МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ПОЛТАВСЬКА ПОЛІТЕХНІКА ІМЕНІ ЮРІЯ КОНДРАТЮКА»

НАВЧАЛЬНО НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА РОБОТОТЕХНІКИ

КАФЕДРА КОМП'ЮТЕРНИХ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ І СИСТЕМ

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

спеціальність 122 «Комп'ютерні науки»

на тему

«Створення проєкту комплексного захисту інформації підприємства із використанням програми для шифрування даних»

Студента групи 601-ТН Профатілова Анатолія Олександровича

Керівник роботи
доктор технічних
наук, професор
Краснобаєв В. А.

Консультант
кандидат технічних
наук, доцент
Альошин С.П.

Завідувач кафедри
кандидат фізико-
математичних наук,
Двірна О.А.

Полтава – 2024

РЕФЕРАТ

Загальний обсяг роботи 74 с., 24 рисунки, 1 додаток, 3 таблиці, 41 бібліографічних найменувань.

Об’єкт дослідження: «Освітній центр LINGUALAND Exam+».

Мета роботи: розробити проєкту комплексного захисту інформації підприємства із використанням програми для шифрування даних.

Методи: аналіз стану та розробка комплексного захисту інформації на підприємстві, створення програми передачі даних з мережевою шифрацією.

Ключові слова: освітній центр, захист інформації, інформаційна безпека, безпека інформації, комплексні методи захисту інформації, методи захисту інформації.

ABSTRACT

The total volume of the work is 74 pp., 24 figures, 1 appendice, 3 tables, 41 bibliographic titles.

The object of the study: «Educational centre LINGUALAND Exam+».

The purpose of the work: development of an enterprise information protection project with network encryption.

Methods: state analysis and development of complex information protection at the enterprise, creation of a data transmission program with network encryption.

Keywords: educational centre, information protection, information security, information security, comprehensive methods of information protection, methods of information protection.

ЗМІСТ

СПИСОК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ЕОМ – електронна обчислювальна машина.

ЛОМ – локальна обчислювальна мережа.

ОМ – обчислювальна мережа.

ТЗІ – технічний захист інформації.

ПЗ – програмне забезпечення.

ПЗІ – правовий захист інформації.

ОС – операційна система.

ЗУ – закон України.

ІБ – Інформаційна безпека.

СЗІ – Системи захисту інформації.

ІС – Інформаційна система.

ДТ – Державна таємниця.

ПЗ – Програмне забезпечення.

C# (C Sharp) – об’єктно орієнтована мова програмування.

ВСТУП

У сучасному світі розвиток інформаційних систем і збільшення обсягів даних, які обробляються підприємствами, створюють нові виклики, пов’язані з забезпеченням конфіденційності, цілісності та доступності інформації. Загрози кібербезпеки, витоки даних і несанкціонований доступ до конфіденційної інформації стають серйозною проблемою для багатьох організацій, незалежно від їхнього масштабу та сфери діяльності.

Освітні установи, зокрема центри, які займаються навчанням і підготовкою до екзаменів, як-от LINGUALAND Exam+, також стикаються з необхідністю ефективного захисту інформації. В їхніх системах обробляються персональні дані студентів, фінансова та академічна інформація, що потребує надійного захисту. У цих умовах виникає потреба у створенні комплексного проєкту інформаційної безпеки, який би враховував специфіку діяльності підприємства, особливості його інформаційних потоків і сучасні вимоги до кіберзахисту.

Метою даної дипломної роботи є розробка проєкту комплексного захисту інформації для освітнього центру із впровадженням програмного забезпечення, яке реалізує шифрування даних. Завданням є не лише створення ефективного інструменту для захисту даних, але й забезпечення зручності його використання для співробітників освітнього центру.

У роботі розглянуто теоретичні аспекти захисту інформації, зокрема сучасні методи криптографії, та проаналізовано загрози і ризики, пов’язані з інформаційною безпекою. Практична частина зосереджена на розробці програми для шифрування даних із використанням алгоритму RC5, що є ефективним і гнучким інструментом для забезпечення кібербезпеки.

Результати дослідження мають важливе практичне значення, оскільки запропоноване рішення спрямоване на вирішення актуальних проблем захисту інформації в умовах зростаючого впливу цифрових технологій на всі сфери діяльності.

РОЗДІЛ 1
ТЕОРЕТИЧНІ ОСНОВИ ЗАХИСТУ ІНФОРМАЦІЇ НА ПІДПРИЄМСТВІ

1.1 Основні поняття та принципи інформаційної безпеки

Термін «інформаційна безпека» (ІБ) виник у зв'язку з активним розвитком засобів інформаційних комунікацій у нашому суспільстві. У сучасних умовах прогрес інформаційних технологій став досить звичним явищем. Кількість інформаційних систем (ІС) і програмного забезпечення (ПЗ), що допомагає персоналу підприємств успішно керувати інформаційними потоками, зростає, так як і обсяги цінної інформації. Це безперечно обумовлює підвищену актуальність питання її захисту. Варто зазначити, що у наукових колах немає єдиного підходу до визначення поняття «інформаційна безпека» [1].

Інформаційна безпека – це рівень захисту інформаційних ресурсів, таких як дані та системи, від можливих негативних впливів, які можуть виникнути як випадково, так і навмисно. Такі впливи здатні завдати шкоди самим даним, а також особливо технологіям їх обробки й передачі. Це безумовно може негативно позначитися на власниках інформації, державних інституціях, суспільстві та інших учасниках інформаційного обміну. Сучасні інформаційні системи неможливо розглядати окремо від комплексу факторів, які забезпечують їхню безпеку: це як загрози, так і різноманітні заходи захисту, бар’єри від несанкціонованого доступу та також вразливі місця в системах захисту. У більш широкому сенсі інформаційну безпеку можна розглядати як набір методів, засобів та процедур, спрямованих безпосередньо на захист інформаційних активів та забезпечення постійної стабільності й корисності як технічної інфраструктури, так і даних, що обробляються в системах. Основна мета полягає у збереженні точності, цілісності та

достовірності інформації, а також у зменшенні ризику несанкціонованого втручання. Для досягнення успіху та підвищення конкурентоспроможності підприємству необхідно створити дієву систему управління інформаційною безпекою [2].

Система управління інформаційною безпекою (СУІБ, або ISMS) є складовою загальної системи управління, яка покликана забезпечити розробку, впровадження, експлуатацію, моніторинг, оцінку, підтримку та вдосконалення заходів з інформаційної безпеки.

Для реалізації процесів СУІБ використовується модель PDCA (плануй-виконуй-перевіряй-дій; Plan-Do-Check-Act, PDCA) [3]:

Рисунок 1.1 – Модель PDCA

Розбудова СУІБ дозволяє чітко встановити, як між собою взаємодіють процеси та підсистеми ІБ, хто несе відповідальність за їх виконання, а також які фінансові та людські ресурси необхідні для їх дієвого функціонування.

Основні функції системи управління інформаційною безпекою включають [3]:

• ідентифікацію та аналіз ризиків у сфері ІБ;

• розробку та реалізацію процесів, які спрямовані безпосередньо на зменшення інформаційних ризиків;

• моніторинг даних процесів;

• певне коригування заходів для мінімізації ризиків.

Ефективне керування інформаційною безпекою ґрунтується на таких принципах:

• комплексний підхід – управління має бути всебічним, охоплювати всі елементи ІС та враховувати всі потенційні ризики, які можуть вплинути на інформацію як зсередини підприємства, так і ззовні;

• узгодженість із бізнес-цілями та стратегією компанії;

• висока здатність до управління;

• відповідність інформації, що використовується та створюється;

• ефективність – досягнення збалансованості між потенціалом, ефективністю та витратами у системі управління інформаційною безпекою;

• безперебійність управлінських процесів;

• процесний підхід – інтеграція управлінських процесів у замкнутий цикл, що охоплює планування, виконання, оцінювання, аудит та коригування, а також забезпечення безперервного зв'язку між усіма етапами.

Згідно з ISMS Framework, що є європейським аналогом системи управління інформаційною безпекою (СУІБ) і було розроблено Європейською агенцією з кібербезпеки, управління безпекою здійснюється за схемою, показаною на рисунку 1.2.

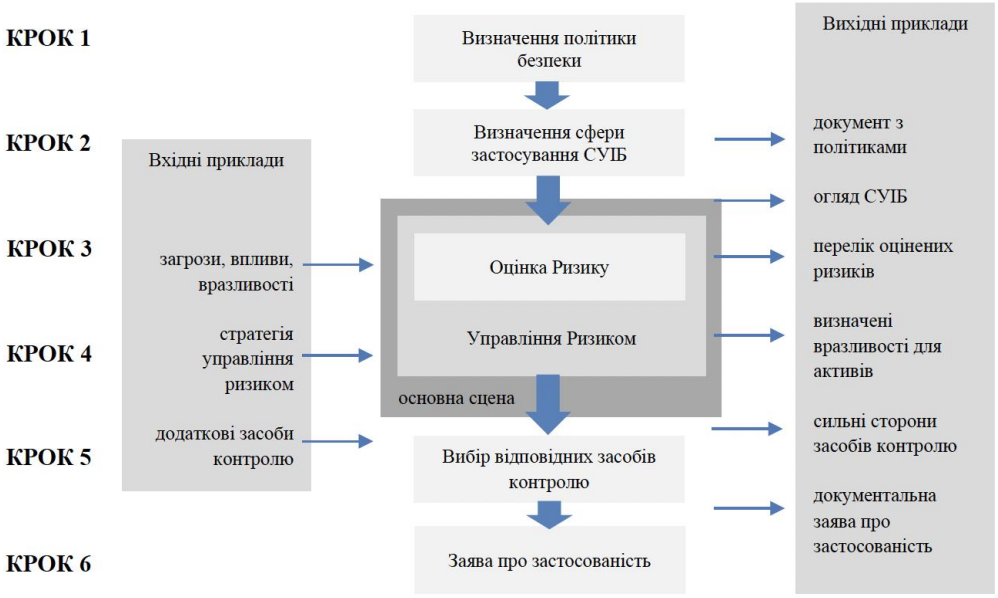

Рисунок 1.2 – СУІБ згідно з ISMS Framework

Як розмір компанії, так і специфіка її діяльності визначають вимоги до безпеки на правовому, регуляторному та операційному рівнях.

Невеликі підприємства з обмеженою інфраструктурою інформаційних систем, які часто не потребують обробки, зберігання або управління персональними чи конфіденційними даними, іноді зустрічаються з малими ризиками або менш ймовірними ризиками та також наслідками. Як правило, такі організації не підтримують окрему систему управління інформаційною безпекою (СУІБ) і зазвичай розглядають питання інформаційної безпеки в контексті загального процесу управління ризиками.

Великі організації, зокрема банки, фінансові інститути, оператори телекомунікацій, лікарні, заклади охорони здоров’я, а також державні органи, мають численні причини серйозно ставитися до питань захисту інформації. Законодавчі та нормативні вимоги, які регулюють захист конфіденційних або персональних даних, у поєднанні з загальними стандартами безпеки, змушують ці організації зосереджувати особливу увагу на ризиках інформаційної безпеки [4].

У цих умовах створення та впровадження автономного процесу управління, зокрема Системи управління інформаційною безпекою, стає єдиним дієвим варіантом.

Як видно з наведеного вище рисунка, розробка системи управління інформаційною безпекою (СУІБ) включає шість основних етапів [5]:

• формулювання політики безпеки;

• визначення меж застосування СУІБ;

• проведення оцінки ризиків (в рамках управління ризиками);

• управління ризиками;

• вибір відповідних заходів контролю;

• підготовка заяви про застосовність.

Етапи 3 та 4, що охоплюють процеси оцінки та управління ризиками, становлять основу системи управління інформаційною безпекою (СУІБ). Вони, з одного боку, «трансформують» правила і рекомендації політики безпеки та її цілі, а з іншого – трансформують ці цілі СУІБ у конкретні плани впровадження заходів контролю та механізмів, спрямованих на зменшення загроз і вразливостей. Важливо відмітити, що ці два етапи розглядаються виключно як єдине ціле, фактично представляючи собою управління ризиками [5].

Процеси та дії, які відповідають крокам 5 і 6, не пов'язані безпосередньо з інформаційними ризиками. Натомість, вони зосереджені на оперативних заходах, необхідних для технічного виконання, обслуговування та нагляду за показниками безпеки.

Відповідні засоби нагляду можуть бути отримані як з уже наявних наборів контролю або механізмів, що зазвичай входять до стандартів захисту інформації та керівних рекомендацій, так і шляхом комбінації або адаптації запропонованих засобів контролю відповідно до певних вимог організації або її функціональних особливостей.

У обох випадках крок 6 полягає в фіксації знайдених ризиків, що стосуються певної організації, разом із експлуатаційним впровадженням засобів безпеки, які організація вирішила використовувати.

Варто зазначити, що, хоча система управління інформаційною безпекою (СУІБ) є циклічним процесом загалом, в більшості з вказаних типів організацій етапи 1 та 2 повторюються в більш тривалих інтервалах, ніж етапи 3, 4, 5 та 6. Це відбувається переважно через те, що формулювання політики безпеки та визначення меж застосування СУІБ частіше стосуються управлінських та, в певній мірі, стратегічних аспектів, тоді як процес управління ризиками є більш «повсякденним» оперативним завданням [5].

Одним із основних факторів ефективності системи управління інформаційною безпекою підприємства є її побудова на основі міжнародних стандартів ISO/IEC 27001. Цей стандарт надає інструменти для створення, впровадження, підтримки, моніторингу та вдосконалення належно задокументованої системи управління інформаційною безпекою в контексті управління бізнес-ризиками [6].

СУІБ надає вибір відповідних і пропорційних методів та засобів контролю і захисту інформації, що, в свою чергу, викликає довіру з боку зацікавлених сторін. Однак важливо враховувати також й інші стандарти в сфері ІБ. На сьогодні в світовій практиці існує безліч стандартів, різних методик та інших документів, що визначають процеси управління ІБ.

Організаційно-управлінська діяльність займає ключове місце в комплексі заходів із захисту інформації, надаючи організаційне забезпечення ІБ. Це один із чотирьох основних керунків у системі заходів з інформаційної безпеки, що також охоплює розробку спеціалізованого ПЗ, застосування спеціальних апаратних засобів і також поліпшення криптографічних методів захисту інформації.

Головні завдання організаційно-управлінської процесів у сфері ІБ охоплюють такі аспекти [6]:

- забезпечення комплексного підходу до рішень, що приймаються в процесі захисту інформації;
- підтримання безперебійності та цілісності процесів ІБ;
- вирішення питань, що є основою ефективного управління інформаційною безпекою, зокрема управління ризиками та економічне моделювання;
- керування людськими ресурсами та певним чином вплив на поведінку персоналу, враховуючи також і завдання ІБ.

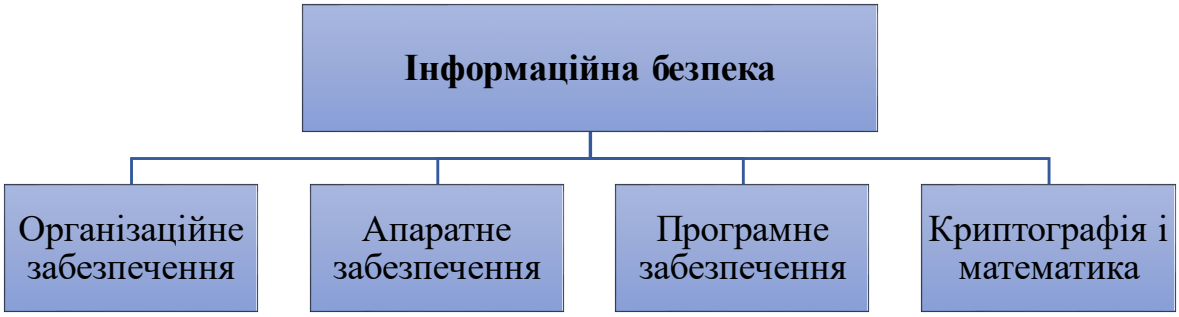

Рисунок 1.3 – Структура діяльності в сфері інформаційної безпеки

Комплексний підхід до вирішення завдань інформаційної безпеки передбачає взаємопов’язане виявлення всіх важливих інформаційних об'єктів, а також врахування існуючих і потенційних загроз. Використовуючи даний аналіз потрібно забезпечити повну та комплексну реалізацію і застосування засобів захисту інформації, які б змогли б певною мірою нейтралізувати всі значущі загрози на всіх теоретично вразливих етапах проходження інформаційних потоків. Безперечно міри щодо нейтралізації ризиків мають бути впроваджені разом з іншими механізмами, такими як страхування. Інакше кажучи, завданням менеджменту є систематичне використання всіх потрібних (вузькоспеціальних) технологій та рішень, адаптованих до кожної конкретної ситуації, щоб у системі заходів із захисту інформаційних ресурсів не виникало «вузьких місць» — вразливих ділянок, через які можуть бути

здійснені напади чи статися випадкові порушення. Складність цих завдань полягає в необхідності максимально повного аналізу всіх інформаційних ресурсів та можливих сценаріїв атак на них, а також у подальшому підборі найбільш відповідних засобів захисту [7].

