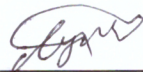


Міністерство освіти і науки України
Державний університет «Житомирська політехніка»
Факультет інформаційно-комп'ютерних технологій
Кафедра комп'ютерної інженерії та кібербезпеки

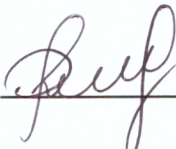
ПОЯСНЮВАЛЬНА ЗАПИСКА

до випускної кваліфікаційної роботи магістра
на тему: «Методи та технології побудови систем кібер-
захисту для об'єктів критичної інфраструктури»

Виконала студентка 2-го курсу групи КБм-23-1
спеціальності 125 «Кібербезпека та захист інформації»

 М.О. Луцевич

Керівник старший викладач кафедри комп'ютерної ін-
женерії та кібербезпеки

 Є.М. Трокоз

Рецензент доцент кафедри комп'ютерної інженерії та кі-
бербезпеки, кандидат технічних наук, доцент

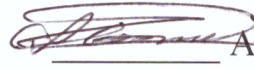
 О.В. Маєвський

Житомир – 2024

Державний університет «Житомирська політехніка»
Факультет інформаційно-комп'ютерних технологій
Кафедра комп'ютерної інженерії та кібербезпеки
Спеціальність 125 «Кібербезпека та захист інформації»
Освітня програма «Кібербезпека»

ЗАТВЕРДЖУЮ

Завідувач
кафедри комп'ютерної
інженерії та кібербезпеки

 Андрій Єфіменко

17 вересня 2024 р.

ЗАВДАННЯ

на випускню кваліфікаційну роботу магістра

Студент: Луцевич Марта Олександрівна

Тема роботи: **«Методи та технології побудови систем кіберзахисту для об'єктів критичної інфраструктури»**

затверджена Наказом університету від 17 вересня 2024 р. № 479/с

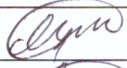
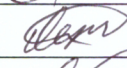
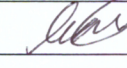
Термін здачі студентом закінченої роботи 25 грудня 2024 р.

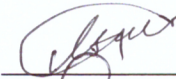
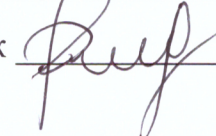
Вихідні дані роботи (зазначається предмет і об'єкт дослідження): Об'єктом дослідження є процеси забезпечення кібербезпеки об'єктів критичної інфраструктури. Предметом дослідження є методи та технології побудови модульних систем кіберзахисту.

Обладнання та програмне забезпечення:

1. Cisco Packet Tracer
2. PyCharm2023
3. Python 3.0

Консультанти з випускної кваліфікаційної роботи із зазначенням розділів, що їх стосуються

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
1	Похотило О.А.	18.09.2024 	18.09.2024 
2	Похотило О.А.	07.10.2024 	07.10.2024 
3	Похотило О.А.	12.11.2024 	12.11.2024 

Студент  М.О. Луцевич
 Керівник  Є.М. Трокоз

Календарний план

№ з/п	Назви етапів випускної кваліфікаційної роботи	Термін виконання етапів роботи	Примітка
1	Постановка завдання	17.09.2024-25.09.2024	
2	Аналіз літературних джерел	25.09.2024-27.09.2024	
3	Аналіз об'єкта дослідження	27.09.2024-05.10.2024	
4	Аналіз аналогічних розробок	05.10.2024-07.10.2024	
5	Проектування модульної системи кіберзахисту	07.10.2024-25.10.2024	
6	Реалізація та тестування програмних модулів	25.10.2024-13.11.2024	
7	Впровадження системи кіберзахисту	13.11.2024-30.11.2024	
8	Написання пояснювальної записки	01.12.2024-23.12.2024	
9	Отримання відгуку та рецензії на кваліфікаційну роботу	23.12.2024-24.12.2024	
10	Захист	25.12.2024	

Студент  М.О. Луцевич

Керівник  Є.М. Трокоз

АНОТАЦІЯ

Пояснювальна записка на тему «Методи та технології побудови системи кіберзахисту для об'єктів критичної інфраструктури» складається із трьох розділів, висновку, списку використаної літератури та додатків.

Текстова частина викладена на 99 сторінках друкованого тексту.

Пояснювальна записка має список використаних джерел, що містить 30 найменувань та займає 4 сторінки. В роботі наведено 44 рисунки. Загальний обсяг роботи 113 сторінок.

Заваданням кваліфікаційної рооботи було:

- Ідентифікація потенційних загроз;
- Оцінка ризиків;
- Розробка захисних модулів;
- Розробка рекомендацій;
- Розробка програмних застосунків;
- Визначення перспектив вдосконалення;

В першому розділі було викладено теоретичні основи дослідження, зокрема, аналіз літературних джерел та нормативно-правової бази з кібербезпеки, визначено основні загрози для критичної інфраструктури, а також наведено ключові принципи побудови системи кіберзахисту.

В другому розділі було викладено практичні аспекти створення системи кіберзахисту, включаючи модульний підхід до її розробки. Детально описано окремі модулі, виконано аналіз відповідності запропонованої системи міжнародних стандартів безпеки.

В третьому розділі було викладено можливості впровадження розробленої системи на різних типах підприємств критичної інфраструктури

Висновок містить в собі результати побудови системи кіберзахисту для об'єктів критичної інфраструктури

КЛЮЧОВІ СЛОВА: КРИТИЧНА ІНФРАСТРУКТУРА, МОДУЛЬНИЙ ПІДХІД, ФАКТОРИ НЕБЕЗПЕКИ.

ANNOTATION

Explanatory Note on the Topic "Methods and Technologies for Building a Cybersecurity System for Critical Infrastructure Objects" consists of three chapters, a conclusion, a list of references, and appendices.

The text part is presented on 99 pages of printed text.

The explanatory note includes a list of 30 sources used, which occupies 4 pages. The work contains 44 figures. The total volume of the work is 113 pages.

The tasks of the qualification work were:

- Identification of potential threats;
- Risk assessment;
- Development of protective modules;
- Development of recommendations;
- Development of software applications;
- Determination of improvement prospects.

In the first chapter, the theoretical foundations of the research were outlined, including the analysis of literary sources and the regulatory and legal framework on cybersecurity. The main threats to critical infrastructure were identified, and the key principles for building a cybersecurity system were presented.

The second chapter outlined the practical aspects of creating a cybersecurity system, including a modular approach to its development. The individual modules were described in detail, and an analysis of the proposed system's compliance with international security standards was performed.

The third chapter presented the possibilities of implementing the developed system in different types of critical infrastructure enterprises.

The conclusion contains the results of building a cybersecurity system for critical infrastructure objects.

KEYWORDS: CRITICAL INFRASTRUCTURE, MODULAR APPROACH, THREAT FACTORS.

ЗМІСТ

ВСТУП	8
РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ КІБЕРЗАХИСТУ КРИТИЧНОЇ ІНФРАСТРУКТУРИ	9
1.1. Загрози та вразливості об'єктів критичної інфраструктури	9
1.2. Огляд сучасних методів і технологій кіберзахисту	13
1.3. Теоретичні основи модульності у побудові систем захисту	17
Висновки до першого розділу	25
РОЗДІЛ 2. РОЗРОБКА ПРОЄКТУ МОДУЛЬНОЇ СИСТЕМИ КІБЕРЗАХИСТУ	26
2.1. Опис модулів системи кіберзахисту.....	26
2.2. Фактори небезпеки та відповідні модулі для їх нейтралізації.....	71
2.3. Програмна реалізація	73
Висновки до другого розділу.....	87
РОЗДІЛ 3 ОЦІНКА ЕФЕКТИВНОСТІ СИСТЕМИ ТА НАПРЯМИ ЇЇ ВДОСКОНАЛЕННЯ.....	88
3.1. Впровадження системи на різних типах підприємств.....	88
3.2. Можливі напрями покращення системи	92
ВИСНОВКИ.....	94
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ	95
ДОДАТКИ.....	99

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

KI — Критична інфраструктура

IoT — Інтернет речей (Internet of Things)

IDS — Система виявлення вторгнень (Intrusion Detection System)

IPS — Система запобігання вторгнень (Intrusion Prevention System)

AI — Штучний інтелект (Artificial Intelligence)

ML — Машинне навчання (Machine Learning)

SIEM — Управління подіями інформаційної безпеки (Security Information and Event Management)

VPN — Віртуальна приватна мережа (Virtual Private Network)

HIPAA — Акт про переносимість та відповідальність за медичне страхування (Health Insurance Portability and Accountability Act)

PCI DSS — Стандарт безпеки даних платіжних карток (Payment Card Industry Data Security Standard)

IDS/IPS — Система виявлення та запобігання вторгнень (Intrusion Detection and Prevention System)

SET — Social-Engineer Toolkit

ВСТУП

Актуальність теми. Необхідність розробки модульних систем кіберзахисту, які поєднують ефективне протистояння новітнім формам кібератак, адаптивність до змінних умов та масштабованість для різних типів об'єктів критичної інфраструктури в умовах поточної ситуації. Використання національних стандартів шифрування, розвиток адаптивних систем моніторингу та впровадження інноваційних підходів до захисту від соціоінженерних атак дозволяють створити комплексні рішення, які покращать стійкість та надійність критичних інфраструктур.

Метою роботи є розробка та обґрунтування методів і технологій побудови модульних систем кіберзахисту для об'єктів критичної інфраструктури, здатних забезпечити оперативне виявлення, попередження та реагування на кібератаки в умовах сучасних загроз.

Завданням роботи є створення комплексної стратегії кібербезпеки для захисту критичної інфраструктури.

Об'єктом дослідження є процеси забезпечення кібербезпеки об'єктів критичної інфраструктури.

Предметом дослідження є методи та технології побудови модульних систем кіберзахисту, що включають застосування криптографії, моніторингових систем, адаптивних механізмів реагування, а також інноваційних підходів до виявлення та запобігання кібератакам.

Практичне значення роботи полягає у можливості застосування розроблених рекомендацій та методик під час проектування та вдосконалення модульних систем кіберзахисту критичних інфраструктур, особливо в умовах гібридних загроз і військових конфліктів.

Наукова новизна полягає у розробці комплексного підходу до побудови модульних систем кіберзахисту, що поєднують сучасні криптографічні стандарти, методи аналізу трафіку, інтелектуальні системи виявлення інцидентів та принципи адаптації до різних типів загроз.

РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ КІБЕРЗАХИСТУ КРИТИЧНОЇ ІНФРА-СТРУКТУРИ

1.1. Загрози та вразливості об'єктів критичної інфраструктури

Зростання кількості кіберзагроз і зростання складності кібератак є ознакою сучасного стану розвитку інформаційних технологій. Об'єкти критичної інфраструктури, такі як об'єкти енергетичної сфери, транспортні вузли, системи водопостачання та зв'язку, державні реєстри та сервери постійно піддаються тиску з боку можливих зловмисників, які намагаються дестабілізувати їх роботу, отримати несанкціонований доступ до важливої інформації або спричинити порушення роботи систем. Це особливо важливо в світлі гібридних загроз і поточної війни в Україні.

Критична інфраструктура складається з компонентів соціальної структури та економіки країни, які є найважливішими для функціонування суспільства. Електростанції, системи водопостачання, транспортні мережі, державні реєстри, системи зв'язку та інші можуть входити до цієї категорії. У разі їх зупинки або порушення суспільство може зазнати значних збитків, бути вразливим, отримати тимчасове або довгострокове послаблення.

У країнах по всьому світу для забезпечення національної безпеки використовується поняття «критична інфраструктура», яке охоплює системи та об'єкти, які є життєво важливими для забезпечення життя людей і держави, і дестабілізація роботи яких може призвести до тяжких негативних або навіть катастрофічних наслідків. Каскадні ефекти, також відомі як «ефект доміно», представляють значну небезпеку, оскільки порушення роботи одного об'єкта критичної інфраструктури впливає на роботу інших об'єктів і систем. Критична інфраструктура також включає особливо небезпечні виробництва. Аварії, які виникають через будь-яку причину, наприклад природні або техногенні надзвичайні ситуації, зловмисне чи необережне поводження, також можуть мати катастрофічні наслідки для певних районів і їхніх жителів [1].

Згідно з даними Міністерства Розвитку Громад та Територій України, існує щонайменше 9 видів загроз критичній інфраструктурі, такі як терористичні, інформаційні та диверсійні загрози, кіберзагрози, загрози соціального, природного, воєнного та техногенного характеру, загрози, що пов'язані з діями персоналу, що спричинені недостатньою кваліфікацією. Цей список не є вичерпним та може бути доповнений [2].

Наведемо приклади кожного з типів загроз на критичну інфраструктуру України для ширшого розуміння проблематики.

Терористичні загрози. У 2014 році почалася російсько-українська війна, яка призвела до низки атак на енергетичну інфраструктуру України. Наприклад, у грудні 2015 року кібератака на енергетичну систему Західної України призвела до відключення понад 230 тисяч людей від електроенергії. Зловмисники використали програмне забезпечення BlackEnergy, щоб отримати доступ до систем управління та відключити електроенергію. Це був перший відомий випадок кібератаки, яка призвела до відключення електроенергії [3].

Інформаційні загрози. У січні 2022 року було здійснено масовану кібератаку на веб-сайти українського уряду, включаючи сайти Кабінету Міністрів, Міністерства закордонних справ та інших державних установ. Зловмисники створили паніку серед населення, розмістивши погрози та неправдиву інформацію на цих сайтах. Цей напад був лише одним із елементів гібридної війни, яку Росія веде проти України, використовуючи кібератаки та дезінформацію, щоб дестабілізувати ситуацію в країні [4].

Диверсійні загрози. У травні 2022 року нафтопровід «Дружба», який є одним із основних шляхів постачання нафти з Росії до Європи, був атакований диверсійними групами. Злочинці зруйнували частину трубопроводу, що призвело до витіку нафти, що призвело до тимчасової зупинки транспортування. Це спричинило значні економічні втрати та негативні наслідки для навколишнього середовища [5].

Кіберзагрози. Industroyer, також відомий як Crashoverride, є першим відомим фреймворком, розробленим для атак на електромережі. 17 грудня 2016 року він був використаний для кібератак на енергетичну мережу України. Industroyer має модульну структуру, яка включає основний backdoor для управління всіма компонентами, альтернативний backdoor для збереження доступу, ланцюг для запуску компонентів навантаження та видалення даних. Крім того, він містить чотири частини навантаження, які спрямовані на певні протоколи промислового зв'язку. Industroyer був створений для перешкоджання роботі промислових систем керування, особливо електричних підстанцій [6].

Загрози соціального характеру. Під час повномасштабного вторгнення чимало терористичних атак відбувається не лише на об'єкти критичної інфраструктури, а й безпосередньо у житлові райони, бізнес-центри та торгові центри, громадські місця скупчення людей, пологові будинки, станції метро і подібне. Також однією з наймасштабніших атак була атака на компанію «Київстар», результатом якої стало відключення 24 мільйонів абонентів від усіх послуг.

Загрози природного характеру. У квітні 2020 року через сильні дощі та поєднані постраждали цілі райони на заході України, зокрема на Івано-Франківщині та в Чернівцях. Сотні будинків, дороги, мости та інша інфраструктура були зруйновані повінню. Деякі мости були повністю зруйновані, що ускладнило доступ до постраждалих місць і уповільнило евакуацію людей [7].

Загрози воєнного характеру. У ході активної фази бойових дій у 2022-2024 роках російські війська кілька разів атакували енергетичну інфраструктуру України. Трансформаторні підстанції, лінії електропередачі та інші енергетичні об'єкти були зруйновані. Це призвело до масових відключень електроенергії, які вплинули на життя мільйонів людей і роботу промислових підприємств [8].

Загрози техногенного характеру. У 2013 році сталася аварія на заводі «Стирол» у Горлівці, внаслідок якої стався витік аміаку. Це спричинило забруднення навколишнього середовища та отруєння працівників заводу та місцевих жителів. Технічні проблеми та порушення правил безпеки призвели до аварії. [9].

Загрози, що пов'язані з діями персоналу, що у зв'язку з недостатньою кваліфікацією. У 1986 році на Чорнобильській АЕС сталася аварія, частково спричинена діями некваліфікованого персоналу. Грубі дії персоналу призвели до погіршення ситуації. Через незнання небезпеки реактора працівники АЕС припустили низку помилок і ненавмисно порушили настанови [10].

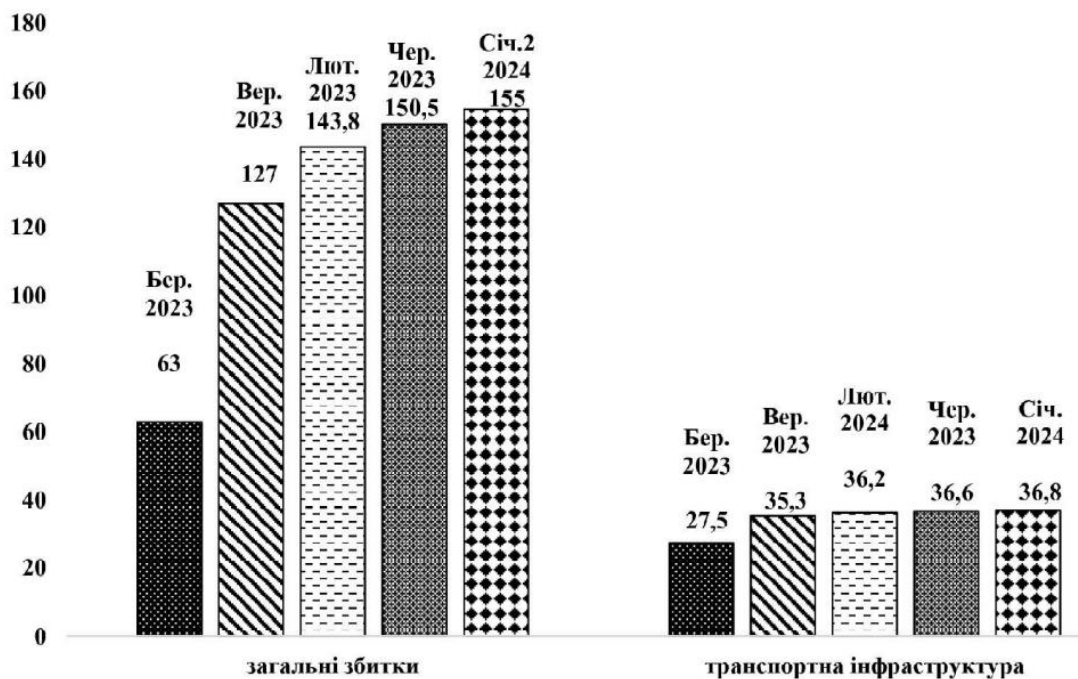


Рис. 1. Обсяги збитків критичної інфраструктури України протягом 2022—2023 років, млрд. \$

Джерело: узагальнено авторами на основі даних аналітичного центру при Київській школі економіки [1].

Рисунок 1.1 — Обсяги збитків критичної інфраструктури України протягом 2022—2023 років, млрд. \$

На Рисунку 1.1 наведено обсяги збитків критичної інфраструктури України у 2022—2023 роках, що демонструють зростання економічних втрат унаслідок війни та атак на об'єкти інфраструктури, зокрема транспортні мережі та енергосистеми [2].

Загрози та вразливості критичної інфраструктури вказують на потребу впровадження адаптивних та модульних систем кіберзахисту. Такий підхід до-

звояє інтегрувати новітні технології та швидко реагувати на нові загрози. Оскільки концепція критичної інфраструктури охоплює велику кількість об'єктів, таких як електростанції, системи водопостачання, об'єкти зберігання та виробництва продуктів харчування, критичні транспортні вузли, телекомунікаційні мережі та медичні установи [11], модульний підхід дозволяє інтегрувати лише ті компоненти, які відповідають потребам конкретного об'єкта. У наступних розділах розглядатимуться методи і способи забезпечення надійного захисту критично важливих об'єктів з урахуванням їхніх потреб у масштабованості та модульності.

1.2. Огляд сучасних методів і технологій кіберзахисту

1. Загальні тенденції розвитку технологій кіберзахисту

Проактивні та адаптивні системи: замість використання реактивних стратегій, адаптивні та проактивні системи передбачають і реагують на загрози в режимі реального часу. Вони не просто реагують на вже відомі атаки, вони передбачають нові загрози. Наприклад, використання проактивного аналізу трафіку мережі може допомогти знайти аномальну активність до того, як вона призведе до пошкодження.

Zero Trust Architecture: це концепція, яка передбачає, що жоден користувач або пристрій не є надійним за замовчуванням. Ідея полягає в тому, що не можна довіряти нікому, навіть відомим користувачам усередині мережі. Такі системи включають багаторівневу автентифікацію, контроль доступу та постійний моніторинг активності. Наприклад, працівники компанії або підприємства повинні проходити автентифікацію щоразу, коли вони намагаються отримати доступ до ресурсів компанії.

Shift-Left у безпеці (DevSecOps): впровадження безпеки на ранніх стадіях розробки програмного забезпечення, щоб дозволяє виявити вразливості до запуску системи. Це знижує ризик і витрати на виправлення вразливостей на більш

пізніх етапах. Наприклад, інструменти статичного аналізу коду можуть бути інтегровані в процес розробки для виявлення потенційних загроз до початку тестування.

Machine Learning та AI-driven Security: використання алгоритмів машинного навчання та штучного інтелекту для виявлення нетипової поведінки, аналізу логів у реальному часі, прогнозування атак і реагування на інциденти. До прикладу, алгоритми машинного навчання можуть аналізувати поведінку користувачів у мережі та автоматично блокувати підозрілі дії.

SOAR (Security Orchestration, Automation and Response): зменшує людський фактор, автоматизуючи рутинні завдання та швидке реагування на інциденти. SOAR-платформа може автоматично блокувати та ізолювати заражений комп'ютер із мережі, відправляти повідомлення фахівцям із безпеки та записувати процес розслідування інциденту.

2. Методи виявлення та запобігання атакам

Поведінковий аналіз та Heuristic Detection: виявлення за допомогою аналізу поведінки програмного забезпечення та систем. Цей метод дозволяє виявити нові небезпеки, які не матимуть відомих підписів. Наприклад, поведінковий аналіз може виявити незвичну та підозрілу активність працівника, що може свідчити про спробу кібератаки.

Використання машинного навчання в IDS/IPS: системи виявлення та запобігання вторгнень, які виконують адаптивне навчання та виявлення аномалій за допомогою алгоритмів машинного навчання. До прикладу, такі системи можуть виявляти нетипову мережеву активність, що свідчить про спробу несанкціонованого доступу.

3. Системи централізованого моніторингу та управління безпекою

SIEM (Security Information and Event Management): системи централізованого збору, аналізу та зберігання логів з різних джерел, що полегшує виявлення небезпеки в режимі реального часу. Вони забезпечують кореляцію подій, виявлення аномалію і швидку реакцію на загрози. Наприклад, система SIEM може

автоматично перевірити підозрілу активність, аналізуючи журнали подій з мережевих пристроїв і серверів. Поєднання роботи SIEM-систем та алгоритмів ШІ підвищує ефективність виявлення та реагування на загрози [12].

SOAR (Security Orchestration, Automation and Response): платформи, що доповнюють SIEM-системи можливістю автоматизованого реагування на інциденти. Вони дозволяють налаштувати сценарії автоматичного реагування, такі як блокування доступу, ізоляція підозрілого хосту, відправлення повідомлення фахівцям з безпеки.

В разі підозрілої активності, SOAR-платформа автоматично блокуватиме доступ до мережі без допомоги або контролю спеціаліста. Це, у свою чергу, дозволить спеціалістам зосередитися на більш пріоритетних задачах, заощадити бюджет компанії та звільнити час спеціаліста від монотонної роботи обробки кожного сповіщення та зменшить можливість помилки через людський фактор [13].

XDR (Extended Detection and Response): системи, які об'єднують дані з різних джерел (пристроїв кінцевих користувачів, серверів, мережевих пристроїв, хмарних середовищ і систем електронної пошти), щоб забезпечити цілісне уявлення про загрози та вдосконалити підходи до виявлення загроз і реагування на них. XDR-система може аналізувати дані з антивірусного програмного забезпечення, фаєрволів та інших систем безпеки для виявлення складних атак.

4. Хмарні технології та безпека

CASB (Cloud Access Security Broker): рішення для забезпечення безпеки при доступі до хмарних сервісів. Наприклад, CASB дозволяє контролювати доступ користувачів до корпоративних хмарних сервісів та забезпечувати шифрування переданих даних.

SASE (Secure Access Service Edge): інтеграція мережевих і безпекових функцій у єдину хмарну платформу. SASE-платформа може забезпечувати захищений доступ до корпоративних ресурсів для віддалених співробітників, об'єднуючи функції VPN та антивірусу.

5. Криптографічні технології

Post-Quantum Cryptography: це криптоалгоритми, стійкі до квантових комп'ютерів. Сучасні алгоритми шифрування, можуть бути розкриті квантовими комп'ютерами у відносно короткий час за допомогою використання дискретного логарифмування або ж розкладання цілого числа на прості множники [14]. Тому розробляються нові алгоритми, які зможуть протистояти квантовим атакам.

