Харківський національний університет імені В.Н. Каразіна
Факультет комп'ютерних наук
Безпека інформаційних систем і технологій

«Допущено до захисту»
Зав.кафедрою БІСТ
Сватовський І.І._____
« » червня 2023р.

Пояснювальна записка
до кваліфікаційної роботи бакалавра
спеціальність: 125 Кібербезпека

на тему: «Аналіз та дослідження застосування криптографічних методів у хмарних сервісах»

оцінка « » Керівник к.т.н. Єсіна М.В.
(прізвище та ініціали/підпис)

Голова ЕК Рецензент к.т.н. Бобух В. А.
(прізвище та ініціали/підпис)

Лемешко О.В._____ Виконавець студентка групи КБ-42

Лазарєва Є. В
(прізвище та ініціали/підпис)

Харків – 2023

РЕФЕРАТ

Пояснювальна записка даної дипломної роботи містить чотири розділи, висновки, список джерел посилання та додаток. Всього в роботі 55 сторінок, 5 рисунків, 2 таблиці, 30 пунктів у списку використаних джерел та 1 додаток на двох сторінках.

Метою цієї дипломної роботи є проведення детального аналізу та дослідження застосування криптографічних методів у хмарних сервісах.

Об’єктом дослідження є криптографічні методи, які використовуються у хмарних сервісах, у тому числі методи постквatової криптографії.

Предметом дослідження дипломної роботи є аналіз та дослідження застосування криптографічних методів у хмарних сервісах.

У процесі виконання роботи було проведено огляд, аналіз та дослідження застосування криптографічних методів у хмарних сервісах, зокрема методів постквантової криптографії. Також були сформульовані моделі загроз та порушника.

У результаті дослідження найбільш популярних постачальників хмарних послуг було визначено основні криптографічні методи, що на практиці використовуються в хмарах, а саме симетричне та асиметричне шифрування, шифрування за методом envelope encryption та гешування.

Отримані результати є корисними для всіх, хто займається забезпеченням безпеки хмарних сервісів і розробкою нових криптографічних методів захисту. Робота дає змогу краще зрозуміти важливість криптографічного захисту в контексті хмарних технологій та надає базові знання про різні методи шифрування та їх використання в хмарних сервісах.

Ключові слова: КРИПТОГРАФІЯ, ПОСТКВАНТОВА КРИПТОГРАФІЯ, ХМАРНІ ТЕХНОЛОГІЇ, ШИФРУВАННЯ, ХМАРНІ СЕРВІСИ.

ABSTRACT

The explanatory note of this thesis contains four sections, conclusions, a list of reference sources and an appendix. In total, the work has 55 pages, 5 figures, 2 tables, 30 items in the list of used sources and 1 appendix on two pages.

The purpose of this thesis is to conduct a detailed analysis and study of the application of cryptographic methods in cloud services.

The object of the research is cryptographic methods used in cloud services, including methods of post-quatic cryptography.

The subject of the thesis research is the analysis and research of the application of cryptographic methods in cloud services.

In the course of the work, a review, analysis and study of the application of cryptographic methods in cloud services, in particular post-quantum cryptography methods, was carried out. Threat and offender models were also formulated.

As a result of the study of the most popular cloud service providers, the main cryptographic methods used in practice in clouds were determined, namely symmetric and asymmetric encryption, encryption using the envelope encryption method and hashing.

The obtained results are useful for everyone who is engaged in ensuring the security of cloud services and developing new cryptographic protection methods. The work enables a better understanding of the importance of cryptographic protection in the context of cloud technologies and provides basic knowledge about different encryption methods and their use in cloud services.

Keywords: CRYPTOGRAPHY, POST-QUANTUM CRYPTOGRAPHY, CLOUD TECHNOLOGIES, ENCRYPTION, CLOUD SERVICES.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ ТА СИМВОЛІВ

SaaS	-	Програмне забезпечення як послуга (Software as a Service)
PaaS	-	Платформа як послуга (Platform as a Service)
IaaS	-	Інфраструктура як послуга (Infrastructure as a Service)
AWS	-	Веб-сервіси Амазон (Amazon Web Services)
FaaS	-	Функція як послуга (Function as a Service)
IAM	-	Управління ідентифікацією та доступом (Infrastructure and Access Management)
ACL	-	Список прав доступу до об’єкта (Access Control List)
КШД	-	Ключ шифрування даних
КШК	-	Ключ шифрування ключа
TLS	-	Transport Layer Security
Amazon S3	-	Сервіс-сховище від Amazon (Amazon Simple Storage Service)
GCM	-	Лічильник з автентифікацією Галуа (Galois/Counter Mode)
SSE-S3	-	Ключі, що керуються Amazon S3
SSE-KMS	-	Ключі, що керуються KMS
KMS	-	Сервіс керування ключами (Key Management Service)
SSE-C	-	Ключі, надані клієнтом
CMK	-	Customer Master Key
HMAC	-	Код автентифікації повідомлень на основі гешування (Hash-based message authentication code)
KMI	-	Інфраструктура керування ключами (Key Management Infrastructure)
GCP	-	Google Cloud Platform
CSEK	-	Ключ шифрування, наданий клієнтом (Customer-Supplied Encryption Key)

ВСТУП

Все більше зростає популярність хмарних технологій як ефективного рішення для задоволення різноманітних потреб людства. При чому останнім часом дана технологія стала доступною не тільки для IT-спеціалістів, а і для звичайних користувачів. Тож, люди кожен день зіштовхуються із хмарними сервісами такими, як Gmail, Google Docs, інші сервіси від компанії Google, Hotmail від Microsoft та навіть Netflix. Вони дозволяють отримувати доступ до інформації з будь-якого пристрою, що має підключення до мережі Інтернет, а також переглядати й/або редагувати дані одночасно з іншими користувачами. До того ж для роботи немає потреби мати велику кількість вільної пам’яті на локальному комп’ютері, адже для зберігання даних використовуються віддалені сервери.

Разом із розвитком хмарних сервісів все більш актуальним стає питання їхнього захисту, адже все частіше з’являються нові загрози безпеки інформації. Також з’являються перспективні напрямки захисту хмарних технологій:

- управління ідентифікацією користувача за допомогою використання сучасних методів автентифікації;
- використання комплексних систем захисту для забезпечення безпеки серверів, маршрутизаторів та іншого обладнання від можливих загроз безпеки (несанкціонований доступ, крадіжка, природні загрози тощо);
- нові підходи до підбору та навчання персоналу;
- покращення безпеки застосунків шляхом підвищення їхньої стійкості до шкідливого програмного забезпечення;
- використання комбінацій методів автентифікації користувачів і криптографічних перетворень задля підвищення захищеності персональних даних;
- а також підвищення криптостійкості алгоритмів шифрування шляхом об’єднання методів, що існують.

Таким чином, актуальність даної роботи полягає в необхідності аналізу та дослідження застосування криптографічних методів у хмарних сервісах для забезпечення можливості подальшого їх використання на практиці та їхнього розвитку задля протидії різноманітним загрозам безпеки інформації.

Метою дослідження є аналіз та дослідження застосування криптографічних методів у хмарних сервісах.

Перед початком роботи було поставлено наступні завдання:

- Провести огляд, аналіз та отримати досвід використання хмарних сервісів;
- Провести огляд, аналіз та дослідження застосування криптографічних методів у хмарних сервісах;
- Сформулювати моделі загроз та порушника хмарних сервісів.

Об’єктом дослідження є криптографічні методи у хмарних сервісах.

Предмет дослідження: аналіз та дослідження застосування криптографічних методів у хмарних сервісах.

1 ХМАРНІ СЕРВІСИ. ВИЗНАЧЕННЯ ТА КЛАСИФІКАЦІЯ

1.1 Поняття хмарних сервісів

Відповідно до стандарту ISO/IEC 22123-1 хмарні сервіси визначаються як можливості, які надаються за допомогою хмарних обчислень з використанням певних інтерфейсів, а хмарні обчислення в свою чергу – це модель забезпечення мережевого доступу до масштабованого розподіленого пулу спільних фізичних або віртуальних ресурсів. При цьому ресурсами можуть бути сервери, мережі, операційні системи, програмне забезпечення тощо.

Переваги хмарних технологій [1]:

- Створюється можливість швидко отримати доступ до необхідної інформації з будь-якого комп’ютера, який має підключення до мережі Інтернет;
- Можливість переглядати та редагувати інформацію одночасно з іншими користувачами;
- Немає необхідності мати персональний комп’ютер з великою обчислювальною потужністю;
- Гнучкість та масштабованість, резервування та відновлення після збоїв;
- Висока обчислювальна потужність, швидкість обробки інформації та великий обсяг файлового сховища;
- Можливість централізовано керувати хмарною інфраструктурою, встановлювати певний рівень безпеки тощо.

Недоліки хмарних технологій:

- Необхідність постійного доступу до мережі Інтернет;
- Існує небезпека атак на сервер;
- Необхідно мати довіру до провайдера хмарних послуг (постачальники хмарних послуг відповідальні за забезпечення безпеки даних користувачів);
- Необхідно самостійно контролювати захист персональних даних (не рекомендується розміщувати конфіденційні дані на публічній хмарі);
- Побудова власної хмари потребує значних матеріальних ресурсів.

1.2 Моделі розгортання хмар

Існує наступні види хмар: суспільні (community cloud), публічні (public cloud), приватні (private cloud) та гібридні (hybrid cloud) [2].

Публічні хмари [3] – модель хмарної інфраструктури, де хмарні служби потенційно доступні до будь-якого клієнта хмарної служби. При чому користувачі не можуть управляти ресурсами хмари – відповідальність за це лежить на власнику даної хмари (сервіс-провайдери). До прикладів публічних хмар відносяться: Amazon Elastic Compute Cloud, Google Docs, Microsoft Office Web тощо.

На рисунку 1.1 зображено схематичне зображення публічної хмари.

Рисунок 1.1 – Публічна хмара

Переваги публічних хмар:

- Простота й ефективність використання;
- Економія на персоналі;
- Гнучкість;
- Здатність до масштабування;
- Надійність та відмовостійкість.

Недоліки публічних хмар:

- Контроль над інфраструктурою відбувається сторонньою компанією;
- Зручність використання залежить від швидкості та стабільності підключення до мережі Інтернет;

- Необхідно використовувати нові підходи захисту даних.

Приватні хмари [4, 5] – модель хмарної інфраструктури, де хмарні служби використовуються виключно одним клієнтом хмарної служби (в межах однієї організації). Приватні хмари можуть бути розташовані на території підприємства, якому вони належать, або можуть розміщуватися сторонньою компанією. Однак, незалежно від розташування інфраструктури, апаратне та програмне забезпечення утиліт приватної хмари доступне лише власнику, який працює у визначеній приватній мережі. В цілому, ідеальним варіантом є хмара, що розташовується на території певної організації та обслуговується і контролюється її співробітниками.

На рисунку 1.2 нааведено схематичне зображення приватної хмари, що розташовується на території організації (ліворуч) та віддалено (праворуч).