Безперервність процесів ІБ передбачає надання потрібних ресурсів і організацію здійснення функцій захисту інформації на всіх етапах роботи ІС та здійснення бізнес-операцій.

Створення, покращення та актуалізація методичної бази для управління ІБ передбачає, насамперед, впровадження загальних концепцій і теорій менеджменту — наприклад, математичних моделей оцінювання ризиків чи принципів інвестиційного аналізу — у сферу ресурсів, які застосовуються для підтримання ІБ та обробки інформації.

Конкретна структура та склад завдань управління і організації в галузі ІБ, а також використовувані методи визначаються як рівнем управлінської та організаційної діяльності, так і умовами, в яких діють інформаційні системи, яким необхідний захист. Концепція курсу ґрунтується на поділі методів і завдань організації та управління у сфері ІБ на декілька ключових ступенів з подальшим визначенням відповідних організаційно-управлінських методів для кожного з цих ступенів.

Отже, з розвитком інформаційних технологій та збільшенням інтенсивності обміну інформацією, організаційна та управлінська діяльність у сфері інформаційної безпеки стає спрямованою не лише на захист конкретних інформаційних ресурсів, але й на більш широкий об'єкт – формування та удосконалення безпечної інформаційної інфраструктури.

Збільшення сфери інтересів менеджменту ІБ пояснюється потребою розподілу на кілька відносно самостійних організаційних рівнів. Кожен рівень відзначається особливими завданнями, підходами до їх вирішення та організаційними методами, що використовуються для ефективного досягнення цілей безпеки.

Рівень міжнародних професійних об'єднань, зазвичай неурядових і некомерційних, пов'язаний з інформаційними технологіями, телекомунікаціями та інформаційною безпекою, включає організації, які активно працюють над розвитком стандартів, керівних принципів та рекомендацій у цих сферах. Ці об'єднання сприяють обміну досвідом, підвищенню кваліфікації фахівців, а також визначають глобальні тенденції і найкращі практики в управлінні інформаційною безпекою [7].

Рівень великих компаній, що працюють у сфері інформаційних технологій, відіграє ключову роль у формуванні стану інформаційної безпеки як у межах своїх організацій, так і в цілому серед користувачів інформаційних систем. Ці компанії, через свої технологічні інновації та стандарти, можуть значно впливати на безпеку різних елементів інформаційної інфраструктури, задаючи вектор розвитку для більш широкого співтовариства. Їхні рішення, впровадження новітніх технологій і стратегій захисту створюють передумови для забезпечення інформаційної безпеки на ринку в цілому.

Державний рівень включає державні та міжурядові організації, які мають значний вплив на соціально-економічне життя, правову систему та технологічний розвиток країни. Вони визначають політику в галузі ІБ, розробляють та впроваджують відповідні закони і нормативні акти, а також координують міждержавні ініціативи, що стосуються захисту інформації та забезпечення національної безпеки в ІС.

Рівень окремих компаній, підприємств та організацій складається з користувачів ІС, які мають спільний інтерес у забезпеченні власної інформаційної безпеки. Ці організації самостійно організовують заходи з захисту своїх інформаційних ресурсів, впроваджуючи власні рішення для забезпечення конфіденційності, цілісності та доступності даних, а також для запобігання загрозам та виявлення вразливостей у своїх системах.

Окремо можна виділити проміжний рівень, який складається з консалтингових та впроваджувальних компаній, а також навчальних центрів, що включають спільноту фахівців, які надають консультаційні послуги,

займаються впровадженням рішень та проводять навчання в індивідуальному порядку. Ці організації діють у сфері ІБ, виконуючи роль сполучної ланки між різними рівнями організаційної структури та представляють інтереси учасників інформаційної взаємодії. Вони забезпечують обмін знаннями, технологіями та найкращими практиками у галузі захисту інформації [7].

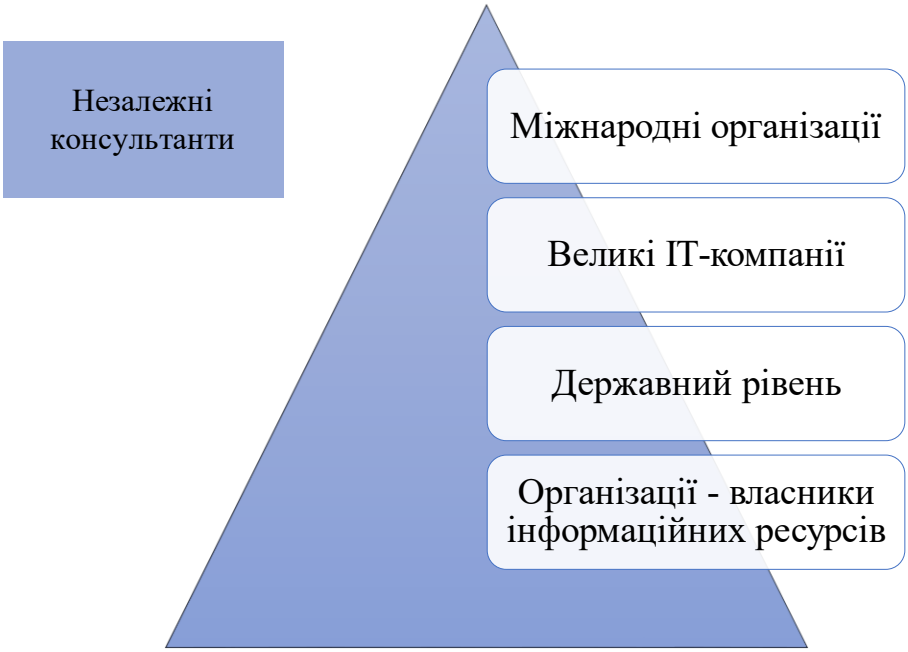

Рисунок 1.4 – Ієрархія рівнів організаційної роботи у сфері інформаційної безпеки

Такий поділ на рівні має стати базою для більш ефективного розвитку системи управління та встановлення взаємозв'язків між різними рівнями організаційної діяльності.

1.2 Методи шифрування даних у мережевих системах

SSL-сертифікати досить часто недооцінюють, хоча вони безумовно є ключовим елементом для забезпечення безпеки та конфіденційності даних в Інтернеті. Шифрування передбачає перетворення інформації в закодований формат, доступний лише для авторизованих користувачів. Це здійснюється

завдяки криптографічним ключам, які використовуються разом із спеціальними математичними алгоритмами. Розглянемо два основні методи шифрування — симетричне та асиметричне — а також п'ять найпоширеніших алгоритмів шифрування [8].

Симетричне шифрування використовує один і той самий криптографічний ключ для обох операцій – як для зашифровування, так і для розшифрування даних. Така особливість робить метод відносно простим і швидким у використанні, що є особливо корисним для великих обсягів даних. Однак необхідність передачі ключа між відправником і отримувачем створює потенційні ризики, адже перехоплення ключа може призвести до компрометації всієї зашифрованої інформації. Тому симетричне шифрування найчастіше застосовують у поєднанні з іншими методами для забезпечення більшої безпеки [9].

Симетричне шифрування відоме своєю легкістю в реалізації, адже для обох операцій — як шифрування, так і дешифрування — використовується один і той самий ключ. Завдяки цьому симетричний метод ідеально підходить для роботи з великими обсягами даних, забезпечуючи високу швидкість обробки. Основні переваги симетричних алгоритмів такі:

- вони працюють значно швидше за асиметричні алгоритми, які потребують більш складних розрахунків (асиметричні методи розглянемо далі);
- вимагають менших обчислювальних ресурсів, що полегшує їх використання на більшості пристроїв;
- практично не впливають на швидкість інтернет-з’єднання, що важливо для передачі даних у режимі реального часу.

Завдяки цим перевагам симетричне шифрування широко застосовується для захисту великих масивів даних, де критичними є швидкість і мінімальні вимоги до обладнання. Серед численних алгоритмів симетричного шифрування, що використовуються сьогодні, три виділяються особливою популярністю: AES, DES і 3DES.

Розглянемо три з найбільш відомих симетричних алгоритмів [9]:

- AES (Advanced Encryption Standard) – це сучасний стандарт шифрування, який забезпечує високий рівень захисту завдяки використанню ключів різних довжин: 128, 192 або 256 біт. AES відзначається високою швидкістю обробки даних і є широко впровадженим у різних сферах, включаючи фінансові послуги та урядові установи;
- DES (Data Encryption Standard) – це один із перших стандартів, який здобув популярність для шифрування даних. Проте, через свою коротку довжину ключа (56 біт) та вразливість до атак методом перебору, він вважається застарілим для більшості сучасних застосувань;
- 3DES (Triple DES) – це вдосконалена версія DES, яка використовує три послідовних цикли шифрування для підвищення рівня безпеки. Хоча 3DES забезпечує кращий захист у порівнянні з DES, алгоритм поступово замінюється більш ефективними методами, такими як AES, через його вищі вимоги до обчислювальних ресурсів та повільнішу швидкість.

Асиметричне шифрування, на відміну від симетричного, використовує пару ключів для процесів шифрування та дешифрування інформації. Ці ключі мають математичний зв’язок між собою, проте виконують різні функції: один з них називається «відкритим ключем», а інший – «закритим ключем» [10].

Відкритий ключ може бути вільно розподілений та використаний будь-ким для шифрування повідомлень, тоді як закритий ключ, що зберігається в таємниці, використовується для розшифровування отриманих даних. Завдяки цій структурі асиметричне шифрування, яке також відоме як «криптографія з відкритим ключем», забезпечує високий рівень безпеки, оскільки навіть якщо відкритий ключ стане відомим, зламати закритий ключ буде надзвичайно складно [10].

Асиметричне шифрування часто використовується в ситуаціях, коли потрібно забезпечити захищений обмін даними між сторонами, які не мають попередньо встановленого секретного каналу зв’язку. Це робить його

ідеальним для використання в електронній комерції, електронному підписі, а також для захисту конфіденційних даних у Інтернеті.

Прикладами популярних асиметричних алгоритмів є RSA (Rivest-Shamir-Adleman), DSA (Digital Signature Algorithm) та ECC (Elliptic Curve Cryptography), які забезпечують надійний захист інформації і є основою для багатьох сучасних протоколів безпеки.

У 1977 році три вчених із Массачусетського технологічного інституту, Рон Ривест, Аді Шамір та Леонард Адлеман, розробили алгоритм RSA, який на сьогодні є одним з найбільш поширених і застосовуваних алгоритмів асиметричного шифрування. Основний принцип його роботи базується на складності факторизації великих чисел. Для генерації пари ключів обираються два великі прості числа, які множаться між собою для створення величезного складеного числа. Ключ до дешифрування полягає в тому, щоб розкласти це число на прості множники, що є надзвичайно важким завданням навіть для потужних обчислювальних систем [11].

Значна складність факторизації великих чисел робить злом цього алгоритму практично неможливим. Наприклад, в 2010 році команда добровольців витратила більше 1500 років обчислювального часу (розподіленого між сотнями комп’ютерів), щоб зламати 768-бітний ключ RSA, що значно слабший за сучасні 2048-бітні ключі, що використовуються сьогодні.

Однією з головних переваг RSA є його масштабованість. Ключі можуть бути різної довжини, наприклад, 768, 1024, 2048, 4096 біт і більше, що дозволяє вибирати оптимальний рівень безпеки залежно від потреб. Оскільки RSA засноване на відносно простих математичних принципах, його впровадження в інфраструктуру відкритих ключів (PKI) є простим і ефективним. Висока безпека та адаптивність зробили RSA стандартом для багатьох сфер, включаючи: сертифікати SSL/TLS для безпечних з'єднань у Інтернеті, криптовалюти, де використовується для захисту транзакцій,

шифрування електронної пошти, яке гарантує конфіденційність повідомлень [11].

RSA також активно використовується для захисту даних у багатьох урядових і фінансових установах, що підтверджує його надійність і ефективність у забезпеченні безпеки.

У 1985 році математики Ніл Кобліц та Віктор Міллер запропонували концепцію використання еліптичних кривих у криптографії, що стало важливим кроком у розвитку сучасних методів захисту інформації. Ідея полягає в тому, щоб застосовувати математичні властивості еліптичних кривих для створення більш ефективних і безпечних алгоритмів шифрування.

Проте реальне застосування цієї концепції в криптографії почалося лише через майже два десятиліття. Алгоритм ECC (Elliptic Curve Cryptography) здобув популярність і почав активно використовуватися в середині 2000-х років, зокрема, в 2004-2005 роках. Завдяки своїй здатності забезпечити високий рівень безпеки при значно меншій довжині ключа, ECC став одним із найефективніших алгоритмів для шифрування даних [12].

Основною перевагою алгоритму ECC є його ефективність у використанні коротших ключів для досягнення того ж рівня безпеки, який забезпечує RSA при значно довших ключах. Це робить ECC дуже привабливим для використання в пристроях з обмеженими ресурсами, таких як мобільні телефони та інші портативні пристрої. Наприклад, для забезпечення рівня безпеки, еквівалентного 3072-бітному ключу RSA, достатньо 256-бітного ключа ECC. Алгоритм ECC використовується в багатьох сучасних додатках, таких як [12]:

- VPN та захищені канали зв'язку;
- мобільні додатки;
- блокчейн технології та криптовалюти;
- SSL/TLS сертифікати для безпечних інтернет-з'єднань.

Завдяки своїй ефективності і високому рівню безпеки, алгоритм ECC продовжує здобувати популярність серед розробників та організацій, що потребують високого рівня захисту даних.

Гібридне шифрування не є окремим типом шифрування, а поєднує в собі переваги двох основних методів: симетричного та асиметричного шифрування, створюючи потужну та ефективну систему захисту даних. Такий підхід дозволяє використовувати найкращі характеристики кожного методу, забезпечуючи високий рівень безпеки та швидкість шифрування.

Кожен з методів шифрування має свої слабкі сторони. Симетричне шифрування ідеально підходить для шифрування великих обсягів даних завдяки своїй швидкості, однак не має вбудованої механізму для перевірки особистості, що важливо для забезпечення безпеки в Інтернеті. Асиметричне шифрування, з іншого боку, дозволяє здійснити перевірку особистості та забезпечує надійний доступ до даних лише авторизованим користувачам. Однак цей процес є значно повільнішим через складність операцій з великими ключами [13].

Гібридне шифрування було розроблено для того, щоб вирішити ці проблеми, поєднуючи швидкість симетричного шифрування і переваги асиметричного шифрування. Це дозволяє не лише забезпечити захист даних, але й гарантувати перевірку особистості учасників обміну.

У сучасних технологіях, таких як SSL/TLS сертифікати, гібридне шифрування застосовується для забезпечення безпеки під час комунікації між клієнтами (наприклад, веб-браузерами) та серверами. Процес, що отримав назву «TLS handshake», починається з перевірки особистості обох сторін за допомогою відкритого і закритого ключа. Після успішної автентифікації сторін, дані шифруються за допомогою симетричного шифрування, де для кожної сесії генерується новий «сеансовий» (ефемерний) ключ. Це дозволяє забезпечити високу швидкість обміну даними, що є критично важливим для повсякденного використання Інтернету, при цьому зберігаючи високий рівень захисту інформації [13].

Таким чином, гібридне шифрування є оптимальним рішенням для забезпечення балансу між безпекою та ефективністю при обміні інформацією через Інтернет.

Коли постає питання: «Який тип шифрування є кращим?», варто зазначити, що однозначної відповіді немає.

З точки зору безпеки асиметричне шифрування безумовно має переваги, оскільки забезпечує не лише конфіденційність даних, а й аутентифікацію користувачів. Це робить його ідеальним для ситуацій, де важливо підтвердити особу відправника або отримувача. Проте продуктивність також є важливим фактором, і саме тому симетричне шифрування залишається невід'ємною частиною безпечних комунікацій, особливо коли йдеться про обробку великих обсягів даних.

Таким чином, вибір між симетричним і асиметричним шифруванням залежить від конкретних вимог безпеки, швидкості обробки даних та ресурсів, доступних для реалізації. У багатьох випадках оптимальним рішенням може бути використання комбінації обох методів, що дозволяє досягти ідеального балансу між безпекою та продуктивністю

Для кращого розуміння переваг та недоліків обох методів, підготовано таблицю, в якій узагальнені основні характеристики симетричного та асиметричного шифрування:

Таблиця 1.1 – Основні характеристики симетричного та асиметричного шифрування

Метод шифрування	Переваги	Недоліки
Симетричне	Висока швидкість шифрування	Відсутність автентифікації
	Менша потреба в обчислювальних діях	Ключ потрібно безпечно зберігати та передавати
Асиметричне	Високий рівень безпеки та автентифікації	Повільніший процес шифрування
	Можливість безпечного обміну ключами	Потребує більше обчислювальних ресурсів

.

1.3 Аналіз загроз і ризиків витоку інформації

Процес оцінки ризиків інформаційної безпеки є критичним на всіх етапах функціонування системи захисту даних і має особливе значення для власників інформаційних ресурсів. Це дозволяє визначити можливі економічні втрати, що можуть виникнути через загрози для безпеки даних. Вибір методу оцінки ризиків, як правило, залежить від ряду чинників, таких як доступність часових, фінансових і інформаційних ресурсів, рівень невизначеності в оцінці, а також можливість здобуття кількісних даних для аналізу [14].

