Міністерство освіти і науки України
Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра обчислювальної техніки

Пояснювальна записка

до дипломної роботи
магістр
(освітньо-кваліфікаційний рівень)

на тему: **«СТЕГАНОГРАФІЧНИЙ МЕТОД ПЕРЕДАЧІ ІНФОРМАЦІЇ В ЗАГОЛОВКАХ ПРОТОКОЛЬНИХ БЛОКІВ ДАНИХ»**

Виконав: студент 2 курсу, групи 1КІ-18м
спеціальності
123 – Комп'ютерна інженерія
(шифр і назва напряму підготовки, спеціальності)

Моторнюк Дмитро Андрійович
(прізвище та ініціали)
Керівник доц. каф. ОТ, к.т.н. Захарченко С.М.
(прізвище та ініціали)
Рецензент доц. каф. МБІС, к.т.н. Карпінець В.В.
(прізвище та ініціали)

м. Вінниця - 2019 рік

Вінницький національний технічний університет

Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра обчислювальної техніки
Освітньо-кваліфікаційний рівень магістр
Напрям підготовки 12 – Інформаційні технології
Спеціальність 123 – Комп'ютерна інженерія

ЗАТВЕРДЖУЮ
Завідувач кафедри Т. Б. Мартинюк

_____
"____" _____20___року

З А В Д А Н Н Я
НА ДИПЛОМНУ РОБОТУ СТУДЕНТУ
Моторнюку Дмитру Андрійовичу

1. Тема проекту (роботи) Стеганографічний метод передачі інформації в заголовках протокольних блоків даних
Керівник проекту (роботи) Захарченко Сергій Михайлович, к.т.н., доц. каф. ОТ
затверджені наказом вищого навчального закладу від "___"_____20__року №___
2. Строк подання студентом проекту (роботи)_____
3. Вихідні дані до проекту (роботи) список технічної літератури, аналіз, вивчення та дослідження процесів захисту інформації, технічне завдання на магістерську роботу.
4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) Аналіз сучасних методів та засобів захисту інформації. Аналіз полів протокольних блоків даних на можливість використання їх в якості контейнера для передачі прихованого повідомлення. Аналіз отриманих даних. Визначення методів передачі прихованих повідомлень на мережевому і транспортному рівнях. Вибір найдоцільнішого методу. Програмна реалізація обраного методу. Економічна доцільність розробки.
5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) Загальний вигляд стеганосистем. Класифікація цифрової стеганографії. Мережева модель OSI. Структура стеку протоколів TCP/IP. Структура заголовку UDP. Структура заголовку TCP. Структура заголовку IP-пакету. Структура кадру Ethernet. Порівняння розміру контейнерів із загальним розміром заголовку.

6. Консультанти розділів проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Технічний розділ	Захарченко С.М.		
Економічний розділ	Глущенко Л. Д.		

7. Дата видачі завдання_____

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту (роботи)	Примітка
	Огляд існуючих підходів до розв'язання задачі		
	Аналіз сучасних методів захисту інформації		
	Аналіз протокольних блоків даних		
	Аналіз полів за головків протокольних блоків даних на можливість використання їх в якості контейнерів		
	Визначення методів передачі прихованого повідомлення		
	Проведення економічних розрахунків		
	Оформлення пояснювальної записки до дипломної роботи		
	Оформлення додатків та графічного матеріалу		

Студент _____ Моторнюк Д.А.
(підпис) (прізвище та ініціали)

Керівник роботи _____ Захарченко С.М.
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Дана магістерська кваліфікаційна робота присвячена розробці та програмній реалізації стеганографічному методу передачі повідомлення у протокольних блоках даних. Цей програмний засіб дозволить передавати приховане повідомлення використовуючи службові поля заголовків протокольних блоків даних.

В даній магістерській роботі виконано дослідження і аналіз сучасних методів та засобів захисту інформації, розглянуто існуючі методи стеганографії та приховування передачі повідомлень, визначено декілька методів передачі прихованих повідомлень, проаналізовано та обрано з них найдоцільніший, реалізовано даний метод у вигляді прикладного додатку

ABSTRACT

This master qualificational work is dedicated to developing and program realization of the stenography method of sending messages in protocol data blocks. This program will allow transfer hidden data using header field of protocol data blocks.

In this master work was done researches and analyses of modern methods and ways of protecting the information. Here was examined existing methods of stenography and hiding of transferring messages analysed and choose the most vital of them. This method was realised in the way of making the application.

ЗМІСТ

ВСТУП

Актуальність теми дослідження. Основна частина інформації перебувають у цифорвому вигляді. За результатами дослідження Big Data, Bigger Digital Shadows and Biggest Growth in the Far East, що були проведені IDC, об'єм інформації на 2012 рік становив 2,8 зеттабайт. За прогнозами на 2020 рік її кількість збільшиться до 40 зеттабайт. Серед цих даних 53,3% будуть відноситись не до сфери розваг, і серед них захищено менше 20%. [1]

Основними напрямками захисту інформації, що передається є її шифрування та створення тунелів для передачі, що являє собою по факту також шифруванням.