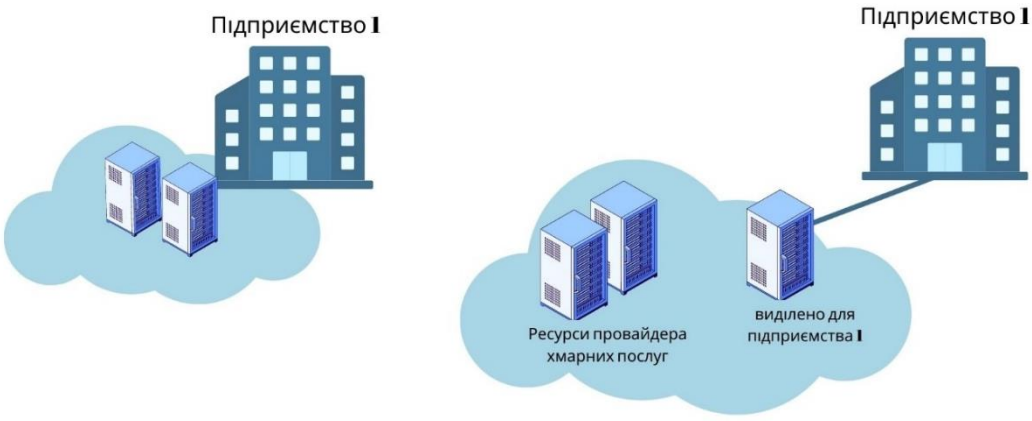

Рисунок 1.2 – Приватна хмара

Переваги приватної хмари:

- Швидкість роботи;
- Безпека;
- Використання ресурсів хмари тільки однією компанією;
- Повний контроль хмари.

Недоліки приватної хмари:

- Великі фінансові витрати на початку;
- Додаткові витрати на підтримку хмари;
- Необхідність залучення висококваліфікованих спеціалістів;
- Необхідність планування навантаження наперед.

Суспільні хмари [6] – це модель хмарної інфраструктури, що призначена для використання спільнотою користувачів, які мають спільні завдання та належать до одної спільноти. Наприклад, суспільними хмарами можуть бути компанії охорони здоров’я, уряди, деякі великі виробничі компанії.

Відмінністю суспільної хмари від приватної є те, що в моделі приватної хмари володіє та використовує інфраструктуру тільки одна організація, в той час як у моделі суспільної хмари інфраструктурою володіє одна організація, але використовують її ресурси спільно кілька компаній зі схожими характеристиками.

На рисунку 1.3 зображено схематичне зображення суспільної хмари.

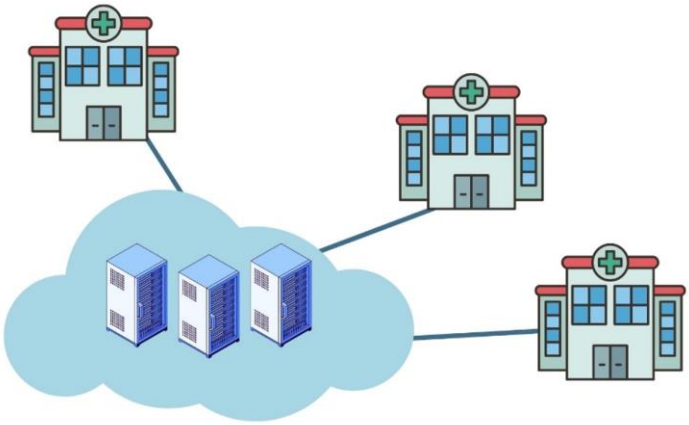

Рисунок 1.3 – Суспільна хмара

Переваги суспільних хмар:

- Гнучкість та масштабованість;
- Безпека;
- Надійність та доступність;
- Контрольований рівень конфіденційності.

Недоліки суспільних хмар:

- Складність;
- Безпека залежить від рівня знань IT-персоналу;
- Віддалений доступ може бути обмежений;
- Збільшення кількості IT-персоналу;
- Постійні витрати на технічне обслуговування;
- Повільна інтеграція/оновлення технології.

Гібридні хмари [7] – модель хмарної інфраструктури з використанням принаймні двох різних моделей хмарної інфраструктури (публічна, приватна, суспільна). Конфіденційні дані залишаються в приватній хмарі, де можна підтримувати високі стандарти безпеки. Операції, які не використовують конфіденційні дані, виконуються в загальнодоступній хмарі, де інфраструктура може масштабуватися відповідно до потреб, а витрати зменшуються. При цьому вся хмарна інфраструктура працює як єдина система, якою централізовано керує одна компанія.

Гібридні хмари добре підходять для виконання операцій з великими даними над неконфіденційними даними в загальнодоступній хмарі, зберігаючи захист конфіденційних даних у приватній хмарі [8]. Гібридні хмари також дають компаніям можливість запускати свої загальнодоступні додатки або платформи інтенсивної розробки в загальнодоступній частині хмари, при цьому їхні конфіденційні дані залишаються захищеними.

На рисунку 1.4 зображено схематичне зображення гібридної хмари.

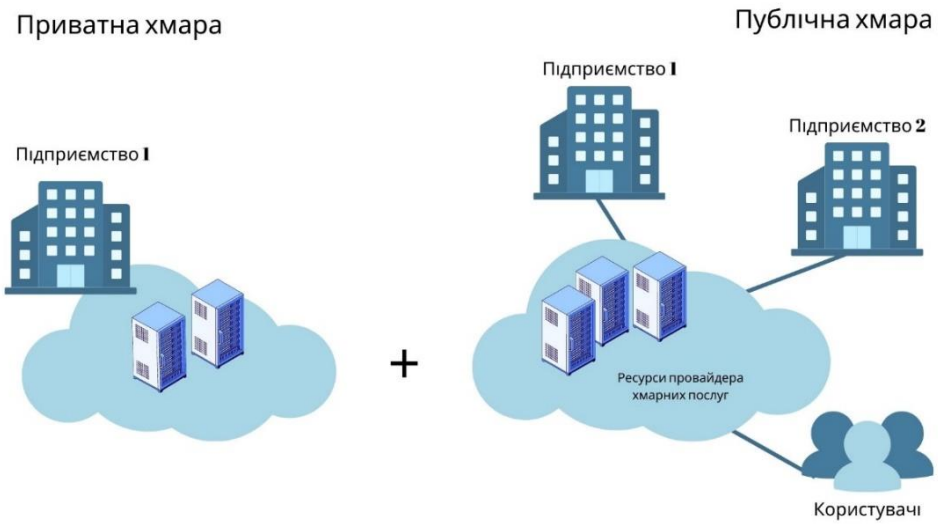

Рисунок 1.4 – Гібридна хмара

Переваги гібридної хмари:

- Зниження витрат (порівняно із приватними хмарами);
- Баланс контролю, продуктивності та масштабованості;
- Гнучкість.

Недоліки гібридної хмари:

- Складність реалізації та обслуговування;
- Безпека: існує ризик витоку даних між різними типами хмар, є потреба належного захисту від кібератак;
- Більша вартість у порівнянні із публічними хмарами.

1.3 Моделі надання послуг

Очевидно, що хмарні сервіси використовуються не лише для зберігання даних. Далі наведено перелік основних послуг [9], які можуть надаватися хмарними технологіями:

- Програмне забезпечення як послуга (SaaS)

Так програми розміщуються на хмарному сервері і користувачу не потрібно встановлювати її на свій ПК – він отримує доступ до неї через Інтернет.

Прикладами SaaS є Netflix, Microsoft Office 365, Cisco, Slack.

- Платформа як послуга (PaaS)

PaaS є середовищем розробки у хмарі з ресурсами, які дозволяють створювати хмарні застосунки. Провайдер контролює сервери та системи зберігання даних, в той час як розробники створюють власні програми та займаються їх підтримкою.

Прикладами PaaS є Heroku, Google App Engine, Microsoft Azure.

- Інфраструктура як послуга (IaaS)

Користувачам надається доступ до хмарних серверів та сховищ. Таким чином вони можуть створювати власні програми, проте заощаджувати на створенні власної фізичної інфраструктури.

Прикладами IaaS є DigitalOcean, Amazon (AWS), OpenStack.

- Функція як послуга (FaaS)

FaaS – це модель обчислювального хмарного сервісу, яка розбиває хмарні програми на менші компоненти, які запускаються лише у відповідь на певні події або запити. У цій моделі розробники завантажують код на сервер, а хмарний провайдер автоматично виділяє ресурси, необхідні для його виконання. FaaS може використовуватись для запуску невеликих фрагментів коду, які виконують

певну функцію (обробка даних, обчислення тощо), яка може викликатися з інших додатків або сервісів.

Прикладами FaaS є Azure Functions, AWS Lambda, Google Cloud Functions.

2 МОДЕЛІ ЗАГРОЗ ТА ПОРУШНИКА

Під час аналізу впровадження безпеки інформаційної системи необхідно враховувати загрози, яким може піддаватися система. Загрози використовують вразливі місця в системі для використання системних ресурсів або даних. Розробники систем розгортають механізми безпеки, щоб захистити вразливі місця та, таким чином, зменшити або усунути вплив певної загрози або класу загроз.

Для хмарних технологій додатково необхідно враховувати характерні для них особливості:

1) Користувачі можуть отримувати доступ до сервісів без потреби у взаємодії з провайдером хмарних сервісів.

2) Можливість отримати доступ до сервісів за допомогою тонких (надають лише інтерфейс для роботи користувача, проте використовують забезпечення сервера для обробки інформації) або товстих (використовують власні апаратні і програмні можливості) клієнтів [10].

3) Об’єднання обчислювальних ресурсів в одному місці для обслуговування споживачів з можливістю динамічного масштабування ресурсів залежно від потреб споживачів.

Ці особливості є перевагами використання хмарних сервісів для користувачів, проте разом із цим призводять до виникнення додаткових загроз безпеки інформації. Використовуючи хмарні сервіси, користувач не може використовувати додаткові засоби контролю доступу до інформації (наприклад, різноманітні технічні та організаційні заходи). Ще однією проблемою є те, що зловмисники можуть спричинити зменшення кількості доступних обчислювальних ресурсів шляхом інформаційних впливів на систему.

Таким чином, важливим є визначення моделей загроз, порушника та безпеки хмарних сервісів, про що і піде мова в цьому розділі. Відповідно до даних моделей можна сформувати вимоги до системи захисту хмарних сервісів.

2.1 Модель порушника

Під моделлю порушника розуміється абстрактний опис дій порушника, що відображає його можливості (практичні та теоретичні), знання, можливі дії тощо. При побудові моделі порушника необхідно враховувати інфраструктуру хмари, модель надання послуг, рівень контролю інформації тощо.

Для визначення моделі порушника в даній роботі виконується аналіз еталонної інфраструктури хмарних сервісів NIST [11] (рисунок 2.1).

