НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ
Кафедра інформаційної безпеки

До захисту допущено
В.о. завідувача кафедри

_____ **Микола ГРАЙВОРОНСЬКИЙ**
(підпис)

«_____»_____2021 р.

Дипломна робота

на здобуття ступеня бакалавра

за освітньо-професійною программою «Системи, технології та математичні методи кібербезпеки»
спеціальності: 125 «Кібербезпека»

на тему: Моделювання та порівняльний аналіз багаторівневої безпеки в реляційних СУБД

Виконав (-ла): здобувач вищої освіти **IV** курсу, групи ФБ-72
(шифр групи)

Солдатова Катерина Іванівна _____
(прізвище, ім'я, по батькові) (підпис)

Керівник к.т.н., доцент Коломицев Михайло Володимирович _____
(посада, науковий ступінь, вчене звання, прізвище, ім'я, по батькові) (підпис)

Рецензент _____ _____
(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище, ім'я, по батькові) (підпис)

Засвідчую, що у цій дипломній роботі немає запозичень з праць інших авторів без відповідних посилань.
Здобувач вищої освіти _____
(підпис)

Київ - 2021 року

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ
Кафедра інформаційної безпеки

Рівень вищої освіти – перший (бакалаврський)
Спеціальність – 125 «Кібербезпека»
Освітньо-професійна програма «Системи, технології та математичні методи кібербезпеки»

ЗАТВЕРДЖУЮ
В.о. завідувача кафедри
_____ Микола ГРАЙВОРОНСЬКИЙ
(підпис)
«___»_____2021 р.

ЗАВДАННЯ
на дипломну роботу здобувачу вищої освіти

Солдатова Катерина Іванівна
(прізвище, ім'я, по батькові)

1. Тема роботи "Моделювання та порівняльний аналіз багаторівневої безпеки в реляційних СУБД", керівник роботи к.т.н., доцент Коломицев Михайло Володимирович затверджені наказом по університету
від « » _____ 2021 р. №

2. Термін подання здобувачем вищої освіти роботи: 07 червня 2021 р.

3. Вихідні дані до роботи: попередні дослідження багаторівневої безпеки

4. Зміст роботи: огляд існуючих моделей багаторівневої безпеки в реляційних СУБД, їх порівняльний аналіз методом прийняття рішень з використанням безпосереднього експертного оцінювання

5. Перелік ілюстративного матеріалу (із зазначенням плакатів, презентацій тощо): Моделювання та порівняльний аналіз багаторівневої безпеки в реляційних СУБД - презентація

6. Дата видачі завдання 01.10.2020

Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів дипломної роботи	Примітка
1	Отримання завдання	01.10.2020	виконано
2	Вивчення та аналіз літератури	05.10.2020-03.01.2021	виконано
3	Ознайомлення з особливостями багаторівневої безпеки	04.01.2021-25.01.2021	виконано
4	Ознайомлення з моделями багаторівневої безпеки	26.01.2021-03.03.2021	виконано
5	Вибір методу порівняльного аналізу	04.03.2021-14.03.2021	виконано
6	Порівняльний аналіз моделей багаторівневої безпеки	15.03.2021-10.04.2021	виконано
7	Проходження переддипломної практики	12.04.2021-16.05.2021	виконано
8	Написання дипломної роботи	01.04.2021-01.06.2021	виконано
9	Отримання допуску до захисту	02.06.2021	виконано
10	Захист дипломної роботи	16.06.2021	

Здобувач вищої освіти _____ Катерина СОЛДАТОВА
(підпис) (Власне ім’я, ПРІЗВИЩЕ)

Керівник роботи _____ Михайло КОЛОМИЦЕВ
(підпис) (Власне ім’я, ПРІЗВИЩЕ)

РЕФЕРАТ

Дипломна робота містить: 63 сторінки, 3 ілюстрації, 18 таблиць, 1 додаток, 13 джерел літератури.

Метою роботи є висновки щодо особливостей практичної реалізації багаторівневих моделей безпеки в реляційних СУБД та порівняльний аналіз моделей для обрання найбільш прийнятного варіанту.

Об’єктом дослідження є моделі багаторівневої безпеки в реляційних СУБД.

Предметом дослідження є порівняльний аналіз моделей багаторівневої безпеки в реляційних СУБД.

Методами дослідження є аналіз літературних джерел за обраною тематикою, визначення найважливіших особливостей існуючих моделей багаторівневої безпеки, порівняльний аналіз розглянутих моделей.

Було досліджено такі моделі багаторівневої безпеки в реляційних СУБД: модель Secure Data Views, модель Jajodia-Sandhu, модель Smith-Winslett, модель MultiLevel Relation та модель Belief-Consistent Multilevel Secure Data Model. Було проведено порівняльний аналіз даних моделей методом прийняття рішень на основі експертної оцінки. За результатами аналізу, найкраща для реалізації модель багаторівневої безпеки за визначеними коефіцієнтами – модель Multilevel Relation.

Ключові слова: реляційна модель даних, база даних, захист даних, системи управління базами даних, багаторівнева безпека, контроль доступу, безпека баз даних.

ABSTRACT

Thesis contains: 63 pages, 3 figures, 18 tables, 1 applications, 13 sources of literature. Purpose: conclusions about features of practical implementation of multilevel security models in relational DBMS and comparative analysis of models to choose the most acceptable variant.

The object of the study is multilevel security models in relational DBMS.

The subject of the study is the comparative analysis of multilevel security models in relational DBMS.

Research methods are the analysis of literature sources on the chosen subject, definition of the most important features of existing multilevel security models, the comparative analysis of the considered models.

The following models of multilevel security in relational DBMS have studied: the Secure Data Views model, the Jajodia-Sandhu model, the Smith-Winslett model, the MultiLevel Relation model and the Belief-Consistent Multilevel Secure Data Model. A comparative analysis of these models by the decision-making method that based on the expert evaluation has performed. According to the results of the analysis, the best model for implementation of multilevel security according to defined coefficients is the MultiLevel Relation model.

Keywords: relational data model, database, data protection, database management systems, multilevel security, access control, database security.

ЗМІСТ

ВСТУП

Сьогодні реляційна модель організації баз даних є найбільш поширеною. А найбільш використовуваним інструментом роботи з ними є системи управління базами даних (СУБД). За умови стрімкого розвитку інформаційної сфери та збільшення потоку інформації, забезпечення інформаційної безпеки реляційних СУБД має велике значення.

Більш складним для вирішення завданням щодо безпеки реляційних СУБД вважається організація політики доступу користувачів, що мають різні рівні допуску, до розподіленої за категоріями інформації. Тому була сформована багаторівнева безпека (Multilevel Security – MLS) – політика безпеки, яка створюється не тільки за допомогою дискреційних засобів контролю доступу, але і певних засобів контролю, які можуть запобігти отриманню несанкціонованого доступу до недосяжних для користувача категорій інформації або зміні категорії доступної користувачеві інформації.

Загалом, політика MLS була створена для роботи з системами баз даних критичних державних установ. Але з плином часу її почали дотримуватися і деякі комерційні підприємства. Наразі існують організації, що створили та підтримують різні форми багаторівневої безпеки. Ці рішення створені на основі різних моделей розмежування доступу та є механізмом впровадження політики безпеки MLS.

В багаторівневій безпеці кожний елемент бази даних визначений як об’єкт захисту та має свою класифікаційну мітку безпеки, а кожний користувач визначається як суб’єкт і також отримує свою мітку безпеки, яка визначає його рівень доступу. І саме на реалізації правил роботи суб’єктів з об’єктами засновані різні моделі багаторівневої безпеки СУБД.

Зараз відомі такі моделі створені на основі моделі Белл-ЛаПадула: модель Secure Data Views (SeaView), модель Jajodia-Sandhu, модель Smith-Winslett,

модель MultiLevel Relation (MLR) та модель Belief-Consistent Multilevel Secure Data Model (BCMLS).

Актуальність роботи зумовлена тим, що наразі найбільш використовуваним типом систем управління базами даних є реляційні. Тому активно прикладаються зусилля для підвищення рівня їх безпеки. Найбільш перспективною для цієї цілі можна вважати моделі багаторівневої безпеки, тому їх моделювання та докладне дослідження дуже актуальні.

Метою роботи є висновки щодо особливостей практичної реалізації багаторівневих моделей безпеки в реляційних СУБД та порівняльний аналіз моделей для обрання найбільш прийнятного варіанту.

Завданням дослідження є:

- огляд основних відомостей щодо реляційних систем управління базами даних;
- огляд моделей багаторівневої безпеки реляційних СУБД та особливостей їх побудови;
- порівняльний аналіз розглянутих моделей.

Об’єктом дослідження є моделі багаторівневої безпеки в реляційних СУБД.

Предметом дослідження є порівняльний аналіз моделей багаторівневої безпеки в реляційних СУБД.

Методами дослідження є аналіз літературних джерел за обраною тематикою, визначення найважливіших особливостей існуючих моделей багаторівневої безпеки, порівняльний аналіз розглянутих моделей.

Наукова новизна полягає в тому, що вперше було проведено порівняльний аналіз моделей багаторівневої безпеки у реляційних СУБД за допомогою методу прийняття рішень.

Практичне значення одержаних результатів

Проведений порівняльний аналіз моделей багаторівневої безпеки може сприяти поширенню використання даних моделей, а також розумному вибору моделі для реалізації. Також проведений аналіз може набути подальшого розвитку шляхом покращення певних його аспектів.

Апробація результатів роботи

Роботу було оприлюднено у формі доповіді на XIX Всеукраїнській науково-практичній конференції студентів, аспірантів та молодих вчених «Теоретичні і прикладні проблеми фізики, математики та інформатики».

Публікації

Роботу було опубліковано у збірнику тез XIX Всеукраїнської науково-практичної конференції студентів, аспірантів та молодих вчених «Теоретичні і прикладні проблеми фізики, математики та інформатики».

1 РОЛЬ БАГАТОРІВНЕВОЇ БЕЗПЕКИ В РЕЛЯЦІЙНИХ СУБД

1. 1 Основні поняття безпеки реляційних баз даних

1. 1. 1 Основні поняття безпеки баз даних

На сьогоднішній день, безпека баз даних спирається на такі три основні аспекти([1]): конфіденційність, цілісність, доступність. Розглянемо ці аспекти дещо докладніше.

Конфіденційність передбачає, що інформація захищена від доступу до неї нелегітимного користувача. Тобто, база даних повинна бути захищена від несанкціонованого доступу до певних атрибутів чи кортежів, розголошення інформації, що зберігається у базі даних.

Цілісність передбачає, що інформація захищена від її несанкціонованої модифікації, втрати достовірності. Для бази даних вона полягає в тому, що можуть бути несанкціоновано змінені значення деяких атрибутів.

У свою чергу, доступність передбачає, що інформація захищена від несанкціонованого знищення, вона є доступною, і не утримується іншим користувачем. База даних повинна бути працездатною, і користувачі повинні отримувати від неї відповідь за таку кількість часу, яка є зручною для роботи з нею.

Система вважається безпечною, якщо її управління доступом працює так, що тільки легітимні користувачі чи процеси, які діють від їх імені, можуть читати, створювати, модифікувати та знищувати інформацію. Через те, що у реальному житті не існує ідеально безпечних систем, це визначення ми розглядаємо як визначення системи, якій ми можемо довіряти. Основними параметрами, які підтверджують надійність, є політика безпеки та гарантованість.

Політика безпеки відображає той набір законів, правил і норм поведінки, яким користується конкретна організація при обробці, захисті та поширенні інформації ([2]).