Наприклад, всім відомий алгоритм RSA, алгоритм ECC (шифр на базі еліптичних кривих) або ж Lattice-based cryptography (базується на складній математиці латинських решіток) пропонують стійкі методи шифрування, які забезпечують високу безпеку навіть у квантовій епосі. Це означає, що навіть якщо зломщик матиме квантовий комп'ютер, він не зможе розшифрувати дані, зашифровані за допомогою постквантових алгоритмів.

Варто зауважити, що квантові шифри ще перебувають на стадії розробки і ця галузь обіцяє значні прориви в майбутньому [15][16].

Гомоморфне шифрування: це метод шифрування, який дозволяє виконувати обчислення над зашифрованими даними (шифротекст) без необхідності їх розшифрування. Це означає, що можна здійснювати різні обчислювальні операції, зберігаючи конфіденційність даних.

Наприклад, банківські установи можуть переглядати або аналізувати фінансові дані клієнтів, виконуючи обчислення без розшифрування даних, забезпечуючи високий рівень конфіденційності та безпеки.

У хмарних обчисленнях також можна використовувати гомоморфне шифрування, що дозволяє без ризику опрацювати конфіденційні дані, які провайдер хмарних послуг надає.

Крім того, алгоритми ШІ можуть використовувати такий тип шифрування для обробки конфіденційних даних, що забезпечує приватність під час аналізу великих кількостей даних. Це дозволяє виконувати машинне навчання на зашифрованих даних, зберігаючи конфіденційність навіть під час складних обчислювальних операцій.

6. Біометрична автентифікація та управління ідентифікацією

IAM (Identity and Access Management): управління обліковими записами та привілеями. Як приклад, використання системи IAM для контролю доступу співробітників до корпоративних ресурсів та управління їхніми правами доступу.

ZTNA (Zero Trust Network Access): безпечний доступ до додатків з мінімальними привілеями. Як зазначалось раніше, ZTNA дозволяє користувачам отримувати доступ лише до тих додатків і даних, які необхідні для їхньої роботи, без доступу до інших ресурсів мережі. Крім того, ZTNA забезпечує детальний контроль доступу та моніторинг дій користувачів, що сприяє оперативному виявленню та реагуванню на потенційні загрози.

7. Системи резервного копіювання та відновлення даних (Disaster Recovery)

Застосовується для резервування даних та інфраструктури на випадок кібератак або технічних збоїв. Використовується автоматизовані системи резервного копіювання для швидкого відновлення критичних даних та послуг.

1.3. Теоретичні основи модульності у побудові систем захисту

Забезпечення надійності та безпеки критичної інфраструктури залежить від її модифікації. Різноманіття небезпек і складність їхньої природи вимагають гнучких, адаптивних і здатних швидко реагувати на нові виклики систем. Для забезпечення високої ефективності системи кіберзахисту модульний підхід дозволяє розподіляти функціональні обов'язки між усіма модулями.

Система захисту може бути модульною, якщо вона розділена на окремі функціональні блоки або модулі, які можуть бути незалежно інтегровані у вже існуючі структури та налаштовані відповідно до конкретних вимог кожного підприємства. Це гарантує гнучкість, масштабованість та ефективність кіберзахисту, що дозволяє адаптувати систему до різноманітних типів важливої інфраструктури. Якщо розглядаються конкретні потреби різних об'єктів критичної інфраструктури, таких як енергетичні системи, медичні заклади, транспортні мережі тощо, то значення модульного підходу стає ще більш очевидним.

Модульний підхід розділяє систему на окремі функціональні блоки, які можуть бути окремо розроблені, інтегровані та змінені відповідно до конкретних потреб. Це дозволяє кожному модулю виконувати спеціальні завдання, що забезпечує гнучкість, масштабованість і надійність загальної системи.

Такий підхід дозволяє легко налаштовувати систему під потреби конкретного об'єкта інфраструктури, оновлювати або замінювати окремі компоненти без впливу на загальну роботу системи, а також забезпечує масштабованість системи з урахуванням її розширення або ж модифікації.

Модульний підхід до побудови системи кіберзахисту базується на ряді принципів, які забезпечують ефективність, гнучкість і масштабованість захисних механізмів.

Нижче ми детально розглянемо кожен принцип.

1. Гнучкість та адаптивність

Системи кіберзахисту повинні бути здатними адаптуватися до швидкого розвитку кіберзагрози в сучасному світі. Модульний підхід дозволяє швидко інтегрувати нові частини, що дозволяє оперативно реагувати на нові загрози та адаптувати структуру захисту до нових вимог та сценаріїв загроз.

Якщо енергетичний об'єкт потребує посиленого захисту мережевих протоколів, можна додати модуль IDS/IPS для виявлення атак. Для енергетичних об'єктів, таких як ТЕЦ чи насосні станції, актуальним є захист мережевої інфраструктури та протидія атакам дронів, тому інтегруються мережевий модуль і модуль фізичного захисту.

У той же час, для медичних установ важливішими будуть модулі шифрування даних пацієнтів для захисту персональної інформації.

Для адміністративних установ (такі як наприклад, мерії, військові адміністрації або держустанови) важливо забезпечити резервне копіювання, технічний захист інформації та надійне шифрування для збереження функціонування під час потенційних атак.

Ця адаптивність дозволяє ефективно використовувати ресурси, орієнтуючись на реальні потреби об'єкта, а не на використання універсального рішення.

2. Сумісність та інтеграція

Модульна архітектура забезпечує взаємодію різних компонентів системи кіберзахисту через стандартизовані інтерфейси та протоколи обміну даними. Це дозволяє інтегрувати як існуючі, так і нові технологічні рішення, зберігаючи загальну цілісність системи.

Деякі модулі системи кіберзахисту можуть управлятися через централізовану платформу, що спрощує моніторинг, контроль і координацію дій. Це дозволяє модулям ефективно взаємодіяти та обмінюватися інформацією.

SIEM (Security Information and Event Management) як модуль моніторингу подій інтегрується з SOAR (Security Orchestration, Automation, and Response), що дозволяє автоматизувати реагування на інциденти безпеки.

IDS/IPS системи (Intrusion Detection/Prevention Systems) можуть взаємодіяти з модулями поведінкового аналізу, що забезпечує своєчасне виявлення загроз та мінімізацію помилкових спрацювань.

Модулі шифрування даних можуть інтегруватися з системами резервного копіювання для захисту даних на випадок збоїв або атак.

Таким чином, сумісність і інтеграція модульних систем кіберзахисту дозволяє забезпечити багаторівневий захист, гнучку адаптацію до загроз та підвищену ефективність використання ресурсів.

3. Локалізація загроз

Модульна структура дозволяє локалізувати загрози, утримуючи безпеку в ураженій частині системи та мінімізуючи ризик її поширення на інші частини системи. Це зберігає цільність і стабільність у всій системі, дозволяючи швидко виявляти та реагувати на інциденти безпеки.

Кожен модуль працює самостійно та має чіткі межі взаємодії з іншими модулями. Це означає, що система може відключити або ізолювати уражений модуль, якщо ви виявите загрозу, щоб запобігти посиленню шкідливого впливу.

Наприклад, мережевий модуль може бути тимчасово відключений для деталь-ного аналізу, якщо він виявляє ризиковану активність. Тим часом інші модулі, такі як шифрування даних або резервне копіювання, продовжують працювати безперебійно.

Сегментація мережі — це підхід до локалізації загрози, яка поділяє мережу на окремі частини або зони безпеки. Кожен сегмент має правила доступу та за-хисту, що дозволяє локалізувати загрозу. Наприклад, у медичному закладі можна підвищити безпеку, створивши окремі розділи для медичних пристроїв, даних пацієнтів та адміністративної інформації.

Моніторинг і виявлення: модульний підхід дозволяє впроваджувати спеці-алізовані модулі для моніторингу та виявлення загроз у різних частинах системи. Це включає в себе використання системи і запобігання вторгненню (IDS/IPS), які здатні виявляти та запобігати вторгненню, а також систему SIEM, яка викорис-товується для централізованого збору та аналізу логів. Такі модулі можуть шви-дко виявити активність і автоматично ввести певні заходи, щоб усунути загрозу.

Автоматизація реагування: після загрози системи автоматизації реагування (SOAR) можуть автоматично виконувати дії з нейтралізації загрози. Це може включати блокування зловмисних IP-адрес, ізоляцію пошкоджених систем або блокування пошкоджених облікових записів. Автоматизація реагування значно скорочує час, необхідний для реакції на інциденти, і зменшує ймовірність того, що вони будуть поширюються далі по системі. Локалізація загроз дозволяє уни-кнути "ефекту доміно", коли збій одного компонента призводить до збоїв у всій системі.

Відновлення та аналіз: після ізоляції загрози важливо провести ретельний аналіз інциденту та відновити системи, які постраждали. Скориставшись моду-лями резервного копіювання та відновлення даних, можна швидко відновити ро-боту системи до стану, який стався до інциденту, мінімізуючи втрату даних і по-легшуючи роботу. Аналіз інциденту допомагає визначити причини та слабкі мі-сця, що дозволяє вдосконалювати захисні механізми.

4. Масштабованість:

У системах модульного кіберзахисту масштабованість є важливим принципом, який дозволяє системам адаптуватися до змін вимог і завантаження без втрати ефективності або безпеки. Модульна архітектура дозволяє поступово розширювати, зменшувати або ж оновлювати систему відповідно до потреб конкретної організації.

Масштабовані системи кіберзахисту можна легко розширити, щоб задовольнити зростаючі потреби, без значних переробок існуючої структури. Це означає, що при збільшенні рівня загрози або обсягу оброблюваних даних можна додати додаткові модулі, такі як аналітика загрози, додаткові засоби моніторингу або кращі функції реагування.

Базові модулі, такі як моніторинг і шифрування даних, можуть бути достатніми для малих об'єктів, таких як невеликі офіси або окремі відділи. Але великі інфраструктурні проекти, такі як національні енергетичні мережі або великі медичні заклади, можуть вимагати додаткових модулів для аналітики, реагування на інциденти та резервування даних.

Окрім того, підприємства можуть додавати або зменшувати кількість модулів у залежності від змін у кіберзагрозах, бюджетних обмежень або операційних вимог. Наприклад, у періоди підвищеної кіберзагрози можна тимчасово збільшити кількість модулів для додаткового моніторингу та аналізу.

Фактори небезпеки як основа захисту

Усі об'єкти критичної інфраструктури мають характеристики та піддаються різним факторам небезпеки. Це означає, що для проектування системи кіберзахисту потрібно використовувати адаптивний підхід. Успішний захист залежить від хорошого розуміння особливостей об'єкта та вразливості його середовища. Також необхідно враховувати динамічність сучасних загроз, які постійно еволюціонують, вимагаючи регулярного оновлення стратегій захисту. Лише комплексний підхід, що охоплює технічні, організаційні та людські фактори, може забезпечити ефективний і довгостроковий кіберзахист критичних об'єктів.

Наприклад, у медичних установах, де особлива увага приділяється збереженню конфіденційності даних пацієнтів та запобігання соціальній інженерії, акцент робиться на:

- Навчання співробітників правилам кібергігієни. Це включає підвищення обізнаності про основні кіберзагрози та постійні тренінги з виявлення фішингових атак.
- Забезпечення даних за допомогою модулів криптозахисту та резервного копіювання. Забезпечення безперервності роботи навіть під час кібератаку або технічних збоїв є реальною необхідністю.
- Підтримка психічної стійкості співробітників, для зменшення кількості стресових факторів. У критичних ситуаціях психологічна підтримка може допомогти вам уникнути помилкових дій.

Для інфраструктур, де участь людини мінімальна (наприклад, насосні станції чи енергетичні об'єкти), основна увага приділяється:

- Захисту від фізичних загроз, таких як атаки БПЛА. Впровадження систем виявлення та нейтралізації дронів забезпечує захист об'єктів від повітряних атак.
- Моніторингу технічного стану обладнання та впровадженню аварійних планів. Регулярні перевірки і використання сенсорних мереж дозволяють мінімізувати ризики техногенних аварій.
- Виявленню та запобіганню кібератакам через системи IDS/IPS та аналіз поведінкових аномалій. Це допомагає ефективно реагувати на нові типи загроз.

Крім того, важливим аспектом є резервне копіювання критичних даних та забезпечення їхньої доступності у разі інцидентів. Це включає створення географічно розподілених резервних сховищ, які дозволяють оперативно відновити роботу об'єктів навіть у випадку серйозних кібератак або фізичних пошкоджень. Також слід враховувати постійне вдосконалення захисних технологій та проведення тестувань на проникнення для виявлення потенційних вразливостей системи.

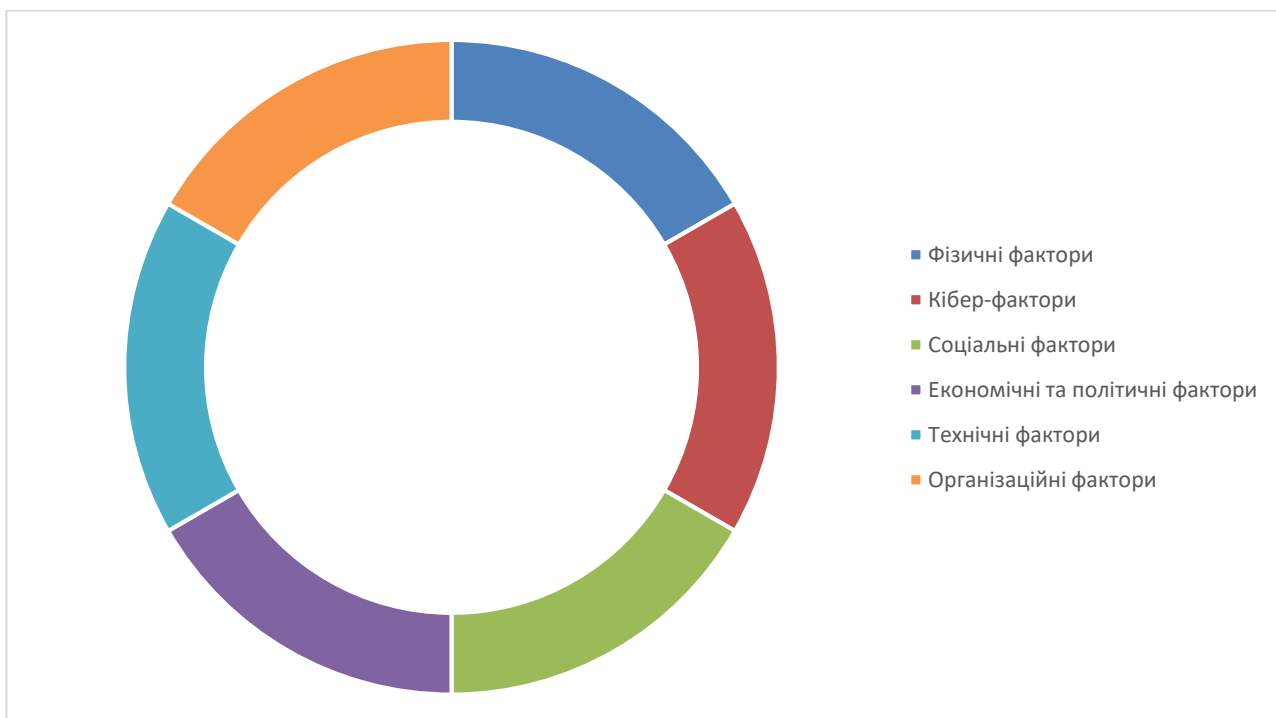


Рисунок 1.2 — Модель комплексного захисту критичної інфраструктури

На діаграмі 1.1 зображено модель комплексного захисту критичної інфраструктури (КІ), яка базується на врахуванні шести ключових факторів небезпеки: фізичних, кібер-факторів, соціальних, технічних, економічних та політичних, а також організаційних аспектів.

Залежно від конкретних потреб об'єкта критичної інфраструктури кожен із цих модулів може використовуватися одночасно, створюючи комплексний захист. Наприклад, у той час як насосні станції потребують фізичного захисту та технічного огляду, медичні заклади зосереджені на соціальних питаннях і криптозахисті. Такий метод гарантує ефективність і гнучкість кіберзахисної системи.

Фізичні фактори стосуються захисту інфраструктури від фізичних атак, природних катастроф чи диверсій. Наприклад, для захисту насосних станцій пропонується впровадження антидронових систем і регулярного моніторингу технічного стану обладнання.

Кібер-фактори включають захист від кібератак, вразливостей у програмному забезпеченні та загроз для IoT-пристроїв. Для таких випадків використовуються системи IDS/IPS, багаторівнева автентифікація та впровадження криптографічних алгоритмів, як-от шифрування даних.

Соціальні фактори спрямовані на мінімізацію ризиків, пов'язаних із людським фактором. Наприклад, у медичних установах акцент робиться на навчанні персоналу правилам кібергігієни, впровадженні резервного копіювання даних та захисті від соціальної інженерії.

Технічні фактори передбачають регулярне технічне обслуговування та впровадження новітніх технологій, таких як використання штучного інтелекту для аналізу аномалій у роботі систем. Для теплових станцій це може включати сенсори моніторингу температури та автоматизовані системи аварійного відключення.

Економічні та політичні фактори включають залежність від постачальників, зміни у законодавстві чи політичну нестабільність. Для запобігання цим загрозам рекомендується впроваджувати диверсифікацію постачальників і забезпечувати відповідність систем міжнародним стандартам безпеки, наприклад, ISO/IEC 27001.

Організаційні фактори зосереджуються на розробці процедур та політик безпеки, а також управлінні ризиками. Для адміністративних центрів важливо впровадити централізовані платформи управління доступом і плануванням кризових ситуацій.

Завдяки багатофункціональності модулів, деякі з них можуть одночасно вирішувати завдання, пов'язані з кількома типами ризиків.



Рисунок 1.3 — Модель комплексного захисту критичної інфраструктури за факторами ризику

Ця модель демонструє, як різні фактори небезпек впливають на вибір модулів захисту, забезпечуючи адаптивний і цілісний підхід до захисту об'єктів КІ.

Висновки до першого розділу

У першому розділі було проведено огляд літературних джерел та аналіз сучасних підходів до кіберзахисту критичної інфраструктури. Розглянуто основні загрози, з якими стикаються об'єкти критичної інфраструктури, та визначено ключові аспекти, що потребують посиленого захисту.

РОЗДІЛ 2. РОЗРОБКА ПРОЄКТУ МОДУЛЬНОЇ СИСТЕМИ КІБЕРЗАХИСТУ

2.1. Опис модулів системи кіберзахисту

Система кіберзахисту критичної інфраструктури складається з низки незалежних модулів, кожен із яких виконує окрему функцію. Такий підхід дозволяє забезпечити гнучкість, адаптивність і надійність захисту, а також мінімізувати вплив потенційних загроз. Кожен модуль розроблено для виконання специфічних завдань. У цьому розділі детально розглянуто основні модулі системи кіберзахисту, їхні функції, технічні аспекти та можливі способи реалізації.

У цьому підрозділі ми детально розглянемо ключові модулі, що становлять основу модульної системи кіберзахисту, та їхню програмну реалізацію. Кожен модуль виконує конкретні функції, спрямовані на нейтралізацію певних факторів небезпеки та забезпечення цілісності, конфіденційності й доступності даних і систем.

Модульний підхід дозволяє створювати гнучкі та масштабовані рішення, які можуть адаптуватися до змінних загроз і потреб різних об'єктів критичної інфраструктури. Ми розглянемо опис функціональних можливостей кожного модуля, а також наведемо приклади програмного коду та сценарії інтеграції цих модулів у існуючі системи [17].

Модулі, які ми опишемо, включають диверсифікацію постачальників, відповідність стандартам безпеки, розробку процедур безпеки, управління ризиками, антидронові системи, моніторинг стану обладнання, апаратний захист КІ за допомогою IoT, систему IDS/IPS, криптозахист, інцидент-менеджмент, моніторинг та аудит, навчання персоналу, захист від соціальної інженерії, використання штучного інтелекту та резервне копіювання. Програмна реалізація кожного з них буде представлена детально, що дозволить зрозуміти, як ці рішення можуть бути впроваджені для забезпечення всебічного захисту критичної інфраструктури. Крім того, кожен модуль буде розглянуто з точки зору його адаптивності до змінних умов та потенційного масштабування в майбутньому.

Нижче розглянемо детально кожен модуль.

Модуль криптозахисту

Шифрування даних є важливим елементом інформаційної безпеки, особливо в критично важливих інфраструктурах. Використання надійних алгоритмів шифрування дозволяє захистити дані від несанкціонованого доступу та зберегти конфіденційність. Серед українських національних стандартів у галузі криптографії основними інструментами для захисту даних є криптографічні алгоритми «Калина» та «Купина». Також використовується для шифрування даних банківської таємниці. Серед сучасних методів шифрування набуває популярності гомоморфне шифрування та пост-квантове шифрування. Нижче розглянемо детальніше конкретні приклади та їх реалізації.

Гомоморфне шифрування є доволі новим і перспективним напрямком у криптографії, але наразі воно все ще знаходиться на стадії розробки та дослідження. Хоча воно має великий потенціал для захисту даних, виконання обчислень над зашифрованими даними без їх розшифровки, на сьогоднішній день воно не є широко розповсюдженим у практичних застосуваннях [18].

Основні сфери, де гомоморфне шифрування може бути корисним, включають обчислювальні сервіси в хмарі, аналіз даних з збереженням конфіденційності, захист даних у фінансових установах та охорону здоров'я. Проте, численні перешкоди, такі як висока складність реалізації та велика кількість ресурсів, що необхідно для обчислень над зашифрованими даними, затримують широке впровадження цього технологічного напрямку.

Згідно з останніми даними, Cisco активно досліджує та впроваджує різні методи криптографії, включаючи гомоморфне шифрування. Вони використовують його для покращення безпеки обчислень у хмарі та забезпечення конфіденційності даних. Однак, повне гомоморфне шифрування (FHE) залишається досить повільним для реального часу, тому його застосування ще не стало повсякденним.[19].

Гомоморфне шифрування класифікують як повністю гомоморфні системи та частково гомоморфні системи [20]. Проте за даними деякими джерел, частково гомоморфні системи класифікують за поколіннями [21]. Частково гомоморфні криптосистеми — це такі криптосистеми, які гомоморфні щодо лише однієї операції — або операції додавання, або операції множення. Доволі відома криптосистема RSA та криптосистема Ель-Гамала є частково гомоморфними. RSA є гомоморфною системою за множенням, а криптосистема Ель-Гамала має функцію гомоморфного шифрування щодо операції множення відкритих текстів [22].

Таблиця 2.1

Типи гомоморфного шифрування

Тип гомоморфного шифрування	Частково гомоморфне	Певною мірою гомоморфне	Рівневе повністю гомоморфне	Повністю гомоморфне
Оцінка	Просте	Середнє	Просунутий	Найбільш просунутий
Обчислення	Додавання або множення	Додавання або множення	Складні, але обмежені	Складні та необмежені
Застосування	Сума або добуток	Основний статистичний аналіз	AI/ML алгоритми, MPS	AI/ML алгоритми, MPS

Шифр Пайє

Розглянемо покроково криптосистему Пайє.

Генерація ключів. Для початку обирається для великих довільних простих числа. Для демонстрації обираються невеликі числа.

$$p = 7, q = 5$$

При цьому перевіряється умова найбільшого спільного дільника $pq, p - 1$ та $q - 1$ за формулою 2.1:

$$\gcd(pq, p - 1, q - 1) \quad (2.1)$$

Обчислюємо значення n та λ :

$$n = pq = 7 \cdot 5 = 35$$

$$\lambda = \text{НСК}(p - 1, q - 1) = \text{НСК}(6, 4) = 12$$

Додатково можна перевірити значення λ за формулою 2.2:

$$\lambda = ((p-1)(q-1))/(\text{НСД}(p-1, q-1)) \quad (2.2)$$

$$\lambda = ((7-1)(5-1))/(\text{НСД}(7-1, 5-1)) = 24/(\text{НСД}(6, 4)) = 12$$

Обираємо довільне число g та повинно задовольняти умову $g \in Z_{n^2}^*$

При цьому $Z_{(n^2)}^*$ це множина цілих чисел, взаємно простих з n^2 . Кількість ймовірних довільних чисел g має бути рівним $n\varphi(n)$.

Z_n^* це множина цілих чисел взаємно простих з n . Кількість ймовірних довільних чисел g має бути рівним $\varphi(n)$.

Z_n це множина цілих чисел по модулю з n .

Наприклад:

Якщо $n = 6$, то:

Z_n : 0, 1, 2, 3, 4, 5.

Z_n^* : 1, 5, бо значення $\text{НСД}(1, 6) = 1$ та $\text{НСД}(5, 6) = 1$. При цьому $\varphi(n) = 2$

$Z_{(n^2)}^*$: 1, 5, 7, 11, 13, 17, 19, 23, 29, 31, 33, 35. При цьому $n\varphi(n) = 6 \cdot 2 = 12$

Отже, обираємо значення $g = 3$

Визначаємо значення μ за формулою 2.3:

$$\mu = [L(g^\lambda \bmod n^2)]^{-1} \bmod n \quad (2.3)$$

$$\text{де } L(u) = \frac{u-1}{n}$$

$$\mu = (L(3^{12} \bmod 35^2))^{-1} \bmod 35$$

$$u = 3^{12} \bmod 1225 = 531441 \bmod 1225 = 1016$$

$$L(u) = \frac{u-1}{n} = \frac{1016-1}{35} = 29$$

$$\mu = 29^{-1} \bmod 35 = 29$$

Для визначення значення μ скористаємося розширеним методом Евкліда.

