

Центральноукраїнський національний технічний університет
Механіко-технологічний факультет
Кафедра кібербезпеки та програмного забезпечення

”Допущено до захисту”
Завідувач кафедри кібербезпеки
та програмного забезпечення
д.т.н., професор
_____ Олексій СМІРНОВ
“___” _____ 2026 р.

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА
за першим (бакалаврським) рівнем вищої освіти
на тему
“Програмне забезпечення системи захисту та відновлення
даних у хмарних сервісах”

Виконав здобувач вищої освіти
IV курсу, групи КІ-23-МБ
ОПП «Комп’ютерна інженерія»
спеціальності 123 «Комп’ютерна інженерія»
_____ Почкай О.О.
«___» _____ 2026 р.

Керівник проекту
кандидат фізико-математичних наук, доцент
_____ Якименко Н.М.
«___» _____ 2026 р.
Рецензент _____

Центральноукраїнський національний технічний університет
Факультет Механіко-технологічний
Кафедра Кібербезпеки та програмного забезпечення
Освітній ступінь бакалавр
Галузь знань ._12 _____“Інформаційні технології”
Спеціальність 123 “Комп’ютерна інженерія”
Освітньо-професійна (освітньо-наукова) програма “Комп’ютерна інженерія”

ЗАТВЕРДЖУЮ
Завідувач кафедри
д.т.н., проф.
_____ Олексій СМІРНОВ
« 27 » лютого 2026 року

ЗАВДАННЯ НА ВИПУСКНУ КВАЛІФІКАЦІЙНУ РОБОТУ ЗА ПЕРШИМ (БАКАЛАВРСЬКИМ) РІВНЕМ ВИЩОЇ ОСВІТИ ЗДОБУВАЧА ВИЩОЇ ОСВІТИ

Почкай Олексію Олександровичу

(прізвище, ім’я, по батькові)

1. Тема роботи *Програмне забезпечення системи захисту та відновлення даних у хмарних сервісах*

2. Керівник роботи *Якименко Наталія Миколаївна, кандидат фізико-математичних наук, доцент*

(прізвище, ім’я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу № 126-02 від 27.02.2026 року

3. Строк подання студентом роботи до захисту 23.05.2026 р.

4. Мета та завдання випускної кваліфікаційної роботи: *Метою роботи є розробка програмного забезпечення системи захисту та відновлення даних у хмарних сервісах*

5. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

1. Призначення та область використання.

2. Перегляд аналогічних існуючих систем.

3. Опис і обґрунтування проектних рішень.

4. Етапи програмування системи.

5. Впровадження системи в промислову експлуатацію.

6. Висновки

6. Перелік графічного матеріалу (з точним зазначенням обов’язкових креслень)

Структурна схема системи 1 аркуш

Функціональна схема системи 1 аркуш

Діаграма процесів 1 аркуш

Блок-схема алгоритму роботи додатку 2 аркуша

7. Дата видачі завдання « 27 » лютого 2026 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти	Строк виконання етапів випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти	Примітка
1.	Аналіз існуючих систем	10.03.2026 р.	
2.	Постановка задачі, оформлення ТЗ	15.03.2026 р.	
3.	Розробка моделі компонента	20.03.2026 р.	
4.	Розробка структур даних	25.03.2026 р.	
5.	Розробка алгоритмів зв'язку та відображення	30.03.2026 р.	
6.	Програмування алгоритмів	10.04.2026 р.	
7.	Оформлення ПЗ	17.04.2026 р.	
8.	Попередній захист роботи	23.05.2026 р.	

Дата видачі завдання
« 27 » лютого 2026 р.

Підпис керівника

Якименко Н.М.
(прізвище та ініціали)

Завдання прийнято до виконання
« 27 » лютого 2026 р.

Підпис здобувача

Почкай О.О.
(прізвище та ініціали)

АНОТАЦІЯ

Почкай О.О. Програмне забезпечення системи захисту та відновлення даних у хмарних сервісах. 123 Комп'ютерна інженерія. Центральноукраїнський національний технічний університет. Кропивницький. 2026.

В даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи захисту та відновлення даних у хмарних сервісах.

Метою розробки є програмне забезпечення системи захисту та відновлення даних у хмарних сервісах.

Результат роботи – програмна реалізація системи захисту та відновлення даних у хмарних сервісах.

В процесі роботи над програмною моделлю виконано аналіз існуючих апаратних та програмних засобів. В повній мірі описані всі компоненти розробленого програмного забезпечення.

Розроблено зручний інтерфейс користувача. Наведені інструкції по роботі з програмними засобами.

Програма може використовуватися на ПЕОМ з ОС Windows 10/11.

Програму розроблено в середовищі Visual C#.

Ключові слова: комп'ютерна інженерія, захист та відновлення даних, хмарні сервіси

ABSTRACT

Pochkai O.O. Software for data protection and recovery systems in cloud services. 123 Computer Engineering. Central Ukrainian National Technical University. Kropyvnytskyi. 2026.

In this final qualification work for the first (bachelor's) level of higher education, software has been developed, which is intended for a data protection and recovery system in cloud services.

The purpose of the development is software for a data protection and recovery system in cloud services.

The result of the work is a software implementation of a data protection and recovery system in cloud services.

In the process of working on the software model, an analysis of existing hardware and software was performed. All components of the developed software are fully described.

A convenient user interface has been developed. Instructions for working with software are provided.

The program can be used on a PC with Windows 10/11 OS.

The program was developed in the Visual C# environment.

Keywords: computer engineering, data protection and recovery, cloud services

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ	2
ВСТУП.....	3
1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ	5
1.1 Призначення системи.....	5
1.2 Область застосування.....	8
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ	10
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти.....	10
2.2 Обґрунтування вибору засобів для побудови системи та мови програмування.....	22
2.3 Розгорнута постановка завдання	25
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ	27
3.1 Опис функціонування системи	27
3.2 Розробка структурної схеми.....	32
3.3 Розробка функціональної схеми	42
3.4 Розробка діаграми процесів.....	60
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ.....	62
4.1 Розробка блок-схем та опис алгоритмів функціонування системи.....	62
4.2 Захист розробленого програмного забезпечення.....	78
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ	81
6 ОСНОВНІ ВИСНОВКИ.....	87
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	89

					ВКРБ-123.26.0037.00.00.ПЗ			
Вим.	Арк.	№ докум.	Підп.	Дата	Програмне забезпечення системи захисту та відновлення даних у хмарних сервісах	Літ.	Аркуш	Аркушів
Розроб.	Почкай О.О.					Б	1	99
Перев.	Якименко Н.М.							
Н.контр.	Коваленко А.С.					ЦНТУ КІ-23-МБ		
Затв.	Смірнов О.А.							

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

ВМ	—	Віртуальна машина
НП	—	Надзвичайна подія
ОС	—	Операційна система
ПЗ	—	Програмне забезпечення
СЗД	—	Системи зберігання даних
ЦОД	—	Центр обробки даних
AWS	—	Amazon Web Services
BIOS	—	Basic Input-Output System
CDP	—	Continuous Data Protection – технології безперервного захисту даних
LINQ	—	Language Integrated Query
RAID	—	Redundant Array of Inexpensive Disks
RPO	—	Recovery Point Objective – цільова точка відновлення
RTO	—	Time Objective – цільовий час відновлення

ВСТУП

Актуальність теми. Системи захисту та відновлення даних у хмарних сервісах актуальні у наступних випадках. Втрата периметра безпеки: Компанії втратили контроль над визначеним периметром безпеки через залежність від «агентського» штучного інтелекту та сотень сторонніх SaaS-інструментів. Зловмисники зосередилися на компрометації ідентифікаційних даних, а не на мережевих зламах. Зловмисники використовують технології штучного інтелекту для автоматизації обходу багатофакторної автентифікації (MFA) за допомогою захоплення сеансів та атак соціальної інженерії. Організації стикаються з проблемою ручного керування мільярдами нелюдських ідентифікаційних даних, які дедалі частіше отримують надмірні привілеї. Щоб пережити 2026 рік, організаціям потрібно буде переключити увагу зі статичних аудитів безпеки в певний момент часу на безперервне управління станом даних та управління, орієнтоване на автоматизацію, для захисту даних у русі в безмежній багатохмарній екосистемі. Організації втратили контроль над тим, де зберігаються їхні конфіденційні дані. Це створило величезну прогалину у видимості, коли конфіденційні дані «приховані» та некеровані.

Хмарним обчисленням зараз близько 20 років. Протягом багатьох років компанії впроваджували хмарні технології як частину своєї інфраструктури.

Варто зазначити, що значний обсяг даних, який організації продовжують переносити в хмару. Згідно зі звітом Hacker News « SaaS Backup and Recovery Report in 2025» («Звіт про резервне копіювання та відновлення SaaS за 2025 рік»), 54 відсотки робочих навантажень функціонують у хмарних середовищах. Експерти прогнозують, що до 2026 року цей відсоток зросте до 61 відсотка.

Згідно зі статистикою хмарних обчислень G2, 65 відсотків підприємств планують збільшити свої хмарні бюджети. Крім того, згідно з дослідженням IDC, очікується, що витрати на публічну хмару подвоїться між 2024 і 2028 роками.

					ВКРБ-123.26.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		3

Очікується, що світові інвестиції в SaaS-додатки подвоїться за той самий період часу.

Що це означає для захисту даних. Захист даних у хмарних та SaaS-середовищах незабаром стане ще важливішим, ніж сьогодні. Організації не лише використовуватимуть хмарні та SaaS-додатки для більшої частини своїх робочих навантажень, але й будуть піддаватися дедалі більшому ризику, якщо не матимуть належних рішень для резервного копіювання та відновлення, включаючи резервне копіювання для Microsoft 365 або резервне копіювання для Google Workspace, які захищають деякі з найважливіших для бізнесу середовищ даних SaaS.

Мета й завдання дослідження. Метою роботи є програмне забезпечення системи захисту та відновлення даних у хмарних сервісах.

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

- Огляд існуючих систем захисту та відновлення даних у хмарних сервісах.
- Дослідження системи захисту та відновлення даних у хмарних сервісах.
- Програмна реалізація системи захисту та відновлення даних у хмарних сервісах.

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі захисту та відновлення даних у хмарних сервісах.

Таким чином, виходячи з вищеперерахованого, програмне забезпечення системи захисту та відновлення даних у хмарних сервісах, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти.

					ВКРБ-123.26.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		4

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

Коли ми говоримо про захист даних у хмарі, то ми маємо на увазі... Простої відповіді на це питання немає, тому що він у значній мірі виникає з тих інструментів і технологій, які використовуються. З погляду ІТ-фахівців, достаток потенційних векторів атаки робить захист хмарних систем куди як більше складним, ніж захист традиційних систем. Але це зовсім не виходить, що потрібно планувати захист від всіх цих факторів – при первісному формулюванні моделі погроз значна їхня частина відсівається.

Логіка така:

- По-перше, потрібно розуміти, які моделі захисту даних можуть застосовуватися в хмарі.
- По-друге, визначитися які із завдань захисту інформації будуть вирішуватися самотужки, а які – передаватися на аутсорсинг.
- І по-третє, потрібний хоча б приблизний список інструментів, які будуть використовуватися при перенесенні даних у хмару.
- І на основі цих суджень визначаються завдання, які будуть вирішуватися – усередині компанії й ззовні.

Процеси захисту даних

Резервне копіювання й відновлення, забезпечення безперервності бізнесу, відновлення після НП – все це теми, які напевно сплинуть в обговоренні, що стосується захисту даних. Є й інші терміни – комбінація двох або більше технік. Варто погодити визначення з партнерами й вендорами – якщо ви маєте на увазі під ними різні поняття, то ви можете одержати не зовсім те що хочете.

Резервне копіювання – мабуть, найпростіший в організації процес. Це захист даних дуплікацією, створенням копії даних, які перебувають у роботі. Копія зберігається в окремому сховищі, відділеному від основного логічно й

					ВКРБ-123.26.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		5

фізично (бажано відстанню). У випадку збоїти, робота сервісу/застосунку відновлюється завантаженням резервної копії з наступним її відновленням.

Відновлення в аварійних ситуаціях – це процес відновлення працездатності всієї ІТ-інфраструктури (яка забезпечувала функціонування застосунку/сервісу) після того як інфраструктура стала недоступна на неприпустимий проміжок часу. Відновлюються не тільки дані, але й сервери, застосунки працюючі на серверах, схема мережних взаємодій між серверами й так далі.

Визначення безперервності бізнесу містить у собі як оперативне відновлення, так і відновлення в аварійних ситуаціях. Оперативне відновлення має на увазі відновлення роботи який-небудь із систем ІТ-структури. Це може бути технічна відмова – вихід з ладу жорсткого диска або поломка системи кондиціонування або неполадка, пов'язана з ПЗ – збій мережного протоколу або ушкодження бази даних. Головна відмінність від відновлення в аварійних ситуаціях у тому, що зусилля по відновленню проходять у штатному режимі і як правило не мають на увазі порушень у роботі сервісу – поки йдуть роботи, навантаження переноситься на інші сервери/площадки, у той час як при НП можливо припинення роботи сервісів до усунення НП.

Аварійні ситуації здатні нанести значну втрату репутації, даним і прибуткам, але на щастя вони відносно рідкі, більшість інцидентів можна віднести до ситуацій оперативного відновлення. Хоча деякі процедури, які застосовуються в сценаріях відновлення в аварійних ситуаціях також задіюються при оперативному відновленні, але як правило в цьому немає необхідності. У більшій частині ситуацій, відбудовні роботи обмежуються відновленням даних з резервної копії.

Визначення «хмари» і захист даних

Національний інститут стандартів і технології визначає публічний «хмарний» сервіс як «інфраструктуру, що втримується й підтримується хмарним провайдером і надається для вільного використання всім бажаним». Приватна «хмара», відповідно до визначення того ж NIST, це «хмарна» комп'ютерна

					ВКРБ-123.26.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		6

інфраструктура, надавана хмарним провайдером для ексклюзивного використання єдиною організацією, що включає в себе численних користувачів (наприклад бізнес-підрозділу організації). Вона («хмара») перебуває у власності й управляється організацією, що користується його послугами, третьою особою, або тим і іншим у різному ступені й може розташовуватися на або поза потужностями замовника.

Захист даних може виконуватися як самотужки організацією або передаватися третім особам – провайдерам сервісів резервного копіювання (backup-as-a-service), відновлення даних (recovery-as-a-service) або відновлення в аварійних ситуаціях (disaster as-a-service). Хоча не існує ні однієї компанії, яка б надавала послуги саме під такими назвами, ці моделі існують, просто, як правило, вони включені до складу більшої загальної IaaS, PaaS і так далі. У кожному разі участь третьої сторони в створенні й виконанні схеми захисту має на увазі певну (і чималу) довіра до провайдера, що повинен бути включений до складу хмарної інфраструктури організації, причому бажано із самого початку.

Багато організацій не довіряють хмарній моделі досить, щоб переходити на неї. У такому випадку можливе залучення вендорів у рамках моделі «керованої приватної хмари». Це модель взаємодії, у рамках якої сервіс-провайдер надає певні послуги в рамках своєї хмарної інфраструктури але кожному клієнтові виділяється окремий сервер, індивідуальний набір ПЗ й послуг. Наприклад, послуги захисту даних.

Коли питання про використання хмари в організації тільки обговорюється, то завдання розподілу варто почати з інвентаризації й класифікації виконуваних завдань. Які зараз виконуються й можуть виконуватися тільки на «внутрішніх» потужностях? Які можуть бути передані на аутсорсинг? Які можна винести в «хмару»? Які пріоритети в кожного класу завдань? Як організовується весь процес захисту даних зараз і як він буде організовуватися в новій структурі роботи з даними? Спробуйте категоризувати наявні завдання по цих принципах. Чітке розуміння своїх потреб – те що вам необхідно, коли ви будете проводити переговори з вендорами, інакше можна втратитися й одержати не те що потрібно.

					ВКРБ-123.26.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		7

1.2 Область застосування

Хмарні технології стали основою захисту даних і двигуном розвитку інновацій. Вони дозволяють компаніям більш спокійно переносити коливання економіки, кризи й навіть форс-мажорні ситуації. Привабливість і швидкий ріст попиту на хмари пов'язані з тим, що вони дозволяють швидше й дешевше впроваджувати інновації, і досягати ринкової переваги – наприклад, такого як швидка масштабованість без капітальних вкладень в інфраструктуру, що є однією з найважливіших умов успіху стартапа або просто нового проекту.

Багато керівників не знають, що хмара дозволяє бізнесу продовжувати працювати й зберігати дані “у себе”, і паралельно страхувати бізнес від втрат за допомогою резервного копіювання важливих даних або процесів у європейські дата-центри. Така страховка для даних дозволяє бізнесу відновити роботу в лічені хвилини після рейдерської атаки, простоїв, крадіжок і будь-яких несанкціонованих дій третіх осіб.

Наприклад, порівняно недавно стався випадок із клієнтом з фінансової галузі, що через специфіку своєї діяльності й географічного розташування (дві філії в Донбасі), вчасно запустив план аварійного відновлення даних в одному з дата-центрів у Європі. Після того, як один з філій вийшов з ладу, рішення приватної хмари клієнта дозволило зробити частковий винос даних (технічно більш відомо, як Partial Site Failover) на резервний дата-центр без зупинки діяльності. Після того, як через кілька днів був загублений контроль і над другою філією, всі робітники процеси продовжували здійснюватися на базі аварійного дата-центра в Європі. Менеджмент клієнта був готовий якщо буде потреба зробити перемикання на резервну хмарну інфраструктуру, і зміг продовжити роботу із шифрованого з'єднання з резервним дата-центром, віддалено працюючи в так званому мобільному домашньому офісі mobile home office.

Таким чином, резервне копіювання даних (BaaS) дозволяє істотно знизити ризики простою систем у випадку втрати даних, кібератак або технічних неполадок. Крім цього, хмари вже стали звичним порятунком від рейдерських

					ВКРБ-123.26.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		8

захватів і інших випадків втрати фізичного контролю над даними, і навіть цілим офісом. Завжди можна запустити резервну копію даних у хмарній інфраструктурі, і таким чином, забезпечити безперервність бізнес-процесу у випадку форс-мажорних ситуацій, що для сервісного бізнесу, особливо фінансового, дозволяє уникнути паніки з боку клієнтів і зміцнити репутацію надійної компанії. Безпека даних і бізнесу є першочерговим стимулом попиту на хмарні технології в Україні, у те час як глобальний попит на хмари спонукуваний потребами розвитку й впровадження інновацій.

Таким чином, виходячи з вищеперерахованого, програмне забезпечення системи захисту та відновлення даних у хмарних сервісах, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти.

К6ПЗ_2026

					ВКРБ-123.26.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		9

2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ

2.1 Огляд існуючих систем, технологій, архітектур, програмних рішень за профілем теми випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти

Рішення Dell EMC для резервного копіювання й відновлення в хмарі для AWS

Переваги Dell EMC і AWS

Одержіть максимальну віддачу від інвестицій у сервіси AWS завдяки технологіям для захисту, мобільності даних і керування ними від підрозділу Dell EMC Data Protection Solutions. Amazon Web Services (AWS) і Dell EMC допомагають замовникам по усьому світі розгортати й масштабувати захист даних у хмарі незалежно від розміщення даних і вигідно використовувати експлуатаційні й економічні переваги хмари.

Рішення Dell EMC Data Protection допомагають замовникам захистити інвестиції в хмару:

- Перенос робочих навантажень у хмару для більшої мобільності даних.
- Захист робочих навантажень у хмарі для підвищення відказостійкості даних.
- Резервне копіювання й відновлення в хмарі для експлуатаційної й економічної ефективності.
- Відновлення після збоїв у хмарі для спокою й упевненості.

Більше швидке й економічне резервне копіювання й відновлення в хмарі.

Захист даних у хмарі AWS

Рішення Dell EMC для захисту даних допомагають замовникам одержати максимальну віддачу від інвестицій у загальнодоступну хмару. Замовники

					ВКРБ-123.26.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		10

можуть легко переміщати дані між цільовими системами зберігання AWS для відновлення після збоїв, тривалого зберігання й операційного резервного копіювання. Замовники також можуть захистити хмарні застосунки для прискорення переходу до хмари.

Захист VMware Cloud on AWS

Рішення Dell EMC для захисту даних призначені для сучасних програмно-визначаємих ЦОД з інтеграцією із провідною в галузі технологією VMware. Рішення Dell EMC для VMware Cloud on AWS забезпечують користувачам миттєве відновлення ВМ і масштабованість для задоволення потреб зростаючого віртуального середовища.

Відновлення після збоїв у хмарі

Рішення Dell EMC Data Protection дозволяють замовникам використовувати сховище AWS S3 для захисту даних від природних погроз, людського фактора й зловмисників. Відновлення даних і застосунків безпосередньо в хмарі займає лічені хвилини завдяки прозорій інтеграції з локальною інфраструктурою захисту даних.

Тривале зберігання

Замовники можуть прискорити свою відмову від стрічкових технологій або підвищити ефективність тривалого зберігання даних за допомогою рішень Dell EMC Data Protection і AWS.

Керування захистом даних для AWS

Dell EMC PowerProtect Cloud Snapshot Manager – це рішення, надаване по моделі «ПЗ як послуга», що спрощує для замовників захист робочих навантажень в AWS. При цьому воно не вимагає установки або наявності інфраструктури. Замовники можуть виконувати виявлення, оркестрацію й автоматизацію захисту робочих навантажень, використовуючи політики для безперебійного резервного копіювання й відновлення після збоїв.

					ВКРБ-123.26.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		11

Зниження вартості захисту даних у хмарі за допомогою Dell EMC

Організаціям, які переміщують виробничі застосунки в хмару, часто буває складно виконувати умови своїх угод про рівень обслуговування за допомогою убудованих служб захисту даних, пропонованих постачальниками хмарних рішень. Dell EMC пропонує економічне уніфіковане рішення для захисту корпоративних даних, що централізує, автоматизує й прискорює процес резервного копіювання й відновлення у всієї ІТ-середовищу.

Переваги рішень Dell EMC для захисту робочих навантажень VMware

Перед ІТ-службами й адміністраторами віртуальних середовищ і так коштує безліч завдань, тому захист даних VMware не повинна забирати багато часу й ресурсів. Рішення Dell EMC допомагають упоратися із цим завданням. Вони надають кошти автоматизації, інтеграції з VMware і забезпечують комплексний охват найбільшої екосистеми застосунків локально й у хмарі. Вони також мають високу продуктивність, використовують кращу в галузі функцію дедуплікації, не перевантажують мережу й не вимагають більших витрат для забезпечення захисту.

Модернізація

Перейдіть на сучасний програмно-визначаємий центр обробки даних і захистите своє середовище VMware за допомогою рішення з комплексним охоптом. Це рішення масштабується без накладних витрат, захищає комплексну віртуалізовану екосистему, забезпечує оптимізований захист даних з використанням HCI і інтегрованих пристроїв і знижує вимоги до ємності й пропускної здатності.

Автоматизація

Одержите високу продуктивність і низьку вартість завдяки кращій у галузі автоматизації всього стека захисту даних. Наші рішення забезпечують високу продуктивність із низькою сукупною вартістю володіння й прискорюють резервне копіювання й відновлення, а для розгортання й налаштування проксі-сервера потрібно менш 5 хвилин.

