Список літератури

1. Herrmann, J. B. Metaphor in specialist discourse. J. B. Herrmann, T. Berber Sardinha. – Amsterdam, Netherlands: John Benjamins Publishing Company, 2015. 185 p.
2. Kuhn T.S. Metaphor in science. Ortony A. Metaphor and thought. - Cambridge: Cambridge University Press, 2014. 409–419 p.
3. Newmark Peter. A Textbook of Translation. Harlow: Pearson Education Limited, 2008. 292 p.
4. Яковенко Р. В. Тлумачний англо-український словник економічних термінів з елементами теорії та проблематики. Дидактичний довідник. Роман Яковенко. – [Вид. 2–ге, випр.]. – Кіровоград: видавець Лисенко В.Ф., 2015. – 130 с. URL: http://dspace.kntu.kr.ua/jspui/bitstream/123456789/5566/1/Tlumachniy%20slovnyk.pdf (дата звернення: 12.10.2022
5. Abbreviations Dictionary. URL: http://www.acronymfinder.com (дата звернення: 12.09.2022).
6. Collins English Dictionary. Harper Collins Publishers, 2006. 774 p.
7. Levy J. Translation as a Decision Process. Translation Studies Reader. London and New York, 2003. p. 148–159.

УДК 004

В.Сушков, магістр гр. КІ-21М-1,4,
Центральноукраїнський національний технічний університет

ДОСЛІДЖЕННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ ХМАРНОГО СЕРВІСУ З ВИКОРИСТАННЯМ АЛГОРИТМУ TDEA

У статті розроблено програмне забезпечення, яке призначено для системи хмарного сервісу з використанням алгоритму TDEA. Метою розробки є дослідження та програмна реалізація системи хмарного сервісу з використанням алгоритму TDEA. Об'єктом дослідження є процес хмарного сервісу з використанням алгоритму TDEA. Предметом дослідження є методи хмарного сервісу з використанням алгоритму TDEA. Методи дослідження базуються на методах захисту інформації, методах математичної статистики, методах розробки програмного забезпечення. Результат роботи – програмна реалізація системи хмарного сервісу з використанням алгоритму TDEA. В процесі роботи над програмною моделлю виконано аналіз існуючих апаратних та програмних засобів. В повній мірі описані всі компоненти розробленого програмного забезпечення.

комп'ютерна інженерія, хмарний сервіс, TDEA

Постановка проблеми. Хмарні обчислення – це архітектура для надання обчислювальних послуг через Інтернет на вимогу та платного доступу до пулу спільних ресурсів без їх фізичного отримання. Таким чином, це заощаджує кошти та час для організації. Хмарні обчислення є новою парадигмою. Яка стала найактуальнішою сферою досліджень сьогодні завдяки своїй здатності зменшувати витрати, пов'язані з обчислювальною технікою. Сьогодні це найцікавіша та приваблива технологія, яка пропонує своїм користувачам послуги на вимогу через Інтернет. Оскільки хмарні обчислення зберігають дані та їх розповсюджені ресурси в середовищі, безпека стала основною перешкодою, яка заважає розгортанню хмарних середовищ [1]. Багато користувачів використовують хмару для зберігання своїх особистих даних. Таким чином, необхідна безпека зберігання даних на носії інформації [3]. Ця концепція використовується алгоритмом DNS. Це симетричний блоковий шифр, який можна використовувати як додаткову заміну DES або IDEA. Він використовує ключ змінної довжини, від 32 біт до 448 біт, що робить його ідеальним як для внутрішнього використання, так і для експорту. Це алгоритм шифрування, який можна використовувати як заміну алгоритмам DES або IDEA. Ця стаття використовується, щоб дізнатися про безпеку хмарних обчислень за допомогою алгоритму DES [4]. Практична атака Sweet32 на набори шифрів на основі 3DES у TLS вимагала блоків

(785 ГБ) для повної атаки, але дослідникам пощастило отримати зіткнення одразу після блоків, що зайняло лише 25 хвилин. На безпеку TDEA впливає кількість блоків, оброблених одним набором ключів [8]. Один пакет ключів не повинен використовуватися для застосування криптографічного захисту (наприклад, шифрування) більш ніж 64-бітних блоків даних.

Аналіз останніх досліджень і публікацій. При аналізі останніх досліджень і публікацій [1-10] було виявлено певні прогалини у забезпеченні системи хмарного сервісу з використанням алгоритму TDEA.

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи хмарного сервісу з використанням алгоритму TDEA.

Для досягнення поставленої мети визначена програма дослідження, що складається з наступних завдань:

– Огляд існуючих систем хмарного сервісу з використанням алгоритму TDEA.
– Дослідження системи хмарного сервісу з використанням алгоритму TDEA.
– Програмна реалізація системи хмарного сервісу з використанням алгоритму TDEA.

Об'єктом дослідження є процес хмарного сервісу з використанням алгоритму TDEA.

Предметом дослідження є методи хмарного сервісу з використанням алгоритму TDEA.

Методи дослідження базуються на методах захисту інформації, методах математичної статистики, методах розробки програмного забезпечення.

Виклад основного матеріалу

Більшість організацій вже досить давно використовують переваги хмарних обчислень. Важко нехтувати перевагами гнучкості, гнучкості та масштабованості, коли було б важко підтримувати лише фізичне обладнання.

Де ускладнюється пошук способів захисту робочих ресурсів, які розміщені за межами вашого приміщення. Це відкриває шлях до хмарної безпеки як дисципліни захисту хмарних систем даних. Давайте заглибимося в тему, щоб знайти все, що потрібно знати про хмарні рішення безпеки та як вони працюють.

Що таке хмарна безпека?

