захист інфраструктури. Для надання ресурсів на вимогу часто використовується віртуалізація. Переваги. Зниження капіталовкладень в апаратне забезпечення. Оскільки в цій моделі зазвичай використовуються методи віртуалізації, можна домогтися економії в результаті більш ефективного використання ресурсів. Зменшення ризику втрати інвестицій і порога впровадження, можливість

плавного автоматичного масштабування. Недоліки. Бізнес-ефективність і продуктивність дуже залежать від можливостей постачальника. Існує ймовірність, що будуть потрібні потенційно великі довгострокові витрати.

Централізація вимагає нових підходів до заходів безпеки. Прикладами послуг інфраструктури служать IBM SmartCloud Enterprise, VMWare, Amazon EC2, Windows Azure, Google Cloud Storage, Parallels Cloud Server і багато інших.

Послуги платформи (Platform as a Service - PaaS) - це модель обслуговування, в якій споживачеві надаються застосунки (створені або придбані) як набір послуг. Сюди входить, зокрема, проміжне ПЗ як послуга, обмін повідомленнями як послуга, інтеграція як послуга, інформація як послуга, зв'язок як послуга і т.д. Наприклад, робоче місце як послуга (Workplace as a Service - WaaS) дозволяє компанії використовувати хмарні обчислення для організації робочих місць своїх співробітників, налаштувавши і встановивши все необхідне для роботи персоналу ПЗ. Дані як послуга (Data as a Service - DaaS) надають користувачу дисковий простір, яке він може використовувати для зберігання великих обсягів інформації. Безпека як послуга (Security as a Service - SaaS) дає можливість користувачам швидко розгортати продукти, що дозволяють забезпечити безпечне використання веб-технологій, безпеку електронного листування, а також безпеку локальної системи. Цей сервіс дозволяє користувачам заощаджувати на розгортанні і підтримці своєї власної системи безпеки. Іншими словами, модель PaaS - це IaaS разом з операційною системою і її інтерфейсом прикладного програмування (API - Application Programming Interface). Споживач при цьому не керує базовою інфраструктурою хмари, в тому числі мережами, серверами, операційними системами і системами зберігання даних, але має контроль над розгорнутими застосунками і, можливо, деякими параметрами конфігурації середовища хостингу. Таким чином, споживач повинен подбати про забезпечення захисту застосунків, які будуть розгорнуті на наданих платформах[3].

Застосунки можуть працювати як в хмарі, так і в традиційних ЦОД підприємства. Для досягнення масштабованості, необхідної в хмарі, різні

пропоновані послуги часто віртуалізуються, як і розглянуті раніше послуги інфраструктури. Переваги. Плавне розгортання версій. Плавність означає, що в ідеалі користувач повинен слабо відчувати або навіть взагалі не відчувати зміни ПЗ в хмарі. Недоліки. Як і у попередньої моделі обслуговування, централізація вимагає надійних заходів безпеки. Прикладами послуг платформи служать IBM SmartCloud Application Services, Amazon Web Services, Windows Azure, Boomi, Cast Iron, Google App Engine і інші.

Застосунки як послуга (Software as a Service - SaaS) припускають доступ до застосунків як до сервісу, тобто застосунки провайдера запускаються в хмарі і надаються користувачам на вимогу як послуги. Іншими словами, користувач може отримувати доступ до ПЗ, розгорнутого на віддалених серверах, за допомогою Інтернету, причому всі питання оновлення та ліцензій на дане ПЗ регулюються постачальником даної послуги. Оплата в даному випадку здійснюється за фактичне використання ПЗ. Іноді ці послуги постачальники роблять безкоштовними, так як у них є можливість отримувати дохід, наприклад, від реклами. Програма є доступною за допомогою різних клієнтських пристроїв або через інтерфейси тонких клієнтів, такі, наприклад, як веб-браузер, або веб-пошта, або інтерфейси програм. Споживач при цьому не керує базовою інфраструктурою хмари, в тому числі мережами, серверами, операційними системами. Насамкінець ви несете відповідальність тільки за збереження параметрів доступу (логінів, паролів і т.д.) і виконання рекомендацій провайдера щодо безпечних налаштувань застосунків. Послуги застосунків найбільше знайомі повсякденному користувачеві. Найпоширенішим прикладом застосунків даного типу є поштові сервіси GMail, Mail.ru, Yahoo Mail. Взагалі існують тисячі застосунків SaaS, і завдяки технології Web 2.0 їх число зростає з кожним днем. Серед служб застосунків є безліч застосунків, націлених на корпоративне співтовариство. Існує ПЗ, що управляє нарахуванням заробітної плати, кадровими ресурсами, колективною роботою, взаємовідносинами з клієнтами та бізнес-партнерами і т.п.

Переваги: Зниження капіталовкладень в апаратне забезпечення і трудові ресурси; зменшення ризику втрати інвестицій; плавне ітеративне оновлення.

Недоліки: Як і в попередніх двох моделях, централізація вимагає надійних заходів безпеки. Прикладами SaaS є Gmail, Google Docs, Netflix, Photoshop.com, Acrobat.com, Intuit QuickBooks Online, IBM LotusLive, Unyte, Salesforce.com, Sugar CRM і WebEx. Значна частина зростаючого ринку мобільних застосунків також є реалізацією SaaS.

Моделі обслуговування за засобами доступу вмістом можна побачити в таблиці 1.2.

Таблиця 1.2 - Моделі обслуговування за засобами доступу і управління

| Моделі обслуговування | Засоби доступу і управління | Вміст |
|---|---|---|
| ПЗ як сервіс (SaaS) | Веб-браузер | Хмарні програми: соціальні мережі, офісні застосунки, системи управління вмістом, інтелектуальна обробка даних. |
| Платформа як сервіс (PaaS) | Хмарна середовище розробки | Хмарна платформа: мови програмування, бібліотеки, утиліти конфігурації композицій сервісів, структуровані дані. |
| Інфраструктура як сервіс (IaaS) | Система управління віртуальної інфраструктурою | Хмарна інфраструктура обчислювальні сервера, сховища даних, організація мережевих з'єднань (Брандмауери, балансування навантаження). |

Однак це далеко не всі моделі обслуговування хмарних технологій. Існують також інші.

Апаратне забезпечення як послуга (Hardware as a service, HaaS) - це модель закупівель, аналогічна лізингу або ліцензуванню, при якій устаткування, що належить постачальнику керованих послуг (MSP), встановлюється на об'єкті замовника, а угода про рівень обслуговування (SLA) визначає відповідальність обох сторін. Іноді клієнт платить щомісячну плату за використання апаратного забезпечення, іноді його використання включається в структуру плати за установку, моніторинг та обслуговування апаратного забезпечення MSP. У будь-якому випадку, якщо обладнання виходить з ладу або застаріває, MSP відповідає за його виведення з експлуатації та заміну. Залежно від умов угоди SLA, виведення з експлуатації може включати стирання пропрієтарних даних, фізичне знищення жорстких дисків і підтвердження того, що старе обладнання було законно перероблено.

Модель HaaS може бути економічно ефективним способом для малого і середнього бізнесу, що дозволяє забезпечити співробітників найсучаснішим обладнанням при мінімальних витратах. HaaS можна протиставити моделям закупівель, заснованим на принципі "інфраструктура як послуга" (IIS) і керованому хостингу, в яких апаратне забезпечення розміщується на майданчику MSP.

DBaaS (Database as a Service, база даних як послуга) - це різновид PaaS. Використовуючи DBaaS, користувач може отримати доступ до бази даних будь-якого типу за запитом. Користувач може швидко розгорнути БД на будь-якому класі устаткування в середовищі обраної ним програмної платформи (операційної системи). [4]

Користувач може вибрати базу даних, вказавши її версію, загальну конфігурацію, ряд інших особливостей (наприклад, розміщення). БД за запитом можна розмістити в ОС на віртуальній машині або підключити в рамках контейнера.

За останні пару років постачальники хмарних послуг значно збільшили кількість пропозицій DBaaS. Компанія IBM, наприклад, надає доступ до масштабованої і повністю керованої бази даних через стандартні об'єктно-орієнтовані API.

DBaaS складається з компоненту управління базою даних, який керує всіма базовими екземплярами баз даних через API. Цей API доступний користувачеві через консоль управління, зазвичай веб-застосунок, що користувач може використовувати для управління і настройки бази даних і навіть для надання або скасування доступу до баз даних.

Переваги і недоліки баз даних як сервісів DBaaS :

Створення і ведення традиційної бази даних може бути дуже дорогим і трудомістким процесом, управління яким може бути ускладнене, особливо для підприємств з обмеженими ресурсами та вимогами, які використовують малі або середні бази даних.

DBaaS означає, що великі і малі підприємства можуть змінювати розміри своїх баз даних відповідно до своїх потреб і бюджета, а також динамічно масштабувати свої потреби в БД в міру зміни ситуації від дня до року.

DBaaS пропонує пакет послуг з управління даними, в якому компаніям не потрібно розгортати і керувати власними серверами і інфраструктурою БД. Бази даних розміщуються і керуються третьою стороною, а доступ до них надається користувачам на Cloud across the Globe за певною ціною.

Крім того, існує безліч інших факторів, які вимагають DBaaS, на відміну від традиційних БД. Деякі з них згадуються нижче:

- необхідність управління величезними обсягами даних;
- продуктивність. (Вартість зберігання і обслуговування такого величезного обсягу даних значно знижується);
- зміцнення систем післяаварійного відновлення і забезпечення безперебійного функціонування.

За прогнозом Міжнародної інформаційної корпорації (IDC), розгортання застосунків в хмарі збільшиться на 15,3% в порівнянні з аналогічним періодом

минулого року. У звіті також говориться, що використання хмарних обчислень дає 520% окупності інвестицій із наступних причин:

- На 70% швидше рух до ринку. Оскільки БД вже є, затримки щодо закупівель не потрібні. Ми можемо безпосередньо користуватися послугами БД і розмішувати наш застосунок.
- Зниження витрат на інфраструктуру на 75-85%. Як вже зазначалося вище, всі витрати на інфраструктуру, а також мережеві витрати і пов'язані з ними накладні витрати і технічне обслуговування не потрібні. Постачальник послуг DBaaS про все це подбає, і компанія отримує тільки перевагу використання послугою.

Наявні вагомі переваги даної полслуги.