Застосуємо алгоритм Евкліда для знаходження $\text{НСД}(a, b)$, де $a = 29, b = 35$. Використаємо $a = bq + r$

$$35 = 29 \cdot 1 + 6$$

$$29 = 6 \cdot 4 + 5$$

$$6 = 5 \cdot 1 + 1$$

$$1 = 1 \cdot 1 + 0$$

Отже, НСД $(29, 35) = 1$, оскільки він є останнім ненульовим залишком.

Застосуємо розширений алгоритм Евкліда. Потрібно знайти таке число x та y , яке задовольняє рівняння:

$$29 \cdot x + 35 \cdot y = \text{НСД}(a, b) \Rightarrow 29 \cdot x + 35 \cdot y = 1$$

З останнього ненульового залишку маємо:

$$1 = 6 - 5 \cdot 1$$

Підставимо вираз для 5 з попереднього кроку:

$$1 = 6 - (29 - 6 \cdot 4) \cdot 1$$

$$1 = 6 - 29 + 6 \cdot 4$$

$$1 = 6 \cdot 1 - 29 + 6 \cdot 4$$

$$1 = 6 \cdot 5 - 29 \cdot 1$$

Підставимо вираз для 6:

$$1 = (35 - 29) \cdot 5 - 29 \cdot 1$$

$$1 = 35 \cdot 5 - 29 \cdot 6 \Rightarrow 29 \cdot (-6) + 35 \cdot 5 = 1$$

Це означає, що:

$$29 \cdot (-6) \equiv 1 \pmod{35}$$

Звідси, обернений елемент для 29 за модулем 35 є -6.

У модульній арифметиці ми не використовуємо від'ємні числа. Щоб отримати додатне значення, додамо модуль 35 до -6:

$$-6 + 35 = 29$$

Таким чином, обернене число до 29 за модулем 35 є 29.

Перевіримо:

$$29 \cdot 29 \equiv 1 \pmod{35}$$

Таким чином маємо:

Відкритий ключ: $(g, n) = (3, 35)$

Закритий ключ: $(\lambda, \mu) = (12, 29)$

Виконуємо шифрування:

Повідомлення $m = 8$. Обираємо довільне значення $r \in Z_n^*$: $r = 9$

За формулою 2.4 шифруємо повідомлення:

$$\begin{aligned}c &= g^m \times r^n \bmod n^2 \\c &= 3^8 \times 9^{35} \bmod 35^2 \\c &= 436 \times 949 \bmod 1225 = 939\end{aligned}\tag{2.4}$$

Виконуємо дешифрування:

Криптошифр $c = 939$, $c \in Z_{n^2}$

За формулою дешифруємо криптошифр:

$$\begin{aligned}m' &= L(c^\lambda \bmod n^2) \times \mu \bmod n \\m' &= L(939^{12} \bmod 12225) \times 29 \bmod 35 \\u &= 939^{12} \bmod 12225 = 771 \\L(u) &= \frac{u-1}{n} = \frac{771-1}{35} = 22 \\m' &= 22 \times 29 \bmod 35 = 8 = m\end{aligned}\tag{2.5)}$$

Розглянемо програмну реалізацію шифру Пайє, який дозволяє виконувати обчислення над зашифрованими даними без їх розшифрування, забезпечуючи максимальний рівень конфіденційності. Ця програма реалізована на мові програмування Python у середовищі розробки PyCharm і передбачає інтерактивний графічний інтерфейс для демонстрації роботи алгоритму. Основна мета програми — надати користувачам зрозумілий і наочний інструмент для вивчення основ гомоморфного шифрування.

Програма реалізує три основні функції: генерацію ключів, шифрування та дешифрування чисел. Покроковий опис дій, який можна увімкнути або вимкнути за бажанням, дозволяє легко зрозуміти кожен етап роботи алгоритму, що робить програму ідеальною для навчальних і демонстраційних цілей.

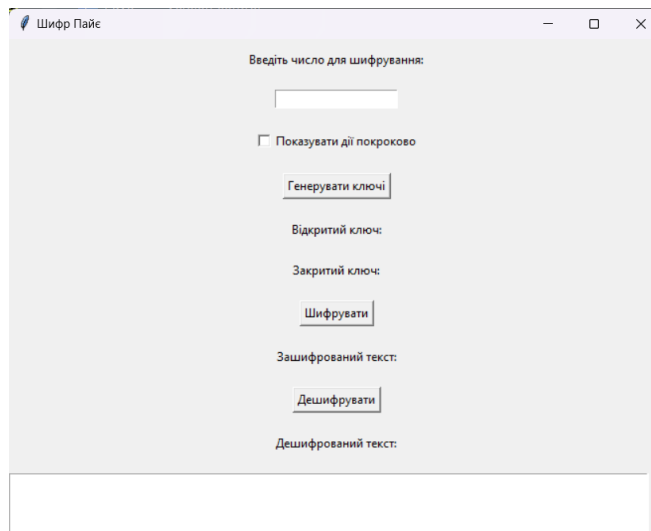


Рисунок 2.1 — Вікно програми

Державний стандарт шифрування «Калина»

Шифр «Калина» є національним криптографічним стандартом України, розробленим для забезпечення високого рівня безпеки даних. Цей шифр базується на сучасних математичних методах, що дозволяють ефективно захищати інформацію в умовах зростання кіберзагроз. Завдяки своїй архітектурі, «Калина» забезпечує стійкість до різних атак.

За деякими даними «Калина» є стійкими до атак, що базуються на квантових обчисленнях. «Калина» та «Купина» використовуються для шифрування інформації з обмеженим доступом, який містить гриф секретності «Для службового користування». Також цей тип шифрування використовується для шифрування даних банківської таємниці.

Розглянемо покроково криптосистему «Калина».

Генерація ключів. Для початку обирається ключ шифрування K . На основі цього ключа генерується допоміжний ключ K_t . На основі допоміжного ключа формуються раундові ключі [23].

В залежності від розміру блоку повідомлення (128, 256 або 512 біт) та довжини ключа (128, 256 або 512 біт) формується відповідна кількість раундових

ключів — 10, 14 або 18 раундів відповідно (Таблиця 2.2). Тобто, довжина ключа дорівнює або вдвічі перевищує розмір блоку.

Таблиця 2.2

Кількість раундів шифрування «Калина»

Розмір повідомлення	Довжина ключа та кількість раундів		
	128 біт ($N_k = 2$)	256 біт ($N_k = 4$)	512 біт ($N_k = 8$)
128 біт ($N_b = 2$)	10	14	-
256 біт ($N_b = 4$)	-	14	18
512 біт ($N_b = 8$)	-	-	18

Константна матриця стану, формується по формулі $N_b + N_k + 1$, де N_b — це кількість блоків даних, а N_k — кількість раундів ключів. У процесі шифрування після кожного кола застосовується ключ K , у межах якого виконуються дві основні операції: додавання за модулем 2^{64} або побітова операція XOR. Після трьох раундів в результаті маємо K_t .

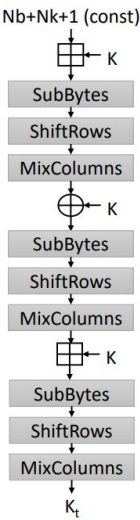


Рисунок 2.2. — Схема розробки ключа K_t

Після завершення трьох раундів шифрування формується проміжний ключ K_t . Цей ключ є основою для подальшого процесу генерації ключів. На основі початкового ключа та K_t відбувається формування ключів, індекс яких є парними.

Ключі, індекси яких не є парними, генеруються на основі вже сформованих парних ключів. У цьому процесі використовується метод ShiftPows, який забезпечує обертання та маніпуляцію бітами, що підвищує стійкість алгоритму до атак.

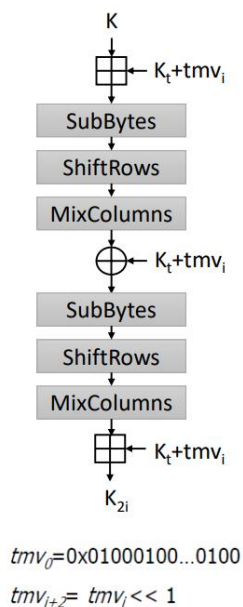


Рисунок 2.3. — Схема розробки ключів K_i

Шифрування:

На початку процесу відкритий текст розбивається на блоки фіксованого розміру, відповідно до параметра N_b . Кожен блок доповнюється до необхідної довжини, якщо це потрібно. Далі застосовується початковий раундовий ключ K_0 , який формується на основі стартового ключа.

Основні раунди шифрування (за винятком останнього) виконуються за наступною схемою: підстановка, зсув, перемішування та додавання ключа.

Підстановка: вхідний блок даних обробляється чотирма S-блоками (таблицями заміни), кожен із яких замінює байти блоку на нові значення, підвищуючи нелінійність.

Зсув рядків: рядки матриці стану циклічно зміщуються вправо на певну кількість байтів, залежно від їх положення та розміру блоку.

Перемішування стовпців: рядки й стовпці блоку змішуються шляхом множення на фіксовану матрицю в полі Галуа, забезпечуючи дифузію даних і ускладнюючи статистичний аналіз.

Додавання ключа: до блоку додається раундовий ключ шляхом операцій XOR або додавання за модулем 2^{64} в останньому раунді.

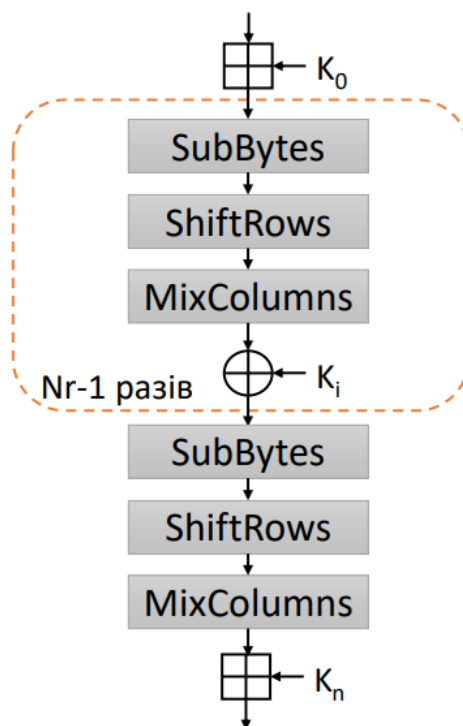


Рисунок 2.4. — Схема шифрування

Дешифрування:

Схема дешифрування відбувається дуже схоже на схему шифрування. Дешифрування використовує інверсні функції шифрування.

Основні раунди дешифрування (за винятком першого) виконуються за наступною схемою: додавання ключа, операція, зворотна до перемішування стовпців, зсув в іншому порядку та операція, обернена до підстановки.

XOR або віднімання по модулю 2^{64} для першого раунду: процес аналогічний додаванню. Якщо результат є від'ємним, то додається модуль 2^{64} для отримання додатного значення.

Підстановка байтів: вхідний блок даних обробляється чотирма S-блоками для дешифрування (таблицями заміни), кожен із яких замінює байти блоку на нові значення, підвищуючи нелінійність.

Зворотна операція до перемішування стовпців: стовпці множать на фіксований многочлен $s^{-1}(x)$, що є оберненим до $s(x)$.

Зсув рядків в зворотному порядку: рядки циклічно зсуваються вліво на задану кількість байтів.

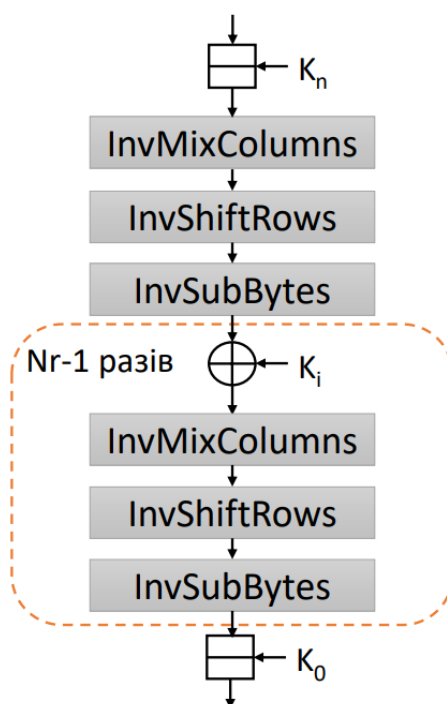


Рисунок 2.5. — Схема дешифрування

Розглянемо програмну реалізацію шифру «Калина», національного стандарту України, який забезпечує високий рівень криптографічного захисту інформації. Ця програма реалізована мовою програмування Python із використанням бібліотеки NumPy для матричних обчислень. Реалізація забезпечує підтримку основних функцій шифрування та дешифрування даних відповідно до специфікацій алгоритму.

Програма дозволяє виконувати генерацію ключів, шифрування та дешифрування текстових даних, а також надає змогу зрозуміти принципи роботи шифру через покрокову демонстрацію його основних етапів. Гнучкість архітектури

забезпечує легке налаштування параметрів, таких як розмір блоку даних і ключа, відповідно до потреб користувача.

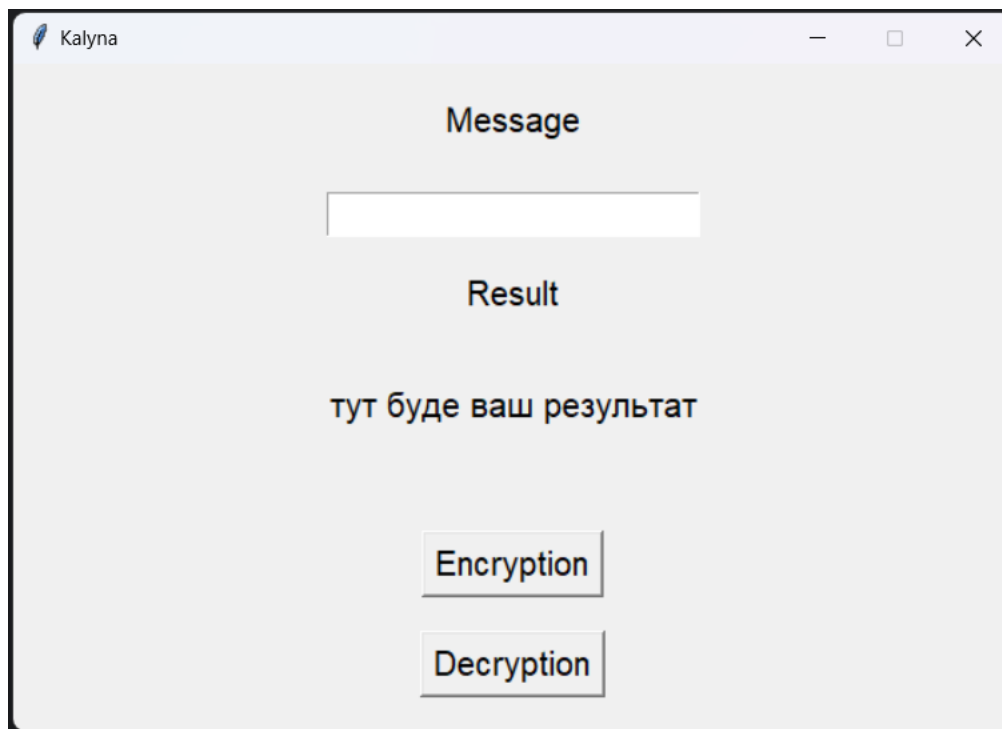


Рисунок 2.6 — Вікно програми

Державний стандарт хешування «Купина»

Хеш-функція «Купина» є національним криптографічним стандартом України, створеним для забезпечення високого рівня захисту даних [24]. Ця хеш-функція базується на сучасних математичних і криптографічних принципах, що гарантують ефективність та надійність при збереженні та передачі інформації. Завдяки своїй архітектурі, «Купина» забезпечує високу швидкість обчислення та стійкість до різних атак, включаючи колізії та криптоаналіз.

Особливістю «Купини» є її оптимізація для роботи в умовах сучасних викликів кібербезпеки, зокрема протидія квантовим обчисленням.

Функція відповідає найвищим стандартам криптографічного захисту та є невід'ємною складовою багатьох систем шифрування й автентифікації. Вона активно використовується для цифрового підпису, захисту паролів, створення контрольних сум та інших сфер, де необхідна висока надійність хешування.

Щодо стійкості до квантових обчислень, Купина, як і інші сучасні хеш-функції, не є абсолютно імунною до можливостей квантових комп'ютерів [25].

Однак, на сьогоднішній момент, квантові комп'ютери ще не досягли такого рівня розвитку, щоб здійснювати шифрування або розшифрування великих об'єктів даних швидко і ефективно [26].

Тобто, Купина залишається досить надійною хеш-функцією для більшості сучасних застосувань.

Розглянемо покроково хеш-функцію Купина.

Генерація хешу:

Для початку ведеться підготовка вхідного повідомлення. Початково обирається вхідне повідомлення, яке потрібно захешуватив. Для демонстрації обираємо невелике повідомлення.

Вхідне повідомлення: "Hello, World!"

Далі відбувається процес перетворення повідомлення. Вхідне повідомлення перетворюється у бінарний формат за допомогою ASCII-коди.

Бінарний формат: 01001000 01100101 01101100 01101100 01101111
00101100 00100000 01010111 01101111 01110010 01101100 01100100 00100001

Розбиття повідомлення на блоки. Повідомлення розбивається на блоки фіксованого розміру, які обробляються окремо. Наприклад, блоки по 512 бітів.

Паддінг (доповнення) повідомлення. Додаємо паддінг для вирівнювання повідомлення до необхідної довжини.

Ініціалізація стану. Встановлюється початковий стан хеш-функції.

Ініціалізація стану: $H_0, H_1, H_2, \dots, H_n$

Обробка блоків. Кожен блок повідомлення проходить через серію криптографічних перетворень, включаючи заміну, перестановку та змішування бітів. Це забезпечує стійкість до атак.

Остаточне значення хешу. Остаточне значення хешу представляє собою унікальну послідовність бітів фіксованої довжини, яка відповідає заданому повідомленню і слугує його цифровим відбитком."

Застосування хеш-функції Купина:

- Перевірка цілісності даних. Хеш-функція Купина використовується для створення контрольних сум, що дозволяють швидко перевіряти цілісність великих обсягів даних.
- Захист паролів. Хеш-функція часто застосовується для захисту паролів у базах даних. Замість збереження паролів у відкритому вигляді, зберігаються їх хеші, що унеможлиблює зломисникам дізнатися справжні паролі навіть у випадку доступу до бази даних.
- Цифровий підпис. Купина забезпечує надійний механізм для створення цифрових підписів, що гарантують автентичність і цілісність документів.
- Створення сертифікатів. Хеш-функція може використовуватися для створення сертифікатів, що підтверджують походження і невідомість документа.

Модуль IDS/IPS

Системи виявлення вторгнення (IDS) і запобігання вторгненню (IPS) є компонентами кібербезпеки важливих інфраструктур. За допомогою цієї системи організація може захистити свої мережі від кібератак і зменшити ризики за допомогою моніторингу, аналізу мережевого трафіку та виявлення загроз.

Основні функції IDS та IPS

Системи IDS/IPS аналізують вхідний і вихідний трафік у реальному часі, використовуючи різні методи для виявлення аномальної активності, як-от підозрілі шаблони трафіку або незвичайні поведінкові ознаки. IDS-системи орієнтовані на виявлення вторгнень через аналіз даних і порівняння їх із шаблонами атак. Вони можуть ідентифікувати ознаки кібератак, такі як сканування портів або спроби використання вразливостей. IPS-системи, крім виявлення, вживають заходів для запобігання вторгненням, наприклад, блокують підозрілий трафік, ізолюють пристрої або застосовують інші захисні дії.

Типи IDS та IPS систем

Мережеві системи (NIDS/NIPS) аналізують трафік у всій мережі, розташовуючись у критичних точках, через які проходить увесь трафік. Хостові системи (HIDS/HIPS) встановлюються на окремих пристроях для аналізу локального трафіку і поведінки, що дозволяє виявляти специфічні загрози. Гібридні системи поєднують обидва підходи, забезпечуючи комплексний захист.

Переваги та обмеження IDS/IPS систем

Впровадження IDS/IPS систем має ряд переваг, таких як підвищення рівня захисту мережі, виявлення загроз у реальному часі та зменшення ризиків компрометації. Однак, існують також обмеження, зокрема вимоги до обчислювальних ресурсів та можливість виникнення хибних спрацювань.



Рисунок 2.7 — Переваги та недоліки IDS/IPS систем

Впровадження IDS та IPS та їх інтеграція з іншими системами кібербезпеки

Перед впровадженням систем проводять оцінку потреб організації, визначаючи критичні точки мережі й можливі загрози. Встановлення й налаштування систем враховують специфіку мережі, конфігуруючи правила виявлення, моні-

торинг і заходи реагування на інциденти. Після впровадження забезпечується постійний моніторинг роботи систем та оновлення баз даних сигнатур атак, що підтримує їхню актуальність і ефективність.

IDS/IPS системи можна інтегрувати з іншими засобами кібербезпеки, такими як SIEM, антивірусні програми та фаєрволи. Це дозволяє забезпечити комплексний підхід до захисту мережі.

Реальні приклади впровадження

Наприклад, фінансова установа впровадила систему IDS для моніторингу транзакцій і виявлення підозрілих дій. У медичному закладі IPS система забезпечила захист від несанкціонованого доступу до медичних записів пацієнтів.

Майбутні тенденції та розвиток IDS/IPS систем

Використання штучного інтелекту та машинного навчання для підвищення ефективності IDS/IPS систем, автоматизації процесів та зменшення кількості хибних спрацювань.

Диверсифікація постачальників

Диверсифікація постачальників є важливою стратегією управління ризиками, яка може допомогти зменшити залежність від одного постачальника та забезпечити безперервність постачання критично важливих ресурсів і послуг. Наявність кількох надійних постачальників особливо важлива для захисту критичної інфраструктури, оскільки це значно зменшує вразливість до збоїв і загроз.

Однією з переваг диверсифікації є зменшення ризиків. Розподіл постачання між кількома постачальниками дозволяє уникнути залежності від одного джерела і забезпечує стабільність постачання, навіть якщо один з постачальників зазнає труднощів. Це мінімізує вплив можливих збоїв на діяльність організації. Диверсифікація також сприяє підвищенню конкуренції між постачальниками, що може позитивно вплинути на якість продукції та послуг. До того ж, вона забезпечує гнучкість у виборі альтернативних джерел у разі змін на ринку або введення санкцій. Таким чином, диверсифікація спрямована на збереження безперервності операцій та стійкості бізнес-процесів.



Рисунок 2.8 — Переваги та недоліки диверсифікації постачальників

Конкурентоспроможність також зростає завдяки диверсифікації, оскільки наявність кількох постачальників створює конкурентне середовище, що може призвести до зниження витрат і підвищення якості продукції або послуг. Гнучкість та адаптивність організації підвищується, оскільки можливість швидкого переходу до альтернативного постачальника у разі виникнення проблем дозволяє швидко реагувати на зміни на ринку та у середовищі.

Посилення безпеки є ще одним важливим аспектом диверсифікації постачальників. Використання кількох постачальників зменшує ризик компрометації постачання через кібератаки або інші загрози, оскільки потенційні нападники не можуть сфокусуватися на одному вразливому постачальнику.

Для успішної диверсифікації постачальників важливо проводити детальний аналіз та оцінку потенційних постачальників для визначення їх надійності, фінансової стабільності та відповідності вимогам безпеки. Розробка стратегій резервного постачання передбачає наявність альтернативних постачальників та запасів критичних ресурсів, що забезпечує додаткову стабільність.

Постійний моніторинг та контроль якості продукції та послуг від різних постачальників допомагають забезпечити відповідність встановленим стандартам та вимогам. Важливо також встановлювати та підтримувати довгострокові партнерства з кількома надійними постачальниками для забезпечення стабільності та передбачуваності постачання.

Диверсифікація постачальників є важливою стратегією для забезпечення безпеки критичної інфраструктури. Вона допомагає знизити ризики, підвищити конкурентоспроможність, забезпечити гнучкість та посилити загальний рівень безпеки. Використання різноманітних постачальників дозволяє організаціям бути краще підготовленими до можливих загроз та збоїв у постачанні, забезпечуючи безперервність роботи критичних систем.

Відповідність стандартам безпеки

Захист критичної інфраструктури залежить від підтримки міжнародних стандартів безпеки, таких як ISO/IEC 27001, NIST і CIS, серед інших. Ці стандарти забезпечують системний і структурований підхід до управління інформаційною безпекою з метою мінімізації ризиків.

Основні стандарти безпеки

- ISO/IEC 27001 ISO/IEC 27001 є міжнародним стандартом для систем управління інформаційною безпекою (ISMS). Він надає вимоги для встановлення, впровадження, підтримки та постійного вдосконалення системи управління безпекою інформації. Стандарт допомагає організаціям захищати свою інформацію за допомогою підходу, ґрунтованого на управлінні ризиками

- NIST (Національний інститут стандартів і технологій США) NIST розробляє та публікує спеціальні публікації з кібербезпеки, включаючи NIST SP 800-53, який забезпечує всеосяжний набір заходів контролю безпеки для інформаційних систем федерального уряду США. Ці рекомендації є корисними для різних організацій в усьому світі.