Хмарна безпека – це набір процедур і технологій, призначених для захисту даних і захисту від зовнішніх і внутрішніх загроз. Із зростанням інтеграції з хмарою зростають і потенційні ризики, і компаніям потрібні рішення для захисту їх мережевої інфраструктури. Встановлення правильного балансу між продуктивністю та безпекою має першочергове значення.

Хмарні рішення безпеки розгортаються подібно до інструментів, що використовуються для захисту фізичного обладнання. Ключова відмінність полягає в тому, що ними також можна керувати та розгортати віддалено. Відповідальність за захист даних розподіляється між постачальником хмарних технологій і клієнтом. Перший постачальник повинен забезпечити безпеку налаштування свого апаратного забезпечення та правил доступу, тоді як другий має подбати про шифрування сховища та різні конфігурації політик безпеки.

Це одна з ключових причин, чому вважається, що хмарну безпеку підтримувати набагато важче, ніж локальні моделі. Оскільки є більше залучених сторін, це також означає, що щось важливе може бути пропущено. Не кажучи вже про те, що використання зовнішніх постачальників позбавляє клієнта значної видимості та контролю.

Чому хмарна безпека важлива?

Організації значною мірою покладаються на хмарні обчислення для багатьох своїх повсякденних операцій. Динамічний характер хмарної інфраструктури надає багато чудових можливостей для компаній, які прагнуть отримати переваги під час досягнення своїх бізнес-цілей. Оскільки потенціал великий, компанії, які знаходять способи приборкати хмарні

обчислення, можуть подолати багато викликів ІТ.

Однак, оскільки хмарні обчислення все ще є новою територією для більшості підприємств, ризики, пов'язані зі збереженням ваших даних поза межами, є більш помітними. Відповідно до домовленості між хмарним провайдером, кожен клієнт несе відповідальність за безпеку своїх даних. Таким чином, кожна організація має розглянути, як підійти до хмарної безпеки для свого унікального бізнес-випадку.

Кібербезпека завжди вимагає активного внеску з боку організації. В іншому випадку вони ризикують привернути небажану увагу з боку хакерів, які спеціально націлені на хмарні мережі. Тому хмарні обчислення актуальні незалежно від розміру вашої організації чи галузі.

Основні переваги хмарної безпеки

Хмарна безпека приносить користь організаціям кількома способами:

– Допомагає запобігти кібератакам. Хмарна безпека може бути основою для стримування або припинення вхідних спроб злому.

– Покращує безпеку даних. Різні технології допомагають захистити конфіденційні дані, допомагаючи захистити дані, щоб вони не потрапили в чужі руки.

– Полегшує обслуговування хмари. Більшість хмарних служб пропонують моніторинг і підтримку в реальному часі, що допомагає підвищити надійність обслуговування.

– Швидше одужання. У разі порушення даних хмарні інструменти безпеки допомагають легше організувати процес відновлення.

– Відповідність нормативним вимогам. Часто хмарна безпека є обов'язковою вимогою для безпечної акредитації на відповідність нормативним вимогам.

Як працює хмарна безпека?

Хмарна безпека допомагає організаціям, надаючи різні елементи керування для захисту від загроз для програм даних і хмарних систем. Оскільки платформи хмарних обчислень є основним рішенням для більшості підприємств, загрози, націлені на бізнес, часто спрямовані на хмару.

Тому хмарні рішення безпеки допомагають бізнесу кількома способами:

– Збільшити прозорість. Набагато легше захистити організацію, коли мережеві адміністратори знають, до чого мають доступ користувачі.

– Моніторинг стану мережі. Знання про те, яка діяльність відбувається в хмарі, може допомогти зупинити різні ризики на їх шляху.

– Підвищує рівень безпеки. Найважливіші ресурси можна краще захистити від доступу неавторизованих користувачів до конфіденційної інформації.

– Забезпечує ефективніше керування ідентифікацією. Збільшення вимог до доступу допомагає захистити облікові записи користувачів від захоплення.

– Порівнює безпеку з вимогами відповідності. Оскільки більшість компаній зберігають багато конфіденційної інформації, хмарна безпека допомагає їм відповідати визначеним стандартам безпеки.

Типи хмарних середовищ

Незважаючи на такий загальний термін, хмарні обчислення можна налаштувати різними способами. Важливо також відзначити, що навіть один і той же тип хмари може бути організований по-різному. Проте кожен тип хмарних обчислень має слабкі та сильні сторони, які можуть суттєво вплинути на ваш бізнес.

Загальнодоступні хмари

Загальнодоступна хмара – це середовище, яке на вимогу розповсюджується через загальнодоступний Інтернет постачальником послуг. Деякі загальнодоступні хмари є безкоштовними для всіх, тоді як інші вимагають підписки або тарифікуються згідно з моделями оплати за користування. Найбільші публічні хмарні постачальники включають Google Cloud, Amazon Web Services, Microsoft Entra ID і IBM Cloud.

Такі послуги допомагають перспективним компаніям переносити робочі навантаження назовні та легко збільшувати або зменшувати масштаб відповідно до своїх потреб. Це звільняє локальних мережевих адміністраторів і допомагає знизити ІТ-витрати.

Набагато дешевше використовувати спільну інфраструктуру, якою керує третя сторона, ніж мати такий самий масштаб налаштування всередині компанії.

Приватні хмари

Приватна хмара – це хмарне середовище, у якому всі апаратні та програмні ресурси зарезервовані та доступні для одного клієнта. Часто ці середовища захищені брандмауером групи. Це створює повністю ізольований доступ без збігів з іншими користувачами хмари.

Більшість компаній віддають перевагу налаштуванням приватної хмари, оскільки це набагато простіший спосіб забезпечити безпеку та відповідати вимогам відповідності. Однак один великий недолік цієї установки полягає в тому, що вона не така масштабована, як публічна хмара. Приватні хмари зазвичай мають фіксований розмір, і їх неможливо миттєво збільшити або зменшити. Для підвищення масштабу приватної хмари знадобляться додаткові ліцензії на обладнання та програмне забезпечення.