Переваги використання DBaaS:

- висока масштабованість - майже нескінченна ємність сховища даних.
- економічність - це найбільша перевага, коли ви платите за те, що використовуєте. Також виключається вартість апаратного забезпечення і мережі.
- для компаній, що зазнають труднощі з управлінням своїми даними, хмара може надати недорогу альтернативу інвестиціям в інфраструктуру для управління всіма цими даними на своїх власних майданчиках.
- з DBaaS компанія платить за те, що вона використовує, і за час, який вона використовує. Його нескінченна масштабованість є великою перевагою, коли мова заходить про збільшення або зменшення обсягу пам'яті.
- витрати на ліцензування та оновлення бази даних несе постачальник послуг, і бізнес повинен інвестувати в це.
- про безпеку даних і безперебійності роботи також піклується постачальник послуг.

Існують деякі очевидні недоліки даної послуги по управлінню БД, які проявляються в наступному.

Недоліки використання DBaaS:

- немає прямого контролю доступу до бази даних. Якщо щось піде не так, ти безпорадний.
- відсутність контролю за фізичною безпекою серверів. У разі стихійного лиха в місці, де розташований ваш сервер або система виходить з ладу, вам може знадобитися час простою, якщо не відбудеться втрата даних.
- ви знаходитесь на милості управління сервером Хмарної бази даних без прямого контролю над конфіденційними даними.
-

Висновки до розділу 1

В цьому розділі було проведено аналіз сучасних хмарних технологій, різних моделей хмарних сервісів, дано визначення хмарним обчисленням та розглянута архітектура, що дозволяє хмарним сервісам бути такими якими вони є.

Також було виявлено актуальність використання хмарних технологій, що призводить до питання існування певних загроз, які пов’язані з існуванням даних технологій. Наявні загрози хмарним сервісам будуть розглянуті в наступному розділі. А в третьому розділі будуть розглянуті механізми захисту.

# 2 ЗАГРОЗИ ТА ВРАЗЛИВОСТІ ХМАРНИХ ОБЧИСЛЕНЬ

Хмарні обчислення як нова парадигма розподілених обчислень надають послуги на вимогу, скорочуючи капіталовкладення в інфраструктуру і максимально збільшуючи використання наявних ресурсів. Хмарні технології забезпечують мобільність застосунків і інфраструктурних сервісів, а також незалежність фізичних/апаратних платформ від існуючих розподілених обчислень і мережевих застосунків [5]. З ростом хмарних обчислень і забезпеченням доступності обчислювальних ресурсів для споживачів IT-індустрія стає більш гнучкою, спроможною і економічно ефективною для розробки та хостингу застосунків. Однак, приймаючи на озброєння цю нову потужну систему, безпека виявилась найбільш важливим і критичним питанням. Для розуміння проблем безпеки необхідно аналізувати загрози, вразливості і ризики як різні фактори, що впливають на безпеку хмарних обчислень.

Моделювання загроз у вигляді систематичного і всебічного аналізу загроз і вразливостей необхідно для забезпечення конфіденційності, цілісності та доступності для розгортання системи безпеки хмарних обчислень. Моделювання загроз збирає базову інформацію, необхідну у вигляді сценаріїв використання, зовнішніх залежностей, припущень про реалізацію, деталей реалізації внутрішньої і зовнішньої безпеки [6]. Був розроблений ряд методів моделювання загроз для оцінки та аналізу таких загроз і вразливостей. Розглянемо основні з них.

Моделювання загроз Microsoft представляє собою модель ефективного процесу, що включає п'ять логічних кроків від класифікації активів до усунення і класифікації загроз і вразливостей.

Microsoft’s Threat Analysis and Modeling (TAM) [6] - це модель, заснована на бізнес-цілях, які повинні бути досягнуті за допомогою програми. Інструменти TAM використовуються для генерування і класифікації загроз шляхом вимірювання їх шкідливого впливу на компоненти системи.

Практичний аналіз загроз (Practical Threat Analysis, PTA) [7] визначає ефективний план зниження ризиків для конкретної архітектури системи для отримання вартості активів системи.

Структура і методологія моделі загроз для персональних мереж (PNs) [8]. Ця модель загроз, заснована на аналізі персональної мережі, дає гарне уявлення про активи системи і їх вартості. Визначення всіх ресурсів по UML-діаграмам дає можливість захистити дані та мережеві функції від будь-яких загроз.

Моделювання загроз в широко поширеній обчислювальній парадигмі шляхом розробки хмарних обчислень. Кожен користувач стикається з різними сферами безпеки з множинною ідентифікацією. Наведена вище модель являє собою нове моделювання загроз, що включає проблему широко розповсюдженого комп'ютерного середовища.

На жаль, у зв'язку з ростом хмарних середовищ як великих розподілених обчислень, всі перераховані вище методики не включають в себе усі проблеми, тому для вирішення проблем повсякденної комп'ютерної безпеки необхідно розробити нові підходи до моделювання загроз.

В даному дослідженні представлена методологія моделі загроз для розгортання безпечного обчислювального середовища шляхом демонстрації загроз і вразливостей в хмарних обчисленнях і визначення рішень безпеки. Пропонована методологія побудови моделі загроз складається з декількох етапів, які після розгляду всіх етапів складають остаточну модель загрози для хмарних обчислень.

Питання безпеки хмарних обчислень: Перш ніж більш детально розглянути загрози, вразливості і рішення, характерні для хмарних середовищ, ми визначимо вразливість і загрозу наступним чином:

- загроза - це шкода або несанкціонований доступ, які можуть виникнути в результаті вразливості і знищити активи організації, її діяльність або системну інформацію
- вразливість - це будь-яка слабкість інформаційної системи, процедур системної безпеки, внутрішнього контролю або реалізації, яка може бути використана або викликана ресурсами загроз [9].

## 2.1 Загрози хмарних обчислень

Cloud Security Alliance (CSA) [10] в якості постачальника послуг із забезпечення безпеки випустив посібник з безпеки для критичних областей в хмарних обчисленнях для управління ризиками і розуміння загроз безпеці. Найбільш значні загрози, які пов'язані з характером хмарних обчислень на вимогу, класифікуються наступним чином:

- Втрата або витік даних (T1): Ця загроза оцінюється як найсерйозніша і жахлива загроза для бізнесу і споживачів. Будь-яке видалення даних постачальником послуг або аварія, наприклад, пожежа, може привести до втрати даних споживача.

- Викрадення облікового запису або послуги (T2): Ця слабкість дозволяє зловмисникам красти облікові дані і доступ до критично важливих областей послуг хмарних обчислень. Організація повинна заборонити обмін обліковими даними між різними службами і користувачами і використовувати надійні методи аутентифікації.

- Небезпечний інтерфейс (T3): Клієнти хмарних обчислень використовують інтерфейс прикладного програмування (API) або програмні інтерфейси для взаємодії і управління хмарними службами. Технології аутентифікації, контролю доступу та моніторингу для API захищають обчислювальні ресурси від шкідливих атак.

- Відмова в обслуговуванні (T4): Розподілена відмова в обслуговуванні (DDoS) є основною загрозою безпеці доступності, коли мова заходить про підвищення надійності організацій на загальнодоступних хмарних сервісах [11]. З іншого боку, ця атака не дозволяє користувачам отримати доступ до своїх даних або застосунків і не дає можливості дістатися до місця призначення. Постачальники хмарних послуг повинні бути впевнені в захисті доступності, а клієнти - в рівні захисту доступності всередині провайдера.

- Шкідливий інсайдер (T5): Система була пошкоджена авторизованим співробітником, діловим партнером або адміністратором, який має доступ до мережі або ресурсів. Ця небезпечна загроза зачіпає конфіденційність, цілісність

і доступність ділової інформації.

- Витік даних (T6): Однією з найгірших ситуацій для кожної організації є несанкціонований доступ або незаконний перегляд даних конкурентами. Шифрування даних може знизити ризик цієї загрози, але слід бути обережним з ключем шифрування, тому що якщо ви втратите його, ви втратите і свої дані.

- Зловживання хмарними службами (T7): Провайдери хмарних обчислень не проводять жорстких реєстраційних процедур, і будь-який користувач з діючою кредитною карткою може зареєструватися для отримання хмарних послуг [12]. Інтеграція слабких способів виявлення шахрайства при реєстрації дозволяє зловмисникові ефективно використовувати дані за допомогою агресивних хмарних моделей, таких як PaaS і IaaS.

- Недостатня належна обачність (T8): Зниження витрат, доступ до пулу ресурсів і поліпшення безпеки є найважливішими факторами, які можуть бути корисні для організації, щоб прискорити розвиток хмарних обчислень. Без розуміння середовища постачальника хмарних послуг (Cloud Service Provider, CSP), невідповідність очікувань було створено як критичне питання з контролю безпеки хмарних обчислень. Однак для того щоб ресурси були досить кваліфікованими, організації повинні розуміти пропозиції постачальників послуг і ризики.

- Небезпечна міграція віртуальних машин (T9): Переміщуючи різні віртуальні машини під час гібридних і об'єднаних хмар, зловмисники можуть отримати незаконний доступ до даних і передати їх на ненадійний хост. Віртуалізація як основний компонент IaaS є основною метою атак зловмисників. Надійні хмарні обчислення і технологія шифрування захищають ресурси даних від небезпечної міграції віртуальних машин.

Загроза загальних вразливостей існує у всіх моделях, оскільки базові компоненти, які розгортають інфраструктуру, платформи і застосунки, не забезпечують сильної ізоляції між моделями хмарних обчислень.

## 2.2 Вразливості хмарних обчислень

Перекладаючи критично важливі дані і застосунки організації на хмарні сервіси, слід враховувати їх різні істотні вразливості, основні характеристики хмари, відомі засоби контролю безпеки і найсучасніші хмарні пропозиції.

- Перехват сесії (V1): Під перехватом сесії мається на увазі відправка хакерами команди до веб-застосунку для отримання несанкціонованого доступу до інформації або використання слабких місць веб-служби для надання хакеру можливості здійснити такі ж дії, як видалення призначених для користувача даних або розсилка спаму в мережу через Інтернет.

- Вихід за межі віртуальної машини (V2): Ця вразливість дозволяє зловмисникові запускати на віртуальній машині код, що дозволяє операційній системі зламувати і взаємодіяти безпосередньо з гіпервізором для доступу до хостовой операційній системи та іншим віртуальним машинам. Для запобігання система повинна виявляти шкідливу активність на рівні віртуальних машин.