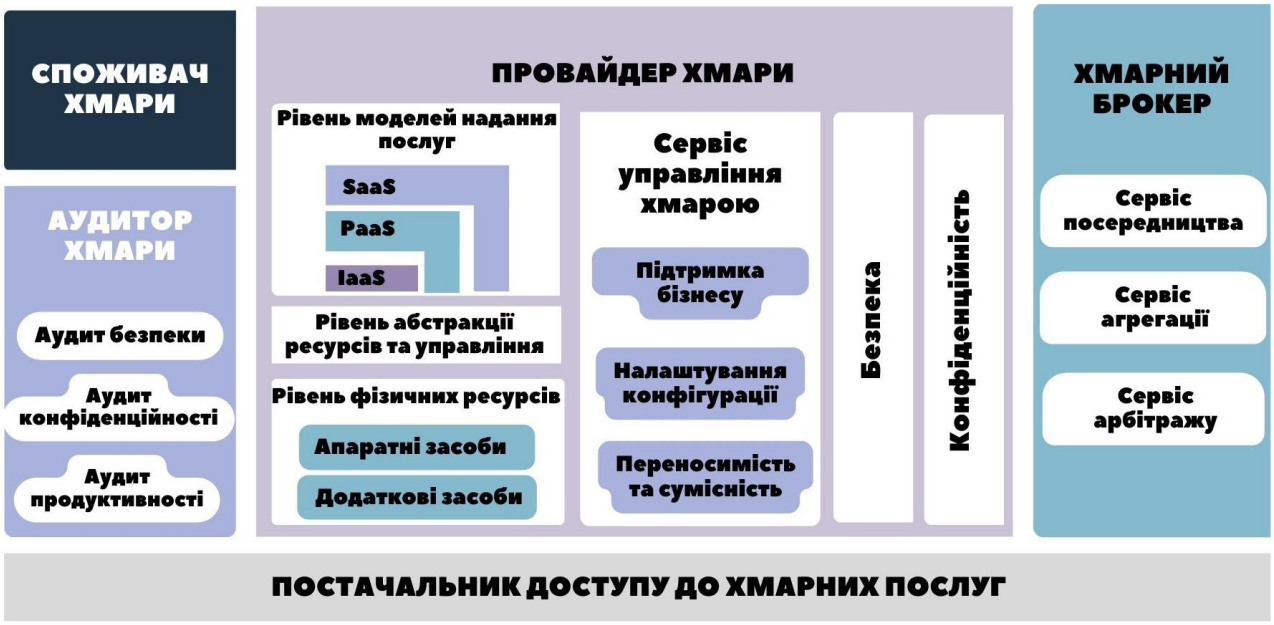

Рисунок 2.1 – Еталонна інфраструктура хмарних сервісів NIST

Побудована модель порушника включатиме наступні категорії:

- Тип порушника;
- Характер дій;
- Мотивація порушника;
- Навички порушника, його кваліфікація та ознайомлення з системо;
- Засоби та методи атаки;
- Шляхи здійснення атаки та елементи системи, які можуть бути атаковані.

Відносно хмари порушники можуть бути внутрішніми або зовнішніми [12]. Особливо небезпечними вважаються внутрішні порушники. До них відносяться співробітники провайдера та користувачі системи. Для їх визначення необхідно проаналізувати всі можливості несанкціонованого доступу

працівників провайдера і користувача до ресурсів та інформації, що обробляється в системі. До зовнішніх порушників відносяться сторонні особи, які не мають доступу до ресурсів хмарної інфраструктури. Для їх визначення необхідно проаналізувати можливі канали витоку інформації, а також вразливі місця системи.

Відносно рівня ознайомленості з системою внутрішніх порушників можна виділити наступні категорії порушників:

- Користувачі (клієнти постачальників хмарних послуг – можуть стати жертвами або сприяти зловмисникам. Наприклад, використовувати слабкі паролі, давати доступ до своїх файлів або папок ненадійним особам, або не оновлювати своє програмне забезпечення);
- Системні адміністратори (займаються конфігуруванням операційної системи та програмного забезпечення, а також мережевого обладнання та системи зберігання даних);
- Адміністратори безпеки (займаються конфігурацією засобів захисту, а також мають доступ до технологічної інформації, що обробляється в системі).

Характер дій порушника [13] може бути випадковий (порушник не навмисно порушив політику безпеки), пасивний (зловмисник навмисно порушив політику безпеки шляхом використання атрибутів доступу інших користувачів) та активний (порушник використовує будь-які доступні йому методи та засоби для порушення безпеки даних, наприклад, проводить віддалені атаки на інфраструктуру задля приведення до збоїв та відмов роботи сервісів, отримує фізичний доступ до серверів з метою перешкоджання їхньої роботи, викрадає чи пошкоджує носії інформації тощо).

Мотивацією порушника [12] може бути здобуток конфіденційної інформації, вимагання викупу, порушення цілісності даних (знищення або модифікація), знищення репутації компанії тощо.

Можливі цілі зловмисника:

- Отримання атрибутів доступу авторизованого користувача (шляхом крадіжок, купівлі, або за допомогою різних технічних засобів) з метою

ознайомлення з конфіденційною інформацією, модифікація або знищення даних, використання хмарних ресурсів у власних цілях.

- Отримання фізичного доступу до компонентів системи (обчислювальних та інформаційних ресурсів, носіїв інформації, телекомунікаційного обладнання тощо) задля нанесення збитків шляхом пошкодження матеріальних та/або інформаційних цінностей.

- Установка фізичних чи інших засобів технічної розвідки з метою знімання інформації.

- Установка програмних засобів з метою перевантаження систем і порушення доступності системи в цілому або її компонентів.

- Зміна режиму функціонування або порушення роботи ресурсів системи.

- Використання фізичних чи інших засобів для генерації хибних сигналів чи повідомлень.

- Отримання несанкціонованого доступу до ресурсів, програмного забезпечення тощо завдяки обходу системи управління доступом.

Навички порушника можуть варіюватися від дуже обмежених до високих. Їх можна умовно поділити на 4 рівні [13]:

1) Порушник здатен запускати фіксований набір програм, які реалізують визначені функції обробки даних.

2) Порушник може створювати і запускати власні програми та запроваджувати нові функції обробки даних.

3) Порушник може управляти функціонуванням ІТС хмари (впливати на базове ПЗ системи, а також склад і конфігурацію її устаткування).

4) Порушник володіє всім обсягом можливостей з проектування, реалізації і ремонту апаратних компонентів системи хмари та має можливість включати до системи власні засоби з певними функціями обробки даних.

Зловмисники можуть використовувати різні засоби для виконання атак на хмарні сервіси, такі як віддалений доступ до мережі, використання хмарних ресурсів для запуску зловмисного програмного забезпечення або виконання атак

на інші системи. Зловмисники можуть також використовувати соціальну інженерію для отримання доступу до важливої інформації або для здійснення атак на користувачів хмарних сервісів [14].

Засоби здійснення атак на хмарні сервіси:

1) Віддалений доступ: зловмисник використовує фішинг, соціальну інженерію або шкідливі програми, для отримання доступу до облікових даних користувачів хмари, що дозволяє входити до хмарних сервісів та здійснювати атаки на них.

2) Використання хмарних ресурсів для запуску зловмисного програмного забезпечення: зловмисник може використовувати віруси, черв'яки або троянські коні, для отримання доступу до конфіденційних даних або використовувати ресурси користувачів хмари без їхнього дозволу.

3) Атаки на віртуалізацію: використання вразливостей в системі віртуалізації хмарних сервісів, дозволяє зловмиснику отримати доступ до даних та ресурсів інших користувачів хмари, що використовують ту ж саму фізичну інфраструктуру.

4) Атаки на захист даних: зловмисник може використовувати перехоплення даних або злам паролів для отримання доступу до конфіденційних даних користувачів хмарних сервісів.

5) Атаки на служби моніторингу та управління: зловмисник використовує різні методи, наприклад, DDoS або злам паролів, для отримання доступу до систем моніторингу та управління хмарними сервісами, що дає йому можливість відключати, змінювати або знищувати дані користувачів.

6) Атаки на систему збереження даних: використання вразливостей в системі збереження даних хмарних сервісів, дозволяє йому знищувати або змінювати дані користувачів, виконувати шифрування або дешифрування даних без дозволу власника даних.

7) Атаки на сумісність та інтеграцію: використання вразливостей в системі сумісності та інтеграції хмарних сервісів, дозволяє зловмиснику виконувати атаки щодо користувачів, що використовують різні хмарні сервіси,

або зламувати користувацькі дані, що зберігаються на різних хмарних сервісах.

Шляхи реалізації атак:

- Технічні канали (побічні випромінювання та наводки, акустичні, оптичні, радіо- та інші канали);
- Канали спеціального впливу (формування полів і сигналів задля знищення системи захисту або викривлення інформації);
- Підключення до пристроїв та ліній зв’язку, застосування закладних пристроїв або вкорінення вірусів, приховування своєї ідентичності, використовуючи дані зареєстрованого користувача, тощо задля отримання несанкціонованого доступу.

Модель порушника описана в таблиці 2.1.

Таблиця 2.1 – Модель порушника у хмарних сервісах

Категорія осіб	Внутрішній
	Зовнішній
Рівень ознайомленості	Користувачі
	Системні адміністратори
	Адміністратори безпеки
Характер дій	Випадковий
	Активний
	Пасивний
Мета	Отримання атрибутів доступу авторизованого користувача з метою ознайомлення з конфіденційною інформацією, модифікація або знищення даних, використання хмарних ресурсів у власних цілях.
	Отримання фізичного доступу до компонентів системи задля нанесення збитків шляхом пошкодження матеріальних та/або інформаційних цінностей.

Продовження таблиці 2.1 – Модель порушника у хмарних сервісах

	Установка фізичних чи інших засобів технічної розвідки з метою знімання інформації.
	Установка програмних засобів з метою перевантаження систем і порушення доступності системи в цілому або її компонентів.
	Зміна режиму функціонування або порушення роботи ресурсів системи.
	Зміна режиму функціонування або порушення роботи ресурсів системи.
	Використання фізичних чи інших засобів для генерації хибних сигналів чи повідомлень.
	Отримання несанкціонованого доступу до ресурсів, програмного забезпечення тощо, завдяки обходу системи управління доступом.
Навички порушника	Порушник здатен запускати фіксований набір програм, які реалізують визначені функції обробки даних.
	Порушник може створювати і запускати власні програми та запроваджувати нові функції обробки даних.
	Порушник може управляти функціонуванням ІТС хмари (впливати на базове ПЗ системи, а також склад і конфігурацію її устаткування).
	Порушник володіє всім обсягом можливостей з проектування, реалізації і ремонту апаратних компонентів системи хмари та має можливість включати до системи власні засоби з певними функціями обробки даних.

Продовження таблиці 2.1– Модель порушника у хмарних сервісах

Методи і засоби здійснення атак	Соціальна інженерія
	Віддалений доступ до мережі
	Використання хмарних ресурсів для запуску зловмисного програмного забезпечення.
	Атаки на віртуалізацію.
	Атаки на захист даних
	Атаки на служби моніторингу та управління
	Атаки на систему збереження даних

2.2 Модель загроз

У моделі загроз мають бути наведені перелік загроз, їх мета, імовірність виконання, об’єкт, щодо якого реалізується загроза та способи захисту від них.