Гібридні хмари

Гібридна хмара – це середовище, у якому програми працюють із різних джерел: хмарних і локальних. Цей метод є найпопулярнішим у хмарних обчисленнях, оскільки більшість компаній отримують найкраще з обох світів. Більшість компаній використовують інфраструктуру, яку вони створили протягом тривалого часу, і розширюють її за допомогою хмарних доповнень.

Підключення хмарних і локальних середовищ зазвичай здійснюється за допомогою локальних мереж (LAN), глобальних мереж (WAN), віртуальних приватних мереж (VPN) та інших методів. Вся установка керується з інтегрованої платформи керування та оркестровки.

Багатохмарність

Мультихмари – це комбінації різних типів хмар, публічних або приватних. Це налаштування створюється, коли різні хмари (часто від різних постачальників послуг) об'єднуються певним методом інтеграції чи оркестровки. Це допомагає уникнути прив'язки до постачальника та створювати більш гнучкі рішення, адаптовані до конкретних потреб бізнесу.

Часто такі налаштування створюються для однієї хмари, яка функціонує як резервна копія на випадок запобігання втраті даних. У разі аварії дані організації можна безпечно відновити з резервної копії.

Типи моделей Cloud Service

Хмарні обчислення можна надати як три різні моделі послуг, кожна з яких надає унікальний набір переваг, які можуть задовольнити різні потреби бізнесу.

IaaS

Інфраструктура як послуга фактично пропонує типові компоненти інфраструктури центру обробки даних, такі як обладнання, обчислювальна потужність, простір для зберігання або мережеві ресурси. Доступ до ресурсів здійснюється через віртуальні або приватні мережі, і клієнт може швидко використовувати їх. Цей метод вирішує проблему підтримки фізичного обладнання для малих, середніх і великих компаній.

SaaS

Програмне забезпечення як послуга – це ліцензія та модель продажу, яка використовується для доставки програмного забезпечення через загальнодоступний Інтернет. Зазвичай користування здійснюється за підпискою. Після сплати комісії ви можете користуватися послугою протягом встановленого періоду часу. Постачальник – це той, хто контролює весь стек обчислень. Тим часом користувач може безпосередньо взаємодіяти з програмним забезпеченням з його кінцевої точки.

PaaS

Платформа як послуга пропонує повний набір інструментів середовища розробки. Це значно спрощує процес розробки програмного забезпечення та корисно під час створення нових програм. Ця структура миттєво надає інструменти проектування, тестування та доставки, що дозволяє клієнтам швидко розпочати роботу над новими проектами.

Типи хмарних рішень безпеки

Доступно кілька типів хмарних рішень безпеки, кожне з яких підходить для конкретного завдання.

Керування ідентифікацією та доступом (IAM)

Управління ідентифікацією та доступом (IAM) – це структура бізнес-процесів, яка полегшує політику та технології для керування цифровою ідентифікацією. IT-менеджери можуть використовувати IAM для контролю доступу до ресурсів організації. IAM створює цифрові ідентифікатори для кожного користувача, що полегшує їх моніторинг і обмеження.

Запобігання втраті даних (DLP)

Запобігання втраті даних (DLP) – це набір інструментів і процесів, які використовуються для забезпечення безпеки бізнес-даних. Він використовує різні інструменти, як-от шифрування, профілактичні заходи та сповіщення про виправлення, щоб захистити дані під час передавання чи зберігання.

Інформація про безпеку та керування подіями (SIEM)

Управління інформацією про безпеку та подіями (SIEM) – це підхід до управління безпекою для оркестрування IT-безпеки організації. Він використовує різні інструменти керування інформацією та подіями для створення єдиної інформаційної панелі за допомогою ШІ для кореляції даних на кількох платформах. Це дозволяє легко мати повний панорамний огляд безпеки організації.

Безперервність бізнесу та аварійне відновлення

Інструменти безперервності бізнесу (BC) і аварійного відновлення (DR) надають організаціям інструменти, послуги та протоколи для відновлення організації після аварії. Ці послуги допомагають організаціям зменшити ризик втрати даних і шкоди репутації та покращити поточні бізнес-операції.

Хмарні загрози безпеці

Хмарні системи піддаються тим самим ризикам, що й ваша локальна інфраструктура. Однак залучення додаткових сторін збільшує загальну суму ризиків.

– Відсутність повного контролю. Оскільки хмарні сервіси існують за межами корпоративних мереж, організації не повністю контролюють усі сфери кібербезпеки.

– Багатоквартирність. Коли кілька клієнтів орендують послуги в одного постачальника, ви можете потрапити під лавину, коли один із ваших сусідів буде зламаний.

– Shadow IT. Хмарні середовища сумно відомі тіньовими IT-налаштуваннями, особливо коли активна політика «принеси свій власний пристрій» (BYOD).

– Неправильні конфігурації. Однією з найпоширеніших причин витоку даних є неправильні налаштування. Інсайдерські аварії часто призводять до витоку клієнтської інформації, що засмучує, навіть якщо налаштування безпеки належні. Дізнайтеся більше про хмарні загрози безпеці, ризики та вразливості

Хмарні інструменти безпеки

Ось деякі з конкретних інструментів, які використовуються для захисту хмари:

– Cloud Workload Protection Platform (CWPPs) – система безпеки, призначена для захисту робочих навантажень

– Cloud Access Security Brokers (CASB) – посередник між хмарними клієнтами та хмарним сервісом, який забезпечує дотримання політик безпеки

– Cloud Security Posture Management (CSPM) – набір інструментів безпеки, які полегшують моніторинг і виявлення неправильної конфігурації

– Secure Access Service Edge (SASE) – конвергенція різноманітних засобів безпеки та мережевих інструментів, щополегшує керування безпекою мережі

Нарешті, численні доповнення, як-от веб-служби IAM, інструменти DLP та інші інструменти безпеки, допомагають користувачам хмари.