- Застаріла криптографія (V3): Розробка недостатньо надійного шифрування або його відсутність дозволяє зловмисникові розшифровувати зашифровані дані. Щоб захистити систему від цієї вразливості, користувач повинен бути впевнений, що справжні дані зашифровані, використовувати правильне зберігання ключів і розробити хороший алгоритм.

- Несанкціонований доступ до інтерфейсу управління (V4): Інтерфейс управління хмарою має доступ до користувачів хмарних послуг для управління службами за запитом. Несанкціонований доступ дозволяє зловмисникам отримати повний контроль над користувачами і застосунками.

- Інтернет-протокол (V5): Відсутність методів аутентифікації, що не входять в базовий протокол, дозволяє зловмисникам впроваджувати в мережу свій шкідливий трафік. З іншого боку, протокол IP або пов'язані з ним протоколи, такі як UDP і TCP, вразливі для різних типів атак типу відмова в обслуговуванні (DoS), включаючи перехоплення сеансу.

- Відновлення даних (V6): Хмарні обчислення дозволяють розподіляти або перерозподіляти ресурси між різними користувачами. Ця еластична характеристика може призвести до крадіжки даних, витік даних і іншим загрозам безпеки. Більшість організацій використовують сторонніх

постачальників для відновлення даних, тому вони повинні враховувати ризик безпеки при роботі з даними з зовнішніми компаніями і забезпечувати належну перевірку безпеки постачальника послуг

- Виставлення рахунків (V7): Лічильники хмарних обчислень і послуги вимірювання, такі як зберігання, обліковий запис користувача і обробка, використовуються для оптимізації надання послуг. Застосовні вразливості включають обробку даних обліку і виставлення рахунків, а також витік рахунків.

- Замок постачальника (V8): Блокування постачальника - це ситуація, коли користувач хмари залежить від одного постачальника і не може мати справу з іншим постачальником без істотних незручностей. Відсутність стандартів є основною причиною того, що користувачі не можуть легко переходити від одного провайдера до іншого.

## 2.3 Порівняння існуючих фреймворків моделювання загроз

Перш ніж розглянути пропоновану модель, ми проілюструємо деталі існуючих моделей в таблиці А.1, щоб проаналізувати і порівняти існуючі моделі із запропонованою моделлю. Таблиця А.1 заснована на характеристиках моделі загрози, яка буде використана в хмарних обчисленнях – приведена в додатку А

## 2.4 Модель Аміні-Джаміла

Пропонована модель забезпечення безпеки хмарних обчислень представлена в додатку А на рисунку А.1. Ця модель складається з чотирьох основних етапів, кожен з яких включає в себе кілька підпунктів. Першим кроком є виявлення активів і роз'яснення того, хто або що має доступ до них. Довіра, представлена терміном надійність, який використовується для позначення безлічі надлишкових веб-служб для позначення кваліфікованої довіри між користувачем і постачальником послуг або між різними постачальниками. Далі, на другому етапі визначається здатність провайдера надавати вимоги користувачів в буквальному сенсі слова. Виявлення унікальних загроз і їх усунення шляхом розробки відповідних контрзаходів

було продемонстровано на третьому етапі. Основною метою даного етапу є виявлення і подальше виявлення нових загроз для підвищення безпеки. На останньому етапі представлений системний рейтинг для виявлення найбільш небезпечних і дієвих загроз і вразливостей.

Визначте активи. Актив IT - це дані, програмне або апаратне забезпечення, що належать компанії і використовуються для ведення бізнесу. Організації повинні бути впевнені в тому, що до цих ресурсів мають доступ авторизовані користувачі, і вони налаштовані на використання новітніх технологій безпеки для захисту від загроз безпеці та вразливостей. Найбільш несанкціонований доступ або ефективні зміни в процесі управління конфігурацією здійснюються неофіційними користувачем в систему. Тому для блокування атаки необхідно знати машини підприємств і їх місце розташування. Сьогодні організації використовують здійсненні і ефективні інструменти управління активами (SAM) для аналізу власних даних, програмного та апаратного забезпечення. У будь-якому випадку, ці інструменти, оптимізують і керують IT-активами, допомагають організації визначити, що у них є, контролювати витрати і ризики і підвищувати безпеку. Для виявлення та моніторингу активів і визначення ролі користувачів в запропонованій нами структурі були запропоновані деякі кроки (рисунок 2.1).

Оцініть надійність. Довіра - це надійність розподілених систем між двома організаціями, які покладаються на свою надійність для забезпечення безпеки. В даний час довіра грає важливу роль в інтеграції різнорідних середовищ для оцінки надійності шляхом забезпечення конфіденційності, цілісності, доступності і достовірності. Для визначення та оцінки довіри між різними розподіленими системами необхідно розділити адміністративні області, виявити уразливості системи і встановити рівень довіри.

Визначення загроз і вразливостей. Розвиваючи інформаційно-комунікаційні технології, організації виявляють все більшу цікавість до аутсорсингу своїх обчислювальних ресурсів на віртуальних доменах. Фактично, цей величезний обсяг даних може бути пошкоджений загрозами з боку різних ресурсів (дії співробітників або зловмисні атаки зловмисників). Тому при управлінні та оцінці ризиків необхідно розуміти і аналізувати загрози і вразливості як найважливіші питання безпеки. Таким чином, ефективна класифікація безпеки необхідна для виявлення і систематизації загроз і

вразливостей за класами, заснованим на передбачуваному впливі атак, і розробки рішень щодо запобігання ефективних загроз для системи.

Визначте результат: Оцінка і ранжування ступеня серйозності загроз і вразливостей для прийняття обґрунтованого рішення про те, що робити для захисту системи від шкідливого впливу, є останнім кроком пропонованої моделі. Наявність системи ранжування загроз і вразливостей необхідна для економії часу запобігання більш серйозних загроз безпеці. Результатом цього кроку може стати список або база даних профілів загроз і вразливостей, що складається з відсортованих ризиків безпеки.

## 2.5 Вибір підсистеми контролю доступом

Контроль доступу, як правило, являє собою політику або процедуру, яка дозволяє, забороняє або обмежує доступ до системи. Крім того, він може відстежувати і реєструвати всі спроби доступу до системи. Контроль доступу може також виявляти користувачів, які намагаються отримати несанкціонований доступ до системи. Це механізм, який дуже важливий для захисту в комп'ютерній безпеці. Існують різні моделі контролю доступу, включаючи найбільш поширені моделі: мандатне керування доступом (англ. Mandatory access control, MAC), вибіркове керування доступом (англ. Discretionary access control, DAC) і керування доступом на основі ролей (англ. Role Based Access Control, RBAC). Всі ці моделі відомі як моделі контролю доступу на основі ідентифікації. У всіх цих моделях управління доступом користувачі (суб'єкти) і ресурси (об'єкти) ідентифікуються за унікальними іменами. Ідентифікація може здійснюватися безпосередньо або за допомогою ролей, призначених суб'єктам. Ці методи контролю доступу ефективні в незмінних розподілених системах, де є відомий набір користувачів з відомим набором сервісів.

В даний час великі відкриті розподілені системи розвиваються дуже швидко. До них відносяться мережеві обчислення і хмарні обчислення. Ці системи схожі на віртуальні організації з різними автономними

доменами(галузями). Взаємини між користувачами і ресурсами динамічні і носять більш вузькоспеціалізований характер в хмарних системах. У цих системах користувачі і постачальники ресурсів не належать до однієї й тієї ж галузі безпеки. Користувачі зазвичай ідентифікуються по їх атрибутам або характеристиками, а не за заздалегідь заданими ідентифікаційними даними. У таких випадках традиційні моделі контролю доступу, засновані на ідентифікації, не надто ефективні, і тому доступ до системи повинен здійснюватися на основі рішень, заснованих на певних атрибутах.

Крім того, в хмарной системі автономні домени мають окремий набір політик безпеки. Отже, механізм контролю доступу повинен бути гнучким для підтримки різних видів областей і політик. З розвитком великих розподілених систем, керування доступом на основі атрибутів (ABAC) стає все більш важливим.

Перший спосіб, яким система забезпечує безпеку своїх ресурсів і даних, - це контроль доступу до ресурсів і самої системи. Однак контроль доступу - це більше, ніж просто контроль того, які користувачі (суб'єкти) можуть отримати доступ до обчислювальних і мережевих ресурсів. Контроль доступу також дозволяє управляти користувачами, файлами і іншими ресурсами. Він контролює права користувача на доступ до файлів або ресурсів (об'єктів). У системах контролю доступу застосовуються різноманінтні кроки, такі як ідентифікація, аутентифікація, та авторизація, перш ніж предоставити доступ до ресурсів або об'єкту в цілому.

На ранніх етапах інформаційних технологій дослідники і технологи усвідомили важливість запобігання втручанню користувачів до роботи один одного в спільних(shared) системах. Були розроблені різні моделі контролю доступу. Особистість користувача була основним показником, що дозволяла користувачам використовувати систему або її ресурси. Цей підхід отримав назву "Контроль доступу на основі ідентифікації" (англ. Identification Based Access Control, IBAC). Однак, із зростанням мереж і кількості користувачів в них, було встановлено, що IBAC виявився слабким для захисту від такої кількості користувачів. Були введені вдосконалені концепції контролю доступу, які

включали власника/групу/громадськість. IBAC також виявився проблематичним і для розподілених систем. Управління доступом до системи і ресурсів стало важким і вразливим для помилок. Був представлений новий метод, відомий як керування доступом на основі ролей (RBAC). Керування доступу на основі ролей визначає доступ користувача до системи на основі його ролі. Роль, призначена користувачу, в основному заснована на принципі найменших привілеїв. Роль визначається з найменшою кількістю дозволів або функцій, необхідних для виконання роботи. Дозволи можуть бути додані або видалені, якщо змінюються привілеї для певної ролі. Однак проблеми стали очевидними, коли RBAC була поширена на адміністративні домени. І виявилося важко домовитися про те, які привілеї асоціювати з тією чи іншою роллю. Тоді і з'явилася модель керування доступом на основі атрибутів (ABAC). У ABAC доступ надається по атрибутам, які користувач може надати, як, наприклад, дата народження або номер телефону. Однак дійти згоди по необхідному набору характеристик дуже важко, особливо між численними установами і організаціями. Всі методи контролю доступу засновані на аутентифікації користувача на сайті та під час запитів. Іноді такі методи називають методами керування доступом на основі аутентифікації. У всіх цих методах потрібен тісний зв'язок між доменами. Крім того, всі ці підходи ускладнюють призначення підмножин прав адміністратора. Це призводить до того, що загальні схеми використання, можуть бути реалізовані шляхом скорочення функціональних можливостей або порушення принципу найменших привілеїв. Даміані Е. в своїй праці New paradigms for access control in open environments зробив спробу забезпечити єдиний фреймворк для специфікації і правозастосування ABAC. П. Бонатті в роботі A unified framework for regulating access and information release on the web представив єдину структуру для формулювання і обгрунтування обмежень доступу до послуг та розкриття інформації на основі відповідних атрибутів організації.