					ВКРБ-123.26.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		12

Трансформація

Скористайтеся ідеальним рішенням, створеним для програмно-визначаємого ЦОД і гібридної хмари. Технологія цього рішення забезпечує розширену інтеграцію із середовищами VMware, захист хмарних робочих навантажень VMware на платформі AWS, а також самообслуговування й консолідацію контролю й автоматизації із захистом даних, що відповідає цільовому рівню обслуговування.

Комплексний, ефективний і безперебійний захист даних для VMware

Рішення Dell EMC для захисту даних:

- створені для програмно-визначаємого ЦОД;
- глибоко інтегровані з користувальницьким інтерфейсом VMware;
- ефективно масштабуються до більшого числа віртуальних машин без втрати продуктивності;
- захищають критично важливі застосунки з інтенсивним введенням-виводом.

Рішення Dell EMC для захисту даних спрощують захист, масштабування й перехід до хмари, а також забезпечують комплексний захист для всього середовища VMware при низькій сукупній вартості володіння.

Програма Future-Proof Loyalty

Програма Future-Proof Loyalty призначена для замовників. Вона гарантує захист інвестицій завдяки набору ІТ-можливостей і програм світового класу, що дозволяють системам зберігання, продуктам для захисту даних і гіперконвергентній інфраструктурі Dell EMC приносити переваги протягом усього терміну служби застосунків. Програма Dell EMC використовує можливості Dell Technologies, щоб запропонувати найкращу й всеосяжну програму в галузі.

Програма Future-Proof Loyalty забезпечує переваги для замовників

Програма Future-Proof Loyalty призначена для забезпечення захисту інвестицій завдяки набору ІТ-можливостей світового класу, що дозволяють

					ВКРБ-123.26.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		13

системам зберігання, продуктам для захисту даних і гіперконвергентній інфраструктурі Dell EMC приносити переваги замовникові.

Гарантія задоволеності

Dell EMC гарантує роботу своїх продуктів відповідно до їхніх технічних характеристик.

Захист інвестицій

Dell EMC пропонує варіанти оптимізації фінансових ресурсів і дозволяє компаніям точно прогнозувати майбутні витрати й коректувати їх у міру необхідності.

Незастаріваюча технологія

Рішення Dell EMC дозволяють безперешкодно адаптувати ІТ-інфраструктуру до майбутніх вимог і одержати доступ до хмарних можливостей.

Пропозиції програми

Трирічна гарантія задоволеності

Здобуваючи сервіс Dell EMC ProSupport, ви одержуєте трирічну гарантію задоволеності для систем зберігання, засобів захисту даних і гіперконвергентної інфраструктури, тоді як галузевий стандарт передбачає гарантію строком 30 днів.

Захист інвестицій в устаткування

Ви завжди можете здати застарілі або конкурентні системи й одержати знижку на системи зберігання, засобу захисту даних або гіперконвергентні продукти Dell EMC наступного покоління.²

Концепція Clear Price

Концепція обслуговування Clear Price від Dell EMC передбачає чітке й прозоре ціноутворення за підтримку з докладною інформацією про підтримку по передоплаті.

Програмне забезпечення за принципом «усе включено»

Все необхідне для зберігання й адміністрування критично важливих даних уже включено в комплект систем зберігання, що здобуваються вами, і гіперконвергентних рішень Dell EMC.

					ВКРБ-123.26.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		14

Безпроблемна міграція даних

Ви можете з легкістю перейти на СЗД і гіперконвергентні рішення наступного покоління Dell EMC завдяки безпроблемній модернізації й убудованим інструментам міграції даних.

Гарантія дедуплікації при захисті даних

Використовуйте рекомендовані кращі практики Dell EMC, щоб забезпечити гарантоване скорочення обсягу даних в 55 разів для пристроїв захисту даних.

Гарантії скорочення обсягу збережених даних

Dell EMC забезпечує гарантований коефіцієнт скорочення обсягу збережених даних для ваших робочих навантажень.

Можливості роботи в хмарі

Dell EMC пропонує засоби хмарної мобільності, послуг, захисти й керування для ресурсів зберігання, рішення для захисту даних і гіперконвергентні продукти.

Використання хмарних ресурсів

Dell EMC надає гнучкі моделі споживання на вимогу для ІТ-інфраструктури від Dell Financial Services (DFS).

НРЕ: система зберігання із захистом даних

Система захисту даних, інтегрована у флеш-накопичувачі й створена для хмари, пропонує виняткову ефективність резервного копіювання, архівування й аварійного відновлення ваших корпоративних застосунків.

Оптимізація засобів захисту даних для гібридних ІТ

Досягнення безпрецедентної гнучкості й масштабованості при захисті критично важливих застосунків завдяки сполученню хмарних і локальних ІТ-рішень. Система захисту даних від НРЕ, інтегрована у флеш-накопичувачі й створена для хмари, забезпечує ефективне використання хмарних функцій резервного копіювання й відновлення, що відрізняються гнучкістю й економічністю, у сполученні з локальною інфраструктурою для забезпечення швидкого й надійного відновлення роботи.

					ВКРБ-123.26.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		15

Захист даних, де б вони не перебували

HPE поєднує краще в галузі ПЗ для захисту інфраструктури й даних із професійними послугами від HPE Pointnext Services, завдяки чому для всього середовища підприємства, від центра обробки даних до хмари, забезпечується комплексна інтегрована основа аварійного відновлення й зменшення ризиків.

Спрощення

Використовуйте рішення для створення інтелектуального автоматизованого багаторівневого рішення для захисту даних на основі політик і з урахуванням потреб підприємства. Забезпечте виконання угоди про рівень обслуговування завдяки найвищій швидкості резервного копіювання в масштабі хмари. Захиститесь від погроз за допомогою надійного рішення резервного копіювання й високодоступної системи зберігання, використовуваної для захисту.

Переміщайте дані в потрібні місця

Забезпечте повну підтримку хмари за допомогою інтегрованого рішення захисту даних. Створюйте для хмари резервні копії тільки унікальних даних, знижуючи витрати на хмарну СЗД в 20 разів. Зберігаєте зашифровані, самоописувані дані для простого аварійного відновлення в хмарі.

Захист даних з урахуванням ваших вимог

Трансформуйте своє рішення резервного копіювання даних, щоб перетворити реактивну політику страхування в ефективну проактивну послугу. Використовуйте екосистему кращих партнерів і модель оплати по факті споживання в масштабованому рішенні, що підійде як для невеликих віддалених офісів, так і для самих великих підприємств і постачальників послуг.

Збереження резервних копій у хмарі

Простота, економічність і безпека резервного копіювання в хмару HPE Cloud Bank Storage дозволяє замовникам HPE StoreOnce використовувати всі переваги недорогої зовнішньої об'єктної СЗД (локальної або хмарної) для тривалого зберігання резервних копій.

					ВКРБ-123.26.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		16

Засоби відновлення, інтегровані у флеш-накопичувачі

Об'єднання служб створення знімків, реплікації й резервного копіювання зі швидкістю роботи флеш-масиву. HPE Recovery Manager Central (RMC) зменшує витрати й складність і прискорює відновлення даних застосунків, які можуть розміщатися як у ЦОД, так і в хмарі. Це рішення задовольняє високим вимогам застосунків, виконуваних на флеш-масивах, і SLA із самими строгими показниками RTO/RPO.

Захист на основі застосунків

Підвищте продуктивність і ефективність дедуплікації для корпоративних застосунків, наприклад SAP HANA, Oracle, SQL, а також систем резервного копіювання незалежних розроблювачів, таких як Commvault, Data Protector, Veritas і Veeam. Скоротить витрати й рівень ризику за рахунок уніфікованого керування.

Пристрій для резервного копіювання й захисту даних HPE StoreOnce

Інтелектуальна система зберігання, що трансформує захист даних у гібридних IT-середовищах і забезпечує простоту, підвищену продуктивність і гнучкість і при цьому коштує дешевше традиційних рішень.

Спрощення захисту даних з гібридних IT-середовищах

Готова для використання в хмарних рішеннях платформа захисту даних дозволяє спростити керування, скоротити витрати й зменшити ризики. HPE StoreOnce забезпечує миттєвий захист центра обробки даних, економічне архівування й відновлення після збоїв у хмарі, а широкі можливості масштабування роблять це рішення підходящою як для малих віддалених офісів, так і для великих підприємств і постачальників послуг.

Останні інновації

Прискорене резервне копіювання й відновлення даних завдяки простому, надійному й доступному рішення HPE для зберігання даних, готовому до використання в хмарі.

					ВКРБ-123.26.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		17

Спрощення операцій

Спростите резервне копіювання в хмарі за допомогою рішення для захисту даних, що спеціально створено для хмари й не вимагає окремого шлюзу або віртуального пристрою. Ви зможете забезпечити повну підтримку хмари в рішеннях резервного копіювання й корпоративних застосунків, забезпечити інтеграцію обраного хмарного рішення, виконувати резервне копіювання в хмару тільки унікальних даних, що дозволить в 20 разів скоротити витрати на зберігання даних у хмарі¹, а також зберігати дані резервних копій у зашифрованому виді зі зрозумілим описом, що спрощує відновлення хмари у випадку збоїти

Виконання угод про рівень обслуговування

Доможіться кращих у галузі показників по зборі даних. Ви зможете в 23 рази прискорити процеси резервного копіювання й в 15 разів – відновлення для корпоративних застосунків з мінімальним впливом на роботу виробничого середовища за допомогою прямого резервного копіювання із систем HPE ZPAR або HPE Nimble в HPE StoreOnce з використанням HPE RMC. Забезпечте контрольований ріст даних і гарантоване скорочення витрат на резервне копіювання й загальну кількість устаткування на 95% завдяки інтелектуальній дедуплікації

Захист даних з урахуванням ваших потреб

Насолоджуйтеся простий і надійний захист даних з оплатою резервного копіювання як послуги з факту використання⁹ і ємністю, що росте разом з вашим бізнесом. Дане рішення можна розвертати в хмарах, віртуальних і фізичних середовищах. Це гарантує захист інвестицій, виключає прив'язку до певного постачальника, а також забезпечує повну інтеграцію у всієї широкої екосистемі незалежних постачальників послуг.

HPE StoreOnce – платформа нового покоління

Забезпечте найвищу продуктивність у масштабі хмари, збільшення ємності до 10 разів і швидкості резервного копіювання до 3 разів. Спростіть

					ВКРБ-123.26.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		18

керування завдяки можливості контролю декількох систем за допомогою однієї панелі. Підвищте оперативність роботи завдяки гнучкій програмно обумовленому захисту.

HPE RMC 6.2

Функції резервного копіювання й керування даними копії, інтегровані в системи HPE ZPAR, HPE Primera і HPE Nimble Storage, забезпечують захист більшого обсягу даних, відрізняються більше високою продуктивністю й меншою вартістю. Необхідні політики для виконання угоди про рівень обслуговування можна створити за допомогою декількох клацань миші.

Інтеграція Commvault

Спростите резервне копіювання хмари завдяки підтримці Commvault для системи зберігання HPE Cloud Bank. Інтеграція Commvault з функціями дедуплікації на стороні джерела HPE StoreOnce і реплікацією з невеликим завантаженням каналу забезпечують більше швидке й ефективне резервне копіювання й відновлення у випадку збоїв.

HPE StoreOnce VSA

Потрібно ефективне резервне сховище з дедуплікацією і без обмежень, характерних для спеціалізованих пристроїв? Якщо так, HPE StoreOnce VSA допоможе задовольнити ваші потреби, маючи гнучкість віртуального пристрою. Програмно обумовлене резервне сховище підтримує всі функції спеціалізованих систем StoreOnce і гарантує сумісність програмного забезпечення для захисту даних. Крім іншого, функції переносу на іншу платформу й доступності в vSphere і Hyper-V підвищують гнучкість і відказостійкість систем. Можна настроїти до 500 Тбайт корисної ємності із кроком 1 Тбайт. Сервер ліцензій спрощує керування ліцензіями у великих або динамічних розгортаннях StoreOnce VSA. Інтелектуальне об'єднане керування через одну консоль ще більше полегшує експлуатацію декількох екземплярів StoreOnce VSA. Реплікація StoreOnce і хмарне сховище Cloud Bank Storage для додаткового захисту дозволяють переносити дані резервних копій за межі підприємства й у малозатратні служби об'єктних систем зберігання даних для тривалого економічного зберігання.

					ВКРБ-123.26.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		19

Нові можливості

- У системі StoreOnce VSA 4.2 розширена підтримка розгортання в середовищах VMware з високою доступністю.
- VSA 4.2 забезпечує автоматичне аварійне перемикання VSA з непрацездатного хоста на хост у режимі очікування при використанні підключень до даних по протоколах IP і FC. Завдяки цьому час простою через відмову хоста скорочується до лічених хвилин.
- Новий віртуальний пристрій StoreOnce High Availability Manager полегшує керування системами StoreOnce VSA у конфігураціях з високою доступністю.

Можливості

Програмно обумовлена резервна система зберігання даних ємністю до 500 Тбайт із HPE StoreOnce VSA.

У порівнянні з попереднім поколінням StoreOnce VSA показники покращилися в 10 разів, що припускає не тільки наявність додаткової ємності, але й ріст продуктивності, числа потоків, джерел реплікації й ресурсів зберігання.

Рішення StoreOnce VSA можна настроїти всього з 4 Тбайт ємності й масштабувати до 500 Тбайт із кроком 1 Тбайт, не переплачуючи за ємність, що знадобиться пізніше зі збільшенням обсягів резервного копіювання.

Доступні й інші віртуальні пристрої для дедуплікації резервної системи зберігання даних, але жодне з них не пропонує 500 Тбайт ємності. Розмір StoreOnce VSA може в 5 разів перевищувати розмір наступного найбільшого віртуального пристрою для дедуплікації.

Гнучке налаштування продуктивності і ємності

Гнучке додавання нової ємності із кроком 1 Тбайт або будь-яким значенням, кратним 1 Тбайт. Ліцензії додаються безпосередньо до екземпляра VSA або із сервера ліцензій HPE StoreOnce VSA.

Програмно обумовлена продуктивність із ресурсами, виділеними для масштабування пропускної здатності, ємності, потоків і числа цільових об'єктів

					ВКРБ-123.26.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		20

резервного копіювання, які необхідні для дотримання графіків резервного копіювання й угод про рівень обслуговування.

Можна використовувати кілька екземплярів VSA, щоб задати різну продуктивність резервного копіювання. Наприклад, можна мати VSA з великою ємністю, але невисокою продуктивністю, або високопродуктивний пристрій VSA з меншою ємністю. Це дозволить оптимально використовувати час, виділений для резервного копіювання.

Вибір гіпервізору й опцій оцінки

Запустите HPE StoreOnce VSA в VMware vSphere і Microsoft Hyper-V* або в обох гіпервізорах за допомогою єдиної консолі, щоб управляти екземплярами VSA. Примітка. Hyper-V у даний момент не підтримується кодом VSA версії 4.2

Пробна версія HPE StoreOnce VSA припускає до 90 днів використання. Вона допоможе оцінити всі можливості цього рішення. Щоб продовжити використання з усіма збереженими конфігураціями, придбайте ліцензію. Див. сторінку <https://www.hpe.com/storage/TryStoreOnceVSA>.

Безкоштовна версія ПЗ StoreOnce VSA на 1 Тбайт для підтримки розширеної оцінки й використання некритичних продуктів з можливістю самопідтримки, яку можна легко оновити для додавання підтримки HPE, хмарного сховища Cloud Bank Storage, шифрування й до 500 Тбайт ємності. Див. сторінку www.hpe.com/storage/freebackup.

Централізоване керування ліцензіями HPE StoreOnce VSA для великих і динамічних розгортань

У динамічних або великомасштабних системах StoreOnce VSA керування ліцензіями може займати багато часу, а сервер ліцензій StoreOnce VSA заощаджує час завдяки централізації керування ліцензіями. Ліцензії додаються на сервер ліцензій AutoPass для формування пула ресурсів, виділюваного підключеним VSA у міру необхідності.

За допомогою сервера ліцензій AutoPass можна повертати ліцензії зі списаних екземплярів StoreOnce VSA на сервер ліцензій і використовувати

					ВКРБ-123.26.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		21

повторно на нових пристроях StoreOnce VSA, щоб скоротити пов'язані з ліцензуванням витрати.

У сполученні з об'єднаним керуванням StoreOnce сервер ліцензій StoreOnce VSA забезпечує керування великими розгортаннями StoreOnce VSA у єдиній консолі.

2.2 Обґрунтування вибору засобів для побудови системи та мови програмування

Програмне забезпечення написане мовою Visual C#. Ця мова обрана виходячи з наступних міркувань. Visual C# – строго типізована об'єктно-орієнтована мова, призначена для розробки різноманітних безпечних і потужних застосунків, виконуваних у середовищі .NET Framework. Мовою Visual C# можна розробляти звичайні клієнтські застосунки Windows, веб-служби XML, розподілені компоненти, застосунки типу “сервер-клієнт”, застосунки баз даних і багато яких інших. В Visual C# є розширений редактор коду, конструктори зі зручним користувацьким інтерфейсом, вбудований відладник і багато інших засобів, покликані спростити розробку застосунків мовою Visual C# версії 5.0 і .NET Framework версії 4.5.

Синтаксис Visual C# дуже виразний, але простий у вивченні. Усі, хто знаком з мовами C, C++ або Java з легкістю визнають синтаксис із фігурними дужками, характерний для мови Visual C#. Розроблювачі, що знають кожную із цих мов, як правило, зможуть домогтися ефективної роботи з мовою Visual C# за дуже короткий час. Синтаксис Visual C# робить простіше те, що було складно в C++, і забезпечує потужні можливості, такі як типи значень Nullable, перерахування, делегати, лямбда-вираження й прямий доступ до пам'яті, чого немає в Java. Visual C# підтримує універсальні методи й типи, забезпечуючи більше високий рівень безпеки й продуктивності, а також ітератори, що дозволяють при реалізації колекцій класів визначати власне поводження ітерації,

					ВКРБ-123.26.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		22

що може легко використовуватися в клієнтському коді. В Visual C# 5.0 вираження LINQ (Language-Integrated Query) роблять строго-типізований запит першокласною конструкцією мови.

Як об'єктно-орієнтована мова, Visual C# підтримує поняття інкапсуляції, спадкування й поліморфізму. Всі змінні й методи, включаючи метод Main – точку входу застосунка – інкапсулюється у визначення класів. Клас може успадковувати безпосередньо з одного родового класу, але може реалізовувати будь-яке число інтерфейсів. Для методів, які перевизначають віртуальні методи в батьківському класі, необхідно ключове слово `override`, щоб виключити випадкове повторне визначення. У мові Visual C# структура схожа на полегшений клас: це тип, що розподіляється по стопках, що реалізує інтерфейси, але не підтримує спадкування.

На додаток до основних описаних об'єктно-орієнтованих принципів, мова Visual C# спрощує розробку компонентів програмного забезпечення завдяки декільком інноваційним конструкціям мови, у число яких входять наступні:

- Інкапсульовані підписи методів, називані делегатами, які підтримують строго-типізовані повідомлення про події.
- Властивості, що виступають у ролі методів доступу для закритих змінних-членів.
- Атрибути з декларативними метаданими про типи під час виконання.
- Вбудовані коментарі XML-документації.
- LINQ (Language-Integrated Query), що пропонує вбудовані можливості запитів у різних джерелах даних.

Якщо буде потрібно забезпечити взаємодію з іншим програмним забезпеченням Windows, таким як об'єкти COM або власні бібліотеки DLL Win32, у мові Visual C# можна використовувати процес, що називається "Interop". Процес Interop дозволяє програмам на Visual C# виконувати практично будь-які дії, які може виконувати вихідний додаток на C++. Мова Visual C# підтримує

навіть покажчики й поняття "небезпечного" коду для тих випадків, коли прямий доступ до пам'яті має вкрай важливе значення.

Процес побудови Visual C# у порівнянні з C і C++ простий і є більше гнучким, ніж в Java. Немає окремих файлів заголовка, а методи й типи не потрібно повідомляти в певному порядку. У вихідному файлі Visual C# може бути визначене будь-яке число класів, структур, інтерфейсів і подій.

Архітектура платформи .NET Framework

Програма мовою Visual C# виконується в середовищі .NET Framework – інтегрованому компоненті Windows, що містить віртуальну систему виконання (середовище CLR) і уніфікований набір бібліотек класів. Середовище CLR являє собою комерційну реалізацію корпорацією Майкрософт інфраструктури CLI, що є міжнародним стандартом, який лежить в основі створення середовищ виконання й розробки, у яких забезпечується тісна взаємодія між мовами й бібліотеками.

Вихідний код, написаний мовою Visual C#, компілюється в проміжну мову (IL) у відповідності зі специфікацією CLI. Код IL і ресурси, такі як растрові зображення й рядки, зберігаються на диску у файлі, що виконується, названому складанням, з розширенням EXE або DLL у більшості випадків. Складання містить маніфест із відомостями про типи складання, версії, мови й регіональні параметри та вимоги безпеки.

При виконанні програми на Visual C# складання завантажується в середовище CLR залежно від відомостей у маніфесті. Далі, якщо вимоги безпеки дотримані, середовище CLR виконує JIT-компіляцію для перетворення коду IL в інструкції машинного коду. Середовище CLR також надає інші служби, що відносяться до автоматичного збору сміття, обробки виключень і керуванню ресурсами. Код, виконуваний середовищем CLR, іноді називають "керованим кодом" у протиставлення "некерованому коду", що компілюється в машинний код, призначений для певної системи. Далі показані відносини під час компіляції

					ВКРБ-123.26.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		24

й час виконання між файлами з вихідним кодом Visual C#, бібліотеками класів .NET Framework, складаннями й середовищем CLR.

Взаємодія між мовами є ключовою особливістю .NET Framework. Оскільки код IL, створюваний компілятором Visual C# відповідає специфікації CTS, код IL на основі Visual C# може взаємодіяти з кодом, створюваним версіями мов Visual Basic, Visual C++, Visual J# платформи .NET Framework і ще більш ніж 20 CTS-сумісних мов. В одному складанні може бути кілька модулів, написаних на різних мовах платформи .NET Framework, і типи можуть посилатися один на одного, як якби вони були написані на одній мові.

Крім служб часу виконання, в.NET Framework також є велика бібліотека, що складається з більш ніж 4000 класів, організованих по просторах імен, які забезпечують різноманітні корисні функції для будь-яких дій, починаючи від введення й виведення файлів для керування рядками для розбивки XML, і закінчуючи елементами керування Windows Forms. У звичайному застосунку мовою Visual C# бібліотека класів .NET Framework інтенсивно використовується для "устрою" коду.

2.3 Розгорнута постановка завдання

Згідно з технічним завданням на випускню кваліфікаційну роботу за першим (бакалаврським) рівнем вищої освіти, реалізації підлягає програмне забезпечення, яке призначено для системи захисту та відновлення даних у хмарних сервісах.

В процесі розробки випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти необхідно виконати наступний обсяг роботи:

а) провести аналіз існуючих систем-аналогів для виявлення їх позитивних і негативних якостей. Результати аналізу врахувати в подальших розробках;

б) вибрати та обґрунтувати методику побудови системи контролю роботи технологічного обладнання на виробництві в автоматизованому режимі.