Як захистити хмару

Ось кілька порад про те, як краще захистити інформацію в хмарі.

– Шифрування. Для каналів зв'язку та постійного зберігання слід використовувати

шифрування. Таким чином дані будуть недоступні під час передачі та коли ваш сервер зламано.

– Безпечні конфігурації. Дотримання належної гігієни управління послугами кібербезпеки. Це передбачає зміну паролів за замовчуванням і отримання додаткової інформації про елементи керування безпекою хмарного постачальника.

– Використовуйте надійні паролі. Жодне налаштування безпеки не допоможе, якщо ваші користувачі повторно використовують ті самі паролі. Надійні паролі піднімають смугу входу в організацію, ускладнюючи проникнення.

– Обмежити дозволи. Їх не слід надавати, якщо для виконання певної посадової ролі не потрібні дозволи. Хоча це здається обмежувальним, це також допомагає запобігти багатьом ризикам кібербезпеки.

Нарешті, для користувачів, які покладаються на сторонніх постачальників, неможливо недооцінити, наскільки важливим є аналіз умов обслуговування. Чіткий розподіл обов'язків допоможе уникнути сірих зон, які можуть бути використані. Це важливий документ, який допомагає зрозуміти слабкі сторони вашого поточного налаштування та кроки, які можна вжити, щоб виправити його налаштування.

TDEA

Сервіси хмарного зберігання даних швидко стають все більш популярними. Користувачі можуть зберігати свої дані в хмарі та отримувати доступ до них будь-де в будь-який час. Через конфіденційність користувачів дані, що зберігаються в хмарі, зазвичай зашифровані та захищені від доступу інших користувачів.

Беручи до уваги властивість хмарних даних працювати над співробітництвом, шифрування на основі атрибутів (ABE) вважається однією з найбільш підходящих схем шифрування для хмарних сховищ [7]. Існує багато запропонованих схем ABE.

Більшість із запропонованих схем припускають, що постачальники послуг хмарного сховища або довірені треті сторони, які займаються керуванням ключами, є надійними та не можуть бути зламані; однак на практиці деякі суб’єкти можуть перехоплювати зв’язок між користувачами та постачальниками хмарних сховищ, а потім змушувати постачальників сховищ розкривати секрети користувачів, використовуючи владу уряду чи інші засоби [3]. У цьому випадку вважається, що зашифровані дані відомі, а постачальники сховищ мають надати секрети користувача. Google надав ФБР документи користувача після отримання ордера на обшук. У 2013 році Едвард Сноуден розкрив існування глобальних програм стеження, які збирають такі хмарні дані, як електронні листи, текстові та голосові повідомлення від деяких технологічних компаній. Після зламу постачальників хмарних сховищ усі схеми шифрування втрачають свою ефективність [4].

Хоча ми сподіваємося, що постачальники хмарних сховищ зможуть боротися з такими організаціями, щоб зберегти конфіденційність користувачів через законні шляхи, це, здається, стає все складнішим.

Наприклад, Lavabit була компанією, що надає послуги електронної пошти, яка захищала всі електронні листи користувачів від зовнішнього зловмисного впливу; на жаль, це не вдалося, і він вирішив закрити свою службу електронної пошти.

Оскільки боротися із зовнішнім зловмисним впливом важко, ми мали на меті створити схему шифрування, яка могла б допомогти постачальникам хмарних сховищ уникнути цієї скрутної ситуації. У цій роботі пропонується постачальникам хмарних сховищ засоби для створення підроблених секретів користувачів. Враховуючи такі фальшиві секрети користувача, сторонні особи можуть отримати лише підроблені дані зі збереженого зашифрованого тексту користувача. Щойно зловмисники подумають, що отримані секрети справжні, вони будуть задоволені, і, що більш важливо, постачальники хмарних сховищ не розкриють жодних справжніх секретів [8]. Тому конфіденційність користувачів все ще захищена.

Ця концепція походить від особливого виду схеми шифрування, яка називається шифруванням із запереченням, запропонована вперше [1]. Шифрування, яке можна

заперечувати, включає відправників і одержувачів, які створюють переконливі підроблені докази підроблених даних у зашифрованих текстах, щоб сторонні зловмисники особи були задоволені. Зверніть увагу, що заперечення випливає з того факту, що зловмисники не можуть довести, що запропоновані докази неправильні, і тому не мають підстав відхиляти надані докази. Цей підхід намагається повністю заблокувати зусилля зловмисного впливу, оскільки зловмисники знають, що їхні зусилля будуть марними.

Ми використовуємо цю ідею, щоб постачальники хмарних сховищ могли надавати послуги зберігання без аудиту.

У сценарії хмарного сховища власники даних, які зберігають свої дані в хмарі, є схожими на відправників у схемі забороненого шифрування [3]. Ті, хто має доступ до зашифрованих даних, відіграють роль одержувачів у схемі шифрування, яка забороняється, включаючи самих постачальників хмарних сховищ, які мають загальносистемні секрети та повинні мати можливість розшифрувати всі зашифровані дані.

У цій роботі ми описуємо заперечувальну схему ABE для хмарних служб зберігання. Ми використовуємо характеристики ABE для захисту збережених даних за допомогою точного механізму контролю доступу та забороненого шифрування для запобігання зовнішньому аудиту. Наша схема заснована на схемі шифрування на основі атрибутів політики Waters (CP-ABE) [4]. Ми вдосконалюємо схему Уотерса від білінійних груп простого порядку до білінійних груп складеного порядку. Згідно з припущенням про проблему вирішення підгрупи, наша схема дозволяє користувачам мати можливість надавати підроблені секрети, які здаються законними зовнішнім зловмисникам.