Процес оцінки ризиків включає в себе визначення критеріїв, за якими буде прийматися рішення про рівень прийнятності цих ризиків. Важливо, щоб ці критерії були чіткими і відповідали специфічним вимогам системи безпеки, що дозволить отримати достовірні й актуальні результати аналізу. Зокрема, треба враховувати такі аспекти як конфіденційність, цілісність та доступність даних, що є основними властивостями інформаційних ресурсів.

Ідентифікація ризиків є одним з перших етапів аналізу. Вона включає виявлення загроз та вразливостей, що можуть вплинути на інформаційні ресурси. Ідентифікація власника ризику – це процес визначення осіб або підрозділів, що відповідають за управління ризиками. Власниками ризиків можуть бути фізичні або юридичні особи, які мають необхідні повноваження для управління загрозами і виконання заходів щодо їх мінімізації.

Оцінка ризиків є складною задачею в сфері управління інформаційною безпекою, оскільки в даний час не існує єдиного загальноприйнятого методу для оцінки ризиків. Оцінка зазвичай здійснюється за допомогою евристичних методів, що включають суб’єктивну складову. Це означає, що більша частина оцінки залежить від досвіду та експертної думки, що може призвести до різних результатів у різних випадках [15].

Аналіз ризиків включає оцінку можливих збитків у разі реалізації ризику, визначення ймовірності реалізації конкретних загроз і вразливостей та обчислення величини ризиків. На основі цього проводиться порівняння

отриманих значень з встановленими критеріями прийнятності, що дає змогу сформулювати стратегію подальшої роботи з ризиками. Пріоритетні напрямки, що виникають з оцінки, дозволяють визначити, які ризики слід обробляти в першу чергу для зниження їх впливу на систему.

У межах кількісного аналізу ризик (R) розглядається як комплексний показник, що визначається взаємодією кількох ключових чинників, таких як загрози, вразливості та можливі збитки. Кожен з цих елементів відіграє важливу роль у формуванні загального рівня ризику для інформаційної системи. Загроза відображає потенційний негативний вплив зовнішніх чи внутрішніх факторів на безпеку даних, вразливості – це слабкі місця системи, які можуть бути використані для реалізації загроз, а збитки визначаються як негативні наслідки для організації в разі реалізації конкретного ризику [16].

Кількісна оцінка ризику допомагає визначити, наскільки ймовірно настання збитків у результаті конкретних загроз і вразливостей, а також дозволяє оцінити масштаби можливих втрат, що виникають через ці фактори. Це дає змогу приймати обґрунтовані рішення щодо того, як мінімізувати ці ризики та які заходи слід вжити для їх нейтралізації.

У межах кількісного аналізу ризик (R) розглядається як комплексний показник, що визначається взаємодією кількох ключових чинників, таких як загрози, вразливості та можливі збитки. Кожен з цих елементів відіграє важливу роль у формуванні загального рівня ризику для інформаційної системи. Загроза відображає потенційний негативний вплив зовнішніх чи внутрішніх факторів на безпеку даних, вразливості – це слабкі місця системи, які можуть бути використані для реалізації загроз, а збитки визначаються як негативні наслідки для організації в разі реалізації конкретного ризику [16].

Кількісна оцінка ризику допомагає визначити, наскільки ймовірно настання збитків у результаті конкретних загроз і вразливостей, а також дозволяє оцінити масштаби можливих втрат, що виникають через ці фактори. Це дає змогу приймати обґрунтовані рішення щодо того, як мінімізувати ці ризики та які заходи слід вжити для їх нейтралізації.

$R = \lambda P_T P_V(z)$, де [17]:

- **λ** – це розмір збитків, які можуть бути завдані у разі порушення безпеки інформаційного активу. Це може бути фінансовий збиток, репутаційні втрати чи інші негативні наслідки для організації.
- $\mathbf{P_T}$ – ймовірність виникнення загрози, що характеризує шанси на те, що конкретна загроза реалізується в межах визначеного періоду.
- $\mathbf{P_V}$ – функція, яка описує ймовірність реалізації загрози для конкретного інформаційного активу в залежності від витрат (z) на впровадження заходів для забезпечення безпеки. Чим більші витрати на захист, тим менша ймовірність реалізації загрози.

Отже, розмір збитків безпосередньо залежить як від значимості інформації, що потребує захисту, так і від ймовірності виникнення загрози. Зниження ймовірності реалізації загрози може бути досягнуте завдяки впровадженню ефективних заходів з інформаційної безпеки, що зменшують ризики для активів.

Основною метою управління ризиками в компанії є мінімізація впливу негативних факторів на її діяльність, щоб результати роботи наближались до бажаних і відповідали визначеним цілям. Управління ризиками охоплює цілий спектр методів і технік для аналізу і мінімізації ризиків, об'єднаних у систему планування, моніторингу та коригуючих заходів. Цей комплекс включає процеси ідентифікації ризиків, їх аналіз, а також прийняття рішень, спрямованих на зниження ймовірності та ступеня впливу потенційних загроз на результати діяльності підприємства [18].

Дослідження в сфері інформаційної безпеки свідчать, що всі ризики, пов'язані з інформаційною безпекою, повинні бути узгоджені з загальними ризиками підприємства. Це привело до необхідності інтеграції системи управління інформаційними ризиками в загальну систему управління компанією. Використання кількісних методів розрахунку дозволяє обґрунтувати фінансові вкладення в інформаційну безпеку, а також оцінити їх економічну доцільність.

Однак питання оптимізації рівня інвестицій в інформаційну безпеку все ще потребує додаткового вивчення, зокрема визначення тих ділянок системи, де збільшення витрат на захист матиме найбільший вплив на зниження ризику для всієї організації. Важливо також виявити ключові компоненти, де стратегічне підвищення захисту може забезпечити найбільшу ефективність у забезпеченні безпеки, зменшуючи вразливість підприємства загалом.

Аналіз існуючих підходів до управління ризиками в складних системах показує, що ця сфера ще не достатньо формалізована та вивчена. Для зменшення невизначеності у виборі оптимальних рішень щодо управління ризиками застосовуються різні математичні методи, зокрема, методи суб'єктивної ймовірності, нечіткі множини, нейронні мережі та інші.

Зважаючи на велику різноманітність загроз, розробка методик і алгоритмів для оцінки ризику втрати або зниження рівня інформаційної безпеки є складним і важливим завданням для будь-якої інформаційної системи. Найперше, потрібно створювати гнучкі комплексні моделі систем, що враховують як програмні та апаратні ресурси, так і внутрішні та зовнішні загрози й вразливості, при цьому такі моделі повинні бути налаштовані відповідно до специфіки конкретного підприємства.

Крім того, враховуючи велику кількість факторів, що впливають на ризик, математичні моделі для оцінки інформаційної безпеки повинні забезпечувати можливість розробки ефективних числових алгоритмів для обробки даних. Це дозволить більш точно і швидко здійснювати оцінку ризиків, враховуючи всі особливості та зміни в умовах діяльності підприємства [18].

Для ефективної оцінки ризиків інформаційної безпеки важливо виявити та проаналізувати основні чинники, через які реалізуються загрози, що можуть призвести до відмов або зниження працездатності інформаційної системи. Серед різноманітних методів оцінки ризиків, важливе місце займає підхід, заснований на побудові моделі загроз і вразливостей.

Ця методика передбачає використання експертних та статистичних даних щодо загроз і вразливостей, що можуть вплинути на інформаційну систему. Оцінка ризиків у межах підприємства полягає в визначенні рівня захищеності кожного цінного ресурсу шляхом оцінки ймовірності реалізації загроз, які можуть впливати на конкретний ресурс (наприклад, ймовірність збоїв у роботі системи безпеки через низьку кваліфікацію персоналу, застаріле програмне чи апаратне забезпечення тощо). Також враховуються вразливості, через які ці загрози можуть бути реалізовані [19].

Оцінка ймовірностей реалізації загроз і вразливостей дозволяє класифікувати їх за ступенем ризику, що дозволяє прийняти обґрунтовані рішення щодо посилення захисту та пріоритетності заходів безпеки для найуразливіших компонентів системи.

Ризики інформаційної безпеки тісно пов'язані із застосуванням сучасних інформаційних технологій, які значною мірою впливають на ефективність роботи IT-компаній, особливо в контексті їх інноваційної діяльності. Тому ці ризики можна розглядати як частину загального поняття інноваційних ризиків. Якщо інноваційний ризик визначається як ймовірність негативних наслідків, що виникають через не досягнуті або неправильно визначені стратегічні цілі, то для оцінки ризиків відмови в системах інформаційної безпеки можна використовувати такий показник, як рівень витрат на відновлення працездатності системи, виражений у матеріальному чи фінансовому вимірі.

Спираючись на експертні оцінки ризиків, вразливостей і витрат по кожному з ресурсів інформаційної системи, можна створити модель, яка відображатиме актуальні для підприємства умови. Це дозволяє провести детальний аналіз функціонування системи з точки зору зниження ймовірності відмов або погіршення її працездатності. Зрештою, це сприяє максимізації ефективності роботи інформаційної системи відповідно до критерію інформаційної безпеки, дозволяючи оптимізувати витрати на її захист і підтримку [16].

Спочатку для вирішення цієї задачі необхідно виділити найбільш критичні напрямки діяльності підприємства, які безпосередньо впливають на рівень інформаційної безпеки з точки зору керівництва організації. На другому етапі для кожного з цих напрямків на основі експертних оцінок ймовірності реалізації загроз проводиться розрахунок важливості кожної окремої загрози. Окрім цього, визначаються витрати на відновлення працездатності системи після реалізації цих загроз у вартісному вимірі. Після цього розраховується загальний ризик відмови системи, який являє собою суму ризиків для кожного напрямку.

Результатом вирішення цієї задачі є оптимальний розподіл фінансових ресурсів за виділеними напрямками діяльності підприємства, що дозволяє мінімізувати ризики відмови системи відповідно до критеріїв інформаційної безпеки.

Зважаючи на велику кількість загроз інформаційній безпеці, для кількісної оцінки ризиків можна застосовувати методи оптимізації. Одним з можливих підходів є побудова математичної моделі для мінімізації ризиків, що забезпечить ефективне управління інформаційною безпекою підприємства та дозволить скоротити ймовірність збоїв і витрат на відновлення працездатності системи [17].

Нехай у технічній або соціально-економічній системі є відомі $r_i = f(\chi_i)$ залежності ризиків (r_i) відмови працездатності системи від витрат (χ_i), спрямованих на їх усунення або зменшення в і-му напрямку забезпечення інформаційної безпеки (наприклад, відмова апаратного або програмного забезпечення, зниження ефективності через недостатню кваліфікацію фахівців, таких як програмісти чи менеджери). Для мінімізації ризиків інформаційної безпеки можна використовувати показник, який визначає рівень витрат (у матеріальному або фінансовому вимірі), необхідних для відновлення працездатності системи в разі її відмови в одному або кількох напрямках [17].

На практиці кількість категорій ризиків, що становлять загрозу інформаційній безпеці підприємства, є відносно обмеженою. Теоретичні дослідження вказують, що основні загрози безпеці організації випливають із дії п’яти ключових конкурентних факторів: ризик появи товарів-замінників, внутрішньогалузева конкуренція, можливість появи нових конкурентів, ризик втрати клієнтів, а також загроза нестабільності або проблем із постачальниками. На інтенсивність цих ризиків впливають такі фактори, як умови попиту, виробничі процеси, стратегія компанії, а також наявність суміжних галузей і партнерських зв’язків. Цей підхід дозволяє оцінити конкурентну позицію компанії на ринку та вибрати довгострокову стратегію, яка забезпечить надійний захист від ризиків і одночасно підвищить конкурентні переваги [20].

Аналіз ризиків також передбачає виділення тих категорій загроз, що мають високу ймовірність реалізації. Це дозволяє оцінити та контролювати основні ризики для компанії, формуючи відповідні стратегії для їхньої мінімізації.

Малі та середні підприємства (МСП) є важливим сегментом економіки, який відзначається високою сприйнятливістю до інновацій у сферах технологій, інформаційних рішень і бізнес-стратегій. Однак, багато з них, перебуваючи у динамічному інформаційному середовищі, не приділяють належної уваги потенційним загрозам, що можуть завдати шкоди їхнім інформаційним системам. Це нехтування може призвести до суттєвих фінансових втрат [21].

Впровадження стратегій зниження ризиків, що є невід’ємною частиною діяльності компанії, допомагає не лише захистити інформаційні активи, але й підвищити конкурентоспроможність та стійкість на ринку загалом.

РОЗДІЛ 2
АНАЛІЗ СТАНУ СИСТЕМИ ІНФОРМАЦІЙНОЇ БЕЗПЕКИ

2.1 Загальні відомості про підприємство

Освітній центр LINGUALAND Exam+ — це мовний освітній центр, що спеціалізується на вивченні іноземних мов і підготовці до міжнародних екзаменів. Даний центр знаходиться у місті Полтава.

Рисунок 2.1 - Логотип освітнього центру LINGUALAND Exam+

Lingualand Exam Plus у Полтаві пропонує комплексні курси з англійської мови для дорослих і дітей. Також у центрі викладають курси, німецької та польської мов для студентів різного рівня, а також спеціальні творчі студії для дітей. Заняття доступні як в індивідуальному форматі, так і в групах до восьми осіб або міні-групах [22].

Заняття відзначаються інтерактивним підходом з акцентом на лексичні навички. Навчання проходить у комфортній атмосфері, а викладачі приділяють увагу індивідуальним потребам кожного студента, допомагаючи подолати мовний бар'єр і впевнено розмовляти англійською.

В центрі використовуються іноземні навчальні матеріали та підручники, а також сучасна тестова платформа. Кваліфіковані викладачі мають

багаторічний досвід і застосовують гнучкий підхід до навчання. Якщо студент пропускає заняття, він може відпрацювати його безкоштовно. Крім курсів іноземних мов, центр надає підтримку у навчанні за шкільною програмою, включаючи такі предмети, як математика та українська мова [22].

Центр має позитивні відгуки від учнів, які відзначають професійність викладачів, зручність навчальних матеріалів та високу якість навчання.

Організаційна структура установи — це система, що визначає ієрархію, ролі та обов'язки, а також взаємодію між підрозділами й окремими працівниками. Вона забезпечує ефективне управління, розподіл функцій, а також контроль за досягненням цілей установи [23].

У центрі існують чітко визначені посади, а також завдання і обов'язки. Однак кожного року можуть створюватися нові ідеї та ініціативи, а також з’являтися додаткові підрозділи через зміни у напрямках діяльності закладу. Тому важливо, щоб розподіл відповідальності та обов'язків був чітким і логічним.

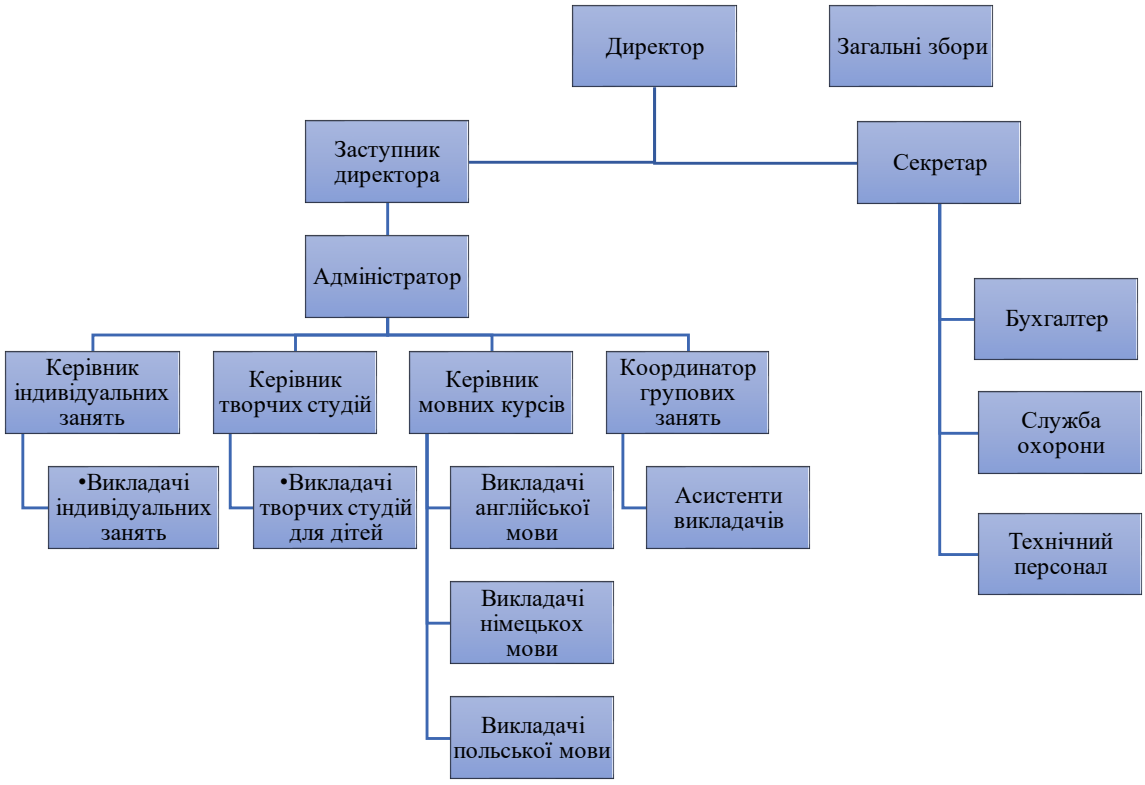

Рисунок 2.2 – Організаційна структура освітнього центру LINGUALAND Exam+

Директор відповідає за загальне управління та стратегічний розвиток центру, контроль якості послуг, прийняття ключових рішень, а також підтримку зв’язку з клієнтами та партнерами. Директор скликає загальні збори (конференції) для вирішення важливих питань щодо роботи центру.