					ВКРБ-123.26.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		25

Розробити функціональну та структурну схеми системи;

в) розробити програмне забезпечення системи, що дозволить реалізувати поставлену технічним завданням задачу. Побудувати блок-схеми алгоритмів програми та підпрограми;

г) організувати інтерфейс користувача з метою формування та виводу на екран ЕОМ повідомлень про некоректні дії користувача та нестандартні ситуації в роботі технологічного обладнання;

д) розробити рекомендації по організаційних та методичних заходах, які забезпечать впровадження системи в промислову експлуатацію та її подальшу успішну експлуатацію;

е) провести розрахунки по визначенню економічної ефективності розробленої системи;

ж) розробити заходи по охороні праці при впровадженні та експлуатації системи, а також розробити заходи з цивільного захисту;

з) сформулювати висновки про виконаний обсяг робіт та одержані результати.

					ВКРБ-123.26.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		26

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

Розширення ландшафту загроз для даних SaaS

Визначення причини втрати даних є важливою частиною запобігання їй в майбутньому. Коли йдеться про резервне копіювання SaaS, існує багато потенційних причин.

Як зазначалося вище, те саме дослідження, на яке цитується The Hacker News, показує, що 50 відсотків респондентів зазнали зловмисного видалення даних, тоді як 36,7 відсотка постраждали від атаки програм-вимагачів, причому випадкове видалення (34 відсотки) та неправильні налаштування програм (30 відсотків) також значною мірою сприяли втраті даних SaaS.

Оскільки більшість компаній впроваджують SaaS-додатки, ваші дані можуть бути втрачені з будь-якої з перелічених вище причин. Саме тому важливо мати SaaS-рішення для резервного копіювання, щоб захистити свої дані від усього переліченого.

Критичні прогалини в покритті резервного копіювання SaaS

У наведеній вище даних показано, які типи даних найчастіше втрачаються із SaaS-додатків. Але чи знаєте ви, які типи даних ваш бізнес насправді захищає за допомогою резервного копіювання SaaS?

Повний огляд вашого середовища SaaS – один із найкращих способів зрозуміти, які фрагменти даних можуть бути втрачені. Компанії, які впроваджують захист SaaS, найчастіше повідомляли, що вони створюють резервні копії електронної пошти та контактних даних з поштових клієнтів.

Однак вони також рідше створюють резервні копії подій календаря, файлів на спільних дисках або даних командного чату/співпраці. Виключення певних даних зі стратегії захисту SaaS може зробити вашу організацію вразливою

					ВКРБ-123.26.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		27

у випадку, якщо вам знадобиться відновити ці типи даних.

Слабкий захист облікових даних резервного копіювання створює прихований ризик

Якщо хтось отримав ваші резервні облікові дані, він може використати їх для самостійного доступу до даних. Розголошення ваших облікових даних значно збільшить ймовірність їх викрадення.

Згідно з дослідженням безпеки офісів, проведеним компанією Sentry Tech Solutions, та подальшим дослідженням гігієни паролів, проведеним компанією Keeper Security, багато компаній досі носять стікери та паролі в голові. Фактично, лише кожен третій співробітник покладається виключно на менеджери паролів для зберігання своїх паролів.

Відсутність видимості резервних копій підриває готовність до відновлення

Ми вже згадували, що багатьом компаніям бракує розуміння типів даних, які перебувають під загрозою. ІТ-командам також часто бракує розуміння того, чи успішно створено резервне копіювання.

Незалежно від того, чи вважають компанії, що вони захищають структуровані дані, неструктуровані дані чи і те, й інше, ІТ-фахівці все ще бачать мало інформації про рівень успішності резервного копіювання, як показує нове дослідження, опубліковане The Hacker News. 35% опитаних фахівців сказали, що вони не знають, чи є резервне копіювання успішним.

Рішення для моніторингу резервного копіювання можуть автоматично повідомляти та сповіщати ІТ-адміністраторів про стан резервного копіювання, щоб вони могли контролювати його стан. Деякі організації, схоже, не мають рішення для захисту даних з такою можливістю або просто не використовують його.

Чому більшості організацій не вдається швидко відновити дані SaaS

Атаки та втрата даних у хмарі не лише часто трапляються, але й компанії повідомляють про дуже погану здатність до відновлення після цих інцидентів,

					ВКРБ-123.26.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		28

коли вони трапляються:

– Згідно з даними видання The Hacker News, лише 14 відсотків ІТ-керівників впевнені у відновленні критично важливих даних SaaS протягом кількох хвилин, тоді як 25 відсотків кажуть, що відновлення займе кілька днів.

– 25 відсотків сказали, що відновлення триватиме кілька днів

А враховуючи, що дослідження IBM, на яке цитується Northdoor, оцінює середні витрати на простої в понад 300 000 доларів на годину, такі затримки можуть мати значний негативний вплив на бізнес.

Зниження довіри до можливостей резервного копіювання та відновлення

Оскільки на керування резервними копіями витрачається більше часу, довіра до захисту даних зменшується.

Довіра до захисту даних продовжує знижуватися: аналіз, про який йдеться у виданні The Hacker News, показує, що лише 40 відсотків підприємств вважають, що їхні критично важливі дані захищені від втрати, тоді як попереднє дослідження, опубліковане Databarracks, показало, що лише п'ять років тому 50 відсотків організацій описували себе як «дуже впевнені» у своїх можливостях резервного копіювання та відновлення.

Я б сказав, що це привід для кошмарів. Незважаючи на інвестування більших ресурсів у рішення для резервного копіювання та відновлення, компанії відчують меншу впевненість у своїй здатності досягати цілей відновлення.

Зростаючий регуляторний тиск та вартість недотримання вимог

Штрафи за витік даних – це лише один із потенційних фінансових наслідків втрати даних. Організації також ризикують отримати великі штрафи за дотримання вимог в результаті збоїв у захисті даних.

Значною мірою це пов'язано з новими регуляторними стандартами, такими як DORA та NIS 2, які посилять дію у 2026 році. На даний момент регулятори менше стурбовані прийняттям законодавства та більше зосереджені на його забезпеченні шляхом аудитів, наглядових перевірок та штрафів для

					ВКРБ-123.26.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		29

організацій, що не дотримуються вимог.

Згідно з DORA, регуляторні органи можуть стягувати штрафи у розмірі до 2 відсотків від річного доходу та, як зазначають Грант Торнтон та regulation-dora.eu, також можуть притягувати керівників підприємств до особистої відповідальності.

Багато старих нормативних актів щодо дотримання нормативних вимог зосереджувалися переважно на забезпеченні того, щоб підприємства вживали розумних заходів для запобігання кібератак та витоку даних. Але новіші нормативні акти мають ширшу сферу застосування, яка включає резервне копіювання та відновлення даних. Непідготовка до стихійних лих може призвести до штрафів згідно з цими регуляторними стандартами.

Керування резервним копіюванням виснажує ІТ-ресурси

Логіка підказує, що якщо ІТ-команди намагаються ефективно створювати резервні копії своїх даних, то вони не витрачають на це достатньо часу. Але, схоже, все навпаки.

Керування резервним копіюванням зараз займає щонайменше десять годин на тиждень у половини ІТ-фахівців, що майже вдвічі більше часу, ніж два роки тому, а галузеве дослідження, на яке посилається ТРх, оцінює, що ці зусилля коштують організаціям близько 25 000 доларів США на рік у середній зарплаті. (ІТ-фахівці заробляють в середньому близько 100 000 доларів США на рік)

Тож яке ж рішення? Підвищте ефективність резервного копіювання, щоб воно не займало стільки часу вашої ІТ-команди. Автоматизація резервного копіювання – це один із методів, який може допомогти ІТ-командам відновити контроль над своїм часом.

Стратегії резервного копіювання не встигають за сучасними ІТ-технологіями

Ймовірно, не допомагає й те, що так мало компаній оновили свою стратегію резервного копіювання та відновлення за останні п'ять років.

Ми спостерігаємо, як робочі навантаження переносяться в хмару швидше,

					ВКРБ-123.26.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		30

ніж будь-коли, а також ширше впровадження хмарних додатків і технологій у різних галузях. Однак MSP Success поділилася галузевим дослідженням, що 28% організацій не змінювали свою стратегію резервного копіювання та відновлення протягом останніх п'яти років.

Якщо ваша стратегія резервного копіювання та відновлення не встигає за оновленнями, вона не буде готова задовольнити потреби вашої ІТ-команди.

Організації активно переглядають постачальників резервного копіювання

Більшість організацій хочуть оновити своїх постачальників резервного копіювання наступного року.

За прогнозами MSP Success, галузеві дослідження, на які посилалася компанія, що цитує дані MSP Success, показують, що 73 відсотки респондентів планують змінити постачальників резервного копіювання, причому вартість названа основним рушійним фактором цих рішень.

Однак вони також зазначили, що їм потрібні рішення для резервного копіювання, які можуть робити більше, ніж просто задовольняти їхні потреби. Такі функції, як тестування аварійного відновлення, автоматизація відновлення та оркестрація резервного копіювання, також були дуже бажаними.

Через сучасні вимоги до захисту даних застарілі засоби резервного копіювання не відповідають потребам сучасного бізнесу. Вони визначають відповідні засоби захисту даних, які перевершують їхні вимоги, шляхом оцінки нових рішень для захисту даних.

Витоки даних продовжуватимуться – це неминуче. Але організації, які зможуть покращити свої можливості відновлення, виживуть.

Як показує наведена вище статистика, компанії просто не впевнені у своїх поточних програмах захисту даних. Але це може змінитися, якщо вони впровадять такі передові методи, як автоматизоване резервне копіювання, звітування про резервне копіювання та тестування відновлення.

Коли компанії автоматизують операції резервного копіювання та

відновлення, вони можуть зменшити витрати, дозволяючи ІТ-фахівцям зосередитися на більш важливих обов'язках. Але не всі постачальники засобів захисту даних надають такі функції, тому обнадійливо бачити, що більшість компаній планують оцінити нове рішення для резервного копіювання цього року.

3.2 Розробка структурної схеми

Що потрібно, щоб дотримуватися вимог у світі, переповненому конфіденційними даними? У даному розділі розглядаються ключові стандарти безпеки даних, які має розуміти кожен бізнес, від конфіденційності та захисту до зберігання та управління.

Стандарти відповідності вимогам безпеки даних – це нормативно-правові рамки, які вимагають від підприємств захищати, регулювати та належним чином обробляти конфіденційну інформацію, щоб уникнути штрафів до 20 мільйонів євро та зберегти довіру клієнтів.

Люди ще ніколи не були більш обізнаними чи обережними щодо того, як обробляються їхні дані. І не без підстав. Від фішингових шахрайств та крадіжки особистих даних до корпоративних порушень, які розкривають мільйони записів, неправомірне використання даних стало однією з найдорожчих та найпоширеніших загроз нашого часу.

Окремі особи розплачуються за це через крадіжку облікових даних, спустошення банківських рахунків та втрату конфіденційності. Організації також страждають, стикаючись з фінансовими втратами, судовими позовами, а іноді й непоправною шкодою для своєї репутації. У ширшому масштабі економічні витрати на злочини, пов'язані з даними, сягають трильйонів щорічно .

Безпека даних є критично важливим пріоритетом для бізнесу, а дотримання вимог безпеки даних – це складна, динамічна система законів, контролю та обов'язків, що визначають, як дані повинні збиратися, захищатися та зберігатися. Ставки високі, очікування зростають, а виклики зростають.

					ВКРБ-123.26.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		32

У даному розділі ми розглянемо найважливіші стандарти безпеки даних, які повинні знати компанії, дослідимо ключові вимоги та проблеми, а також запропонуємо стратегії для забезпечення відповідності вимогам у світі, перенасиченому даними.

Стандарти та положення щодо відповідності основним вимогам безпеки даних

Відповідність вимогам щодо даних охоплює кілька сфер, що перетинаються, кожна з яких має свої ризики, очікування та нормативно-правові рамки. Загалом, більшість зобов'язань щодо дотримання вимог належать до однієї з трьох категорій: конфіденційність даних, безпека та захист даних, а також зберігання та управління даними. Розуміння цієї різниці може допомогти компаніям розробляти розумніші та комплексніші стратегії дотримання вимог.

Конфіденційність даних: прозорість, згода та права користувачів

Дотримання вимог щодо конфіденційності даних стосується того, як компанії збирають та використовують персональну інформацію, а також чи роблять вони це етично, прозоро та з дозволу. Ці правила надають особам контроль над своїми даними, включаючи право знати, що збирається, чому це використовується та як подати запит на їх видалення. Підприємства повинні бути готові дотримуватися запитів на відмову, відповідати на запити на доступ до даних та уникати зловживання конфіденційною інформацією.

Це означає переосмислення того, як отримується згода, як повідомляються політики конфіденційності та як використання даних узгоджується з тим, з чим погодилися користувачі. Закони про конфіденційність підвищують планку для всіх, змушуючи компанії будувати довіру за замовчуванням.

Загальний регламент про захист даних (GDPR) у ЄС та CCPA/CPRA у Каліфорнії – це два найвпливовіші закони про конфіденційність у цій сфері, які надихнули на аналогічне законодавство в Бразилії, Південній Африці та Індії. Хоча деталі відрізняються, спільною метою є забезпечення відповідальності людей за свої персональні дані.

					ВКРБ-123.26.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		33

Безпека та захист даних: захист інформації від порушень

Якщо конфіденційність стосується того, які дані збираються та чому, то захист стосується того, наскільки добре ці дані захищені, коли вони потрапляють до вас. Правила безпеки даних вимагають від компаній впроваджувати технічні та організаційні заходи безпеки для запобігання несанкціонованому доступу, витокам або втратам. До них належать чіткі очікування щодо шифрування, контролю доступу, брандмауерів та протоколів реагування на порушення.

Складністю захисту даних є його широта охоплення. Він стосується не лише даних клієнтів, а й записів співробітників, корпоративної інтелектуальної власності, внутрішнього зв'язку та навіть метаданих. А оскільки загрози постійно розвиваються, закони наголошують на постійному вдосконаленні, а це означає, що вчорашній рівень безпеки може бути неефективним завтра.

Такі закони, як Правило безпеки HIPAA у США, Закон SHIELD у Нью-Йорку та Директива ЄС NIS2, вимагають базових заходів захисту та практик управління ризиками. Крім того, такі стандарти, як PCI DSS та ISO/IEC 27001, забезпечують більш детальні операційні рамки для впровадження цих заходів захисту в різних галузях.

Зберігання та управління даними: підтвердження відповідності та забезпечення нагляду

Правила зберігання та управління даними зосереджені на тому, як довго ви зберігаєте інформацію, як вона зберігається та як її можна отримати для аудитів, розслідувань або юридичних розслідувань. У багатьох секторах незбереження ключової електронної пошти або голосового запису може бути таким же шкідливим, як і їх витік.

Ця сфера дотримання вимог є особливо важливою у фінансовій, юридичній та публічній сферах, де компанії повинні демонструвати регуляторним органам, що їхні записи є повними, точними та захищеними від несанкціонованого доступу. Саме тут також виникають труднощі, оскільки співробітники все частіше використовують неформальні канали, такі як

					ВКРБ-123.26.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		34

WhatsApp або SMS, для ведення бізнесу. Якщо ці повідомлення не архівуються, це вважається порушенням вимог.

Такі нормативні акти, як Закон Сарбейнса-Окслі (SOX), правило 17a-4 Комісії з цінних паперів та бірж США, правила FINRA та MiFID II, встановлюють суворі зобов'язання щодо зберігання записів, особливо щодо комунікацій та фінансових розкриттів. Навіть японський закон J-SOX відображає цей акцент на внутрішньому контролі та управлінні даними.

Ключові вимоги до дотримання вимог щодо даних, які має дотримуватися кожен бізнес

Хоча кожне положення є унікальним, більшість стандартів відповідності мають спільні принципи та очікування. Ось що зазвичай очікується від компаній:

- Чіткі політики управління даними: Компанії повинні встановити документовані політики, в яких детально описано, як збираються дані, де вони зберігаються, хто може отримати до них доступ і як вони зрештою видаляються або архівуються. Ці політики слугують основою для внутрішньої підзвітності та зовнішніх аудитів.

- Правила зберігання та видалення даних: Більшість нормативних актів містять конкретні правила щодо того, як довго компанії можуть зберігати персональні або конфіденційні дані. Компаніям необхідно автоматизувати графіки зберігання та забезпечити остаточне видалення даних, коли вони більше не потрібні, інакше вони ризикують порушити правила «права бути забутим».

- Контроль доступу та автентифікація: Компанії повинні впроваджувати доступ на основі ролей, щоб лише уповноважений персонал міг переглядати або змінювати конфіденційну інформацію. Це часто включає багатофакторну автентифікацію, політики паролів та аудит доступу користувачів.

- Журнали аудиту та ведення журналу: Для забезпечення прозорості та підзвітності системи повинні реєструвати всі доступи, зміни та передачі конфіденційних даних. Ці журнали повинні бути незмінними та такими, що їх

					ВКРБ-123.26.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		35

можна буде отримати у разі регуляторного розслідування або внутрішнього перегляду.

– Шифрування під час зберігання та передачі: Незалежно від того, чи дані зберігаються на сервері, чи передаються мережею, вони мають бути зашифровані за галузевими стандартами. Це запобігає перехопленню або крадіжці, навіть якщо інші системи скомпрометовані.

– Механізми згоди та прозорості: Компанії повинні надавати чіткі та доступні повідомлення про те, які дані збираються, як вони будуть використовуватися та з ким вони будуть передаватися. У багатьох випадках необхідно отримати чітку згоду користувача, і компанія повинна мати змогу її довести.

– Протоколи виявлення та оповіщення про порушення: У разі порушення безпеки даних організації повинні діяти швидко. Багато нормативних актів вимагають повідомлення регуляторних органів та постраждалих осіб протягом 72 годин. Компанії повинні мати план реагування на інциденти та інструменти для швидкого виявлення та оцінки порушень.

Які найбільші труднощі у дотриманні вимог?

Найбільшою перешкодою для багатьох організацій є те, щоб зробити дотримання вимог невід'ємною частиною щоденної діяльності. Ось найпоширеніші перешкоди, з якими стикаються підприємства:

– Фрагментовані середовища даних: дані часто зберігаються в десятках непов'язаних систем – хмарних додатках, внутрішніх базах даних, сторонніх інструментах та платформах обміну повідомленнями. Така фрагментація робить практично неможливим контроль або застосування послідовних політик зберігання та доступу.

– Постійно оновлювані правила: такі закони, як GDPR та CCPA, постійно оновлюються, а нові структури (наприклад, Закон про захист цифрових персональних даних Індії або PIPL Китаю) з'являються в усьому світі. Підтримка

					ВКРБ-123.26.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		36

актуальності та коригування політик у різних регіонах – це робота на повний робочий день.

– Політики віддаленої роботи та BYOD: Коли співробітники використовують особисті пристрої та споживчі месенджери (такі як WhatsApp або Signal) для ведення бізнесу, організації втрачають видимість та контроль. Це створює сліпі зони відповідності, які важко закрити заднім числом.

– Міжюрисдикційні операції: Компанія, яка працює в США, ЄС та Азії, може бути змушена дотримуватися півдюжини суперечливих стандартів одночасно. Це може призвести до операційної складності та правової невизначеності, якщо немає скоординованої стратегії дотримання вимог.

– Обмежена внутрішня експертиза: Особливо для невеликих компаній створення програми комплаєнсу з нуля може бути складним завданням. Без спеціального юридичного, IT-фахівця або фахівця з комплаєнсу підприємствам може бути важко оцінити ризики або впровадити належні запобіжні заходи.

– Поведінка та культура співробітників: Навіть за наявності найкращих інструментів, людська помилка залишається головним ризиком для дотримання вимог. Незахищені файли, слабкі паролі та позаканальний зв'язок можуть легко призвести до витоків даних або порушень, якщо співробітники не проходять навчання та не контролюються.

Як компанії можуть створити надійну систему комплаєнсу?

Створення надійної програми забезпечення відповідності даних – це постійний процес, який вимагає узгодження дій між людьми, процесами та технологіями. Найефективнішими структурами є ті, які інтегрують відповідність вимогам у повсякденні бізнес-операції, а не розглядають її як зовнішній додаток.

Ось деякі основні кроки, які організації можуть зробити для створення проактивного та стійкого підходу до дотримання вимог:

– Почніть з чіткого інвентарю даних: Перш ніж запровадити будь-яку змістовну стратегію відповідності, підприємствам необхідно зрозуміти, які дані вони зберігають, де вони знаходяться, хто має до них доступ і чому вони

					ВКРБ-123.26.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		37

збираються. Це включає структуровані дані в базах даних, неструктуровані дані в повідомленнях або документах, а також дані, що передаються між сторонніми платформами. Відображення життєвого циклу даних від збору до збереження або видалення має вирішальне значення для застосування належних заходів захисту та виконання нормативних зобов'язань.

– Розробка міжфункціональних, практичних політик: Ефективні політики дотримання вимог повинні охоплювати відділи та щоденні операції. Юридичні команди часто розробляють проекти політик, а команди IT, HR, маркетингу та обслуговування клієнтів відіграють певну роль у їх впровадженні. Політики повинні відображати фактичні робочі процеси кожної команди та бути адаптованими до галузі, географії та регуляторного впливу компанії.

– Впровадження технологій, що підтримують відповідність вимогам за проектом: Організації повинні впроваджувати системи, що забезпечують основні функції відповідності, такі як автоматизований збір даних, детальний контроль доступу, ведення журналу аудиту та планування зберігання.

– Інвестуйте в постійне навчання співробітників: регулярне навчання, адаптоване до конкретних ролей та обов'язків, допомагає співробітникам розпізнавати ризики, пов'язані з дотриманням вимог, та розуміти свою роль у їх мінімізації.

– Проводьте регулярні аудити та виявляйте прогалини: Внутрішні перевірки допомагають організаціям оцінити, наскільки добре працюють політики, тестувати системи на практиці та виявляти слабкі місця, перш ніж вони призведуть до порушень. Оцінка прогалин також може виявити сфери, де змінилися правила або де бізнес-операції вийшли за межі поточної структури відповідності.

– Використовуйте комунікаційні платформи, що забезпечують управління: оскільки бізнес-комунікації стають більш мобільними та децентралізованими, організаціям необхідно забезпечити, щоб обмін повідомленнями, дзвінки та інструменти для співпраці не залишалися поза

					ВКРБ-123.26.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		38

екосистемою відповідності. Платформи, що дозволяють збирати, архівувати та перевіряти повідомлення, є важливими для виконання вимог законодавства, особливо в регульованих галузях.

Створюйте відповідно до вимог, працюйте впевнено

Дотримання вимог безпеки даних може спочатку бути юридичною вимогою, але для прогресивних компаній воно стає конкурентною перевагою. Довіра крихка, а наслідки невдачі серйозні, тому компанії, які керують даними ретельно, чітко та послідовно, уникають штрафів, а також завойовують лояльність, захищають свою репутацію та залишаються гнучкими перед обличчям змін.

Відповідність даних вимагає контролю, прозорості та можливості аудиту в усіх аспектах бізнесу. Це особливо актуально, коли йдеться про електронні комунікації, де неконтрольовані канали можуть швидко стати точками впливу регуляторних органів.