Список загроз з класифікацією цілей атак наведено у [15]. Модель визначає спуфінг, фальсифікацію, відмову, розкриття інформації, відмову в обслуговуванні, підвищення привілеїв і латеральний рух (для поступового просування мережею під час пошуку ключових даних і активів, які в кінцевому підсумку є ціллю їхніх атак) [16]. Розгортання хмарної інфраструктури піддається багатьом тим самим загрозам, яким піддається локальна обчислювальна мережа даних. Таким чином, застосування моделі загроз STRIDE-LM виявляє загрози, які є спільними як для локальних, так і для хмарних моделей спільних послуг. Однак використання загальнодоступних хмарних обчислювальних ресурсів створює додаткові, унікальні загрози для організації клієнта через такі характеристики, як відсутність контролю, низька або повна відсутність видимості операцій і незрілі вимоги відповідності.

1) Спуфінг (Spoofing) [16]

Загроза: несанкціонована зміна керування ідентифікацією та доступом (IAM).

Протидія: захист і моніторинг облікових даних користувача та доступу до хмарних ресурсів IAM.

Загроза: несанкціоновані зміни ролей адміністратора та прав доступу, які надають підвищені привілеї для зловмисних цілей.

Протидія: проведення моніторинг діяльності адміністратора (наприклад, за допомогою журналу адміністративного аудиту).

Загроза: скомпрометований доступ користувачів і адміністраторів через зламані облікові дані користувача (наприклад, спроби входу грубою силою).

Протидія: відстеження спроб входу користувачів, щоб шукати ознаки скомпрометованих облікових даних користувача.

2) Підробка даних (Tampering with data)

Загроза: неавторизоване вимкнення служб хмарного журналювання або неавторизоване видалення чи перезапис даних хмарного журналювання.

Протидія: захистити дані журналу.

Загроза: підробка або модифікація даних, зібраних і використаних для реагування на інциденти.

Протидія: проведення аудиту даних щодо персоналу, який відхилив сповіщення, а також коли сповіщення було відхилено та які анотації зроблено.

3) Відмова (Repudiation)

Загроза: неавторизований користувач стверджує, що не отримував доступ до облікового запису.

Протидія: хмарний моніторинг і журналювання.

4) Розкриття інформації (Information disclosure)

Загроза: неавторизована передача файлів до/з робочого середовища з/до хмарних середовищ.

Протидія: використання надійного шифрування даних і безпечних каналів зв’язку за допомогою протоколів, які забезпечують шифрування повідомлень. Також налаштування брандмауерів, які контролюють вхідний і вихідний трафік за допомогою груп безпеки та ACL.

Загроза: неавторизоване вихідне підключення до хмари.

Протидія: збір вихідних з’єднань у виробничих хмарних середовищах, які відповідають нормальній поведінці – слід переглянути будь-які відхилення від

нормальних вихідних з’єднань.

Загроза: вихідні/вхідні мережеві підключення до/від відомих шкідливих IP-адрес.

Протидія: відстеження вихідних з’єднань з адресами, які вважаються такими, що містять або розповсюджують зловмисне програмне забезпечення або пов’язані з активністю ботнету.

Загроза: неавторизований перегляд і редагування критичних файлів вручну.

Протидія: визначення права власності на дані та обмеження доступу для кожної ролі для конфіденційних файлів. Крім того, необхідно налаштувати брандмауери, які контролюють вхідний і вихідний трафік за допомогою груп безпеки та ACL.

Загроза: неавторизований мережевий доступ до сховищ даних (наприклад, сегментів AWS S3).

Протидія: використання автентифікації та авторизації для обмеження доступу до сховищ даних.

Загроза: робота з хмарними конфігураціями високого ризику.

Протидія: відстеження будь-яких небезпечних налаштувань.

Загроза: незахищене або неповне видалення даних із хмарного сховища може призвести до несанкціонованого викрадання даних.

Протидія: перевірка політики видалення хмарного постачальника.

5) Відмова в обслуговуванні (Denial of service)

Загроза: атака типу «відмова в обслуговуванні» (DoS) проти хмарної інфраструктури.

Протидія: блокування IP-адреси; використання білого списку.

6) Підвищення привілеїв (Elevation of privilege)

Загроза: підвищення привілеїв користувача.

Протидія: дотримання принципу найменших привілеїв та використання найменш привілейованих облікових записів служб для запуску процесів і доступу до ресурсів.

7) Латеральний рух (Lateral Movement)

Загроза: несанкціоновані спроби сканувати сусідній хост або спроба доступу до сусіднього хосту. Або несанкціоновані зміни групи безпеки мережі.

Протидія: використання журналів потоків хмарних віртуальних мереж, щоб перевірити трафік хмарної мережі.

Хмарна інфраструктура може мати вразливість до додаткових загроз [15], окрім визначених в STRIDE-LM. Такі загрози включають:

- порушення конфіденційності споживачів;
- несанкціоноване використання ресурсів для виконання неавторизованих завдань;
- скомпрометовані віртуальні машини/пристрої, які можуть використовуватися для запуску атак грубою силою на інші машини, створення спаму, або сканування відкритих портів та інших пристроїв в Інтернеті;
- проблеми з громадською інфраструктурою, що призводять до фізичного вторгнення в центр обробки даних, що призводить до крадіжки незашифрованих даних клієнтів;
- Блокування постачальника послуг;
- Неавторизоване виставлення рахунків зміни конфігурації;
- Потенційні недоліки в безпеці віртуалізації;
- Програми-вимагачі, хакерство, атаки на пристрої критичної інфраструктури.

Загалом середовище хмарних обчислень складається з тих самих або подібних обчислювальних ресурсів, що й традиційний центр обробки даних. Подібним чином середовище хмарних обчислень страждає від подібних загроз і становить інтерес для зловмисників з тих же причин, що й традиційний центр обробки даних. Але при застосуванні моделі загроз до хмарного середовища деякі ключові характерні відмінності відрізняють хмарне середовище від традиційного центру обробки даних.

3 ОГЛЯД КРИПТОГРАФІЧНИХ МЕТОДІВ У ХМАРНИХ СЕРВІСАХ

3.1 Загальний огляд хмарної криптографії

Хмарна криптографія – це сукупність методів та алгоритмів криптографії, які дозволяють забезпечувати безпеку даних в хмарних сервісах. Задачею хмарної криптографії є захист даних без затримки їхньої передачі.

Використання криптографії в хмарних сервісах надає певні переваги. Наприклад, унеможливлюється несанкціонований доступ до даних, які передаються, адже зловмисник не може розшифрувати їх, не маючи ключа. До того ж є можливість визначити, чи зберігається цілісність даних, тобто, чи не були отримані дані пошкоджені.

Хмарна криптографія базується на шифруванні даних [17]. Існують наступні підходи до використання криптографічних методів у хмарах:

- Попереднє шифрування (з хмарними сервісами синхронізуються попередньо зашифровані дані, що унеможливлює їх читання зловмисниками);
- Наскрізне шифрування (відправники та одержувачі надсилають повідомлення, при цьому лише вони можуть їх читати);
- Шифрування файлу (шифрування файлів відбувається, коли дані в стані спокою шифруються таким чином, що, якщо неавторизована особа спробує перехопити файл, вони не зможуть отримати доступ до даних, які в ньому зберігаються);
- Повне шифрування диска (перед збереженням файлів на зовнішньому диску, вони автоматично шифруються. Це ключовий метод захисту жорстких дисків на комп’ютерах).

В загальному криптографічні методи, що використовуються у хмарних сервісах, поділяються на методи з симетричним та з асиметричним ключем.

3.2 Криптографія з симетричним ключем

Симетрична криптографія [18] використовує один і той самий ключ для шифрування та розшифрування даних. При цьому забезпечується автентифікація та авторизація даних, оскільки розшифрувати дані можливо лише одним унікальним ключем, що був використаний для їхнього шифрування. Хоча симетричні методи є швидкими та ефективними, вони стикаються з проблемою безпеки ключа, оскільки він може бути скомпрометований.

Переваги симетричних методів:

- Швидкість;
- Ефективність.
- Недоліки:
- Існує проблема безпеки ключа, оскільки для шифрування та розшифрування даних використовується один і той самий ключ. Якщо ключ скомпрометовано, то всі зашифровані дані будуть під загрозою.
- Забезпечення безпеки обміну ключами між користувачами може бути проблематичним для великої кількості користувачів.

До симетричних методів відносяться: Advanced Encryption Standard (AES), Data Encryption Standard (DES), triple DES (3DES).

3.3 Криптографія з асиметричним ключем

Асиметрична криптографія [18] використовує два ключі – відкритий та особистий. При цьому, відкритий ключ використовується для шифрування повідомлення, а особистий ключ – для його розшифрування.

Переваги асиметричних методів:

- Більш безпечні, ніж методи з симетричним ключем;
- Корисно, коли задіяно багато кінцевих точок (є лише один власник особистого ключа, що значно знижує ймовірність його розкриття);
- Спрощується передача ключа (відкритий ключ не потрібно зберігати в таємниці).

Недоліки:

- Менша швидкість та ефективність у порівнянні з симетричними

методами;

- Через великий розмір ключів асиметричне шифрування проблематично використовувати в системах з великою кількістю даних, що сповільнює роботу сервера.

У хмарних сервісах використовуються: алгоритм цифрового підпису DSA (англ. – Digital Signature Algorithm), RSA (Rivest, Shamir, Adleman) та алгоритм Діффі-Хелмана.

3.4 Гібридне шифрування (Envelope Encryption)

Гібридне шифрування [19] поєднує у собі використання симетричного та асиметричного шифрування ключів. При такому підході для шифрування даних виконуються такі кроки:

- Одноразовий симетричний ключ, який називається ключем шифрування даних (КШД), генерується та використовується для шифрування частини даних;

- Для шифрування КШД генерується окремий симетричний або відкритий ключ, який називається ключем шифрування ключа (КШК). КШК може використовувати як симетричне, так і асиметричне шифрування;

- Зашифрований КШД додається або розміщується поряд із зашифрованим текстом і зберігається разом.

Для процесу розшифрування виконується зворотне:

- Програма отримує зашифрований текст і пов’язаний із ним КШД;

- Програма отримує КШК, якщо це симетричний ключ, або пов’язаний закритий ключ, якщо КШК є асиметричним відкритим ключем;

- Зашифрований КШД розшифровується за допомогою КШК, якщо це симетричний ключ, або за допомогою пов’язаного закритого ключа, якщо КШК є асиметричним відкритим ключем;

- Потім зашифрований текст розшифровується за допомогою КШД.

Шифрування за методом Envelope Encryption має ряд переваг:

- Простіший захист ключа даних – оскільки ключі зашифровані, їх можна зберігати разом із зашифрованими даними;

- Простіше керування ключами. Завдяки меншому набору ключів шифрування для керування можна виконувати ротацію або змінювати лише ключі КШК, а КШД залишити без зміни та повторно шифрувати свої дані;
- Поєднує в собі сильні сторони як симетричних, так і асиметричних методів шифрування з ключем.

3.5 Гешування

Гешування [18] використовується для трансформації будь-якого повідомлення у строку певного формату та фіксованої довжини. Використання геш-функцій дозволяє швидко перевірити ідентичність двох наборів даних, порівнюючи при цьому їхні геш-коди, замість порівняння всіх даних вручну. У сфері хмарних технологій гешування використовується для безпечного зберігання конфіденційної інформації, яка може використовуватись для аналізу даних або проведення обчислень, але не може бути розкрита.