Якщо політику безпеки виразити математично, формально чи схематично, то це вираження буде вважатися моделлю безпеки. Найчастіше використовується суб'єктно-об'єктна модель. В такій моделі існують суб'єкти, які визначаються як певні активні сутності, і об'єкти, що є пасивними сутностями. Тобто, дії суб'єктів над об'єктами породжують процеси.

У свою чергу, гарантованість – це міра довіри, яку ми можемо висловити стосовно архітектури та реалізації системи([2]). Тобто гарантованість передбачає, що система відповідає певному рівню захисту і має певні механізми захисту.

1.1.2 **Реляційні бази даних**

Реляційною моделлю даних вважається така логічна модель даних, в основі якої полягає представлення даних у форматі таблиці. Дані в цій моделі пов'язані між собою за допомогою такого математичного поняття, як відношення. Воно представлене у вигляді таблиці, рядками якої є кортежі, а стовпцями – атрибути. Тому відношення – це поняття, яке задається переліком атрибутів та їх значень. Важливою особливістю є існування у кожному записі первинного ключа, який дає змогу виділяти унікальні записи.

Його математичний запис:

$$R(A_1,A_2,\ldots,A_n) \tag{1.1}$$

де R – відношення;

$A_1,A_2,\ldots,A_n$ – атрибути відношення.

Реляційна алгебра щодо моделі складається із операторів, що допомагають вибрати цілі чи часткові записи з відношень та поєднувати дані в різні відношення([3]). А реляційна модель передбачає існування операцій для роботи з відношеннями: об'єднання, з'єднання, проекція, різниця, добуток тощо([4]).

Створена на основі реляційної моделі даних база даних вважається реляційною. Зв’язки між таблицями у такій базі даних засновані на однакових значеннях певних ключових атрибутів. Існує три види моделі “сутність – зв’язок”([5]):

- один до одного (1:1) – один кортеж першої таблиці пов’язаний своїм елементом з одним кортежем другої таблиці, та навпаки;
- один до багатьох (1:M) – один кортеж першої таблиці пов’язаний своїм елементом з багатьма кортежами другої таблиці, але кортеж другої таблиці має зв’язок лише з одни кортежем першої;
- багато до багатьох (M:M) – один кортежей першої таблиці пов’язаний з багатьма кортежами другої таблиці, і навпаки.

1.2 Реляційні СУБД

1.2.1 Сутність реляційних СУБД

Система управління базами даних (СУБД) – це комплекс програмних і мовних засобів, необхідних для створення баз даних, підтримання їх в актуальному стані та організації пошуку в них необхідної інформації ([2]).

СУБД надають власнику можливість ефективного управління роботою баз даних, а саме створенням та використанням баз даних користувачами. До основних функцій СУБД відносять([5]): забезпечення мовних засобів опису та маніпуляції даними; забезпечення підтримки логічної моделі даних; забезпечення взаємодії логічної та фізичної структур даних; забезпечення захисту та цілісності даних; забезпечення підтримки БД в актуальному стані; адміністрування бази даних; управління даними у зовнішній пам’яті; управління транзакціями; журналізація.

Перевагами використання СУБД є висока ефективність роботи, можливість відновити систему після збою, апарат підтримки конфіденційності та цілісності даних.

1.2.2 Засоби захисту СУБД

Через те, що збої у роботі баз даних можуть мати катастрофічні наслідки для багатьох організацій, досить велика увага приділяється розвитку засобів захисту СУБД.

У свою чергу, система засобів захисту спрямована на реалізацію таких сервісів безпеки([6]):

- управління доступом – організація доступу до полів таблиці за різними моделями безпеки;
- ідентифікація і аутентифікація – користувачеві бази даних надається ім’я та перевіряється, чи є користувач легітимним;
- криптографія – забезпечує цілісність інформації;
- протоколювання і аудит – забезпечення підзвітності, надання інформації для виявлення і аналізу проблем;
- засоби мови SQL – надання і відміна прав доступу, створення збережених процедур та тригерів, резервне копіювання даних.

1.2.3 Управління доступом

Управління доступом визначає керування над використанням ресурсів системи. В контексті баз даних здійснюється керування доступом до таблиць та їх полів. Управління доступом виконує такі функції([1]): ідентифікує користувачів та ресурси системи, здійснює автентифікацію та авторизацію, здійснює аудит, а також реагує на усі спроби отримати несанкціонований доступ.

Наразі існують такі три типи контролю доступу: дискреційна, мандатна та рольова моделі безпеки.

Способом формалізованого представлення дискреційного доступу є матриця доступу або списки управління доступом, що встановлюють перелік користувачів і перелік дозволених операцій відносно кожного об'єкта БД([2]). Вона базується на взаємодії суб’єктів та об’єктів. Кожний суб’єкт має певний перелік можливих дій щодо певного об’єкта. Усі об’єкти та суб’єкти повинні бути однозначно ідентифіковані, та кожний об’єкт має власника.

Мандатна модель доступу передбачає, що кожному суб’єкту та кожному об’єкту системи надається певна мітка безпеки, яка відображає певний рівень конфіденційності. При використанні цієї моделі у базах даних мітки безпеки отримують користувачі, а також рядки та стовпці таблиць. Право читати інформації з об’єкту отримують суб’єкти з рівнем доступу не менше рівня доступу об’єкта. А записувати інформацію у об’єкт можуть суб’єкти, рівень яких дорівнює або нижче рівня доступу об’єкта.

У рольовій моделі управління доступом здійснюється на основі ролей, які надаються користувачам системи. Роль – це певна сукупність дій та привілей користувача щодо об’єктів та інших суб’єктів. Ця сукупність може бути визначена або за допомогою матриці доступу, або певних правил, які описують можливу поведінку користувача. Рольове розмежування доступу поєднує елементи мандатного та дискреційного розмежування доступу

1.3 Концепція багаторівневої безпеки реляційних СУБД

1.3.1 Основні принципи багаторівневої безпеки реляційних СУБД

Багаторівнева безпека (Multilevel Security – MLS) – політика безпеки, яка дає можливість класифікувати суб’єкти та об’єкти на основі ієрархічних рівней безпеки та системі неієрархічних категорій безпеки([7]). Вона попереджує

можливість отримання користувачем інформації більш високого рівня безпеки, ніж рівень безпеки цього користувача. Багаторівнева безпека поєднує в собі особливості дискреційної та мандатної моделі доступу.

Суб'єктами в цій політиці вважаються сутності, які потребують ресурсів системи. Це може бути користувач, збережена процедура або пакетне завдання.

У свою чергу, об'єктами вважається ресурс, доступ до якого може контролюватися. Об'єктом може бути кортеж, таблиця, команда чи набір даних.

Як ми можемо побачити в документації DB2 на сайті IBM([7]), багаторівнева безпека має такі переваги:

- багаторівнева безпека може використовувати більш складні методи, які неможливо виразити через звичайні уявлення та запроси на мові SQL;
- багаторівнева безпека не дозволяє користувачам розсекречувати інформацію;
- управління доступом багаторівневої безпеки інтегрується в систему, тому кожен користувач унікальний;
- не потребує особливих уявлень для контролю безпеки строк.

Мітки безпеки у MLS пов'язані ієрархічно, тобто завжди можна визначити мітки, які будуть нижче чи вище за рівнем від обраної. Саме на домінування міток спирається управління доступом. Одним з прикладів міток безпеки на основі конфіденційності є поділення інформації на інформацію цілком таємну (TOP-SECRET – TS), таємну (SECRET – S), конфіденційну (CONFIDENTIAL – C) та некласифіковану (UNCLASSIFIED – U).

У загальному вигляді схему відношення багаторівневої безпеки позначають так([8]):

$$R(A_1;C_1;\ldots;A_N;C_N;TC) \tag{1.2}$$

де A_i – атрибут відношення;

C_i – його мітка безпеки;

TC – мітка безпеки кортежу.

Відношення може також мати лише мітки безпеки атрибутів:

$$R(A_1;C_1;\ldots;A_N;C_N) \quad (1.3)$$

чи лише мітку безпеки кортежу:

$$R(A_1;\ldots;A_N;TC) \quad (1.4)$$

В основі контролю доступу багаторівневої безпеки лежать принципи моделі Белл-ЛаПадула (Bell-LaPadula). Вона має такі властивості([9]):

- “Читати вниз” : Якщо мітка безпеки користувача більше або дорівнює мітці безпеки об’єкта, то читання дозволено.
- “Писати вгору” : Якщо мітка безпеки користувача менше або дорівнює мітці безпеки об’єкта, то запис дозволено.
- Строга властивість для запису : Якщо мітка безпеки користувача дорівнює мітці безпеки об’єкта, то запис дозволено.

Перші спроби реалізації багаторівневої безпеки у базах даних відбувалися ще у 1982 році([3]). Вже тоді було визначено, що основним інструментом роботи з таблицями в системі будуть представлення, бо вони дають змогу визначити обмежений обсяг інформації для роботи в залежності від контексту.

Реалізація політики багаторівневої безпеки передбачає створення програмного комплексу, який має п’ять рівнів([3]). На першому рівні знаходиться ядро безпеки, яке реалізує принципи моделі Белл-ЛаПадула. На другому рівні реалізуються реляційні операції, методи доступу. Третій рівень передбачає реалізацію представлень, забезпечує відображення багаторівневих відношень на основі однорівневих. Також цей рівень забезпечує дискреційне управління доступом. Четвертий рівень дає можливість управління даними нижчих рівнів, оптимізації запитів. А п’ятий рівень передбачає користувацький інтерфейс.

1. 3. 2 Загрози багаторівневої безпеки

Основним джерелом загроз щодо багаторівневої безпеки СУБД є випадки, що пов’язані з багатозначністю.

Багатозначністю (polyinstantation) називається особливість багаторівневої безпеки СУБД, яка передбачає, що існує декілька об’єктів з однаковими первинними ключами, але з різними мітками безпеки кортежу або атрибутів([9]). Це може призводити до витоку інформації.

Інше визначення багатозначності – ситуація, коли однакові дані виглядають по-різному для різних користувачів([10]).

Існує два випадки, які призводять до багатозначності:

- Коли користувач з міткою безпеки низького рівня додає у базу даних кортеж з первинним ключем, який вже існує на більш високому рівні. Такий випадок призводить невидимої багатозначності сутності.
- Коли користувач з міткою безпеки високого рівня створює у базі даних кортеж з первинним ключем, який вже існує на більш низькому рівні. Такий випадок призводить до видимої багатозначності сутності.

В першому випадку ми маємо такі варіанти рішення проблеми:

- Повідомити користувача, що додавання кортежу неможливе через існування ідентичного первинного ключа на більше високому рівні. Недолік цього рішення полягає в тому, що користувач дізнається про існування інформації, яка йому не повинна бути доступна за його міткою безпеки.
- Перезаписати існуючий кортеж на вищому рівні конфіденційності. Недолік цього рішення полягає в тому, що користувач не побачить перезапис, та буде порушена цілісність даних.
- Додати кортеж на рівні мітки безпеки користувача. Недоліком є те, що тепер ми маємо два відношення з ідентичним первинним ключем на різних рівнях безпеки.

Якщо ми розглянемо видиму багатозначність сутності, то маємо такі варіанти вирішення такої ситуації:

- Повідомити користувача, що кортеж з таким первинним ключем вже існує на більш низькому рівні, та відмовити у операції додавання. Недоліком цього рішення є те, що порушується доступність.
- Перезаписати кортеж на більш низькому рівні безпеки. Недолік цього рішення полягає в тому, що користувач більш низького рівня отримає інформацію, що повинна бути йому недоступна за рівнем конфіденційності.
- Додати кортеж на його рівні безпеки. Недоліком є наявність в результаті у базі даних двох кортежей з ідентичним первинним ключем, але з різними мітками безпеки.