Заступник директора з навчальної роботи забезпечує якість і стандарти навчального процесу, займається плануванням навчальної програми, підбором кадрів, а також організацією підвищення кваліфікації для викладачів.

Секретар веде діловодство, відповідає на телефонні дзвінки, обробляє електронну пошту, допомагає адміністрації у підготовці документів.

Адміністратор забезпечує реєстрацію студентів, підтримує комунікацію з клієнтами (надання інформації про курси, розклад), а також координує розклад і резервування місць у групах.

Керівник мовних курсів координує роботу викладачів іноземних мов та асистентів. Відповідає за програму навчання для кожної мови та контроль якості викладання.

Викладачі англійської, німецької, польської та китайської мов ведуть заняття з відповідної мови для дорослих і дітей на різних рівнях (від початкового до просунутого).

Асистенти викладачів допомагають викладачам під час занять, особливо у великих групах або на заняттях з дітьми.

Керівник творчих студій організовує та координує роботу творчих студій для дітей. Забезпечує розвиток мовних навичок дітей через творчі заняття.

Викладачі творчих студій для дітей проводять інтерактивні заняття з розвитку мовних навичок у дітей через творчість (наприклад, театр, малювання, музика).

Керівник відділу індивідуальних занять відповідає за організацію індивідуальних занять, підбір викладачів для індивідуальних учнів, а також за забезпечення гнучкості розкладу для таких занять.

Викладачі індивідуальних занять проводять індивідуальні уроки для студентів, забезпечуючи індивідуальний підхід і адаптацію програми до потреб конкретного учня.

Координатор групових занять організовує та координує розклад групових занять (до 8 осіб) та міні-груп. Відповідає за комплектування груп за рівнем підготовки та розробку відповідного розкладу.

Бухгалтер відповідає за фінансові питання, включаючи оплату праці співробітників, облік доходів і витрат, а також ведення звітності та бухгалтерських документів.

Служба охорони відповідає за дотримання внутрішніх правил безпеки, забезпечує охорону приміщень після завершення робочого дня.

Технічний персонал відповідає за підтримку приміщень та обладнання у належному стані, забезпечує комфортні умови для занять, а також вирішує технічні питання, які можуть виникнути під час занять.

Ця структура чітко розподіляє обов’язки між співробітниками, забезпечуючи ефективність роботи та високий рівень обслуговування клієнтів мовного центру «Lingualand Exam +».

2.2 Характеристика даних в мережі підприємства

Характеристика даних в мережі підприємства — це опис властивостей, видів і способів організації інформації, яка циркулює в інформаційній системі підприємства. Основна мета характеристики даних — забезпечення їхньої ефективної обробки, зберігання, передачі та захисту в межах корпоративної мережі [24].

Розглянемо детально характеристику даних в мережі підприємства, а саме мовного центру «Lingualand Exam +».

Таблиця 2.1 – Характеристика даних в мережі підприємства

Категорія даних	**Опис**	**Рівень конфіденційності**	**Доступ**	**Вимоги до безпеки**
Загальна інформація	Рекламні матеріали, розклад занять, контактна інформація, програми курсів	Нетаємна (Н).	Відкрита для всіх працівників, студентів, партнерів та відвідувачів.	Мінімальні
Особиста інформація	Дані учнів (ім'я, дата народження, контактна інформація), дані співробітників (контракт, адреса, банківські реквізити).	Конфіденційна (К)	Лише для адміністративного персоналу, психолога, медичної служби та директора	Захист від несанкціонованого доступу (шифрування, обмеження доступу).
Фінансова інформація	Рахунки за навчання, бюджети, платежі, фінансові звіти.	Конфіденційна (К)	Бухгалтерія, директор, заступники директора	Жорсткий контроль доступу, резервне копіювання.
Академічна інформація	Журнали успішності, тести, навчальні плани, методичні матеріали	Частково конфіденційна (К/Н).	Педагогічний колектив, директор, студенти (обмежено).	Захист від несанкціонованого редагування.
Юридична інформація	Ліцензії, угоди з клієнтами, договори з персоналом, нормативно-правова документація	Конфіденційна (К).	Директор, юридичний відділ (за наявності), бухгалтерія	Довготривале збереження, захист від модифікацій.
Технічна інформація	Дані про IT-інфраструктуру, мережеві настройки, лог-файли, інформація про обладнання	Для службового користування (ДСК).	IT-відділ, технічні фахівці	Резервування даних, захист від кібератак.

2.3 Розмежування прав доступу

Розмежування прав доступу — це система обмеження доступу до інформаційних ресурсів, систем, приміщень чи обладнання для різних категорій користувачів або співробітників. Мета розмежування прав доступу — забезпечити конфіденційність, цілісність та безпеку інформації, а також захистити ресурси від несанкціонованого доступу або використання.

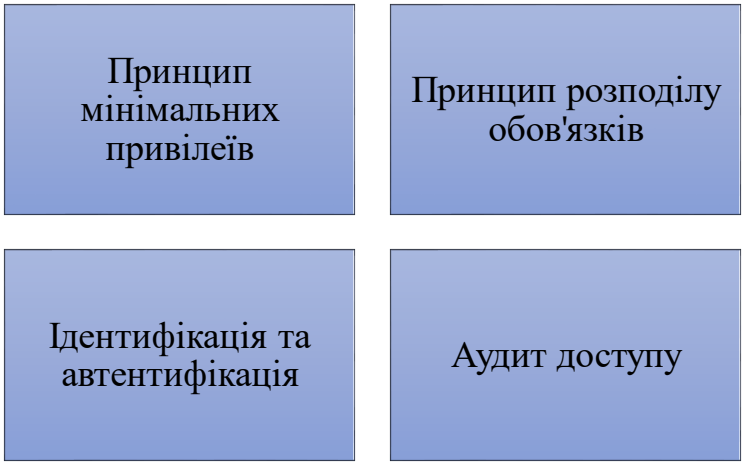

Рисунок 2.3 – Основні принципи розмежування прав доступу

Принцип мінімальних привілеїв означає, що кожному користувачеві або співробітнику надається тільки той обсяг доступу, який необхідний для виконання його конкретних обов’язків. Наприклад, бухгалтеру потрібен доступ до фінансових документів, але немає потреби мати доступ до конфіденційної інформації щодо маркетингових стратегій. Таким чином, зменшується ризик випадкового або навмисного втручання у функції, які не стосуються основних обов'язків цього користувача. Принцип мінімальних привілеїв також допомагає обмежити обсяг інформації, до якої потенційно можна отримати доступ у разі порушення безпеки облікового запису [26].

Принцип розподілу обов’язків спрямований на зниження ризиків шахрайства, зловживань і помилок за допомогою розподілу процесів між кількома людьми. Наприклад, у бухгалтерії одна людина може бути відповідальною за складання фінансових звітів, інша — за їх перевірку, а третя

— за остаточне затвердження. Таке розмежування забезпечує більшу прозорість і контроль, оскільки жоден співробітник не має повного контролю над процесом. Це допомагає виявляти помилки або несанкціоновані дії на ранньому етапі та запобігає зловживанню службовим становищем.

Ідентифікація та автентифікація є важливими етапами, що передують наданню доступу до систем або ресурсів. Ідентифікація — це процес розпізнавання користувача, який зазвичай включає введення імені користувача. Після цього здійснюється автентифікація — підтвердження особи користувача за допомогою пароля, біометричних даних (відбитків пальців, сканування обличчя тощо) або додаткових заходів, таких як двофакторна автентифікація (2FA). Автентифікація дозволяє переконатися, що особа, яка запитує доступ, дійсно є тим, за кого себе видає. Це знижує ризик несанкціонованого доступу до системи або даних [27].

Аудит доступу означає регулярний моніторинг та запис усіх дій, які виконуються користувачами в системі. Це можуть бути записи про входи і виходи користувачів, зміни в документах, завантаження файлів, перегляд конфіденційної інформації тощо. Аудит дозволяє виявляти незвичайну або підозрілу активність, наприклад, якщо користувач намагається отримати доступ до інформації, яка йому зазвичай недоступна. Такий облік дій допомагає не тільки запобігати інцидентам, але й розслідувати можливі випадки порушення безпеки. Аудит доступу може бути корисним під час внутрішніх розслідувань та для підвищення прозорості у використанні ресурсів компанії [28].

Приклади застосування розмежування прав доступу:

- інформаційні системи: користувачі мають різні рівні доступу. Наприклад, менеджери можуть мати повний доступ до внутрішніх документів, тоді як звичайні працівники мають доступ тільки до документів, необхідних для їхніх щоденних завдань;

- приміщення: в офісі можуть бути зони з обмеженим доступом, такі як серверні кімнати або архіви, до яких мають доступ тільки певні співробітники;
- IT-системи: адміністратори можуть мати доступ до серверів і систем, а звичайні користувачі – лише до своїх робочих станцій, що зменшує ризик порушення безпеки на рівні системи.

Для розподілу повноважень суб'єктів по відношенню до об'єктів використовується матрична модель доступу.

Таблиця 2.2 – Матрична модель доступу

Інформація / Відділи	Загальна інформація	Особиста інформація	Фінансова інформація	Навчальна інформація	Правова інформація	Технічна інформація
Директор	+	+	+	+	+	+
Заступник директора з навчальної роботи	+	+	+	+	-	+
Секретар	+	-	+	+	+	+
Адміністратор	+	+	+	+	-	-
Керівник індивідуальних занять	+	+	-	+	-	-
Керівник творчих студій	+	+	-	+	-	-
Керівник мовних курсів	+	+	-	+	-	-
Викладачі	+	-	-	+	-	-
Бухгалтер	+	-	+	-	+	-
Технічний персонал	+	-	-	-	-	+
IT служба	+	+	-	-	-	+
Охорона	+	-	-	-	+	+
Рівень секретності інформації	Відкрита	Відкрита	Таємна	Таємна	Таємна	Частковий доступ

2.4 Інженерно-технічний та серверний захист

Інженерно-технічний захист (ІТЗ) охоплює комплекс спеціалізованих структур, технічних засобів та заходів їх застосування, які спрямовані на недопущення розголошення, витоку, несанкціонованого доступу або іншого незаконного втручання в інформаційні ресурси.

Засоби ІТЗ класифікують за такими критеріями [29]:

- за об’єктами впливу: для захисту персоналу, матеріальних ресурсів, фінансів чи інформації;
- за характером заходів: відповідно до мети та способу дії;
- за методами реалізації: з урахуванням технологічних підходів;
- за масштабом впливу: від локального до глобального рівня;
- за типами технічних засобів: залежно від категорії використовуваного обладнання;
- за типами загроз: протидія різним класам засобів, якими можуть користуватися порушники.

Залежно від функцій, технічні засоби ІТЗ поділяються на окремі категорії.

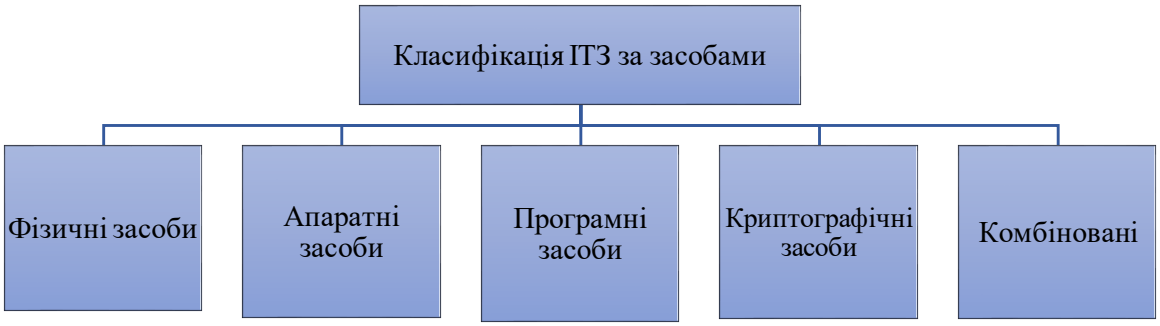

Рисунок 2.4 – Класифікація ІТЗ за засобами

Фізичні засоби захисту охоплюють широкий спектр пристроїв, інженерних споруд та організаційних заходів, які перешкоджають фізичному проникненню чи доступу зловмисників до об’єктів захисту або матеріальних носіїв інформації. Вони забезпечують безпеку персоналу, матеріальних

ресурсів, фінансів та інформаційних активів від протиправних дій чи небажаного впливу (рис. 2.4). До таких засобів належать системи відеоспостереження, охоронні бар’єри, сигналізація, турнікети, системи контролю доступу тощо.

Апаратні засоби захисту включають механічні, електричні, електронні пристрої та інші технічні розробки, спрямовані на забезпечення інформаційної безпеки. Їх головне завдання полягає у створенні стійкого бар'єру проти розголошення, витоку чи несанкціонованого доступу до інформації. Такі засоби також протидіють використанню технічних засобів розвідки. Прикладами є шифрувальні пристрої, генератори перешкод, апаратні брандмауери та пристрої для виявлення прослуховувальних пристроїв [30].

Програмні засоби захисту включають спеціалізовані програми та комплекси, які інтегруються в інформаційні системи для забезпечення безпеки даних. Ці засоби реалізують функції збереження доступності, цілісності, конфіденційності та контролю дій користувачів. До таких програм належать антивірусне програмне забезпечення, міжмережеві екрани, системи шифрування файлів, моніторингові системи та програми для управління доступом до даних.

Криптографічні засоби захисту базуються на математичних методах та алгоритмах, що використовуються для шифрування інформації, яка передається мережами, зберігається або обробляється на комп’ютерах. Основна їхня мета – унеможливити несанкціонований доступ до даних через їхнє кодування. Серед них – шифрувальні протоколи, генератори ключів, цифрові підписи, а також системи керування ключами [31].

Варто зазначити, що поділ засобів захисту на вищезазначені категорії є умовним, оскільки на практиці вони часто комбінуються. Наприклад, апаратні та програмні засоби інтегруються в програмно-апаратні комплекси, які використовують криптографічні алгоритми для забезпечення багаторівневої безпеки інформації. Таке поєднання дозволяє значно підвищити ефективність захисту від сучасних загроз.

Рисунок 2.5 – Категорії технічних засобів ІТЗ

Засоби самозахисту – елементи захисту, притаманні самому програмному забезпеченню або супроводжують його продаж і перешкоджають незаконним діям.

Засоби захисту в складі обчислювальної системи – засоби захисту апаратури, дисків і штатних пристроїв. При використанні таких засобів операційне середовище, на відміну від штатного режиму, постійно змінюється, оскільки виконання таких програм залежить від певних дій, спеціальних запобіжних заходів і умов, що гарантують захист [32].

Засоби захисту із запитом інформації – вимагають для своєї роботи введення додаткової інформації з метою ідентифікації повноважень користувачів.

Засоби активного захисту – ініціюються при виникненні особливих обставин:

- введенні неправильного пароля;
- вказанні неправильної дати або часу при запуску;
- спроби доступу до інформації без дозволу, тощо.

Засоби пасивного захисту – спрямовані на попередження, контроль, пошук доказів з метою створення обстановки неминучого розкриття злочину.

Інформація є важливим ресурсом, і її втрата, особливо конфіденційного характеру, може призвести до серйозних моральних або матеріальних збитків.

Ситуації, які створюють ризики для неправомірного отримання конфіденційної інформації, часто пов’язані з розголошенням, витоком даних або несанкціонованим доступом до джерел цієї інформації.

У сучасних умовах забезпечення безпеки інформаційних ресурсів можливе лише за допомогою комплексного підходу до їхнього захисту.

Комплексна система захисту інформації повинна відповідати низці принципів, зокрема:

- діяти безперервно;
- бути ретельно спланованою та спрямованою на досягнення конкретних цілей;
- функціонувати надійно як у звичайних умовах, так і під час надзвичайних ситуацій;
- бути активною у виявленні та протидії загрозам.

Ефективна система захисту базується на багатокомпонентному забезпеченні, яке дозволяє їй працювати як у повсякденній діяльності, так і у критичних обставинах.

Досягнення комплексної безпеки інформаційних ресурсів передбачає одночасне використання кількох рівнів захисту:

Нормативно-правова база визначає основні правила роботи з інформацією, права та обов’язки співробітників, а також їхню відповідальність за порушення встановлених норм. Вона закладає фундамент для ефективної організації захисту інформації.

Організаційні заходи є ключовою складовою системи, яка включає створення служби безпеки, розробку процедур захисту даних, навчання співробітників та моніторинг виконання правил безпеки. Організація цих заходів забезпечує скоординовану роботу системи захисту.