Дотримання вимог щодо даних допомагає бізнесу захищати конфіденційну інформацію, будувати довіру з клієнтами та уникати значних штрафів. Це гарантує, що компанії обробляють персональні та фінансові дані відповідно до правових та етичних стандартів. Недотримання вимог може призвести до серйозних наслідків, таких як штрафи, судові позови та репутаційна шкода. Але справа не лише в уникненні покарання. Суворі практики дотримання вимог щодо даних також підвищують операційну ефективність, знижують ризики та надають компаніям конкурентну перевагу.

До найбільш широко застосовуваних правил належать Загальний регламент про захист даних (GDPR) у ЄС, Закон Каліфорнії про конфіденційність споживачів (CCPA) та його поправка, CPRA, у США, HIPAA для даних охорони здоров'я, PCI DSS для обробки платежів та вимоги FINRA/SEC для фінансових послуг. Кожен із цих правил встановлює стандарти щодо збору, зберігання, використання та захисту даних. Хоча конкретні особливості різняться, більшість з них зосереджені на згоді, прозорості, безпеці даних та сповіщенні про

					ВКРБ-123.26.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		39

порушення. Підприємства, що працюють на міжнародному рівні, можуть підпадати під дію кількох нормативних актів одночасно, що вимагає єдиної, гнучкої стратегії дотримання вимог.

Дотримання вимог конфіденційності даних стосується того, як організації збирають, обмінюються та використовують персональні дані, з акцентом на прозорість, згоду та права особистості. Уявіть це як правила щодо того, хто отримує доступ до даних та як вони використовуються. З іншого боку, дотримання вимог безпеки даних стосується захисту цих даних від несанкціонованого доступу, крадіжки або втрати. Це включає шифрування, контроль доступу та журнали аудиту. Обидва є важливими, оскільки конфіденційність без безпеки не має сенсу, а безпека без конфіденційності все одно може призвести до юридичних проблем.

Порушення можуть призвести до суворих штрафів, включаючи багатомільйонні штрафи, судові позови та регуляторні розслідування. Наприклад, штрафи за GDPR можуть сягати 20 мільйонів євро або 4% від світового річного доходу, залежно від того, що більше. Окрім фінансових наслідків, компанії часто зазнають шкоди від репутації та втрати клієнтів. Регулятори також можуть запроваджувати обмеження на те, як компанія обробляє дані в майбутньому. У деяких галузях порушення вимог можуть призвести до анулювання ліцензій або примусового закриття. Під загрозою знаходяться не лише великі корпорації – менші фірми також все частіше піддаються ретельній перевірці.

Як мінімум, компанії повинні переглядати свою програму відповідності даних щорічно. Але насправді перевірки повинні проводитися частіше, особливо після суттєвих оновлень нормативних актів, змін у системі або інцидентів безпеки. Регулярні внутрішні аудити допомагають виявити прогалини, забезпечити дотримання політик та виявити ризики до їх загострення. Періодичні оцінки третіх сторін також можуть запропонувати новий погляд. Відповідність вимогам повинна розвиватися разом з вашим бізнесом, вашим технологічним стеком та регуляторним ландшафтом.

					ВКРБ-123.26.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		40

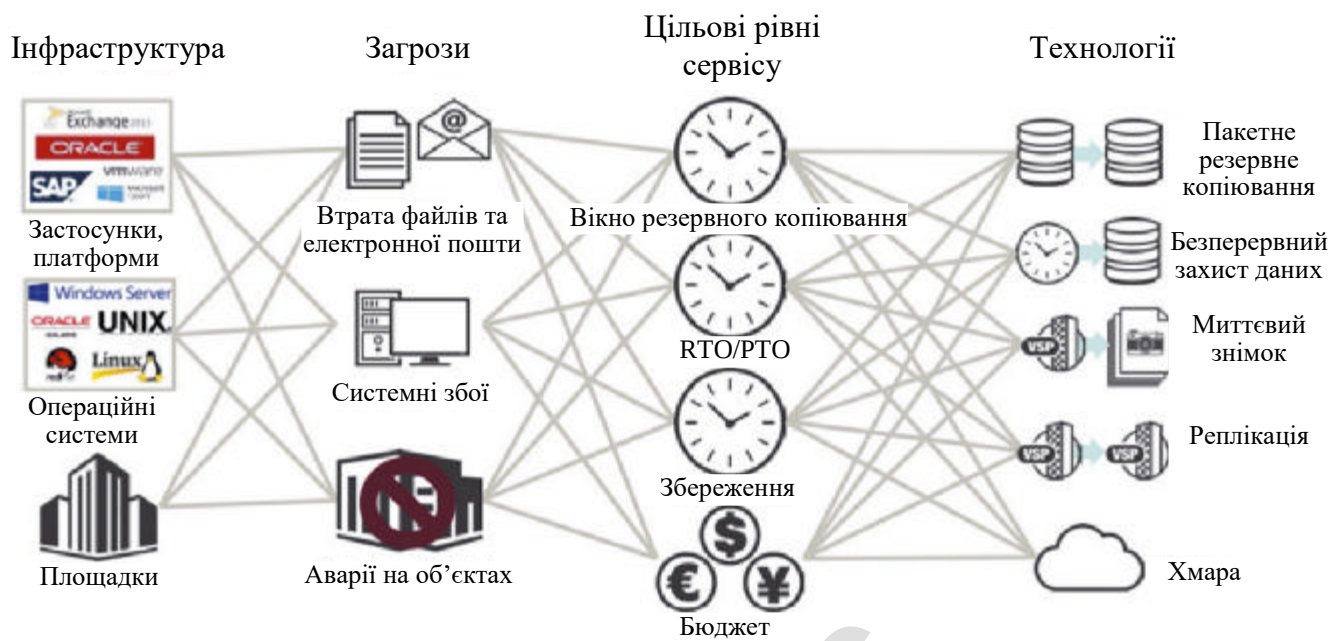


Рисунок 3.1 – Структурна схема системи

Існує низка інструментів, що підтримують дотримання вимог, включаючи програмне забезпечення для запобігання втратам даних (DLP), платформи шифрування, брокери безпеки хмарного доступу (CASB) та рішення для управління, ризиків та дотримання вимог (GRC). Платформи управління комунікаціями, такі як ті, що захоплюють та архівують бізнес-повідомлення, відіграють дедалі більшу роль, особливо в таких секторах, як фінанси та право. Інші корисні інструменти включають системи управління ідентифікацією та доступом (IAM), автоматизоване ведення журналу аудиту та платформи виявлення порушень. Правильний технологічний стек залежить від вашої галузі, профілю ризику та каналів зв'язку.

Ні, єдиного універсального стандарту не існує. Натомість організації повинні орієнтуватися у складній мережі міжнародних, національних та галузевих норм. Однак деякі структури, такі як ISO/IEC 27001, широко визнані як моделі найкращої практики для побудови системи управління інформаційною безпекою (СУІБ). Вони можуть допомогти компаніям створити базову базу контролю, яка відповідає багатьом світовим стандартам. Тим не менш,

дотримання вимог часто вимагає адаптації вашого підходу до конкретних законів, таких як GDPR, HIPAA або CCPA, особливо якщо ви працюєте в кількох регіонах або маєте справу з регульованими типами даних.

3.3 Розробка функціональної схеми

У даній роботі для захисту та відновлення даних в хмарних сервісах використовується технологія RAID. Надамо її опис. Відразу варто помітити, що RAID розшифровується як Redundant Array of Independent Disks – надлишковий масив незалежних дисків. Споконвічно RAID розшифровувався як Redundant Array of Inexpensive Disks – надлишковий масив недорогих дисків. Під недорогими малися на увазі диски, призначені для використання в ПК, – на противагу дорогим дискам для мейнфреймів. Але так як в RAID-масивах стали використовувати SCSI-вінчестери, які були істотно дорожче застосовуваних у більшості комп'ютерів дисків IDE, слово "недорогий" було замінено на "незалежний".

Принцип функціонування RAID-системи полягає в наступному: з набору дискових накопичувачів створюється масив, що управляється спеціальним контролером і визначається комп'ютером як єдиний логічний диск великої ємності. За рахунок паралельного виконання операцій вводу-виводу забезпечується висока швидкодія системи, а підвищена надійність зберігання інформації досягається дублюванням даних або обчисленням контрольних сум. Слід зазначити, що застосування RAID-масивів захищає від втрат даних тільки у випадку фізичної відмови жорстких дисків.

Розрізняють кілька основних рівнів RAID-масивів: RAID 0, 1, 2, 3, 4, 5, 6, 7. Також існують комбіновані рівні, такі як RAID 10, 0+1, 30, 50, 53 і т.п. Розглянемо принципи функціонування, достоїнства й недоліки основних рівнів.

					ВКРБ-123.26.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		42

RAID 0 – Дисковий масив без відказостійкості (Striped Disk Array without Fault Tolerance)

Дисковий масив без надлишкового зберігання даних. Інформація розбивається на блоки, які одночасно записуються на окремі диски, що забезпечує підвищення продуктивності. Такий спосіб зберігання інформації ненадійний, оскільки поломка одного диска приводить до втрати всієї інформації, тому рівнем RAID як таким не є.

RAID 0 – дешевий і продуктивний, але ненадійний. За рахунок можливості одночасного вводу/виводу з декількох дисків масиву RAID 0 забезпечує максимальну швидкість передачі даних і максимальну ефективність використання дискового простору, тому що не потрібно місця для зберігання контрольних сум. Реалізація цього рівня дуже проста. RAID 0, як правило, застосовується в тих областях, де потрібна швидка передача великого обсягу даних. Для реалізації масиву потрібно не менше двох вінчестерів.

Переваги:

- найвища продуктивність у додатках, що вимагають інтенсивної обробки запитів вводу/виводу й даних великого обсягу;
- простота реалізації;
- низька вартість;
- максимальна ефективність використання дискового простору – 100%.

Недоліки:

- не є "сьогоденням" RAID'ом, оскільки не підтримує відказостійкість;
- відмова одного диска спричиняє втрату всіх даних масиву.

RAID 1 – Дисковий масив із відзеркалюванням (Mirroring & Duplexing)

Дисковий масив з дублюванням інформації (відзеркалюванням даних). У найпростішому випадку два накопичувачі містять однакову інформацію і є одним

логічним диском. При виході з ладу одного диска його функції виконує інший. Для реалізації масиву потрібно не менше двох вінчестерів.

RAID 1 – найпростіший відказостійкий масив.

Переваги:

- простота реалізації;
- простота відновлення масиву у випадку відмови (копіювання).

Недоліки:

- висока вартість – 100-процентна надмірність;
- невисока швидкість передачі даних.

RAID 2 – Відказостійкий дисковий масив з використанням коду Хеммінга (Hamming Code ECC)

Схема резервування даних з використанням коду Хеммінга (Hamming code) для корекції помилок. Потік даних розбивається на слова – причому розмір слова відповідає кількості дисків для запису даних. Для кожного слова обчислюється код корекції помилок, що записується на диски, виділені для зберігання контрольної інформації. Їхнє число дорівнює кількості біт у слові контрольної суми.

RAID 2 не одержав комерційного застосування.

Якщо слово складається із чотирьох біт, то під контрольну інформацію приділяється три диски. RAID 2 – один з деяких рівнів, що дозволяють виявляти подвійні помилки й виправляти "на льоту" одиночні. При цьому він є самим надлишковим серед всіх рівнів з контролем парності. Ця схема зберігання даних не одержала комерційного застосування, оскільки погано справляється з великою кількістю запитів.

Переваги:

- досить проста реалізація;
- корекція помилок "на льоту";
- дуже висока швидкість передачі даних;
- при збільшенні кількості дисків накладні витрати зменшуються.

					ВКРБ-123.26.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		44

Недоліки:

- низька швидкість обробки запитів;
- висока вартість;
- більша надмірність.

RAID 3 – Відказостійкий дисковий масив з паралельною передачею даних і парністю (Parallel Transfer Disks with Parity)

Відказостійкий масив з паралельним вводом/виводом даних і диском контролю парності. Потік даних розбивається на порції на рівні байт (хоча можливо й на рівні біт) і записується одночасно на всі диски масиву, крім одного. Один диск призначений для зберігання контрольних сум, що обчислюються при записі даних. Поломка кожного з дисків масиву не приведе до втрати інформації.

В RAID 3 інформація розбивається на порції однакового розміру.

Цей рівень має набагато меншу надмірність, чим RAID 2. У RAID 2 більшість дисків, що зберігають контрольну інформацію, потрібні для визначення несправного розряду. Як правило, RAID-контролери можуть одержати дані про помилку за допомогою механізмів відстеження випадкових збоїв. За рахунок розбивки даних на порції RAID 3 має високу продуктивність. Оскільки при кожній операції вводу/виводу виробляється обіг практично до всіх дисків масиву, те одночасна обробка декількох запитів неможлива.

Цей рівень підходить для додатків з файлами великого обсягу й малою частотою обігів (в основному це сфера мультимедіа). Використання тільки одного диска для зберігання контрольної інформації пояснює той факт, що коефіцієнт використання дискового простору досить високий (як наслідок цього – відносно низька вартість). Для реалізації масиву потрібно не менше трьох вінчестерів.

Переваги:

- відмова диска мало впливає на швидкість роботи масиву;
- висока швидкість передачі даних;
- високий коефіцієнт використання дискового простору.

					ВКРБ-123.26.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		45

Недоліки:

- складність реалізації;
- низька продуктивність при великій інтенсивності запитів даних невеликого обсягу.

RAID 4 – Відказостійкий масив незалежних дисків із загальним диском парності (Independent Data Disks with Shared Parity Disk)

Цей масив дуже схожий на рівень RAID 3. Потік даних розділяється не на рівні байтів, а на рівні блоків інформації, кожен з яких записується на окремий диск. Після запису групи блоків обчислюється контрольна сума, що записується на виділений для цього диск.

В RAID 4 потік даних розділяється на блоки.

В RAID 4 можливо одночасне виконання декількох операцій читання. Цей масив підвищує продуктивність передачі файлів малого обсягу (за рахунок розпаралелювання операції зчитування). Але оскільки при записі повинна змінюватися контрольна сума на виділеному диску, одночасне виконання операцій неможливо (у наявності асиметричність операцій вводу й виводу). Цей рівень має майже всі недоліки RAID 3 і не забезпечує переваги у швидкості при передачі даних великого обсягу. Схема зберігання розроблялася для додатків, у яких дані споконвічно розбиті на невеликі блоки, тому немає необхідності розбивати їх додатково. Ця схема зберігання даних має невисоку вартість, але її реалізація досить складна, як і відновлення даних при збої.

Переваги:

- висока швидкість передачі даних;
- відмова диска мало впливає на швидкість роботи масиву;
- високий коефіцієнт використання дискового простору.

Недоліки:

- досить складна реалізація;
- дуже низька продуктивність при записі даних;
- складне відновлення даних.

RAID 5 – Відказостійкий масив незалежних дисків з розподіленою парністю (Independent Data Disks with Distributed Parity Blocks)

Найпоширеніший рівень. Блоки даних і контрольних сум циклічно записуються на всі диски масиву, відсутній виділений диск для зберігання інформації про парність, немає асиметричності конфігурації дисків.

У випадку RAID 5 всі диски масиву мають однаковий розмір – але один з них не бачимий для операційної системи. Наприклад, якщо масив складається з п'яти дисків ємністю 10 Гб кожний, те фактично розмір масиву буде дорівнює 40 Гб – 10 Гб приділяється на контрольні суми. У загальному випадку корисна ємність масиву з N дисків дорівнює сумарної ємності N- 1 диска.

В RAID 5 відсутній виділений диск для зберігання інформації про парність.

Найбільший недолік рівнів RAID від 2-го до 4-го – це наявність окремого диска (або дисків), що зберігає інформацію про парність. Швидкість виконання операцій зчитування досить висока, тому що не вимагає звертання до цього диска. Але при кожній операції запису на ньому змінюється інформація, тому схеми RAID 2-4 не дозволяють проводити паралельні операції запису. RAID 5 не має цього недоліку, тому що контрольні суми записуються на всі диски масиву, що уможливорює виконання декількох операцій читання або записи одночасно. RAID 5 має досить високу швидкість запису/читання й малу надмірність.

Переваги:

- висока швидкість запису даних;
- досить висока швидкість читання даних;
- висока продуктивність при великій інтенсивності запитів читання/запису даних;
- високий коефіцієнт використання дискового простору.

Недоліки:

- низька швидкість читання/запису даних малого обсягу при одиничних запитах;

					ВКРБ-123.26.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		47

- досить складна реалізація;
- складне відновлення даних.

RAID 6 – Відказостійкий масив незалежних дисків із двома незалежними розподіленими схемами парності (Independent Data Disks with Two Independent Distributed Parity Schemes)

RAID 6 – це відказостійкий масив незалежних дисків з розподілом контрольних сум, обчислених двома незалежними способами. Цей рівень багато в чому схожий з RAID 5. Тільки в ньому використовується не одна, а дві незалежні схеми контролю парності, що дозволяє зберігати працездатність системи при одночасному виході з ладу двох накопичувачів. Для обчислення контрольних сум в RAID 6 використовується алгоритм, побудований на основі коду Ріда-Соломона (Reed-Solomon).

RAID 6 використовує дві незалежні схеми контролю парності.

Цей рівень має дуже високу відказостійкість, більшу швидкість зчитування (дані зберігаються блоками, немає виділених дисків для зберігання контрольних сум). У той же час через великий обсяг контрольної інформації RAID 6 має низьку швидкість запису. Він дуже складний у реалізації, характеризується низьким коефіцієнтом використання дискового простору: для масиву з п'яти дисків він становить усього 60%, але з ростом числа дисків ситуація виправляється.

RAID 6 по багатьом характеристикам програє іншим рівням, тому на сьогодні не одержав комерційного застосування.

Переваги:

- висока відказостійкість;
- досить висока швидкість обробки запитів;

Недоліки:

- низька швидкість читання/запису даних малого обсягу при одиничних запитах;
- дуже складна реалізація;

- складне відновлення даних;
- низька швидкість запису даних.

RAID 7 – Відказостійкий масив, оптимізований для підвищення продуктивності (Optimized Asynchrony for High I/O Rates as well as High Data Transfer Rates)

На відміну від інших рівнів, RAID 7 не є відкритим індустріальним стандартом – це зареєстрована торговельна марка компанії Storage Computer Corporation. Масив ґрунтується на концепціях, використаних у третьому й четвертому рівнях. Додалася можливість кешування даних. До складу RAID 7 входить контролер з убудованим мікропроцесором під керуванням операційної системи реального часу (real-time OS). Вона дозволяє обробляти всі запити на передачу даних асинхронно й незалежно.

RAID 7 – зареєстрована торговельна марка компанії Storage Computer Corporation.

Блок обчислення контрольних сум інтегрований із блоком буферизації; для зберігання інформації про парність використовується окремий диск, що може бути розміщений на будь-якому каналі. RAID 7 має високу швидкість передачі даних і обробки запитів, гарну масштабованість. Самим більшим недоліком цього рівня є вартість його реалізації.

Переваги:

- дуже висока швидкість передачі даних і висока швидкість обробки запитів (в 1,5...6 разів вище інших стандартних рівнів RAID);
- гарна масштабованість;
- значно зросла (завдяки наявності кешу) швидкість читання даних невеликого обсягу;
- відсутність необхідності в додатковій передачі даних для обчислення парності.

Недоліки:

- власність однієї компанії;

					ВКРБ-123.26.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		49

- складність реалізації;
- дуже висока вартість на одиниці об'єму;
- не може обслуговуватися користувачем;
- необхідність використання блоку безперебійного живлення для запобігання втрати даних з кеш-пам'яті;
- короткий гарантійний строк.

RAID 10

RAID 10 – відказостійкий масив з дублюванням і паралельною обробкою. Ця архітектура виявляє собою масив типу RAID 0, сегментами якого є масиви RAID 1. Він поєднує в собі високу відказостійкість і продуктивність.

Комбіновані рівні

Крім базових рівнів RAID 0 – RAID 5, описаних у стандарті, існують комбіновані рівні RAID 1+0, RAID 3+0, RAID 5+0, RAID 1+5, які різні виробники інтерпретують кожний по-своєму.

- RAID 1+0 – це сполучення дзеркалювання й чергування.

Нинішні контролери використовують цей режим за замовчуванням для RAID 1. Тобто, 1 диск основний, 2-й диск – дзеркало, причому читання виробляється з них по черзі, як для RAID 0. Властиво, зараз можна вважати що RAID 1 і RAID 1+0 – це просто різні назви того самого методу апаратного дзеркалювання дисків. Але не варто забувати, що повноцінний RAID 1+0 повинен містити як мінімум 4 диски.

- RAID 5+0 – це чергування томів 5-го рівня. RAID 1+5 – RAID 5 із дзеркальованою парою.

Комбіновані рівні успадковують як переваги, так і недоліки своїх «батьків»: поява чергування в рівні RAID 5+0 анітрошки не додає йому надійності, але зате позитивно відбивається на продуктивності. Рівень RAID 1+5, напевно, дуже надійний, але не найшвидший і, до того ж, украй неекономічний: корисна ємність тому менше половини сумарної ємності дисків.

					ВКРБ-123.26.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		50

Варто відзначити, що кількість жорстких дисків у комбінованих масивах також зміниться. Наприклад для RAID 5+0 використовують 6 або 8 жорстких дисків, для RAID 1+0 – 4, 6 або 8.

На рисунку 3.2 зображена функціональна схема системи. Нижче розглянемо її більш докладно.

Система складається з наступних блоків.

Блок вибору кількості дисків – призначений для визначення кількості дисків з яких буде складатися RAID-масив. Якщо дисків буде 2 то буде використовуватися технологія RAID 0 та RAID 1. Якщо дисків буде більше 2 то буде використовуватися технологія RAID 0 та RAID 6.

Блок формування віртуальних дисків використовується у разі коли диск тільки один. Тоді на цьому диску формуються декілька віртуальних дисків, з якими відбуваються дії аналогічні як ті, що відбуваються над фізичними дисками. Тобто RAID-масив формується з віртуальних дисків.

Блок розподілу інформації за критерієм критичності/важливості визначає який саме вид RAID-масиву необхідно буде використовувати. Якщо інформація не є критичною то буде використовуватися RAID 0. Якщо інформація є важливою, то в залежності від ступеня важливості, та вимог до швидкодії, буде використовуватися або RAID 1, який є більш швидким, та менш надійним, або RAID 6, який є дуже надійним але менш швидким.

Блок вибору типу RAID масиву дозволяє автоматично, в залежності від заданих параметрів у попередніх блоках, визначити, який саме тип RAID-масиву потрібен для зберігання тієї, або іншої інформації. Блоки RAID 0, RAID 1 та RAID 6 реалізують ту, або іншу технологію. У блоці RAID 6 використовується кодек Ріда-Соломона. Нижче надамо опис цього кодеку.

Коди Ріда-Соломона – недвійкові циклічні коди, що дозволяють виправляти помилки в блоках даних. Елементами кодового вектора є не біти, а групи біт (блоки). Дуже поширені коди Ріда-Соломона, що працюють із байтами (октетами).