Випадок, коли первинний ключ однаковий, але мітки безпеки кортежів різні, називається багатозначністю сутності.

Випадок, коли кортежі мають однаковий первинний ключ та його мітку безпеки, але різні мітки безпеки для атрибутів, називається багатозначністю атрибутів.

Висновки до розділу 1

На сьогоднішній день, безпека баз даних спирається на конфіденційність, цілісність та доступність. А основними параметрами, які підтверджують надійність системи, є політика безпеки та гарантованість.

Розглянута реляційна модель даних – логічна модель даних, в основі якої полягає представлення даних у форматі таблиці. Дані в цій моделі пов’язані між собою за допомогою такого математичного поняття, як відношення.

В свою чергу, СУБД, яка заснована на реляційній моделі даних, вважається реляційною. Вона реалізує різні сервіси безпеки, але найважливішим для нас

серед них є управління доступом. Воно визначає керування над використанням ресурсів системи.

Поєднання дискреційного та мандатного керування доступом характеризує багаторівневу безпеку. В основі контролю доступу цієї політики лежать принципи моделі Белл-ЛаПадула (Bell-LaPadula).

Основним джерелом загроз щодо багаторівневої безпеки СУБД є випадки, що пов’язані з багатозначністю. Вони можуть призводити до витоку інформації.

2 МОДЕЛЮВАННЯ БАГАТОРІВНЕВОЇ БЕЗПЕКИ БАЗ ДАНИХ

2.1 Модель Secure Data Views (SeaView)

Модель Secure Data Views (SeaView) передбачає, що кожному з атрибутів кортежу надається мітка безпеки, та відношення позначається як у формулі (1.3). В даній моделі зберігання даних здійснюється у вигляді набору одноріневих фрагментів, а багаторівневі відношення отримуються у вигляді представлень над цими відношеннями([9]). Перевагою такого підходу є максимальна прозорість роботи над відношеннями, що максимально зменшує ризик розкриття конфіденційних даних.

Для роботи з відношеннями використовуються два алгоритми: алгоритм декомпозиції та алгоритм відновлення. Алгоритм декомпозиції ділить багаторівневі відношення на відношення з одним рівнем доступу. Для алгоритму декомпозиції використовуються два типи фрагментації([11]): вертикальна та горизонтальна, і створюються відношення для первинних ключів (для кожного рівня безпеки окремо), а також для пар Ключ-Атрибут (для кожного рівня безпеки окремо). У свою чергу, алгоритм відновлення повертає однорівневі відношення до початкового стану.

Модель SeaView має багато недоліків, які пов’язані з алгоритмами декомпозиції та відновлення. Розглянемо основні з них([8]):

- В моделі SeaView використання алгоритму відновлення буде призводити до появи додаткових кортежів шляхом багатьох з’єднань однорівневих відношень. Ці кортежі називають хибними.
- Існують такі відношення до яких неможливо використовувати алгоритм декомпозиції даної моделі. Тобто є певні обмеження її використання.
- Основою алгоритму відновлення багаторівневого відношення є ліве зовнішне з’єднання відношень. Операція потребує багато ресурсів, а модель SeaView з

вертикальною фрагментацією використовує велику кількість лівих зовнішніх з’єднань для відновлення багаторівневого відношення.

2.1.1 Операція SELECT в моделі SeaView

Команда на мові SQL для операції SELECT в моделі SeaView має такий вигляд:

$$\text{SELECT } [A_1,A_2,...,A_n] \quad \text{FROM R} \quad \text{WHERE P} \tag{2.1}$$

де R – відношення;

$[A_1,A_2,...,A_n]$ – його атрибути;

P – умова, за якою визначаються потрібні користувачеві дані.

Команда операції SELECT має такий вигляд для всіх моделей багаторівневої безпеки.

Алгоритм роботи команди SELECT в моделі SeaView буде працювати таким чином:

- На першому кроці ми повинні отримати клас безпеки суб’єкта, який виконав операцію SELECT.
- На другом кроці однорівневі відношення поєднуються шляхом здійснення операції з’єднання, оберненої до вертикальної фрагментації, та операції об’єднання, оберненої до горизонтальної фрагментації.
- Третій крок. Вибираємо з отриманих багаторівневих відношень лише ті, які задовольняють умові P та рівень безпеки яких менше за рівень безпеки суб’єкта або дорівнює йому.

2. 1. 2 **Операція INSERT в моделі SeaView**

Команда операції INSERT на мові SQL в моделі SeaView має такий вигляд:

$$\text{INSERT} \quad \text{INTO R } [A_1,A_2,...,A_n] \quad \text{VALUES } [a_1,a_2,...,a_n] \qquad (2.2)$$

де R – відношення;

$[A_1,A_2,...,A_n]$ – його атрибути;

$[a_1,a_2,...,a_n]$ – значення цих атрибутів.

Команда операції INSERT має такий вигляд для всіх моделей багаторівневої безпеки.

Алгоритм роботи команди INSERT в моделі SeaView буде працювати таким чином:

- Перший крок. Отримуємо рівень безпеки суб’єкта, який виконує операцію.
- Якщо атрибут таблиці входить у список наведених у команді, то цьому атрибуту встановлюється відповідне значення.
- На третьому кроці всім доданим значенням атрибутів буде наданий рівень безпеки ідентичний рівню безпеки суб’єкта.
- Четвертим кроком отримуємо однорівневі відношення, та додаємо їх до вже наявних у базі даних. Рівень безпеки відношень буде дорівнювати рівню безпеки суб’єкта.

2. 1. 3 **Операція UPDATE в моделі SeaView**

Команда операції UPDATE в моделі SeaView має такий вигляд:

$$\text{UPDATE R} \quad \text{SET } [A_1 = a_1,\ A_2 = a_2,...,A_n = a_n] \quad \text{WHERE P} \qquad (2.3)$$

де R – відношення;

$[A_1,A_2,...,A_n]$ – його атрибути;

$[a_1,a_2,...,a_n]$ – значення цих атрибутів;

P – умова, за якою визначаються дані, які треба оновити.

Команда операції UPDATE має такий вигляд для всіх моделей багаторівневої безпеки.

Розглянемо алгоритм даної операції покроково:

- Крок 1. Отримуємо рівень безпеки суб'єкта, який виконує операцію.
- Крок 2. Отримуємо усі кортежі, які відповідають умові P та мають ідентичний чи нижчий рівень безпеки за рівень суб'єкта.
- Крок 3. Оновлюємо відповідні поля отриманих кортежів, які мають рівень безпеки ідентичний рівню користувача.
- Крок 4. Для кортежів, які мають рівень безпеки нижче рівня безпеки суб'єкта, ми підіймаємо їх рівень безпеки до рівня користувача та оновлюємо дані. В цьому випадку виникає багатозначність.

2. 1. 4 **Операція DELETE в моделі SeaView**

Команда на мові SQL для операції DELETE в моделі SeaView має такий вигляд:

DELETE FROM R WHERE P (2.4)

де R – відношення;

P – умова, за якою визначаються дані, які треба видалити.

Команди операції DELETE має такий вигляд для всіх моделей багаторівневої безпеки.

Ця операція має лише два кроки:

- Спочатку ми визначаємо рівень безпеки суб'єкта, що виконує операцію.
- Потім ми видаляємо усі кортежі, які задовольняють умові P та рівень безпеки яких відповідає рівню безпеки користувача.

2. 2 Модель Jajodia-Sandhu

Модель Jajodia-Sandhu заснована на моделі багаторівневої безпеки SeaView і була створена шляхом її модернізації. Основною відмінністю є робота алгоритмів декомпозиції та відновлення([8]).

В даній моделі використовується лише горизонтальна фрагментація. Як результат, щоб відновити багаторівневі відношення, вже не потрібно використовувати операції з'єднання, а можна обмежитися операціями об'єднання. Це сильно спрощує процес відновлення багаторівневих відношень, зменшує кількість необхідних ресурсів.

Незважаючи на певні покращення відповідно до моделі SeaView, у моделі Jajodia-Sandhu залишаються певні проблеми([12]):

- Семантична двозначність: Розглянемо випадок, коли існують відношення з однаковим первинним ключем не для всіх рівней безпеки. Тоді якщо користувач має мітку безпеки, для якої не існує відношення, і існують декілька відношень нижчих рівнів, то потрібно вирішувати, яке відношення відображати.
- Неповнота операцій: нехай є два рівні безпеки M1 та M2, найменша верхня межа яких – рівень безпеки S, а найбільша нижня межа – C. Тоді користувач з рівнем безпеки S не зможе додавати кортежі з мітками безпеки атрибутів C, M1 та M2.

У даній моделі передбачається, що відношення має в собі первинний ключ та набір атрибутів, що не завжди мають однакові мітки безпеки. Багаторівнева реляційна схема виглядає таким чином:

$$R\ (PK, C_{PK}; A_1, C_1; ...; A_n, C_n) \qquad (2.5)$$

де PK – первинний ключ;

C_{PK} – мітка безпеки первинного ключа;

A_i – атрибут відношення;

C_i – мітка безпеки атрибуту відношення.

2. 2. 1 Операція SELECT в моделі Jajodia-Sandhu

Команда операції SELECT в моделі Jajodia-Sandhu має вигляд як команда (2.1) і працює за таким алгоритмом:

- Першим кроком ми отримуємо рівень безпеки суб'єкта, який виконує операцію.
- Другим кроком ми отримуємо логічне представлення однорівневих відношень бази даних шляхом операції їх об'єднання.
- На третьому кроці ми беремо лише ті кортежі, які задовольняють умові P та рівень безпеки яких менше або дорівнює рівню безпеки суб'єкта, та відображаємо їх.

2. 2. 2 Операція INSERT в моделі Jajodia-Sandhu

Команда операції INSERT в моделі Jajodia-Sandhu має вигляд (2.2). В свою чергу, алгоритм операції ідентичний такому алгоритму у моделі SeaView. Єдина відмінність полягає в тому, що при додаванні кортежу він декомпонується у однорівневі відношення лише горизонтально.

2. 2. 3 Операція UPDATE в моделі Jajodia-Sandhu

Команда операції UPDATE в моделі Jajodia-Sandhu має вигляд як команда (2.3). Алгоритм даної операції майже цілком повторює алгоритм у моделі SeaView. Але відношення, дані яких замінюються, декомпоновані лише горизонтально.

2. 2. 4 Операція DELETE в моделі Jajodia-Sandhu

Команда операції DELETE в моделі Jajodia-Sandhu має вигляд як у схемі (2.4). Алгоритм даної операції також майже цілком аналогічна алгоритму у моделі SeaView. Відмінність полягає лише в способі декомпозиції відношень.

2. 3 Модель Smith-Winslett

Модель Smith-Winslett розглядає багаторівневу реляційну СУБД як набір реляційних баз даних, які побудовані за однаковою схемою, але для різних рівней безпеки. У моделі не використовуються мітки безпеки для атрибутів. Існують лише мітки безпеки первинного ключа та кортежу в цілому.

Схема відношення у моделі Smith-Winslett має такий вигляд:

$$R\ (PK, C_{PK}, A_1 ..., A_n, TC) \tag{2.6}$$

де PK – первинний ключ;

C_{PK} – мітка безпеки первинного ключа;

$A_1 ..., A_n$ – атрибути відношення;

TC – мітка безпеки кортежа.

Мітка безпеки кортежа завжди буде або дорівнювати мітці безпеки первинного ключа, або буде більша за неї.

Користувач у даній моделі буде бачити усі відношення його рівня безпеки та нижче. Але довіряти він буде лише інформації його рівня безпеки. Відповідно, змінювати, додавати та видаляти кортежі користувач може лише на своєму рівні безпеки. Ця властивість моделі дуже сильно обмежує її функціонал. Модель заснована на довірі (belief-based)([8]).