- CIS (Center for Internet Security) CIS Controls – це набір найбільш важливих дій, що забезпечують кращий захист від найбільш поширених та небезпечних кібератак. Вони включають конкретні заходи, такі як контроль доступу, управління конфігураціями та захист даних. Ці заходи спрямовані на підвищення кіберстійкості організацій та зниження ризиків шляхом впровадження перевірених практик безпеки.

Впровадження стандартів безпеки

Оцінка поточних систем. Перше, що потрібно зробити, це оцінити поточний стан інформаційних систем та їх відповідність вимогам стандартів безпеки. Це включає аудит інформаційної безпеки, тестування на проникнення та аналіз вразливостей.

Розробка політик та процедур. На основі результатів оцінки необхідно розробити або оновити політики та процедури безпеки, що відповідають стандартам. Це включає політики управління доступом, реагування на інциденти, управління ризиками та безперервності бізнесу.

Навчання персоналу є однією з окремих частин відповідності стандартам. Всі співробітники повинні знати і розуміти політики та процедури безпеки, а також бути підготовленими до реагування на інциденти.

Моніторинг та аудит. Після впровадження стандартів необхідно постійно моніторити стан систем безпеки та проводити регулярні аудити для перевірки відповідності. Це допомагає виявляти та усувати порушення безпеки вчасно.

Відповідність стандартам безпеки є основою ефективної стратегії кібербезпеки. Вона забезпечує систематичний підхід до захисту інформаційних систем, зменшує ризики та допомагає організаціям відповідати регуляторним вимогам.

Розробка процедур безпеки

Розробка процедур безпеки є важливим кроком у забезпеченні кібербезпеки критичної інфраструктури. Процедури безпеки встановлюють чіткі правила та кроки, які необхідно виконувати для захисту інформаційних систем і даних.

Хоча каркас процедур безпеки є схожим для різних типів підприємств, конкретні заходи та деталі можуть різнитися залежно від специфіки галузі та ризиків.

Основні етапи розробки процедур безпеки:

Оцінка ризиків. Перший крок у розробці процедур безпеки – це проведення оцінки ризиків. Це включає визначення потенційних загроз, вразливостей та наслідків можливих інцидентів. Оцінка ризиків допомагає пріоритезувати заходи безпеки.

Встановлення політик безпеки. На основі оцінки ризиків необхідно розробити політики безпеки, які включають загальні принципи та підходи до захисту інформаційних систем. Політики безпеки повинні бути зрозумілими та доступними для всіх співробітників організації

Розробка планів реагування на інциденти. Плани реагування на інциденти включають конкретні покрокові дії, які необхідно виконати в разі виявлення загрози або порушення безпеки. Це допоможе швидко реагувати на інциденти та мінімізувати їхні негативні наслідки.

Управління доступом. Важливо розробити процедури управління доступом, які включають правила надання та відкликання доступу до інформаційних систем. Це допомагає запобігти несанкціонованому доступу до конфіденційних даних.

Захист даних. Процедури захисту даних включають заходи шифрування, резервного копіювання та зберігання даних. Вони забезпечують конфіденційність, цілісність та доступність інформації.

Моніторинг та аудит. Необхідно встановити процедури постійного моніторингу та аудиту інформаційних систем. Це допомагає вчасно виявляти та реагувати на аномалії та потенційні загрози.

Навчання персоналу. Персонал повинен знати та розуміти політики та процедури безпеки, а також бути підготовленими до реагування на інциденти. Спеціальні тренінги та сертифікаційні програми, такі як провідний аудитор ISO/IEC 27001, CISSP або CISM, допомагають підвищити кваліфікацію персоналу.

Регулярний перегляд та оновлення. Процедури безпеки повинні регулярно переглядатися та оновлюватися з урахуванням нових загроз та змін в організаційній структурі. Це допомагає підтримувати ефективність заходів безпеки.

Для кожного типу підприємства процедури безпеки можуть різнитися. Наприклад, медичні заклади приділяють особливу увагу захисту персональних даних пацієнтів відповідно до вимог HIPAA, тоді як фінансові установи зосереджуються на захисті фінансової інформації та відповідності вимогам PCI DSS. Незважаючи на ці відмінності, каркас процедур безпеки залишається схожим і включає основні етапи, такі як оцінка ризиків, управління доступом та моніторинг.

Управління ризиками

Управління ризиками допомагає ідентифікувати, оцінювати та управляти потенційними загрозами для критичної інфраструктури. Впровадження ефективних методів управління ризиками дозволяє мінімізувати негативні наслідки інцидентів безпеки та забезпечити безперебійну роботу організації.

Ідентифікація ризиків визначає потенційні загрози і вразливості. Оцінка ризиків включає їхню ймовірність і вплив. Розробка стратегії управління ризиками вибирає методи для зменшення ризиків, такі як уникнення, передача дії іншому виконавцю або пом'якшення. Впровадження заходів передбачає оновлення програмного забезпечення, навчання персоналу та моніторинг. Моніторинг і аналіз включає постійний контроль ризиків і оцінку ефективності заходів.

Методи управління ризиками відрізняються залежно від типу підприємства. Фінансові установи захищають конфіденційні дані клієнтів, тоді як промислові підприємства зосереджуються на безпеці виробничих процесів.

Приклади у різних сферах КІ

Енергетичні системи впроваджують резервне живлення та зміцнюють фізичну безпеку об'єктів. Медичні заклади впроваджують багатofакторну аутентифікацію та шифрування даних.

Фінансові установи використовують аналітичні системи для виявлення підозрілих транзакцій. Транспортні системи включають резервні системи та навчання персоналу.

Промислові підприємства встановлюють системи моніторингу та розробляють плани реагування.

Моніторинг стану обладнання

Системи моніторингу дозволяють вчасно виявляти та усувати потенційні проблеми, забезпечуючи безперебійне функціонування обладнання та знижуючи ризики. Датчики та сенсори забезпечують постійний збір даних про стан обладнання. Ці дані включають інформацію про температуру, вологість, вібрації та інші параметри, які можуть вказувати на потенційні проблеми. Спеціалізовані системи збору даних об'єднують інформацію, отриману від датчиків, і передають її до центральної системи для подальшого аналізу. Центральна система моніторингу обробляє зібрані дані, аналізує їх та відображає інформацію у зручному для користувача форматі. Це дозволяє операторам швидко виявляти аномалії або відхилення від норми та приймати відповідні рішення. Системи оповіщення сповіщає операторів у разі виявлення проблем. Це може включати SMS, електронну пошту або push-сповіщення. Моніторинг стану обладнання має ряд переваг. Регулярний моніторинг дозволяє виявляти потенційні проблеми на ранніх стадіях, що запобігає непередбаченим поломкам та збоям. Дані, отримані з систем моніторингу, допомагають планувати технічне обслуговування на основі фактичного стану обладнання, що знижує витрати та підвищує ефективність. Моніторинг також дозволяє вчасно реагувати на потенційні загрози, зменшуючи ризики для критичної інфраструктури та підвищуючи загальний рівень безпеки. Автоматизація процесів моніторингу зменшує потребу в ручному контролі, що дозволяє зекономити час та ресурси.

Системи моніторингу можна інтегрувати з іншими системами управління, такими як ERP, що забезпечує централізоване управління та обмін даними. Також можливе поєднання моніторингу стану обладнання з системами фізичної

безпеки та кібербезпеки для забезпечення комплексного захисту інфраструктури. Дані з систем моніторингу можуть бути використані для аналітики та прогнозування, що дозволяє підвищити точність планування та прийняття рішень.

Приклади використання моніторингу стану обладнання включають енергетичні системи, де моніторинг стану трансформаторів та генераторів дозволяє забезпечити безперебійне постачання енергії та запобігти аваріям. У промислових підприємствах моніторинг стану виробничого обладнання допомагає запобігти простоям та забезпечити високу якість продукції. У транспортних системах моніторинг стану рухомого складу та інфраструктури допомагає забезпечити безпеку та ефективність транспортних операцій.

Моніторинг стану обладнання продовжує розвиватися з використанням новітніх технологій, таких як штучний інтелект для прогнозової аналітики та інтеграція з блокчейн для забезпечення незмінності даних. Ці інновації сприяють підвищенню ефективності та точності моніторингу, що є важливим для зниження ризиків та забезпечення надійності критичної інфраструктури.

Технології та платформи для моніторингу стану обладнання значною мірою впливають на те, наскільки добре функціонує управління промисловими інфраструктурами. Розгляймо на порівняльну таблицю, яка показує основні функціональні можливості, переваги та застосування провідних технологій і платформ, які широко застосовуються в різних галузях.

Таблиця 2.3

Порівняльна таблиця технологій та платформ для моніторингу стану обладнання

Технологія/Платформа	Функціональність	Переваги	Приклади використання
Azure IoT	Платформа для збору, аналізу та управління даними IoT	Широкі можливості інтеграції, аналітика в реальному часі, масштабованість	Енергетика, промисловість, транспорт
AWS IoT	Платформа для підключення, керування та захисту IoT	Висока безпека, масштабованість, інтеграція з іншими сервісами AWS	Промислові підприємства, сільське господарство

Google Cloud IoT	Платформа для аналізу IoT даних та управління пристроями	Глибока аналітика, штучний інтелект, машинне навчання, масштабованість	Розумні міста, охорона здоров'я, транспорт
SCADA	Система для збору та контролю даних з промислових систем	Контроль у реальному часі, висока надійність, інтеграція з промисловими системами	Промислові підприємства, енергетика, водопостачання
OPC UA	Стандарт зв'язку для промислових систем	Безпека, інтероперабельність, масштабованість	Промислові підприємства, автоматизація виробництва
Splunk	Платформа для збору та аналізу машинних даних	Глибока аналітика, візуалізація даних, інтеграція з різними джерелами даних	ІТ-безпека, моніторинг інфраструктури, управління подіями
IBM Watson IoT	Платформа для аналізу IoT даних з використанням AI	Штучний інтелект, машинне навчання, глибока аналітика	Розумні будинки, промислові підприємства, охорона здоров'я

Загалом, моніторинг стану обладнання є невід'ємною частиною забезпечення безпеки та ефективного управління критичною інфраструктурою. Він дозволяє підвищити надійність обладнання, оптимізувати обслуговування, знизити ризики та економити ресурси. Інтеграція з іншими системами забезпечує комплексний підхід до управління та безпеки.

Апаратний захист КІ за допомогою IoT

Інтернет речей (IoT) став важливою складовою сучасних систем захисту критичної інфраструктури. Завдяки можливості постійного моніторингу та управління, IoT-пристрої дозволяють підвищити рівень безпеки та забезпечити безперебійну роботу важливих об'єктів, таких як енергетичні системи, водопостачання, транспортні мережі та промислові об'єкти.

Основні елементи технічного захисту IoT

Шифрування даних забезпечує конфіденційність та цілісність інформації, що передається між IoT-пристроями та центральними системами управління, використовуючи алгоритми шифрування, такі як AES, та протоколи безпеки, як-от TLS або VPN.

Для запобігання несанкціонованому доступу, користувачі та пристрої повинні бути авторизовані та аутентифіковані. Використання OAuth 2.0 та токенів безпеки гарантує надійну аутентифікацію.

Захист мережі включає використання фаєрволів для фільтрації трафіку та запобігання несанкціонованим запитам. Віртуальні локальні мережі (VLAN) ізолюють IoT-пристрої від інших частин мережі, знижуючи ризик компрометації.

Оновлення програмного забезпечення запобігає використанню вразливостей через автоматичні механізми оновлення, що впроваджують патчі безпеки.

Моніторинг та виявлення загроз включає постійний аналіз трафіку та поведінки IoT-пристроїв. Використання систем виявлення аномалій, таких як Splunk та Darktrace, допомагає ідентифікувати підозрілу активність та швидко реагувати на інциденти.

Виявлення та реагування на аномалії в IoT-системах

Моніторинг та виявлення аномалій забезпечує постійний аналіз трафіку та поведінки пристроїв для виявлення відхилень від норми. Використання спеціалізованого програмного забезпечення для виявлення аномалій дозволяє швидко реагувати на загрози через автоматичне блокування підозрілої активності та ізоляцію уражених пристроїв.

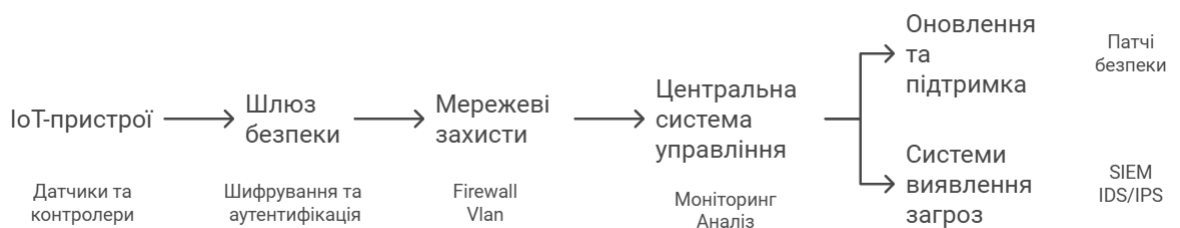


Рисунок 2.9 — Схема апаратного захисту критичної інфраструктури за допомогою IoT

Інцидент-менеджмент

Інцидент-менеджмент є важливим компонентом стратегії кібербезпеки, орієнтованим на виявлення, аналіз та реагування на інциденти безпеки. Управління інцидентами допомагає організаціям мінімізувати вплив інцидентів на бізнес-процеси, забезпечити безперервність бізнесу та знизити ризики.

Нижче розглянемо основні етапи.

Основні етапи

Виявлення інциденту включає моніторинг систем, аналіз логів та використання засобів виявлення загроз. Виявлені інциденти реєструються в системі управління інцидентами для подальшого аналізу. Детальний аналіз інциденту дозволяє визначити причини, обсяг та наслідки.

Реагування на інцидент включає нейтралізацію загрози, відновлення систем та застосування засобів захисту. Після нейтралізації загрози проводиться відновлення постраждалих систем і даних. Після завершення реагування документуються всі дії, пов'язані з інцидентом, та створюються звіти. Регулярний аналіз інцидентів допомагає вдосконалювати процедури безпеки.

Для оцінки ефективності інцидент-менеджменту важливо збирати метрики, такі як середній час виявлення інциденту (MTTD) та середній час реагування (MTTR).

Процедури інцидент-менеджменту можуть варіюватися залежно від типу та розміру підприємства, що вимагає різних підходів та ресурсів.

Сучасні SIEM-системи дозволяють інтегрувати виявлення, аналіз та реагування на інциденти в єдину платформу, а штучний інтелект та машинне навчання допомагають передбачати можливі інциденти та виявляти аномалії.

Використання аналітичних інструментів, таких як Splunk, Kibana або Elasticsearch, дозволяє проводити глибокий аналіз даних журналів та кореляцію подій.

Інцидент-менеджмент в системах КІ

Енергетичні системи. Енергетичні системи зосереджені на забезпеченні безперебійного постачання електроенергії. Вони мають складну мережеву інфраструктуру та велику кількість обладнання, що потребує постійного моніторингу та захисту.

Процедури: енергетичні компанії використовують системи моніторингу в реальному часі для контролю стану обладнання та мереж. Інциденти, такі як перебої в постачанні електроенергії або кібератаки на системи управління, потребують негайного реагування та використання резервних джерел живлення для мінімізації перебоїв.

Медичні заклади. Медичні заклади зберігають критично важливі медичні дані, що потребують захисту відповідно до регуляторних стандартів, таких як HIPAA.

Процедури: резервне копіювання медичних даних виконується щоденно, а доступ до систем захищений багатофакторною аутентифікацією. У разі інциденту, наприклад, витоку даних або атаки з використанням шкідливого ПЗ, проводиться ізоляція уражених систем, відновлення даних та посилення мережевого захисту. Також регулярно проводяться аудити безпеки та навчання персоналу для підвищення готовності до реагування на кіберзагрози.

Моніторинг та аудит

Моніторинг стану обладнання є важливим елементом кібербезпеки та ефективного управління критичною інфраструктурою. Він забезпечує своєчасне виявлення та усунення потенційних проблем, дозволяючи підтримувати безперебійне функціонування обладнання і мінімізувати ризики.

Основною складовою систем моніторингу є датчики та сенсори, які забезпечують постійний збір даних про стан обладнання. Ці дані включають параметри, такі як температура, вологість, вібрація та інші показники, що дозволяють виявляти потенційні проблеми на ранніх стадіях.

Спеціалізовані системи збору даних інтегрують інформацію з різних сенсорів і передають її до центральної системи моніторингу. Центральна система

аналізує отримані дані, виявляє аномалії або відхилення від норми, а також виводить інформацію у зручному форматі для оператора. Завдяки цьому оператори можуть швидко приймати рішення щодо усунення виявлених проблем.

Важливу роль у моніторингу відіграють системи оповіщення. Вони сповіщають операторів про виявлення несправностей за допомогою SMS, електронної пошти або push-сповіщень, забезпечуючи швидку реакцію на потенційні загрози.

Моніторинг стану обладнання має низку переваг.

Він дозволяє запобігати несподіваним поломкам та збоям шляхом виявлення проблем на ранніх стадіях. Отримані дані допомагають оптимізувати планування технічного обслуговування, знижуючи витрати та підвищуючи ефективність.

Крім того, своєчасна реакція на потенційні загрози підвищує загальний рівень безпеки критичної інфраструктури. Автоматизація моніторингу зменшує потребу у ручному контролі, що економить час і ресурси.

Системи моніторингу легко інтегруються з іншими системами управління, такими як ERP, що забезпечує централізоване управління та обмін даними. Вони також можуть працювати у поєднанні із системами фізичної безпеки та кібербезпеки для забезпечення комплексного захисту інфраструктури.

Дані з моніторингових систем можуть використовуватись для аналітики та прогнозування, що покращує точність планування і прийняття рішень.

Практичне застосування моніторингу стану обладнання охоплює різні галузі. Наприклад, у енергетичному секторі моніторинг трансформаторів і генераторів дозволяє забезпечити безперебійне постачання енергії та уникнення аварій.

У промисловості контроль стану виробничого обладнання допомагає уникати поломки пристроїв і підтримувати якість продукції.

У транспортних системах моніторинг інфраструктури та рухомого складу забезпечує безпеку і ефективність роботи.

Таким чином, моніторинг стану обладнання є необхідним для забезпечення безпеки і надійності критичної інфраструктури.

Він дозволяє підвищити ефективність управління, оптимізувати технічне обслуговування, зменшити ризики і витрати. Інтеграція з іншими системами забезпечує комплексний підхід до управління та захисту критично важливих об'єктів.

Навчання персоналу

Критична інфраструктура є основою функціонування сучасного суспільства та економіки. З огляду на зростаючі ризики кібератак, терористичних загроз та стихійних лих, забезпечення безпеки об'єктів критичної інфраструктури потребує кваліфікованого та підготовленого персоналу.

Навчання персоналу має першорядне значення, оскільки людський фактор часто відіграє вирішальну роль у запобіганні аварійних ситуацій. Навіть найзахищеніші системи можуть бути частково або повністю зруйновані через погано реалізовані протоколи поведінки співробітників.

Ефективне навчання гарантує, що співробітники не тільки розуміють потенційні загрози, а й здатні своєчасно реагувати на них, використовуючи відповідні технології та алгоритми поведінки, розроблені або зазначені в допоміжних матеріалах.

Основна мета навчання — підвищення кваліфікації персоналу з виявлення, запобігання та усунення загроз.

Крім того, важливо використовувати отримані навички на практиці та підтримувати свої знання, коли виникнуть нові загрози.

Регулярне оновлення навчальних програм відповідно до сучасних викликів забезпечує актуальність знань і підготовку до роботи в динамічному середовищі кібербезпеки.

Основними задачами навчання є:

- Розвиток навичок ідентифікації ризиків, оцінки їх рівня та планування заходів безпеки.

- Практичне відпрацювання сценаріїв реагування на надзвичайні ситуації, включаючи інциденти кібербезпеки.
- Формування у співробітників культури безпеки, що передбачає відповідальне ставлення до захисту.

Сучасне навчання персоналу у сфері безпеки критичної інфраструктури включає використання інтерактивних методик та інструментів, які забезпечують максимальну ефективність.

Серед них, тренінги з моделювання надзвичайних ситуацій і відпрацювання алгоритмів дій, онлайн-курси та вебінари, що дозволяють оперативно оновлювати знання про нові загрози та технології, навчальні симуляції для відпрацювання реагування на техногенні, природні чи кіберзагрози, та регулярне тестування для перевірки знань і навичок співробітників.

Найбільш ефективними є саме комплекси таких методів навчання, а не вузьконаправлений вибір лише одного методу.

Також, найкраще використовувати можливості навчання від тих компаній, які безпосередньо стикаються з практичними викликами кібератак.

Наприклад, це можуть бути курси Cisco Networking Academy від компанії Cisco.

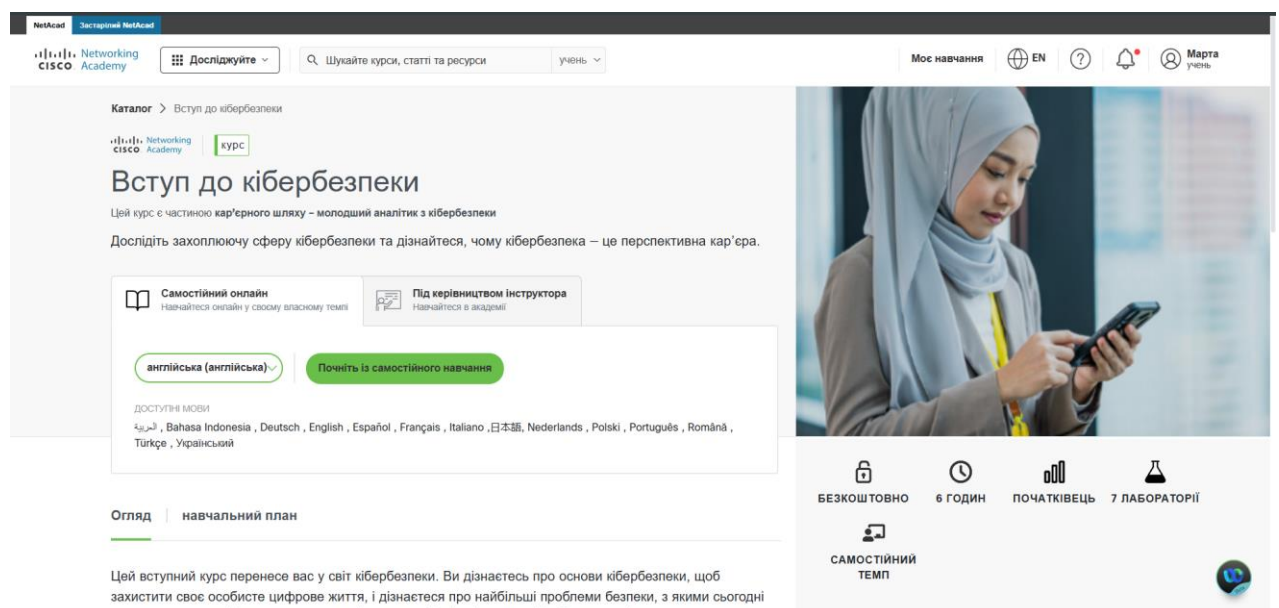


Рисунок 2.10 — Курси на платформі Cisco Networking Academy

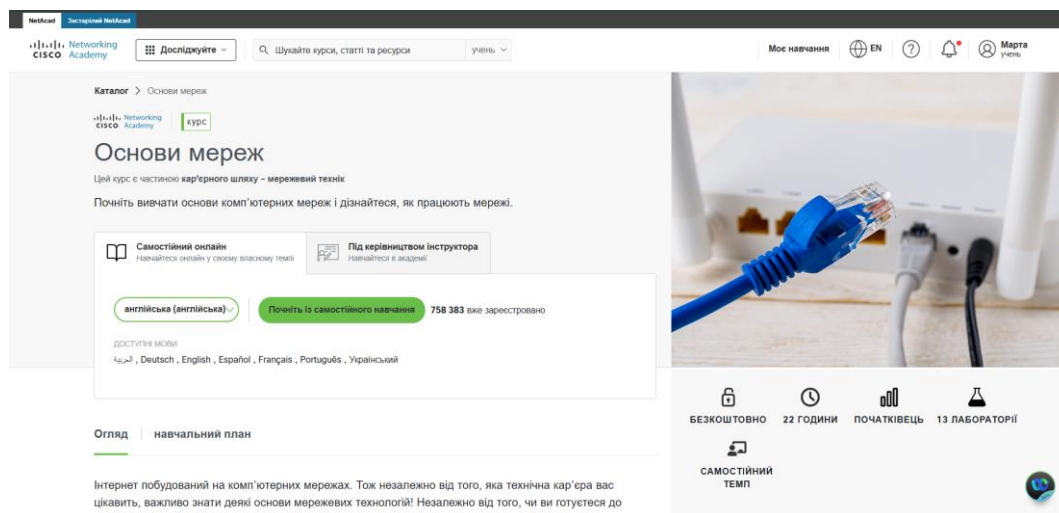


Рисунок 2.11 — Курси на платформі Cisco Networking Academy

Компанія Cisco пропонує курси, які допомагають спеціалістам отримати необхідні знання та навички для ефективного виявлення, запобігання та ліквідації загроз.