					ВКРБ-123.26.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		51

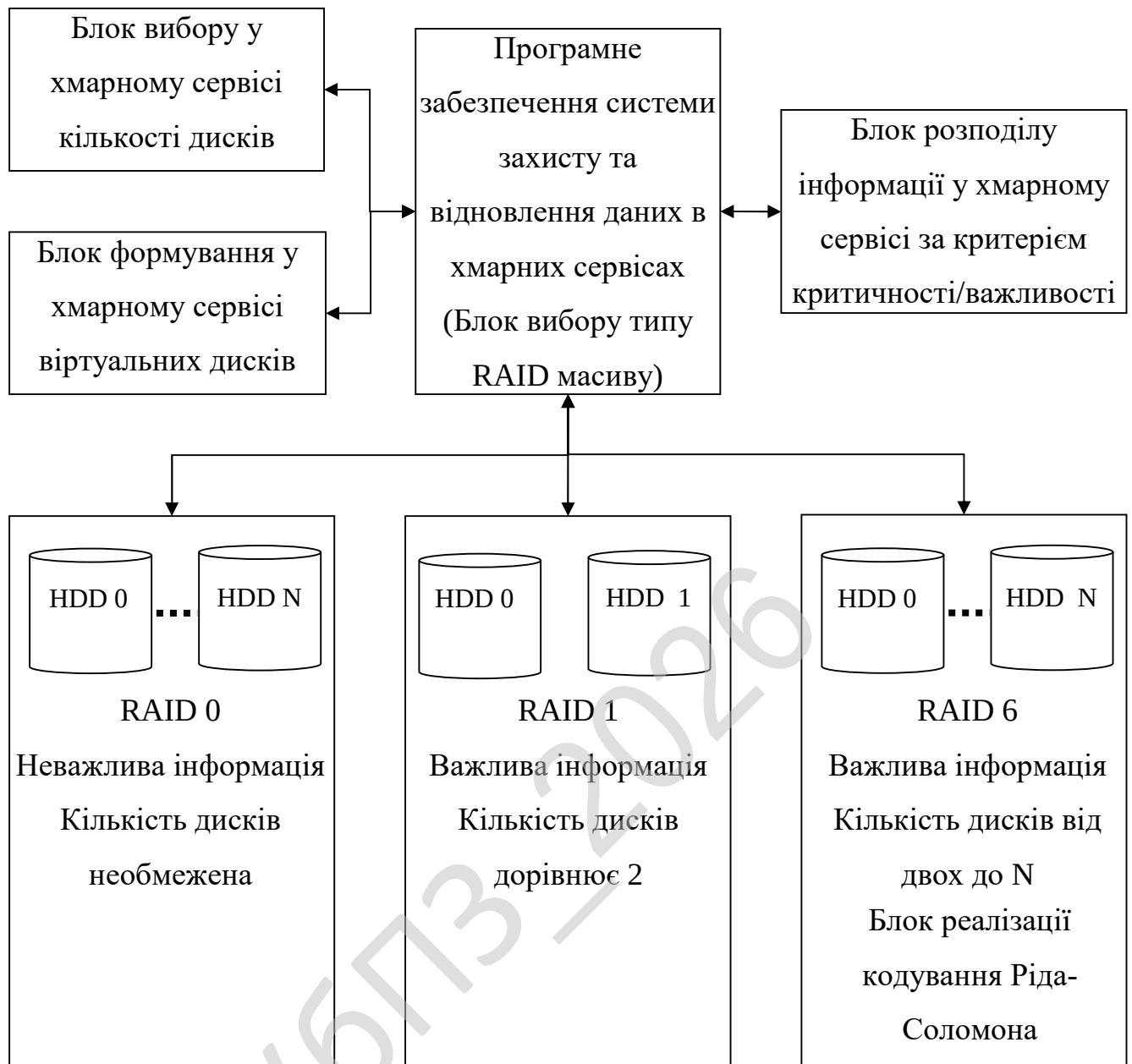


Рисунок 3.2 – Функціональна схема системи

У цей час широко використовується в системах відновлення даних з компакт-дисків, при створенні архівів з інформацією для відновлення у випадку ушкоджень, у завадостійкому кодуванні.

Коди Ріда-Соломона є важливим частковим випадком БЧХ-коду, корінь полінома, що породжує, якого лежать у тім же полі, над яким і будується код ($m = 1$).

Коди Боуза-Чоудхури-Хоквінгхема (БЧХ коди) – у теорії кодування це

широкий клас циклічних кодів, застосовуваних для захисту інформації від помилок. Відрізняється можливістю побудови коду із заздалегідь певними коригувальними властивостями, а саме, мінімальною кодовою відстанню.

Поліном, що породжує

Визначення поліномом, що породжує, циклічного (n, k) коду C називається такий ненульовий $g(x) = \sum_{i=0}^r g_i x^i$ поліном з C , ступінь якого найменша й коефіцієнт при старшому ступені $g_r = 1$.

Теорема 3.1

Якщо C – циклічний (n, k) код і $g(x)$ – його поліном, що породжує, тоді ступінь $g(x)$ дорівнює $r = n - k$ і кожне кодове слово може бути єдиним чином представлено у вигляді $c(x) = m(x)g(x)$, де ступінь $m(x)$ менше або дорівнює $k - 1$.

Теорема 3.2

$g(x)$ – поліном, що породжує, циклічного (n, k) коду є дільником двочлена $x^n - 1$.

Наслідок: у такий спосіб як поліном, що породжує, можна вибирати будь-який поліном, дільник $x^n - 1$. Ступінь обраного полінома буде визначати кількість перевірочних символів r , число інформаційних символів $k = n - r$.

Матриця, що породжує

Поліноми $g(x), xg(x), x^2g(x), \dots, x^{k-1}g(x)$ лінійно незалежні, інакше $m(x)g(x) = 0$ при ненульовому $m(x)$, що неможливо.

Значить кодові слова можна записувати, як і для лінійних кодів, наступним чином:

$$\overline{m}G = (m_0, m_1, \dots, m_{k-1}) \begin{bmatrix} g(x) \\ xg(x) \\ \dots \\ x^{k-1}g(x) \end{bmatrix} = m(x)g(x), \quad (3.1)$$

де G є матрицею, що породжує, $m(x)$ – інформаційним поліномом.

Матрицю G можна записати в символьній формі:

					ВКРБ-123.26.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		53

$$G = \begin{bmatrix} g_0 & g_1 & \dots & g_{r-1} & g_r & 0 & \dots & 0 \\ 0 & g_0 & \dots & g_{r-2} & g_{r-1} & g_r & \dots & 0 \\ \vdots & \vdots & \dots & \vdots & \vdots & \vdots & \dots & \vdots \\ 0 & 0 & \dots & 0 & g_0 & g_1 & \dots & g_r \end{bmatrix}$$

Перевірочна матриця

Для кожного кодового слова циклічного коду справедливо $c(x) = 0 \pmod{g(x)}$. Тому *перевірочну матрицю* можна записати як $H = [1 \ x \ x^2 \ \dots \ x^{n-2} \ x^{n-1}] \pmod{g(x)}$.

Тоді:

$$\bar{c}H^T = \sum_{i=0}^{n-1} c_i x^i \pmod{g(x)}. \quad (3.2)$$

Нехай α – елемент поля $GF(q)$ порядку n . Якщо α – *примітивний* елемент, то його порядок дорівнює $q-1$, т.е. $\alpha^{q-1}=1$, $\alpha^i \neq 1$, $0 < i < q-1$.

Тоді нормований поліном $g(x)$ мінімального ступеня над полем $GF(q)$, коріннями якого є $d-1$ підряд, що йдуть ступенів, $\alpha^{l_0}, \alpha^{l_0+1}, \dots, \alpha^{l_0+d-1}$ елемента α , є поліномом, що породжує, коду над полем $GF(q)$ $g(x) = (x - \alpha^{l_0})(x - \alpha^{l_0+1}) \dots (x - \alpha^{l_0+d-1})$; де l_0 – деяке ціле число (у тому числі 0 і 1), за допомогою якого іноді вдається спростити кодер. Звичайно покладається $l_0 = 1$. Ступінь багаточлена $g(x)$ дорівнює $d-1$.

Довжина отриманого коду n , мінімальна відстань d (мінімальна відстань d лінійного коду є мінімальним із всіх відстаней Хеммінга всіх пар кодових слів). Код містить $r = d-1 = \deg(g(x))$ перевірочний символ, де $\deg()$ позначає ступінь полінома; число інформаційних символів $k = n - r = n - d + 1$. У такий спосіб $d = n - k - 1$ і код Ріда-Соломона є *роздільним кодом з максимальною відстанню* (є оптимальним у змісті границі Синглтона).

Кодовий поліном $c(x)$ може бути отриманий з інформаційного полінома $m(x)$, $\deg m(x) \leq k-1$, шляхом перемножування $m(x)$ і $g(x)$: $c(x) = m(x)g(x)$.

Властивості

Код Ріда-Соломона над $GF(q^m)$, що виправляє t помилок, вимагає $2t$ перевірочних символів і з його допомогою виправляються довільні пакети

довжиною t і менше. Відповідно до теореми про границю Рейгера, коди Ріда-Соломона є оптимальними з погляду співвідношення довжини пакета й можливості виправлення помилок – використовуючи $2t$ додаткових перевірочних символів виправляються t помилок (t менш).

Теорема (границя Рейгера). Кожний лінійний блоковий код, що виправляє всі пакети довжиною t і менш, повинен містити щонайменше $2t$ перевірочних символів.

Виправлення багаторазових помилок

Код Ріда-Соломона є одним з найбільш потужних кодів, що виправляють багаторазові пакети помилок. Застосовується в каналах, де пакети помилок можуть утворюватися настільки часто, що їх уже не можна виправляти за допомогою кодів, що виправляють одиночні помилки. $(q^m - 1, q^m - 1 - 2t)$ -код Ріда-Соломона над полем $GF(q^m)$ з кодовою відстанню $d = 2t + 1$ можна розглядати як $((q^m - 1)m, (q^m - 1 - 2t)m)$ -код над полем $GF(q)$, що може виправляти будь-яку комбінацію помилок, зосереджену в t або меншому числі блоків з m символів.

Найбільше число блоків довжини m , які може торкнути пакет довжини l_i , де $l_i \leq mt_i - (m - 1)$, не перевершує t_i , тому код, що може виправити t блоків помилок, завжди може виправити й будь-яку комбінацію з r пакетів загальної довжини l , якщо $l + (m - 1) \leq mt$.

Практична реалізація

Кодування за допомогою коду Ріда-Соломона може бути реалізовано двома способами: систематичним і несистематичним.

При несистематичному кодуванні інформаційне слово множиться на якийсь полином, що неприводиться, у полі Галуа. Отримане закодоване слово повністю відрізняється від вихідного й для добування інформаційного слова потрібно виконати операцію декодування й уже потім можна перевірити дані на зміст помилок. Таке кодування вимагає більші витрати ресурсів тільки на добування інформаційних даних, при цьому вони можуть бути без помилок.

					ВКРБ-123.26.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		55

При систематичному кодуванні до інформаційного блоку з k символів приписуються $2t$ перевірочних символів, при обчисленні кожного перевірочного символу використовуються всі k символів вихідного блоку.

У цьому випадку немає витрат ресурсів при добуванні вихідного блоку, якщо інформаційне слово не містить помилок, але кодер/декодер повинен виконати $k(n - k)$ операцій додавання й множення для генерації перевірочних символів. Крім того, тому що всі операції проводяться в поле Галуа, те самі операції кодування/декодування вимагають багато ресурсів і часу. Швидкий алгоритм декодування, заснований на швидкому перетворенні Фур'є, виконується за час порядку $\ln n^2$.

Кодування

При операції кодування інформаційний поліном множиться на багаточлен, що породжує. Множення вихідного слова S довжини k на не приводиться полином, що, при систематичному кодуванні можна виконати в такий спосіб:

- До вихідного слова приписуються $2t$ нулів, виходить поліном $T = Sx^{2t}$.
- Цей поліном ділиться на поліном, що породжує, G , перебуває залишок R , $Sx^{2t} = QG + R$, де Q – частка.
- Цей залишок й буде коригувальним кодом Ріда-Соломона, він приписується до вихідного блоку символів. Отримане кодове слово $C = Sx^{2t} + R$.

Кодер будується зі регістрів зсуву, суматорів і перемножувачів. Регістр зсуву складається з комірок пам'яті, у кожній з яких перебуває один елемент поля Галуа.

Наведений як приклад кодер Ріда-Соломона генерує 16 коригувальних байт, що дозволяє виправляти до 8 і виявляти до 16 помилок у кадрі даних.

Перемножувачі на константи $GF(0)...GF(15)$ у полі Галуа реалізуються в такий спосіб: спочатку вихідне число й константа перетворюються в індексну форму, потім складаються в межах байта без обліку переносу.

Результатом операції є результат додавання, перетворена обернено в поліноміальну форму.

					ВКРБ-123.26.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		56

При переході від однієї форми подання даних до іншої доцільно використовувати таблицю істинності розміром 256 байт, що становить ємність одного ЕАВ (Embedded Array Block – блок зосередженої пам'яті). Для реалізації кодера потрібно 16 таких перемножувачів, при цьому те саме число множиться на різні константи, що дозволяє використовувати для його перекладу в індексну форму один ЕАВ.

Для перекладу результатів у поліноміальну форму потрібно вже 16 таких таблиць, що вимагає застосування ІМС FPGA дуже великої ємності. У запропонованій схемі використовується тактування кодера із частотою, в 8 разів перевищуючу частоту надходження байт даних.

Це дає можливість використовувати дві пари «суматор – ЕАВ», мультиплексує константи на входах суматорів і дозволяючи роботу регістрів-накопичувачів у моменти появи відповідних даних на виходах засувки ЕАВ.

На структурній схемі кодера (рисунок 3.2) символу «С» відповідають дві константи $GF(n)$.

Символ «L» у логіці регістрів-накопичувачів відповідає наступний: вихід компаратора нуля (символи CMP0 і SYNC) дозволяє роботу схеми «АБО що виключає », на входи якої подаються вихід попереднього регістра й ЕАВ. Якщо ж вектор зворотного зв'язку дорівнює "0", схема пропускає дані з виходу попереднього регістра-накопичувача на вхід наступного.

У результаті кодер з урахуванням схеми синхронізації (на рисунку не показана) займає 255 LE (Logic Element – логічний елемент) і 3 ЕАВ, що дозволяє розмістити його в ІМС EPF10K10. Після оптимізації розміщення схеми на кристалі FPGA швидкодія схеми досяглася 11,57 МГц (частота надходження байт даних, далі – байтова частота).

При використанні ІМС EPF10K20, у складі якої 6 ЕАВ, використовуючи 4 пари "суматор – ЕАВ", можна тактувати кодер із частотами, що перевищують байтову частоту не в 8, а в 4 рази, що дозволить підняти її до 25...30 МГц.

Декодування

Декодер, що працює по авторегресивному спектральному методі декодування, послідовно виконує наступні дії:

- Обчислює синдром помилки.
- Будує поліном помилки.
- Знаходить корінь даного полінома.
- Визначає характер помилки.
- Виправляє помилки.

Обчислення синдрому помилки

Обчислення синдрому помилки виконується синдромним декодером, що ділить кодове слово на багаточлен, що породжує. Якщо при діленні виникає остача, то в слові є помилка. Остача від ділення є синдромом помилки.

Побудова полінома помилки

Обчислений синдром помилки не вказує на положення помилок. Ступінь полінома синдрому дорівнює $2t$, що багато менше ступеня кодового слова n . Для одержання відповідності між помилкою і її положенням у повідомленні будується поліном помилок.

Поліном помилок реалізується за допомогою алгоритму Берлекемпа-Мессі, або за допомогою алгоритму Евкліда. Алгоритм Евкліда має просту реалізацію, але вимагає більших витрат ресурсів. Тому частіше застосовується більше складний, але менш затратоємний алгоритм Берлекемпа-Мессі. Коефіцієнти знайденого полінома безпосередньо відповідають коефіцієнтам помилкових символів у кодовому слові.

Алгоритм Евкліда виконується наступним чином.

Нехай a і b суть цілі числа, не рівні одночасно нулю, і послідовність чисел $a, b, r_1 > r_2 > r_3 > r_4 > \dots > r_n$ визначена тим, що кожне r_k це остача від ділення перед-попереднього числа на попереднє, а передостаннє ділиться на останнє націло, тобто

$$a = bq_0 + r_1$$

$$\begin{aligned}
 b &= r_1 q_1 + r_2 \\
 r_1 &= r_2 q_2 + r_3 \\
 &\dots \\
 r_{n-1} &= r_n q_n
 \end{aligned}
 \tag{3.3}$$

Тоді (a,b) , найбільший загальний дільник a і b , дорівнює r_n , останньому ненульовому члену цієї послідовності.

Існування таких $r_1, r_2, \dots$, тобто можливість ділення з остачею m на n для будь-якого цілого m і цілого $n \neq 0$, доводиться індукцією по m .

Коректність цього алгоритму випливає з наступних двох тверджень:

- Нехай $a = bq + r$, тоді $(a,b) = (b,r)$.
- $(0,r) = r$. для будь-якого ненульового r .

Знаходження корня

На цьому етапі шукаються коріння полінома помилки, що визначають положення перекручених символів у кодовому слові. Реалізується за допомогою процедури Ченя, рівносильній повному перебору. У поліном помилок послідовно підставляються всі можливі значення, коли поліном звертається в нуль – коріння знайдені.

Визначення характеру помилки і її виправлення

По синдрому помилки й знайдених корінь полінома за допомогою алгоритму Форни визначається характер помилки й будується маска перекручених символів. Ця маска накладається на кодове слово за допомогою операції XOR і перекручені символи відновлюються. Після цього відкидаються перевірочні символи й виходить відновлене інформаційне слово.

Розглянувши усі блоки функціональної схеми перейдемо до розгляду діаграми взаємодії процесів, які відбуваються у системі.

3.4 Розробка діаграми процесів

Розглянемо розроблену діаграму процесів яка зображена на рисунку 3.3. Основна будова діаграми процесів полягає у графічному представленні складу сукупностей даних, що характеризуються як співвідношення різних частин кожної з сукупностей.

Склад статистичної сукупності графічно може бути представлений як за допомогою абсолютних, так і відносних показників. Графічне зображення складу сукупності по абсолютними і відносними показниками сприяє проведенню більш глибокого аналізу і дозволяє проводити аналіз системи.

Діаграма взаємодії процесів використовується для візуалізації процесів обробки даних (структурне проектування).

Для розробника вважається звичним спочатку креслити діаграму взаємодії процесів даних рівня контексту, завдяки чому буде показано взаємодію системи.

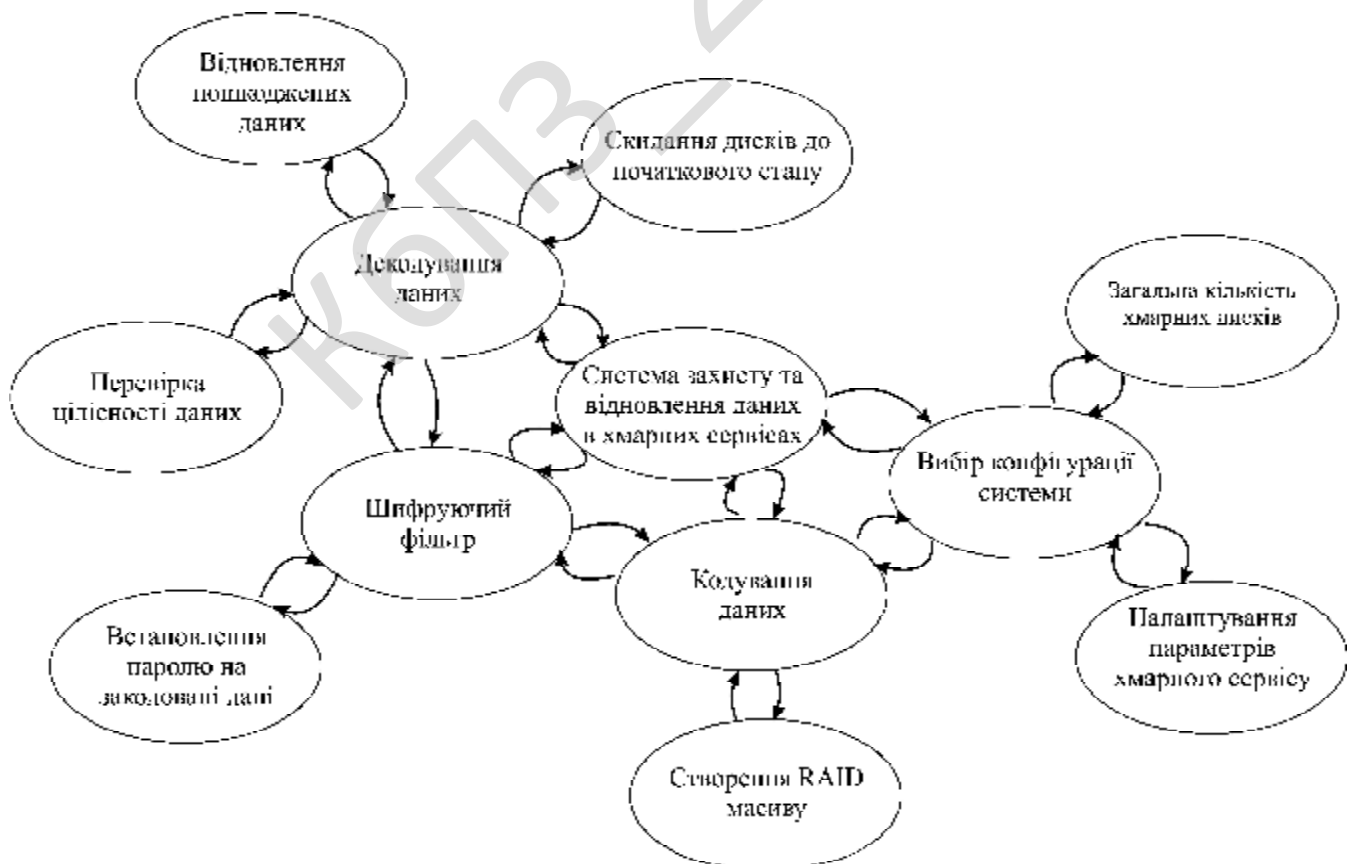


Рисунок 3.3 – Діаграма взаємодії процесів

Ця діаграма в подальшому підлягає уточненню шляхом деталізації процесів та потоків даних з метою показати систему що розробляється.

При детальному її розгляді можна побачити як саме проходить взаємодія у розробленій системі. Використовується модель проектування, графічне представлення «потоків» даних в інформаційній системі.

Діаграми потоків даних містять чотири типи елементів:

- Зовнішні по відношенню до системи сутності.
- Потоки даних між елементами трьох попередніх типів.
- Процеси які являють собою трансформацію даних в рамках описуваної системи.
- Сховища даних (репозиторії).

Таким чином, розглянувши опис системи, структурну, функціональну схеми системи, та діаграму взаємодії процесів перейдемо до опису блок-схем основної програми, та підпрограм, які використовуються, для реалізації системи.

4 РЕАЛІЗАЦІЯ ПРОЕКТУ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ПРАВИЛЬНІСТЬ ПРОЕКТНИХ РІШЕНЬ

4.1 Блок-схеми та опис алгоритмів функціонування системи

Розглянемо реалізацію бакалаврської дипломної роботи. Були проведені розрахунки і підібрані набори тестових даних для перевірки правильності реалізації проектних рішень.