Переваги:

- Порівняння геш-кодів виконується швидше, ніж зіставлення двох великих наборів даних.
- Геш-коди можна зберігати у вигляді індексів для забезпечення зручного зберігання та пошуку даних у базах даних.

Недоліки:

- Гешування неефективне, коли є велика кількість колізій;
- Для великого набору можливих ключів колізій практично неможливо уникнути;
- Геш-таблиці може бути складно реалізувати.

У хмарних сервісах використовують алгоритми SHA (Secure Hash Algorithm) та MD5 (Message-Digest Algorithm 5). SHA-256 та SHA-512, відносяться до криптографічно стійких алгоритмів гешування. Натомість MD5 та SHA-1 є менш стійкими алгоритмами.

3.6 Важливість хмарної криптографії

Криптографія – це один із основних засобів захисту, які можуть використовуватися для захисту даних, інтелектуальної власності та іншої

конфіденційної інформації.

Переваги хмарного шифрування включають наступне [20].

Безпека: шифрування забезпечує наскрізний захист конфіденційної інформації, коли вона перебуває в стані спокою або передається.

Відповідність вимогам: Положення та стандарти щодо конфіденційності та захисту даних, такі як FIPS і HIPPA вимагають забезпечити шифрування всіх конфіденційних даних клієнтів.

Цілісність: хоча зашифровані дані можуть бути модифіковані зловмисниками, таку активність легко виявити авторизованими користувачами.

Зниження ризику: якщо відбувається витік даних, проте вони були зашифровані, це значно знижує ризик як шкоди репутації, так і судових позовів або інших правових дій, пов’язаних із безпекою.

3.7 Використання постквантової криптографії у хмарних сервісах

Класичні методи шифрувань з відкритим ключем використовують складність обчислення дискретного алгоритму еліптичної кривої, факторизації або задачі дискретного логарифмування. Враховуючи потенційну здатність квантових комп’ютерів зламувати більшість таких шифрувань, доцільно розглядати можливість використання постквантової криптографії для забезпечення безпеки хмарних сервісів.

Наприклад, можна використовувати постквантовий протокол обміну ключами. Він додає постквантові шифри в протокол TLS (Transport Layer Security), забезпечуючи таким чином захист як від класичних, так і постквантових атак. Постквантовий протокол обміну ключами гарантує, що жодна зі сторін не зможе отримати значення ключа без додаткової інформації. Крім того, в квантових системах заборонено копіювання квантових станів, тому неможливо навіть отримати таку інформацію.

Квантова криптографія поділяється на 4 види [22]:

- Криптографія на основі коду (базується на розв’язанні невідомих кодів з виправленням помилок, які вважаються NP-складними. Прикладом є криптосистема, названа на честь Роберта Макеліса (McEliece). Вона дозволяє

кодувати біти даних під час передачі та отримання);

- Криптографія на основі решітки (використовує багатовимірні решітки, найчастіше базується на задачі пошуку найближчого вектора або пошуку найкоротшого вектора; прикладом є NTRU та Dilithium);
- Криптографія на основі гешу (прикладом є розширена схема підпису дерева Меркле (eXtended Merkle Signature Scheme, XMSS));
- Багатовимірна криптографія з відкритим ключем (ґрунтується на проблемі розв’язання випадкових наборів квадратних рівнянь над скінченними полями і використовує певні точки для шифрування та розшифрування; прикладами є Rainbow, GeMSS).

4 ЗАСТОСУВАННЯ КРИПТОГРАФІЧНИХ МЕТОДІВ У ХМАРНИХ СЕРВІСАХ

4.1 Веб-сервіси Amazon (AWS)

Amazon S3 (Amazon Simple Storage Service) [23] – це служба зберігання об’єктів в AWS, яка на сьогодні є найпоширенішою у світі. Як постачальник загальнодоступних хмар із найбільшою довговічністю, AWS пропонує послуги шифрування та керування ключами з найбільшою кількістю варіантів.

4.1.1 Методи шифрування

Amazon S3 підтримує шифрування як на стороні сервера, так і на стороні клієнта з кількома параметрами для кожного [24]. Користувачі мають можливість увімкнути шифрування на стороні сервера за замовчуванням для всіх об’єктів, завантажених на S3. Для шифрування як на стороні сервера, так і на стороні клієнта AWS використовує AES-256 із використанням лічильника з автентифікацією Галуа (GCM) для будь-яких операцій шифрування з симетричним ключем. GCM забезпечує автентифіковане шифрування, додаючи унікальний тег до зашифрованого тексту, який підтверджує, що зашифровані дані не були жодним чином підроблені. Шифрування за методом Envelope encryption [25] використовується для всіх параметрів на стороні клієнта та для всіх параметрів на стороні сервера, за винятком випадків, коли клієнт надає ключ шифрування.

Для шифрування на стороні сервера Amazon S3 підтримує три варіанти [24]:

- Ключі, що керуються Amazon S3 (SSE-S3);
- Ключі, що керуються AWS Key Management Service (KMS) (SSE-KMS);
- Ключі, що надаються клієнтом (SSE-C).

За допомогою SSE-S3 як ключ шифрування ключа (КШК), так і ключ шифрування даних (КШД) зберігаються та керуються службою S3. Усі функції керування ключами, включаючи періодичну ротацію ключів, виконуються

службою без введення даних від користувача. S3 робить це за допомогою служби керування ключами (KMS), керованої AWS.

Процес шифрування для SSE-S3 такий:

- Дані завантажуються в Amazon S3;
- Служба S3 генерує унікальний одноразовий ключ шифрування даних (КШД);
- Завантажені дані шифруються за допомогою КШД;
- Потім КШД шифрується за допомогою ключа шифрування ключа (КШК), який зберігається та керується службою S3;
- Зашифрований КШД зберігається як метадані разом із даними зашифрованого тексту, тоді як версія КШД у відкритому вигляді видаляється з пам’яті.

Процес розшифрування виглядає наступним чином:

- Amazon S3 отримує зашифрований КШД для запитуваного об’єкта та розшифровує його за допомогою пов’язаного КШК;
- S3 розшифровує об’єкт зашифрованого тексту за допомогою розшифрованого КШД, а потім видаляє ключ із пам’яті;
- Розшифрований об'єкт завантажується в запитуючий клієнт або програму.

Перш, ніж заглибитися в опцію SSE-KMS, важливо зазначити, що KMS використовує термін Customer Master Key (CMK) для опису того, що зазвичай називається ключем шифрування ключа (КШК). Подібним чином AWS використовує термін «ключ даних» (Data Key) для опису того, що зазвичай називається «ключ шифрування даних» (Data Encryption Key).

За допомогою SSE-KMS ключ даних шифрується або за допомогою CMK за замовчуванням, який автоматично створюється, коли користувач вирішує зашифрувати об’єкт S3 вперше, або за допомогою CMK, створеного користувачем. Використання CMK, створеного користувачем, забезпечує більшу гнучкість і контроль над CMK.

Процес шифрування для SSE-KMS такий:

- Дані завантажуються в Amazon S3;
- Служба S3 запитує як відкриту, так і зашифровану версії ключа даних у контексті CMK за замовчуванням або CMK, створеного користувачем;
- Служба управління ключами (KMS) AWS використовує CMK для створення нового унікального одноразового ключа даних і шифрує ключ за допомогою CMK;
- AWS KMS надсилає на S3 версії ключа даних у відкритому та зашифрованому вигляді;
- S3 використовує відкритий текстовий ключ даних для шифрування об’єкта та видаляє ключ із пам’яті;
- Зашифрований ключ даних зберігається в S3 як метадані разом із зашифрованим об’єктом.

Процес розшифрування виглядає наступним чином:

- Amazon S3 отримує зашифрований ключ даних для запитуваного об’єкта та надсилає його до AWS KMS;
- KMS розшифровує ключ даних за допомогою пов’язаного CMK і надсилає розшифрований ключ до S3;
- S3 розшифровує об’єкт за допомогою розшифрованого ключа даних, а потім видаляє ключ із пам’яті;
- Розшифрований об'єкт завантажується в запитуючий клієнт або програму.

SSE-C повністю покладає обов’язок керування ключами на користувача. Amazon S3 усе ще виконує процес шифрування та розшифрування, але користувач надає ключі шифрування, які мають бути симетричними ключами AES-256. Користувач надає ці ключі для кожної операції шифрування та розшифрування. AWS не зберігає фактичні ключі, але виконує операцію автентифікації повідомлень на основі гешування (HMAC) щодо ключів і зберігає отриманий геш. Геш HMAC – це, по суті, електронний підпис, який можна використовувати для перевірки автентичності ключів для майбутніх операцій без необхідності зберігати надані клієнтом ключі в AWS. Сам геш не можна

використовувати для розшифровки даних або відновлення втраченого ключа.

Процес шифрування для SSE-C такий:

- Дані завантажуються в Amazon S3 разом із ключем шифрування, наданим клієнтом;
- Дані шифруються службою S3;
- Створюється геш ключа шифрування, а сам ключ видаляється з пам’яті;
- Геш і зашифрований об’єкт зберігаються в S3.

Процес розшифрування виглядає наступним чином:

- Клієнт або програма запитує об’єкт і надає симетричний ключ, який використовується для шифрування;
- Amazon S3 перевіряє симетричний ключ, використовуючи геш, створений під час шифрування;
- S3 розшифровує об’єкт за допомогою симетричного ключа, а потім видаляє ключ із пам’яті;
- Розшифрований об'єкт завантажується в запитуючий клієнт або програму.

Переходячи до шифрування на стороні клієнта, Amazon S3 підтримує два варіанти:

- Використання Customer Master Key (CMK) під керуванням KMS;
- Використання CMK на стороні клієнта.

Щоб допомогти користувачам, які обирають варіант шифрування на стороні клієнта, AWS надає клієнт шифрування Amazon S3 [24], який вбудовано в AWS SDK для ряду мов, зокрема Java, Go тощо. Клієнт шифрування виконує всі операції шифрування та розшифрування даних за допомогою симетричного шифрування AES-256 GCM із головним ключем (AWS-еквівалент ключа шифрування ключа), згенерованим у KMS або наданим користувачем.

За допомогою шифрування на стороні клієнта, яке використовує AWS KMS, користувач створює CMK у KMS і отримує ідентифікатор CMK, який є логічним представленням фактичного CMK. Користувачі надають ідентифікатор CMK, коли запитують об’єкт, гарантуючи, що фактичний CMK ніколи не покине

KMS.

Процес шифрування виглядає наступним чином:

- Дані передаються до клієнта шифрування AWS;
- Клієнт шифрування запитує ключ даних у KMS, використовуючи вказаний ідентифікатор CMK;
- KMS використовує пов’язаний CMK для створення нового унікального одноразового ключа даних;
- KMS передає клієнту шифрування версії ключа даних у відкритому та зашифрованому вигляді;
- Клієнт шифрування шифрує дані за допомогою відкритого текстового ключа даних, а потім видаляє ключ із пам’яті;
- Клієнт шифрування повертає зашифроване повідомлення, яке містить зашифрований ключ даних разом із зашифрованими даними;
- Зашифроване повідомлення завантажується в S3.