Деякі курси Cisco є безкоштовними, що дозволяє персоналу підвищувати свою кваліфікацію без додаткових витрат. Більшість курсів можна проходити самостійно, без інструктора, що надає гнучкість у навчанні. Вони забезпечують сучасні знання та допомагають покращити поточні навички персоналу.

Практичні знання мають важливе значення для спеціалістів у сфері кібербезпеки. Cisco також пропонує програму Cisco Packet Tracer, де можна віртуально навчатися працювати на обладнанні, що дозволяє відпрацьовувати різні сценарії та підвищувати практичні навички. Це допомагає персоналу бути готовим до будь-яких викликів та забезпечувати надійний захист критичної інфраструктури. Окрім того, на платформі діє постійна форма для зворотнього зв'язку, яка дозволяє покращувати та оновлювати курси.

Навчання персоналу має важливе значення для забезпечення безпеки критичної інфраструктури. Приклади реальних подій свідчать про те, що правильне навчання може запобігти великим небезпекам і зберегти стабільність бізнесу.

Окрім теоретичних знань, особливу увагу слід приділяти практичним тренуванням, що імітують реальні загрози та сценарії атак. Це дозволяє персоналу ефективно реагувати на інциденти та знижувати ризики для інфраструктури.

Захист від соціальної інженерії

Методи несанкціонованого доступу до інформації класифікуються як з використанням методів соціальної інженерії, так і без них. У першому випадку зловмисник використовує соціологію та психологію для отримання конфіденційних даних, тоді як у іншому випадку він повинен мати досвід у сфері ІТ.

Такі характеристики людської психіки, як когнітивні упередження, є психологічними підґрунтям для використання методів соціальної інженерії. Через це надійність комп'ютерної системи є не вищою, ніж надійність її оператора. Зловмисники проникають навіть у добре спроектовані, захищені комп'ютерні системи, скориставшись неуважністю довірених користувачів або умисно вводячи їх в оману (наприклад, відрекомендувавшись системним адміністратором або амбасадором комерційного бренду, надсилають повідомлення із запитом паролів).

Кібератаки включають у себе впровадження шкідливого коду в код веб-сайтів або використання шкідливих програм (вірусів, троянів тощо). Атаки такого типу перешкоджають керуванню або ремонту пошкодженого продукту. Що стосується соціальної інженерії, цей тип атаки спрямований на користувачів комп'ютерної системи, а не безпосередньо на комп'ютерну систему.

У сфері соціальної інженерії цей тип атаки спрямований на користувачів — «найуразливішої ланки» комп'ютера, а не на систему безпосередньо. Він досягає тих самих результатів, обходячи інфраструктуру, яка захищає комп'ютери від шкідливих програм.

Оскільки такі прийоми значно складніше виявити чи запобігти їм, цей напрям атак є набагато ефективнішим за інші. Такі атаки вимагають більше часу, але менших ресурсів і досліджень. Не завжди вдається одразу знайти людину, яка готова надати доступ зсередини (свідомо чи ні), іноді перші спроби виявля-

ються скомпрометованими і на певний час провокується збільшення уваги до даного методу атаки. Але рано чи пізно людський фактор, неуважність чи втома дозволять знайти вразливе місце.

Основна тактика соціальної інженерії полягає в тому, щоб переконати користувачів розкрити інформацію про особистий характер (наприклад, паролі, доступи тощо) за допомогою психологічних методів.

Іноді навіть вплив на власні переконання і політичні погляди людини можна використати для отримання доступів у організацію зсередини.

Показовим прикладом вдалого застосування соціальної інженерії можна вважати атаку на найбільшого телефонного оператора в Україні. Атака відбулася 12 грудня 2023 року, близько 24 мільйонів користувачів «Київстар» не мали доступу до Інтернету та м'обільного зв'язку по всій країні, при цьому користувачі не могли і використати альтернативні засоби зв'язку — приєднатися до мереж інших операторів у рамках внутрішньоукраїнського роумінгу [27]. Мета атаки полягала в центральній мережі, яка є ядром мережі та відповідає за обробку та маршрутизацію трафіку між користувачами та службами.

Усі системи, пов'язані з цим оператором, зіткнулися з проблемами. Так, у багатьох містах необхідно було вручну включати та відключати вуличне освітлення. Частина банкоматів і платіжних терміналів українських банків припинила роботу. Райффайзен Банк, Ощадбанк і Приватбанк заявили про серйозні проблеми. Інтернет-магазини, Нова пошта та деякі сервіси, такі як [Tabletki.ua](https://tabletki.ua) та [Glovo](https://glovo.ua), зіткнулися з проблемами. Уряд України також повідомив про збої в роботі охоронних систем і гарячих ліній.

Хакери зуміли не лише зламати центр управління мережею «Київстар», але й «знести» конфігурацію на транзитних базових станціях. Тобто відновлення мало бути повноцінним і довготривалим. Повне відновлення зайняло у компанії дев'ять днів.

Під час термінового розслідування виявилось, що хакери зламали комп'ютерний захист компанії за допомогою скомпрометованого облікового запису одного із співробітників «Київстар».

Відомо, що компанія "Київстар" з моменту повномасштабного вторгнення, на момент цієї атаки відбила понад 500 середніх та великих хакерських атак, але саме проникнення зсередини стало найбільш ефективним.

Саме тому дуже важливо використовувати і методи боротьби з соціальною інженерією.

Зважаючи на варіативність обраних методів соціоінженерного підходу та форм маніпулятивного впливу, виділяються такі напрями протидії – тестування на проникнення; моделювання дій об'єкта та суб'єкта соціоінженерного впливу; метод виявлення і повідомлення про атаки соціальної інженерії людиною; метод протидії використанню соціальної інженерії.

Тестування на проникнення використовує автоматизовану систему захисту від тестових атак на користувачів. Для цього використовується набір інструментів соціального інженера (Social-Engineer Toolkit, SET). Ці засоби дозволяють на практиці забезпечити правдоподібність атак соціальної інженерії, надають варіативність атак [28].

Метод моделювання дій об'єкта та суб'єкта соціоінженерного впливу можна назвати "оберненим" до атаки. У даному випадку досліджується мета та способи кожної сторони, моделюється актуальна реакція на надану відповідь. Тобто, нападник аналізує дані, які прийшли від жертви з метою отримання якомога більш точної, детальної, критичної інформації. В той час як жертва намагається передбачити наступний крок нападника і протидіяти. Дана вправа має бути динамічною і виконаною в обох ролях, для того, щоб потенційна жертва бачила вплив аналогічних дій на нападника і розуміла найбільш та найменш ефективні засоби боротьби.

Метод виявлення та повідомлення про атаки соціальної інженерії, що здійснюються людиною, є найпростішим. Достатньо одного повідомлення від користувача, щоб виявити її, відповідно, повідомити про випадок атаки соціальної інженерії [29].

Такий підхід дозволяє залучати працівників організації не тільки для профілактики за допомогою, для прикладу, кібергігієни, а й для активного повідомлення про загрози кібербезпеці у реальних випадках.

Структура захисту соціальної інженерії SEDF (Social Engineering Defensive Framework) описує метод запобігання використанню соціальної інженерії. (Social Engineering Defensive Framework). На основі його використання взято наступні етапи:

- Визначення соціоінженерного впливу
- Оцінка можливості протидії
- Навчання персоналу
- Організація поточних політик і технологій

Підсумовуючи, можна сказати, що протидіяти соціальній інженерії передбачає дуже багато вимог для контролюючого відділу. Одними з головних є ефективне навчання персоналу, актуалізація навчальних матеріалів, постійне тестування персоналу наближеними до реальних випадками, спонукання персоналу до комунікацій за найменшої ознаки втручання, актуальні технічні засоби для автоматизації перевірок (за необхідності — додатково налаштованими), замовлення незалежних перевірок та консультацій.

Модуль «Штучний інтелект»

Сучасні системи кіберзахисту критичної інфраструктури базуються на модулях штучного інтелекту (ШІ). Його основним призначенням є автоматизація обробки, аналізу та реагування на кіберзагрози за допомогою прикладних технологій машинного та глибокого навчання.

Цей модуль автоматизує процеси реагування на інциденти, прогнозує загрози та моніторить активність мережі.

Модуль ІІІ виконує ряд важливих функцій, включаючи:

- Автоматизація процесів виявлення загроз.
- Аналіз та класифікація кіберзагроз.
- Прогнозування потенційних атак.
- Автоматизоване реагування на інциденти в реальному часі.
- Постійне навчання моделей для адаптації до нових типів загроз.

Основні компоненти модуля

Збір та агрегація даних: Дані для аналізу збираються з різних джерел, таких як мережеві журнали, лог-файли систем моніторингу, антивірусні програми, дані IoT пристроїв і хмарних сервісів [30]. Після збору дані нормалізуються та агрегуються для уніфікації, що забезпечує їх готовність до обробки моделями ІІІ.

Аналіз даних: на етапі аналізу проводиться попередня обробка даних: очищення від шуму, заповнення пропусків та видалення зайвих даних. Дані перетворюються у формат, придатний для аналізу, зокрема виконуються генерація ознак (фічей), кодування змінних та нормалізація.

Навчання моделей ІІІ: навчання моделей базується на історичних даних із використанням як супервізованих, так і несупервізованих методів. Для виконання завдань застосовуються різні алгоритми, такі як нейронні мережі, дерева рішень, кластеризація тощо. Постійне оновлення моделей гарантує їхню адаптацію до змін у кіберсередовищі.

Виявлення загроз та автоматизоване реагування: модуль аналізує мережеву активність у реальному часі, виявляючи аномалії, які можуть сигналізувати про загрозу. Виявлені загрози класифікуються за рівнем ризику, а система автоматично застосовує відповідні заходи: блокування IP-адрес, ізоляцію підозрілих систем або зміну політик безпеки.

Інтеграція з іншими модулями: ІІІ інтегрується з модулями моніторингу, аудиту, резервного копіювання та відновлення для забезпечення комплексного

захисту. Це дає змогу зменшити кількість помилкових спрацьовувань і покращити реакцію на загрози.

Переваги та недоліки використання ШІ

Переваги:

- Адаптивне навчання: модуль здатний постійно навчатися на нових даних та адаптуватися до змін у кіберзагрозах, що дозволяє забезпечити ефективність навіть у динамічних середовищах.
- Миттєва реакція: система забезпечує оперативне виявлення та реагування на загрози у реальному часі, що мінімізує потенційні збитки.
- Прогнозний аналіз: завдяки аналізу історичних даних, модуль може прогнозувати можливі атаки, дозволяючи заздалегідь вживати необхідні заходи.
- Протидія фішинговим атакам: ШІ автоматично аналізує електронну пошту, виявляє потенційно шкідливі листи та блокує їх, забезпечуючи додатковий рівень захисту.
- Захист від DDoS-атак: система виявляє та блокує аномальний мережевий трафік, що дозволяє зменшити навантаження на інфраструктуру і запобігти зупинці роботи.

Недоліки:

- Можливі хибнопозитивні результати: у деяких випадках модуль може видавати помилкові тривоги, що може створювати додаткове навантаження на персонал.
- Висока вартість впровадження: інтеграція ШІ в системи кіберзахисту вимагає значних фінансових ресурсів для розробки, налаштування та підтримки.
- Залежність від якості даних: ефективність роботи модуля прямо залежить від якості вхідних даних. Неповні або некоректні дані можуть знижувати точність прогнозів і виявлення загроз.
- Проблеми конфіденційності: аналіз великої кількості даних може створювати ризики для конфіденційності, якщо система не має достатніх засобів захисту.

- Необхідність регулярних оновлень: для збереження актуальності та ефективності модуль потребує постійного оновлення моделей і алгоритмів відповідно до нових загроз і змін у кіберсередовищі.



Рисунок 2.12 — Переваги та недоліки модулю ШІ в захисті КІ

Тривалий термін роботи і постійне навчання ШІ може зменшити кількість помилок і підвищувати точність та ефективність ШІ у забезпеченні кіберзахисту.

Розглянемо вже готові спеціалізовані варіанти ШІ для кібербезпеки, які можуть бути інтегровані.

1. Darktrace. Darktrace використовує технологію Enterprise Immune System, яка імітує роботу імунної системи людини. Вона застосовує алгоритми машинного навчання для виявлення аномалій у реальному часі. Система автоматично аналізує мережевий трафік, поведінку користувачів і пристроїв, і реагує на підозрілі активності.

2. Cylance. Cylance використовує штучний інтелект для прогнозування загроз до їх появи. Її технологія базується на аналізі великих обсягів даних та виявленні патернів, що вказують на потенційні загрози. Cylance забезпечує захист кінцевих точок та інших пристроїв.

3. Vectra AI. Vectra AI використовує штучний інтелект для автоматизованого виявлення та реагування на загрози. Система аналізує поведінку користувачів і пристроїв, щоб виявити аномальні активності, які можуть свідчити про потенційні загрози.

4. FireEye. FireEye є комплексним рішенням для виявлення, реагування та розслідування інцидентів кібербезпеки. Вона поєднує у собі інструменти для захисту від передових постійних загроз (APT) та інших типів атак. Система постійно оновлюється для протидії новим загрозам.

Таблиця 2.4

Порівняльна таблиця спеціалізованих ШІ для кібербезпеки

Функції	Переваги	Недоліки
Darktrace	Технологія Enterprise Immune System автоматично виявляє та реагує на аномалії Система постійно аналізує мережевий трафік та поведінку користувачів Автоматично самостійно реагує на загрози, мінімізуючи шкоду.	Висока вартість для малих підприємств Є доволі складною системою, що потребує фахівців для налаштування та підтримки
CyLance	Використовує штучний інтелект для прогнозування загроз перед їх появою. Працює з мінімальними ресурсами, не впливаючи на продуктивність систем. Легко інтегрується з існуючими системами безпеки.	Має обмежений набір функцій, і може не покривати всі аспекти Для деяких функцій потрібен постійний доступ до мережі.
Vectra AI	Використовує ШІ для автоматизованого виявлення та реагування на загрози. Аналізує поведінку користувачів та пристроїв для виявлення аномалій. Легко інтегрується з іншими системами безпеки.	Доволі дороговартісний для малих підприємств Складний для налаштування та оптимізації
FireEye	Комплексний підхід поєднує інструменти для виявлення, реагування та розслідування інцидентів. Захищає від передових постійних загроз APT Система постійно оновлюється для протидії новим загрозам.	Доволі дороговартісний Можливі труднощі з інтеграцією з іншими системами

Ці спеціалізовані рішення ШІ для кібербезпеки пропонують конкретні можливості для захисту критичної інфраструктури. Вибір найбільш підходящого рішення залежить від конкретних потреб вашої організації, доступних ресурсів

та вимог до безпеки. Тривалий термін роботи і постійне навчання ШІ може зменшити кількість помилок і підвищувати точність та ефективність ШІ у забезпеченні кіберзахисту.

Резервне копіювання

Для захисту важливої інфраструктури резервне копіювання зберігає дані та дозволяє їх відновити у разі втрати, пошкодження або кібератаки.

Процедури резервного копіювання включають кілька основних етапів. Планування резервного копіювання передбачає визначення частоти та часу проведення резервного копіювання, враховуючи критичність даних та вимоги до їх доступності. Автоматизація процесу резервного копіювання за допомогою спеціального програмного забезпечення забезпечує регулярність виконання процедур і знижує ризик людської помилки. Вибір місця зберігання для резервних копій є критичним аспектом. Резервні копії можуть зберігатися на віддалених серверах, у хмарних сховищах або на окремих фізичних носіях, що забезпечує додатковий рівень захисту.

Захист резервних копій включає використання методів шифрування для запобігання несанкціонованому доступу. Важливо обмежити доступ до резервних копій лише уповноваженому персоналу. Регулярне тестування резервних копій шляхом перевірки працездатності відновлення даних допомагає переконатися у надійності та готовності резервних копій до використання у випадку надзвичайної ситуації.

Таблиця 2.5

Приклади застосування у різних типах критичної інфраструктури

Тип КІ	Коли?	Чому?
Енергетичні системи	Резервне копіювання даних повинно проводитися регулярно, наприклад, щодня або щотижня.	Втрата даних про стан обладнання, операційні процеси чи технічне обслуговування може призвести до серйозних перебоїв у постачанні електроенергії.
Медичні заклади	Резервне копіювання повинно проводитися щодня, враховуючи постійне оновлення медичних записів.	Втрата критично важливих даних про пацієнтів може мати серйозні наслідки для здоров'я пацієнтів та їхнього лікування.

Фінансові установи	Резервне копіювання повинно виконуватися кілька разів на день.	Втрата даних про фінансові транзакції та конфіденційні дані клієнтів може призвести до фінансових втрат та репутаційних ризиків.
Транспортні системи	Резервне копіювання даних має проводитися щоденно або навіть частіше.	Втрата даних про маршрути, розклади руху та технічний стан транспорту може спричинити затримки та підвищити ризик аварій.
Промислові підприємства	Резервне копіювання має виконуватися щоденно, особливо перед важливими операціями або оновленнями.	Втрата даних може призвести до зупинки виробництва, втрати продукції та порушення виробничих ланцюгів.

Модуль антидронових систем

Сучасні виклики в галузі безпеки об'єктів критичної інфраструктури сприяли розвитку автоматизованих систем повітряної охорони, які базуються на безпілотних літальних комплексах (БЛК). Ці системи розроблені для виконання ключових функцій спостереження, контролю, виявлення та блокування несанкціонованих повітряних об'єктів у межах контрольованих зон.

Розвиток технологій, зокрема безпілотних і роботизованих систем у поєднанні зі штучним інтелектом, забезпечує можливість виконання складних задач із мінімальним залученням людських ресурсів. Це дозволяє значно зменшити витрати часу та ресурсів, одночасно охоплюючи великі території, як на земній поверхні, так і у повітрі.

Автоматизована система безпеки включає центральний вузол управління, сенсори, розташовані на об'єктах та БЛК, щонайменше два безпілотні літальні апарати, комбіновану систему передачі даних та віддалений сервер. Центральний вузол управління інтегрований у глобальну систему збирання та обробки інформації (ГСЗОІ), яка моніторить переміщення повітряних об'єктів на охоронюваній території. Він виконує аналіз, обробку та комбінування даних, отриманих із ГСЗОІ, сенсорів, встановлених на об'єктах та БЛК.

Центральний вузол також здійснює керування мінімум двома безпілотними літальними апаратами, які знаходяться в автономних місцях стоянки з по-

садковими зонами. Один із БЛК оснащений модулем блокування роботи несанкціонованих повітряних об'єктів, тоді як інший виконує спостереження за допомогою відповідних сенсорів і має модуль, який здатен змінювати курс польоту виявлених об'єктів. Обидва дрони здатні обмінюватися даними між собою та з центральним вузлом управління.

Система може збирати інформацію про повітряні об'єкти як у межах охоронюваної території, так і поблизу неї, використовуючи сторонні інформаційні вузли ГСЗОІ та власні сенсори. Вона аналізує такі параметри, як радіочастотний, звуковий, тепловий та візуальний сліди, доповнюючи ці дані інформацією з фото- та відеофіксації.

Літальні апарати БЛК перебувають у режимі очікування на місцях стоянки, доки не отримають сигнал від центрального вузла управління про виявлення несанкціонованих об'єктів.

Вони оснащені інтегрованими зарядними пристроями та системами збору й передачі інформації, що доповнені елементами штучного інтелекту.

За командою центрального вузла БЛК можуть працювати як автономно, використовуючи власні засоби збору інформації, так і в координації з іншими компонентами системи. Вони виконують виявлення, супровід та перехоплення несанкціонованих повітряних об'єктів.

Центральний вузол обчислює координати положення, траєкторію руху, висоту та швидкість об'єкта і передає ці дані до БЛК для подальшого супроводу стороннього об'єкту або його перехоплення. За необхідності оператор може вносити корективи в керування БЛК, змінювати їх курс, а також активувати та коректувати роботу модулів блокування та відхилення

Реалізація цієї моделі дозволяє досягти високої ефективності у виявленні та блокуванні несанкціонованих повітряних об'єктів. Система є інтегрованою у глобальну інфраструктуру збирання та обробки інформації, що забезпечує оперативність і надійність у реагуванні на надзвичайні ситуації.

Детальний опис патенту наведено в Додатку А.

Сценарій використання

Уявімо сценарій використання автоматизованої системи безпеки під час виявлення несанкціонованого БПЛА в зоні критичної інфраструктури: Розглянемо покроково кожен етап реагування: від виявлення об'єкту до його нейтралізації.

1. Детекція загрози. Центральний вузол управління аналізує інформацію про повітряні об'єкти як у межах охоронюваної території, так і поблизу неї, використовуючи сторонні інформаційні вузли ГСЗОІ. В той же час сенсори, розташовані на об'єкті та дронах, фіксують радіочастотний сигнал, характерний для роботи дронів. Одночасно теплові та акустичні сенсори визначають присутність об'єкта, підтверджуючи загрозу. Зібрана інформація передається для аналізу до центрального вузла управління.

2. Аналіз і прийняття рішення. Центральний вузол обробляє отримані дані, використовуючи алгоритми штучного інтелекту. Визначається тип загрози, траєкторія польоту несанкціонованого БПЛА та можливі варіанти нейтралізації. У разі підтвердження загрози активується сценарій реагування.

3. Активізація дронів. Центральний вузол надсилає команди на автономні місця стоянки дронів. Дрон-блокатор піднімається в повітря для виконання задачі нейтралізації. Одночасно дрон-спостерігач забезпечує супровід і контроль за діями несанкціонованого об'єкта. При цьому обидва дрони забезпечують трансляцію керуючих сигналів від центрального вузла до дронів, а також трансляцію відеосигналу до оператора.

4. Нейтралізація загрози. Дрон, обладнаний модулем відхилення, за прорахованим центральним вузлом маршрутом, вилітає на перехоплення за допомогою спеціальних засобів глушить радіочастотний сигнал, розриваючи зв'язок між несанкціонованим дроном та його оператором.

5. У разі ситуаційної необхідності задіюється другий дрон, обладнаний модулем блокування для забезпечення фізичного перешкоджання польоту стороннього об'єкта, наприклад, відстрілу сітки для захоплення або спеціалізоване обладнання для знищення об'єкта.

6. Завершення операції. Після усунення загрози дрони повертаються на місце стоянки для заряджання та підготовки до подальшого використання. Вся інформація про інцидент зберігається в центральному вузлі для подальшого аналізу та вдосконалення системи.

Процес реагування на загрози БПЛА



Рисунок 2.13 — Сценарій використання

Опис архітектури системи

Автоматизована система повітряної безпеки є комплексним рішенням, що забезпечує багаторівневий захист об'єктів критичної інфраструктури. Вона поєднує сучасні технології виявлення, аналізу та нейтралізації загроз.

Архітектура системи включає наступні ключові компоненти:

1. Центральний вузол управління. Це основний елемент системи, що здійснює збір, обробку й аналіз інформації з усіх сенсорів і безпілотних літальних комплексів (БЛК). Центральний вузол інтегрований у глобальну систему збору та обробки інформації (ГСЗОІ), що забезпечує доступ до даних про переміщення повітряних об'єктів у зоні відповідальності. Завдяки використанню штучного інтелекту вузол може аналізувати комплексні дані в реальному часі, прогнозувати траєкторії руху повітряних об'єктів і приймати рішення про активізацію засобів нейтралізації.

2. Системи сенсорів. Сенсори розташовані як на об'єкті, що охороняється, так і безпосередньо на самих БЛК. Радіочастотні датчики виявляють сигнали дронів. Теплові сенсори реагують на теплові відбитки двигунів дронів. Акустичні сенсори здатні фіксувати звуки роботи пропелерів. Оптичні сенсори забезпечують візуальне розпізнавання та ідентифікацію об'єктів.

3. Безпілотні літальні комплекси (БЛК). До складу системи входять щонайменше два БЛК:

- Дрон-блокатор, обладнаний модулем для фізичного блокування роботи несанкціонованих повітряних об'єктів.
- Дрон-спостерігач, обладнаний модулем для електронного блокування роботи несанкціонованих повітряних об'єктів, оснащений сенсорами для моніторингу та супроводу несанкціонованих об'єктів та системою ретрансляції сигналів.

4. Комбінована система передачі даних. Інформація передається через дротові та бездротові канали зв'язку, що забезпечує безперебійний обмін даними між компонентами системи навіть у разі збоїв.

5. Автономні місця стоянки. Місця стоянки дронів забезпечені зарядними пристроями та системами передачі даних. Це дозволяє підтримувати безперервну готовність літальних засобів до виконання задач.



Рисунок 2.14 — Опис архітектури системи

Після аналізу кожного модуля стає очевидним, що модулі системи кіберзахисту мають тісне взаємозв'язок і часто доповнюють функціональність одного. Така інтеграція модулів дозволяє створити багаторівневу та ефективну систему захисту, де кожен компонент не тільки виконує свої основні функції, але підтримує інші модулі для досягнення максимальної безпеки критичної інфраструктури.

2.2. Фактори небезпеки та відповідні модулі для їх нейтралізації

Фактори небезпеки, які можуть вплинути на критичну інфраструктуру, включають кібератаки, природні катастрофи, технічні збої, соціальну інженерію та шкідливих інсайдерів. Кібератаки мають на меті крадіжку даних, злам систем або поширення шкідливого програмного забезпечення. Природні катастрофи, такі як землетруси, буревії та повені, можуть спричинити пошкодження інфраструктури та втрату даних.