Інженерно-технічний захист охоплює використання апаратних і програмних рішень, криптографічних методів та інших інструментів для збереження конфіденційності, цілісності та доступності інформації.

Інженерно-технічний захист, як один із ключових елементів, використовує інноваційні рішення для попередження витоку інформації та її компрометації. Це можуть бути апаратно-програмні комплекси, системи моніторингу активності в мережі, засоби шифрування, а також пристрої для запобігання технічному прослуховуванню [32].

Таким чином, лише інтеграція правових, організаційних та технічних заходів може гарантувати повноцінний захист інформаційних ресурсів, мінімізуючи ризики, пов’язані з внутрішніми та зовнішніми загрозами.

2.5 Антивірусний захист системи

Існують програми, які спочатку створювалися для знищення даних на чужих комп'ютерах, крадіжки інформації, незаконного використання ресурсів тощо, або придбали ці властивості з певних причин. Такі програми мають шкідливу функціональність і називаються шкідливими програмами. Залежно від методів поширення і шкоди, що завдається, шкідливе ПЗ можна класифікувати на чотири основні категорії: віруси, черв'яки, трояни та інші види програм.

Комп'ютерний вірус є програмою, здатною копіювати саму себе і впроваджуватися у файли, системні області комп'ютера або мережі. Ці копії зберігають здатність до подальшого розповсюдження. Основні цілі вірусу — поширення інші ресурси системи та виконання особливих дій, найчастіше шкідливих, при певних подіях. Залежно від способу активації віруси поділяються на завантажувальні, файлові, макровіруси та скрипт-віруси. За методами маскування від антивірусів виділяють зашифровані віруси, метаморфні віруси (які змінюють частини коду, щоб уникнути антивірусне ПЗ) і поліморфні віруси, які комбінують різні методи приховування [33].

Існують програми, які спочатку створювалися для знищення даних на чужих комп'ютерах, крадіжки інформації, незаконного використання ресурсів тощо, або придбали ці властивості з певних причин. Такі програми мають

шкідливу функціональність і називаються шкідливими програмами. Залежно від методів поширення і шкоди, що завдається, шкідливе ПЗ можна класифікувати на чотири основні категорії: віруси, черв'яки, трояни та інші види програм.

Хробак (або мережевий хробак) — це шкідлива програма, що розповсюджується по мережах, здатна самостійно долати захист мережевих систем та створювати свої копії для подальшого розповсюдження. Ці копії можуть відрізнятися від вихідного хробака. Черв'яки класифікуються за методами проникнення в систему, включаючи мережеві, поштові, IM-хробаки, IRC-хробаки та P2P-хробаки. Залежно від того, чи вимагає програма активності користувача для активації, черв'яка поділяються на два типи: активні та пасивні [34].

Троян (троянський кінь) — це шкідливе програмне забезпечення, яке виконує несанкціоновані дії на комп'ютері, такі як крадіжка, знищення або зміна конфіденційних даних, порушення роботи системи або використання ресурсів машини з метою отримання вигоди. Трояни можуть бути розділені за типами шкідливих навантажень: клавіатурні шпигуни, програми для крадіжки паролів, утиліти для прихованого віддаленого доступу, анонімні поштові сервери, програми для дозвону, а також логічні бомби та організатори DDoS-атак [34].

Серед інших видів шкідливих програм можна назвати [35]:

- Riskware — утиліти, які зазвичай використовуються адміністраторами для віддаленого управління, клієнти IRC, програми для завантаження файлів з Інтернету, утиліти відновлення паролів та інші;
- Adware — програми, що надаються безкоштовно, але в обмін показують користувачеві рекламу, часто у вигляді графічних банерів;
- Pornware – утиліти, пов'язані з показом порнографічної інформації;
- утиліти для злому - програми, що приховують код заражених файлів від антивірусної перевірки, а також конструктори вірусів;

- злі жарти — програми, які створюють помилкові повідомлення або загрози для обману користувача.

Антивірусні програми призначені для захисту від шкідливих програм. Основними методами антивірусного захисту є сигнатурні методи (засновані на розпізнаванні відомих загроз за їх цифровими підписами) і евристичні методи (які аналізують поведінку програм для виявлення нових або невідомих загроз).

Сигнатурні методи — це точні методи виявлення вірусів, що базуються на порівнянні файлів з відомими зразками вірусів. Вірусна сигнатура – це набір характеристик, який дозволяє точно ідентифікувати наявність вірусу у файлі, включаючи випадки, коли файл є цілком вірусом. Усі сигнатури відомих вірусів становлять антивірусну основу. Експерти в галузі вірусології витягують код вірусу та формулюють його характерні ознаки для подальшого пошуку. Однак цей метод не є ефективним для захисту від нових вірусів, оскільки сигнатури можна створити тільки після того, як вірус був проаналізований. Саме тому нові віруси можуть поширюватись до того, як для них будуть створені сигнатури [36].

Евристичні методи — це методи, засновані на припущенні, що нові віруси часто схожі з відомими. Вони дозволяють виявити віруси, які є точними копіями відомих, але мають схожі риси. Цей метод допомагає виявити нові віруси до того, як для них створять сигнатури. Він також включає аналіз шкідливих дій, таких як видалення файлів, зміна реєстру або перехоплення даних з клавіатури. Позитивна якість евристичного методу полягає в тому, що він може виявити нові шкідливі програми, навіть якщо вони не схожі на старі віруси [36].

Недоліками обох методів є:

- можливість хибних спрацьовувань;
- відсутність можливості відновлення заражених файлів;
- низька ефективність проти нових та інноваційних вірусів.

Для підвищення ефективності антивірусних програм використовують комбіновані методи виявлення. Одним із ключових компонентів є модуль

оновлення, який дозволяє отримувати актуальні сигнатури вірусів. Коли експерти аналізують нові віруси та створюють їхні сигнатури, ці оновлення завантажуються через інтернет. Модуль оновлення перевіряє наявність нових файлів та оновлює антивірусну програму.

Другим важливим допоміжним модулем антивірусної програми є модуль планування. Цей модуль відповідає за регулярне виконання різних дій, таких як перевірка системи на віруси та оновлення бази даних антивірусу. Щоб забезпечити максимальний захист, рекомендується налаштовувати частоту оновлень антивірусної бази, наприклад, раз на одну чи три години, залежно від можливостей інтернет-з'єднання. Це особливо важливо, оскільки шкідливі програми регулярно оновлюються, і антивірусні компанії часто випускають нові сигнатури, щоб протистояти новим загрозам [33].

Якщо користувач проводить багато часу в Інтернеті, його комп'ютер схильний до більшого ризику зараження, тому оновлення бази антивіруса має відбуватися якнайчастіше. Також необхідно регулярно проводити повну перевірку системи, оскільки нові віруси можуть проникнути до системи до появи відповідних сигнатур. Рекомендується налаштувати перевірку системи хоча б раз на тиждень. Основне завдання модуля планування - це надати користувачеві можливість вибрати найбільш підходящий розклад для різних дій, включаючи перевірку та оновлення.

Крім того, зі збільшенням числа модулів в антивірусній програмі зростає необхідність у модулі керування та налаштування. Це інтерфейс, який дозволяє користувачеві легко налаштовувати та керувати функціями антивірусу. Такий модуль має бути зручним та інтуїтивно зрозумілим, з докладною довідковою системою, яка пояснює всі параметри та дозволяє захистити налаштування від змін. У домашніх антивірусних програмах цей модуль є інтерфейсом з базовими функціями. Однак для захисту великих мереж потрібен складніший модуль управління, який допомагає адміністраторам централізовано керувати антивірусами на безлічі пристроїв, спрощуючи їх налаштування та моніторинг безпеки.

У багатьох антивірусних програмах передбачені спеціальні технології, які допомагають захистити дані від втрати у разі неправильного впливу антивірусу. Наприклад, може виникнути ситуація, коли евристичний аналізатор виявляє підозрілий файл і вирішує видалити його. Однак, оскільки евристичний метод не дає стовідсоткової впевненості в тому, що файл справді заражений, існує можливість, що антивірус видалити незаражений файл. Також можливо, що антивірус знаходить заражений документ і намагається його вилікувати, але відновлення може завершитися помилкою, внаслідок чого разом з вірусом втрачається важлива інформація. Щоб запобігти таким втратам, багато антивірусів пропонують створити резервні копії файлів перед їх лікуванням або видаленням. Це дозволяє у разі помилки відновити втрачені дані, звернувшись до резервної копії [35].

Брандмауер - це технологія, що є системою або комбінацією систем, яка дозволяє розділяти мережу на кілька частин і створювати правила для передачі даних між ними. Часто цей кордон встановлюється між корпоративною локальною мережею та Інтернетом, але може також перебувати всередині корпоративної мережі. Брандмауер управляє всім трафіком, що проходить через нього, і для кожного пакета даних визначає, пропускати його або блокувати, спираючись на заздалегідь задані правила.

Існує три основні типи брандмауерів [37]:

- пакетні фільтри (packet filters), які вирішують, пропускати пакет чи ні, аналізуючи IP-адреси, TCP-прапори та номери портів, зазначені у заголовку пакета. Вони працюють на мережному та транспортному рівнях, але іноді враховують інформацію прикладного рівня, оскільки в TCP/IP стандартні послуги прив'язані до конкретних номерів портів;
- шлюзи додатків (application gateways) — це сервери, які контролюють передачу даних на прикладному рівні, фільтруючи доступ до програм на основі заданих параметрів;
- шлюзи з'єднань (circuit gateways), які контролюють мережеві з'єднання, забезпечуючи їхню безпеку на рівні взаємодії вузлів.

Ці типи можуть використовуватися окремо або комбінуватись для підвищення рівня безпеки мережі.

Для опису правил проходження пакетів складаються таблиці типу:

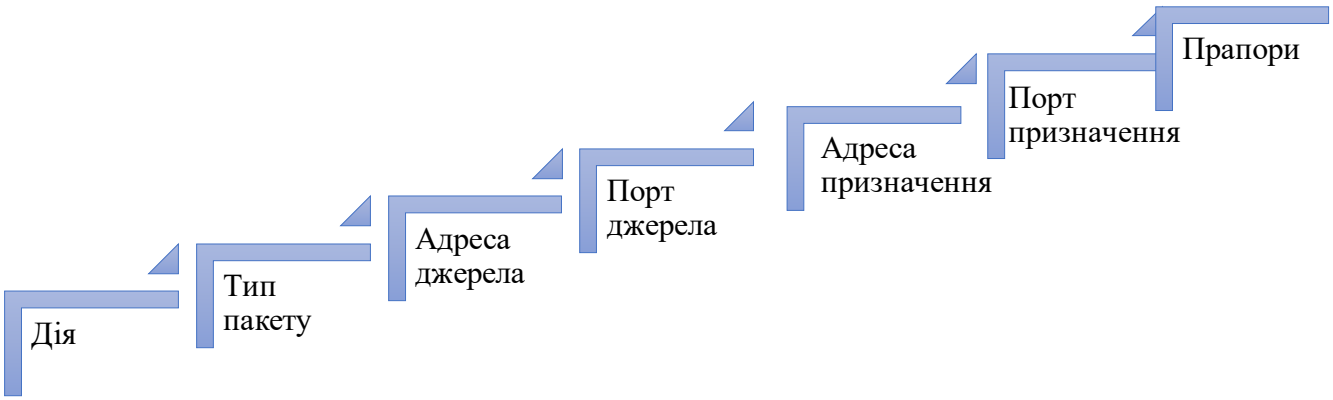

Рисунок 2.6 – Правило проходження пакетів

Поле «дія» в налаштуваннях брандмауера може приймати значення «пропустити» або «відкинути». Тип пакета може бути TCP, UDP або ICMP, а прапори відносяться до заголовка IP-пакету. Поля «порт джерела» та «порт призначення» є актуальними лише для пакетів типу TCP та UDP.

Брандмауери, що працюють на рівні програм, використовують спеціалізовані проксі-сервери для конкретних сервісів, таких як TELNET, FTP та інші. Ці проксі запускаються безпосередньо на брандмауер і обробляють весь трафік, що відноситься до свого сервісу. В результаті між клієнтом і сервером створюються два з’єднання: перше – від клієнта до брандмауера, і друге – від брандмауера до кінцевого сервера.

Можливості брандмауера залежать від серверів, що підтримуються, які зазвичай включають [37] :

- термінальні підключення (наприклад, Telnet, Rlogin);
- передача файлів (FTP);

- електронна пошта (SMTP, POP3);
- інтернет-сервіси (HTTP, Gopher, Wais);
- графічні інтерфейси (X Window System);
- друк (принтери);
- віддалені команди (Rsh);
- користувацька інформація (Finger);
- новини (NNTP).

Ці послуги підтримуються щодо різноманітних застосувань і забезпечують контроль над мережним трафіком лише на рівні додатків.

Використання серверів прикладного рівня вирішує важливе завдання – приховує структуру локальної мережі від зовнішніх користувачів, включаючи інформацію в заголовках поштових пакетів або DNS-запитах. Ще однією перевагою є можливість аутентифікації на рівні користувачів (підтвердження особистості, що дозволяє переконатися, що користувач дійсно є тим, за кого себе видає). Про процеси аутентифікації буде розказано докладніше. При налаштуванні правил доступу можна вказати такі параметри, як назва сервісу, ім’я користувача, допустимий часовий інтервал для використання, список дозволених пристроїв та методи автентифікації. Протоколи серверів прикладного рівня забезпечують високий рівень безпеки, оскільки вони спрямовують взаємодію із зовнішніми мережами через обмежену кількість додатків, що повністю контролюють весь вхідний та вихідний трафік.

Сервер рівня з’єднання, у свою чергу, є перетворювачем TCP-з’єднання. Користувач підключається до певного порту брандмауера, а потім брандмауер встановлює з’єднання з потрібним сервером з іншого боку захисту. У процесі сеансу цей транслятор копіює дані обох напрямках, діючи як канал. Зазвичай кінцевий сервер заздалегідь вказано, тоді як до нього можуть підключатися різні джерела (з’єднання типу «один до багатьох»). Для різних конфігурацій можна використовувати різні порти. Такий сервер дозволяє налаштувати трансляцію для будь-яких сервісів на основі TCP, а також контролювати доступ та збирати статистику їхнього використання.

Нижче наведено основні переваги та недоліки пакетних фільтрів та серверів прикладного рівня порівняно один з одним.

Переваги пакетних фільтрів мають відносно низьку вартість, гнучко налаштовуються за допомогою правил фільтрації, а також забезпечують мінімальні затримки під час проходження пакетів.

Недоліки пакетних фільтрів у тому, що локальна мережа стає видимою та доступною для маршрутизації з мережі інтернет, правила фільтрації пакетів вимагають глибоких знань tcp та udp, що може ускладнити налаштування. Також при збої роботи брандмауера всі пристрої за ним стають повністю незахищеними або недоступними, а аутентифікація на основі IP-адреси може бути обійдена за допомогою підробки IP (IP-спуфінг), коли зловмисник використовує IP-адресу іншої системи.

Переваги серверів прикладного рівня полягають у тому, що локальна мережа залишається прихованою від зовнішніх користувачів з Інтернету, а при порушенні роботи брандмауера трафік блокується, запобігаючи загрозам захищеним системам. Також захист на рівні програм дозволяє проводити додаткові перевірки, що знижує ймовірність атак, пов’язаних з уразливістю в ПЗ. Реалізується автентифікація на рівні користувача та можливість системи попередження про спроби злому.

До недоліків серверів прикладного рівня можна віднести вищу вартість порівняно з пакетними фільтрами. Також вони не підтримують роботу з протоколами RPC та UDP, а продуктивність нижча, ніж у пакетних фільтрів.

Зручність управління – один із ключових факторів у створенні надійної системи захисту. Помилки в налаштуванні правил доступу можуть залишити «пролом», через який можливе проникнення зловмисників. Тому більшість брандмауерів оснащені утилітами, що полегшують додавання, видалення та перегляд правил доступу. Ці утиліти також дозволяють перевіряти правила на наявність синтаксичних та логічних помилок, полегшуючи адміністрування. Вони можна групувати інформацію, наприклад, по користувачеві чи сервісу, що полегшує аналіз.

Важливою частиною брандмауера є також система моніторингу та оповіщення про можливі атаки. Дані про події, такі як невдалі спроби доступу, з’єднання, обсяг переданих даних, сервіси та час з’єднання, фіксуються в статистичних файлах. Багато брандмауерів дозволяють налаштовувати типи подій, які слід протоколювати, а також визначати дії при атаці або спробах несанкціонованого доступу – від виведення повідомлення на консоль до повідомлення адміністратора електронною поштою. Миттєві сповіщення про спроби злому допомагають реагувати на загрозу в реальному часі. Також у брандмауери часто вбудовані генератори звітів, які обробляють дані статистики та дозволяють аналізувати використання ресурсів користувачами, доступ до сервісів, відмови та спроби злому, а також їх джерела [37].