Було створено блок-схеми роботи системи. Перед їх розглядом необхідно провести роз'яснення який саме тип блок-схем використовується. Блок-схема це представлення задачі для її аналізу або розв'язування за допомогою спеціальних символів (геометричних образів), які позначають такі елементи, як операції, потік, дані тощо.

Блок вхідних та вихідних даних прийнято позначати паралелограмом, блок обчислень (обробки) даних – прямокутником, блок прийняття рішень – ромбом, еліпсом – початок та кінець алгоритму.

У інформаційних технологіях функціональна схема складається з функціональних блоків, які являють собою конструктивно відособлені частини (елементи або пристрої) автоматичних систем, які виконують певні функції. Функціональні блоки на схемі позначають прямокутниками, всередині яких надписують їх найменування відповідно до функцій, що виконуються. Зв'язки між функціональними блоками (внутрішні впливи) позначаються лініями зі стрілками, які вказують напрям впливів.

Функціональні схеми можуть виконуватися в укрупненому і розгорненому вигляді. У першому випадку на схемі зображають найважливіші блоки системи і зв'язки між ними.

					ВКРБ-123.26.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		62

У другому варіанті схема відображається більш детально, що полегшує її читання та ілюструє принцип роботи.

Основні елементи схем алгоритму це термінатор, процес, рішення, зумовлений процес (підпрограма), дані та з'єднувач. Термінатор це елемент відображає вхід із зовнішнього середовища або вихід з неї (найчастіше застосування – початок і кінець програми). Всередині фігури записується відповідна дія.

Процес це виконання однієї або кількох операцій, обробка даних будь-якого виду (зміна значення даних, форми подання, розташування). Всередині фігури записують безпосередньо самі операції. Рішення це показує рішення або функцію перемикального типу з одним входом і двома або більше альтернативними виходами, з яких тільки один може бути обраний після обчислення умов, визначених всередині цього елемента.

Вхід в елемент позначається лінією, що входить зазвичай у верхню вершину елемента. Якщо виходів два чи три то зазвичай кожен вихід позначається лінією, що виходить з решти вершин (бічних і нижній). Якщо виходів більше трьох, то їх слід показувати однією лінією, що виходить з вершини (частіше нижній) елемента, яка потім розгалужується.

Відповідні результати обчислень можуть записуватися поруч з лініями, що відображають ці шляхи. Зумовлений процес (підпрограма) це символ відображає виконання процесу, що складається з однієї або кількох операцій, що визначені в іншому місці програми (у підпрограмі, модулі). Всередині символу записується назва процесу і передані в нього дані. Дані це перетворення у форму, придатну для обробки (введення) або відображення результатів обробки (виведення). Цей символ не визначає носія даних (для вказівки типу носія даних використовуються специфічні символи). З'єднувач це символ відображає вихід в частину схеми і вхід з іншої частини цієї схеми.

Використовується для обриву лінії та продовження її в іншому місці (приклад: поділ блок-схеми, що не поміщається на листі). Відповідні сполучні символи повинні мати одне (при тому унікальне) позначення.

Блок-схеми показують весь процес роботи системи з підсистемами та частково доказують правильність вибраних проектних рішень. Тому від точності і детальної блок-схеми залежить результат всієї програми.

При виборі початкової точки відліку при побудові схем було враховано, що виходячи з вибору мови програмування і інших технічних засобів, програма буде об'єктно-орієнтована що вимагає високого рівня декомпозиції задач на класи.

На рисунку 4.1 зображена основна блок-схема програми, на рисунку 4.2 зображено роботу підпрограми.

З яких видно що робота основної програми складається з початкових етапів ініціалізації ПЗ, перевірки наявності ресурсів системи, блоку початку основного циклу з чеканням запиту від користувача в якому відбувається виклик підсистеми та останньої стадії – перевірка поточного стану з завершенням роботи розробленого ПЗ. При роботі підпрограми виконується основний функціонал системи з циклічними послідовностями, перевіркою поточного стану та поверненням в основну програму прапорів стану виконання.

Розглянемо визначення API. Це прикладний програмний інтерфейс (Application Programming Interface, API) – набір визначень підпрограм, протоколів взаємодії та засобів для створення програмного забезпечення.

Спрощено – це набір чітко визначених методів для взаємодії різних компонентів. API надає розробнику засоби для швидкої розробки програмного забезпечення. API може бути для веб-базованих систем, операційних систем, баз даних, апаратного забезпечення, програмних бібліотек.

Одним з найпоширеніших призначень API є надання набору широко використовуваних функцій, наприклад для малювання вікна чи іконок на екрані.

					ВКРБ-123.26.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		64

API є абстрактним поняттям – програмне забезпечення, що пропонує деякий API, часто називають реалізацією (implementation) даного API.

У багатьох випадках API є частиною набору розробки програмного забезпечення, водночас, набір розробки може включати як API, так і інші інструменти/апаратне забезпечення, отже ці два терміни не є взаємозамінювані.

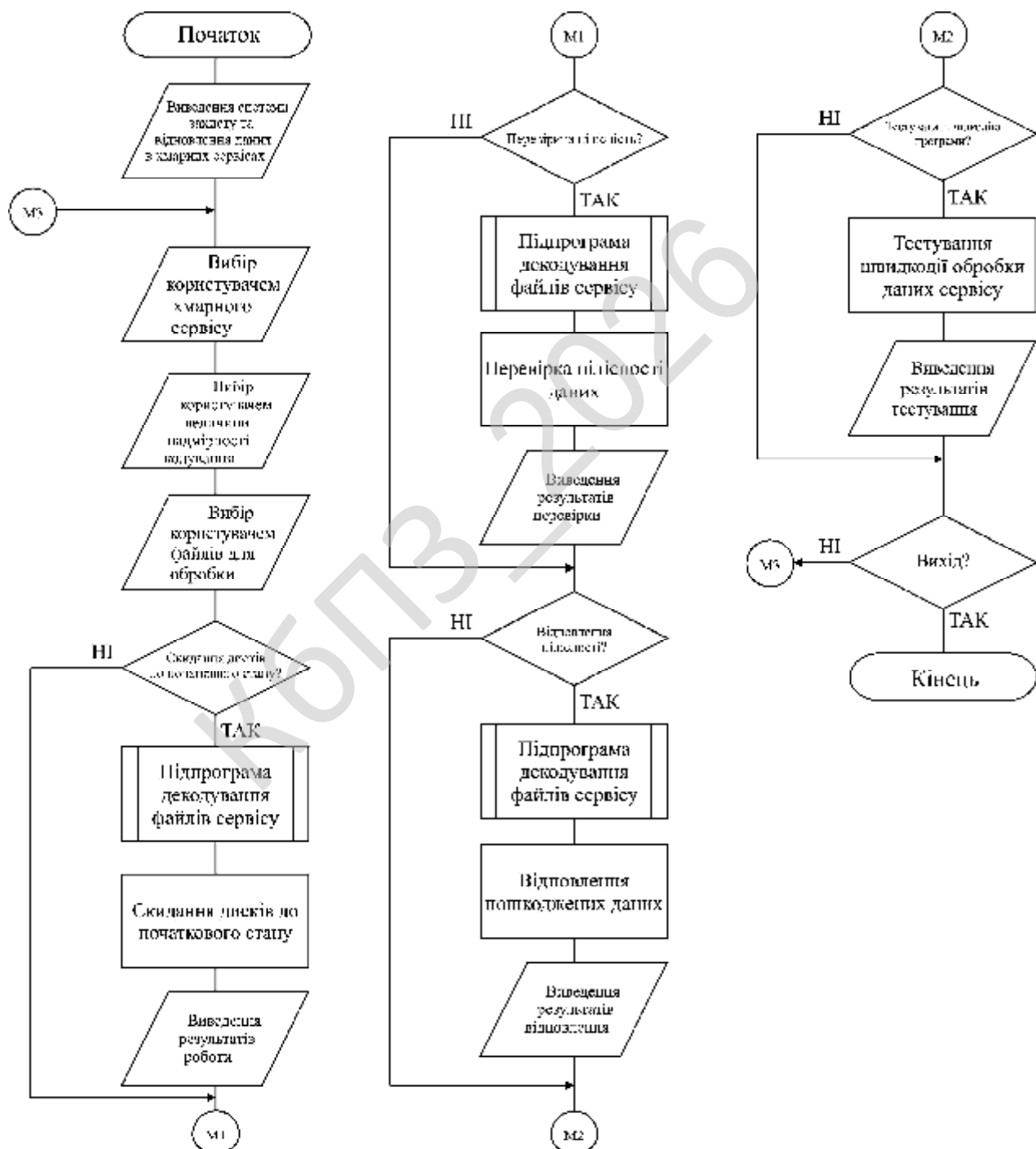


Рисунок 4.1 – Блок-схема основної програми

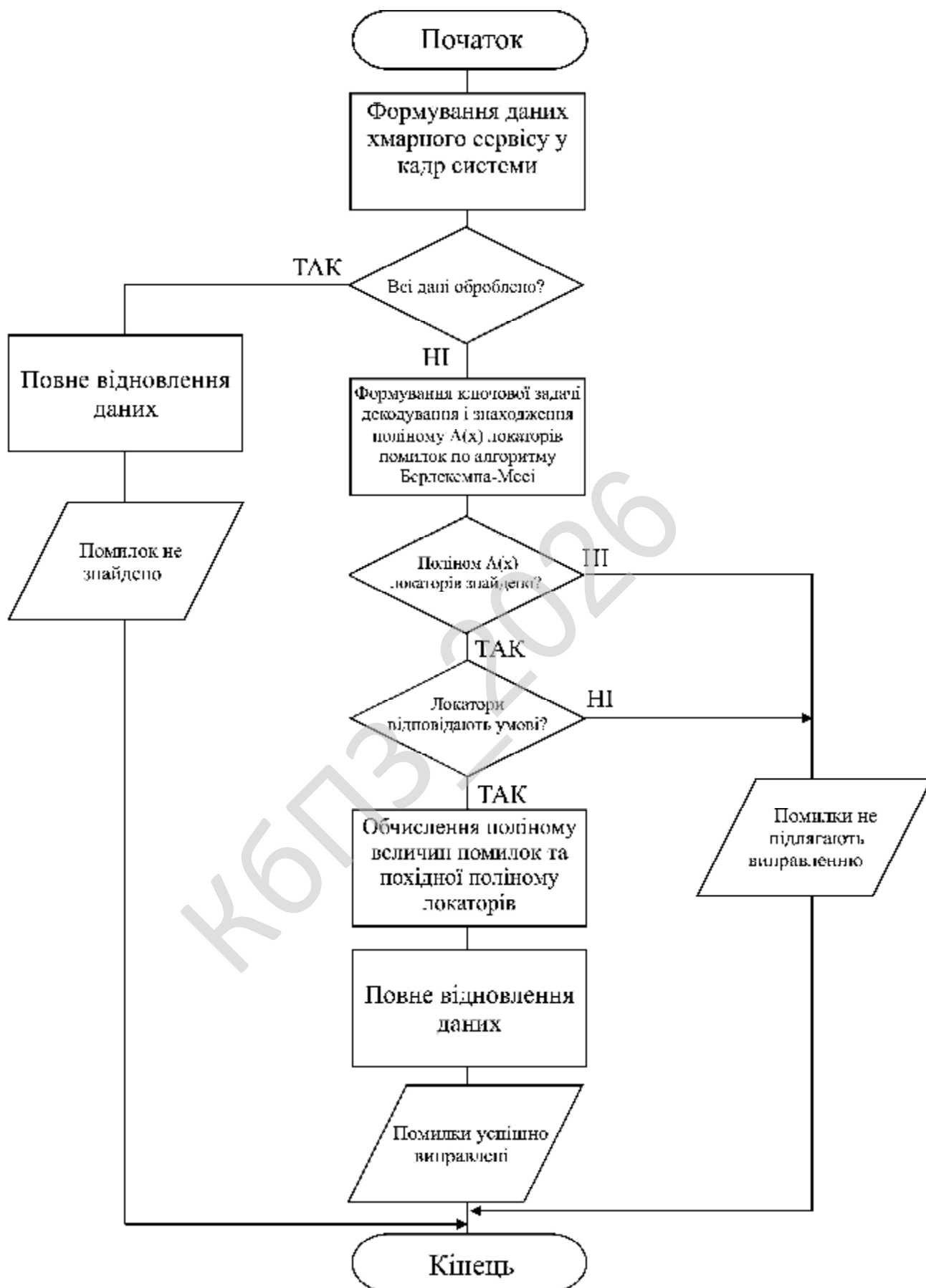


Рисунок 4.2 – Блок-схема роботи підпрограми

Високорівневі API часто програють у гнучкості. Виконання деяких функцій нижчого рівня стає набагато складнішим, або навіть неможливим.

В об'єктно-орієнтованих мовах, прикладний програмний інтерфейс зазвичай включає в себе опис набору визначень класу, з набором форм поведінки, пов'язаних з цими класами. Це абстрактне поняття пов'язане з реальними функціями, які надані або надаватимуться, класами, які реалізуються в методах класу.

Прикладний програмний інтерфейс в даному випадку можна розглядати як сукупність всіх методів, які публічно доступні в класах (зазвичай званий інтерфейс класу). Це означає, що прикладний програмний інтерфейс вказує методи, за допомогою яких взаємодіє з об'єктами, отриманими з визначень класів і обробляє їх.

У більш загальному плані можна визначити Прикладний Програмний Інтерфейс як сукупність усіх видів об'єктів, які можна вивести з визначення класу, і пов'язаних з ними можливих варіантів поведінки.

Наприклад: клас, що представляє Stack, може просто виставити публічно два методи Push() (для додавання нового елемента в стек) і Pop() (для вилучення останнього пункту, ідеально розташований на вершині стека).

У цьому випадку Прикладний Програмний Інтерфейс може бути інтерпретованим як два методи pop() і push(), або, більш широко, використовується варіант, коли можна використовувати елемент типу Stack, який реалізує поведінку стека, надаючи йому можливість для додавання / видалення елементів з вершини. Друга інтерпретація видається більш доречною в дусі об'єктно-орієнтованого підходу.

Якість документації, пов'язаної з Прикладним Програмним Інтерфейсом, є часто ключовим фактором, що визначає його успішність з точки зору простоти використання.

ППІ, як правило, пов'язаний із бібліотеками програмного забезпечення: ППІ описує і вказує очікувану поведінку в той час, як бібліотека є фактичною

					ВКРБ-123.26.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		67

реалізацією даного набору правил. Один ППІ може мати декілька реалізацій (або жодної, будучи абстрактним) у вигляді різних бібліотек, які мають такий же інтерфейс.

Прикладний програмний інтерфейс також може бути пов'язаним з платформами програмування: платформа може бути заснована на кількох бібліотеках реалізує декілька інтерфейсів ППІ, але на відміну від звичайного використання ППІ, доступ до поведінки вбудований в платформу опосередкований шляхом розширення його змісту новими класами і вставлений в саму платформу. Крім того, загальний потік управління програми може бути під контролем абонента. Прикладний програмний інтерфейс може бути також реалізацією протоколу. Коли ППІ реалізує протокол, він може бути заснованим на проксі-методах віддалених викликів, що засновані на протоколі зв'язку. Роль ППІ може заключатися саме в тому, щоб приховати деталі транспортного протоколу. Наприклад: RMI є ППІ, який реалізує протокол або JRMP IIOP як RMI-IIOP.

Протоколи, як правило, розподіляються між різними технологіями і зазвичай дозволяють різним технологіям обмінюватися інформацією, діючи як абстракція між двома світами. ППІ, як правило, є специфічним для конкретної технології: звідси, інтерфейси даної мови не можуть бути використані на інших мовах, якщо виклики функції не будуть перетворені з конкретної адаптації бібліотеки.

Деякі мови, серед яких такі, що працюють на віртуальних машинах (наприклад: мови, сумісні з NET CLI середовища CLR і JVM сумісних мов у віртуальній машині Java) можуть ділитися програмними інтерфейсами.

У цьому випадку віртуальна машина дозволяє мові взаємодії завдяки спільному знаменнику віртуальної машини, що абстрагується від конкретної мови, використовувати проміжний байт-код і його мову.

При використанні прикладного програмного інтерфейсу в контексті веб-розробки, як правило, ППІ визначається набором повідомлень запиту HTTP, також визначається структура повідомлень-відповідей, зазвичай у розширенні

					ВКРБ-123.26.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		68

мови розмітки XML або в форматі об'єктного запису JavaScript (JSON). У той час як прикладний програмний інтерфейс у Web історично був практично синонімом для веб-служби, останнім часом тенденція змінилась (так званий Web 2.0) на відхід від Simple Object Access Protocol (SOAP) на основі веб-сервісів і сервіс-орієнтованої архітектури (SOA) на більш прямі передачі репрезентативного стану (REST) стилів веб-ресурсів та ресурсів-орієнтованої архітектури (ROA).

Частина цієї тенденції пов'язана з рухом Семантичного веб-ресурсу до Опису Платформ (RDF), Концепції розвитку веб-технологій інженерних онтологій. Прикладні програмні інтерфейси у Web, що дозволяють комбінувати декількома прикладними програмними інтерфейсами в нові додатки називають гібридними.

Архітектура клієнт-сервер є одним із архітектурних шаблонів програмного забезпечення та є домінуючою концепцією у створенні розподілених мережних програм і передбачає взаємодію та обмін даними між ними. Вона передбачає такі основні компоненти:

- набір серверів, які надають інформацію або інші послуги програмам, які звертаються до них;
- набір клієнтів, які використовують сервіси, що надаються серверами;
- мережа, яка забезпечує взаємодію між клієнтами та серверами.

Сервери є незалежними один від одного. Клієнти також функціонують паралельно і незалежно один від одного. Немає жорсткої прив'язки клієнтів до серверів. Більш ніж типовою є ситуація, коли один сервер одночасно обробляє запити від різних клієнтів; з іншого боку, клієнт може звертатися то до одного сервера, то до іншого. Клієнти мають знати про доступні сервери, але можуть не мати жодного уявлення про існування інших клієнтів.

Дуже важливо ясно уявляти, хто або що розглядається як «клієнт». Можна говорити про клієнтський комп'ютер, з якого відбувається звернення до інших комп'ютерів. Можна говорити про клієнтське та серверне програмне забезпечення. Нарешті, можна говорити про людей, які бажають за допомогою

					ВКРБ-123.26.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		69

відповідного програмного та апаратного забезпечення отримати доступ до тієї чи іншої інформації.

Загальноприйнятим є положення, що клієнти та сервери – це перш за все програмні модулі. Найчастіше вони знаходяться на різних комп'ютерах, але бувають ситуації, коли обидві програми – і клієнтська, і серверна, фізично розміщуються на одній машині; в такій ситуації сервер часто називається локальним.

Модель клієнт-серверної взаємодії визначається перш за все розподілом обов'язків між клієнтом та сервером. Логічно можна відокремити три рівні операцій:

- рівень представлення даних, який по суті являє собою інтерфейс користувача і відповідає за представлення даних користувачеві і введення від нього керуючих команд;
- прикладний рівень, який реалізує основну логіку ПЗ і на якому здійснюється необхідна обробка інформації;
- рівень управління даними, який забезпечує зберігання даних та доступ до них.

Дворівнева клієнт-серверна архітектура передбачає взаємодію двох програмних модулів – клієнтського та серверного. В залежності від того, як між ними розподіляються наведені вище функції, розрізняють:

- модель тонкого клієнта, в рамках якої вся логіка ПЗ та управління даними зосереджена на сервері. Клієнтська програма забезпечує тільки функції рівня представлення;
- модель товстого клієнта, в якій сервер тільки керує даними, а обробка інформації та інтерфейс користувача зосереджені на стороні клієнта. Товстими клієнтами часто також називають пристрої з обмеженою потужністю: кишенькові комп'ютери, мобільні телефони та ін.

Типовим прикладом клієнт-серверної взаємодії є WWW. Існує величезна кількість веб-серверів, на яких розміщується та чи інша інформація. У

					ВКРБ-123.26.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		70

найпростішому випадку ця інформація являє собою набір веб-сторінок, які можуть зберігатися на сервері у вигляді файлів, розмічених за допомогою мови розмітки HTML. Але ситуація, як правило, є складнішою; значна частина веб-ресурсів на сучасному етапі є динамічними, тобто вони не існують в заздалегідь підготовленому вигляді, а створюються безпосередньо в процесі обробки запиту від користувача.

Для того, щоб людина, яка працює в Інтернеті, могла переглянути ту чи іншу сторінку, на її комп'ютері повинно бути встановлено відповідне програмне забезпечення. Програми для перегляду веб-сторінок називаються браузером.

Але, крім браузерів, до серверів можуть звертатися і інші клієнти, а саме – автономні програми. Вони можуть передбачати взаємодію з людиною, а можуть працювати в цілком автоматичному режимі. Типовим класом таких програм є роботи, призначені для автоматичного перегляду веб-ресурсів. Зокрема, роботи є важливим елементом пошукових систем і використовуються ними для перегляду сторінок і збору інформації про них.

Для запиту до веб-сервера клієнтська програма повинна задати місцезнаходження комп'ютера, на якому розміщується серверна програма, назву потрібного документа і, можливо, інші дані, які специфікують запит. Мережа забезпечує знаходження сервера і передачу йому клієнтського запиту. Серверні програми обробляють цей запит, відповідь пересилається по мережі клієнтові.

Трирівнева клієнт-серверна архітектура, яка почала розвиватися з середини 90-х років, передбачає відділення прикладного рівня від управління даними. Відокремлюється окремий програмний рівень, на якому зосереджується прикладна логіка ПЗ. Програми проміжного рівня можуть функціонувати під управлінням спеціальних серверів ПЗ, але запуск таких програм може здійснюватися і під управлінням звичайного веб-сервера. Нарешті, управління даними здійснюється сервером даних.

Для роботи з системою користувач використовує стандартне програмне забезпечення – звичайний браузер. Це позбавляє його необхідності завантажувати

					ВКРБ-123.26.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		71

та інсталиувати спеціальні програми (хоча інколи така необхідність все-таки виникає).

Але користувачеві слід надати в розпорядженні інтерфейс, який дозволяв би йому взаємодіяти з системою і формувати запити до неї. Форми, що визначають цей інтерфейс, розміщуються на веб-сторінках та завантажуються разом з ними.

Веб-оглядач формує запит та пересилає його до сервера, який здійснює обробку. При необхідності сервер викликає серверні програмні модулі, які забезпечують обробку запиту і в разі потреби звертаються до сервера даних. Сервер даних здійснює операції з даними, що зберігаються в системі та складають її інформаційну основу. Зокрема, він може здійснити вибірку з інформаційної бази відповідно до запиту та передати її модулю проміжного рівня для подальшої обробки. Дані, з якими працює сервер даних, найчастіше організовані як реляційна база даних.

Найчастіше веб-сервер і серверні модулі проміжного рівня розміщуються на одному комп'ютері, хоч і являють собою окремі і логічно незалежні програмні модулі.