Процес розшифрування виглядає наступним чином:

- Зашифровані дані у вигляді шифрувального повідомлення завантажуються в клієнт або програму користувача;
- Користувач передає зашифроване повідомлення клієнту шифрування AWS;
- Клієнт шифрування отримує зашифрований ключ даних із повідомлення шифрування та надсилає зашифрований ключ даних до KMS;
- KMS використовує пов’язаний CMK для розшифровки ключа даних зашифрованого тексту;
- KMS передає відкритий текстовий ключ даних клієнту шифрування;
- Клієнт шифрування розшифровує дані за допомогою відкритого текстового ключа даних, а потім видаляє ключ із пам’яті;
- Клієнт шифрування повертає розшифровані дані.

Використовуючи цей варіант, користувачі можуть зашифрувати дані перед тим, як вони покинуть центр обробки даних і завантажаться в S3. Однак вони можуть обійтися без відповідальності за підтримку криптографічної системи або

власної інфраструктури керування ключами (KMI).

Остаточний варіант шифрування на стороні клієнта вимагає від користувача надання головного ключа (КШК), який використовується для шифрування будь-яких ключів даних. Цей головний ключ може бути симетричним або асиметричним відкритим ключем. Головний ключ користувача надається клієнту шифрування AWS, який виконує всі операції шифрування та розшифрування даних. Користувач несе повну відповідальність за зберігання та керування головним ключем.

Процес шифрування виглядає наступним чином:

- Дані разом із головним ключем передаються до клієнта шифрування AWS;
- Клієнт шифрування створює унікальний одноразовий ключ даних і шифрує дані за допомогою цього ключа;
- Клієнт шифрування шифрує відкритий текстовий ключ даних за допомогою наданого головного ключа, а потім видаляє відкритий текстовий ключ із пам’яті;
- Клієнт шифрування повертає зашифроване повідомлення, яке містить зашифровані дані, ключ зашифрованих даних і метадані, які пов’язують цей ключ даних із головним ключем;
- Зашифроване повідомлення завантажується в S3.

Процес розшифрування виглядає наступним чином:

- Зашифровані дані у вигляді шифрувального повідомлення завантажуються в клієнт або програму користувача з S3;
- Користувач передає зашифроване повідомлення клієнту шифрування AWS;
- Клієнт шифрування отримує зашифрований ключ даних із повідомлення шифрування та знаходить відповідний головний ключ, використовуючи метадані в зашифрованому повідомленні;
- Клієнт шифрування використовує пов’язаний симетричний головний ключ або асиметричний особистий головний ключ для розшифровки ключа

даних зашифрованого тексту;

- Клієнт шифрування розшифровує дані за допомогою відкритого текстового ключа даних, а потім видаляє ключ із пам’яті;
- Клієнт шифрування повертає розшифровані дані.

Користувачі також можуть вибрати шифрування даних перед завантаженням на S3 за допомогою власної криптографічної інфраструктури та інфраструктури керування ключами без клієнта шифрування AWS. Процес шифрування прозорий для S3, і зашифровані дані зберігаються так само, як і незашифровані дані.

4.1.2 Управління ключами

З точки зору керування ключами, AWS пропонує три варіанти [24]:

- Ключі, що керуються Amazon S3;
- CloudHSM для клієнтів із власним програмним забезпеченням для керування ключами;
- Служба керування ключами AWS (KMS).

Як було сказано раніше, сервіс Amazon S3 може зберігати та керувати ключами шифрування від імені користувача, включаючи періодичну ротацію ключів. За допомогою цієї опції користувачі перекладають усі обов’язки з управління ключами на AWS.

CloudHSM пропонує сховище ключів для користувачів, які не хочуть керувати власною системою зберігання ключів, проте вже мають із власним програмне забезпечення для керування ключами. CloudHSM надає клієнтам спеціальний апаратний пристрій, захищений від несанкціонованого доступу, який працює в центрі обробки даних Amazon. Пристрій CloudHSM інтегрується з програмним забезпеченням для керування ключами, яке користувачі можуть запускати локально або в AWS. При цьому користувач несе відповідальність за повний життєвий цикл ключів, які зберігаються в CloudHSM.

Служба керування ключами AWS – це повністю керована послуга, яка, по суті, надається користувачам як програмне забезпечення для керування ключами. Користувачі можуть взаємодіяти з KMS через веб-інтерфейс, який дає

їм можливість керувати повним життєвим циклом ключів шифрування, що зберігаються в KMS.

4.2 Microsoft Azure

Azure Blob Storage [26] – це пропозиція служб зберігання об’єктів Microsoft Azure.

4.2.1 Методи шифрування

Azure підтримує шифрування як на стороні сервера, так і на стороні клієнта, при цьому користувачі мають можливість увімкнути шифрування на стороні сервера за замовчуванням для всіх завантажених об’єктів [27].

В Azure шифрування на стороні сервера називається Storage Service Encryption. Azure використовує Envelope Encryption за допомогою симетричних ключів AES-256 для шифрування даних або вмісту (Microsoft використовує термін Content Encryption Key замість Data Encryption Key) і підтримує використання симетричних або асиметричних ключів для ключа шифрування ключа (КШК), залежно від того, хто генерує та керує ключами.

Storage Service Encryption підтримує використання КШК, який:

- Керується самою службою зберігання за допомогою внутрішньої інфраструктури керування ключами Microsoft;
- Клієнт керує та зберігає в Key Vault, пропозицію служби керування ключами Azure.

Процес шифрування для Storage Service Encryption виглядає так:

- Дані завантажуються в Azure Blob Storage;
- Azure Blob Storage викликає криптографічну бібліотеку для створення унікального одноразового ключа шифрування вмісту (КШВ);
- Завантажені дані шифруються за допомогою КШВ;
- Потім КШВ шифрується за допомогою загальнодоступного КШК RSA, який зберігається та керується службою зберігання або зберігається в Azure Key Vault;
- Зашифрований КШВ зберігається як метадані разом із даними зашифрованого тексту, тоді як версія КШВ у відкритому вигляді видаляється з

пам’яті.

Процес розшифрування виглядає наступним чином:

- Коли запитуються дані, Azure Blob Storage отримує зашифрований КШД і надсилає його до внутрішньої служби керування ключами служб зберігання або до Azure Key Vault;
- КШВ розшифровується за допомогою закритого ключа, пов’язаного з КШК, і надсилається назад до Azure Blob Storage;
- Дані розшифровуються за допомогою відкритого тексту КШВ;
- Azure Blob Storage відкидає КШВ і надсилає розшифровані дані клієнту, який запитав дані.

Для шифрування на стороні клієнта Azure надає клієнтську бібліотеку сховища, написану для Java, .NET і Python, яка інтегрується з шифруванням Azure. За допомогою цього параметра користувачі можуть зберігати та керувати власними КШК або використовувати Azure Key Vault.

Процес шифрування такий:

- Клієнтська бібліотека сховища генерує унікальний одноразовий ключ шифрування вмісту (КШВ);
- Дані шифруються за допомогою КШВ;
- Клієнт сховища викликає алгоритм упаковки ключів, викликаючи КШК, який зберігається у користувача, або в Azure Key Vault (КШК може бути симетричним або асиметричним ключем);
- КШВ шифрується за допомогою КШК;
- Зашифрований КШВ зберігається як метадані разом із даними зашифрованого тексту, тоді як версія КШВ у відкритому вигляді видаляється з пам’яті;
- Зашифровані дані завантажуються та зберігаються в Azure Blob Storage.

Процес розшифрування виглядає наступним чином:

- Зашифровані дані витягуються з Azure Blob Storage;
- Клієнт сховища викликає алгоритм розгортання ключа, який викликає КШК, який або зберігається користувачем, або зберігається в Azure Key Vault;

- Зашифрований КШВ розшифровується за допомогою КШК;
- Дані розшифровуються за допомогою відкритого тексту КШВ, який потім видаляється з пам’яті.

Користувачі також можуть вибрати шифрування даних перед завантаженням в Azure за допомогою власної криптографічної інфраструктури та інфраструктури керування ключами без клієнтської бібліотеки зберігання. Процес шифрування прозорий для Azure Blob Storage, і зашифровані дані зберігаються так само, як і незашифровані дані.

4.2.2 Управління ключами

Для шифрування на стороні сервера ключами керують за допомогою одного з двох варіантів [27]:

- Усі ключі генеруються та зберігаються самою службою Azure Blob Storage. Корпорація Майкрософт забезпечує зберігання та керування ключами без участі клієнтів;
- КШВ генеруються та зберігаються в Azure Key Vault. КШК зберігаються в Azure Key Vault, але ними керує користувач. КШК можна створити в Key Vault або імпортувати в Key Vault.

Для шифрування на стороні клієнта ключами можна керувати за допомогою одного з трьох варіантів:

- КШВ генеруються клієнтською бібліотекою сховища Azure. КШК зберігаються в Azure Key Vault, але ними керує користувач;
- КШВ генеруються клієнтською бібліотекою сховища Azure. КШК генеруються, зберігаються та керуються користувачем за допомогою власної інфраструктури керування ключами;
- І КШВ, і КШК генеруються, зберігаються та керуються користувачем за допомогою власної криптографічної системи. Azure Blob Storage не знає, що зберігає зашифровані дані.

Шифрування на стороні сервера за допомогою ключів, керованих сервісом, можна ввімкнути за замовчуванням для сховища Blob Azure. Це хороший варіант для користувачів, які не мають власної інфраструктури керування ключами

локально або в Azure та не хочуть брати на себе відповідальність за управління ключами.

Azure Key Vault – це служба, яка надає клієнтам Azure інфраструктуру керування ключами з підтримкою HSM і може бути інтегрована як із шифруванням на стороні сервера, так і на стороні клієнта. Користувачі можуть генерувати ключі через Azure Key Vault або імпортувати їх у Key Vault. Відповідальність за керування життєвим циклом ключів лягає на користувача, який використовує інструменти Azure Key Vault.

4.3 Google Cloud Platform

Google Cloud Storage [28] – це служба зберігання об’єктів Google Cloud Platform (GCP). Ймовірно, через новизну сервісу Google Cloud Storage має найменше можливостей для шифрування даних і керування ключами [29].

4.3.1 Методи шифрування

Google Cloud Storage за замовчуванням виконує шифрування на стороні сервера для всіх завантажених об’єктів. Усі дані розбиваються на частини, розмір яких може досягати кількох гігабайтів. Використовуючи шифрування за методом Envelope Encryption [30], кожна частина даних шифрується за допомогою унікального ключа шифрування даних (КШД), який також шифрується за допомогою ключа шифрування ключа (КШК), а зашифрована версія КШД потім зберігається разом із зашифрованими даними. Потім зашифровані фрагменти даних розподіляються між системами зберігання Google. І КШК, і КШД використовують симетричний AES-256 із шифром у режимі лічильника Галуа.