Модель Smith-Winslett, у порівнянні з моделлю Jajodia-Sandhu, має кращій механізм оновлення, але є дуже обмеженою в наборі об’єктів, які можна оновлювати.

2. 3. 1 Операція SELECT в моделі Smith-Winslett

В даній моделі операція SELECT виглядає як у схемі (2.1).

Алгоритм цієї операції має такий вигляд:

- Крок 1. Отримуємо рівень безпеки суб’єкта, який виконує операцію.
- Крок 2. Обираємо усі кортежі, які задовольняють умові P та загальний рівень безпеки яких дорівнює або менше рівня безпеки суб’єкта.

2. 3. 2 Операція INSERT в моделі Smith-Winslett

Команда операції INSERT в цій моделі виглядає як у схемі (2.2).

Алгоритм операції має такі кроки:

- Першим кроком ми отримуємо рівень безпеки суб’єкта, який виконує операцію.
- Якщо атрибут таблиці входить у список наведених у команді, то цьому атрибуту встановлюється відповідне значення.
- На третьому кроці рівень безпеки первинного ключа кортежа встановлюється відповідно до рівня безпеки суб’єкта, який виконує операцію.
- Четвертим кроком нові кортежі додаються до багаторівневих відношень.

2. 3. 3 Операція UPDATE в моделі Smith-Winslett

Команда операції UPDATE будується за схемою (2.3).

Операція працює за таким алгоритмом:

- Першим кроком ми повинні отримати рівень безпеки суб'єкта, який виконує операцію.
- Другим кроком ми оновлюємо усі кортежі, які відповідають умові P та мають рівень безпеки дорівнює рівню безпеки суб'єкта.

2. 3. 4 Операція DELETE в моделі Smith-Winslett

В моделі Smith-Winslett операція DELETE визначається за схемою (2.4).

Алгоритм цієї операції має такий вигляд:

- На першому кроці ми отримуємо рівень безпеки суб'єкта, який виконує операцію видалення.
- На другому кроці ми видаляємо усі кортежі, які задовольняють умові P та їх рівень безпеки дорівнює рівню безпеки суб'єкта.

2. 4 Модель MultiLevel Relation (MLR).

Модель MultiLevel Relation (MLR) реалізує концепцію цілісності даних, шляхом організації потоків інформації, що восходять([8]). Тобто зміна інформації на нижчому рівні безпеки призводить до зміни інформації на вищому.

Багаторівнева реляційна схема має вигляд([13]):

$$R\ (PK, C_{PK}, A_1, C_1, \ldots, A_n, C_n, TC) \qquad (2.7)$$

де PK – первинний ключ;

C_{PK} – мітка безпеки первинного ключа;

A_i – атрибут відношення;

C_i – мітка атрибуту відношення;

TC – мітка безпеки кортежу.

Ця модель даних створена на основі моделі Jajodia-Sandhu, але в ній була вирішена проблема семантичної двозначності. Це досягається завдяки додаванню команди UPLEVEL, за допомогою якої користувач може вказати, яким кортежам більш низького рівня він може довіряти.

Користувач моделі бачить не тільки дані його рівня безпеки, але й дані від користувачів більш низьких рівнів доступу ([8]). В моделі існує успадкування, яке працює аналогічно операції “запис вгору”, але не відбувається перезапис кортежів з більш високим рівнем безпеки. Успадковані дані можуть бути змінені їх власником з більш низьким рівнем безпеки.

Обмеженням даної моделі є відсутність можливості вказати, яким кортежам даних не можна довіряти.

2. 4. 1 **Операція SELECT в моделі MLR**

Команда операція SELECT виглядає як у схемі (2.1).

Алгоритм цієї операції має такий вигляд:

- Крок 1. Отримуємо рівень безпеки суб’єкта, який виконує операцію.
- Крок 2. Обираємо усі кортежі, які задовольняють умові P, а також загальний рівень безпеки яких дорівнює або менше рівня безпеки суб’єкта.

2. 4. 2 **Операція INSERT в моделі MLR**

В цій моделі команда операції INSERT виглядає як у схемі (2.2).

Алгоритм операції має такі кроки:

- Першим кроком ми отримуємо рівень безпеки суб'єкта, який виконує операцію.
- Якщо атрибут таблиці входить у список наведених у команді, то цьому атрибуту встановлюється відповідне значення. Якщо атрибут не вказаний, то йому надається нульове значення.
- На третьому кроці рівень безпеки первинного ключа кортежа встановлюється відповідно до рівня безпеки суб'єкта, який виконує операцію.
- Четвертим кроком нові кортежі додаються до багаторівневих відношень.

2. 4. 3 **Операція UPDATE в моделі MLR**

Команда операції UPDATE в моделі MLR виглядає як у схемі (2.3).

Вона працює за таким алгоритмом:

- Спочатку ми отримуємо рівень безпеки суб'єкта, який виконує операцію оновлення.
- Другим кроком ми перевіряємо, чи існує у SET атрибут первинного ключа. Якщо його немає, то ми оновлюємо усі кортежі, які задовольняють умові P та мають рівень безпеки, який дорівнює рівню безпеки суб'єкта. А також усі запозичені користувачами вищого рівня кортежі, які задовольняють умові P, теж будуть оновлені.
- Третім кроком, якщо існує атрибут первинного ключа у SET, то ми оновлюємо кортеж, що задовольняє умові P та має рівень безпеки ідентичний рівню безпеки суб'єкта. У свою чергу, запозичені записи користувачів вищих рівнів, які задовольняють умові P, видаляються.

2. 4. 4 Операція DELETE в моделі MLR

В цій моделі команда операції DELETE виглядає як у схемі (2.4).

Операція здійснюється таким чином:

- Першим кроком отримуємо рівень безпеки суб'єкта, що виконує операцію видалення.
- Другим кроком ми видаляємо усі кортежі, які задовольняють умові P та рівень безпеки яких дорівнює рівню безпеки суб'єкта.
- На третьому кроці ми перевіряємо, чи є на вищих рівнях запозичені записи. У такому випадку користувачу вищого рівня буде попереджено про видалення цього запису або буде повідомлено про виключення.

2. 4. 5 Операція UPLEVEL в моделі MLR

У даній моделі, на відміну від попередніх, існує додаткова операція UPLEVEL. Команда цієї операції виглядає так:

$$\text{UPLEVEL} \quad R \quad \text{GET} \ [A_1,A_2,...,A_n] \quad \text{FROM} \ [C_1,C_2,...,C_n] \quad \text{WHERE} \ P \qquad (2.8)$$

де R – відношення;

$A_1,A_2,...,A_n$ – атрибути відношення;

$C_1,C_2,...,C_n$ – рівні безпеки атрибутів $A_1,A_2,...,A_n$;

P – умова, за якою виконується операція.

Розглянемо алгоритм даної операції.

- Крок 1. Отримуємо рівень безпеки суб'єкта, який виконує дану операцію.
- Крок 2. Отримуємо усі кортежі, які задовольняють умові підвищення рівня P та мають рівень безпеки ідентичний або менше рівня безпеки суб'єкта.
- Крок 3. На третьому кроці для всіх отриманих кортежів будуємо нові таким чином. Первинний ключ нового кортежу має значення атрибуту та його рівень

безпеки ідентичні значенню та рівню первинного ключа у початковому кортежі. Для кожного не первинного ключа, якщо цей атрибут знаходиться в частині команди GET, то він буде мати те ж саме значення та його рівень безпеки буде дорівнювати рівню безпеки початкового атрибута. Якщо ж атрибута не має в частині команди GET, то він записується як нульовий і отримує рівень безпеки ідентичний рівню безпеки суб'єкта.

- Крок 4. Якщо кортеж з таким первинним ключем та рівнем безпеки як і створений існує, то він замінюється щойно створеним. Якщо не існує, то новий кортеж додається до багаторівневих відношень.

2.5 Модель Belief-Consistent Multilevel Secure Data Model (BCMLS)

В моделі Belief-Consistent Multilevel Secure Data Model (BCMLS) кожному атрибуту надається мітка безпеки. Мітка безпеки може складатися з однієї чи декількох літер. Перша літера визначає рівень безпеки, на якому значення атрибуту було створено, і називається первинним рівнем безпеки цього атрибуту.

Ті літери, що записані після першої літери мітки, називаються вторинними рівнями. Вони показують рівні безпеки тих користувачів, які довіряють наведеній інформації. Літери повинні стояти у порядку зростання, тобто кожна наступна літера повинна задавати вищий рівень безпеки, ніж попередня.

Також якщо перед літерами стоїть символ заперечення (–), то це буде означати, що користувачі з такими рівнями безпеки не довіряють наданій інформації. А якщо такого символу немає, то довіряють.

Кортеж в цілому теж має мітку безпеки. Користувач бачить його, тільки якщо його мітка безпеки присутня в мітці безпеки кортежу. Як і атрибутам, користувачі можуть або довіряти, або не довіряти кортежам. Важливо, що кортеж нижчого рівня, який ідентичний кортежу на більш високому рівні, вважається хибним.

Розглянемо побудову мітки безпеки на прикладі. Якщо мітка безпеки кортежу має вигляд “U -S TS” позначає, що: первинний рівень безпеки U, користувач рівня S не довіряє даному кортежу, а користувач рівня безпеки TS довіряє кортежу.

Первинний рівень безпеки в даній моделі визначається як рівень, на якому кортеж додали до відношення. І цей кортеж називається первинним кортежем. Вважаємо, що користувач довіряє інформації, якщо його мітка безпеки дорівнює рівню безпеки атрибута даних.

Загалом, користувач бачить та довіряє інформації у базі даних, яка знаходиться на його рівні. І також він може отримати доступ до вмісту бази даних на більш низьких рівнях.

Модель BCMLS має перевагу у тому, що інформацію не потрібно тиражувати, якщо у кортежа мітки атрибутів однакові.

Роглянута модель є більш повною за інші, але вона дуже складна в реалізації, тому повністю ніколи не була реалізована ([8]).

2.5.1 Процедури в моделі BCMLS

2.5.1.1 Процедура Xview (Label)

Ця процедура повертає видиму частину мітки безпеки для користувача, який її виконує.

Спочатку у якості вхідних даних процедури отримується мітка безпеки., яка ділиться на частини. Далі отримується рівень безпеки суб’єкта, який виконує операцію. Створюється лічильник, що дорівнює одиниці. Далі за допомогою збільшення лічильника створюється нова мітка безпеки з тих частин, які може бачити суб’єкт. Вона і повертається по закінченню процедури.

2. 5. 1. 2 **Процедура Pl (Label)**

Ця процедура повертає первинний рівень мітки безпеки.

Отримується мітка безпеки, яка потім розділяється на частини. Далі повертається первинний рівень безпеки, який відповідає першій частині мітки, якій довіряють.

2. 5. 1. 3 **Процедура Sl (Label)**

Ця процедура повертає вторинний рівень мітки безпеки.

Отримується вхідним параметром мітка безпеки. Вона розбивається на частини. Первинний рівень безпеки відкидається, а решта повертається як результат процедури.

2. 5. 1. 4 **Процедура Ib (Label)**

Ця процедура повертає значення, яке показує, чи довіряє суб’єкт з певним рівнем безпеки інформації з певною міткою безпеки.

Отримується мітка безпеки та ділиться на частини. Далі отримується рівень безпеки суб’єкта, який виконує операцію. Створюється лічильник зі значенням 1. І поки його значення не буде дорівнювати рівню безпеки суб’єкта, вихідне значення буде дорівнювати mod (рівень безпеки, кількість рівнів безпеки), а мітка безпеки буде дорівнювати її значенню, поділеному на кількість рівнів безпеки. Інакше процедура повертає останнє вихідне значення.