Технічні збої можуть виникати через несправності обладнання або програмного забезпечення, що призводить до перебоїв у роботі систем. Соціальна інженерія використовує людський фактор для отримання доступу до конфіденційної інформації або систем. Шкідливі інсайдери - це співробітники, які зловмисно використовують свій доступ для пошкодження систем або викрадення даних.

Для нейтралізації цих факторів небезпеки можуть бути використані різні модулі. Система IDS/IPS забезпечує виявлення та запобігання вторгненням, криптозахист шифрує дані для запобігання несанкціонованому доступу, а ІШ допомагає прогнозувати та виявляти аномальну активність. Резервне копіювання дозволяє зберігати дані для їхнього відновлення після катастроф, а моніторинг стану обладнання допомагає виявляти проблеми та запобігати збоєм. Моніторинг та аудит забезпечують постійний контроль за системами та виявлення проблем, а резервне копіювання допомагає відновити дані після технічних збоїв. Навчання персоналу підвищує обізнаність щодо методів соціальної інженерії, а заходи захисту від соціальної інженерії запобігають фішинг-атакам та іншим методам. Відповідність стандартам безпеки включає впровадження політик доступу, а інцидент-менеджмент забезпечує швидке реагування на інциденти та нейтралізацію загроз. Управління ризиками враховує різні джерела загроз та мінімізує залежність від одного постачальника. Антидронові системи захищають від загроз, які можуть бути використані дронами для збору інформації або пошкодження інфраструктури. Використання IoT-пристроїв для моніторингу та управління інфраструктурою забезпечує захист через шифрування та авторизацію.

Таблиця 2.6

Фактори небезпеки та відповідні модулі для їх нейтралізації

Фактори небезпеки	Модулі для нейтралізації
Кібератаки	- Система IDS/IPS - Криптозахист - Штучний інтелект
Природні катастрофи	- Резервне копіювання - Моніторинг стану обладнання
Технічні збої	- Моніторинг та аудит - Резервне копіювання
Соціальна інженерія	- Навчання персоналу - Захист від соціальної інженерії
Шкідливі інсайдери	- Відповідність стандартам безпеки - Інцидент-менеджмент
Диверсифікація постачальників	- Управління ризиками
Антидронові системи	- Захист від дронів
Апаратний захист КІ за допомогою IoT	- Використання IoT-пристроїв - Шифрування та авторизація

2.3. Програмна реалізація

Програмна архітектура шифру Пайє

Програма реалізує три основні функції: генерацію ключів, шифрування та дешифрування даних. У процесі генерації ключів використовуються великі прості числа, які перевіряються на взаємну простоту. Це дозволяє створити відкритий і закритий ключі, які зберігаються та виводяться для подальшого використання.

Шифрування відбувається шляхом введення користувачем повідомлення (числа), яке зашифровується за допомогою відкритого ключа. Алгоритм використовує складні математичні операції та випадкові числа для забезпечення високої стійкості шифрування. Дешифрування виконується із застосуванням закритого ключа, що дозволяє відновити початкове повідомлення із зашифрованих даних. У цьому процесі використовуються функції для обчислення обернених значень та модульної інверсії.

Користувач взаємодіє з програмою через графічний інтерфейс. Він вводить повідомлення, яке потрібно зашифрувати, і має змогу увімкнути режим покрокового опису дій. Це дозволяє спостерігати всі етапи роботи алгоритму у вигляді текстових пояснень. Генерація ключів, шифрування та дешифрування активуються відповідними кнопками, після чого результати відображаються на екрані та в текстовому полі.

Приклад роботи програми шифру Пайє

Програма демонструє генерацію ключів, використовуючи прості числа, шифрування тексту шляхом піднесення до ступеня та модульних операцій, а також дешифрування через обчислення обернених значень. Наприклад, користувач вводить число "5", яке шифрується у вигляді "34892". Після дешифрування програма відновлює початкове число "5", підтверджуючи коректність роботи алгоритму. Таким чином, програма наочно ілюструє основні принципи роботи асиметричного шифрування та забезпечує перевірку його ефективності. Цей процес

дозволяє користувачам зрозуміти, як працюють криптографічні алгоритми, забезпечуючи надійний захист даних у цифровому середовищі.

Опис роботи коду шифру Пайє

Реалізація виконана на мові програмування Python з використанням бібліотеки `tkinter` для створення графічного інтерфейсу та бібліотеки `random` для генерації випадкових чисел.

Функція `get_prime(bits)`: генерує випадкове просте число з заданою кількістю бітів та використовує функцію `random.getrandbits(bits)` для генерації випадкового числа і перевіряє його простоту за допомогою функції `is_prime(n)`.

```
2 usages
7  def get_prime(bits):
8      while True:
9          num = random.getrandbits(bits)
10         if is_prime(num):
11             return num
12
```

Рисунок 2.15 — Функція `get_prime(bits)`

Функція `extended_gcd(a, b)`: реалізує розширений алгоритм Евкліда для обчислення найбільшого спільного дільника (НСД) двох чисел та коефіцієнтів, які задовольняють рівняння Безу.

```
1 usage
23 def extended_gcd(a, b):
24     x0, x1, y0, y1 = 1, 0, 0, 1
25     while b != 0:
26         q, a, b = a // b, b, a % b
27         x0, x1 = x1, x0 - q * x1
28         y0, y1 = y1, y0 - q * y1
29     return a, x0, y0
30
```

Рисунок 2.16 — Функція `extended_gcd(a, b)`

Функція `mod_inverse(a, n)`: обчислює обернене число за модулем n за допомогою розширеного алгоритму Евкліда.

```
1 usage
32 def mod_inverse(a, n):
33     gcd, x, y = extended_gcd(a, n)
34     if gcd != 1:
35         raise Exception('Модульний обернений елемент не існує')
36     else:
37         return x % n
38
```

Рисунок 2.17 — `mod_inverse(a, n)`

Функція `generate_keys(bits=5)`: генерує пару простих чисел p і q . Обчислює модуль n , значення g , λ_n , u , L_u та m_u .

Функція `encrypt(public_key, plaintext)`: виконує шифрування вхідного числа `plaintext` за допомогою відкритого ключа `public_key`. Використовує випадкове число r для шифрування.

Функція `decrypt(public_key, private_key, ciphertext)`: виконує дешифрування зашифрованого тексту `ciphertext` за допомогою відкритого ключа `public_key` та закритого ключа `private_key`.

Однією з проблем при реалізації шифру Пайє є обмежені можливості комп'ютера для генерації великих чисел. Це може призвести до помилок при генерації ключів, обчисленні обернених чисел за модулем та виконанні інших криптографічних операцій, що вимагають великих чисел. Через ці обмеження виникають ситуації, коли НСД чисел не є 1, і це унеможлиблює обчислення обернених чисел, що призводить до помилок в алгоритмі.

Це може призвести до помилок в алгоритмі. У випадку виникнення помилки або неправильної роботи програми рекомендується її перезапустити для повторного виконання обчислень.

```

1 usage
103 def decrypt(public_key, private_key, ciphertext):
104     n = public_key[1]
105     lambda_n = private_key[0]
106     mu = private_key[1]
107     u = (ciphertext ** lambda_n) % (n ** 2)
108     Lu = (u - 1) // n
109     plaintext = (Lu * mu) % (n * 2)
110     if toggle_var.get():
111         append_output("Дешифрування...")
112         append_output(f"m'={L(c^lambda_n mod n^2)*(mu mod n)}:")
113         append_output(f"1) c^lambda_n mod n^2: {ciphertext}^{lambda_n} mod {n ** 2} = {u}")
114         append_output(f"2) L(u)=(u-1)/n: ({u}-1/{n})={Lu}")
115         append_output(f"3) m' = {Lu}*{mu} mod {n}={plaintext}")
116         append_output(f"Розшифрований текст = {plaintext}")
117     return plaintext
118

```

Рисунок 2.18 — decrypt(public_key, private_key, ciphertext)

Лістинг програмного коду буде наведено в Додатку Б

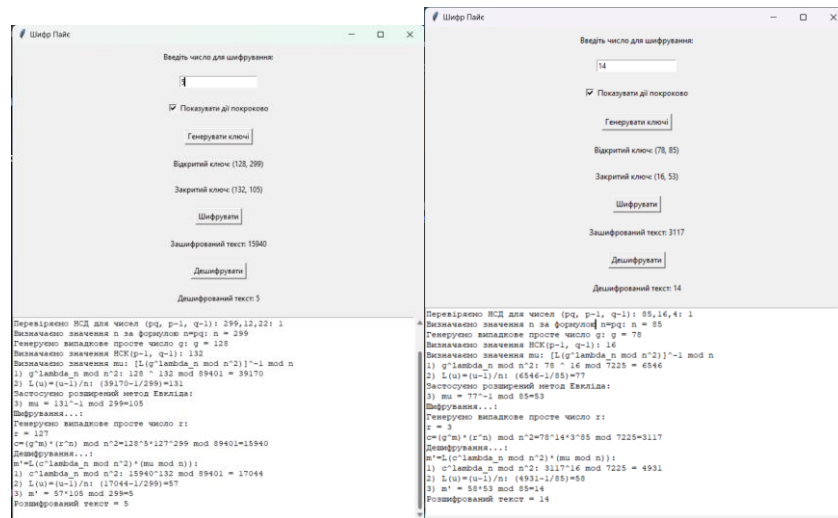


Рисунок 2.19 — Приклад роботи програми

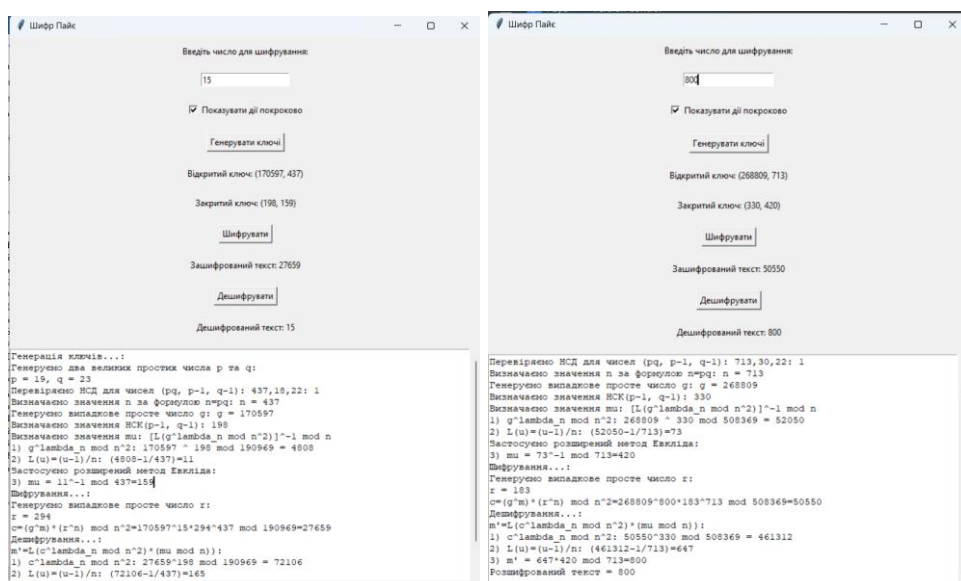


Рисунок 2.20 — Приклад роботи програми

```

Генерація ключів...:
Генеруємо два великих простих числа p та q:
p = 19, q = 23
Перевіряємо НСД для чисел (pq, p-1, q-1): 437,18,22: 1
Визначаємо значення n за формулою n=pq: n = 437
Генеруємо випадкове просте число g: g = 170597
Визначаємо значення НСК(p-1, q-1): 198
Визначаємо значення mu: [L(g^lambda_n mod n^2)]^-1 mod n
1) g^lambda_n mod n^2: 170597^198 mod 190969 = 4808
2) L(u)=(u-1)/n: (4808-1/437)=11
Застосуємо розширений метод Евкліда:
3) mu = 11^-1 mod 437=159
Шифрування...:
Генеруємо випадкове просте число r:
r = 294
c=(g^r*m) mod n^2=170597^15*294^437 mod 190969=27659
Дешифрування...:
m'=L(c^lambda_n mod n^2)*(mu mod n):
1) c^lambda_n mod n^2: 27659^198 mod 190969 = 72106
2) L(u)=(u-1)/n: (72106-1/437)=165
3) m' = 165*159 mod 437=15
Розшифрований текст = 15

```

Рисунок 2.21— Приклад роботи програми

Крім того, для демонстрації можливостей гомоморфного шифрування реалізовано додаткову програму, яка демонструє процес додавання зашифрованих чисел. Користувач вводить два числа для шифрування, після чого система генерує відкритий і закритий ключі. Обидва числа шифруються незалежно, а потім сума зашифрованих значень обчислюється без необхідності їх розшифрування. Цей результат можна дешифрувати, щоб отримати суму вихідних чисел. Такий підхід дозволяє наочно показати переваги гомоморфного шифрування, зокрема можливість обчислень із зашифрованими даними без розкриття конфіденційної інформації. Інтерфейс програми інтуїтивно зрозумілий і підтримує покроковий режим пояснення дій, що робить її корисною для навчальних і демонстраційних цілей.

Лістинг програмного коду буде наведено в Додатку В.

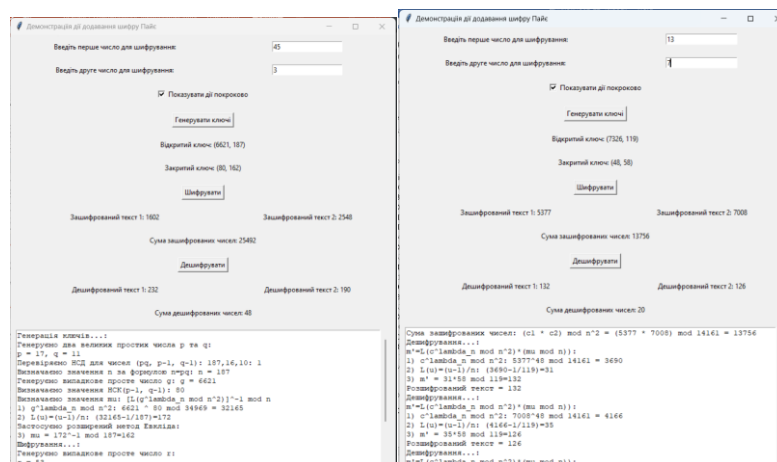


Рисунок 2.22 — Приклад роботи програми

Програмна архітектура шифру “Калина”

Програма реалізує три основні функції: генерацію раундових ключів, шифрування та дешифрування даних. У процесі генерації ключів формується початкова матриця стану, яка базується на розмірах блоку даних (N_b) та ключа (N_k). Далі, початковий ключ (K_0) обчислюється через додавання за модулем 2^{64} і використовується для формування раундових ключів на кожному етапі шифрування.

Шифрування починається з розбиття вхідного повідомлення на блоки фіксованого розміру. Для кожного блоку виконується початкове додавання раундового ключа. Кожен раунд складається з підстановки байтів через S-блоки, циклічного зсуву рядків, перемішування стовпців із використанням матриці у полі Гаула, а також додавання відповідного раундового ключа. У фінальному раунді замість перемішування стовпців застосовується лише додавання ключа.

Дешифрування виконується у зворотному порядку з використанням обернених операцій. На кожному етапі застосовуються інверсні S-блоки, обернений зсув рядків і зворотнє перемішування стовпців, що дозволяє відновити вихідний текст із зашифрованих даних.

Користувач взаємодіє з програмою через текстовий інтерфейс. Він може задати вхідні дані повідомлення для шифрування або дешифрування, обрати параметри блоку і ключа, а також активувати покроковий режим для спостереження за всіма етапами роботи алгоритму. Результати обчислень на кожному етапі відображаються у вигляді тексту або матриць, що дозволяє користувачам краще зрозуміти принципи роботи шифру «Калина».

Приклад роботи програми шифру “Калина”

Програма демонструє генерацію раундових ключів, використовуючи початковий ключ і матрицю стану, шифрування тексту через послідовність операцій, таких як підстановка, циклічний зсув рядків, перемішування стовпців та додавання за модулем 2^{64} , а також дешифрування за допомогою обернених перетворень. Наприклад, користувач вводить текст "Привіт", який шифрується у вигляді

блоку даних, представленого у шістнадцятковій системі, наприклад, "7E45A6F0...". Після дешифрування програма відновлює вихідний текст "Привіт", що підтверджує правильність роботи алгоритму.

Опис роботи коду шифру “Калина”

Реалізація виконана на мові програмування Python з використанням бібліотеки tkinter для створення графічного інтерфейсу, бібліотек numpy для роботи з матрицями та для обробки рядків.

Функція ContBytes(list): розбиває вхідну строку на окремі байти (групи по два символи). Забезпечує підготовку тексту до шифрування.

```
31 # розділення стрічки по байтам
5 usages
32 def ContBytes(list):
33     reinitiallist = re.findall( pattern: r'[0-9A-Za-z]{2}', list) # ділимо по два символи (буква або ж число)
34     return reinitiallist
35
```

Рисунок 2.23 — ContBytes(list)

Функція ConvertToHex(arr): конвертує масив чисел у шістнадцятковий формат. Використовується для представлення даних у форматі, зручному для криптографічних операцій.

```
6 # перетворення даних в 16кову сч # для строк
5 usages
7 def ConvertToHex(arr):
8     # print(f'arr={arr}')
9     hexResult = [[0x00] * int(len(arr) / 8) for i in range(8)]
10    k = 0
11    # print(f'int(len(arr)/8)={int(len(arr)/8)}')
12    for j in range(int(len(arr) / 8)): # 2
13        for i in range(8): # 8
14            hexResult[i][j] = hex(int(arr[k], 16))
15            # print(i, j, k, hex(int(arr[k], 16)), hexResult[i][j])
16            k += 1
17    # print(f'hexResult={hexResult}')
18    return hexResult
```

Рисунок 2.24 — ConvertToHex(arr)

Функція `ConntRounds(mess, key)`: визначає кількість раундів шифрування залежно від розміру блоку даних і ключа. Задає параметри алгоритму для конкретного використання.

```

37 # визначення кількості раундів
2 usages
38 def ConntRounds(mess, key):
39     # print(f'mes={mess}, key={key}')
40     # print(f'len(key)={len(key)*len(key[0])}')
41     NbNkRound = [0, 0, 0]
42     if (len(mess) * len(mess[0])) == 16 and (len(key) * len(key[0])) == 16:
43         NbNkRound = [2, 2, 10]
44     elif (len(mess) * len(mess[0])) == 16 and (len(key) * len(key[0])) == 32:
45         NbNkRound = [2, 4, 14]
46     elif (len(mess) * len(mess[0])) == 32 and (len(key) * len(key[0])) == 32:
47         NbNkRound = [4, 4, 14]
48     elif (len(mess) * len(mess[0])) == 32 and (len(key) * len(key[0])) == 64:
49         NbNkRound = [4, 8, 18]
50     elif (len(mess) * len(mess[0])) == 64 and (len(key) * len(key[0])) == 64:
51         NbNkRound = [8, 8, 18]
52     return NbNkRound
53

```

Рисунок 2.25 — `ConntRounds(mess, key)`

Функція `Add2in64(a, b)`: реалізує додавання чисел за модулем 2^{64} . Ця операція є важливою для забезпечення дифузії в алгоритмі.

Функція `Mod2in64(a, b)`: виконує віднімання за модулем 2^{64} , що використовується під час дешифрування.

```

72 # Додавання по модулю 2^64
73 def Mod2in64(a, b):
74     a = int(a, 16)
75     b = int(b, 16)
76     result = (a + b) # результат в десятковій СЧ
77     # print(f'a+b in 10: {result}') # 264dec 364dec
78     # print(f'a+b in 16: {hex(result)}') # 108hex 16chex
79     if result > 255:
80         result %= 64 #
81         # print(f'result={result}') # 264dec 364dec
82         # print(f'a+b mod 64 in dec: {result % 64}') # 44dec
83         # print(f'a+b mod 64 in hex: {hex(result)}') # 8 2c
84     return hex(result)
85

```

Рисунок 2.26 — `Mod2in64(a, b)`

Функція `ShiftRows(arr)`: циклічно зсуває рядки матриці стану на різну кількість байтів залежно від їхнього положення. Забезпечує перемішування даних.

Функція `InvShiftRows(arr)`: виконує обернений зсув рядків для відновлення вихідного порядку даних у процесі дешифрування.

Функція `SubBytes(arr)`: реалізує підстановку байтів за допомогою S-блоків, що забезпечує нелінійність алгоритму.

Функція `InvSubBytes(arr)`: виконує обернену підстановку байтів для дешифрування.

Функція `multiply(a, b)`: реалізує множення у полі Галуа, що використовується для операції перемішування стовпців.

Функція `multiplyArr(arr1, arr2)`: виконує множення матриці стану на фіксовану матрицю в полі Галуа. Забезпечує дифузю (перемішування) даних у блоці.

Функція `XortwoArr(arr1, arr2)`: виконує побітове додавання (XOR) двох матриць. Застосовується для додавання ключа до блоку даних.

```
4 usages
713 def XortwoArr(arr1, arr2):
714     res = [[0x00] * len(arr2[0]) for i in range(8)]
715     for i in range(8):
716         for j in range(len(arr2[0])):
717             res[i][j] = hex(int(str(arr2[i][j]), 16) ^ int(str(arr1[i][j]), 16))
718     return res
719
```

Рисунок 2.27 — `XortwoArr(arr1, arr2)`

Функція `InitialStateMatrix(Nb, Nk, arr)`: формує початкову матрицю стану на основі параметрів алгоритму.

```

1 usage
55 def InitialStateMatrix(Nb, Nk, arr):
56     # print(f'Nb={Nb},Nk={Nk}, arr={arr}')
57     res = [[0x00] * len(arr[0]) for i in range(len(arr))]
58     # res = [0] * (8 * (int(len(arr) / 8)))
59     # print(f'res={res}')
60     for j in range(len(arr[0])):
61         for i in range(len(arr)):
62             if i == 0 and j == 0:
63                 res[i][j] = hex(Nb + Nk + 1)
64             else:
65                 res[i][j] = hex(res[i][j])
66             # print('')
67     # res[0][0] = hex(Nb + Nk + 1)
68     # print(f'res={res}')
69     return res
70

```

Рисунок 2.28 — InitialStateMatrix(Nb, Nk, arr)

Функція Rotate(message, key): реалізує циклічне обертання ключа, що є частиною генерації раундових ключів.

```

747 def Rotate(message, key):
748     # print(f'key={key}')
749     # print(f'message={message}')
750     res128 = [[0x00] * len(key[0]) for i in range(8)]
751     if len(message[0] * 8) == 16:
752         for i in range(8):
753             res128[i] = key[i][::-1]
754     # print(res128)
755     key = [['0x0', '0x8', '0x10', '0x18'], ['0x1', '0x9', '0x11', '0x19'],
756           ['0x2', '0xa', '0x12', '0x1a'], ['0x3', '0xb', '0x13', '0x1b'],
757           ['0x4', '0xc', '0x14', '0x1c'], ['0x5', '0xd', '0x15', '0x1d'],
758           ['0x6', '0xe', '0x16', '0x1e'], ['0x7', '0xf', '0x17', '0x1f']]
759     return res128
760     # if len(message[0] * 8) == 32:
761

```

Рисунок 2.29 — Rotate(message, key)

Проект IoT-системи КІ



Рисунок 2.36 — Схема мережі

Система є інтегрованою інфраструктурою на основі технологій Інтернету речей (IoT), яка забезпечує безпеку, моніторинг та автоматизацію процесів управління критичною інфраструктурою. Всі компоненти мережі взаємодіють між собою через локальні шлюзи або хмарні сервіси, забезпечуючи централізований контроль і можливість оперативного реагування на зміни в середовищі. Система побудована на основі демонстраційної мережі, яка є простою у структурі та конфігурації, але водночас забезпечує функціональність для демонстрації основних принципів роботи IoT-інфраструктури.

Мережа складається з датчиків, сенсорів, мережевого шлюзу, камер відоспостереження та інших елементів.

Мережевий шлюз виконує функції маршрутизації та управління потоком даних між пристроями та центральною системою.

Система функціонує за допомогою бездротового з'єднання. Для захисту з'єднань використовується протокол аутентифікації WPA2-PSK. IP-адреси видаються протоколом DHCP.

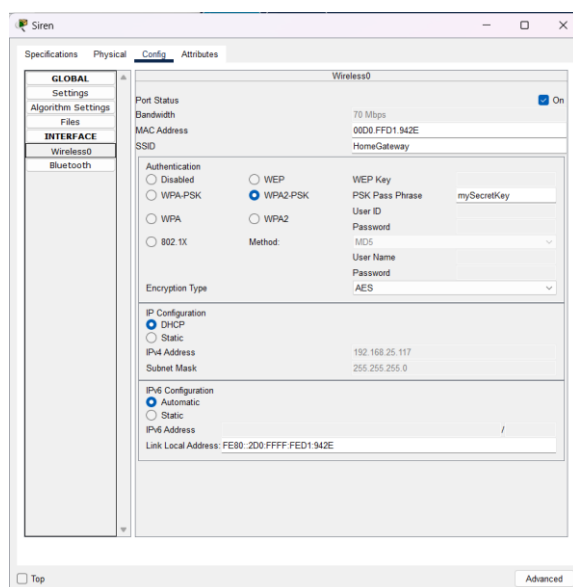


Рисунок 2.37 — Налаштування кінцевих пристроїв

Система постійно збирає дані з датчиків і сенсорів, розташованих у різних частинах інфраструктури. Ці датчики відслідковують ключові параметри, такі як температура, вологість, вібрації, рівень диму, рух, доступ до зон, а також енергоспоживання. Отримані дані передаються до центральної системи обробки, яка аналізує їх у реальному часі, виявляє відхилення від норми та активує відповідні механізми реагування. Наприклад, у разі аномалії система може автоматично заблокувати двері, увімкнути сигналізацію або припинити роботу певного обладнання.