Загалом, потрійний DES із трьома незалежними ключами має довжину ключа 168 біт (три 56-бітні ключі DES), але завдяки зустрічі в середині атаки ефективний захист, який він забезпечує, становить лише 112 біт. Варіант ключа 2 зменшує ефективний розмір ключа до 112 біт (оскільки третій ключ такий самий, як і перший). Однак цей параметр сприйнятливий до певних атак із відкритим текстом або відомим текстом, і, таким чином, NIST визначає лише 80 біт безпеки. Це можна вважати несправним, оскільки доступний споживач може ретельно шукати весь простір ключів 3des.

Сахай і Уотерс першими представили концепцію ABE, у яку власники даних можуть вбудовувати спосіб обміну даними з точки зору шифрування [4]. Тобто лише ті, хто відповідає умовам власника, можуть успішно розшифрувати збережені дані. Тут ми зауважимо, що ABE – це шифрування для привілеїв, а не для користувачів. Це робить ABE дуже корисним інструментом для хмарних служб зберігання, оскільки обмін даними є важливою функцією для таких служб [2]. Користувачів хмарних сховищ так багато, що власникам даних непрактично шифрувати свої дані за допомогою парних ключів. Крім того, для багатьох людей також непрактично шифрувати дані багато разів. За допомогою ABE власники даних вирішують лише те, які користувачі можуть отримати доступ до їхніх зашифрованих даних [5]. Користувачі, які задовольняють умови, можуть розшифрувати зашифровані дані.

Існує два типи ABE: CP-ABE і Key-Policy ABE (KP-ABE). Різниця між цими двома полягає в перевірці політики. KP-ABE – це ABE, у якому політику вбудовано в секретний ключ користувача, а набір атрибутів – у зашифрований текст.

І навпаки, CP-ABE вбудовує політику в зашифрований текст, а секрет користувача має набір атрибутів [3].

Goyal та ін. запропонувала першу KP-ABE. Вони створили виразний спосіб зв'язати будь-яку монотонну формулу як політику для секретних ключів користувача.

Бетенкур запропонував перший CP-ABE в [4]. Ця схема використовувала деревоподібну структуру доступу для вираження будь-якої монотонної формули через атрибути як політику в зашифрованому тексті.

Перший повністю експресивний CP-ABE був запропонований Вотерсом, у якому використовувалися схеми лінійного секретного обміну (LSSS) для побудови політики шифрованого тексту [5].

Левко та ін. покращив схему Waters до повністю безпечного CP-ABE, хоча й з деякою втратою ефективності.

На сучасному етапі аналізується техніко-економічне обґрунтування проекту та висувається бізнес-пропозиція з дуже загальним планом проекту та деякими оцінками витрат. Під час аналізу системи має бути проведено техніко-економічне обґрунтування запропонованої системи. Це зроблено для того, щоб запропонована система не була тягарем для компанії. Для аналізу здійсненності важливе певне розуміння основних вимог до системи. Три ключові міркування, пов'язані з аналізом здійсненності:

– економічний аналіз.

– технічний аналіз.

– соціальний аналіз.

Економічний аналіз

Це дослідження проводиться для перевірки економічного впливу від система матиме на організації. Сума коштів, яку компанія може влити в дослідження та розробку системи, обмежена. Витрати мають бути обґрунтованими. Таким чином, розроблена система також в межах бюджету, і це було досягнуто завдяки тому, що більшість використовуваних технологій знаходяться у вільному доступі. Треба було купувати лише індивідуальні продукти.

Технічний аналіз

Це дослідження проводиться для перевірки технічної здійсненності, тобто технічних вимог системи. Будь-яка розроблена система не повинна мати високих вимог до наявних технічних ресурсів. Це призведе до високих вимог до наявних технічних ресурсів. Це призведе до високих вимог до клієнта. Розроблена система повинна мати скромні вимоги, оскільки для впровадження цієї системи потрібні лише мінімальні або нульові зміни.

Соціальний аналіз

Аспектом дослідження є перевірка рівня прийняття системи користувачем. Це включає процес навчання користувача ефективному використанню системи. Користувач не повинен відчувати загрозу з боку системи, натомість повинен прийняти її як необхідність. Рівень прийняття користувачами залежить виключно від методів, які використовуються для навчання користувача системі та ознайомлення його з нею. Його рівень довіри потрібно підвищити, щоб він також міг зробити певну конструктивну критику, що вітається, оскільки він є кінцевим користувачем системи.

Вхід та вихід алгоритму DES

Дизайн вводу є ланкою між інформаційною системою та користувачем. Він включає в себе розробку специфікації та процедур для підготовки даних, і ці кроки, необхідні для переведення даних транзакцій у придатну для використання форму для обробки, можуть бути досягнуті шляхом перевірки комп'ютера для зчитування даних із письмового чи друкованого документа або це може відбутися за допомогою людей, які вводять ключі дані безпосередньо в систему.

Дизайн введення зосереджується на контролі необхідного обсягу введення, контролі помилок, уникненні затримок, уникненні додаткових кроків і збереженні простоти процесу.

Вхід розроблений таким чином, щоб забезпечити безпеку та легкість використання зі збереженням конфіденційності.

– Дизайн вхідних даних – це процес перетворення орієнтованого на користувача опису вхідних даних в комп'ютерну систему. Цей дизайн важливий, щоб уникнути помилок у процесі введення даних і показати правильний напрямок керівництву для отримання правильної інформації з комп'ютеризованої системи.

– Це досягається шляхом створення зручних екранів для введення даних для обробки великого обсягу даних. Метою проектування вхідних даних є полегшення введення даних і відсутність помилок. Екран введення даних розроблено таким чином, що можна виконувати всі операції з даними. Він також надає можливість перегляду записів.