2. 5. 2 Операція SELECT в моделі BCMLS

Команда операції SELECT в даній моделі виглядає як у схемі (2.1).

Спочатку беруться усі кортежі, значення Xview яких для суб'єкта, що виконує операцію, не дорівнює нулю та які відповідають умові P. Потім для кожного атрибуту цих кортежів отримується значення процедури Xview.

2. 5. 3 Операція INSERT в моделі BCMLS

Команда цієї операції має вигляд як (2.2).

Алгоритм операції має такий вигляд:

- На першому кроці отримується рівень безпеки суб'єкта, який виконує операцію.
- Якщо атрибут таблиці входить у список наведених у команді, то цьому атрибуту встановлюється відповідне значення. Якщо атрибут не вказаний, то йому надається нульове значення.
- Рівень безпеки усіх атрибутів буде дорівнювати рівню безпеки суб'єкта.
- Новий кортеж додається до багаторівневих відношень.

2. 5. 4 Операція VERIFY в моделі BCMLS

Команда операції VERIFY має такий вигляд:

VERIFY (TRUE/FALSE) R WHERE P (2.9)

де R – відношення;

P – умова, яка визначає, які кортежі повинні перевірятися.

Ця операція надає можливість суб'єкту вказати, якім кортежам він довіряє або ні.

Алгоритм цієї операції має такий вигляд:

- Першим кроком ми отримуємо рівень безпеки суб’єкта, який виконує операцію.
- На другому кроці для кожного атрибуту кортежа, для якого виконується умова P, розбити мітку безпеки на частини.
- Для кожної отриманої на попередньому кроці частини, якщо вона дорівнює рівню безпеки суб’єкта, то буде підтверджуватися довіра суб’єкту.
- Створити нову мітку безпеки атрибуту.
- Повернути мітку безпеки атрибуту.

2.5.5 Операція UPDATE в моделі BCMLS

Команда операції UPDATE в моделі BCMLS будується за схемою (2.3).

Алгоритм складається з таких кроків:

- Крок 1. На першому кроці отримується рівень безпеки суб’єкта, який виконує операцію.
- Крок 2. Якщо результат процедури Pl від мітки безпеки кортежу дорівнює рівню безпеки суб’єкта, то кортеж оновлюється.
- Крок 3. Перевіряється чи результат процедури Pl від мітки безпеки кортежу менше рівня безпеки суб’єкта, та результат процедури Ib від рівня безпеки кортежа та рівня безпеки суб’єкта дорівнює рівню безпеки суб’єкта. Тоді будується новий кортеж на основі даного, та вставляється на рівні безпеки суб’єкта. Значення атрибутів кортежа, який перевірявся, не змінюються.
- Крок 4. Перевіряється чи результат процедури Pl від мітки безпеки кортежу менше рівня безпеки суб’єкта, та результат процедури Ib від рівня безпеки кортежа та рівня безпеки суб’єкта дорівнює нулю. У такому випадку якщо існують інші кортежі з відповідними первинними ключами та їх первинними рівнями безпеки, які менше рівня безпеки користувача, то серед усіх таких

кортежів (включаючи перший обраний) користувач обирає кортеж, на базі якого створюється новий кортеж на рівні користувача. Значення атрибутів обраного кортежа не змінюються.

- Крок 5. Якщо результат процедури Pl від мітки безпеки кортежу менше рівня безпеки суб'єкта, та результат процедури Ib від рівня безпеки кортежа та рівня безпеки суб'єкта дорівнює нулю, але не існує інших подібних кортежів, то створюється новий кортеж на рівні суб'єкта на основі обраного кортежа. Значення атрибутів обраного кортежа не змінюються.

2.5.6 Операція DELETE в моделі BCMLS

Команда операції DELETE в моделі будується за схемою (2.4).

Операція виконується за таким алгоритмом:

- Спочатку отримуємо рівень безпеки суб'єкта, який виконує операцію.
- Якщо результат процедури Sl(Xview(мітка безпеки кортежу)), то кортеж видаляється.
- Якщо результат процедури не нульовий, вказується, що суб'єкт не довіряє цьому кортежу та встановити флаг для вищих рівнів.
- Встановлюємо помітку про недовіру для всіх кортежів, які мають відповідний первинний ключ та рівень безпеки як видалений кортеж, а також результат процедури Ib від рівня безпеки кортежа дорівнює l(belief: false).

Висновки до розділу 2

Наразі відомі п'ять моделей багаторівневої безпеки: модель Secure Data Views (SeaView), модель Jajodia-Sandhu, модель Smith-Winslett, модель MultiLevel Relation (MLR) та модель Belief-Consistent Multilevel Secure Data Model (BCMLS).

Модель SeaView потребує великої кількості ресурсів, може призводити до появи хибних кортежів та може бути використана не в усіх випадках. Модель Jajodia-Sandhu має проблеми неповноти операцій та семантичної двозначності. В моделі Smith-Winslett дуже обмежений набор об’єктів для оновлення. Модель MLR не передбачає можливості недовіри до кортежів. В моделі BCMLS вирішені усі можливі проблеми, але вона є дуже складною та ніколи не була повністю реалізована.

3 ПОРІВНЯЛЬНИЙ АНАЛІЗ МОДЕЛЕЙ БАГАТОРІВНЕВОЇ БЕЗПЕКИ

3. 1 Вибір параметрів порівняння

Для порівняння моделей багаторівневої безпеки скористаємося методом прийняття рішень на основі безпосередньої експертної оцінки. Метою аналізу та порівняння побудованих моделей ми будемо вважати вибір найкращої для реалізації моделі багаторівневої безпеки. Цей аналіз буде базуватися на певних проблемах, які присутні в цих моделях, та особливостях їх реалізації.

Нехай $X = \{x_1, x_2, x_3, x_4, x_5\}$ – множина наших альтернатив, де:

x_1 – модель SeaView;

x_2 – модель Jajodia-Sandhu;

x_3 – модель Smith-Winslett;

x_4 – модель MLR;

x_5 – модель BCMLS.

Для порівняння даних моделей визначимо параметри, за якими будемо здійснювати аналіз. Вони будуть базуватися на недоліках моделей та особливостях їх реалізації. Нас цікавлять проблеми, які впливають на рівень забезпеченості безпеки моделі, а також кількість ресурсів, які потрібні для впровадження моделі. Також моделі багаторівневої безпеки обмежені вимогами до безпеки моделі Белл-ЛаПадула, які запобігають появі прихованих каналів між різними рівнями безпеки.

Перший параметр, який ми будемо використовувати для аналізу – це вирішеність проблеми надлишковості даних. Тобто, ми визначаємо, які моделі з побудованих мають проблему надлишковості даних. Вона полягає в тому, що ми зберігаємо однакові дані з різними рівнями безпеки у декількох різних кортежах.

У таблиці 3.1 ми бачимо три однакових кортежі даних з різними загальними рівнями безпеки. Тобто, дані одночасно стосуються різних класифікаційних

рівнів. У цьому випадку ми бачимо проблему надлишковості даних. Користувач вищого рівня побачить одразу декілька однакових кортежів.

Таблиця 3.1 – Приклад проблеми надлишковості даних.

Співробітник	Відділ	Заробітна плата	Загальний рівень безпеки
Іван U	Продажу U	10000 U	U
Іван U	Продажу U	10000 U	S
Іван U	Продажу U	10000 U	TS

Проблема надлишковості даних може призводити до ускладнення керування базою даних, збільшує ризик пошкодження даних та збільшує необхідний обсяг сховища.

Другим параметром ми візьмемо проблему семантичної двозначності. Розглянемо випадок, коли існують відношення з однаковим первинним ключем не для всіх рівней безпеки. Тоді якщо користувач має мітку безпеки, для якої не існує відношення, і існують декілька відношень нижчих рівнів, то потрібно вирішувати, яке відношення відображати. Тобто присутня семантична двозначність.

Розглянемо на прикладі. У таблиці 3.2 ми бачимо, що існують два кортежі з рівнями безпеки U та S. Припустимо, що користувач з рівнем безпеки TS захоче отримати дані про співробітника. Тоді будуть виведені обидва кортежі і користувач не зможе вирішити, який з отриманих кортежів нижчих рівнів є достовірним.

Таблиця 3.2 – Приклад проблеми семантичної двозначності.

Співробітник	Відділ	Заробітна плата
Іван U	Продаж U	10000 U
Іван U	Менеджмент S	12000 S

Проблема обмеження обсягу оновлення – це третій параметр порівняння, який ми будемо використовувати. Вона полягає в тому, що користувач має деякий обмежений обсяг даних, які він може оновлювати. Зокрема, ця проблема присутня у моделі Smitt-Winslett, де користувач може додавати, оновлювати та видаляти кортежі лише на своєму рівні безпеки.

Четвертим параметром є проблема недовіри до кортежів. Ця проблема полягає у відсутності в багатьох моделях можливості недовіряти кортежам. Тобто, користувач вищого рівня не може визначити кортежі нижчих рівнів, яким він не довіряє. Ця проблема присутня у багатьох моделях. Зокрема, у моделі MLR користувач має можливість позначити, якім кортежам він довіряє. Але користувач не може позначити кортежі, які він вважає хибними.

П’ятим параметром ми будемо вважати простоту побудови моделі. Цей параметр є дуже важливим через те, що ми повинні розуміти доцільність реалізації певної моделі відносно кількості ресурсів, яку вона потребує для реалізації. Складність моделі прямо залежить від складності алгоритмів, які використовуються для її реалізації.

Працездатність моделі відносно кількості рівнів безпеки ми будемо вважати шостим параметром. Це обумовлено тим, що одним з важливих для нас факторів є можливість реалізації за потреби такої кількості рівнів безпеки, що нам потрібна. А серед розглянутих моделей є такі, швидкодія яких сильно погіршується при збільшенні кількості рівнів безпеки.

Сьомий параметр для нашого аналізу – працездатність моделі відносно кількості атрибутів у запиті. При збільшенні кількості атрибутів у запиті база даних буде витрачати більше часу для його обробки. Тобто нам важливо, щоб при збільшенні кількості атрибутів, час відповіді бази був достатній для зручної роботи з нею.

Восьмим параметром ми беремо працездатність моделі відносно кількості записів у базі даних. Саме кількість записів впливає на розмір нашої бази даних. В свою чергу, це впливає на час обробки кожної транзакції. І для зручної роботи

користувача з базою даних цей час не повинен бути досить великим. Тобто ми розглянемо як збільшення кількості записів впливає на час відповіді бази даних користувачеві.

Отже, C ={c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8} – множина параметрів оцінювання, де:

c_1 – проблема надлишковості даних;

c_2 – проблема семантичної двозначності;

c_3 – проблема обмеження обсягу оновлення;

c_4 – проблема недовіри до кортежів;

c_5 – простота побудови моделей;

c_6 – працездатність моделі відносно кількості рівнів безпеки;

c_7 – працездатність моделі відносно кількості атрибутів у запиті;

c_8 – працездатність моделі відносно кількості записів у базі даних.

3.2 Оцінка значень параметрів для кожної моделі

Будемо використовувати таку систему оцінки параметрів. Для кожної моделі показник певного параметру повинен бути визначений в межах від 1 до 10, де 1 – найгірше значення параметру, а 10 – найкраще значення.

3.2.1 Проблема надлишковості даних

Визначемо значення даного параметру для кожної з розглянутих моделей багаторівневої безпеки. Ці значення ми будемо обумовлювати особливостями алгоритмів моделей та їх реалізації.