Автентифікація – один із найважливіших елементів брандмауера. Перед тим як надати користувачеві доступ до певного сервісу, необхідно переконатися, що він дійсно той, за кого себе видає, і що йому дозволено скористатися цим сервісом. Цей процес називається авторизацією, і зазвичай авторизація йде після успішної аутентифікації: як тільки користувач підтверджений, брандмауер перевіряє, до яких сервісів він має доступ. Коли надходить запит на використання сервісу, брандмауер визначає метод аутентифікації цього користувача і передає управління серверу аутентифікації. Якщо сервер підтверджує особу користувача, брандмауер встановлює з'єднання, що запитується.

Найчастіше використовується метод, званий «знання секрету»: користувач підтверджує особистість, відправляючи серверу певне секретне слово. Однією з найпростіших схем є використання пароля, але цей спосіб досить вразливий, тому що зловмисники можуть його перехопити та використати.

Безпечніший метод — одноразові паролі, які діють лише для одного сеансу. Навіть якщо такий пароль буде перехоплено, його не можна використовувати повторно. Одноразові паролі можуть генеруватися як програмними засобами, так і апаратними пристроями - в останньому випадку

пристрої можуть бути вбудовані в комп'ютер і вимагають знання секретного слова для активації.

Деякі брандмауери підтримують Kerberos, популярний протокол аутентифікації, хоча він може вимагати оновлення клієнтського програмного забезпечення, що не завжди зручно. Більшість комерційних брандмауерів пропонують кілька методів аутентифікації, що дозволяє адміністраторам вибрати найбільш вдалий метод для конкретної ситуації.

Методи боротьби зі спамом можна розділити на кілька категорій [35].

Ручна або автоматична фільтрація за заголовками листів. Користувачі можуть перейти з протоколу POP3 на IMAP4 або Web-інтерфейс і перевіряти листи лише за заголовками, не завантажуючи текст. Багато поштових програм також дозволяють налаштувати автоматичне фільтрування за заголовками, але це вимагає точного налаштування для мінімізації помилок, інакше можливі скарги на пропуск потрібної кореспонденції.

Сервіси фільтрації у поштового провайдера або на окремому сервері. Такі служби можуть фільтрувати пошту до доставки користувача. Цей спосіб зручний, тому що користувач отримує вже очищені листи, однак він також менш контролюємо: є ризик втрати корисної інформації, про що користувач може не дізнатися.

IP-фільтри, що базуються на чорних списках (DNSBL). Фільтрація IP-адрес, які раніше використовувалися для відправлення спаму, посилається на загальні бази даних спамерів. Проте з розвитком технологій спаму цей метод поступово втрачає ефективність.

Фільтрування з автоматичним поповненням списків доступу (access-list). Тут можна аналізувати підозрілі IP-адреси відправників. Однак для цього потрібні регулярні оновлення та налаштування. Цей спосіб дозволяє пропустити перший лист спамеру, тому його ефективність обмежена.

Програми та модулі для аналізу вмісту. Спеціальні програми або модулі аналізують вміст листів, роблячи оцінку та пропонуючи дії. Поштова програма

передає дані для аналізу, а програма повертає свої рекомендації щодо подальшого поводження з листом.

Ці методи допомагають зменшити обсяг небажаної пошти, хоча ефективність кожного з них залежить від контексту та вимог користувача.

Серед антиспам-програм особливий інтерес становлять рішення, що використовують принципи Байєса і які навчаються у процесі обробки повідомлень. Ці програми застосовують байєсовські методи виявлення шаблонів у спамі, аналізуючи його структуру і загальні елементи. Чим більше спам-повідомлень фільтр аналізує, тим точніше його здатність розпізнавати нові листи як спам. Завдяки автокорекції фільтр може адаптуватися до змін у структурі листів автоматично, що робить його більш стійким до нових типів спаму [34].

Крім захисту від небажаних листів, байєсівський фільтр може виявляти і деякі поштові черв'яки. Наприклад, при отриманні першого зараженого листа його можна відзначити як небажане, і фільтр блокуватиме подальші схожі листи, що особливо корисно для хробаків з невеликими варіаціями. У цьому плані антиспам-фільтр може працювати ефективніше за антивірус, оскільки не вимагає оновлення баз даних для виявлення нових модифікацій вірусів.

Для зручності роботи, особливо на операційних системах, де немає вбудованого брандмауера, часто застосовуються сторонні програмні рішення. Серед них популярні такі продукти, як Kaspersky Anti-Hacker, Agnitum Outpost Firewall, ZoneAlarm та інші. В останні роки великого поширення набули комплексні захисні програми, які поєднують функції брандмауера та антивірусу. Наприклад, до них відносяться Kaspersky Internet Security, Norton Internet Security, McAfee Internet Security та інші аналогічні рішення.

РОЗДІЛ 3
КРИПТОГРАФІЧНИЙ ЗАХИСТ ІНФОРМАЦІЇ

3.1 Загальні принципи криптографічного захисту інформації

Криптографічний захист інформації є однією з основних складових забезпечення інформаційної безпеки. Він покликаний забезпечити конфіденційність, цілісність, автентичність та доступність даних, які обробляються, передаються або зберігаються в інформаційних системах. Сучасна криптографія використовує математичні методи та алгоритми для досягнення цих цілей [38].

Одним із основних принципів криптографічного захисту є конфіденційність інформації. Це означає, що доступ до зашифрованих даних має лише той, хто володіє відповідним ключем для їх розшифрування. Конфіденційність досягається за допомогою шифрування — процесу перетворення відкритих даних (plaintext) у зашифрований текст (ciphertext), який неможливо зрозуміти без спеціального ключа.

Другим важливим принципом є цілісність даних, яка забезпечує те, що дані не були змінені під час їх передачі або зберігання. Для цього використовуються хеш-функції, які генерують унікальний контрольний «відбиток» для кожного набору даних. Якщо навіть незначна частина інформації змінюється, хеш-функція створює зовсім інший відбиток, сигналізуючи про порушення цілісності.

Автентичність даних є ще одним ключовим принципом криптографічного захисту. Вона дозволяє впевнитися, що дані були створені або відправлені саме тією стороною, яка заявляє про це. Для цього використовуються цифрові підписи, засновані на асиметричних алгоритмах шифрування [39].

Неможливість відмови від авторства є продовженням принципу автентичності. Цей принцип означає, що відправник не може заперечувати

своє авторство щодо переданих даних. Ця властивість є важливою в юридичних і фінансових транзакціях.

Для забезпечення криптографічного захисту інформації використовуються два основних типи алгоритмів: симетричні та асиметричні. Симетричні алгоритми, такі як AES, використовують один і той самий ключ для шифрування та дешифрування, що забезпечує високу швидкість роботи, але вимагає безпечного обміну ключами. Асиметричні алгоритми, наприклад RSA, використовують пару ключів: відкритий і закритий, що дозволяє уникнути проблем із передачею ключів, проте вони є менш ефективними за швидкістю [11].

Ще одним елементом криптографії є хеш-функції, такі як SHA-256. Вони не шифрують дані, але створюють їх «контрольну суму», яка допомагає перевірити цілісність і забезпечити автентичність.

Таким чином, криптографічний захист інформації базується на використанні математичних методів для забезпечення конфіденційності, цілісності та автентичності даних. Його ефективність залежить від правильного вибору алгоритмів, довжини ключів та їх налаштування, що враховує специфіку інформаційної системи.

3.2 Обґрунтування вибору криптографічного шифру

Криптографічні алгоритми відіграють ключову роль у забезпеченні захисту інформації. Вибір конкретного алгоритму залежить від численних факторів: типу даних, вимог до швидкості, стійкості до атак і ресурсів системи. Кожен із них має свої особливості, переваги та недоліки, які важливо розуміти для побудови надійної системи захисту.

AES став стандартом симетричного шифрування, замінивши застарілий DES завдяки своїй високій ефективності та безпеці. Його робота базується на шифруванні блоків фіксованого розміру (128 біт) із використанням ключів довжиною 128, 192 або 256 біт. Алгоритм реалізує послідовність операцій,

таких як заміна, перестановка та побітовий зсув, що забезпечує стійкість до криптоаналітичних атак [11].

AES відзначається високою швидкістю роботи, що робить його придатним для різних сфер, включаючи захист мережевого трафіку (TLS/SSL), шифрування файлів і зберігання даних у хмарних сервісах. Його популярність зумовлена тим, що він є стандартом для багатьох урядів і корпорацій [13].

RSA є одним із перших і найвідоміших алгоритмів асиметричного шифрування. Він використовує пару ключів: відкритий для шифрування та закритий для дешифрування. Це дозволяє уникнути ризиків, пов’язаних із передачею ключів, характерних для симетричних алгоритмів.

Безпека RSA базується на складності факторизації великих чисел. Хоча алгоритм є дуже стійким до атак, його швидкість нижча, ніж у симетричних методів. Це обмежує його застосування шифруванням невеликих обсягів даних або передачею ключів, які потім використовуються іншими алгоритмами, такими як AES [11].

ChaCha20 – сучасний симетричний потоковий алгоритм, який став популярним завдяки своїй високій швидкості та ефективності. Алгоритм створює послідовність псевдовипадкових чисел, які комбінуються з даними для їхнього шифрування.

Завдяки своїй простоті та швидкості ChaCha20 ідеально підходить для пристроїв із обмеженими ресурсами, таких як мобільні пристрої або системи Інтернету речей (IoT). Його часто використовують у VPN-сервісах, захищених месенджерах і інших реального часу застосунках.

ECC використовує математичні властивості еліптичних кривих, що дозволяє забезпечити високий рівень безпеки за допомогою коротших ключів порівняно з RSA. Наприклад, ECC із ключем 256 біт еквівалентний RSA із ключем 3072 біт [12].

Це робить ECC особливо корисним у системах із обмеженими ресурсами, таких як мобільні пристрої чи вбудовані системи. ECC також знаходить широке застосування в цифрових підписах, автентифікації та

криптовалютних транзакціях, забезпечуючи високу ефективність і компактність.

SHA-256 – це хеш-функція, яка створює унікальний цифровий відбиток даних, що дозволяє перевіряти їхню цілісність. Навіть найменші зміни у вхідних даних призводять до повністю іншого хешу, що гарантує високу стійкість до колізій.

SHA-256 широко використовується у фінансових системах, цифрових підписах і криптовалютах, таких як Bitcoin. Попри появу SHA-3, SHA-256 залишається одним із найнадійніших алгоритмів для багатьох застосувань.

RC5 – це симетричний блочний шифр, розроблений Рональдом Рівестом у 1994 році. Він є надзвичайно гнучким алгоритмом, параметри якого можна налаштовувати: довжина блоку (32, 64 або 128 біт), довжина ключа (до 2048 біт) і кількість раундів [40].

Головною перевагою RC5 є його простота реалізації та висока швидкість. Алгоритм використовує прості операції, такі як додавання, XOR і циклічний зсув, що забезпечує ефективність навіть на пристроях із обмеженими обчислювальними ресурсами. RC5 знаходить застосування в різних системах, включаючи шифрування файлів, мережевий захист і мобільні додатки.

3.3 Огляд алгоритму RC5

RC5 є сучасним блоковим алгоритмом симетричного шифрування, створеним відомим криптографом Рональдом Рівестом у 1994 році. Цей алгоритм привертає увагу своєю унікальною структурою, гнучкістю в налаштуванні параметрів і високою ефективністю, що робить його придатним для широкого спектра застосувань у сфері інформаційної безпеки. У цьому розділі докладно розглянемо причини вибору RC5 для проєкту, його переваги та можливості, які забезпечують високу ефективність і стійкість системи до атак [40].

Однією з головних переваг RC5 є його гнучкість. Алгоритм дозволяє змінювати три основні параметри: розмір блоку, кількість раундів і довжину ключа. Це робить RC5 універсальним інструментом, який можна адаптувати до різних потреб і обмежень інформаційних систем. Наприклад, розмір блоку може становити 32, 64 або 128 біт, що дозволяє налаштувати рівень безпеки залежно від вимог до продуктивності та обсягу оброблюваних даних. У нашому проєкті обрано блок розміром 64 біти, оскільки він забезпечує баланс між високою продуктивністю та достатнім рівнем захисту для сучасних інформаційних систем.

Кількість раундів у RC5 також може варіюватися, що дозволяє підвищити стійкість алгоритму до криптоаналітичних атак. Зазвичай використовують від 12 до 20 раундів: чим більше раундів, тим складніше алгоритм для злому, хоча це й трохи знижує його швидкість. У рамках нашого проєкту було обрано 16 раундів, що є оптимальним для забезпечення надійного захисту даних без значного зниження продуктивності.

Довжина ключа є ще одним важливим параметром RC5, який можна налаштувати відповідно до потреб безпеки. RC5 підтримує ключі довжиною від 0 до 2040 біт, що дає можливість вибирати рівень захисту залежно від типу даних і ризиків, пов’язаних із їх компрометацією. У цьому проєкті обрано ключ довжиною 128 біт, який вважається стандартом для більшості сучасних криптографічних систем і забезпечує високий рівень захисту від атак типу «грубої сили».

Простота реалізації RC5 також стала важливим аргументом на його користь. Алгоритм базується на використанні кількох простих операцій, таких як додавання, віднімання і побітовий циклічний зсув. Ці операції легко реалізувати як у програмному забезпеченні, так і на апаратному рівні, що робить RC5 доступним для широкого спектра платформ — від серверів до мобільних пристроїв. Простота реалізації знижує ризик помилок під час інтеграції алгоритму в програмне забезпечення підприємства, а також сприяє зменшенню витрат на розробку [41].

Важливим аспектом при виборі RC5 є його висока швидкість роботи. Завдяки простим і ефективним операціям RC5 демонструє чудову продуктивність навіть на пристроях із обмеженими ресурсами, що є ключовим для нашого проєкту. Це дозволяє алгоритму шифрувати великі обсяги даних у реальному часі, не створюючи значного навантаження на апаратні ресурси системи.

Стійкість RC5 до криптоаналітичних атак заслуговує окремої уваги. Структура алгоритму забезпечує надійний захист від диференціального та лінійного криптоаналізу завдяки залежності результату кожного раунду від попередніх операцій і ключа. Побітові зсуви, які є основною частиною алгоритму, ускладнюють аналіз, що робить RC5 надзвичайно стійким до зломів.

Ще однією перевагою є універсальність RC5. Він може бути ефективно застосований у різних середовищах: серверних системах, мобільних пристроях, вбудованих системах і мережевих протоколах. У нашому проєкті RC5 інтегрується в програмне забезпечення підприємства для забезпечення захищеного шифрування даних, що передаються по мережі, та збереження конфіденційності інформації.

З огляду на всі ці переваги, RC5 був обраний як основний криптографічний алгоритм для реалізації захисту інформації в рамках цього проєкту. Його гнучкість, простота реалізації, висока продуктивність і стійкість до атак забезпечують надійний рівень безпеки та відповідають вимогам сучасних інформаційних систем.

3.4 Реалізація алгоритму RC5 у програмі

У цьому підрозділі розглядається процес реалізації алгоритму RC5 у програмі Lingualand EXAM+, що забезпечує ефективне шифрування даних для гарантії їхньої конфіденційності.

Програма надає користувачам можливість виконувати низку ключових дій, зокрема:

- реєстрація користувачів;
- авторизація через логін і пароль;
- шифрування даних;
- дешифрування даних;
- збереження інформації у текстовий файл;
- відкриття текстового файлу для подальшої роботи з даними в програмі;
- вихід із програми.

Доступ до програми забезпечується директором освітнього центру, який надає працівникам логіни та паролі для входу в систему.

На рисунках 3.1–3.12 показано етапи роботи програми шифрування.

Головне меню програми дає змогу зареєструвати нового користувача або виконати вхід у систему, якщо обліковий запис уже створено. У разі введення неправильного логіна або пароля програма сповіщає про помилку.

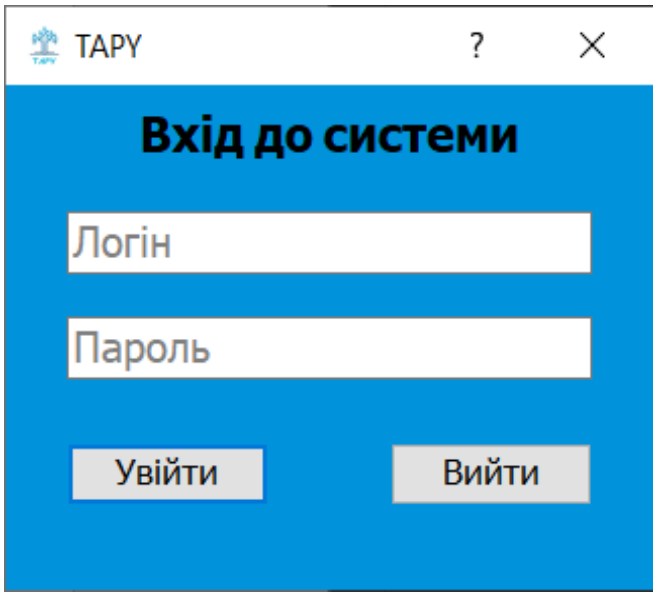

Рисунок 3.1 – Вікно авторизації користувача

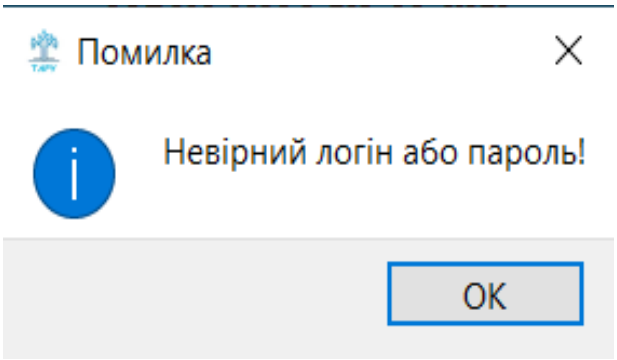

Рисунок 3.2 – Помилка авторизації користувача

Після успішного входу в систему відображається поле для введення тексту та меню, яке містить три підпункти.