Контроль доступу на основі атрибутів розширює контроль доступу на основі ролей, в цілому, наступними функціями:

- делегування повноважень на визначення атрибутів;
- децентралізація атрибутів і функцій;
- Інтерференція(перетин) атрибутів.

ABAC забезпечує політику конфіденційності повноважень. Це дозволяє організації зберігати свою автономність при ефективній співпраці. Крім того, вона забезпечує автоматичні переговори про довіру, які можна перевіряти по міру необхідності.

## 2.6 Вибір підсистеми аудиту

Хмарні обчислення дуже перспективні для застосунків в області інформаційних технологій (IT), однак для персональних користувачів і підприємств ще належить вирішити ряд питань, пов'язаних із зберіганням даних і розгортанням застосунків в середовищі хмарних обчислень. Безпека даних є одним з найбільш значних перешкод на шляху їх впровадження, і за нею йдуть такі питання, як нормативно-правова відповідність, конфіденційність, довіра і правові питання. Тому однією з найважливіших цілей є підтримка безпеки і цілісності даних, що зберігаються в хмарі, з огляду на критичний характер хмарних обчислень і великий обсяг складних даних, які оброблюються. Спочатку слід усунути стурбованість користувачів безпекою, щоб зробити хмарне середовище надійним і допомогти користувачам і підприємствам адаптуватися до нього у великих масштабах.

Основні проблеми в області безпеки хмарних даних включають конфіденційність даних, доступність даних, їх розміщення і безпечну передачу. Загрози, втрата даних, перебої в обслуговуванні, зовнішні шкідливі атаки і проблеми з багатокористувацької атакою - ось основні проблеми безпеки в хмарі.

Цілісність даних в хмарній системі означає збереження цілісності інформації, що зберігається. Дані не повинні бути втрачені або змінені неуповноваженими користувачами. Постачальникам хмарних обчислень довіряють підтримувати цілісність і точність даних. Конфіденційність даних

також є важливим аспектом з точки зору користувача, оскільки він зберігає свої особисті або конфіденційні дані в хмарі. Для забезпечення конфіденційності даних використовується стратегія контролю доступу. Проблема конфіденційності даних може бути вирішена за рахунок підвищення надійності хмарних обчислень. Тому безпека, цілісність та конфіденційність даних, що зберігаються в хмарі, повинні враховуватися і є важливими вимогами з точки зору користувача. Для досягнення всіх цих цілей необхідно розробляти і впроваджувати нові методи або прийоми.

Аудит даних вводиться в хмарні обчислення для безпечного зберігання даних. Аудит - це процес перевірки даних користувача, який може бути здійснений як самим користувачем (власником даних), так і стороннім аудитором (англ. third party auditor). Це допомагає підтримувати цілісність даних, що зберігаються в хмарі. Роль верифікатора розділена на дві частини: перша - приватний аудит, тобто тільки користувач або власник даних має право перевіряти цілісність даних, що зберігаються. Ніхто інший не має права допитувати сервер щодо цих даних. Але це призводить до збільшення верифікаційних операцій на користувача. Друга - можливість публічного аудиту, який дозволяє будь-якій людині, а не тільки клієнту, зробити запит до сервера і виконати перевірку достовірності даних за допомогою TPA. TPA - це сутність, яка використовується для того, щоб діяти від імені клієнта. Вона володіє всіма необхідними знаннями, можливостями і професійними навичками, які необхідні для виконання роботи з перевірки цілісності даних. Важливо, щоб TPA ефективно перевіряв хмарне сховище даних без запиту локальної копії даних. Він повинен володіти нульовими знаннями про дані, що зберігаються на хмарному сервері.

В хмарному середовищу присутні три мережевих об'єкта - клієнт, хмарний сервер і TPA. Клієнт зберігає дані на сервері, що надається провайдером хмарних послуг (англ. cloud service provider, CSP). TPA здійснює перевірку даних клієнта, періодично перевіряючи цілісність даних на вимогу і

повідомляє клієнта про будь-які зміни або помилки, виявлені в даних клієнта. На рисунку 2.1 показана архітектура хмарного сховища даних.


Рисунок 2.1 Хмарна архітектура зберігання даних

В додатку А в таблиці А.1 проводиться порівняння різних факторів, таких як використовуваний метод, підтримка громадського аудиту, збереження конфіденційності, динаміка даних і пакетний аудит. Вона також показує, чи підтримується цілісність і конфіденційність даних, що зберігаються на хмарному сервері, чи ні. З таблиці 2.2 ясно видно, що для перевірки цілісності даних застосовувалися різні методи. Але з кожним з методів пов'язані певні проблеми.

Необхідно обрати ефективний протокол публічного аудиту, який дозволив би подолати обмеження, що накладаються іншими системами аудиту і як ми можемо побачити з таблиці А.1 - це система аудиту Свапналі Мор.

## 2.7 Вибір підсистему криптографічного захисту БД

Зберігання та обробка конфіденційних даних в системі, яка надана третьою стороною збільшує ризик несанкціонованого розголошення, якщо система скомпрометована зловмисником (який сам може бути агентом від цього стороннього постачальника послуг).

Одне з можливих рішень цієї проблеми - шифрування даних на клієнтській машині (яка вважається довіреною) перед завантаженням їх на сервер, а запити виконувати, отримуючи назад зашифровані дані з сервера, розшифровувати їх та виконувати запит на машині клієнта. Однак для запитів до бази даних і аналітичних навантажень потрібно передати набагато більше даних ніж необхідно, оскільки велика частка бази даних зчитується для виконання запиту, але сам результат зазвичай є невеликим агрегованим набором данних чи згорткою даних, наприклад, сумою вартості товарів.

При аналізі літературних джерел було знайдено три системи для вирішення цього питання: CryptDB, MONOMI та система Віктора Телло.

MONOMI спирається на попередню роботу з виконання запитів в зашифрованих базах даних CryptDB, та вирішує деякі її проблеми, не привносячи нових недоліків. Тому вважаю розглядання CryptDB не доцільним, через існування тієї ж за ідеологією, але кращою по реалізації системи.

MONOMI представляє підхід, заснований на роздільному виконанні клієнт/сервер. Використовуючи алгоритми для шифрування, які дозволяють виконувати операції над зашифрованими даними, такі як порівняння та групування. Роздільне виконання запиту дозволяє MONOMI виконувати частину запиту на сервері. Для інших частин запиту, які взагалі не можуть бути виконані на сервері MONOMI завантажує клієнту проміжні результати і виконує там остаточні обчислення. Схему MONIMI можна побачити на рисунку 2.3

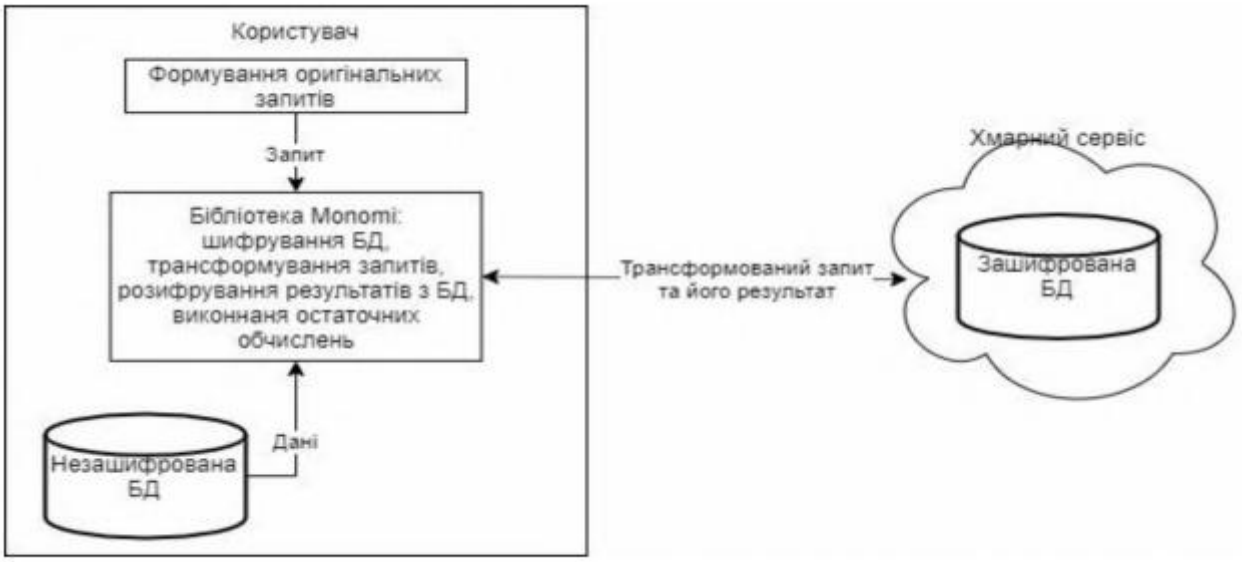


Рисунок 2.3 - Схема MONOMI

MONOMI можна порівняти с системою Віктора Телло, яка схематично зображена на рисунку 2.4.


Рисунок 2. 4 - Схематичне зображення системи Віктора Телло

В MONOMI досить суттєва частина обчислень виконується на стороні клієнта, що заперечує суттєву перевагу використання хмарних баз даних та алгоритми, що використовуються в даній системі для виконання операцій над зашифрованими даними, допускають витік даних, а саме дублікати, порядок та частково відкритий текст.

В той час як в системі Віктора Телло використовується довірена третя сторона (окрема сутність), а не клієнт, для виконання усіх проміжних обчислень,

таких як: виконання проміжних обчислень, розбиття та індексування бази даних в зашифрованому вигляді до хмарної бази даних, та не допускається витік даних. Тому в сучасних хмарних базах даних повинна використовуватися система Віктора Телло.