Розглянемо модель SeaView. В цій моделі надлишковість даних витікає з наявної багатозначності сутності. Коли користувачі більш високого рівня отримують результат запиту SELECT, вони можуть отримувати декілька однакових кортежів з різним рівнем безпеки. Також алгоритм відновлення багатоелементних кортежів у цій моделі може призвести до появи багатьох

хибних кортежів, що ще більше може заплутати користувачів. Тому ми вважаємо, що показник для цього параметру дорівнює: $c_1(x_1)=1$.

Модель Jajodia-Sandhu, яка побудована на базі моделі SeaView, вже вирішує цю проблему завдяки покращенню алгоритмів декомпозиції та відновлення. Показник параметру для цієї моделі дорівнює: $c_1(x_2)=10$.

Також ця проблема відсутня у моделі Smith-Winslett через особливості побудови в ній кортежів. А саме через відсутність рівнів безпеки у вторинних атрибутів. Отже, показник щодо наявності проблеми надлишковості даних для цієї моделі дорівнює: $c_1(x_3)=10$.

Для моделі MLR через відсутність проблеми надлишковості даних показник теж дорівнює: $c_1(x_4)=10$. Це досягається завдяки тому, що на вищих рівнях зберігаються покажчики на дані нижчих рівнів.

Модель BCMLS може вирішити проблему надлишковості даних, якщо дані мають вигляд як у таблиці 3.1, тоді дані будуть зберігаться в такому вигляді, як у таблиці 3.3. Але якщо дані мають вигляд як у таблиці 3.4, то ми будемо вимушені зберігати усі три кортежі. Але це майже не впливає на загальну працездатність, тому показник цього параметру для моделі буде дорівнювати: $c_1(x_5)=8$.

Таблиця 3.3 – Рішення проблеми надлишковості даних у моделі BCMLS.

Співробітник	Відділ	Заробітна плата	Загальний рівень безпеки
Іван U S TS	Продажу U S TS	10000 U S TS	U S TS

Таблиця 3.4 – Другий випадок проблеми налишковості даних у моделі BCMLS.

Співробітник	Відділ	Заробітна плата	Загальний рівень безпеки
Іван U S TS	Менеджменту U -S -TS	10000 U S -TS	-U S TS
Іван U S TS	Продажу -U -S TS	10000 -U -S -TS	S -TS
Іван U S TS	Продажу U -S TS	10000 S TS	S TS

Отже, у таблиці 3.5 ми бачимо значення параметру c_1 для кожної моделі.

Таблиця 3.5 – Значення показників параметру c_1.

Модель	Показник параметру c_1
SeaView	$c_1(x_1)=1$
Jajodia-Sandhu	$c_1(x_2)=10$
Smith-Winslett	$c_1(x_3)=10$
MLR	$c_1(x_4)=10$
BCMLS	$c_1(x_5)=8$

3. 2. 2 **Проблема семантичної двозначності**

Розглянемо модель SeaView. В цій моделі можлива ситуація, коли існують кортежі, як у таблиці 3.2. Тоді якщо користувач рівня S чи рівня TS зробить запит SELECT, то він побачить обидва кортежі і не зможе визначити, якому довіряти. З цього випливає, що показник параметру дорівнює: $c_2(x_1)=1$.

Для моделі Jajodia-Sandhu ця проблема також є актуальною. Можливості вказати, яким кортежам довіряти, немає. Отже, під час виводу запиту SELECT користувачем рівня S даних вигляду як у таблиці 3.2, він не зможе визначити,

інформація якого кортежу достовірна. Тому показник даного параметру дорівнює: $c_2(x_2)=1$.

У моделі Smith-Winslett ця проблема вже відсутня. Це зумовлено тим, що ця модель вважається заснованою на довірі (belief-based). Користувач при виводі запиту SELECT бачить кортежі і свого, і нижчих рівней безпеки, але довіряє лише інформації свого рівня. Тому двозначності у визначенні того, інформація якого кортежу вірна, не існує. Показник параметру для цієї моделі дорівнює: $c_2(x_3)=10$.

Проблема відсутня також і в моделі MLR. Ця модель створена на основі моделі Jajodia-Sandhu, але семантична двозначність не з'являється, бо за допомогою команди UPLEVEL з'являється можливість вказати, яким кортежам можна довіряти. Показник для моделі MLR дорівнює: $c_2(x_4)=10$.

Так само, у моделі BCMLS теж не існує проблеми семантичної двозначності. Це зумовлено тим, що модель передбачає можливість довіри та недовіри до кортежів. Тому проблеми у визначенні, яка інформація достовірна, не існує. Для цієї моделі показник дорівнює також: $c_2(x_5)=10$.

У таблиці 3.6 ми бачимо значення параметру c_2 для кожної моделі.

Таблиця 3.6 – Значення показників параметру c_2.

Модель	Показник параметру c_2
SeaView	$c_2(x_1)=1$
Jajodia-Sandhu	$c_2(x_2)=1$
Smith-Winslett	$c_2(x_3)=10$
MLR	$c_2(x_4)=10$
BCMLS	$c_2(x_5)=10$

3. 2. 3 Проблема обмеження обсягу оновлення

Проблема обмеження обсягу оновлення відсутня у моделі SeaView. Користувач може оновлювати інформацію на його рівні безпеки та нижчих, але рівень безпеки змінених атрибутів буде дорівнювати рівню безпеки користувача. Тоді показник даного параметру визначаємо так: $c_3(x_1)=10$.

У моделі Jajodia-Sandhu ця проблема також відсутня. Обсяг оновлення відповідає обсягу оновлення у моделі SeaView. Отже, показник параметру дорівнює: $c_3(x_2)=10$.

Розглянемо модель Smith-Winslett. В цій моделі на відміну від попередніх обсяг оновлення обмежений. Користувач може оновлювати кортежі лише на своєму рівні безпеки. Це зумовлено тим, що сутність у цій моделі визначається парою (PK, C_{PK}). Тому показник щодо проблеми обмеження обсягу оновлення дорівнює: $c_3(x_3)=1$.

У моделях MLR та BCMLS дана проблема відсутня. Користувач може оновлювати інформацію на своєму та нижчих рівнях, але отриманий кортеж буде мати рівень безпеки, що відповідає рівню безпеки суб'єкта. Тоді показники параметру для цих моделей: $c_3(x_4)=10$, $c_3(x_5)=10$.

У таблиці 3.7 ми бачимо значення параметру c_3 для кожної моделі.

Таблиця 3.7 – Значення показників параметру c_3.

Модель	Показник параметру c_3
SeaView	$c_3(x_1)=10$
Jajodia-Sandhu	$c_3(x_2)=10$
Smith-Winslett	$c_3(x_3)=1$
MLR	$c_3(x_4)=10$
BCMLS	$c_3(x_5)=10$

3. 2. 4 Проблема недовіри до кортежів

Розглянемо модель SeaView. Відповідно до алгоритму даної моделі, користувач не має можливості визначити, яким кортежам він не довіряє. Показник даного параметру для цієї моделі дорівнює: $c_4(x_1)=1$.

У моделі Jajodia-Sandhu ця проблема не була вирішена. Тому показник параметру також дорівнює: $c_4(x_2)=1$.

Модель Smith-Winslett вже вважається заснованою на довіри. Але користувач за замовченням довіряє кортежам на своєму рівні безпеки і не може визначити, яким кортежам він не довіряє. Тому показник щодо проблеми недовіри до кортежів дорівнює: $c_4(x_3)=1$.

Модель MLR передбачає, що користувач може визначити самостійно, яким кортежам нижчого рівня безпеки він може довіряти. Але проблема недовіри до кортежів не вирішена. Показник параметру для цієї моделі дорівнює: $c_4(x_4)=1$.

Модель BCMLS – це єдина модель, де проблема недовіри до кортежів вирішена. Користувач може визначити, яким кортежам він не довіряє. Отже, показник параметру для моделі BCMLS дорівнює: $c_4(x_5)=10$.

У таблиці 3.8 ми бачимо значення параметру c_4 для кожної моделі.

Таблиця 3.8 – Значення показників параметру c_4.

Модель	Показник параметру c_4
SeaView	$c_4(x_1)=1$
Jajodia-Sandhu	$c_4(x_2)=1$
Smith-Winslett	$c_4(x_3)=1$
MLR	$c_4(x_4)=1$
BCMLS	$c_4(x_5)=10$

3. 2. 5 Простота побудови моделі

Значення для цього параметру визначити досить складно. Будемо спиратися на особливості коду та складність операцій у моделях, які були роглянуті в другому розділі.

Розглянемо модель SeaView. Ця модель використовує дуже прості алгоритми. Вони базуються на процесах горизонтальної та вертикальної декомпозиції та відновлення. Операції здійснюються за простою схемою. При реалізації модель додатково потребує лише функції для визначення класу безпеки користувача. Показник даного параметру для цієї моделі дорівнює: $c_5(x_1)=10$.

У моделі Jajodia-Sandhu алгоритми схожі на алгоритми попередньої моделі, але не використовується вертикальна фрагментація. Це дещо ускладнює реалізацію моделі. Але суттєвої різниці у складності побудови моделі не спостерігається. Отже, будемо вважати, що показник параметру дорівнює: $c_5(x_2)=8$.

Модель Smith-Winslett майже не відрізняється за складністю від моделі Jajodia-Sandhu. Проте ця модель передбачає реалізацію довіри користувача до об’єктів, а також при процессі оновлення даних доступ надається до об’єкту, визначеного парою (PK,C_{PK}). Тому показник простоти побудови моделі дорівнює: $c_5(x_3)=6$.

Модель MLR досить складна для побудови у порівнянні до попередніх моделей. В цій моделі з’являється операція UPLEVEL з досить складним алгоритмом. А також через реалізацію “запису вгору” ускладнюються алгоритми операцій UPDATE та DELETE. Показник параметру для цієї моделі визначаємо так: $c_5(x_4)=4$.

Модель BCMLS є дуже складною для реалізацію у повному обсязі. Модель потребує реалізації чотирьох процедур для роботи з мітками безпеки, операції VERIFY. Також у порівнянні з іншими моделями сильно ускладнений алгоритм

операції UPDATE. Отже, показник параметру для моделі BCMLS дорівнює: $c_5(x_5)=1$. Вона є найскладнішою для реалізації серед розглянутих моделей.

У таблиці 3.9 ми бачимо значення параметру c_5 для кожної моделі.

Таблиця 3.9 – Значення показників параметру c_5.

Модель	Показник параметру c_5
SeaView	$c_5(x_1)=10$
Jajodia-Sandhu	$c_5(x_2)=8$
Smith-Winslett	$c_5(x_3)=6$
MLR	$c_5(x_4)=4$
BCMLS	$c_5(x_5)=1$

3. 2. 6 Працездатність відносно кількості рівнів безпеки

Працездатність моделей будемо визначати за експериментальними результатами, які були отримані єгипетськими дослідниками([8]). Відповідно, отримані в ході експериментів графіки наведені у додатку А. Для отримання показників даного параметру розглянемо графік залежності часу відповіді моделі в залежності від кількості рівнів безпеки (Рисунок А.1).

У таблиці 3.10 ми бачимо експериментальні дані, які відображені на графіку.

Таблиця 3.10 – Залежність часу відповіді моделі від кількості рівней безпеки.