– Після введення даних буде перевірено їх дійсність. Дані можна вводити за

допомогою екранів. Відповідні повідомлення надаються, коли це необхідно, щоб користувач не перебував у кукурудзяному стані.

Таким чином, мета дизайну вхідних даних полягає в тому, щоб створити макет вхідних даних, яким легко слідувати.

Вихідні дані

Якісні вихідні дані відповідають вимогам кінцевого користувача та чітко представляють інформацію. У будь-якій системі результати обробки повідомляються користувачам та іншій системі через виходи. У проекті виводу визначається, як інформація має бути переміщена для негайної потреби, а також виведення друкованої копії. Це найважливіше і пряме джерело інформації для користувача. Ефективна та інтелектуальна конструкція виводу покращує взаємозв'язок системи, щоб допомогти користувачеві приймати рішення:

– Розробка комп'ютерного виходу повинна відбуватися організовано, добре продумано; необхідно розробити правильний результат, забезпечуючи, щоб кожен вихідний елемент був розроблений таким чином, щоб люди могли легко й ефективно використовувати систему. Під час аналізу проектування вихідних даних комп'ютера вони повинні визначити конкретний результат, який необхідний для задоволення вимог.

– Виберіть методи для представлення інформації.

– Створення документів, звітів або інших форматів, які містять інформацію, створену системою.

Вихідна форма інформаційної системи повинна досягати однієї або більше з наступних цілей.

– Передавати інформацію про минулу діяльність, поточний стан або прогнози – майбутнього.

– Сигналізуйте про важливі події, можливості, проблеми або попередження.

– Викликати дію.

– Підтвердити дію.

Безпека.

Загалом, потрійний DES із трьома незалежними ключами має довжину ключа 168 біт (три 56-бітні ключі DES), але завдяки атаці зустрічі посередині ефективний захист, який він забезпечує, становить лише 112 біт. Варіант 2 ключа зменшує ефективний розмір ключа до 112 біт (оскільки третій ключ такий самий, як і перший). Однак ця опція сприйнятлива до певних атак обраного відкритого тексту або відкритого тексту, і, таким чином, NIST визначає лише 80-бітний захист [3]. Це можна вважати несправним, оскільки доступний споживач може ретельно шукати весь простір ключів 3des.

Апаратна атака сьогодні на пакети шифрів на основі 3DES у TLS вимагала блоків (785 ГБ) для повної атаки, але дослідникам пощастило отримати колізію відразу після блоків, що зайняло лише 25 хвилин [4]. Безпека TDEA постраждала за кількістю блоків, оброблених одним пакетом ключів.

Крім того, ми забезпечуємо, що цей автентифікатор може бути ефективно згенерований власником даних одночасно з процедурою кодування.

Широкий аналіз показує, що наша схема є доведено безпечною, а оцінка продуктивності показує, що наша схема є високоефективною та може бути реально інтегрована в систему хмарного зберігання на основі відновленого коду.

Розробка структурної схеми

На рисунку 1 зображена структурна схема системи хмарного сервісу з використанням алгоритму TDEA.

Забезпечення безпеки інформації при зберіганні й обробці більших інформаційних масивів – одна із самих актуальних проблем сучасних інформаційних технологій. Інтенсивний розвиток методів розподіленої обробки даних і різке збільшення обсягів інформації, що накопичується в комп'ютерних системах, привели останнім часом до кардинальної зміни методів довгострокового зберігання даних. Традиційні підходи до

організації зберігання більших інформаційних масивів перестали задовольняти зрослим вимогам до ємності носіїв і швидкості доступу до даних. Всі частіше споживач довіряє зберігання своєї власної інформації зовнішнім центрам або мережам зберігання даних (так званий аутсорсинг). Одна з основних сфер застосування мережного зберігання даних – формування банків даних електронних документів, а також електронних архівів і бібліотек. Такі сховища даних можуть бути як публічними, так і обмеженого доступу, залежно від характеру документів, що накопичуються в них. Нерідко перед приміщенням документів у мережні сховища вони піддаються стиску або іншим спеціальним видам кодування. У зв'язку із цим загострюється необхідність забезпечення керованості, надійності й безпеці зберігання й доступу до електронних документів, а також процедур їхньої передачі між прикладними програмами й пристроями зберігання.

Якщо навіть дані зберігаються локально, виникає інша проблема: адміністратори, що управляють СУБД і персонал так чи інакше звичайно мають права доступу до всієї збереженої інформації. Для захисту від їхніх несанкціонованих дій у деяких випадках доцільне застосування апаратно-програмних засобів шифрування даних перед записом їх на засоби зберігання. Часто шифрувальні модулі вбудовуються, наприклад, у засоби резервного копіювання даних. Однак при зберіганні шифрованих масивів утруднений пошук окремих файлів і оперативний доступ до елементів масиву, необхідним для роботи прикладних програм. Тому що масив зберігається в зашифрованому виді, і серверу, на якому він зберігається (або СУБД), не можуть бути довірені ключі шифрування, користувач (або прикладна програма від його ім'я) змушений завантажувати копії всіх файлів масиву, розшифровувати їх і потім виконувати пошук на локальній машині. Очевидно, що такий спосіб пошуку дуже неекономічний. У зв'язку із цим вимальовується проблема забезпечення можливості пошуку даних по шифрованим і (або) стислим даним, що може бути конкретизована залежно від застосовуваної в системі моделі шифрування даних.

Для шифрування великих масивів даних, що поміщаються в зовнішні стосовно власника інформації сховища, ефективні лише симетричні схеми шифрування. Можливості їхнього практичного застосування, мабуть, визначаються можливостями організації схеми керування секретними ключами, для яких необхідно забезпечити виконання двох почасти суперечливих вимог: забезпечення високої схоронності ключів (зокрема, за рахунок резервування) і обмеження середовища їхнього поширення тільки тими пристроями, яким довіряє власник інформації.