## Висновки до розділу 2

В даному розділі були розглянуті вразливості та загрози в сфері хмарних послуг, існуючі фреймворки для моделювання загроз, та було обрано найкращий з запропонованих фреймворків за декількома важливими критеріями.

За допомогою методологій моделювання загроз можна ефективно виявляти та оцінювати ризики безпеки в складних системах, що дозволяє архітекторам систем пом'якшувати потенційні проблеми безпеки на ранніх етапах, коли їх відносно легко вирішити, та й на пізніх етапах життєвого циклу систем.

Розглянуті загрози та вразливості показують, що для безпечного збереження інформації потрібно застосовувати засоби захисту від неправомірного її використання.

Було визначено які з існуючих алгоритмів, протоколів, моделей має сенс використовувати в сучасних підсистемах безпеки хмарних баз даних.

# 3 СИСТЕМА БЕЗПЕКИ ХМАРНИХ БАЗ ДАНИХ

На основі аналізу, який було проведено в попередніх розділах було зроблено висновок про важливість підсистем системи захисту та про технології які повинні в них використовуватися, а саме підсистеми безпеки хмарних баз даних про які піде мова:

- контроль доступу;
- аудит:
- криптографічний захист БД.

## 3.1 Підсистема контролю доступом

Керування доступом на основі атрибутів.

Найбільш важливим механізмом безпеки в хмарних сервісах є контроль доступу, і традиційні моделі контролю доступу не можуть бути застосовані для хмарних сервісів через їх особливості. В результаті аналізу минулого розділу стало очевидно, що в сучасних системах хмарних обчислень потрібно використовувати керування доступом на основі атрибутів. Велика кількість ресурсів, велика кількість динамічних користувачів, динамічні і гнучкі конструкції - ось деякі з важливих особливості хмарних сервісів. Крім того, кожен автономний домен в хмарній системі має свою власну політику безпеки, а саме ACL (список контролю доступу), декларація рішення про авторизацію SAML чи заява про політику XACML. Тому важливо мати гнучку модель керування доступом, щоб мати всі ці різноманітні політики в різних доменах. В ABAC рішення по надання доступу приймаються на основі атрибутів запитуючої сторони, сервісу, ресурсів і середовища. Керування доступом на основі атрибутів складається з чотирьох елементів:

- Запитуюча сторона (Requestor, скорочено Req): посилає запити в хмарний сервіс, викликаючи цим дії в сервісі;

- Сервіс (Serv): програмне і апаратне забезпечення з мережевим інтерфейсом і заздалегідь визначеними операціями;
- Ресурс (Res): то над чим здійснюється дія одним або декількома хмарними сервісами;
- Навколишнє середовище (Env): містить інформацію, яка може бути корисна для прийняття рішення про надання доступу, таку як дата і час. Воно може бути не пов'язане з будь-якою сутністю. Кожна сутність має атрибути, що визначають її особистість і ознаки. Атрибути особистості визначені наступним чином:

$$Attr(Req) = [Req\ Attr_i \mid i \in [1,/]\};$$
$$Attr(Serv) = \{Serv\ Attr_j \mid j \in [1\ ,J]\};$$
$$Attr(Res) = [Res\ Attr_k \mid k \in [1,K]\};$$
$$Attr(Env) = [Env\ Attr_t \mid t \in [1, L]\},$$

де I, J, K і L є цілими числами, яку відображають максимальну кількість атрибутів для кожної сутності.

Політики безпеки підтримуються системою авторизації хмарного сервісу. Кожна система може мати свій власний метод опису політики. Щоб забезпечити інтеграцію різних політик і зробити контроль доступу більш масштабованим, кожна політика інкапсулюється як незалежна одиниця. Політика, яку ABAC підтримує в якості більш широкого набору політик, визначається наступним чином:

- *Policy = [Pm Є [1,M ],P m* — це політика}

Рішення про доступ приймається функцією прийняття рішень *df(). $P_n$df ( )* є функцією оцінки в рамках політики *Pn* , яка визначається наступним чином:

*$P_n$df(Attr(Req), Attr (Serv),Attr(Res),Attr(Env)* = дозволити або заборонити

Атрибути можуть мати наступний вигляд:

- ReqAttrl = Attribute (GID=”admm”=”######”)
- ServAttrl = Attribute (Special Type=”Paas”, Service Name = “platform creation”)

- ResAttr1= Attribute (Computing-”Node1 and Node2”, networiking=”switch1”)
- EnvAttrl = Attribute (Service Time=”current Time”, domain=”Cloud1 .Clusterl and Cloud2.Cluster1”)

*Характеристики ABAC:*

- Ієрархічна структура політики, заснована на концепції абстракції та інкапсулювання;
- Набір політик складається з різних стратегій, які потребують підтримки;
- Політики мають власні алгоритми прийняття рішень;
- Не використовує уніфікований метод для опису кожної політики;
- Ефективна підтримка різних стратегій;
- Модель більш гнучка і розширювана.

Рішення по контролю доступу дуже важливі для будь-якої спільної системи. Однак для такої великої розподіленої системи, як хмарна система, рішення про доступ має бути більш гнучким і масштабованим. Саме тому для аудиту використовується керування доступом на основі атрибутів.

## 3.2 Підсистема аудиту

Обрана система Свапналі Мор розроблена для перевірки коректності хмарних даних стороннім аудитором, періодично або за запитом, без отримання всіх даних або створення додаткового навантаження на користувачів хмарних середовищ в режимі онлайн, і на самі хмарні сервери. Вона забезпечує відсутність розкриття даних стороннім аудитором в ході процесу аудиту. Вона підтримує правильність зберігання даних, їх цілісність та конфіденційність.

Пропонована схема складається з трьох основних елементів: власник даних, сховище хмарного серверу і сторонній аудитор. Власник або користувач даних несе відповідальність за поділ файлу на блоки, шифрування з використанням алгоритму AES, генерування хеш-значення SHA-2 для кожного

файлу, конкатенацію хешів і генерацію сигнатури RSA на них. Хмарний сервер використовується тільки для зберігання зашифрованих блоків файлів. Таким чином, у нього немає додаткового навантаження по обчисленню верифікаційних доказів. Під верифікаційним доказом тут мається на увазі генерація хешей для зашифрованих блоків, їх конкатенація та генерація цифрового підпису для верифікації. Це завдання виконується самим стороннім аудитором. Коли клієнт або власник даних запитує аудит даних у стороннього аудитора, він одразу запитує зашифровані дані з хмарного сервера. Після отримання даних аудитор генерує хеш-значення для кожного блоку зашифрованих файлів. Він використовує той самий алгоритм SHA-2, що і клієнт. Пізніше він сконкатенує ці хеш-значення і сгенерує сигнатуру RSA для цього файлу. У процесі верифікації підпис, що генерується аудитором, і підпис, що зберігається аудитором, який надається користувачем даних, порівнюються аудитором. Якщо вони збігаються одна з одним, це означає, що дані не пошкоджені і дані не були підроблені сторонніми особами або зловмисниками. Якщо підписи не збігаються, це вказує на те, що цілісність даних була порушена або підроблена. Результати перевірки цілісності даних надаються власнику даних. На рисунку 3.1 показана архітектура пропонованої схеми аудиту

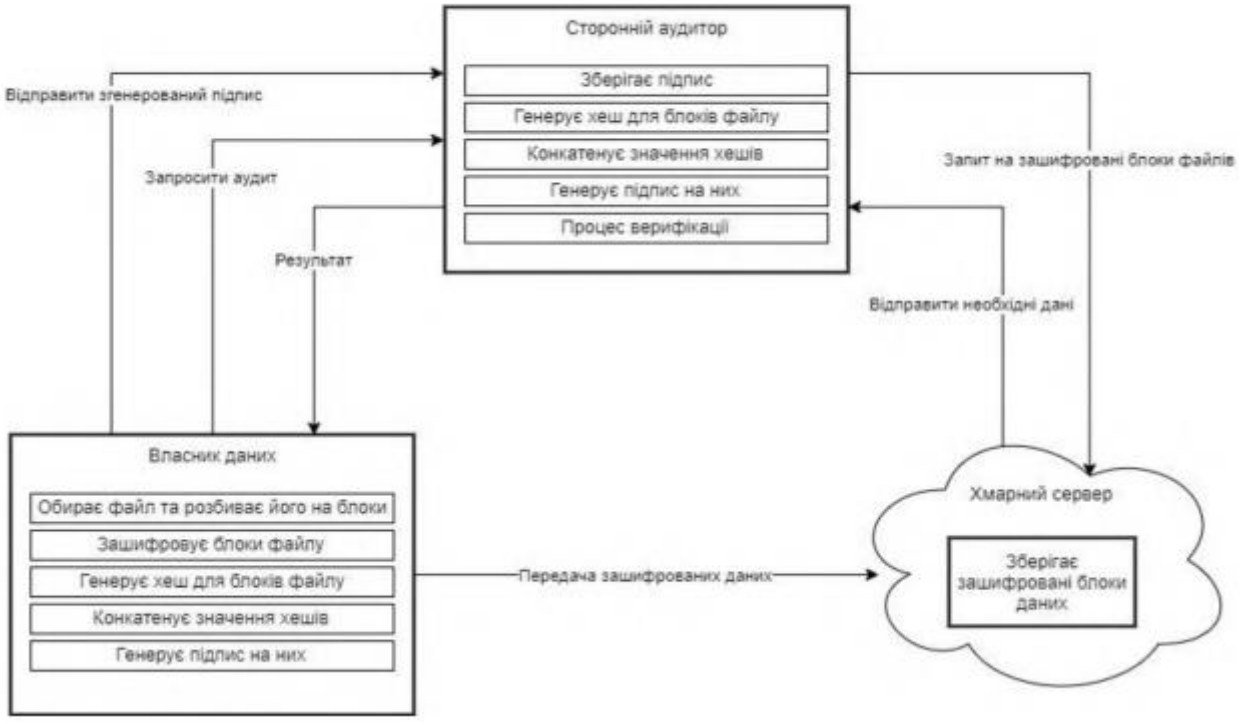


Рисунок 3.1 - Архітектура схеми аудиту

Власник даних є важливою частиною цієї системи. Він виконує більшу частину обов'язків, пов'язаних з даними. У запропонованій схемі аудиту власник даних спочатку виконує вхід і реєстрацію на хмарному сервері і в сервісі стороннього аудитора. Новий користувач повинен спочатку зареєструватися, заповнивши реєстраційну форму, і бути активним в системі. Йому буде видано повідомлення про успішну реєстрацію. Якщо користувач вже є користувачем системи, то він може виконати вхід в систему. Якщо логін користувача та пароль існують в базі даних, то вони успішно увійдуть в неї як дійсні користувачі, інакше вони отримають повідомлення про помилку.