Система також підтримує автоматизовані сценарії реагування. Наприклад, при перевищенні допустимого рівня температури в кімнаті серверів автоматично активується система охолодження та надсилається сповіщення оператору. У разі виявлення руху в забороненій зоні система запускає сирену та автоматично починає запис відео. Завдяки таким сценаріям мінімізується вплив людського фактора та прискорюється процес прийняття рішень.

Дані, зібрані системою, зберігаються у централізованій базі, що дозволяє забезпечити їхній аналіз і доступ навіть у разі збоїв живлення чи втрати локального підключення. Система інтегрується з резервними джерелами живлення, що забезпечує безперервність роботи.

Система моніторингу стану обладнання дозволяє виявляти потенційні проблеми на ранніх етапах, запобігати поломкам і знижувати ризики для критичної інфраструктури. Інтеграція з іншими системами управління, такими як ERP або системи фізичної безпеки, забезпечує комплексний підхід до управління та безпеки. Завдяки цьому забезпечується оптимізація ресурсів, підвищується надійність обладнання та ефективність роботи об'єкта.

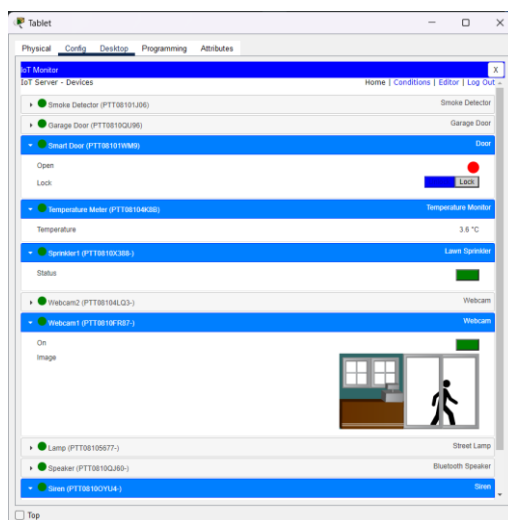


Рисунок 2.38 — Система віддаленого керування IoT пристроїв

Висновки до другого розділу

У другому розділі було розроблено модульну систему кіберзахисту. Здійснено аналіз факторів небезпеки, що впливають на безпеку критичної інфраструктури, та визначено відповідні модулі для їх нейтралізації. Детально описано функціональність кожного модуля, зокрема системи моніторингу, шифрування, IDS/IPS, антидронових систем та модулів інцидент-менеджменту. Особливу увагу приділено інтеграції цих модулів у єдину архітектуру захисту, що дозволяє ефективно реагувати на загрози та забезпечувати безперервність роботи систем.

РОЗДІЛ 3. ОЦІНКА ЕФЕКТИВНОСТІ СИСТЕМИ ТА НАПРЯМИ ЇЇ ВДОСКОНАЛЕННЯ

3.1. Впровадження системи на різних типах підприємств

Ефективне впровадження системи кібербезпеки та управління ризиками вимагає глибокого розуміння специфіки кожного типу підприємства. Різні галузі мають свої унікальні потреби, виклики та вразливості, що визначають підходи до захисту критичної інфраструктури. У цьому розділі ми розглянемо, як впровадження системи кібербезпеки може бути адаптоване для трьох різних типів підприємств, таких як міські реєстри, теплоелектроцентралі (ТЕЦ) та лікарні. Буде детально проаналізовано, які фактори загроз є найбільш значимими для кожного з цих типів інфраструктур та як відповідні модулі безпеки можуть бути застосовані для їх нейтралізації.

Міський реєстр

Кібер-фактори є найбільш значимою загрозою для міського реєстру через велику кількість конфіденційних даних, включаючи персональні дані громадян. Кібератаки можуть призвести до викрадення цих даних, що матиме серйозні наслідки для конфіденційності та безпеки.

Технічні фактори також є важливими, оскільки збої в роботі систем можуть призвести до втрати доступу до реєстрів або втрати даних, що може паралізувати роботу міської адміністрації.

Організаційні фактори мають значення, оскільки неадекватні внутрішні політики та процедури можуть призвести до витоку даних або неправильного управління реєстрами.

Соціальні фактори є критичними, оскільки атаки, які використовують соціальну інженерію, можуть призвести до несанкціонованого доступу до реєстрів через підроблені запити від імені посадовців.

Фізичні фактори вважаються менш значущими у порівнянні з іншими, але фізичний доступ до серверів та обладнання міського реєстру все ж становить загрозу у випадку неправильного фізичного захисту.

Економічні та політичні фактори можуть вплинути на фінансування та підтримку систем кібербезпеки, але їхня вага відносно менша.



Рисунок 3.1 — Відсоткове співвідношення різних факторів загроз в міському реєстрі

Теплоелектроцентрально (ТЕЦ)

Технічні фактори мають найбільшу значимість для ТЕЦ, оскільки збої в роботі обладнання можуть призвести до зупинки виробництва енергії та серйозних економічних втрат.

Кібер-фактори також є дуже важливими, оскільки атаки на системи управління ТЕЦ можуть призвести до перебоїв у постачанні енергії та саботажу.

Фізичні фактори мають значний вплив, оскільки фізичні атаки, такі як саботаж або терористичні акти, можуть пошкодити критичне обладнання.

Організаційні фактори можуть призвести до збоїв у роботі та витоку інформації про технологічні процеси через недоліки в управлінні та організаційних процедурах.

Атаки, пов'язані з людським фактором, можуть включати соціальну інженерію, де зловмисники використовують психологічні маніпуляції для доступу до систем. Це може включати фішинг атаки або підроблені запити від імені співробітників.

Економічні та політичні фактори можуть вплинути на фінансування та стабільність роботи ТЕЦ, але їхня вага є відносно меншою.

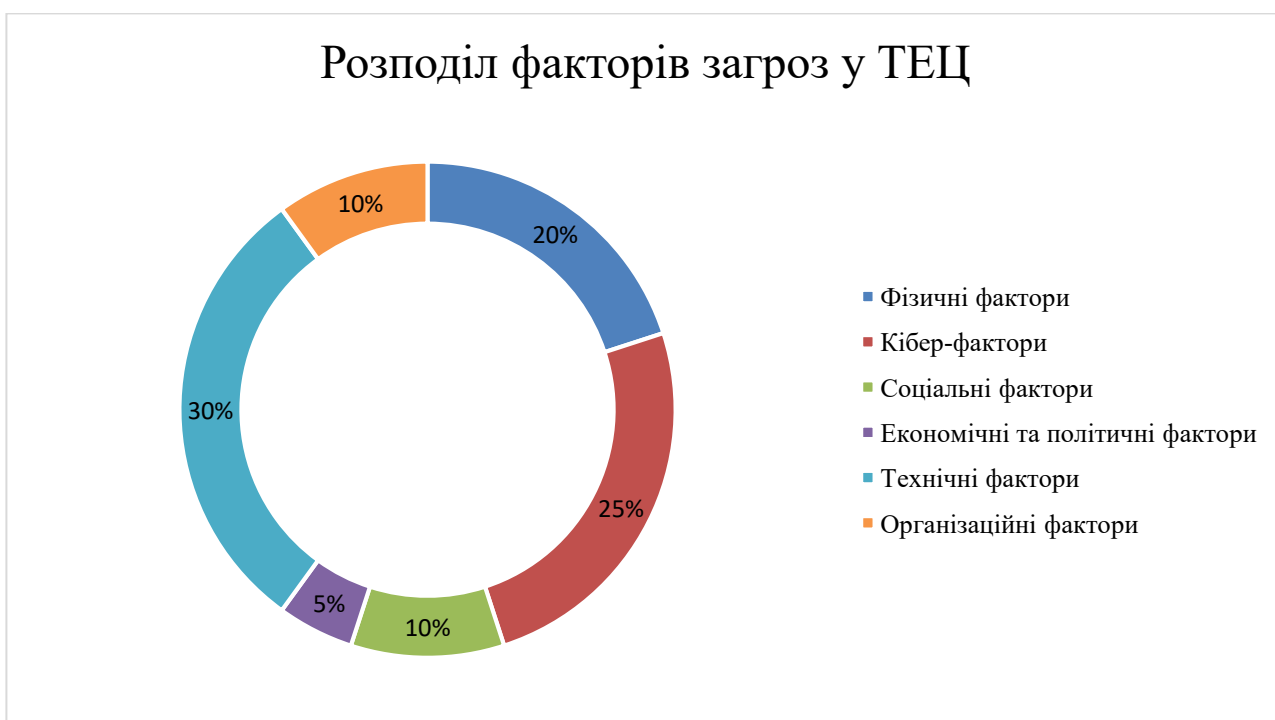


Рисунок 3.2 — Відсоткове співвідношення різних факторів загроз в міському реєстрі

Лікарня

Кібер-фактори є найважливішими для лікарні, оскільки зберігання конфіденційних медичних даних пацієнтів робить лікарні вразливими до кібератак, які можуть призвести до викрадення даних або блокування доступу до них, що матиме серйозні наслідки для пацієнтів. "Крім того, порушення роботи медичних

інформаційних систем через атаки може вплинути на оперативність надання медичної допомоги, що прямо загрожує життю та здоров'ю пацієнтів."

Технічні фактори мають значний вплив, оскільки збої в роботі медичного обладнання або програмного забезпечення можуть мати критичні наслідки для лікування пацієнтів.

Соціальні фактори є важливими, оскільки атаки, які використовують людський фактор, можуть призвести до несанкціонованого доступу до медичних даних через підроблені запити.

Організаційні фактори також мають значення, оскільки недоліки в управлінні та організаційних процедурах можуть призвести до витоку даних або неправильного управління медичними записами.

Фізичні фактори вважаються менш значущими у порівнянні з іншими, але недостатній фізичний захист медичного обладнання та даних може спричинити їх втрату або пошкодження.

Економічні та політичні фактори можуть вплинути на фінансування та підтримку систем кібербезпеки в лікарнях, але їхня вага є відносно меншою.

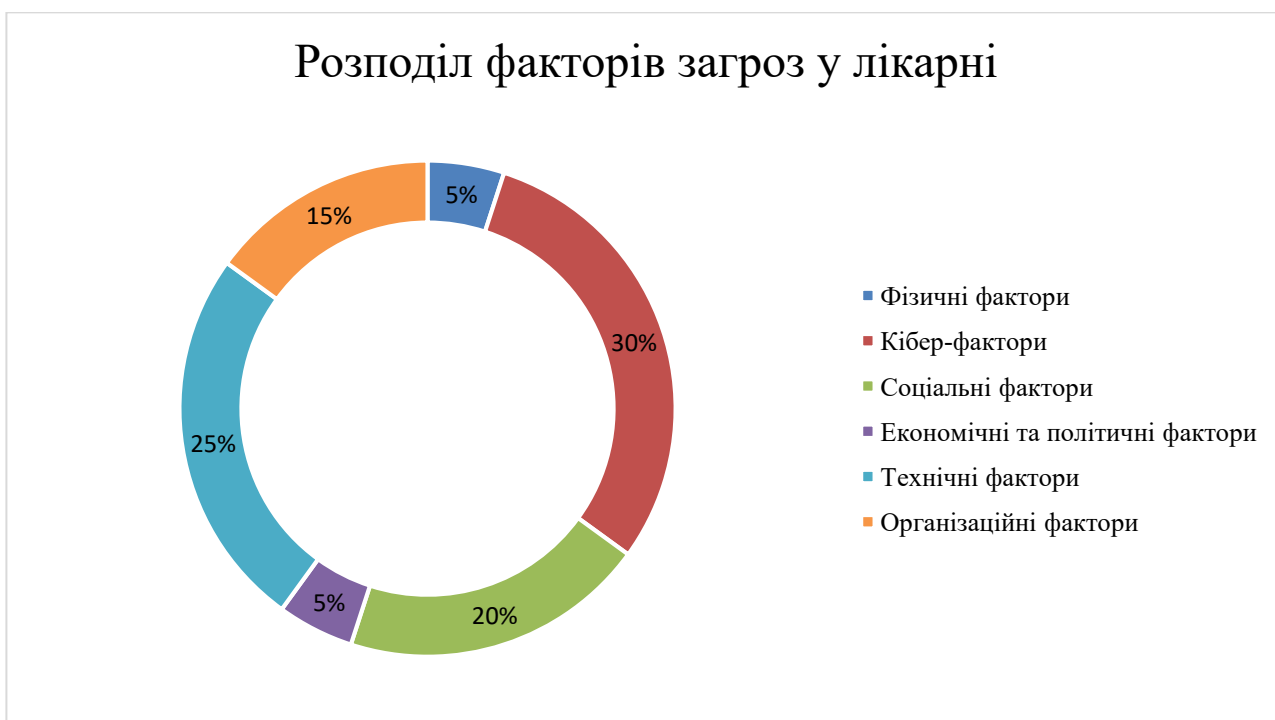


Рисунок 3.2 — Відсоткове співвідношення різних факторів загроз в лікарні

3.2. Можливі напрями покращення системи

У майбутньому систему можна покращити шляхом впровадження нових технологій та оптимізації існуючих компонентів. Це дозволить забезпечити вищий рівень безпеки, функціональності та ефективності управління.

Покращення включає впровадження штучного інтелекту (AI), щоб автоматизувати спостереження та аналіз даних. Алгоритми машинного навчання покращують точність виявлення аномалій у поведінці пристроїв і показують окремі загрози до їх реалізації.

Просування квантової криптографії є ще одним перспективним напрямком. Ця технологія забезпечує найвищий рівень захисту переданих даних і захищає від майбутніх загроз, пов'язаних із розвитком квантових обчислень.

Ще один важливий напрям – це посилення фізичної безпеки пристроїв IoT. Використання захищених корпусів, антивандальних конструкцій та додаткових сенсорів для моніторингу фізичного доступу дозволить запобігти несанкціонованому втручанню.

Розширення функціональності системи також можливе через додавання нових модулів, таких як модулі енергозбереження або автономного управління. Це забезпечить більшу незалежність системи та її стійкість у разі відсутності зовнішніх ресурсів.

Крім того, варто передбачити сценарії масштабування системи для роботи з більшим числом пристроїв і даних. Це дозволить використовувати її для складніших об'єктів критичної інфраструктури або в різних галузях.

Інтеграція з іншими системами кібербезпеки, такими як SIEM (системи управління подіями інформаційної безпеки), дозволить створити єдину екосистему, яка забезпечить комплексний підхід до управління безпекою.

Висновки до третього розділу

У третьому розділі було розглянуто можливості впровадження розробленої системи на різних типах підприємств критичної інфраструктури, таких як енер-

гетика, транспорт, водопостачання та промисловість. Здійснено оцінку ефективності запропонованих рішень у реальних умовах. Продемонстровано приклад створення демонстраційної мережі з використанням модулів захист.

ВИСНОВКИ

У процесі виконання дослідження на тему "Методи та технології побудови систем кіберзахисту для об'єктів критичної інфраструктури" було проведено аналіз сучасних кіберзагроз, які становлять ризик для критичної інфраструктури, включаючи атаки на інформаційні мережі, соціоінженерні атаки та загрози, пов'язані з використанням безпілотних літальних апаратів (БПЛА). Обґрунтовано важливість використання модульного підходу для створення гнучких і масштабованих систем захисту, що дозволяє адаптувати їх до змінних умов і специфіки об'єктів критичної інфраструктури.

Розроблено рекомендації щодо впровадження сучасних технологій, таких як криптографічні системи, системи виявлення та запобігання вторгнень (IDS/IPS), використання IoT для моніторингу стану обладнання, а також антидронові системи для захисту від фізичних загроз. Доведено, що інтеграція різних компонентів у єдину систему з урахуванням особливостей критичної інфраструктури значно підвищує її стійкість до атак та забезпечує надійний захист.

Описано приклади застосування запропонованих методів і технологій у реальних умовах, особливо в контексті викликів, з якими стикається Україна під час війни. Це дозволяє адаптувати системи кіберзахисту до специфічних умов та зростаючого обсягу загроз. Звернуто увагу на перспективні інноваційні технології, такі як штучний інтелект для аналізу поведінки загроз і оптимізації процесів виявлення аномалій.

Таким чином, запропоновані методи та технології забезпечують створення ефективної, адаптивної та масштабованої системи кіберзахисту, яка відповідає сучасним викликам. Результати дослідження можуть бути використані для вдосконалення існуючих систем кіберзахисту, зокрема для об'єктів критичної інфраструктури, що є особливо важливим у контексті національної безпеки України.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Ірина Віталіївна Г., Олег Володимирович Б. Окремі питання захисту об'єктів критичної інфраструктури: зарубіжний досвід. Соціально-правові студії. 2021. № 1(11). С. 132–139. URL: https://sls-journal.com.ua/web/uploads/pdf/S&LS_2021_Vol.%204,%20No.%201_132-139.pdf.
2. Магомедов А. О. Ризики та загрози для об'єктів критичної інфраструктури та шляхи їх подолання. *Investytsiyi: praktyka ta dosvid*. 2024. № 15. С. 216–221. URL: <https://doi.org/10.32702/2306-6814.2024.15.216> (дата звернення: 23.12.2024).
3. Учасники проектів Вікімедіа. Кібератака на енергетичні компанії України – Вікіпедія. *Вікіпедія*. URL: https://uk.wikipedia.org/wiki/Кібератака_на_енергетичні_компанії_України (дата звернення: 23.12.2024).
4. BBC News. Ukraine cyber-attack: Russia to blame for hack, says Kyiv. *BBC Home - Breaking News, World News, US News, Sports, Business, Innovation, Climate, Culture, Travel, Video & Audio*. URL: <https://www.bbc.com/news/world-europe-59992531> (date of access: 23.12.2024).
5. Reuters. Russia's Transneft says Druzhba pipeline 'attacked' near Ukraine border. <https://www.reuters.com/>. URL: <https://www.reuters.com/world/europe/russias-transneft-says-druzhba-pipeline-attacked-near-ukraine-border-tass-2023-05-10/>.
6. Contributors to Wikimedia projects. Industroyer - wikipedia. *Wikipedia, the free encyclopedia*. URL: <https://en.wikipedia.org/wiki/Industroyer> (date of access: 23.12.2024).
7. Повені у Карпатах: наслідки колосальні, є загиблі - BBC News Україна. *BBC News Україна*. URL: <https://www.bbc.com/ukrainian/news-53176538> (дата звернення: 23.12.2024).
8. Russian missiles bombard cities across Ukraine | CNN / K. Nechyporenko et al. *CNN*. URL: <https://edition.cnn.com/2022/10/31/europe/russian-missile-strikes-ukraine-intl/index.html> (date of access: 23.12.2024).

9. Учасники проектів Вікімедіа. Аварія в горлівці (2013) – вікіпедія. *Вікіпедія*.

URL: [https://uk.wikipedia.org/wiki/Аварія_в_Горлівці_\(2013\)](https://uk.wikipedia.org/wiki/Аварія_в_Горлівці_(2013)) (дата звернення: 23.12.2024).

10. Учасники проектів Вікімедіа. Чорнобильська катастрофа – Вікіпедія. *Вікіпедія*. URL: https://uk.wikipedia.org/wiki/Чорнобильська_катастрофа (дата звернення: 23.12.2024).

11. Все, що ви повинні знати про об'єкти критичної інфраструктури в Україні. *Головні новини з України сьогодні - Kyiv Post*. URL: <https://www.kyivpost.com/uk/post/28283> (дата звернення: 23.12.2024).

12. Що таке SIEM? | Захисний комплекс Microsoft. *Your request has been blocked. This could be due to several reasons*. URL: <https://www.microsoft.com/uk-ua/security/business/security-101/what-is-siem> (дата звернення: 23.12.2024).

13. Що таке SOAR? Технології та рішення | Захисний комплекс Microsoft. *Your request has been blocked. This could be due to several reasons*. URL: <https://www.microsoft.com/uk-ua/security/business/security-101/what-is-soar> (дата звернення: 23.12.2024).

14. Учасники проектів Вікімедіа. Постквантова криптографія – Вікіпедія. *Вікіпедія*. URL: https://uk.wikipedia.org/wiki/Постквантова_криптографія (дата звернення: 23.12.2024).

15. Soutar C., Barmes I., Beato F. Why the new NIST standards mean quantum cryptography may just have come of age. *World Economic Forum*. URL: <https://www.weforum.org/stories/2024/10/quantum-cryptography-nist-standards/> (date of access: 23.12.2024).

16. Federal Office for Information Security. The status of quantum computer development. *Federal Office for Information Security*. URL: <https://www.bsi.bund.de/EN/Themen/Unternehmen-und-Organisationen/Infor->

[mationen-und-Empfehlungen/Quantentechnologien-und-Post-Quanten-Kryptografie/Entwicklungsstand-Quantencomputer/entwicklungsstand-quantencomputer.html](https://www.bsi.de/Presse/Pressemitteilungen-und-Empfehlungen/Quantentechnologien-und-Post-Quanten-Kryptografie/Entwicklungsstand-Quantencomputer/entwicklungsstand-quantencomputer.html) (date of access: 23.12.2024).

17. Луцевич М. О., Бродський Ю. Б. Модульність системи кіберзахисту: можливості адаптації до сучасних викликів. *Комп'ютерні технології: інновації, проблеми, рішення* : Всеукр. науково-техн. конф., 2–3 груд. 2024 р.

18. Byun H. The advantages and disadvantages of homomorphic encryption. *Baffle*. URL: <https://baffle.io/blog/the-advantages-and-disadvantages-of-homomorphic-encryption/> (date of access: 23.12.2024).

19. Santos O. Securing retrieval augmented generation (RAG) implementations and the LLM stack. Cisco Live!, 2024. 44 p. URL: <https://www.ciscolive.com/c/dam/r/ciscolive/global-event/docs/2024/pdf/AIHUB-2001.pdf>.

20. What is homomorphic encryption? - chainlink. *Chainlink: The Standard for Onchain Finance*. URL: <https://chain.link/education-hub/homomorphic-encryption> (date of access: 23.12.2024).

21. Contributors to Wikimedia projects. Homomorphic encryption - Wikipedia. *Wikipedia, the free encyclopedia*. URL: https://en.wikipedia.org/wiki/Homomorphic_encryption (date of access: 23.12.2024).

22. Contributors to Wikimedia projects. Гомоморфное шифрование – Википедия. *Википедия – свободная энциклопедия*. URL: https://ru.wikipedia.org/wiki/Гомоморфное_шифрование#Частично_гомоморфные_системы (дата звернення: 23.12.2024).

23. Національний стандарт шифрування ДСТУ 7624:2014 («Каліна»). *Освітній портал*. URL: https://learn.ztu.edu.ua/pluginfile.php/330286/mod_resource/content/1/Лекція_6.pdf (дата звернення: 23.12.2024).

24. Учасники проєктів Вікімедіа. Купина (геш-функція) – Вікіпедія. *Вікіпедія*. URL: [https://uk.wikipedia.org/wiki/Купина_\(геш-функція\)](https://uk.wikipedia.org/wiki/Купина_(геш-функція)) (дата звернення: 23.12.2024).

25. Дослідження криптосистем на основі кодів хеш функцій. 1-й міжнародний науково-практичний форум з інформаційної безпеки «*global cyber security forum 2019*» : Міжнар. науково-практ. форум, м. Харків, 14–16 жовт. 2019 р. 2019. URL: <https://openarchive.nure.ua/server/api/core/bitstreams/8d27b82f-7260-4b53-b59a-b1ddabf80044/content>.
26. Горбенко Ю. І., Ганзя Р. С. *Прикладна електроніка* : Том. 13-те вид. 2014. С. 268–274. URL: <https://openarchive.nure.ua/server/api/core/bitstreams/b36c9482-4963-49ad-91d4-48dd907ad7c6/content>.
27. Учасники проектів Вікімедіа. Кібератака на «київстар» (2023) – вікіпедія. *Вікіпедія*. URL: [https://uk.wikipedia.org/wiki/Кибератака_на_«Київстар»_\(2023\)](https://uk.wikipedia.org/wiki/Кибератака_на_«Київстар»_(2023)) (дата звернення: 23.12.2024).
28. Цуркан О., Герасимов Р., Крук О. Методипротидії використанню соціальної інженерії. *Information technology and security*, 21 серп. 2019 р. 2019. С. 161–170. URL: <https://doi.org/10.20535/2411-1031.2019.7.2.190563>.
29. Луцевич М. О., Бродський Ю. Б. Інноваційні підходи до захисту від соціоінженерних атак в умовах цифровізації. *Інноваційна наука: пошук відповідей на виклики сучасності* : Міжнар. наук. конф., м. Могилів-Подільський, 6 груд. 2024 р. Вінниця, 2024. С. 258–259. URL: <https://archives.mcnd.org.ua/index.php/conference-proceeding/issue/view/06.12.2024/26>.
30. Ticong L. AI in cybersecurity: the comprehensive guide to modern security. *Datamation*. URL: <https://www.datamation.com/security/ai-in-cybersecurity/> (date of access: 23.12.2024).