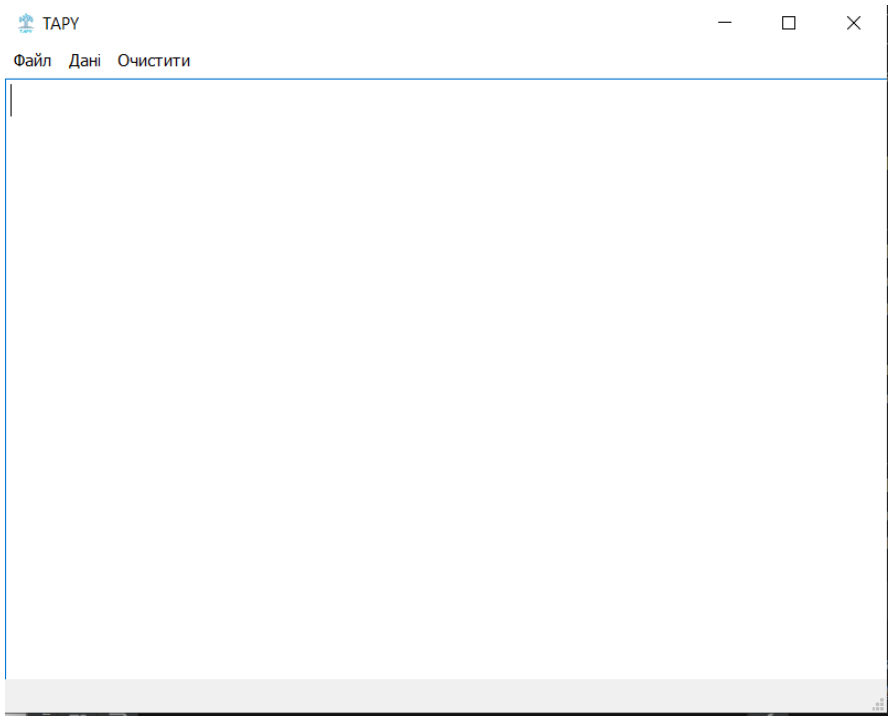

Рисунок 3.3 – Вікно для введення даних для шифрування та розшифрування

Щоб створити зашифроване повідомлення, виконайте наступні дії:

1. Введіть текст повідомлення у текстовому блоці.
2. Виберіть у меню пункт «Дані» → «Зашифрувати» для шифрування повідомлення.
3. У меню виберіть пункт «Файл» → «Зберегти», щоб зберегти зашифроване повідомлення.

Рисунок 3.4 – Введення інформації для шифрування

Рисунок 3.5 – Зашифроване повідомлення.

Рисунок 3.6 – Опція меню «Зберегти файл»

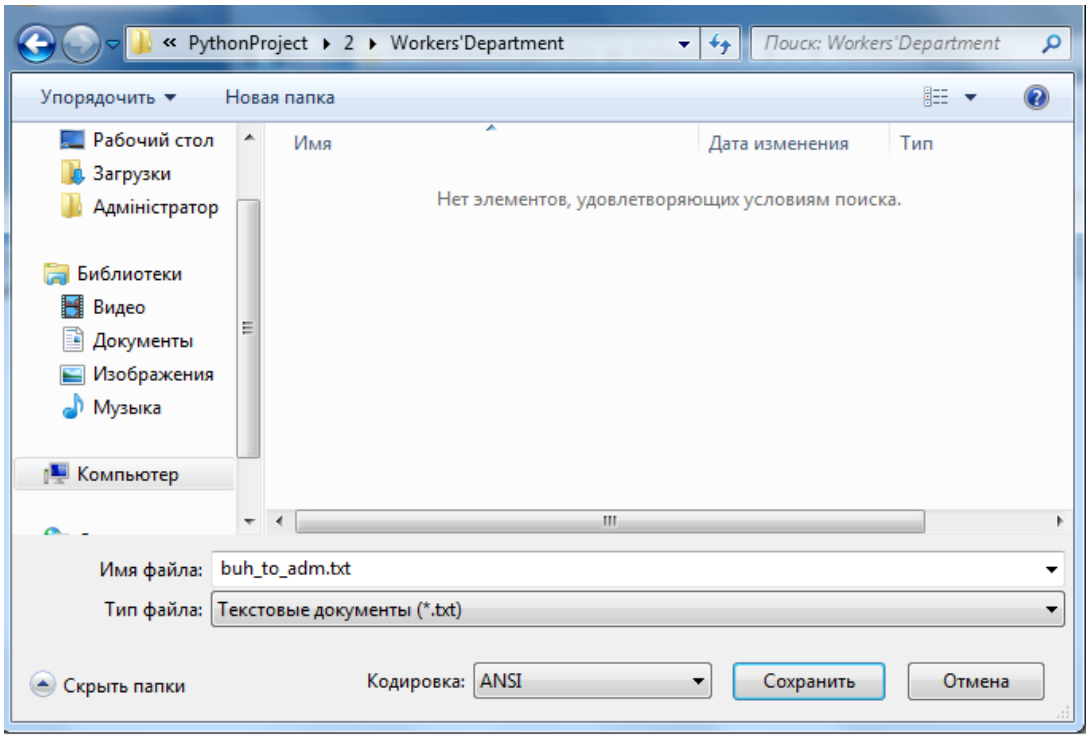

Рисунок 3.7 – Збереження інформації.

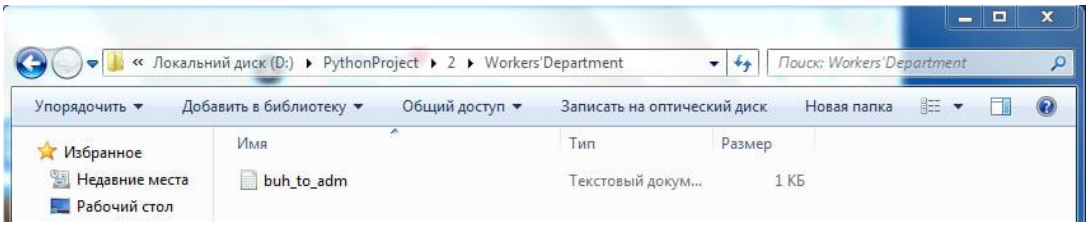

Рисунок 3.8 – Збережене повідомлення

Також є можливість відкривати раніше створені файли. Для розшифрування файлу необхідно:

1. Відкрити файл.
2. Виконати розшифрування.
3. Натиснути «Зберегти» та вибрати місце для збереження файлу.

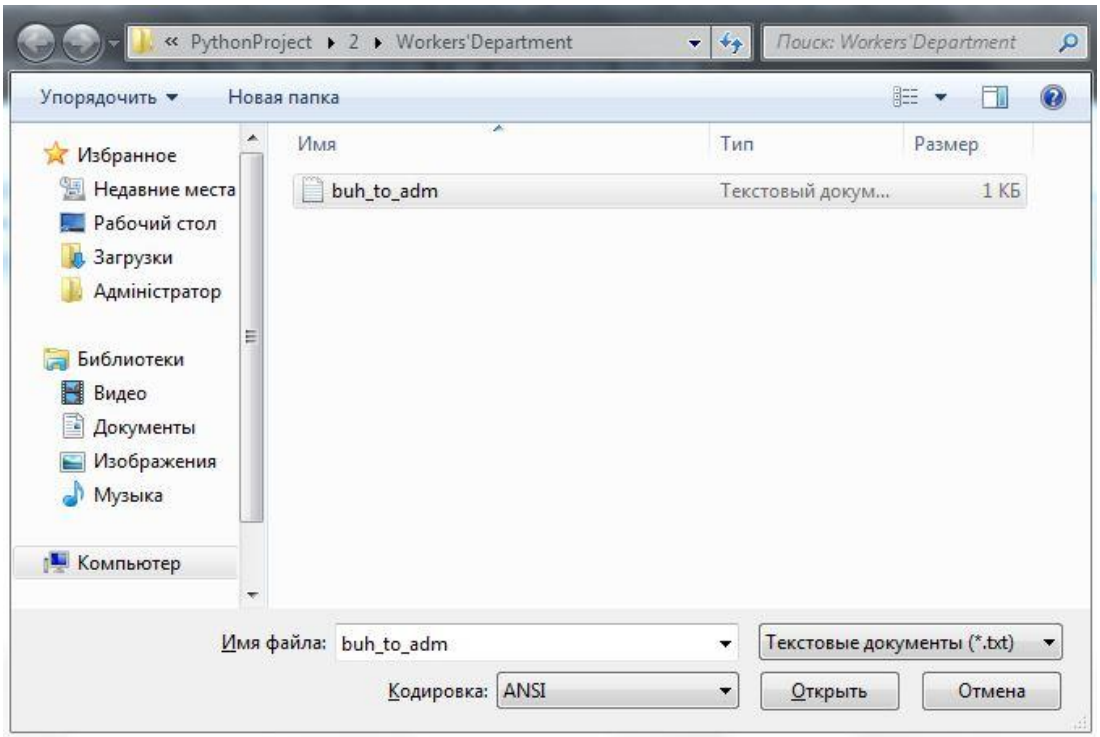

Рисунок 3.9 – Відкриття файлу

Рисунок 3.10 – Зашифрований файл

Рисунок 3.11 – Виконати розшифрування файлу

Рисунок 3.12 – Розшифрований файл

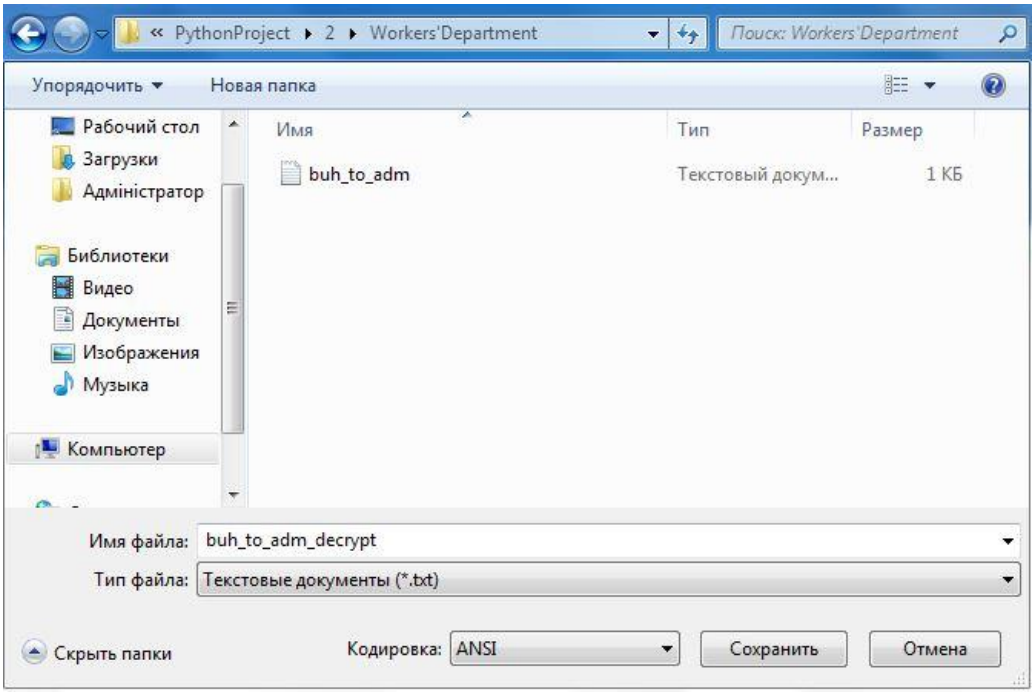

Рисунок 3.13 – Процес збереження файлу

Таким чином, інформація зберігається в окремих папках, до яких мають доступ лише певні посадові особи

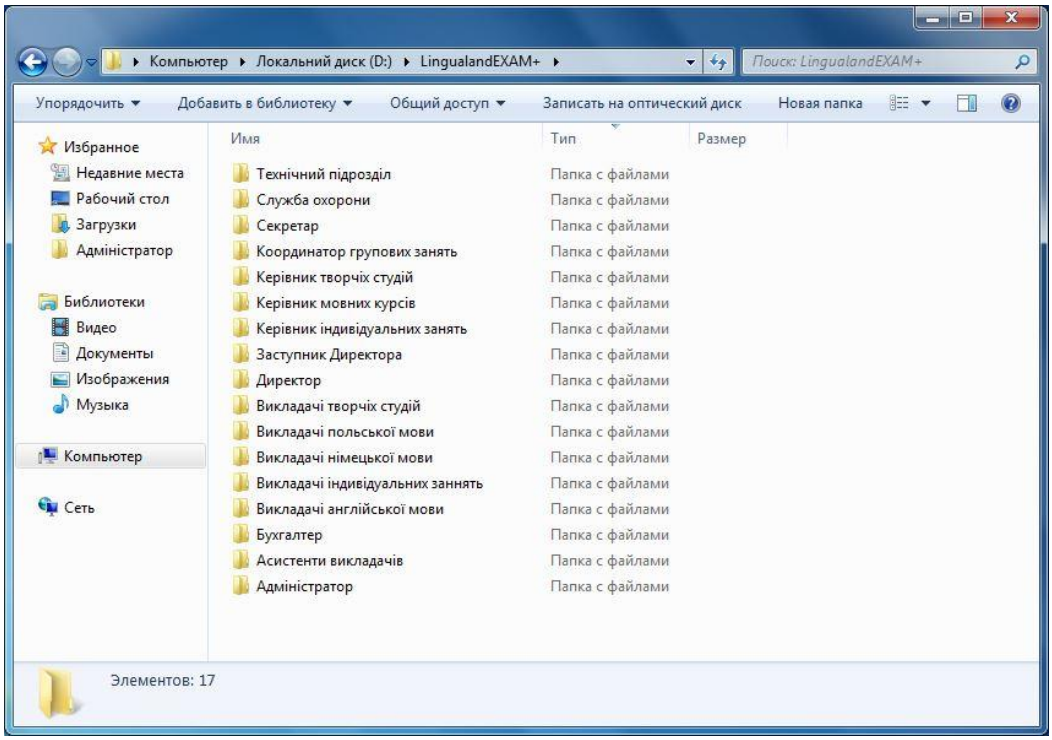

Рисунок 3.14 – Папки зі збереженою інформацією

На рисунку 3.14 показано папки, в яких зберігається інформація (згідно з рисунком 2.2 – Організаційна структура освітнього центру LINGUALAND Exam+).

ВИСНОВКИ

У ході дипломної роботи на тему «Створення проєкту комплексного захисту інформації підприємства із використанням програми для шифрування даних» було досягнуто поставленої мети — розроблено ефективний проєкт захисту інформації для освітнього центру LINGUALAND Exam+. Основний акцент було зроблено на забезпеченні конфіденційності, цілісності та доступності даних через впровадження сучасних методів криптографії.

Робота ґрунтувалася на вивченні теоретичних аспектів інформаційної безпеки, зокрема, розглядалися особливості симетричного та асиметричного шифрування, а також гібридних підходів. Було проаналізовано сучасні загрози інформаційній безпеці та запропоновано оптимальні методи їхньої мінімізації. Особливу увагу приділено алгоритму RC5, який, завдяки своїй простоті, гнучкості й ефективності, став основою для розробки програмного забезпечення.

Освітній центр LINGUALAND Exam+ як об’єкт дослідження має специфічні вимоги до захисту інформації, адже в його мережі обробляються як загальні дані, так і конфіденційна інформація про студентів, викладачів і фінансові операції. Проведений аналіз дозволив сформувати чітке розмежування доступу до даних, що зменшує ризики несанкціонованого втручання.

Практична частина роботи зосереджена на розробці програми, яка реалізує шифрування даних за допомогою алгоритму RC5. Програма дозволяє вводити, шифрувати, дешифрувати та зберігати інформацію, забезпечуючи її захист від стороннього доступу. Реалізація функціоналу була адаптована до специфіки діяльності освітнього центру, що робить систему зручною у використанні та водночас надійною.

Запропоноване рішення демонструє високу ефективність у захисті інформації та здатність мінімізувати ризики, пов’язані з витоком даних. Крім того, його можна масштабувати для застосування в інших організаціях, які мають схожі вимоги до безпеки інформаційних ресурсів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Основи інформаційної безпеки. URL: https://pdf.lib.vntu.edu.ua/books/IRVC/Dudikevich_2018_316.pdf.

2. Основи інформаційної та кібернетичної безпеки / В. Бурячок. URL: https://elibrary.kubg.edu.ua/27370/1/V_Buriachok_Posibnik_2019_FITU.pdf.