Після успішного входу власник даних вибере файл, який він чи вона хоче зберегти на хмарному сервері. Обраний ним файл буде розділений на кілька блоків. Для того щоб здійснити розбиття потрібного файлу на блоки, використовується алгоритм FileSplitter. У цьому алгоритмі перевіряється, чи існує файл чи ні. Якщо існує, то файл розбивається на блоки заданого розміру, який залежить від розміру файлу. Наприклад, якщо розмір файлу 23kb, то він

буде розділений на 20kb і 3kb. Тут в прикладі розмір розбиття встановлений в 20 кб. Далі, використовується стійкий алгоритм шифрування AES (Advanced Encryption Standard) для забезпечення конфіденційності даних. Розділені блоки тепер шифруються власником даних за допомогою алгоритму AES. Кожен блок файлу буде зашифрований і збережений на клієнті. Копія зашифрованого файлу буде передана на хмарний сервер для зберігання. Він шифрує 128-бітові блоки даних за допомогою симетричних ключів розміром 128 біт. Після шифрування блоків, хеш-значення для блоків генерується окремо. Для цього використовується алгоритм хешування SHA-2. Після генерації хешей, хеші для кожного блоку конкатенуються і на ньому ставиться цифровий підпис RSA. Цифрові підписи використовуються для перевірки походження повідомлень. Пізніше цей підпис відправляється в сервіс стороннього аудитора, де він використовує цей підпис для перевірки цілісності даних, що зберігаються в сховищі хмарного сервера. Власник даних має право вимагати проведення перевірки цілісності даних у стороннього аудитора. На рисунку 3.2 показана робота власника даних в рамках цієї схеми аудиту.

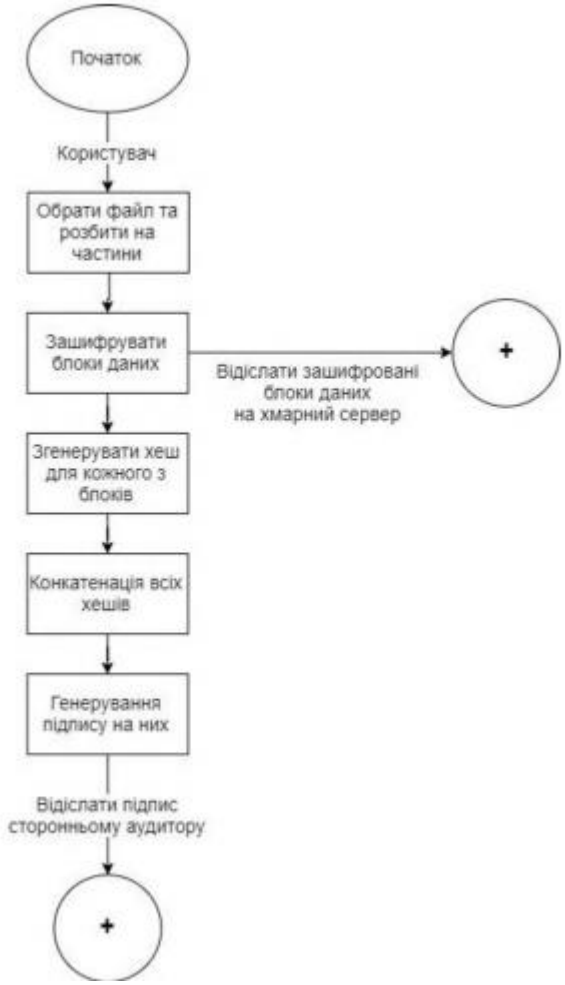


Рисунок 3.2 - Блок-схема роботи власника даних

Власник даних використовує хмарне сховище для зберігання зашифрованих даних. Так як дані зберігаються в зашифрованому вигляді, то хмарний сервер не має ніяких знань про дані. Як і в разі перетворення хмарного сервера в шкідливий сервер або атаки з боку зовнішнього зловмисника, дані не буде легко отримати, так як вони знаходяться в зашифрованому вигляді і сервер не знає про алгоритм шифрування, реалізований власником даних.

В даній схемі для виконання завдання з перевірки даних використовується сторонній аудитор. Він проводить аудит даних або періодично, або на вимогу клієнта. Після отримання запиту від користувача або власника даних про проведення аудиту сторонній аудитор починає процес аудиту. Аудитор також зберігає підпис, який був створений власником даних. Він дотримується того ж

алгоритму, який виконується власником даних, а саме, генерування хеша для зашифрованих блоків даних, їх конкатенація та генерування підпису на них. Пізніше, в процесі верифікації, він порівнює два підписи. Якщо вони збігаються, це означає, що цілісність даних підтримується. Інакше не підтримується. Це означає, що дані не були підроблені або змінені. Аудитором відповідні результати надаються власнику даних. На рисунку 3.3 показана робота стороннього аудитора в рамках розробленої  схеми аудиту.

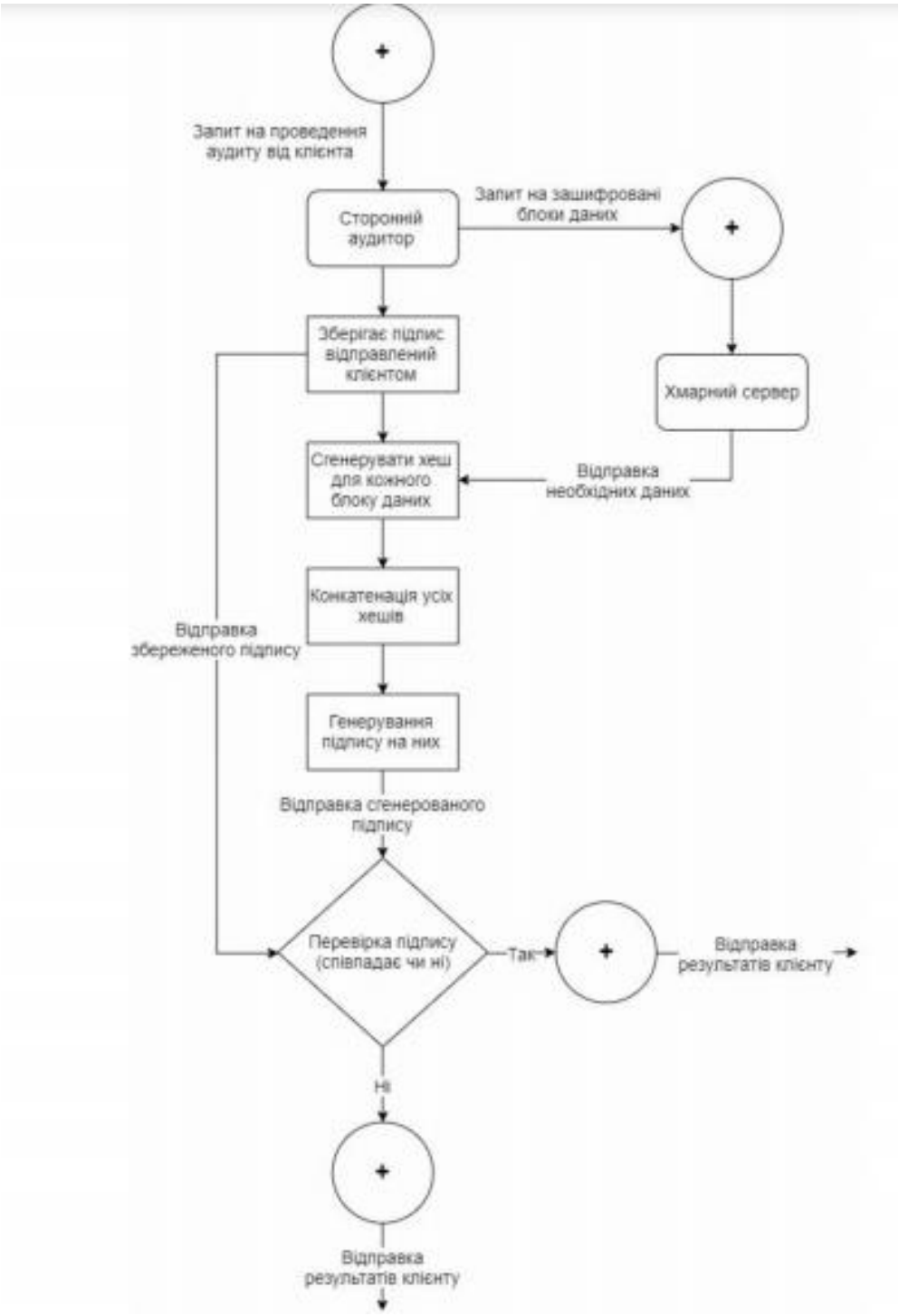


Рисунок 3.3 - Технологічна схема роботи стороннього аудитора

## 3.3 Підсистема криптографічного захисту бази даних

Незважаючи на те, що хмарні обчислення мають ряд переваг, при обговоренні проблем пов'язаних з безпекою даних, зберігаються деякі побоювання щодо безпеки даних, особливо коли виникають питання про те, хто

може бачити дані. Віктор Фуентес Телло запропонував рішення[13], яке допомагає захистити дані за допомогою таких методів, як шифрування і розбиття на секції, яке ми розглянемо далі.

### 3.3.1 Використання довіреної третьої сторони

Постачальники хмарних послуг пропонують різні можливості компаніям, які зацікавлені у використанні послуг зберігання, обробки і зв'язку для зниження витрат, пов'язаних з високодоступною технологічною інфраструктурою (апаратне і програмне забезпечення). Різні автори розробили методи з використанням довіреної третьої сторони (ДТС), в яких ДТС виконує функцію управління діяльністю і зв'язку між сторонами. У цьому типі рішень дані контролюються ДТС, і користувачі в організаціях повинні запитувати доступ до даних і послуг. У моделі, що розглядається в цій роботі, ДТС визначається як зовнішній сервер, який виконує кілька операцій зі зберігання і вилучення даних із хмари. ДТС працює з хмарним сервером для підвищення безпеки даних.