Google Cloud Storage підтримує шифрування на стороні сервера з двома варіантами:

- Ключі генеруються та зберігаються в KMS Google;
- Ключ шифрування, надається клієнтом.

Із шифруванням на стороні сервера з використанням КШК, які зберігаються та керуються внутрішньою KMS Google (або надаються клієнтом). Процес шифрування із використанням ключів, керованих KMS, такий:

- Дані розбиваються на кілька фрагментів після завантаження в Google

Cloud;

- Система Google Cloud Storage викликає загальну криптографічну бібліотеку CrunchyCrypt, що підтримується Google, для створення унікального одноразового КШД;
- Кожна частина даних шифрується за допомогою КШД;
- Потім система зберігання надсилає КШД до служби керування ключами Google (KMS) для шифрування за допомогою пов'язаного з цією системою ключа шифрування ключа (КШК);
- Зашифрований КШД надсилається та зберігається разом із фрагментом шифрованого тексту, який він зашифрував, у Google Cloud Storage, тоді як версія КШД у відкритому вигляді видаляється з пам'яті.

Процес розшифрування виглядає наступним чином:

- Коли надходять запити на дані, Google Cloud Storage визначає фрагменти, у яких зберігаються дані, і місце розташування блоків, і отримує їх;
- Для кожного блоку даних система зберігання отримує зашифрований КШД і надсилає його в KMS Google для розшифрування;
- KMS надсилає розшифрований КШД до системи зберігання, де він використовується для розшифровки даних;
- Система зберігання відкидає КШД і надсилає розшифровані дані клієнту, який запитав дані.

За допомогою параметра ключа шифрування, наданого клієнтом (Customer-Supplied Encryption Key або CSEK), користувачі повинні створити власний симетричний ключ AES-256 і надати його в Google Cloud Storage для операцій шифрування/розшифрування. CSEK зберігається лише в пам'яті системи зберігання й ніколи не зберігається на жодному пристрої Google Cloud.

Процес шифрування виглядає наступним чином:

- CSEK надається в Google Cloud Storage разом із завантаженням даних;
- Дані розбиваються на кілька фрагментів підфайлів;
- Система Google Cloud Storage викликає загальну криптографічну бібліотеку CrunchyCrypt, для створення унікального одноразового КШД;

- Кожна частина даних шифрується за допомогою КШД;
- Потім система зберігання використовує CSEK як КШК і шифрує КШД;
- Зашифрований КШД надсилається та зберігається разом із фрагментом шифрованого тексту, який він зашифрував, у Google Cloud Storage, тоді як версія КШД у відкритому вигляді видаляється з пам’яті;
- Ключ шифрування, наданий клієнтом, гешується, а потім видаляється із системи зберігання. Криптографічний геш використовується для перевірки майбутніх запитів, але не може використовуватися для розшифрування даних або реконструкції ключа.

Процес розшифрування виглядає наступним чином:

- Клієнт або програма запитує дані з Google Cloud Storage, надаючи CSEK;
- Google Cloud Storage визначає блоки, у яких зберігаються дані, і де вони знаходяться, та отримує їх;
- Для кожного блоку даних система зберігання отримує зашифрований КШД і розшифровує його за допомогою CSEK;
- Система зберігання відкидає КШД і надсилає розшифровані дані клієнту або програмі, яка запитала ці дані.

Оскільки служба KMS від Google не задіяна, користувачі несуть відповідальність не лише за генерацію ключів, але й за керування ними.

Google Cloud дозволяє шифрувати на стороні клієнта, але наразі не пропонує жодної конкретної інтеграції такої як, наприклад, бібліотеки на стороні клієнта для генерації КШД. Користувач несе відповідальність за створення ключів шифрування та власне шифрування даних перед завантаженням їх у Google Cloud Storage. Процес шифрування є прозорим для Google Cloud Storage, зашифровані дані зберігаються так само, як і незашифровані, тобто зашифровані на стороні клієнта дані будуть фактично знову зашифровані службою.

4.3.2 Управління ключами

Шифрування Google Cloud Storage за замовчуванням використовує внутрішню службу керування ключами Google [29]. Клієнтські КШК

генеруються та централізовано зберігаються у внутрішній KMS Google. KMS захищено за допомогою ієрархії ключів шифрування.

- Кожен КШК зберігається в KMS Google, що працює на кількох машинах у різних центрах обробки даних по всьому світу. Ці КШК зашифровано головним ключем KMS за допомогою AES-256.
- Головний ключ KMS зберігається в окремій системі під назвою Root KMS, яка розподіляється між кількома меншими спеціальними машинами. Головний ключ KMS також зашифровано за допомогою AES-256 за допомогою головного ключа Root KMS.
- Кореневий головний ключ KMS зберігається в системі під назвою Root KMS Master Key Distributor, яка реплікує ключ глобально. Ця система розповсюджувача ключів зберігає ключі в оперативній пам’яті та працює на тій самій машині, на якій працює кореневий KMS.
- Голоаний ключ KMS також створює резервну копію для захисту апаратних пристроїв, які зберігаються у фізичних сейфах. Ці сейфи зберігаються окремо в добре охоронюваних приміщеннях, доступ до яких мають лише кілька співробітників Google.

4.4 Порівняння розглянутих сервісів

У таблиці 4.1 наведено порівняння шифрування в службах зберігання об’єктів: Amazon S3, Azure Blob Storage, Google Cloud Storage. Видно, що рівень безпеки та різноманітність методів шифрування, що використовуються, залежить від віку постачальника хмарних послуг.

Таблиця 4.1 – Порівняння шифрування в службах зберігання об’єктів

	Amazon S3	Azure Blob Storage	Google Cloud Storage
Шифрування			
Шифрування на боці сервера	Так	Так	Так

Продовження таблиці 4.1 – Порівняння шифрування в хмарних сервісах

Шифрування на боці клієнта	Так	Так	Дозволено, проте клієнт шифрування не надається
Симетричне шифрування	AES-256, GCM; Для шифрування даних та ключа з використанням SSE-S3, SSE-KMS; Для шифрування даних з використанням SSE-C	AES-256; Для шифрування даних; Для шифрування ключа на боці клієнта.	AES-256, GCM; Для шифрування ключа та даних.
Асиметричне шифрування	RSA; Для шифрування ключа на стороні клієнта.	RSA; Для шифрування ключа на боці сервера та клієнта.	Може використовуватися для шифрування на боці клієнта.
Envelope encryption	Так, для всіх опцій, окрім SSE-C.	Так, для всіх опцій.	Так
Керування ключами			
Зберігання та керування користувачами	Так, для SSE-C та шифрування на стороні клієнта з використанням головного ключа клієнта.	Так, для шифрування на боці клієнта.	Так, для шифрування на боці сервера з використанням КШК, що надається користувачем. Для шифрування на боці клієнта, але не

Продовження таблиці 4.1 – Порівняння шифрування в хмарних сервісах

			надаються інтеграції для цього.
Зберігання провайдером та керування користувачами (з використанням власної інфраструктури управління ключами)	Так. Для SSE-C та шифрування на боці клієнта з використанням головного ключа клієнта.	Ні	-
Зберігання провайдером та керування користувачами (з використанням хмарних сервісів управління ключами)	Так, для SSE-KMS та шифрування на боці клієнта з використанням CMK, що керується KMS.	Так, для шифрування на боці сервера з використанням ключів, що керуються користувачами. Та для шифрування на боці клієнта.	-
Зберігання та керування провайдером	Так, для SSE-S3	Так, для шифрування на боці сервера з використанням ключів, що керуються сервісом.	Метод шифрування за замовченням.

ВИСНОВКИ

Отже, хмарні сервіси стають все більш популярними, але разом із цим виникають нові загрози безпеці даних. Важливо враховувати, що для забезпечення безпеки хмари необхідно застосовувати комплексний підхід. Одним із аспектів є використання різноманітних методів хмарної криптографії: симетричних, асиметричних, Envelope Encryption, гешування тощо.

Також варто зазначити, що в майбутньому є сенс детальніше досліджувати постквантові методи криптографії. У разі появи квантових комп'ютерів, які мають потенціал зламати більшість існуючих методів криптографії, постквантові методи зможуть забезпечити достатній рівень захисту.

В ході виконання роботи було досліджено методи шифрування, які використовують три найбільші провайдера хмарних сервісів: Amazon Web Services, Azure та Google Cloud. Цікавим є те, що рівень безпеки та кількість методів шифрування, що використовуються, залежить від віку провайдера. Таким чином, найбільше методів шифрування використовує Amazon Web Services, а Google Cloud має найменше можливостей для шифрування даних та керування ключами.

Таким чином, визначено, що провідні сучасні провайдери хмарних послуг використовують поєднання симетричних та асиметричних методів шифрування із Envelope Encryption.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Конференції Державного університету «Житомирська політехніка». URL: https://conf.ztu.edu.ua/wp-content/uploads/2016/06/3.pdf (дата звернення: 03.01.2023).
2. ISO/IEC 22123-1:2023. ISO. URL: https://www.iso.org/standard/82758.html (дата звернення: 03.01.2023).
3. What Is a Public Cloud? | Google Cloud. Google Cloud. URL: https://cloud.google.com/learn/what-is-public-cloud (дата звернення: 04.01.2023).
4. IBM Documentation. IBM - Deutschland | IBM. URL: https://www.ibm.com/docs/en/tarm/8.8.6?topic=configuration-private-cloud (дата звернення: 04.01.2023).
5. Cloud Deployment Models: Advantages & Disadvantages. SaM Solutions. URL: https://sam-solutions.us/advantages-and-disadvantages-of-cloud-deployment-models/#Cloud_Deployment_Model_3_-_Community_Cloud (дата звернення: 04.01.2023).
6. Cloud Deployment Models: Advantages & Disadvantages. SaM Solutions. URL: https://sam-solutions.us/advantages-and-disadvantages-of-cloud-deployment-models/#Cloud_Deployment_Model_3_-_Community_Cloud (дата звернення: 04.01.2023).
7. What is Hybrid Cloud? | IBM. IBM - Deutschland | IBM. URL: https://www.ibm.com/topics/hybrid-cloud?mhsrc=ibmsearch_a&mhq=hybrid%20cloud (дата звернення: 04.01.2023).
8. What's the Difference between Public, Private, Hybrid, and Community

Clouds? - AbacusNext. AbacusNext. URL: https://www.abacusnext.com/blog/whats-difference-between-public-private-hybrid-and-community-clouds/ (дата звернення: 04.01.2023).