У зв'язку із цим у деяких випадках більше раціональним виглядає застосування схем відкритого шифрування, що дає можливість невизначеному колу осіб поміщати свої дані в сховище, але доступ до них залишати лише для власників секретного ключа. Така схема може бути корисна, наприклад, для систем електронної пошти або систем планування потоків завдань (workflows), де циркулюють переважно повідомлення невеликої довжини. Для таких схем тим більше необхідні механізми пошуку за шифрованим даними, що операції розшифрування в асиметричних криптосхемах, як відомо, виконуються на кілька порядків повільніше в порівнянні із симетричними.

Інша проблема, пов'язана із забезпеченням конфіденційності пошуку в масивах даних, пов'язана з бажанням унеможливити одержання адміністратором СУБД і сторонніми особами відомостей про те, до яких саме записів (або фрагментів) бази даних здійснювався доступ при кожному конкретному запиті. У закордонній літературі це завдання зветься “Private Information Retrieval” (PIR). Вона особливо актуальна, наприклад, при обробці й зберіганні електронних документів, що містять відомості приватного характеру: фінансові, юридичні, майнові, медичні й інші.

Якщо навіть самі поля бази даних зашифровані, характер і частота запитів до них уже можуть дати зловмисникові деяку непряму інформацію, розголошення якої небажано для власника. Ці завдання до визначеної міри аналогічні виникаючої в телекомунікаційних системах завданню маскування інтенсивності трафіка між вузлами, що, як відомо, вирішується шляхом суцільного заповнення каналу псевдовипадковими послідовностями.

Виходячи зі структурної схеми системи зображеної на рисунку 3, система хмарного сервісу з використанням алгоритму TDEA, працює наступним чином.

Спершу при вході в систему, користувач звертається до блоку розмежування доступу.

Блок розмежування доступу отримує пароль користувача, та звертається до менеджеру паролів, де отримує сеансовий пароль перевірки правильності паролю користувача, та правильності прав доступу користувача, які зберігаються у відповідних зашифрованих базах даних.

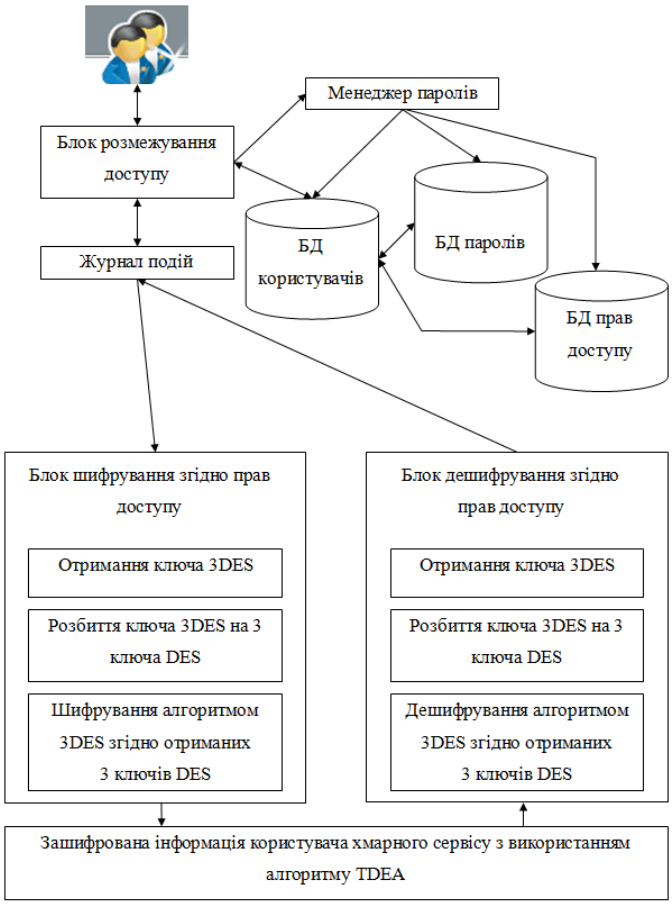

Рисунок 1 – Структурна схема системи

Розмежування цих баз зроблено з метою підвищення стійкості системи зберігання інформації.

Після підтвердження прав доступу, та правильності введеного паролю, користувачеві видається сеансовий ключ 3DES для роботи з інформацією.

У блоці шифрування згідно прав доступу, з отриманого ключа 3DES добуваються 3 ключа алгоритму DES, за допомогою яких й відбувається шифрування інформації алгоритмом 3DES. Процедура дешифрування відбувається аналогічним чином.

Висновки. У статті наведені теоретичне узагальнення й рішення наукового завдання дослідження методів хмарного сервісу з використанням алгоритму TDEA. Рішення даного завдання полягало у вирішенні наступних задач: Був проведений огляд існуючих систем хмарного сервісу з використанням алгоритму TDEA. Досліджена система хмарного сервісу з використанням алгоритму TDEA. На основі отриманих результатів досліджень створена програмна реалізація системи хмарного сервісу з використанням алгоритму TDEA. Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання хмарного сервісу з використанням алгоритму TDEA. Проведено аналіз предметної галузі в ході якого були виявлені об'єкти, взаємодія яких носить істотний характер для функціональної діяльності

предметної галузі, і їхні основні характеристики; побудована алгоритм і вибраний середовище розробки.