### 3.3.2 Шифрування даних

Користувачі і компанії турбуються про свої дані, коли дані зберігаються на хмарних серверах. Основним рішенням є шифрування даних перед відправкою їх на хмарні сервери. Коли клієнт запитує будь-яку інформацію з зашифрованої таблиці, він повинен запросити всю таблицю, так як неможливо відфільтрувати кортежі в зашифрованій таблиці. Це призведе до надмірного робочого навантаження на клієнті, що вимагають шифрування і розшифровки даних. Для зменшення цього навантаження можна додати додаткову інформацію індексування, яка зберігається разом з зашифрованими таблицями в хмарній базі даних. Індексні дані можуть бути використані системою керування базою даних для вибору необхідних даних з сервера або хмари, які будуть повернуті у відповідь на запит.

### 3.3.3 Система Віктора Телло

Система складається з трьох основних сегментів: середовище клієнта, довірена третя сторона і хмарне середовище. Спочатку, оригінальні дані будуть відправлені ДТС з клієнтської машини для зберігання корпоративної бази даних в хмарному середовищі зберігання. Організація і ДТС спільно використовують закритий ключ для шифрування і розшифровки файлів, якими вони обмінюються. ДТС виконує наступні функції: виконання методів розбиття над даними, шифрування записів і відправки зашифрованих записів в хмарну службу зберігання даних. На рисунку 3.4 показана схема, що ілюструє дану систему.


Рисунок 3.4 - Модель Віктора Телло

Далі, ДТС управляє запитом від клієнта, як тільки користувач в організації надсилає цей запит. Для цього ДТС аналізує і модифікує запит в залежності від методу розбиття, використовуваного для отримання індексів. Потім в хмарний сервіс зберігання даних надсилається новий запит на отримання записів, які відповідають умові запиту. Нарешті, ДТС отримує записи, розшифровує зашифровані оригінальні кортежі, застосовує умови запиту, шифрує результати, використовуючи спільний ключ між ДТС і клієнтом, і відправляє результати клієнту. Для захисту зв'язку між ДТС і хмарою використовується криптографічний метод.

Під час первинного зберігання ДТС ділить дані на розділи і відправляє їх в хмарну базу даних. Для кожного відношення (R) з його атрибутами (a) в базі даних $R(a_i , a_2 , a_3 , a_n)$ існує зашифроване відношення (E) у хмарній базі даних КБ[ідентифікатор, $(a_i , a_2 , a_3 , ..., a_n)$ E]. Ідентифікатор, в зашифрованому відношенні, відповідає індексу для кожного кортежу в базі даних. Кожен атрибут в базі даних пов'язаний з індексом розбиття за допомогою функції ідентифікації.

Метою даної моделі є зниження навантаження на машину клієнта. Для забезпечення безпеки даних хмарний сервіс зберігає дані в зашифрованому вигляді, що ускладнює розшифровку інформації зловмисником. В цій моделі ДТС виконує більшу частину робочого навантаження. Спочатку ДТС розбиває дані на секції, а потім керує всім запитом, використовуючи результат процесу розбиття. Для цієї моделі важливо відзначити, що у ДТС з кожною організацією свій ключ. Таким чином, ДТС може запропонувати дану послугу кільком організаціям.

Беручи до уваги міркування безпеки, ця система дозволяє організації працювати над своєю діяльністю, а не турбуватися про питання безпеки, оскільки управління нею здійснює третя сторона. Крім того, проблеми безпеки в хмарному середовищі мінімальні через використання зашифрованих повідомлень в процесі комунікації між сторонами. Переваги даної системи коротко викладені нижче:

- Основною перевагою даної системи є те, що роботи, які виконуються на машині клієнта, мінімальні.

Весь процес розділений на дві частини: ДТС, яка використовує функції співставлення для пошуку даних, і хмарний сервіс зберігання даних.

Зберігання даних в зашифрованому вигляді, а також використання криптографічних методів для комунікації між сторонами дозволяють забезпечити безпеку даних.

### 3.3.4 Методи розбиття даних

В моделі Віктора Телло може використовуватися будь-який метод розбиття даних. Автор пропонує два методи: метод розбиття на основі бісекції дерева (Bisection tree based partition) та метод розбиття на основі гістаграм (Histogram based partition)[13]. Останній буде описано детальніше.

Стратегія полягає в обробці якомога більшої кількості запитів в хмарі або на сервері ДТС без розшифровки даних. Така обробка зводить до мінімуму розрахунки на клієнтській машині. Методика починається з визначення атрибутів, які будуть використовуватися в усіх запитах, і ці атрибути будуть оброблятися деякими операціями для отримання певних секцій розбиття. Для отримання секцій розбиття, потрібно кожне значення будь-якого атрибута співставити з певним діапазоном. Тобто потрібно зіставити значення атрибутів з секціями розбиття таким чином, щоб ці секції розбиття охоплювали всі значення і не перетинались між собою.

Метод розбиття на основі гістограм (Histogram based partition).

Розбиття на основі гістограм - це метод відображення статистичної інформації, одним з видів якої є метод рівномірного розподілу по ширині. Для поділу даних можна використовувати метод рівної ширини, який ділить значення на частини однакової ширини. Цей метод віднімає мінімальне значення від максимального значення для атрибута, який буде розбитий, і ділить результати на бажану кількість частин. На рисунку 3.5 показані ідентифікатори присвоєні п'яти секціям атрибута. У таблиці 3.1 показаний результат поділу за секціями.

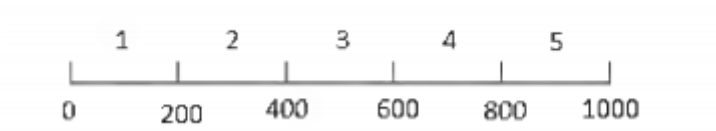


Рисунок 3.5 - Розбиття значень атрибута на 5 частин

Таблиця 3.1 - Результат розбиття на секції

| Діапазон значень | Секція |
|---|---|
| [0 - 200] | 1 |
| [200- 400] | 2 |
| [400 - 600] | 3 |
| [600 - 800] | 4 |
| [800-1000] | 5 |

Алгоритм розбиття за цим методом:

Алгоритм починається з читання таблиці бази даних, яку необхідно розбити і зберегти на хмарі або сервері, і в кінці алгоритм видає секції розбиття.

Вхідні дані:

Таблиця бази даних, в якій присутній атрибут, який треба розбити

Вихідні дані: секції розбиття

1). Ініціалізуйте Набір секцій розбиття = {}

2). Визначте Кількість секцій

3). Визначте Мінімальне та Максимальне значення атрибута, що потрібно розбити

4). Розмір діапазону = ((Максимальне значення - Мінімальне значення) / Кількість секцій)

5). Початок діапазону = Мінімальне значення; Номер секції = 0

6). Кінець діапазону = Початок діапазону + Розмір діапазону

7). Додайте Початок діапазону, Кінець діапазону та Номер секції до Набору секцій розбиття

8). Початок діапазону = Кінець діапазону; Номер секції = Номер секції + 1

9). Якщо (Номер секції < Кількість секцій) Перейдіть до кроку 6

10). Поверніть Набір секцій розбиття

11). Кінець

## 3.3 Розробка методу розбиття на основі частоти використання

В цьому підрозділі пропонується новий метод розбиття, який базується на частоті запитів певних значень атрибута.

Цей метод розбиття починається з отримання лог-файлу запитів до бази даних за деякий період часу, і за допомогою цього файлу генерується статистична матриця для всіх умов оператора WHERE, які будуть виконуватися на цих таблицях. Ця статистична матриця повинна бути створена для кожного атрибута даних, які необхідно розбити на секції. Далі зображено приклад лог-файлу:

*Where заробітня_плата>=400 та заробітня_плата<=750*
*Where заробітня_плата>=400 та заробітня_плата<=800*
*Where заробітня_плата>=400 та заробітня_плата<=750*
*Where заробітня_плата>=700 та заробітня_плата<=1200*
*Where заробітня_плата>=5000 та заробітня_плата<=7000*
*Where заробітня_плата>=1000 та заробітня_плата<=3000*
*Where заробітня_плата>=1200 та заробітня_плата<=2500*
*Where заробітня_плата>=400 та заробітня_плата<=4000*
*Where заробітня_плата>=400 та заробітня_плата<=800*
*Where заробітня_плата>=400 та заробітня_плата<=2500*
*Where заробітня_плата>=2500 та заробітня_плата<=5000*
*Where заробітня_плата>=2500 та заробітня_плата<=7000*

В таблиці 3.2 показана статистична матриця для атрибута заробітної плати в таблиці БД. Вона показує, скільки разів секція розбиття використовувалася в умовах оператору WHERE. Після створення статистичної матриці потрібно прибрати будь-яку секцію розбиття, яка зустрічається менше заданого порогу. Оптимальне значення порогу відрізняється для кожного набору запитів та технічних характеристик системи, але загальні рекомендації

наступні - чим менше значення порогу тим більше потрібно мати вільного місця на сервері та тим швидше клієнт отримає результат, вірно і навпаки.

Таблиця 3.2 - Статистична матриця

| Діапазон до / Діапазон з | 750 | 800 | 1200 | 2500 | 3000 | 4000 | 5000 | 7000 |
|---|---|---|---|---|---|---|---|---|
| 400 | 5 | 100 | 50 | 200 | 7 | 100 | 100 | 50 |
| 700 | | 4 | 15 | 3 | 1 | 0 | 0 | 5 |
| 800 | | | 200 | 100 | 4 | 200 | 100 | 50 |
| 1000 | | | 17 | 13 | 10 | 5 | 2 | 5 |
| 1200 | | | | 300 | 5 | 200 | 200 | 100 |
| 2500 | | | | | 5 | 300 | 200 | 100 |
| 4000 | | | | | | | 600 | 100 |
| 5000 | | | | | | | | 200 |

В таблиці 3.3 зображено результат усунення деяких секцій розбиття, оскільки вони зустрічались менше порогового значення.

Таблиця3.3 -Статистична матриця після усунення значень менше порогового (25)

| Діапазон до / Діапазон з | 800 | 1200 | 2500 | 4000 | 5000 | 7000 |
|---|---|---|---|---|---|---|
| 400 | 100 | 50 | 200 | 100 | 100 | 50 |
| 800 | | 200 | 100 | 200 | 100 | 50 |
| 1200 | | | 300 | 200 | 200 | 100 |
| 2500 | | | | 300 | 200 | 100 |
| 4000 | | | | | 600 | 100 |
| 5000 | | | | | | 200 |

Таблиця 3.4 зображує кількість записів в кожній секції розбиття, і за допомогою цієї таблиці можна продемонструвати, що отримання менших секцій розбиття може поліпшити швидкодію отримання записів з бази даних.