9. Про хмарні послуги : Закон України від 17.02.2022 р. № 2075-IX. URL: https://zakon.rada.gov.ua/laws/show/2075-20#Text (дата звернення: 08.03.2023).
10. Thin Clients vs. Thick Clients: A Comparison | Parallels. Parallels: Mac & Windows Virtualization, Remote Application Server, Mac Management Solutions. URL: https://www.parallels.com/tips/thin-clients/vs-thick/ (дата звернення: 18.05.2023).
11. NIST cloud computing reference architecture / F. Liu та ін. Gaithersburg, MD : National Institute of Standards and Technology, 2011. URL: https://doi.org/10.6028/nist.sp.500-292 (дата звернення: 05.02.2023).
12. Threat Modelling Cloud Platform Services by Example: Google Cloud Storage. NCC Group Research Blog. URL: https://research.nccgroup.com/2023/01/31/threat-modelling-cloud-platform-services-by-example-google-cloud-storage/ (дата звернення: 01.05.2023).
13. Будько, М. Методика оцінки загроз для інформації автоматизованих систем / Микола Будько // Правове, нормативне та метрологічне забезпечення системи захисту інформації в Україні : науково-технічний збірник. – 2005. – Вип. 10. – С. 35-46. – Бібліогр.: 5 назв. URL: https://ela.kpi.ua/handle/123456789/11427 (дата звернення: 05.02.2023).
14. Ghetau C. The 10 Major Types of Cloud Malware Attacks | Buchanan Technologies. Buchanan Technologies. URL: https://www.buchanan.com/cloud-malware/ (дата звернення: 04.05.2023).

15. STRIDE-LM Threat Model. URL: https://csf.tools/reference/stride-lm/ (дата звернення: 05.02.2023).
16. Applying a Threat Model to Cloud Computing / V. E. Urias та ін. 2018 International Carnahan Conference on Security Technology (ICCST), м. Montreal, QC, 22–25 жовт. 2018 р. 2018. URL: https://doi.org/10.1109/ccst.2018.8585471 (дата звернення: 05.02.2023).
17. An Overview of Cloud Cryptography - GeeksforGeeks. GeeksforGeeks. URL: https://www.geeksforgeeks.org/an-overview-of-cloud-cryptography/ (дата звернення: 05.02.2023).
18. Andress J. The Basics of Information Security. Elsevier, 2014. URL: https://doi.org/10.1016/c2013-0-18642-4 (дата звернення: 03.05.2023).
19. IBM Cloud Docs. IBM Cloud. URL: https://cloud.ibm.com/docs/key-protect?topic=key-protect-envelope-encryption (дата звернення: 03.05.2023).
20. Cloud Encryption: Benefits, Challenges, & More - CrowdStrike. crowdstrike.com. URL: https://www.crowdstrike.com/cybersecurity-101/cloud-security/cloud-encryption/ (дата звернення: 16.04.2023).
21. Post-Quantum TLS - Microsoft Research. Microsoft Research. URL: https://www.microsoft.com/en-us/research/project/post-quantum-tls/ (дата звернення: 03.05.2023).
22. What is post-quantum cryptography?. Educative: Interactive Courses for Software Developers. URL: https://www.educative.io/answers/what-is-post-quantum-cryptography (дата звернення: 04.05.2023).
23. Cloud Object Storage – Amazon S3 – Amazon Web Services. Amazon Web Services, Inc. URL: https://aws.amazon.com/s3/ (дата звернення: 25.04.2023).

24. Protecting data using encryption - Amazon Simple Storage Service. URL: https://docs.aws.amazon.com/AmazonS3/latest/userguide/UsingEncryption.html (дата звернення: 17.05.2023).
25. Use envelope encryption with AWS KMS keys - Financial Services Industry Lens. URL: https://docs.aws.amazon.com/wellarchitected/latest/financial-services-industry-lens/use-envelope-encryption-with-customer-master-keys.html (дата звернення: 25.04.2023).
26. Azure Blob Storage | Microsoft Azure. Cloud Computing Services | Microsoft Azure. URL: https://azure.microsoft.com/en-us/products/storage/blobs/ (дата звернення: 25.04.2023).
27. About keys - Azure Key Vault. Microsoft Learn: Build skills that open doors in your career. URL: https://learn.microsoft.com/uk-UA/azure/key-vault/keys/about-keys (дата звернення: 25.04.2023).
28. Cloud Storage | Google Cloud. Google Cloud. URL: https://cloud.google.com/storage (дата звернення: 25.04.2023).
29. Default encryption at rest | Documentation | Google Cloud. Google Cloud. URL: https://cloud.google.com/docs/security/encryption/default-encryption (дата звернення: 25.04.2023).
30. Envelope encryption | Cloud KMS Documentation | Google Cloud. Google Cloud. URL: https://cloud.google.com/kms/docs/envelope-encryption (дата звернення: 25.04.2023).

ДОДАТОК А

КРИПТОГРАФІЧНІ МЕТОДИ У ХМАРНИХ СЕРВІСАХ

Єлизавета ЛАЗАРЄВА[a], Юрій ГОРБЕНКО[b]

[a] *Харківський національний університет імені В. Н. Каразіна, майдан Свободи 4, Харків, Україна*
[b] *АТ «Інститут Інформаційних технологій», вул. Коломенська 15, Харків, Україна*

Анотація. *У цій роботі наведено огляд криптографічних методів для забезпечення безпеки у хмарах. Зокрема, один із розділів присвячений використанню постквантової криптографії у хмарних сервісах.*
Ключові слова. *Хмарні сервіси, хмарна криптографія, постквантова криптографія*

1. Вступ

Хмарні технології стають все більш популярним засобом для задоволення різноманітних потреб людства. Проте, разом із розвитком хмарних сервісів з'являються нові загрози безпеці хмарних сервісів. Тож, актуальною є задача підвищення крипостійкості алгоритмів шифрування шляхом об'єднання існуючих методів. У зв'язку із цим, у даній роботі буде розглянуто криптографічні методи, які використовуються в хмарних сервісах.

2. Застосування криптографічних методів у хмарних сервісах

2.1. Хмарна криптографія

Хмарна криптографія [1] дозволяє забезпечити високий рівень безпеки хмарних сервісів. Вона заснована на шифруванні. Існує декілька підходів використання криптографічних методів у хмарних сервісах:

- Попереднє шифрування (з хмарним сервісом синхронізуються вже зашифровані дані);
- Наскрізне шифрування (бачити повідомлення можуть тільки відправник та одержувач);
- Шифрування файлів (дані шифруються таким чином, що, якщо неавторизована особа спробує перехопити файл, вона не зможуть отримати доступ до даних, які він містить);
- Повне шифрування диска (файли автоматично шифруються під час зберігання на зовнішньому диску).

Криптографічні методи, що використовуються у хмарних сервісах [2] можна поділити на симетричні та асиметричні.

2.2. Криптографія із симетричним ключем

Симетричні методи використовують один ключ як для шифрування, так і для розшифрування даних. Криптографія із симетричним ключем забезпечує автентифікацію та авторизацію даних, оскільки дані, зашифровані одним ключем, неможливо розшифрувати будь-яким іншим ключем.

Симетричні методи є дуже швидкими та ефективними, проте проблемою є безпека ключа, оскільки для шифрування та розшифрування даних використовується один і той самий ключ. Якщо ключ скомпрометовано, то всі зашифровані дані будуть під загрозою. До того ж безпечний обмін ключами може бути проблематичним для великої кількості користувачів.

До симетричних методів відносяться: Data Encryption Standard (DES), triple DES (3DES), Advanced Encryption Standard (AES).

2.3. Криптографія із асиметричним ключем

Асиметричні методи використовують два ключі – відкритий та особистий. За допомогою відкритого ключа повідомлення шифрується, а особистий ключ використовується для розшифрування.

Асиметричні методи є більш безпечними, добре підходять, для випадків, коли задіяно багато кінцевих точок (особистий ключ зберігається лише у одного користувача, що зменшує імовірність його розкриття), а також спрощується передача ключа (відкритий ключ не потрібно зберігати в таємниці). Проте вони є менш швидкими та ефективними у порівнянні з симетричними методами, а також мають великий розмір ключів, отже, їх проблематично використовувати в системах з великою кількістю даних.

У хмарних сервісах використовуються: DSA, RSA та алгоритм Діффі-Хелмана.

2.4. Гешування

Геш-функції використовуються для перетворення будь-якого повідомлення на строку певного виду та заданої довжини. Гешування дозволяє швидко перевірити, чи два набори даних ідентичні, порівнюючи їх геш-коди, замість того, щоб порівнювати всі дані вручну. У хмарних технологіях, гешування використовується для зберігання конфіденційної інформації, які можуть бути потрібні для аналізу даних або обчислень, але не можуть бути розкриті.

Перевірка геш-коду є набагато швидшою, ніж порівняння двох великих наборів даних та геш-коди можна зберігати у вигляді індексів для зберігання та пошуку даних у великих базах даних. Проте гешування неефективне, коли є багато колізій (для великого набору можливих ключів геш-колізій практично не уникнути), а також геш-таблиці можуть бути складними для реалізації.

У хмарних сервісах використовують алгоритми SHA та MD5.

3. Застосування постквантової криптографії у хмарних сервісах

Квантові комп'ютери потенційно здатні зламати більшість таких шифрувань. Отже, для захисту хмарних сервісів доцільно розглядати можливість використання методів постквантової криптографії [3].

Одним із таких методів є постквантовий протокол обміну ключами. Так, до протоколу TLS додаються нові постквантові шифри та забезпечується захист від класичної та постквантової схеми. Квантовий обмін ключами гарантує, що жодна сторона не може дізнатися значення ключа без використання додаткової інформації. Більш того, виключена можливість навіть здобути таку інформацію, оскільки квантові системи забороняють копіювання квантових станів.

Власне квантову криптографію можна поділити на 4 види:

- Криптографія на основі коду (наприклад, McEliece – це криптосистема, яка дозволяє кодувати та перекодувати біти даних під час передачі та отримання);
- Криптографія на основі решітки [4] (використовує багатовимірні решітки, прикладом використовуваної криптосистеми є NTRU);
- Криптографія на основі гешу (наприклад, система підпису відкритого ключа геш-дерева Меркле);
- Багатовимірна криптографія з відкритим ключем (використовує випадкові набори квадратних рівнянь і використовує певні точки для зашифрування та розшифрування).

4. Висновки

Варто зазначити, що необхідно використовувати комплексний підхід до захисту хмарних сервісів. Одним із аспектів захисту може бути використання хмарної криптографії за допомогою методів, розглянутих у даній роботі: з симетричним ключем, з асиметричним ключем, гешування тощо. Також є сенс досліджувати постквантові методи, що можуть бути використані для підвищення безпеки хмарних сервісів, адже вони забезпечують достатній рівень захисту у разі появи квантових комп'ютерів, які здатні зламати більшість існуючих методів криптографії.

5. Список літератури

[1] An Overview of Cloud Cryptography – GeeksforGeeks. GeeksforGeeks. URL: https://www.geeksforgeeks.org/an-overview-of-cloud-cryptography/.

[2] Rajendirakumar S. Cryptographic Algorithms used in Cloud Computing – an Analysis and Comparison. *International Journal for Research in Applied Science and Engineering Technology*. 2018. Vol. 6, no. 1. P. 2718–2728. URL: https://doi.org/10.22214/ijraset.2018.1373.

[3] Melu E. Quantum Computing: A Blessing or A Curse (Part 2). Medium. URL: https://medium.com/@14edymelu/quantum-computing-a-blessing-or-a-curse-part-2-126f0649496c.

[4] Substantiation of promising post-quantum national lattice-based electronic signature standard / A. M. Олексійчук et al. Radiotekhnika. 2020. Vol. 1, no. 200. P. 5–14. URL: https://doi.org/10.30837/rt.2020.1.200.01.