ДОДАТКИ

Опис паткету

МПК B64C39/02; G05D1/00; G07C5/00; G08G5/00**Автоматизована система безпеки**

Заявлена корисна модель відноситься до галузі систем безпеки, а саме, до автоматизованих систем повітряної охорони промислових об'єктів чи об'єктів критичної інфраструктури на базі безпілотних літальних комплексів (БЛК), які виконують функції спостереження, контролю, виявлення та блокування роботи несанкціонованих повітряних об'єктів в контрольованій зоні.

З розвитком технологій безпілотні та роботизовані системи у поєднанні з можливостями штучного інтелекту надають можливості без залучення людини забезпечувати виконання складних задач із меншою витратою ресурсів, часу та з можливістю охоплення значних територій, як на земній поверхні, так і у повітрі.

На сьогодні із рівня техніки відомі наступні засоби безпеки на базі безпілотних літальних комплексів.

З патенту США US10607461B2 відома система безпеки, яка містить, щонайменше безпілотник і базу безпілотника, налаштовану в місцях, які є частиною традиційних систем безпеки, і де традиційна система безпеки підтримує зв'язок із базою безпілотника, що дозволяє безпілотнику реагувати на події, що відбуваються де безпілотник також забезпечений можливістю слідувати заздалегідь визначеним шляхом за контрольними точками.

Також, із патенту на винахід US11215986B2 відома автоматизована система безпеки на базі безпілотників для спостереження за місцевістю яка включає один або кілька безпілотників із вбудованими датчиками та пристроєм обробки зобра-

жень для вимірювання даних спостереження. Дані спостереження можуть включати зображення, дані телеметрії, інфрачервоні дані або іншу інформацію про місцезнаходження, яку можна виявити. Безпілотні літальні апарати можуть виконувати одну або кілька польотів, а також зберігати та передавати дані спостереження на вузол сервера, який функціонує для координації роботи безпілотної та отримання даних спостереження. Док-станція для безпілотників може бути включена для запуску, посадки та/або зберігання. Обчислювальний пристрій користувача може підтримувати зв'язок із вузлом сервера та безпілотниками, причому обчислювальний пристрій користувача здатний отримувати дані користувача та відображати дані спостереження з безпілотної. Польотні операції, пов'язані зі спостереженням за місцезнаходженням, можуть автоматично та/або вручну керуватися обчислювальним пристроєм користувача та/або вузлом сервера.

Відоме з винаходу US11215986B2 технічне рішення прийняте за найближчий аналог.

Технічною задачею на вирішення якої направлено заявлену корисну модель є створення автоматизованої системи безпеки підвищеної ефективності.

Технічний результат від реалізації заявленої корисної моделі полягає у підвищенні ефективності автоматизованої системи безпеки виконаної на базі безпілотних літальних комплексів (БЛК).

Суть заявленої корисної моделі полягає в тому, що автоматизована система безпеки інтегрована в глобальну систему збирання обробки інформації (ГСЗОІ) про переміщення повітряних об'єктів на окремій (охороняємій) території та виконана з можливістю обробки, аналізу та комбінування даних від ГСЗОІ та від власних сенсорів встановлених на БЛК та на місцевості. При цьому один БЛК включає, принаймні, одне автономне місце стоянки (АМС) з посадковими місцями принаймні для двох літальних засобів виконаних з можливістю керування від першої

особи. Кожен літальний засіб виконаний з можливістю блокування роботи несанкціонованих повітряних об'єктів на охороняємій території.

Відповідно до одного з варіантів виконання системи, окремі літальні засоби БЛК виконані з можливістю обміну даними між собою та системою управління.

Відповідно до одного з варіантів виконання системи, окремі літальні засоби БЛК виконані з можливістю переходу від автономного режиму роботи в режим прямого ручного керування від першої особи.

Відповідно до одного з варіантів виконання системи, АМС виконане з можливістю підсилення сигналів управління та їх ретрансляції літальними засобами.

Відповідно до одного з варіантів виконання системи, АМС виконані з можливістю формування сигналів керування літальним засобами БЛК.

Відповідно до одного з варіантів виконання системи, АМС є складовою частиною системи управління.

Також відповідно до одного з варіантів виконання системи, інформація про переміщення повітряних об'єктів включає принаймні відбиток радіочастотного сліду, звукового сліду, теплового сліду, візуальних параметрів.

Відповідно до одного з варіантів виконання системи, літальні засоби БЛК виконані з можливістю роботи в режимі очікування на місцях стоянки до моменту надходження сигналів від системи управління з інформацією про наявність та положення несанкціонованих повітряних об'єктів.

Суть заявленої корисної моделі пояснюється, але не обмежується наведеними графічними матеріалами:

Фіг.1 - принципова блок-схема автоматизованої системи безпеки.

Заявлена корисна модель може бути реалізована за принциповою схемою зображеною на Фіг. 1, відповідно до якої, автоматизована система безпеки включає систему управління, реалізовану на базі віддаленого серверного пристрою 1 або персонального комп'ютера, який функціонально пов'язаний з безпілотним літальним комплексом 2 (БЛК). Безпілотний літальний комплекс 2 виконаний з можливістю тривалої безперебійної роботи в режимі очікування, автономного патрулювання за вказаним маршрутом, автономного вильоту на перехоплення несанкціонованих літальних об'єктів, виявлення та ідентифікації об'єктів на контрольованій території, захоплення та трансляції даних з інформацією про несанкціоновані літальні об'єкти, та оснащений набором сенсорів, наприклад переносною або стаціонарною станцією 3 пеленгації сигналів з відбитками принаймні мікрохвильового сліду, радіочастотного сліду, звукового сліду, теплового сліду, візуальних параметрів про літальні об'єкти на охороняємій території. При цьому система додатково збирає показники засобів фото-відеофіксації, розміщених безпосередньо на елементах БЛК.

До складу БЛК входять автономні місця стоянки 4 з інтегрованим зарядним пристроєм, а також системою збирання обробки та передачі інформації від компонентів БЛК та сенсорів з інтегрованими елементами систем штучного інтелекту.

Відповідно до переважаючого варіант реалізації, автоматизована система безпеки може бути інтегрована в глобальну систему 5 збирання обробки інформації (ГСЗОІ) про переміщення повітряних об'єктів 6 на окремій території та виконана з можливістю обробки, аналізу та комбінування даних від ГСЗОІ та від власних сенсорів, встановлених на компонентах БЛК та на місцевості. Кожен БЛК включає, принаймні, одне автономне місце стоянки 4 (АМС) з посадковими місцями принаймні для двох літальних засобів 7 виконаних з можливістю керування від першої особи, де кожен літальний засіб 7 виконаний з можливістю фізичного блокування роботи та змінення курсу польоту несанкціонованих повітряних об'єктів на певній території.

Окремі літальні засоби 7 БЛК, виконані з можливістю обміну даними між собою та системою управління, яка може бути реалізована як на базі АМС 4 та і на базі віддаленого сервера 1 залежно від умов експлуатації.

Окремі літальні засоби 7 БЛК можуть бути виконані з можливістю переходу з автономного режиму в режим прямого ручного керування від першої особи оператора віддаленого сервера 1.

Відповідно до одного з можливих варіантів реалізації, АМС 4 можуть бути виконані з можливістю підсилення та ретрансляції сигналу управління літальними засобами, що суттєво підвищує дальність польоту літальних засобів 7.

Також АМС 4 можуть виконані з можливістю формування сигналів управління літальними засобами БЛК, зокрема можуть бути задіяні у обчисленні та передачі координат положення, напрямку руху, висоти та швидкості польоту несанкціонованих повітряних об'єктів 6, таким чином АМС 4 є складовою частиною системи управління.

В ході експлуатації автоматизованої системи безпеки, засобами фіксації та збору інформації, здійснюється відбір та актуалізація інформації про переміщення повітряних об'єктів 6, яка включає, принаймні, відбиток радіочастотного сліду, звукового сліду, теплового сліду, візуальних параметрів. Відбір та актуалізація інформації про переміщення повітряних об'єктів 6 може виконуватись із застосуванням елементів систем штучного інтелекту.

Також відповідно до переважаючого варіанту реалізації корисної моделі, літальні засоби 7 БЛК, виконані з можливістю роботи в режимі очікування на місцях стоянки 4 до моменту надходження сигналів від системи управління з інформацією про наявність та положення несанкціонованих повітряних об'єктів 6.

Літальні засоби 7 БЛК спроможні як автономно, за власними засобами фіксації та збору інформації, так і за комбінування даних від ГСЗОІ, виходити на виявлення, відстеження, супроводження чи перехват несанкціонованих повітряних

об'єктів 6. Літальні засоби 7 БЛК за рішенням першої особи оператора віддаленого сервера 1 з метою перехоплення несанкціонованих повітряних об'єктів 6 можуть задіювати модулі фізичного блокування роботи та змінення курсу польоту несанкціонованих повітряних об'єктів

Реалізація заявленої корисної моделі дозволяє забезпечити досягнення зазначеного технічного результату, підвищуючи ефективність виявлення та блокування роботи несанкціонованих повітряних об'єктів у межах визначеної території за рахунок ефективної автономної системи виявлення та індикації об'єктів інтегрованої у глобальну систему збирання обробки інформації про надзвичайні ситуації та небезпечні об'єкти.

Програмна реалізація шифру Пайє

```
import math
import random
from tkinter import *
from tkinter import messagebox, scrolledtext

def get_prime(bits):
    while True:
        num = random.getrandbits(bits)
        if is_prime(num):
            return num

def is_prime(n):
    if n <= 1:
        return False
    for i in range(2, int(n ** 0.5) + 1):
        if n % i == 0:
            return False
    return True

def extended_gcd(a, b):
    x0, x1, y0, y1 = 1, 0, 0, 1
    while b != 0:
        q, a, b = a // b, b, a % b
        x0, x1 = x1, x0 - q * x1
        y0, y1 = y1, y0 - q * y1
    return a, x0, y0

def mod_inverse(a, n):
    gcd, x, y = extended_gcd(a, n)
    if gcd != 1:
        raise Exception('Модульний обернений елемент не існує')
    else:
        return x % n

def append_output(text):
    output_text.insert(END, text + "\n")
    output_text.see(END)

def generate_keys(bits=5):
    if toggle_var.get():
        append_output("Генерація ключів...:")
        append_output("Генеруємо два великих простих числа p та q:")
    #p = 7
    p = get_prime(bits)

    #q = 5
    q = get_prime(bits)
```

```

    if toggle_var.get():
        append_output(f"p = {p}, q = {q}")
    if toggle_var.get():
        append_output(
            f"Перевіряємо НСД для чисел (pq, p-1, q-1): {p * q}, {p - 1}, {q - 1}:
{math.gcd(p * q, q - 1, p - 1)}")
        n = p * q
    if toggle_var.get():
        append_output(f"Визначаємо значення n за формулою n=pq: n = {n}")
    #g = 3
    while True: # Генерація випадкового числа g з перевіркою на взаємнопрості
числа з n*n #
        g = random.randint(1, n*n - 1)
        if math.gcd(g, n) == 1:
            break
    if toggle_var.get():
        append_output(f"Генеруємо випадкове просте число g: g = {g}")
    lambda_n = math.lcm(p - 1, q - 1)
    u = (g ** lambda_n) % (n ** 2)
    Lu = (u - 1) // n
    if toggle_var.get():
        append_output(f"Визначаємо значення НСК(p-1, q-1): {lambda_n}")
        append_output(f"Визначаємо значення mu: [L(g^lambda_n mod n^2)]^-1 mod
n")
        append_output(f"1) g^lambda_n mod n^2: {g} ^ {lambda_n} mod {n ** 2} =
{u}")
        append_output(f"2) L(u)=(u-1)/n: ({u}-1/{n})={Lu}")
    mu = mod_inverse(Lu, n)
    if toggle_var.get():
        append_output("Застосуємо розширений метод Евкліда:")
        append_output(f"3) mu = {Lu}^-1 mod {n}={mu}")
    return [g, n, lambda_n, mu]

def encrypt(public_key, plaintext):
    g, n = public_key[0], public_key[1]
    if toggle_var.get():
        append_output("Шифрування...:")
        append_output("Генеруємо випадкове просте число r:")
    while True: # Генерація випадкового числа g з перевіркою на взаємнопрості
числа з n*n #
        r = random.randint(1, n - 1)
        if math.gcd(r, n) == 1:
            break
    # r = 9
    if toggle_var.get():
        append_output(f"r = {r}")
    c = (g ** plaintext) * (r ** n) % (n ** 2)
    if toggle_var.get():
        append_output(f"c=(g^m)*(r^n) mod n^2={g}^{plaintext}*{r}^{n} mod {n **
2}={c}")
    return c

def decrypt(public_key, private_key, ciphertext):
    n = public_key[1]

```

```

lambda_n = private_key[0]
mu = private_key[1]
u = (ciphertext ** lambda_n) % (n ** 2)
Lu = (u - 1) // n
plaintext = (Lu * mu) % (n * 2)
if toggle_var.get():
    append_output("Дешифрування...:")
    append_output(f"m'=L(c^lambda_n mod n^2)*(mu mod n):")
    append_output(f"1) c^lambda_n mod n^2: {ciphertext}^{lambda_n} mod {n ** 2} = {u}")
    append_output(f"2) L(u)=(u-1)/n: ({u}-1/{n})={Lu}")
    append_output(f"3) m' = {Lu}*{mu} mod {n}={plaintext}")
    append_output(f"Розшифрований текст = {plaintext}")
return plaintext

def generate_keys_gui():
    try:
        keys = generate_keys()
        public_key = keys[0], keys[1]
        private_key = keys[2], keys[3]
        public_key_label.config(text=f"Відкритий ключ: {public_key}")
        private_key_label.config(text=f"Закритий ключ: {private_key}")
        global generated_public_key, generated_private_key
        generated_public_key = public_key
        generated_private_key = private_key
    except Exception as e:
        messagebox.showerror("Помилка", str(e))

def encrypt_message():
    try:
        plaintext = int(plaintext_entry.get())
        cipher = encrypt(generated_public_key, plaintext)
        encrypted_label.config(text=f"Зашифрований текст: {cipher}")
    except Exception as e:
        messagebox.showerror("Помилка", str(e))

def decrypt_message():
    try:
        ciphertext = int(encrypted_label.cget("text").split(": ")[1])
        plain = decrypt(generated_public_key, generated_private_key, ciphertext)
        decrypted_label.config(text=f"Дешифрований текст: {plain}")
    except Exception as e:
        messagebox.showerror("Помилка", str(e))

root = Tk()
root.title("Шифр Пайе")

Label(root, text="Введіть число для шифрування:").pack(pady=10)
plaintext_entry = Entry(root)
plaintext_entry.pack(pady=10)

toggle_var = BooleanVar()
toggle_button = Checkbutton(root, text="Показувати дії покроково",

```

```
variable=toggle_var)
toggle_button.pack(pady=10)

Button(root, text="Генерувати ключі", command=generate_keys_gui).pack(pady=10)
public_key_label = Label(root, text="Відкритий ключ:")
public_key_label.pack(pady=10)
private_key_label = Label(root, text="Закритий ключ:")
private_key_label.pack(pady=10)

Button(root, text="Шифрувати", command=encrypt_message).pack(pady=10)
encrypted_label = Label(root, text="Зашифрований текст:")
encrypted_label.pack(pady=10)

Button(root, text="Дешифрувати", command=decrypt_message).pack(pady=10)
decrypted_label = Label(root, text="Дешифрований текст:")
decrypted_label.pack(pady=10)

output_text = scrolledtext.ScrolledText(root, width=80, height=20)
output_text.pack(pady=10)

root.mainloop()
```

Програмна реалізація додавання шифру Пайс

```

import math
import random
from tkinter import *
from tkinter import messagebox, scrolledtext

def get_prime(bits):
    while True:
        num = random.getrandbits(bits)
        if is_prime(num):
            return num

def is_prime(n):
    if n <= 1:
        return False
    for i in range(2, int(n**0.5) + 1):
        if n % i == 0:
            return False
    return True

def extended_gcd(a, b):
    x0, x1, y0, y1 = 1, 0, 0, 1
    while b != 0:
        q, a, b = a // b, b, a % b
        x0, x1 = x1, x0 - q * x1
        y0, y1 = y1, y0 - q * y1
    return a, x0, y0

def mod_inverse(a, n):
    gcd, x, y = extended_gcd(a, n)
    if gcd != 1:
        raise Exception('Модульний обернений елемент не існує')
    else:
        return x % n

def append_output(text, color="black"):
    output_text.insert(END, text + "\n")
    start_idx = output_text.index("end-1c linestart")
    end_idx = output_text.index("end-1c lineend")
    output_text.tag_add(color, start_idx, end_idx)
    output_text.see(END)

def generate_keys(bits=5):
    if toggle_var.get():
        append_output("Генерація ключів...", "blue")
        append_output("Генеруємо два великих простих числа p та q:", "blue")
        #p = 7
        p = get_prime(bits)

        #q = 5
        q = get_prime(bits)
        if toggle_var.get():
            append_output(f"p = {p}, q = {q}", "blue")
        if toggle_var.get():

```

```

        append_output(f"Перевіряємо НСД для чисел (pq, p-1, q-1): {p * q},{p - 1},{q - 1}: {math.gcd(p * q, q - 1, p - 1)}", "blue")
        n = p * q
        if toggle_var.get():
            append_output(f"Визначаємо значення n за формулою n=pq: n = {n}", "blue")
        #g = 3
        while True: # Генерація випадкового числа g з перевіркою на взаємнопрості числа з n*n #
            g = random.randint(1, n*n - 1)
            if math.gcd(g, n) == 1:
                break
        if toggle_var.get():
            append_output(f"Генеруємо випадкове просте число g: g = {g}", "blue")
        lambda_n = math.lcm(p - 1, q - 1)
        u = (g ** lambda_n) % (n ** 2)
        Lu = (u - 1) // n
        if toggle_var.get():
            append_output(f"Визначаємо значення НСК(p-1, q-1): {lambda_n}", "blue")
            append_output(f"Визначаємо значення mu: [L(g^lambda_n mod n^2)]^-1 mod n", "blue")
            append_output(f"1) g^lambda_n mod n^2: {g} ^ {lambda_n} mod {n ** 2} = {u}", "blue")
            append_output(f"2) L(u)=(u-1)/n: ({u}-1/{n})={Lu}", "blue")
        mu = mod_inverse(Lu, n)
        if toggle_var.get():
            append_output("Застосуємо розширений метод Евкліда:", "blue")
            append_output(f"3) mu = {Lu}^-1 mod {n}={mu}", "blue")
        return [g, n, lambda_n, mu]

def encrypt(public_key, plaintext):
    g, n = public_key[0], public_key[1]
    if toggle_var.get():
        append_output("Шифрування...:", "green")
        append_output("Генеруємо випадкове просте число r:", "green")
    while True:
        r = random.randint(1, n - 1)
        if math.gcd(r, n) == 1:
            break
    if toggle_var.get():
        append_output(f"r = {r}", "green")
    c = (g ** plaintext) * (r ** n) % (n ** 2)
    if toggle_var.get():
        append_output(f"c=(g^m)*(r^n) mod n^2={g}^{plaintext}*{r}^{n} mod {n ** 2}={c}", "green")
    return c

def decrypt(public_key, private_key, ciphertext):
    n = public_key[1]
    lambda_n = private_key[0]
    mu = private_key[1]
    u = (ciphertext ** lambda_n) % (n ** 2)
    Lu = (u - 1) // n
    plaintext = (Lu * mu) % (n * 2)
    if toggle_var.get():
        append_output("Дешифрування...:", "red")
        append_output(f"m'=L(c^lambda_n mod n^2)*(mu mod n):", "red")

```

```

        append_output(f"1)  $c^{\lambda_n} \bmod n^2$ : {ciphertext}^{{lambda_n}} mod {n ** 2} = {u}", "red")
        append_output(f"2)  $L(u) = (u-1)/n$ :  $((u-1)/n) = \{Lu\}$ ", "red")
        append_output(f"3)  $m' = \{Lu\} * \{\mu\} \bmod \{n\} = \{\text{plaintext}\}$ ", "red")
        append_output(f"Розшифрований текст = {plaintext}", "red")
    return plaintext

def add_encrypted_numbers(public_key, c1, c2):
    n = public_key[1]
    sum_c = (c1 * c2) % (n ** 2)
    if toggle_var.get():
        append_output(f"Сума зашифрованих чисел:  $(c1 * c2) \bmod n^2 = (\{c1\} * \{c2\}) \bmod \{n^2\} = \{\text{sum\_c}\}$ ", "purple")
    return sum_c

def generate_keys_gui():
    try:
        keys = generate_keys()
        public_key = keys[0], keys[1]
        private_key = keys[2], keys[3]
        public_key_label.config(text=f"Відкритий ключ: {public_key}")
        private_key_label.config(text=f"Закритий ключ: {private_key}")
        global generated_public_key, generated_private_key
        generated_public_key = public_key
        generated_private_key = private_key
    except Exception as e:
        messagebox.showerror("Помилка", str(e))

def encrypt_message():
    try:
        plaintext1 = int(plaintext_entry1.get())
        plaintext2 = int(plaintext_entry2.get())
        cipher1 = encrypt(generated_public_key, plaintext1)
        cipher2 = encrypt(generated_public_key, plaintext2)
        encrypted_label1.config(text=f"Зашифрований текст 1: {cipher1}")
        encrypted_label2.config(text=f"Зашифрований текст 2: {cipher2}")
        sum_cipher = add_encrypted_numbers(generated_public_key, cipher1, cipher2)
        sum_cipher_label.config(text=f"Сума зашифрованих чисел: {sum_cipher}")
    except Exception as e:
        messagebox.showerror("Помилка", str(e))

def decrypt_message():
    try:
        ciphertext1 = int(encrypted_label1.cget("text").split(": ")[1])
        ciphertext2 = int(encrypted_label2.cget("text").split(": ")[1])
        sum_ciphertext = int(sum_cipher_label.cget("text").split(": ")[1])
        plain1 = decrypt(generated_public_key, generated_private_key, ciphertext1)
        plain2 = decrypt(generated_public_key, generated_private_key, ciphertext2)
        sum_plain = decrypt(generated_public_key, generated_private_key, sum_ciphertext)
        decrypted_label1.config(text=f"Дешифрований текст 1: {plain1}")
        decrypted_label2.config(text=f"Дешифрований текст 2: {plain2}")
        sum_plain_label.config(text=f"Сума дешифрованих чисел: {sum_plain}")
    except Exception as e:

```

```

        messagebox.showerror("Помилка", str(e))

root = Tk()
root.title("Демонстрація дії додавання шифру Пайє ")

Label(root, text="Введіть перше число для шифрування:").grid(row=0, column=0,
padx=10, pady=10)
plaintext_entry1 = Entry(root)
plaintext_entry1.grid(row=0, column=1, padx=10, pady=10)

Label(root, text="Введіть друге число для шифрування:").grid(row=1, column=0,
padx=10, pady=10)
plaintext_entry2 = Entry(root)
plaintext_entry2.grid(row=1, column=1, padx=10, pady=10)

toggle_var = BooleanVar()
toggle_button = Checkbutton(root, text="Показувати дії покроково",
variable=toggle_var)
toggle_button.grid(row=2, column=0, columnspan=2, padx=10, pady=10)

Button(root, text="Генерувати ключі", command=generate_keys_gui).grid(row=3,
column=0, columnspan=2, padx=10, pady=10)
public_key_label = Label(root, text="Відкритий ключ:")
public_key_label.grid(row=4, column=0, columnspan=2, padx=10, pady=10)
private_key_label = Label(root, text="Закритий ключ:")
private_key_label.grid(row=5, column=0, columnspan=2, padx=10, pady=10)

Button(root, text="Шифрувати", command=encrypt_message).grid(row=6, column=0,
columnspan=2, padx=10, pady=10)
encrypted_label1 = Label(root, text="Зашифрований текст 1:")
encrypted_label1.grid(row=7, column=0, padx=10, pady=10)
encrypted_label2 = Label(root, text="Зашифрований текст 2:")
encrypted_label2.grid(row=7, column=1, padx=10, pady=10)

sum_cipher_label = Label(root, text="Сума зашифрованих чисел:")
sum_cipher_label.grid(row=8, column=0, columnspan=2, padx=10, pady=10)

Button(root, text="Дешифрувати", command=decrypt_message).grid(row=9, column=0,
columnspan=2, padx=10, pady=10)
decrypted_label1 = Label(root, text="Дешифрований текст 1:")
decrypted_label1.grid(row=10, column=0, padx=10, pady=10)
decrypted_label2 = Label(root, text="Дешифрований текст 2:")
decrypted_label2.grid(row=10, column=1, padx=10, pady=10)

sum_plain_label = Label(root, text="Сума дешифрованих чисел:")
sum_plain_label.grid(row=11, column=0, columnspan=2, padx=10, pady=10)

output_text = scrolledtext.ScrolledText(root, width=80, height=20)
output_text.grid(row=12, column=0, columnspan=2, padx=10, pady=10)
output_text.tag_configure("blue", foreground="blue")
output_text.tag_configure("green", foreground="green")
output_text.tag_configure("red", foreground="red")
output_text.tag_configure("purple", foreground="purple")

root.mainloop()

```