Таблиця 3.4 - Кількість записів по діапазонах

| Діапазон з | Діапазон по | Кількість записів |
|---|---|---|
| 400 | 800 | 20 |
| 800 | 1200 | 30 |
| 1200 | 2500 | 30 |
| 2500 | 4000 | 25 |
| 4000 | 5000 | 10 |
| 5000 | 7000 | 5 |

Наприклад, секція розбиття [800, 1200] має 200 запитів і 30 записів з таблиці, тому в цілому вона повинна приносити 200 * 30 = 6000 записів з таблиці БД. Однак, якщо ми виберемо секцію розбиття [400, 1200], то вона буде мати 20+30 = 50 записів і принести 250 * 50 = 12500 записів, коли користувач шукає значення від 800 до 1200. Таким чином, розбиття секції [400, 1200] на дві різні секції [400, 800] і [800, 1200] дозволить поліпшити показники. Чисте поліпшення становить 12500-6000 = 6500 записів. У таблиці 3.5 наведені підсумкові секції розбиття.

Таблиця 3.5 - Підсумкові секції розбиття

| Секція розбиття | Індекс |
|---|---|
| [400- 800] | 1 |
| [800 - 1200] | 2 |
| [1200- 2500] | 3 |
| [2500- 4000] | 4 |
| [4000- 5000] | 5 |
| [5000- 7000] | 6 |

Крім того, атрибути можуть мати дискретні дані, які можуть приймати тільки певні значення. Дискретні дані можуть бути числовими, як кількість студентів, і можуть бути категорійними, як чоловік або жінка чи менеджер або програміст. Наступний приклад пояснює, як індексувати і розбивати дискретні дані. Наприклад, розглянемо атрибут Посада, який має дискретні значення, і можливими значеннями для цього атрибута є Менеджер, Програміст, Архітектор БД, або Бухгалтер. Після вивчення лог-файлу для цього атрибута буде створена статистична матриця, як показано в таблиці 3.6. Наступним кроком видаляємо будь-яку секцію розбиття значення якої менше заданого порогу, який становить

Таблиця 3.6 зображує результат цього кроку, усуваючи деякі секції розбиття. Після цього потрібно зібрати всі усунуті секції і об'єднати їх в одну секцію, або кожну групу в окрему секцію залежно від загального числа частот для кожної групи. В таблиці 3.8 зображено результат розбиття на секції

70

Таблиця 3.6 - Статистична матриця для атрибуту Посада

| Посада | Менеджер | Програміст | Архітектор БД | Бухгалтер |
|---|---|---|---|---|
| Менеджер | 250 | 50 | 40 | 200 |
| Програміст | | 30 | 20 | 20 |
| Архітектор БД | | | 20 | 30 |
| Бухгалтер | | | | 300 |

Таблиця 3.7 - Статистична матрицяпісля усунення

| Посада | Менеджер | Бухгалтер | |
|---|---|---|---|
| Менеджер | 250 | | 200 |
| Бухгалтер | | | 300 |

Таблиця 3.8 - Результат розбиття на секції

| Секція розбиття | Індекс |
|---|---|
| Менеджер | 1 |
| Бухгалтер | 2 |
| Програміст & Архітектор БД | 3 |

Алгоритм розбиття за цим методом: Алгоритм починається з читання лог-файлу запитів до таблиці бази даних, яку необхідно розбити і зберегти на хмарі або сервері, і в кінці алгоритм видає секції розбиття.

Вхідні дані: Лог-файл

Вихідні дані: секції розбиття

1. Ініціалізуйте Набір секцій розбиття = {}
2. Прочитайте лог-файл, знайдіть всі команди WHERE для атрибуту, що потребує розбиття, розпарсіть значення та додайте їх до двовимірного масиву Статистична матриця, в якому значення початків діапазону знаходяться на місці рядків, а значення кінців діапазона знаходяться на місці стовпчика та на перетині рядка і стовпчика проставлено значення кількості запитів з таким діапазоном
3. Відсортуйте масив Статистична матриця за першим стовпцем і першим рядком у порядку зростання
4. Визначте поріг мінімально допустимого значення частоти
5. Отримайте масив Список_пар_діапазонів, який складається з пар виду { Початок_секції_розбиття, Кінець_ секції_розбиття } взявши попарно значення з нового масиву Статистична матриця після видалення секцій, значення яких менше за поріг
6. Ініціалізуйте Індекс = 0
7. Отримайте Поточний_діапазон = значення на місці Індекс в масиві Список_пар_діапазонів
8. Якщо (Поточний_діапазон £ Набір секцій розбиття) Додайте Поточний_діапазон в Набір секцій розбиття

9. Якщо (Індекс < Довжина Список_пар_діапазонів - 1) Установіть Індекс = Індекс + 1 та перейдіть до кроку 7
10. Додайте до Набір секцій розбиття усі інші діапазони, які ще не присутні в Набір секцій розбиття, та поєднайте їх в одну чи декілька секцій розбиття
11. Додайте випадкові чи порядкові номера до кожної секції розбиття з Набір секцій розбиття
12. Поверніть Набір секцій розбиття
13. Кінець

## 3.4 Схема структури системи безпеки

Після розглядання методів аудиту, контролю доступу та криптографічного захисту бази даних можна скласти загальну результуючу схему з використанням запропонованого мною методу розбиття даних на основі частоти використання, яка зображена на рисунку А.2.

## Висновки до розділу 3

В даному розділі був запропонований новий метод розбиття даних, описані технології, які потрібно використовувати в системі захисту хмарних баз даних та на їх основі побудована система захисту хмарних баз даних з використанням розробленого нами методу розбиття даних на основі частоти використання.

За допомогою запропонованої системи безпеки хмарних баз даних можна краще орієнтуватися в необхідних технологіях для побудови системи безпеки хмарних баз даних та будувати свою систему на основі запропонованої.

# ВИСНОВКИ

У даній роботі були проаналізовані базові поняття хмарних обчислень, моделі розгортання хмарних технологій, їх основні властивості та архітектура. Була виявлена актуальність їх використання майже в усіх сферах нашого життя. Провайдери хмарних послуг не завжди ставлять безпеку інформації їх користувачів на перше місце, що зумовлює появу вразливостей, які були розглянуті в другому розділі. Ми повинні будувати систему безпеки хмарних баз даних, беручи до уваги наявність різноманітних вразливостей.

Було проаналізовано і виявлено найкращі технологічні рішення для деяких з важливих підсистем, які складають систему безпеки хмарних баз даних, а саме:

- для аудиту - схема громадського аудиту третьої сторони;
- для контролю доступом - керування доступом на основі атрибутів;
- для криптографічного захисту БД- система Віктора Телло.

Запропоновано новий метод розбиття даних на основі частоти використання, який повинен давати приріст швидкодії при використанні визначених запитів.

Використовуючи зазаначені вище технології та новий метод розбиття даних, було побудовано загальну структуру системи безпеки хмарних баз даних.

Було знайдено набір технологій для гарантування безпеки в одній із послуг, яку включає в себе концепція хмарних обчислень, а саме в хмарних базах даних.

Результати роботи можуть бути використані для побудови системи захисту інформації, застосовної до хмарних баз даних, компаніями будь-яких розмірів, від маленьких стартапів до великих концернів.

# ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1 Miller R. Who Has the Most Web Servers? [Electronic resource] / R. Miller - Access mode: http://www.datacenterknowledge.com/archives/2009/05/14/whos-got-the-most-web-servers/ - 27.09.2020 .

2 Оплачко Е.С. Облачные технологии и их применение в задачах вычислительной биологии [Электронный ресурс] / Е.С. Оплачко., Д.М. Устинин., М.Н. Устинин - Режим доступа: http://www.matbio.org/2013/Oplachko_8_449.pdf - 28.09.2020 г.

3 PaaS, DBaaS, SaaS... Что все что значит? [Электронный ресурс] - Режим доступа: https://habr.com/ru/company/kingservers/blog/310022/ - 28.09.2020 г.

4 Wadiwala R. Cloud Database - DBaaS (Database as a Service) [Electronic resource] / R. Wadiwala - Access mode: https://labs.sogeti.com/cloud-database-dbaas-database-as-a-service/ - 29.09.2020

5 Demchenko Y. Defining inter-cloud architecture for interoperability and integration [Text] / Y. Demchenko, C. Ngo, M.X. Makkes, R. Strijkers, C. de Laat // Proceeding of the 3rd International Conference on Cloud Computing, GRIDs and Virtualization, Nice, France, July 22-27, 2012 y. - pp. 174-180

6 Malik N.A. Threat modeling in pervasive computing paradigm [Text] / N.A. Malik, M.Y. Javed, U. Mahmud // Proceedings of the Mobility and Security, New Technologies, Tangier, November 5-7, 2008 y. - pp. 1-5

7 McRee R. PTA: Practical threat analysis [Text] / R. McRee // Proceedings of the Information Systems Security Association, London, September 15 16, 2008 y. - pp. 37-40

8 Jehangir A. Securing inter-cluster communication in personal networks [Text] / A. Jehangir // Proceedings of the 4th Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services, Philadelphia, August 6-10, 2007 y. - pp. 1-6

9 Bertino E. L. Security for Web Services and Service-Oriented Architectures [Text] / E. L. Bertino // Proceedings of the 2th Annual International Conference on Information Security, New York, USA., September 2-7, 2012 y. - pp. 35-69

10 Soares L.F.B. Secure user authentication in cloud computing management interfaces [Text] / L.F.B. Soares // Proceedings of the IEEE 32nd International Performance Computing and Communications Conference, San Diego, CA, December 6-8, 2013 y. - pp. 1-2.

11 Lohman T. DDoS is cloud's security Achilles heel [Electronic resource] / T. Lohman - Access mode: http://www.computerworld.com.au/article/401127/ddos_cloud_security_ac hilles_heel/ - 05.10.2020 .

12 Hamza Y.A. Cloud computing security: Abuse and nefarious use of cloud computing [Text] / Y.A. Hamza - Int. J. Comput. Eng. Res, 2013 - 53 p.

13 Fuentes V.T. Enforcing database security on cloud using a trusted third party based model [Text] / V.T. Fuentes // 2438, Theses and Dissertations, ScholarWorks@UARK, 2017 y. - 50 p.