На сучасному етапі для програмування модулів проміжного рівня використовується мова серверних сценаріїв PHP, а для управління даними – СУБД MySQL. Таким чином, зв'язку PHP-MySQL слід розглядати як стандартний інструмент для створення порівняно простих інтерактивних веб-сайтів та систем електронної комерції; близько 90% комерційних систем сьогодні створюється саме на цій основі. Водночас як засоби управління даними, так і middleware-засоби можуть бути найрізноманітнішими. Так, для створення серверних програм, крім PHP, широко застосовуються Java, Perl, Python, Delphi.

Взагалі, технології створення розподілених, зокрема веб-програм, стрімко розвиваються. Слід згадати про технології EJB (Enterprise Java Beans), CORBA, а також про .NET – порівняно нову ініціативу компанії Microsoft. Для зберігання даних та їх передачі часто використовується так звана розширювана мова

					ВКРБ-123.26.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		72

розмітки XML (Extensible Markup Language).

При розробці ПЗ було використано V-Model (або VEE модель) є моделлю розробки інформаційних систем (ИС), спрямованої на спрощення розуміння складнощів, пов'язаних з розробкою систем. Вона використовується для визначення єдиної процедури розробки програмного забезпечення, апаратного забезпечення та людино-машинного інтерфейсу.

Концепція V-подібної моделі була розроблена Німеччиною та США в кінці 1980-х років незалежно один від одного:

– Німецька V-модель була розроблена аерокосмічної компанією IABG в Оттобрунні поряд з Мюнхеном у сприянні з Федеральним департаментом з закупівлі озброєнь в Кобленці, для Міністерства оборони Німеччини. Модель була прийнята німецькою федеральною адміністрацією для цивільних потреб влітку 1992.

– Американська V-Model (VEE) була розроблена національною радою з системної інженерії (міжнародна – з 1995 року) для супутникових систем, включаючи обладнання, програмне забезпечення та взаємодію з користувачами.

Сучасною версією V-Model є V-Model XT, яка була затверджена в лютому 2005 року. V-модель використовується для управління процесом розробки програмного забезпечення для німецької федеральної адміністрації.

Зараз вона є стандартом для німецьких урядових і оборонних проектів, а також для виробників ПЗ в Німеччині. V-Model являє собою скоріше набір стандартів у галузі проектів, що стосуються розробки нових продуктів. Ця модель багато в чому схожа з Prince2 і описує методи як для проектного управління, так і для системного розвитку.

Основні принципи

Основний принцип V-подібної моделі полягає в тому, що деталізація проекту зростає при русі зліва направо, одночасно з плином часу, і ні те, ні інше не може повернути назад. Ітерації в проекті виробляються по горизонталі, між лівою і правою сторонами літери.

					ВКРБ-123.26.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		73

Стосовно до розробки інформаційних систем V-Model – варіація каскадної моделі, в якій завдання розробки йдуть зверху вниз по лівій стороні букви V, а завдання тестування – вгору по правій стороні букви V. Усередині V проводяться горизонтальні лінії, що показують, як результати кожної з фаз розробки впливають на розвиток системи тестування на кожній із фаз тестування.

Модель базується на тому, що прийнятно-здавальні випробування ґрунтуються, насамперед, на вимогах, системне тестування – на вимогах та архітектурі, комплексне тестування – на вимогах, архітектурі та інтерфейсах, а компонентне тестування – на вимогах, архітектурі, інтерфейсах та алгоритмах

Цілі

V-модель забезпечує підтримку у плануванні та реалізації проекту. В ході проекту ставляться такі завдання:

1. Мінімізація ризиків: V-подібна модель робить проект більш прозорим і підвищує якість контролю проекту шляхом стандартизації проміжних цілей і опису відповідних їм результатів та відповідальних осіб. Це дозволяє виявляти відхилення в проекті і ризики на ранніх стадіях і покращує якість управління проектом.

2. Підвищення та гарантії якості: V-Model – стандартизована модель розробки, що дозволяє домогтися від проекту результатів бажаної якості. Проміжні результати можуть бути перевірені на ранніх стадіях. Універсальне документування полегшує читаність, зрозумілість та контрольованість.

3. Зменшення загальної вартості проекту: Ресурси на розробку, виробництво, управління і підтримку можуть бути заздалегідь прораховані та проконтрольовані. Отримувані результати також універсальні і легко прогножуються. Це зменшує витрати на подальші стадії та проекти.

4. Підвищення якості комунікації між учасниками проекту: Універсальний опис усіх елементів та умов полегшує взаєморозуміння всіх учасників проекту. Таким чином, зменшуються неточності у розумінні між користувачем, покупцем, постачальником і розробником.

					ВКРБ-123.26.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		74

Переваги:

– Користувачі V-Model беруть участь у розробці та підтримці V-моделі. Комітет з контролю за змінами підтримує проект і збирається раз на рік для обробки всіх отриманих запитів на внесення змін до V-Model.

– На старті будь-якого проекту V-подібна модель може бути адаптована під цей проект, так як ця модель не залежить від типів організацій та проектів.

– V-model дозволяє розбити діяльність на окремі кроки, кожен з яких буде включати в себе необхідні для нього дії, інструкції до них, рекомендації та докладне пояснення діяльності.

Обмеження. Наступні моменти не враховуються в V-моделі, але можуть бути розглянуті окремо, або можливо адаптувати модель під них:

– Не регулюється розміщення контрактів на обслуговування.

– Організація і виконання управління, обслуговування, ремонту та утилізації системи не враховуються в V-моделі. Однак, планування і підготовка до цих операцій моделлю розглядаються.

– V-подібна модель більше стосується розробки програмного забезпечення в проекті, ніж всієї організації процесу.

Переваги:

– У моделі особливе значення надається плануванню, спрямованому на верифікацію та атестацію розроблювального продукту на ранніх стадіях його розробки. Фаза модульного тестування підтверджує правильність деталізованого проектування. Фази інтеграції та тестування реалізують архітектурне проектування або проектування на вищому рівні. Фаза тестування системи підтверджує правильність виконання етапу вимог до продукту і його специфікації.

– У моделі передбачені атестація та верифікація всіх зовнішніх і внутрішніх отриманих даних, а не тільки самого програмного продукту.

– У V-подібної моделі визначення вимог виконується перед розробкою проекту системи, а проектування ПО – перед розробкою компонентів.

					ВКРБ-123.26.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		75

– Модель визначає продукти, які повинні бути отримані в результаті процесу розробки, причому кожні отримані дані повинні піддаватися тестуванню.

– Завдяки моделі менеджери проекту можуть відслідковувати хід процесу розробки, так як в даному випадку цілком можливо скористатися тимчасовою шкалою, а завершення кожної фази є контрольною точкою.

Недоліки:

– Модель не передбачає роботу з паралельними подіями.

– У моделі не передбачено внесення вимоги динамічних змін на різних етапах життєвого циклу.

– Тестування вимог в життєвому циклі відбувається занадто пізно, внаслідок чого неможливо внести змін, не вплинувши при цьому на графік виконання проекту.

– У модель не входять дії, спрямовані на аналіз ризиків.

– Деякий результат можна отримати тільки при досягненні низу букви V.

Jira – була використана комерційна система відслідковування помилок, призначена для організації взаємодії з користувачами, хоча в деяких випадках використовується і для управління проектами. Розроблено компанією Atlassian, є одним з двох її основних продуктів (поряд з вікі-системою Confluence). Має веб-інтерфейс.

Назва системи отримано шляхом усічення слова «Gojira» – Японського імені монстра Годзилла, що, в свою чергу, є відсиланням до назви конкуруючого продукту – Bugzilla; створювалася в якості заміни Bugzilla і багато в чому повторює її архітектуру. Система дозволяє працювати з декількома проектами. Для кожного з проектів створює і веде схеми безпеки і схеми оповіщення.

До версії 3.13.5 (включно) розрізнялися редакції Enterprise, Professional і Standard, після – Залишилася тільки редакція Enterprise (для великих організацій).

Система заснована на Java EE і працює на кількох популярних системах управління базами даних і операційних системах.

					ВКРБ-123.26.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		76

Основний елемент обліку в системі – завдання (ticket або issue). Завдання містить назву проекту, тему, тип, пріоритет, компоненти і зміст. Завдання може бути розширена додатковими полями (також і нові призначені для користувача поля можуть бути визначені), додатками (наприклад – Фотографіями, скріншотами) або коментарями. Завдання може редагуватися або просто змінювати статус, наприклад, з «відкритий» в «закритий». Які переходи між станами можливі, визначається через настроюється потік операцій. Будь-які зміни в задачі записуються в журнал.

Jira має велику кількість можливостей конфігурації: для кожної програми може бути визначений окремий тип завдання з власним workflow, набором статусів, одним або декількома видами уявлення (screens). Крім того, за допомогою так званих «схем» можна визначити для кожного індивідуального Jira-проекту власні права доступу, поведінку і видимість полів і багато іншого.

Завдяки універсальному підходу можна пристосувати Jira для багатьох непрофільних завдань, наприклад, керування вимогами, керування ризиками, аж до реалізації невеликої системи бронювання, автоматизації процесу рекрутингу.

Для інтеграції з зовнішніми системами підтримує інтерфейси SOAP, XML-RPC і REST. Поставляється із засобами інтеграції з такими системами управління версіями, як Subversion, CVS, Git, Clearcase, Team Foundation Server, Mercurial і Perforce.

Існують доповнення, що дозволяють вбудувати Jira в інтегровані середовища розробки, в тому числі Eclipse і IntelliJ IDEA. Перекладена багатьма мовами, включаючи російську, англійську, японську, німецьку, французьку, іспанську.

Для сторонніх розробників надаються кошти розробки розширень системи – плагінів. Розробники розширень можуть викладати плагіни для продажу на спеціальний розділ сайту Atlassian.

Є комерційним продуктом, який може бути ліцензований для роботи на локальному сервері або доступний в якості віддаленого додатки. Ціноутворення

					ВКРБ-123.26.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		77

залежить від максимального числа користувачів, при цьому близько \$50 за користувача для локального і \$7 на місяць за користувача для віддаленого доступу є типовими цінами.

Для академічних і комерційних клієнтів доступний повний вихідний код під ліцензією розробника.

Для проектів з відкритим вихідним кодом Atlassian надає спеціальну безкоштовну ліцензію при дотриманні наступних правил:

- проект використовує ліцензії, схвалені Open Source Initiative;
- Вихідний код проекту доступний для скачування;
- у проекту є публічно доступна веб-сайт;
- програмне забезпечення від Atlassian є на веб-сайті проекту.

4.2 Захист розробленого програмного забезпечення

Для захисту розробленого програмного забезпечення запропоновано використовувати алгоритм SHACAL-2, який шифрує дані 256-бітними блоками з використанням 512-бітного ключа. Допускається використання ключів менших розмірів (не менш 128 біт), які доповнюються бітовими нулями до 512 біт.

Шифруємий блок даних ділиться на 8 фрагментів по 32 біта (які позначені буквами A...H). Алгоритм виконує 64 раунду перетворень, у кожному з яких дані фрагменти обробляються в такий спосіб:

$$T = H_i + S_1(E_i) + Ch(E_i, F_i, G_i) + M_i + K_i,$$

$$H_{i+1} = G_i,$$

$$G_{i+1} = F_i,$$

$$F_{i+1} = E_i,$$

$$E_{i+1} = D_i + T,$$

$$D_{i+1} = C_i,$$

$$C_{i+1} = B_i,$$

$$B_{i+1} = A_i,$$

$$A_{i+1} = T + S_0(A_i) + Maj(A_i, B_i, C_i).$$

де T – тимчасова змінна.

Використовувані функції визначені в такий спосіб:

$$S_0(x) = (x \ggg 2) \oplus (x \ggg 13) \oplus (x \ggg 22),$$

$$S_1(x) = (x \ggg 6) \oplus (x \ggg 11) \oplus (x \ggg 25),$$

$$Ch(x, y, z) = (x \& y) \oplus (x' \& z),$$

$$Maj(x, y, z) = (x \& y) \oplus (x \& z) \oplus (y \& z),$$

де $\ggg$ – операція побітового циклічного зрушення вправо.

Константи, що модифікують, M_i ($i = 0 \dots 63$) наведено нижче (одна за одною від M_0 до M_{63}):

428A2F98	71374491	B5C0FBCF	E9B5DBA5
3956C25B	59F111F1	923F82A4	AB1C5ED5
D807AA98	12835B01	243185BE	550C7DC3
72BE5D74	80DEB1FE	9BDC06A7	C19BF174
E49B69C1	EFBE4786	0FC19DC6	240CA1CC
2DE92C6F	4A7484AA	5CB0A9DC	76F988DA
983E5152	A831C66D	B00327C8	BF597FC7
C6E00BF3	D5A79147	06CA6351	14292967
27B70A85	2E1B2138	4D2C6DFC	53380D13
650A7354	766A0ABB	81C2C92E	92722C85
A2BFE8A1	A81A664B	C24B8B70	C76C51A3
D192E819	D6990624	F40E3585	106AA070
19A4C116	1E376C08	2748774C	34B0BCB5
391C0CB3	4ED8AA4A	5B9CCA4F	682E6FF3
748F82EE	78A5636F	84C87814	8CC70208
90BEFFFA	A4506CEB	BEF9A3F7	C67178F2

Фрагменти розширеного ключа $K_0 \dots K_{63}$ обчислюються в процесі процедури розширення ключа, що виконується в такий спосіб:

Етап 1. 512-бітний вихідний ключ шифрування ділиться на 16 фрагментів по 32 біта $K_0 \dots K_{15} \dots$

Етап 2. Інші фрагменти розширеного ключа $K_{16} \dots K_{63}$ обчислюються з перших 16 фрагментів у такий спосіб:

$$K_i = O_1(K_{i-2}) + K_{i-7} + O_0(K_{i-15}) + K_{i-16},$$

де функції O_0 і O_1 визначені так:

$$O_0(x) = (x \ggg 7) \oplus (x \ggg 18) \oplus (x \gg 3),$$

$$O_1(x) = (x \ggg 17) \oplus (x \ggg 19) \oplus (x \gg 10),$$

де $\gg$ – операція побітового зрушення (не циклічного) вправо.

Раунди розшифрування алгоритму виконуються у зворотній послідовності:

$$T = A_{i+1} + S_0'(B_{i+1}) + Maj'(B_{i+1}, C_{i+1}, D_{i+1}) + 2,$$

$$H_i = T + S_1'(F_{i+1}) + Ch'(F_{i+1}, G_{i+1}, H_{i+1}) + M_i' + K_i' + 4,$$

$$G_i = H_{i+1},$$

$$F_i = G_{i+1},$$

$$E_i = F_{i+1},$$

$$D_i = E_{i+1} + T' + 1,$$

$$C_i = D_{i+1},$$

$$B_i = C_{i+1},$$

$$A_i = B_{i+1}.$$

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

На рисунку 5.1 зображено розроблене у бакалаврської дипломної роботі система захисту та відновлення даних в хмарних сервісах. З рисунку можна побачити що інтерфейс головного вікна розподілено на наступні функціональні розділи: – Функціональних кнопок ПЗ; Навігаційного меню яке викликається натисканням правої клавіші маніпулятора миші; Розділу обрання групи; Розділу виведення результату роботи системи; Функції представлені у графічному вигляді.

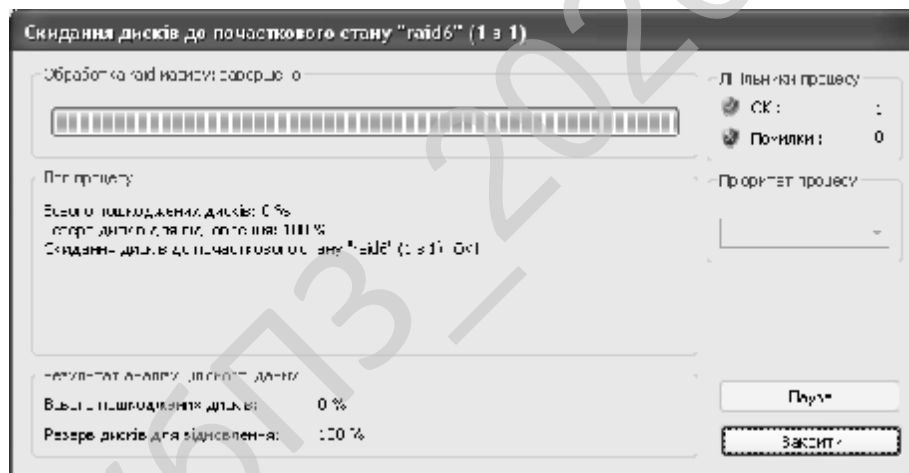


Рисунок 5.1 – Головне вікно ПЗ

Розроблена програма має дуже простий і зрозумілий інтерфейс з користувачем. Кожен, хто в достатньому обсязі володіє операційним середовищем Windows без особливих складностей освоїть і цю програму, оскільки її інтерфейс інтуїтивно зрозумілий. Якщо програма не видала ніяких помилок, і працює, то можна використовувати, інакше слід слідувати інструкціям, які пропонує програма.

На рисунку 5.2 зображено авторські дані розробленого програмного забезпечення.

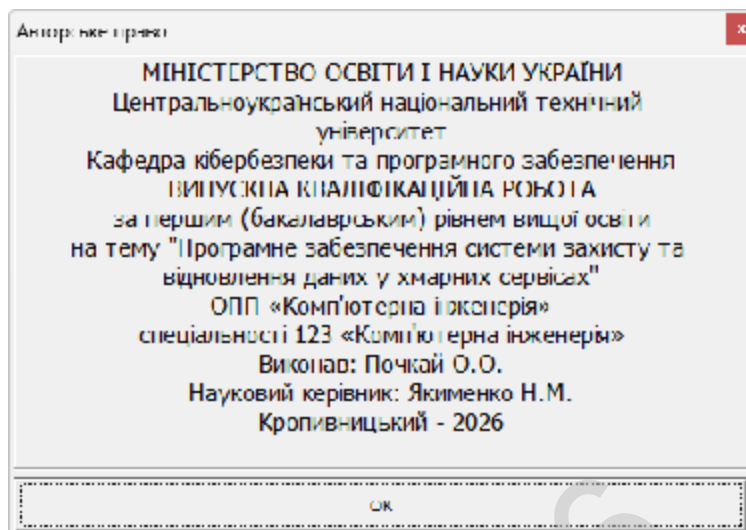


Рисунок 5.2 – Авторське право

Розглянемо процес впровадження програмного забезпечення, це процес налаштування програмного забезпечення під певні умови використання, а також навчання користувачів роботі з програмним продуктом. Впровадження програмного забезпечення це усі дії, що роблять розроблену програмну систему готовою до використання. Даний процес є частинною життєвого циклу програмного забезпечення.

Загалом процес розгортання складається з кількох взаємопов'язаних дій із можливими переходами між ними. Ця активність може відбуватися як з боку виробника так і з боку споживача. Оскільки кожна програмна система є унікальною, то усі процеси та процедури під час розгортання важко передбачити. Тому, "розгортання" можна трактувати як загальний процес відповідно до певних вимог та характеристик. Розгортання може здійснюватись програмістом і в процесі розробки програмного забезпечення.

До діяльностей пов'язаних із розгортанням програмного забезпечення відносять:

					ВКРБ-123.26.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		82

- Випуск.
- Встановлення та активація.
- Деактивація.
- Адаптація.
- Оновлення.
- Вмонтування.
- Відстежування версій.
- Видалення.
- Вилучення з обігу.

При впровадженні програмного забезпечення потрібно урахувати наступні дії:

– Виділення критичних, з точки зору загального результату, процедур в діяльності організації. Коли набір таких процедур визначений, необхідно в першу чергу використовувати ІТ рішення для автоматизації операцій усередині саме цих процедур. Таким чином, розроблене ІТ рішення автоматично стає життєво важливим і затребуваним для організації, а також буде забезпечена публічність процесу впровадження.

– Розширення нормативної бази організації шляхом включення до неї регламентів, що описують порядок виконання процедур автоматизованих процесів. В іншому випадку є небезпека виникнення неузгодженості між автоматизованими процедурами та іншими процесами організації.

– Виконання робіт з загальної стандартизації існуючої діяльності організації, коли виділяються кращі практики виконання процедур і включаються в ІТ рішення за принципом найбільшої корисності для більшості учасників. Відсоток таких процедур щодо загального обсягу автоматизації може бути невеликий, але це надає процесу побудови рішення вагу в організації за рахунок збільшення його необхідності.

Під час роботи над програмою було проведено тестування програмного забезпечення, тобто технічне дослідження, призначене для виявлення інформації про якість продукту відносно контексту, в якому воно має використовуватись.

Тестування включає як процес пошуку помилок або інших дефектів, так і випробування програмних складових з метою їх оцінки.

Проводилась оцінка:

- відповідності поставленим вимогам;
- правильна відповідь для усіх можливих вхідних даних;
- виконання функцій за прийнятний час;
- практичність;
- сумісність з ОС та стороннім ПЗ.

Оскільки число можливих тестів для програмних компонент практично нескінченне, тому стратегія тестування полягала в тому, щоб провести всі можливі тести з урахуванням наявного часу та ресурсів.

Як результат ПЗ тестувалось стандартним виконанням програми з метою виявлення помилок або інших дефектів.

Проводилось тестування чорної скриньки. Основне місце програми тестів «чорної скриньки» – інтерфейс ПЗ. Відомі: функції програми. Досліджується: робота кожної функції на всій області визначення.

Ці тести демонструють:

- Як виконуються функції програми.
- Як приймаються вихідні дані.
- Як виробляються результати.
- Як зберігається цілісність зовнішньої інформації.

При тестуванні «чорної скриньки» розглядаються системні характеристики програм, ігнорується їхня внутрішня логічна структура. Вичерпне тестування, як правило, неможливе.

Наприклад, якщо в програмі 10 вхідних величин і кожна приймає по 10 значень, то кількість тестових варіантів становитиме 10^{10} . Тестування «чорної скриньки» не реагує на багато особливостей програмних помилок.

Тестування «чорної скриньки» (функціональне тестування) дозволяє отримати комбінації вхідних даних, які забезпечують повну перевірку всіх функціональних вимог до програми.

Програмний виріб тут розглядається як «чорна скринька», чію поведінку можна визначити тільки дослідженням його входів та відповідних виходів. При такому підході бажано мати:

- Набір, утворений такими вхідними даними, які призводять до аномалій у поведінці програми (назвемо його ІТс).

- Набір, утворений такими вхідними даними, які демонструють дефекти програми (назвемо його ОТ).

Будь-який спосіб тестування «чорної скриньки» повинен:

- Виявити такі вхідні дані, які з високою ймовірністю належать набору ІТс.
- Сформулювати такі очікувані результати, які з високою ймовірністю є елементами набору ОТ.

Принцип «чорної скриньки» не альтернативний принципу «білої скриньки». Скоріше це доповнює підхід, який виявляє інший клас помилок.

Тестування «чорної скриньки» забезпечує пошук наступних категорій помилок:

- Некоректних чи відсутніх функцій;
- Помилки інтерфейсу;
- Помилки у зовнішніх структурах даних або в доступі до зовнішньої бази даних;
- Помилки характеристик (необхідна ємність пам'яті і т.д.);
- Помилки ініціалізації та завершення.

Обрано умови розповсюдження – Freeware. Це власницьке програмне забезпечення, котре можна Безоплатно використовувати протягом необмеженого терміну без обмежень у функціональності, і поширюване без сирцевих кодів.