3. Руденко, Ю. І. Стандарти інформаційної безпеки в корпоративних системах / Ю. І. Руденко. — Харків: *Видавництво «Харківський університет»*, 2019.

4. Шевченко, О. П. Системи управління інформаційною безпекою в корпоративних мережах / О. П. Шевченко. — Київ: *Видавництво «Техніка»*, 2022.

5. Бєляєв, О. М. Інформаційна безпека в сучасних системах управління / О. М. Бєляєв. — Одеса: *Видавництво «Астропринт»*, 2019.

6. ISO/IEC 27001:2013 — Системи управління інформаційною безпекою. Вимоги. — Женева: *Міжнародна організація зі стандартизації (ISO)*, 2013.

7. Мартин, Д. С. Управління інформаційною безпекою: теорія та практика / Д. С. Мартин. — Київ: *Видавництво «Київський університет»*, 2021.

8. SSL/TLS Сертифікати URL: https://www.ssl.com.ua.

9. Алгоритми симетричного та асиметричного шифрування даних. URL: https://pidruchniki.com/1622012758499/kompyuteri/metodi_sifruvannya.

10. Балаганський Д. Криптографічні методи захисту інформації. URL: https://www.securinfo.com.ua.

11. Петрів М. Історія та розвиток алгоритмів шифрування AES та RSA. URL: https://kpi.ua/crypto_methods.

12. ECC та сучасні криптографічні методи URL: https://itsecurity.ua.

13. Фролов А. Гібридне шифрування та його застосування в SSL/TLS протоколах. URL: https://crypto-it.com.

14. Басіста І. І., Колісник О. М. Оцінка ризиків в інформаційних системах: методи та інструменти // *Журнал інформаційної безпеки*, 2021, №3.

15. Гречухін С. М. Моделі та методи оцінки інформаційних ризиків в умовах невизначеності. — Львів: *Видавництво ЛНУ*, 2022.

16. Каплінська О. В., Шумило Т. П. Кількісна оцінка ризиків інформаційної безпеки: сучасні підходи. — *Вісник економічної кібернетики*, 2021, №5.

17. Кабак В. І., Шкода В. М. Математичні моделі для оцінки ризиків інформаційної безпеки // *Журнал прикладної математики*, 2023, №2.

18. Кальченко С. А. Управління інформаційними ризиками: концепції, методи та моделі. — Київ: *Видавництво «Знання»,* 2020.

19. Ентоні Дж. «Методи оптимізації управління ризиками в кіберпросторі». — Київ: *Видавництво «Кібербезпека»*, 2023.

20. Центр досліджень кібербезпеки НАТО. *«Методології оцінки ризиків у кіберпросторі»*. — Таллінн, 2023.

21. Власенко А. В., Федорчук О. М. Ризики інформаційної безпеки у малих і середніх підприємствах // Бізнес Інформ, 2023, №4.

22. Lingualand Exam+ URL: http://lingua.org.ua/ — Дата звернення: 03.01.2025.

23. Венделін, В. Г. Організаційні структури управління підприємствами: навчальний посібник / В. Г. Венделін. — К.: *Центр учбової літератури*, 2021. — 256 с.

24. Гребеник О. М. Характеристики даних та інформаційних потоків у корпоративних мережах // *Вісник інформаційних технологій*, 2022, №4.

25. Боброва Т. І. Розмежування прав доступу як інструмент забезпечення інформаційної безпеки // *Журнал інформаційної безпеки*, 2021, №3.

26. Кобзар, О. М. Розмежування прав доступу в інформаційних системах: практичні аспекти // *Журнал кібербезпеки*. — 2023. — № 4. — С. 47-55.

27. Нікітін, П. С. Методи ідентифікації та автентифікації в сучасних інформаційних системах / П. С. Нікітін. — Одеса: *Порти і технології*, 2023. — 152 с.

28. Коваленко, Т. В. Аудит доступу та моніторинг інформаційних систем / Т. В. Коваленко. — Львів: *Видавництво ЛНУ*, 2022. — 176 с.

29. Власенко, В. І. Інженерно-технічний захист інформаційних систем // Хмельницький, 2022.

30. Журавльов, О. М. Апаратні засоби захисту інформації: теорія та практика / О. М. Журавльов, Н. В. Ткаченко. — Харків: Фоліо, 2023. — 198 с.

31. Коваленко, Т. В. Основи криптографії: шифрувальні алгоритми та їх застосування / Т. В. Коваленко. — Одеса: *Одеський політехнічний університет,* 2021. — 190 с.

32. Шевченко, С. Г. Інженерно-технічні засоби захисту інформаційних ресурсів // Київ: *Видавництво технічної літератури*, 2023.

33. Гора, О. О., Семенова, В. С. Основи антивірусного захисту та безпеки інформаційних систем / О. О. Гора, В. С. Семенова. — Київ: *Академперіодика*, 2020.

34. Томін, В. М. Антивірусні програми та їх застосування для захисту інформаційних систем / В. М. Томін. — Харків: *Видавничий дім «Фактор»,* 2019.

35. Жуков, О.В. Комп'ютерні віруси: класифікація та методи боротьби / О.В. Жуков. — Харків: *Ранок*, 2021. — 184 с.

36. Федеральне агентство з інформаційної безпеки (FAS). Методи захисту від шкідливих програм. URL: https://www.fas.gov/security.

37. Городецький, А.С. Основи побудови брандмауерів у корпоративних мережах / А.С. Городецький. — Харків: *Фоліо*, 2020. — 272 с.

38. Кузьмін, С. І. Основи криптографії / С. І. Кузьмін. — Київ: *Видавничий центр «Інформатика»,* 2020.

39. Голяк, В. О. Методи криптографічного захисту в інформаційних системах / В. О. Голяк. — Львів: *Академперіодика*, 2021.

40. Рівест, Р. RC5: ефективний блоковий алгоритм шифрування / Р. Рівест. — Массачусетс: *Видавництво «MIT Press»,* 1994.

41. RC5: Огляд алгоритму та його застосування. Journal of Cryptography and Security, 2022.

ДОДАТОК А

ЛІСТИНГ ПРОГРАМИ

```
import sys
import json
from PyQt5 import uic, QtWidgets
from main_window import editor

Form, _ = uic.loadUiType("qt_design/dialog_log_in.ui")


class Check_user(QtWidgets.QDialog, Form):
    """
    Вхідне вікно програми, де відбувається перевірка
    користувачів при вході в програму
    """
    def __init__(self):
        super(Check_user, self).__init__()
        self.setupUi(self)
        self.check_data.clicked.connect(self.check_user)
        self.quit_app.clicked.connect(lambda: self.close())

    def check_user(self):
        login = self.user_login.text()
        password = self.user_password.text()
        users_data = json.load(open("data/users_data.json", "r"))
        if (self.user_login.text() in users_data) and (users_data[login] ==
password):
            self.close()
            self._to_editor(login)
            self._open_main_window()
        else:
            QtWidgets.QMessageBox.information(
                self, "Помилка", "Невірний логін або пароль!")

    def _to_editor(self, logs):
        print(logs)
        editor.verification = logs


    def _open_main_window(self):
        self.w = editor.Main_window()
        self.w.show()

def main(): # Запуск першого вікна програми
    app = QtWidgets.QApplication(sys.argv)
    w = Check_user()
    w.show()
    sys.exit(app.exec_())

if __name__ == "__main__":
    main()
from PyQt5 import uic, QtWidgets
from PyQt5.QtWidgets import QFileDialog, QMainWindow
from RC5.RC5_code import RC5 # Передаємо клас, в якому буде здійсюватися
шифрування

Form, _ = uic.loadUiType("qt_design/main_window.ui")

global nameFile
global verification
```

```
global data_base

class Main_window(QMainWindow, Form):
    """
    Головне вікно програми, де відбувається шифрування
    та розшифрування даних файлів або повідомлень користувачів
    """
    def __init__(self):
        super(Main_window, self).__init__()
        self.setupUi(self)
        self.open_file.triggered.connect(self._open_file_name_dialog)
        self.save_file.triggered.connect(self._save_file_name_dialog)
        self.encrypt_data.triggered.connect(self._encrypt_text)
        self.decipher_data.triggered.connect(self._decipher_text)
        self.clear_data.triggered.connect(self._clear_text)

        self.RC5_key = RC5('tapy')

    def _open_file_name_dialog(self):
        options = QFileDialog.Options()
        options |= QFileDialog.DontUseNativeDialog
        file_name, _ = QFileDialog.getOpenFileName(self,"TAPY",
                                                   "","Text Files (*.txt)",
options=options)
        if (self.check(verification) == True):
            if (f"{verification}" in file_name):
                with open(file_name, "r") as file:
                    self.user_text.setText(file.read())
            else:
                QtWidgets.QMessageBox.information(
                    self, "Помилка", "Відмовлено в доступі!")

    def _save_file_name_dialog(self):
        options = QFileDialog.Options()
        options |= QFileDialog.DontUseNativeDialog
        file_name, _ = QFileDialog.getSaveFileName(self, "TAPY",
f"{verification}",
                                                   "Text Files (*.txt)",
options=options)
        if (self.check(verification) == True):
            if (f"{verification}" in file_name):
                with open(file_name, "w") as file:
                    file.write(self.user_text.toPlainText())
            else:
                QtWidgets.QMessageBox.information(
                    self, "Помилка", "Відмовлено в доступі!")

    def _encrypt_text(self):
        text = self.RC5_key.encrypt_str(self.user_text.toPlainText())
        self.user_text.setText(text)

    def _decipher_text(self):
        text = self.RC5_key.decrypt_str(self.user_text.toPlainText())
        self.user_text.setText(text)

    def _clear_text(self):
        self.user_text.setText("")

    def check(self, verification):
        if (verification == "serverroom"):
            print(f"Hello {verification}")
        elif (verification == "dir"):
            print(f"Hello {verification}")
        elif (verification == "zdir"):
```

```
            print(f"Hello {verification}")
        elif (verification == "sec"):
            print(f"Hello {verification}")
        elif (verification == "adm"):
            print(f"Hello {verification}")
        elif (verification == "kerind"):
            print(f"Hello {verification}")
        elif (verification == "kertv"):
            print(f"Hello {verification}")
        elif (verification == "kermov"):
            print(f"Hello {verification}")
        elif (verification == "vtv"):
            print(f"Hello {verification}")
        elif (verification == "buh"):
            print(f"Hello {verification}")
        elif (verification == "teh"):
            print(f"Hello {verification}")
        elif (verification == "it"):
            print(f"Hello {verification}")
        elif (verification == "ohor"):
            print(f"Hello {verification}")
        elif (verification == "kord"):
            print(f"Hello {verification}")
        elif (verification == "vindiv"):
            print(f"Hello {verification}")
        elif (verification == "vangl"):
            print(f"Hello {verification}")
        elif (verification == "vnim"):
            print(f"Hello {verification}")
        elif (verification == "vpol"):
            print(f"Hello {verification}")
        elif (verification == "assi"):
            print(f"Hello {verification}")
        return True
import base64
import os
from io import BytesIO


class RC5(object):
    def __init__(self, key):
        self.mode = 'CBC'  # Режим для шифрування
        self.blocksize = 32
        self.rounds = 12
        self.iv = os.urandom(self.blocksize // 8)
        self._key = key.encode('utf-8')

    @staticmethod
    def _rotate_left(val, r_bits, max_bits):
        v1 = (val << r_bits % max_bits) & (2 ** max_bits - 1)
        v2 = ((val & (2 ** max_bits - 1)) >> (max_bits - (r_bits %
max_bits)))
        return v1 | v2

    @staticmethod
    def _rotate_right(val, r_bits, max_bits):
        v1 = ((val & (2 ** max_bits - 1)) >> r_bits % max_bits)
        v2 = (val << (max_bits - (r_bits % max_bits)) & (2 ** max_bits - 1))

        return v1 | v2

    @staticmethod
    def _expand_key(key, wordsize, rounds):
        # Частини ключів, які поєднуються в одне слово
```

```
        def _align_key(key, align_val):
            while len(key) % (align_val):
                key += b'\x00'  # Додає 0 біти поки не заповнить розмір ключа

            L = []
            for i in range(0, len(key), align_val):
                L.append(int.from_bytes(key[i:i + align_val],
byteorder='little'))

            return L

        # Генерація констант
        def _const(w):
            if w == 16:
                return (0xB7E1, 0x9E37)  # Повертає значення P та Q
            elif w == 32:
                return (0xB7E15163, 0x9E3779B9)
            elif w == 64:
                return (0xB7E151628AED2A6B, 0x9E3779B97F4A7C15)

        # Генератор псевдо-рандому для списку S
        def _extend_key(w, r):
            P, Q = _const(w)
            S = [P]
            t = 2 * (r + 1)
            for i in range(1, t):
                S.append((S[i - 1] + Q) % 2 ** w)

            return S

        def _mix(L, S, r, w, c):
            t = 2 * (r + 1)
            m = max(c, t)
            A = B = i = j = 0

            for k in range(3 * m):
                A = S[i] = RC5._rotate_left(S[i] + A + B, 3, w)
                B = L[j] = RC5._rotate_left(L[j] + A + B, A + B, w)

                i = (i + 1) % t
                j = (j + 1) % c

            return S

        aligned = _align_key(key, wordsize // 8)
        extended = _extend_key(wordsize, rounds)

        S = _mix(aligned, extended, rounds, wordsize, len(aligned))

        return S

    @staticmethod
    def _encrypt_block(data, expanded_key, blocksize, rounds):
        w = blocksize // 2
        b = blocksize // 8
        mod = 2 ** w

        A = int.from_bytes(data[:b // 2], byteorder='little')
        B = int.from_bytes(data[b // 2:], byteorder='little')

        A = (A + expanded_key[0]) % mod
        B = (B + expanded_key[1]) % mod

        for i in range(1, rounds + 1):
```

```
            A = (RC5._rotate_left((A ^ B), B, w) + expanded_key[2 * i]) % mod
            B = (RC5._rotate_left((A ^ B), A, w) + expanded_key[2 * i + 1]) %
mod

        res = A.to_bytes(b // 2, byteorder='little') + B.to_bytes(b // 2,
byteorder='little')
        return res

    @staticmethod
    def _decrypt_block(data, expanded_key, blocksize, rounds):
        w = blocksize // 2
        b = blocksize // 8
        mod = 2 ** w

        A = int.from_bytes(data[:b // 2], byteorder='little')
        B = int.from_bytes(data[b // 2:], byteorder='little')

        for i in range(rounds, 0, -1):
            B = RC5._rotate_right(B - expanded_key[2 * i + 1], A, w) ^ A
            A = RC5._rotate_right((A - expanded_key[2 * i]), B, w) ^ B

        B = (B - expanded_key[1]) % mod
        A = (A - expanded_key[0]) % mod

        res = A.to_bytes(b // 2, byteorder='little') + B.to_bytes(b // 2,
byteorder='little')
        return res

    def encrypt_file(self, infile, outfile):
        w = self.blocksize // 2
        b = self.blocksize // 8

        if self.mode == 'CBC':
            last_v = self.iv
            # встановлює iv на початку файлу
            outfile.write(last_v)

        expanded_key = RC5._expand_key(self._key, w, self.rounds)

        chunk = infile.read(b)

        while chunk:
            chunk = chunk.ljust(b, b'\x00')
            if self.mode == 'CBC':
                chunk = bytes([a ^ b for a, b in zip(last_v, chunk)])

            encrypted_chunk = RC5._encrypt_block(chunk, expanded_key,
                                                 self.blocksize,
                                                 self.rounds)
            outfile.write(encrypted_chunk)
            last_v = encrypted_chunk

            chunk = infile.read(b)  # Зчитування розміру тексту в файлі

    def decrypt_file(self, infile, outfile):
        w = self.blocksize // 2
        b = self.blocksize // 8
        if self.mode == 'CBC':
            last_v = outfile.read(b)

        expanded_key = RC5._expand_key(self._key, w,
                                       self.rounds)

        chunk = infile.read(b)
```

```
        while chunk:
            decrypted_chunk = RC5._decrypt_block(chunk, expanded_key,
                                                 self.blocksize,
                                                 self.rounds)
            if self.mode == 'CBC':
                decrypted_chunk = bytes([a ^ b
                                         for a, b in zip(last_v,
                                                         decrypted_chunk)])
                last_v = chunk
            chunk = infile.read(b)  # Зчитування розміру тексту в файлі
            if not chunk:
                decrypted_chunk = decrypted_chunk.rstrip(b'\x00')

            outfile.write(decrypted_chunk)

    def encrypt_str(self, input_str):
        str_in = BytesIO()
        str_in.write(input_str.encode('utf-8'))
        str_in.seek(0)
        str_out = BytesIO()

        self.encrypt_file(str_in, str_out)

        return base64.urlsafe_b64encode(str_out.getvalue()).decode("utf-8")

    def decrypt_str(self, input_enc_str):
        enc_bytes = base64.urlsafe_b64decode(input_enc_str)

        byte_in = BytesIO()
        byte_in.write(enc_bytes)
        byte_in.seek(0)
        byte_out = BytesIO()

        self.decrypt_file(byte_in, byte_out)

        return byte_out.getvalue().decode('utf-8')
```