Список літератури

1. О.А. Смірнов, П.С. Усік, «дослідження перспектив використання технологічних рішень в мережах 5g» у Кібербезпека та інформаційні технології: монографія. – Х. : ТОВ «ДІСА ПЛЮС», 2020.С. 122-135.
2. Смірнов О.А., Смірнова Т.В., Якименко Н.М., Смірнов С.А., Поліщук Л.І., «Дослідження стійкості до диференціального криптоаналізу запропонованої функції гешування удосконаленого модуля криптографічного захисту в інформаційно-комунікаційних системах» Системи управління, навігації та зв'язку, 2022, № 3(69). С. 93-98. 2022.
3. Смірнов О.А., Смірнова Т.В., Якименко Н.М., Поліщук Л.І., Смірнов С.А. «Дослідження статистичної стійкості та швидкісних характеристик запропонованої функції гешування удосконаленого модуля криптографічного захисту в інформаційно-комунікаційних системах» Вісник Хмельницького національного університету. Серія: «Технічні науки», № 2 (307). С. 46-52. 2022.
4. Смірнов О.А., Смірнова Т.В., Константинова Л.В., Смірнов С.А., Якименко Н.М., «Дослідження стійкості до лінійного криптоаналізу запропонованої функції гешування удосконаленого модуля криптографічного захисту в інформаційно-комунікаційних системах» Системи управління, навігації та зв'язку, 2022, № 1(67). С. 84-89.
5. Смірнов О.А., Смірнова Т.В., Буравченко К.О., Кравченко С.С., Горбов В.О., «Хмарна система підтримки прийняття рішень технологічного процесу відновлення поверхонь конструкцій і деталей машин». Сучасні інформаційні системи. 2021. Т. 5, № 4. С. 79-95
6. Смирнов А., Кузнецов А., Кузнецова Т. «Шумоподобные дискретные сигналы для асинхронных систем кодового разделения радиоканалов». Радиотехника, № 2(205), 175–183. 2021.
7. Smirnov O., Kuznetsov A., Kovalchuk D., Kuznetsova T. «New Technique for Hiding Data in Cover Images Using Adaptively Generated Pseudorandom Sequences». CEUR Workshop Proceedings Volume 2732, 2020, Pages 214-227.
8. Смірнов, О.А., Полігенько О.О., Одарченко Р.С., Терещенко Л.Ю.Усік П.С., «Інформаційна технологія та програмне забезпечення для підвищення ефективності планування підсистеми базових станцій стільникового зв'язку». Проблеми телекомунікацій. № 1(26). С. 83-96. 2020.
9. Смірнов О.А., Усік П.С., Миронець І.В., Буравченко К.О., Якименко Н.М. «Метод підвищення ефективності розподіленої обробки даних у комп'ютерних системах операторів стільникового зв'язку» Вісник Черкаського державного технологічного університету. Технічні науки. №4. С. 103-110. 2020.
10. О.А.Смірнов, Т.В.Смірнова, Л.І. Поліщук, К.О. Буравченко, А.О.Макевнін, «Дослідження хмарних технологій як сервісів», Кібербезпека: освіта, наука, техніка. № 3(7). С. 43-62. 2020.
11. Smirnov, O., Neskorodieva, T., Fedorov, E., Rudakov, K., Neskorodieva, A. «Method Detection Audit Data Anomalies on Basis Restricted Cauchy Machine» CEUR Workshop Proceedings, Volume 3187, 2022, pp. 1-12..
12. Smirnov O., Smirnova T., Anas M. Al-Oraiqat, Drieiev O., Polishchuk L., Sheroz Khan, Yassin M. Y. Hasan, Aladdein M. Amro, Hazim S. AlRawashdeh «Method for Determining Treated Metal Surface Quality Using Computer Vision Technology». Sensors (Basel, Switzerland) Volume 22, Issue 16, 6223, 2022..
13. Smirnov, O., Lakhno, V., Akhmetov, B., Chubaievskyi, V., Khorolska, K., Bebeshko, B. «Selection of a Rational Composition of İnformation Protection Means Using a Genetic Algorithm». In: Rajakumar, G., Du, KL., Vuppalapati, C., Beligiannis, G.N. (eds) Intelligent Communication Technologies and Virtual Mobile Networks. Lecture Notes on Data Engineering and Communications Technologies, vol 131. 2023. Springer, Singapore. pp. 21-34..
14. Smirnov O.A., Al-Oraiqat A.M., Ulichev O.S., Meleshko Ye.V., Al-Rawashdeh H.S., Polishchuk L.I. «Modeling strategies for information influence dissemination in social networks». Journal of Ambient Intelligence and Humanized Computing Volume 13, Issue 5. Springer, Cham. 2022, pp. 2463-2477..
15. Smirnov O., Kuznetsov A., Kryvinska N., Kiian A., Kuznetsova K. «Full Non-Binary Constant-Weight Codes». SN Computer Science, Vol 2, 337, 2021. https://doi.org/10.1007/s42979-021-00739-w.
16. Smirnov O., Kovalenko O., Kovalenko A., Kavun S. «Quantitative Risk Assessment Method Development in the Context of the SDLC-model». 2021 IEEE 8th International Conference on Problems of Infocommunications, Science and Technology (PIC S&T), 2021, pp. 203-208, doi: 10.1109/PICST54195.2021.9772143.
17. Smirnov O., Neskorodieva T., Fedorov E., Rymar P. «Neural Network Modeling Method of Transformations Data of Audit Production with Returnable Waste». CEUR Workshop Proceedings Volume 3101, 2021, Pages 192-207..
18. Smirnov O., Kuznetsov A., Kiian A., Kuznetsova K. «Data hiding scheme based on spread sequence addressing». CEUR Workshop Proceedings Volume 2805, 2020, Pages 44-58..
19. Smirnov, O., Kuznetsov, A., Potii, O., Poluyanenko, N., Stelnyk, I., Mialkovsky, D. «Combining and filtering functions in the framework of nonlinear-feedback shift register». International Journal of Computing; 2020, Volume 19, Issue 2 – Research Institute for Intelligent Computer Systems – 2020. – P. 247-256..
20. Smirnov O., Kuznetsov A., Kiian A., Kuznetsova T. «Non-binary constant weight coding technique». CEUR Workshop Proceedings. Volume 2740, 2020, Pages 102-114..