Модель	Кількість рівнів безпеки				
	2	3	4	5	6
SeaView	1.49	1.89	1.99	2.1	2.39
Jajodia-Sandhu	1.4	1.49	1.7	1.89	2.09
Smith-Winslett	0.08	0.1	0.13	0.15	0.2
MLR	0.17	0.18	0.2	0.23	0.27
BCMLS	0.78	0.83	0.86	1.01	1.1

Час відповіді згідно графіку знаходиться у проміжку [0;2.5]. Значення оцінки працездатності моделі відповідно до кількості рівней безпеки будемо визначати за формулою:

$$o_j(x_i) = 10 - k_j(x_i) * 4 \quad (3.1)$$

де j=2…6 – кількість рівнів безпеки;

x_i – модель безпеки;

$k_j(x_i)$ – час відповіді моделі при певній кількості рівней безпеки.

У таблиці 3.11 ми бачимо значення показника в залежності від кількості рівнів безпеки для кожної з розглянутих нами моделей. Тепер, щоб отримати показник параметру працездатності для кожної моделі, скористаємося такою формулою:

$$c_6(x_i) = \Sigma_{j=2\ldots6} (o_j(x_i)) /5 \quad (3.2)$$

де x_i – модель безпеки;

j=2…6 – кількість рівнів безпеки;

$o_j(x_i)$ – оцінка працездатності моделі відповідно до кількості рівней безпеки.

Результати обчислень наведені у таблиці 3.12.

Таблиця 3.11 – Оцінка працездатності моделей відносно кількості рівней безпеки.

Модель	Кількість рівнів безпеки				
	2	3	4	5	6
SeaView	$o_2(x_1)$=4.04	$o_3(x_1)$=2.44	$o_4(x_1)$=2.04	$o_5(x_1)$=1.6	$o_6(x_1)$=0.44
Jajodia-Sandhu	$o_2(x_2)$=4.4	$o_3(x_2)$=4.04	$o_4(x_2)$=3.2	$o_5(x_2)$=2.44	$o_6(x_2)$=1.64
Smith-Winslett	$o_2(x_3)$=9.68	$o_3(x_3)$=9.6	$o_4(x_3)$=9.48	$o_5(x_3)$=9.4	$o_6(x_3)$=9.2
MLR	$o_2(x_4)$=9.32	$o_3(x_4)$=9.28	$o_4(x_4)$=9.2	$o_5(x_4)$=9.08	$o_6(x_4)$=8.92
BCMLS	$o_2(x_5)$=6.88	$o_3(x_5)$=6.68	$o_4(x_5)$=6.56	$o_5(x_5)$=5.96	$o_6(x_5)$=5.6

Таблиця 3.12. – Значення показників параметру c_6.

Модель	Показник параметру c_6
SeaView	$c_6(x_1)=2.112$
Jajodia-Sandhu	$c_6(x_2)=3.144$
Smith-Winslett	$c_6(x_3)=9.472$
MLR	$c_6(x_4)=9.16$
BCMLS	$c_6(x_5)=6.336$

3. 2. 7 Працездатність відносно кількості атрибутів у запиті

Для отримання показників даного параметру розглянемо графік залежності часу відповіді моделі в залежності від кількості атрибутів у запиті (Рисунок А.2).

У таблиці 3.13 ми бачимо експериментальні дані, які відображені на графіку.

Таблиця 3.13 – Залежність часу відповіді моделі від кількості атрибутів у запиті.

Модель	Кількість атрибутів				
	2	3	4	5	6
SeaView	1.5	3.1	4.3	5.6	7.3
Jajodia-Sandhu	1.5	1.7	1.86	2.02	2.2
Smith-Winslett	0.09	0.15	0.31	0.45	0.55
MLR	0.19	0.23	0.54	0.62	0.88
BCMLS	0.82	0.98	1.31	1.51	1.82

Час відповіді згідно графіку знаходиться у проміжку [0;8]. Значення оцінки працездатності моделі відповідно до кількості атрибутів у запиті будемо визначати за формулою:

$$o_j(x_i) = 10 - k_j(x_i) * 10 / 8 \quad (3.3)$$

де j=2…6 – кількість атрибутів;

x_i – модель безпеки;

$k_j(x_i)$ – час відповіді моделі при певній кількості атрибутів у запиті.

У таблиці 3.14 ми бачимо значення показника в залежності від кількості атрибутів для кожної з розглянутих нами моделей. Тепер, щоб отримати показник параметру працездатності для кожної моделі, скористаємося такою формулою:

$$c_7(x_i) = \Sigma_{j=2\ldots6} (o_j(x_i)) / 5 \quad (3.4)$$

де j=2…6 – кількість атрибутів;

x_i – модель безпеки;

$o_j(x_i)$ – оцінка працездатності моделі відповідно до кількості атрибутів у запиті.

Результати обчислень наведені у таблиці 3.15.

Таблиця 3.14 – Оцінка працездатності моделей відносно кількості атрибутів у запиті

Модель	Кількість атрибутів				
	2	3	4	5	6
SeaView	$o_2(x_1)=8.125$	$o_3(x_1)=6.125$	$o_4(x_1)=4.625$	$o_5(x_1)=3$	$o_6(x_1)=0.875$
Jajodia-Sandhu	$o_2(x_2)=8.125$	$o_3(x_2)=7.875$	$o_4(x_2)=7.675$	$o_5(x_2)=7.475$	$o_6(x_2)=7.25$
Smith-Winslett	$o_2(x_3)=9.888$	$o_3(x_3)=9.813$	$o_4(x_3)=9.613$	$o_5(x_3)=9.438$	$o_6(x_3)=9.313$
MLR	$o_2(x_4)=9.763$	$o_3(x_4)=9.713$	$o_4(x_4)=9.325$	$o_5(x_4)=9.225$	$o_6(x_4)=8.9$
BCMLS	$o_2(x_5)=8.975$	$o_3(x_5)=8.775$	$o_4(x_5)=8.363$	$o_5(x_5)=8.113$	$o_6(x_5)=7.725$

Таблиця 3.15 – Значення показників параметру c_7.

Модель	Показник параметру c_7
SeaView	$c_7(x_1)=4.55$
Jajodia-Sandhu	$c_7(x_2)=7.68$
Smith-Winslett	$c_7(x_3)=9.613$
MLR	$c_7(x_4)=9.385$
BCMLS	$c_7(x_5)=8.39$

3. 2. 8 Працездатність відносно кількості записів у базі даних

Для отримання показників даного параметру розглянемо графік залежності часу відповіді моделі в залежності від кількості записів у базі даних (Рисунок А.3).

У таблиці 3.16 ми бачимо експериментальні дані, які відображені на графіку.

Таблиця 3.16 – Залежність часу відповіді моделі від кількості записів у базі даних.

Модель	Кількість записів у базі даних			
	500	1000	1500	2000
SeaView	1.24	3.82	4.68	7
Jajodia-Sandhu	1.11	1.89	3.19	3.89
Smith-Winslett	0.07	0.08	0.4	0.55
MLR	0.13	0.2	0.67	0.94
BCMLS	0.76	1.11	1.73	2

Час відповіді згідно графіку для моделей, які ми розглядаємо, знаходиться у проміжку [0;8]. Значення оцінки працездатності моделі відповідно до кількості записів у базі даних будемо визначати за формулою:

$$o_j(x_i) = 10 - k_j(x_i) * 10/8 \quad (3.5)$$

де j=1…4 – відповідно номер кількості записів у множині (500, 1000, 1500, 2000);

x_i – модель безпеки;

$k_j(x_i)$ – час відповіді моделі при певній кількості записів у базі даних.

У таблиці 3.17 ми бачимо значення показника в залежності від кількості записів у базі даних для кожної з розглянутих нами моделей. Тепер, щоб

отримати показник параметру працездатності для кожної моделі, скористаємося такою формулою:

$$c_8(x_i) = \Sigma_{j=1\ldots4} (o_j(x_i)) /4 \quad (3.6)$$

де j=1…4 – кількість записів у базі даних;

x_i – модель безпеки;

$o_j(x_i)$ – оцінка працездатності моделі відповідно до кількості записів у базі даних.

Результати обчислень наведені у таблиці 3.18.

Таблиця 3.17 – Оцінка працездатності моделей відносно кількості записів у базі даних.

Модель	Кількість записів у базі даних			
	500	1000	1500	2000
SeaView	$o_1(x_1)$=8.45	$o_2(x_1)$=5.225	$o_3(x_1)$=4.15	$o_4(x_1)$=1.25
Jajodia-Sandhu	$o_1(x_2)$=8.613	$o_2(x_2)$=7.638	$o_3(x_2)$=6.013	$o_4(x_2)$=5.138
Smith-Winslett	$o_1(x_3)$=9.913	$o_2(x_3)$=9.9	$o_3(x_3)$=9.5	$o_4(x_3)$=9.313
MLR	$o_1(x_4)$=9.838	$o_2(x_4)$=9.75	$o_3(x_4)$=9.163	$o_4(x_4)$=8.825
BCMLS	$o_1(x_5)$=9.05	$o_2(x_5)$=8.613	$o_3(x_5)$=7.838	$o_4(x_5)$=7.5

Таблиця 3.18 – Значення показників параметру c_8.

Модель	Показник параметру c_8
SeaView	$c_8(x_1)$=4.769
Jajodia-Sandhu	$c_8(x_2)$=6.8505
Smith-Winslett	$c_8(x_3)$=9.657
MLR	$c_8(x_4)$=9.394
BCMLS	$c_8(x_5)$=8.25

3. 3 Визначення ваги кожного параметру

Нехай W=(w_1, w_2, w_3, w_4, w_5, w_6, w_7, w_8), де значення w_i – значення ваги і-го параметру від 1 до 5. Визначимо ці значення.

Проблема надлишковості даних впливає на швидкість роботи бази даних, ускладнює керування нею, збільшує потрібний обсяг сховища. Значення ваг параметру визначаємо як: w_1=4.

Семантична двозначність не так сильно впливає на повноту функціоналу та працездатність сховища, але є досить неприємною для користувача. Тоді, w_2=3.

Обмеженість обсягу оновлення зменшує зручність роботи з базою даних та функціонал. Користувачу не буде зручно працювати з системою. Відповідно, w_3=4.

Проблема недовіри до кортежів не має великого значення через те, що її наявність майже не впливає на працездатність моделі. І ця може бути вирішена шляхом певних логічних припущень. Наприклад, припущенням, що ті кортежі, до яких явно не показана довіра користувача, є хибними. Значення ваги параметру: w_4=1.

Простота побудови моделі є найважливішим параметром вибору моделі, бо при реалізації для приватної компанії чи державної критичної інфраструктури має велике значення кількість ресурсів, які для цього будуть потрібні. Ресурсами вважаємо час реалізації, потрібна кількість співробітників, потрібні професійні навички тощо. Також складність моделі впливає на ціну реалізації. Тоді значення ваги параметру: w_5=5.

Працездатність моделі відносно кількості рівней безпеки не є досить важливим параметром, бо не кожна компанія чи організація потребує широкого розгалуження за рівнями безпеки. Найчастіше використовуються три чи чотири рівні безпеки. Тоді: w_6=2.

Працездатність відносно кількості атрибутів у запиті є більш важливим параметром через можливість зберігати у базі даних таблиці з більшою кількістю атрибутів та коректно з ними працювати. Відповідно, $w_7=4$.

Працездатність відносно кількості записів у базі даних є одним з наважливіших параметрів. Найчастіше компанії, які потребують використання моделей багаторівневої безпеки, вимушені зберігати великі обсяги даних. І саме через це потребують розгалуження за рівнями доступу. Тобто, $w_8=5$.

Нормуємо значення ваги. Сума значень дорівнює 28. Тоді значення ваг параметрів: w1=4/28=0.143, w2=3/28=0.107, w3=4/28=0.143, w4=1/28=0.035, w5=5/28=0.179, w6=2/28=0.071, w7=4/28=0.143, w8=5/28=0.179.