Автори такого програмного забезпечення, як правило, хочуть «дати щось спільноті», але хочуть також контролювати його подальшу розробку. Іноді, коли програмісти вирішують припинити розробку, вони передають сирцевий код іншим програмістам, або ж спільноті як вільне програмне забезпечення.

Дуже часто плутають поняття «безплатне програмне забезпечення» та «вільне програмне забезпечення», хоча вони суттєво відрізняються.

Безплатне програмне забезпечення можна безоплатно встановлювати та використовувати (іноді з певними обмеженнями, як, наприклад, «безплатне для домашнього або некомерційного вжитку»), в той час як вільне програмне забезпечення можна продавати за будь-яку суму, але при тому, у користувача, котрий його отримує, повинні бути права на вивчення, модифікацію та поширення сирцевих кодів одержаної програми.

					ВКРБ-123.26.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		86

6 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти, призначено для системи захисту та відновлення даних у хмарних сервісах.

В межах України в недостатній мірі представлені вітчизняні розробки в цій області.

Рішення завдання полягало у вирішенні наступних задач:

- Був проведений огляд існуючих систем захисту та відновлення даних у хмарних сервісах.
- Досліджена система захисту та відновлення даних у хмарних сервісах.
- На основі отриманих результатів досліджень створена програмна реалізація системи захисту та відновлення даних у хмарних сервісах.

Розроблені під час виконання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання захисту та відновлення даних у хмарних сервісах.

Розроблене програмне забезпечення має простий, дружній та зручний інтерфейс користувача, що забезпечує легкість у освоєнні роботи програмного продукту, зручність у використанні, і не потребує особливих спеціальних знань.

При створенні програмного забезпечення було використано об'єктно-орієнтований підхід, що відповідає сучасним тенденціям у галузі розробки комерційних програмних систем.

Програма реалізована на мові високого рівня Visual C#. Дана мова програмування дозволяє найбільш ефективно обробляти дані призначені для системи захисту та відновлення даних у хмарних сервісах. Це дозволило мінімізувати строк розробки програмного забезпечення, і, як слід, зменшити витрати на його розробку. Запропоноване програмне забезпечення ділиться на загальне програмне забезпечення, що поставляється із засобами обчислювальної

					ВКРБ-123.26.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		87

техніки й спеціальне програмне забезпечення, що спеціально розроблене для даної конкретної системи й включає програми, що реалізують її функції.

Програма призначена для виконання під управлінням багатозадачної операційної системи Windows 10/11.

Даються необхідні рекомендації з установки розробленого програмного забезпечення.

Для підвищення рівня безпеки запропоновано застосовувати алгоритм SHACAL-2.

В цілому створене програмне забезпечення підтверджує правильність використаних проектних рішень та повністю відповідає вимогам технічного завдання. Створене програмне забезпечення має потенційну можливість для подальшого вдосконалення і застосування у різних галузях.

К6ПЗ-2026

					ВКРБ-123.26.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		88

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Awais Rashid, Howard Chivers, George Danezis, Emil Lupu, Andrew Martin. CyBOK The Cyber Security Body of Knowledge. The National Cyber Security Centre. 2019. 854 p.
2. Loren Kohnfelder. Designing Secure Software. No Starch Press. 2022. 332 p.
3. Samir Kumar Rakshit. Ethical Hacker's Penetration Testing Guide. BPB Online. 2022. 509 p.
4. Corey J. Ball. Hacking APIs. No Starch Press. 2022. 353 p.
5. Kevin Beaver. Hacking for Dummies. John Wiley & Sons. 2022. 419 p.
6. Mark S. Merkow. Practical Security for Agile and DevOps. CRC Press. 2022. 236 p.
7. Derek Fisher. Application Security Program Handbook. Manning Publications. 2021. 155 p.
8. Cameron Wyatt PH.D. Kali Linux Tutorial. Independently published. 2021. 60 p.
9. Kuznetsov O., Frontoni E., Kuznetsova Y., Chevardin V., Smirnov O. «Architectural foundations for adaptive security in edge computing systems». *Cybersecurity Defensive Walls in Edge Computing*, 2025. pp. 21-61.
10. Петрик В.М., Присяжнюк М.М., Аль-Файюмі Халед та ін. «Системи інформаційної зброї та технології інформаційної війни»: підручник / Петрик В.М., Присяжнюк М.М., Аль-Файюмі Халед, Жарков Я.М., Смірнов О.А, Буравченко К.О., Давидюк А.В., Кононович В.Г., Корчинский В.В., Кудирко В.М., Фесенко А.О.; за заг. ред. В.М. Петрика, М.М. Присяжнюка.— К.: Видавничий центр “Кафедра”, 2025.— 320 с.
11. Усік, П.С., Смірнова, Т.В., Буравченко, К.О., Смірнов, О.А., Улічев, О.С., Смірнов, С.А. «Дослідження технологій забезпечення кібербезпеки

					ВКРБ-123.26.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		89

банківських систем з використанням штучного інтелекту». *Кибербезпека: освіта, наука, техніка*. 2025. Том 1 № 29. С.704–716, 2025

12. Kuznetsov, O., Frontoni, E., Kuznetsova, K., Arnesano, M., Smirnov, O. «A secure biometric authentication architecture for blockchain-driven cyber-physical systems». *Security and Privacy of Cyber Physical Systems Emerging Trends Technologies and Applications*, 2025, pp. 193–224.

13. Kuznetsov, O., Atzeni, G., Arnesano, M., Randieri, C., Smirnov, O. «Secure IoT-based smart wheelchair system: From implementation to security enhancement strategy». *Security and Privacy of Cyber Physical Systems Emerging Trends Technologies and Applications*, 2025, pp. 225–257.

14. Kuznetsov, O., Smirnov, O., Kuznetsova, T., Shaikhanova, A., Svatowsky, I. «Privacy-utility trade-offs in IoT networks: A comparative analysis of differential privacy mechanisms for sensor data aggregation». *Security and Privacy of Cyber Physical Systems Emerging Trends Technologies and Applications*, 2025, pp. 589–622.

15. Lakhno, V., Malyukov, V., Smirnov, O., Bebeshko, B., Chubaievskiy, V., Zhumadilova, M., Malyukova, I., Smirnov, S. «Multifactorial Model for Targeted Attacks Counteracting Within the Framework of a Multi-Step Quality Game with Fuzzy Information». *8th International Symposium on Intelligent Informatics, ISI 2023*, 2025. vol 389. pp 377-389. Springer, Singapore.

16. Kuznetsov O., Frontoni E., Kuznetsova Y., Smirnov O., Moskovchenko I. «Trust-Based Security Architecture for Edge Computing: A Simulation Study of Dynamic Trust Evolution and Attack Detection». *CEUR Workshop Proceedings*, 2024, 3909, pp. 227–241.

17. Kuznetsov, O., Frontoni, E., Kryvinska, N., Chevardin, V., Smirnov, O. «Wireless Network Encryption Stream Ciphers, Computational Modeling, and Security Analysis». *Computational Modeling and Simulation of Advanced Wireless Communication Systems*, 2024, pp. 379–402.

18. Kuznetsov, O., Frontoni, E., Kryvinska, N., Smirnov, O., Imoize, G.L. «Computational Modeling of Enhanced Spread Spectrum Codes for Asynchronous Wireless Communication». *Computational Modeling and Simulation of Advanced Wireless Communication Systems*, 2024, pp. 403–447.

19. Ткаченко, О., Ільєнко, А., Улічев, О., Мелешко, Є., Смірнов, О. «Правові засади поширення інформаційних впливів в соціальних мережах». *Електронне фахове наукове видання «Кібербезпека: освіта, наука, техніка»*, 2024. № 2(26), С. 170–188.

20. Смірнова Т.В., Коноплицька-Слободенюк О.К., Буравченко К.О., Смірнов С.А., Кравчук О.В., Козірова Н.Л., Смірнов О.А. «Дослідження технологій забезпечення кібербезпеки хмарних сервісів IaaS, PaaS та SaaS». *Кібербезпека: освіта, наука, техніка*. 2024. №4(24), С. 6-27.

21. Вінтенко, Б., Миронець, І., Смірнов, О., Кравчук, О., Козірова, Н., Савеленко, Г., Коваленко, А. «Дослідження вимог та аналіз кібербезпеки програмного забезпечення інформаційно-керуючих систем АЕС, важливих для безпеки». *Кібербезпека: освіта, наука, техніка*. 2024. №3(23), С. 111-131.

22. Батрак О., Смірнова Т., Гнатюк В., Одарченко Р., Смірнов О. «Дослідження показників ефективності функціонування та перспектив розвитку систем IP-телефонії». *Підводні технології*, 2024, № 13, с. 28-35.

23. Kuznetsov, O., Kryvinska, N., Ilchenko, O., Smirnova, T., Ulianovska, Y. «Comparative Analysis of Cryptocurrency Trading Platforms Using the Analytic Hierarchy Process». *CEUR Workshop Proceedings*, 2023, 3628, pp. 106-115.

24. Akhalaia, G., Iavich, M., Iashvili, G., Prysiashnyy, D., Smirnova, T. «Secure Encrypted Connection on Georgian Website». *CEUR Workshop Proceedings*, 2023, 3550, pp. 313-320.

25. Al-Mudhafar Aqeel, A.M., Smirnova, T., Buravchenko, K., Smirnov, O. «The method of assessing and improving the user experience of subscribers in software-configured networks based on the use of machine learning». *Advanced Information Systems*, 2023, 7(2), pp. 49-56

26. Smirnov, O., Sydorenko, V., Aleksander, M., Zhyharevych, O., Yenchев, S. «Simulation of the cloud IoT-based monitoring system for critical infrastructures». *CEUR Workshop Proceedings*, Volume 3530, 2023, pp. 256-265.

27. Kuznetsov, O., Kandiy, S., Frontoni, E., Smirnov, O. «Trade-offs in Post-Quantum Cryptography: A Comparative Assessment of BIKE, HQC, and Classic McEliece». *CEUR Workshop Proceedings*, Volume 3504, 2023, pp. 1-11.

28. Smirnov, O., Lakhno, V., Akhmetov, B., Chubaievskyi, V., Khorolska, K., Bebesko, B. «Selection of a Rational Composition of Information 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.

29. Смірнова Т.В., Гнатюк С.О., Бердибаєв Р.Ш., Сидоренко В.М., Жигаревич О.К., «Система корелювання подій та управління інцидентами кібербезпеки на об'єктах критичної інфраструктури». *Кібербезпека: освіта, наука, техніка*, №3(19), 2023, С. 176-196.

30. Смірнов О.А. Козлов Я.О., Смірнова Т.В. «Дослідження застосування SIEM-систем для забезпечення кібербезпеки та захисту інформації». *II Міжнародна науково-практична Інтернет-конференція «Інновації та перспективні шляхи розвитку інформаційних технологій (ІПШРІТ-2023)»* м.Черкаси 6 грудня 2023 року – Черкаси: ЧДТУ.– 2023. – С.251-252.

31. Козлов Я.О., Смірнова Т.В., Смірнов О.А. «Дослідження SIEM-систем для забезпечення кібербезпеки». *VII міжнародна науково-практична конференція “Інформаційна безпека та комп’ютерні технології” до 30-ти річчя кафедри кібербезпеки та програмного забезпечення*, м. Кропивницький. 1 листопада 2023 р. – Кропивницький: ЦНТУ. – 2023. – С. 26.

32. Козлов Я.О., Козірова Н.Л., Смірнов О.А. «Дослідження структури та принципу роботи SIEM-системи». *VII міжнародна науково-практична конференція “Інформаційна безпека та комп’ютерні технології” до 30-ти річчя*

					ВКРБ-123.26.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		92

кафедри кібербезпеки та програмного забезпечення, м. Кропивницький. 1 листопада 2023 р. – Кропивницький: ЦНТУ. – 2023. – С. 59.

33. Вінтенко Б.Ю., Смірнов О.А., Коваленко О.В., Смірнов С.А., Коваленко А.С. «Дослідження нормативних документів та галузевих стандартів розробки програмного забезпечення комп'ютерних систем управління АЕС, важливих для безпеки». *Системи управління, навігації та зв'язку*, 2023, вип. 2(72), С. 170-178.

34. 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,

35. 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.

36. Смірнова Т.В., Гнатюк С.О., Сидоренко В.М., Юдін О.Ю., Сидоренко С.Ю., «Модель визначення критичності галузевих інформаційно-телекомунікаційних систем». *Проблеми інформатизації та управління*, № 2(70). 2022. С. 28-37.

37. 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.

38. 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.

					ВКРБ-123.26.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		93

39. 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.

40. Smirnov O., Kuznetsov A., Zhora V., Onikiychuk A., Pieshkova O. «Hiding Messages in Audio Files Using Direct Spread Spectrum». *11th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, IDAACS 2021*, Cracow, Poland, 22-25 September 2021. P. 414-418

41. Smirnov O., Kuznetsov A., Lokotkova I., Kuznetsova T., Florov S., Lebid O. «Using Orthogonal Signals to Hide Information in Images». *4 IEEE International Conference on Advanced Information and Communication Technologies (AICT) - 2021*, Lviv, Ukraine, September 21-25, 2021. P. 255-260.

42. Smirnov O., Kuznetsov A., Girzheva O., Kiian A., Nakisko O., Kuznetsova T. «Advanced Code-Based Electronic Digital Signature Scheme». *2020 IEEE International Conference on Problems of Infocommunications Science and Technology, PIC S and T 2020*, Kharkiv, 6 October 2020-9 October 2020, P. 358-362.

43. 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.

44. 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.

45. Smirnov O., Kuznetsov A., Kiian A., Kuznetsova T. «Non-binary constant weight coding technique». *CEUR Workshop Proceedings. Volume 2740*, 2020, Pages 102-114.

46. Smirnov O., Alimseitova Zh., Adranova A., Akhmetov B., Lakhno V., Zhilkishbayeva G. «Models and algorithms for ensuring functional stability and cybersecurity of virtual cloud resources». *Journal of theoretical and applied information technology* Vol.98. No 21, 2020, P. 3334-3346.

47. Smirnov O., Kuznetsov A., Arischenko A., Chepurko I., Onikiychuk A., Kuznetsova T. «Pseudorandom sequences for spread spectrum image steganography». *CEUR Workshop Proceedings* Volume 2654, 2020, Pages 122-131.

48. Smirnov O., Kuznetsov A., Kovalchuk D., Kuznetsova T. «New technique for data hiding in cover images using adaptively generated pseudorandom sequences». *CEUR Workshop Proceedings* Volume 2654, 2020, Pages 1-14.

49. Smirnov O., Lutsenko M., Kuznetsov A., Kiian A., Kuznetsova T., «Biometric cryptosystems: overview, state-of-the-art and perspective directions». *Lecture Notes in Networks and Systems*, vol 152. Springer, Cham. 2021, pp 66-84.

50. Smirnov O., Kuznetsov A., Onikiychuk A., Makushenko T., Anisimova O., Arischenko A. «Adaptive pseudo-random sequence generation for spread spectrum image steganography». *2020 IEEE 11th International Conference on Dependable Systems, Services and Technologies (DESSERT)*, Ukraine, Kyiv, May 14-18. 2020. P. 161-165.

51. Smirnov O., Kuznetsov A., Kiian A., Babenko V., Perevozova I., Chepurko I. «New Approach to the Implementation of Post-Quantum Digital Signature Scheme». *2020 IEEE 11th International Conference on Dependable Systems, Services and Technologies (DESSERT)*, Ukraine, Kyiv, May 14-18. 2020. P. 166-171.

52. Smirnov O., Kuznetsov A., Kiian A., Cherep A., Kanabekova M., Chepurko I. «Testing of code-based pseudorandom number generators for post-quantum application». *2020 IEEE 11th International Conference on Dependable Systems, Services and Technologies (DESSERT)*, Ukraine, Kyiv, May 14-18. 2020. P. 172-177.

53. Smirnov O., Kuznetsov A., Pushkar'ov A., Serhiienko R., Babenko V., Kuznetsova T., «Representation of Cascade Codes in the Frequency Domain». In:

Radivilova T., Ageyev D., Kryvinska N. (eds) Data-Centric Business and Applications. Lecture Notes on Data Engineering and Communications Technologies, vol 48. Springer, Cham. 2021. pp 557-587.

54. Smirnov, O., Markovets, O. Vovk, N., Turchyn, Y., «Model of informational support for social network administrators' content creation». *CEUR Workshop Proceedings* Volume 2616, 2020, Pages 125-136.

55. Smirnov, O., Shekhanin, K., Kuznetsov, A., Krasnobayev, V. «Detecting Hidden Information in FAT». *International Journal of Computer Network and Information Security (IJCNIS)*. Vol. 12, No. 3, 2020. PP.33-43.

56. Smirnov, O., Kuznetsov, A., Gorbacheva, L., Babenko, V., «Hiding data in images using a pseudo-random sequence», *CEUR Workshop Proceedings* Volume 2608, 2020, Pages 646-660.

57. Smirnov, O., Kuznetsov, A., Kolovanova, I., Kuznetsova, T., «Noise immunity of the algebraic geometric codes». *International Journal of Computing*; 2019, Volume 18, Issue 4 – Research Institute for Intelligent Computer Systems – 2019. – P. 393-407.

58. Smirnov, O., Kuznetsov, A., Reshetniak, O., Ivko, N., Katkova, T., Kuznetsova, T., «Generators of Pseudorandom Sequence with Multilevel Function of Correlation». *2019 IEEE International Scientific-Practical Conference Problems of Infocommunications, Science and Technology (PIC S&T)*, Kyiv, Ukraine, 8 – 11 October 2019 . P.517-522.

59. Smirnov, O., Ulichev, O., Meleshko, Y., Khokh, V., Goncharenko, I. «Method of Choosing Objects for Informational Influence in Social Networks during Information Campaign Based on the Analytic Hierarchy Process». *CEUR Workshop Proceedings*, Vol 2588, P. 215-227, 2019.

60. Smirnov, O., Krasnobayev, V., Yanko, A., Kuznetsova, T. «Methods of nulling numbers in the system of residual classes». *CEUR Workshop Proceedings*, Vol 2588, P. 90-106, 2019.

61. Smirnov, O., Kuznetsov, A., Kiian, A., Gorbenko, Y., Cherep, O., Bexhter L. «Code-based Pseudorandom Generator for the Post-Quantum Period», *2019 IEEE International Conference on Advanced Trends in Information Theory (IEEE ATIT 2019)*. 18.12.19-20.12.19 Kyiv Ukraine. P. 204 – 209.

62. Smirnov, O., Kuznetsov, A., Nariezhnii, O., Stelnyk, S., Kokhanovska, T., Kuznetsova T., «Side Channel Attack on a Quantum Random Number Generator», *10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, IDAACS 2019*; Metz; France; 18 - 21 September 2019. P.713-718.

63. Smirnov, O., Kuznetsov, A., Kiian, A., Pushkar'ov, A., Mialkovskyi, D., Kuznetsova, T., «Code-Based Schemes for Post-Quantum Digital Signatures», *10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, IDAACS 2019*; Metz; France; 18-21 September 2019. P. 707-712.

64. Smirnov, O., Kuznetsov, A., Stefanovych, O., Gorbenko, Y., Krasnobaev, V., Kuznetsova K. «Information Hiding Using 3D-Printing Technology», *10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, IDAACS 2019*; Metz; France; 18-21 September 2019. P.701-706.

65. Smirnov, O., Hu, Z., Vasiliu, Y., Sydorenko, V., Polishchuk, Y., «Abstract Model of Eavesdropper and Overview on Attacks in Quantum Cryptography Systems», *10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, IDAACS 2019*; Metz; France; 18-21 September 2019. P.399-405.

66. Smirnov, O., Kuznetsov, A., Kovalchuk, D., Averchev, A., Pastukhov, M., Kuznetsova, K., «Formation of Pseudorandom Sequences with Special Correlation Properties», *2019 3rd International Conference on Advanced Information and Communications Technologies, AICT -2019/ Lviv, Ukraine, 2-6 July, 2019*, P. 395-399.

67. Smirnov, O., Kuznetsov, A., Kiian, A., Babenko, B., Zhosan, H., Prokopovych-Tkachenko, D., «Soft Decoding Method for Turbo-Productive Codes», *2019 3rd International Conference on Advanced Information and Communications Technologies, AICT 2019*, Lviv, Ukraine, 2-6 July, 2019, P. 129-134.

68. Smirnov, O., Kuznetsov, A., Kiian, A., Zamula, A., Rudenko, S., Hryhorenko, V., «Variance Analysis of Networks Traffic for Intrusion Detection in Smart Grids», *2019 IEEE 6th International Conference On Energy Smart Systems (2019 IEEE ESS)*, Kyiv, Ukraine April 17-19, 2019 P. 353-358.

69. Smirnov, O., Kuznetsov, A., Kavun, S., Babenko, B., Nakisko, O., Kuznetsova, K., «Malware Correlation Monitoring in Computer Networks of Promising Smart Grids», *2019 IEEE 6th International Conference On Energy Smart Systems (2019 IEEE ESS)*, Kyiv, Ukraine April 17-19, 2019 P. 347-352.

70. Smirnov, O., Kuznetsov, A., Kovalchuk, D., Pastukhov, M., Kuznetsova, K., Prokopovych-Tkachenko, D., «Discrete Signals with Special Correlation Properties», *CEUR Workshop Proceedings Volume 2353, CEUR Workshop Proceedings 2019*, Pages 618-629.

71. Smirnov, O., Kuznetsov, A., Kiian, A., Kuznetsova, K., Ivko, T., Prokopovych-Tkachenko, D., «Soft Decoding Based on Ordered Subsets of Verification Equations of Turbo-Productive Codes», *CEUR Workshop Proceedings Volume 2353, CEUR Workshop Proceedings 2019*, Pages 873-884.

72. Smirnov A.A., Kuznetsov A.A., Danilenko D.A., Berezovsky A., «The statistical analysis of a network traffic for the intrusion detection and prevention systems», *Telecommunications and Radio Engineering*. – Volume 74, Issue 1. – Begel House Inc. – 2015. – P. 61-78.

73. 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.

74. Смірнов О.А., Коноплицька-Слободенюк О.К., Смірнов С.А., Буравченко К.О., Смірнова Т.В., Книшук А.В. «Вступ до кібербезпеки»: навчальний посібник – Кропивницький: ЦНТУ – 2022. – 968 с.

75. Теорія та практика сучасного інформаційно-психологічного протиборства: навчальний посібник / [В.М. Петрик, С.О. Гнатюк, М.М. Присяжнюк та ін.]; за заг. ред. С.О. Гнатюка, В.М. Петрика та О.А Смірнова. – Полтава, 2022. – 334 с.

76. Smirnov O. Kuznetsov A., Zaichenko Yu., Pastukhov M., Oleshko O., Kuznetsova K., «Formation of Discrete Signals with Special Correlation Properties». *International Conference on Information and Telecommunication Technologies and Radio Electronics, UkrMiCo 2019*; Odessa; Ukraine; 9-13 September 2019. P.22-28.

77. Смірнов О.А., Коноплицька-Слободенюк О.К., Смірнов С.А., Буравченко К.О., Смірнова Т.В., Поліщук Л.І. Інформаційна безпека в комп'ютерних мережах. Навчальний посібник – Кропивницький: вид. Лисенко В.Ф. 2020. – 294 с.