3. 4 Вибір за отриманими даними найкращої моделі

Тепер визначимо оцінку прийнятності для нас кожної моделі за такою формулою:

$o(x_i)=w_1*c_1(x_i) + w_2*c_2(x_i) + w_3*c_3(x_i) + w_4*c_4(x_i) + w_5*c_5(x_i) + w_6*c_6(x_i) + w_7*c_7(x_i) + w_8*c_8(x_i)$

Робимо розрахунки:

$o(x_1)= 0.143*1 + 0.107*1 + 0.143*10 + 0.035*1 + 0.179*10 + 0.071*2.112 + 0.143*4.55 + 0.179*4.769 = 5.159$

$o(x_2)=0.143*10 + 0.107*1 + 0.143*10 + 0.035*1 + 0.179*8 + 0.071*3.144 + 0.143*7.68 + 0.179*6.851 = 6.982$

$o(x_3)= 0.143*10 + 0.107*10 + 0.143*1 + 0.035*1 + 0.179*6 + 0.071*9.472 + 0.143*9.613 + 0.179*9.657 = 7.528$

$o(x_4)$= 0.143*10 + 0.107*10 + 0.143*10 + 0.035*1 + 0.179*4 + 0.071*9.16 + 0.143*9.385 + 0.179*9.394 = 8.355

$o(x_5)$= 0.143*8 + 0.107*10 + 0.143*10 + 0.035*10 + 0.179*1 + 0.071*6.336 + 0.143*8.39 + 0.179*8.25 = 7.229

Відповідно, ми отримуємо результат, що найкращою для реалізації за визначеними нами параметрами та їх ваговими коефецієнтами є модель MLR. Вона не є дуже складною для реалізації та вирішує більшість можливих проблем щодо моделей багаторівневої безпеки.

Висновки до розділу 3

Порівняльний аналіз проведено методом прийняття рішень на основі безпосередньої експертної оцінки. Визначено такі параметри порівняння: проблема надлишковості даних, проблема семантичної двозначності, проблема обмеження обсягу оновлення, проблема недовіри до кортежів, простота побудови моделі, працездатність моделі відносно кількості рівнів безпеки, працездатність моделі відносно кількості атрибутів у запиті, працездатність моделі відносно кількості записів у базі даних. Визначено значення даних параметрів щодо розглянутих моделей, а також вагові коефіцієнти параметрів.

За результатами обчислень найкращою для реалізації моделлю є модель Multilevel Relation.

ВИСНОВКИ

Дипломну роботу було виконано з метою вибору найкращої для реалізації моделі багаторівневої безпеки. Було розглянуто основні принципи багаторівневої безпеки, особливості та недоліки алгоритмів моделей даної політики.

Загалом, політика MLS була створена для роботи з системами баз даних критичних державних установ. Але з плином часу її почали дотримуватися і деякі комерційні підприємства. Тому процес вибору кращої моделі для реалізації стає все важливішим.

Але порівняння моделей здійснювалося лише якісно. Також у 2014 році єгипетськими дослідниками було проведено експеримент щодо визначення швидкодії моделей([9]). Тому було прийняте рішення зробити порівняльний аналіз методом прийняття рішень. Результати цього аналізу зокрема були опубліковані у збірнику тез XIX Всеукраїнської науково-практичної конференції студентів, аспірантів та молодих вчених «Теоретичні і прикладні проблеми фізики, математики та інформатики»

Для вибору найкращої для реалізації моделі багаторівневої безпеки обрано метод прийняття рішень з використанням експертної оцінки. Обрано параметри, визначено їх показники для розглянутих моделей та вагові коефіцієнти параметрів. Відповідно, за результатами аналізу найкраща для реалізації модель – модкль Multilevel Relation.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1) Скакун В. В. Защита информации в базах данных и экспертных системах [Текст]: пособие для студентов фак. радиофизики и комп. технологий / В. В. Скакун. – Минск: БГУ, 2015. – 140 с.

2) Касянчук Н.В. Захист інформації в базах даних [Текст] / Касянчук Н.В., Ткачук Л.М. // Секція управління інформаційною безпекою та консолідованими інформаційними ресурсами: XLVIII Науково-технічна конференція факультету менеджменту та інформаційної безпеки ВНТУ, 22 березня 2019 р. – Вінниця, 2019. – 5 с.

3) Denning D.E. Views for Multilevel Database Security [Текст] / D.E. Denning, S.G. Akl, M. Heckman, T.F. Lunt, M. Morgenstern // IEEE Transactions on Software Engineering. – 1987. – Т.13, №2. – С.129-140.

4) Patricia A. Dwyer Multi-level Security in Database Management Systems [Текст] / Patricia A. Dwyer, George D. Jelatis and Bhavani M. Thuraisingham // Computers and Security – 1987. – Т.6, №3. – С.252-260.

5) Мулеса О.Ю. Інформаційні системи та реляційні бази даних [Текст]: навч.посібник. / О.Ю. Мулеса. – Ужгород: УНУ, 2018. – 118 с.

6) Нечипоренко О. В. Механізми забезпечення захисту баз даних в сучасних СУБД [Текст] / Нечипоренко О. В., Міценко С. А. // Вісник ЧДТУ – 2014. – №3. – С.80-86.

7) IBM Corporation. PDF format manuals for Db2 12 for z/OS [Електронний ресурс] / IBM Corporation. – Режим доступу: https://www.ibm.com/docs/en/db2-for-zos/12?topic=zos-pdf-format-manuals-db2-12

8) Коломыцев М.В. Модели многоуровневой безопасности баз данных [Текст] / Коломыцев М.В., Носок С.А., Мазуренко А.Е. // Захист інформації. – 2018. – Т.20, №1. – С.42-48.

9) Osama S. Faragallah Multilevel Security for Relational Databases [Текст] / Osama S. Faragallah, El-Sayad M. El-Rabaie, Fathi E. Abd El-Samie, Ahmed I. Sallam Hala, S. El-Sayed – 1-е изд. – Auerbach Publications, 2018. – 304 с.

10) Дейт К. Дж. Введение в системы баз данных [Текст] / К. Дж. Дейт. – 8-е изд. – М.: Издательский дом "Вильяме", 2005. – 1328 с.

11) Teresa F. Lunt The SeaView Security Model [Текст] / T. F. Lunt, D. E. Denning, R. R. Schell, M. Heckman, W. R. Shockley // IEEE Transactions on Software Engineering. – 1990. – Т.16, №6. – С.593-607.

12) Jajodia S. A Novel Decomposition of Multilevel Relations Into Single-Level Relations [Текст] / S. Jajodia , R. Sandhu // IEEE Computer Society Symposium on Research in Security and Privacy 20-22 May 1991 – Oakland, 1991. – С.300-313.

13) Sandhu R. The Multilevel Relational (MLR) Data Model [Текст] / Ravi Sandhu, Fang Chen // ACM Transactions on Information and System Security. – 1998. – Т.1, №1. – С.93-132.

ДОДАТОК А

Експериментальні результати дослідження 2014 року

Дослідження проводилось такими науковцями: Osama S. Faragallah, El-Sayad M. El-Rabaie, Fathi E. Abd El-Samie, Ahmed I. Sallam, Hala S. El-Sayed.

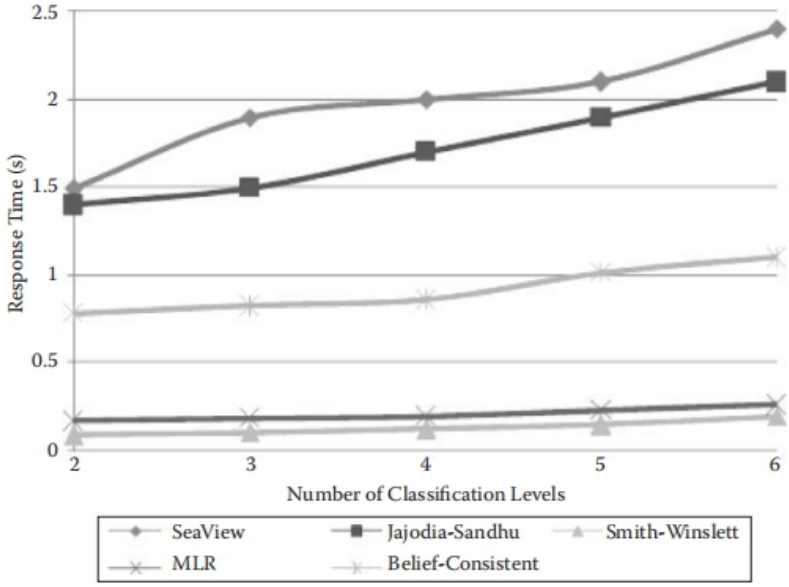

Рисунок А.1 – Залежність часу відповіді бази даних від кількості рівней безпеки.

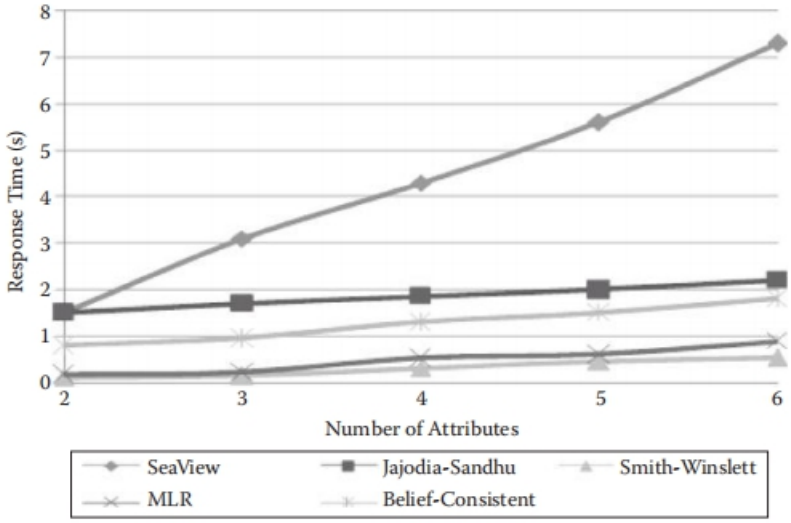

Рисунок А.2 – Залежність часу відповіді бази даних від кількості атрибутів у запиті.

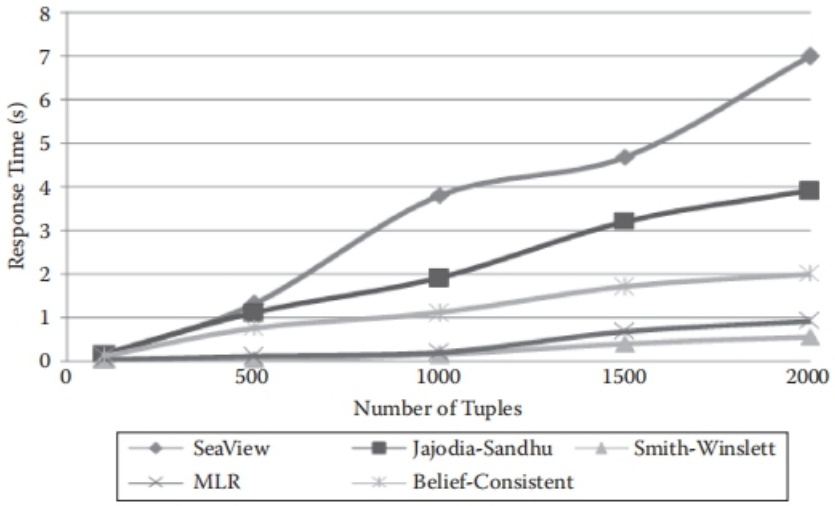

Рисунок А.3 – Залежність часу відповіді бази даних від кількості записів у базі даних.