Для підвищення захищеності збільшується складність шифрування. Але якщо зловмисник отримає зашифровану інформацію, яка не втратить свою актуальність з часом, в нього буду буде час для того щоб розшифрувати ці дані.

Тому існує задача підвищення захищеності інформації за рахунок приховування самого факту її передачі. Технологія, що дозволяє це – стеганографія.

Серед можливих методів стеганографії окремої уваги заслуговує безпосередньо протокольні блоки даних, оскільки вони мають багато можливостей приховування інформації.

Зв'язок роботи з науковими програмами, планами, темами. Магістерська робота виконана відповідно до напрямку наукових досліджень кафедри обчислювальної техніки в галузі комп'ютерних систем та мереж, а також, спеціальності 123 – комп'ютерна інженерія.

Метою дослідження є вдосконалення методів захисту інформації, а саме методу приховування інформації – стеганографііі, а також збільшення кількості інформації, яку можливо передати в прихованому вигляді. Для досягнення поставленої мети необхідно розв'язати такі завдання:

– провести аналіз заголовків протокольних блоків даних на можливість використання полів в якості контейнерів для прихованого повідомлення;

– розглянути існуючі методи стеганографії на можливість їх вдосконалення;

– проаналізувати можливі методи передачі та вибір з них найдоцільніший;

– реалізувати обраний метод.

Об’єктом дослідження є сучасні процеси захисту інформації.

Предметом дослідження є сучасні технології комп’ютерних мереж та захисту інформації.

Методи дослідження. Дослідження, виконані під час роботи над кваліфікаційною магістерською роботою, ґрунтуються на основних поняттях технологій комп’ютерних мереж; технологіях захисту інформації та її прихованої передачі.

Наукова новизна одержаних результатів полягає в такому:

Практичне значення одержаних результатів полягає у такому:

1) Визначено методи приховування інформації в заголовках протокольних блоків даних на мережевому та транспортному рівнях.

2) Визначено швидкість передачі прихованих повідомлень визначених методів.

3) Удосконалено метод передачі прихованої інформації використовуючи псевдовипадкові поля в заголовках протокольних даних.

Отримані результати можуть використовуватись в процесі розробки програмного забезпечення, цілю якого є приховування факту передачі повідомлення.

Достовірність теоретичних положень магістерської кваліфікаційної роботи підтверджується строгістю постановки задач, коректним застосуванням інформаційних технологій під час доведення наукових положень, строгим

виведенням аналітичних співвідношень, збіжністю результатів дослідження з результатами, що отримані під час впровадження розробленої комплексної математичної моделі..

Матеріали роботи були представлені на XLVIII Науково-технічній конференції факультету інформаційних технологій та комп’ютерної інженерії [2].

1 ОГЛЯД ТЕХНІЧНОГО ЗАВДАННЯ І МОЖЛИВИХ ШЛЯХІВ ВИРІШЕННЯ

1.1 Огляд технічної проблеми

Інформація, яку людина отримує і обробляє кожен день, збільшується з кожним роком. Причому значну частину цієї інформації людина отримує з цифрових джерел. Це призводить до небезпек, а саме:

– порушення цілісності, тобто можливості зміни або знищення інформації;

– порушення конфіденційності, можливості отримання доступу до інформації користувачами, які е мають на це необхідні права;

– недоступність, можливість знищення або блокування.

Ці небезпеки змусили створити поняття, таке як захист інформації, що забезпечує її цілісність, конфіденційність, доступність.

Основними способами захисту інформації є її шифрування і приховування, тобто криптографічний і стеганографічний методи.

1.1.1 Криптографія

Криптографія – є наукою про методи забезпечення конфіденційності, цілісності даних та їх аутентифікація. Криптографію, в розумінні секретного запису можна поділити на:

а) Безпосередньо криптографію – відкритий секретний запис

б) Стеганографію – закритий секретний запис

1) Технічна стеганографія – приховування реальними фізичними діями з носієм

2) Лінгвістична стеганографія

– Семограмми – явно прихований секретний запис

– Відкритий код – неявно прихований секретний запис [3]

Загалом криптографія забезпечує конфіденційність повідомлення, яке повинно передаватися. Але при цьому шифрування інформації не допоможе уникнути її можливого знищення. адже приховування передачі інформації є більш безпечним методом, а в ідеалі, для максимального захисту потрібно їх комбінувати.

1.1.2 Стеганографія

Стеганографія являє собою метод організації зв'язку, що приховує саму наявність передачі інформації. На відміну від криптографічних методів, де зловмисник точно може визначити чи є передане повідомлення зашифрованим, методи стеганографії дозволяють вбудовувати приховане повідомлення в звичайні повідомлення так, що неможливо було б запідозрити існування вбудованого таємного повідомлення.

Стеганографія це один із найбільш давніх методів захисту інформації. Перші приклади її використання були відомі задовго до нашої ери. Але переважна більшість цих методів були технічними, а саме приховування відбувалось фізично, так як, наприклад, зникаючі чорнила.

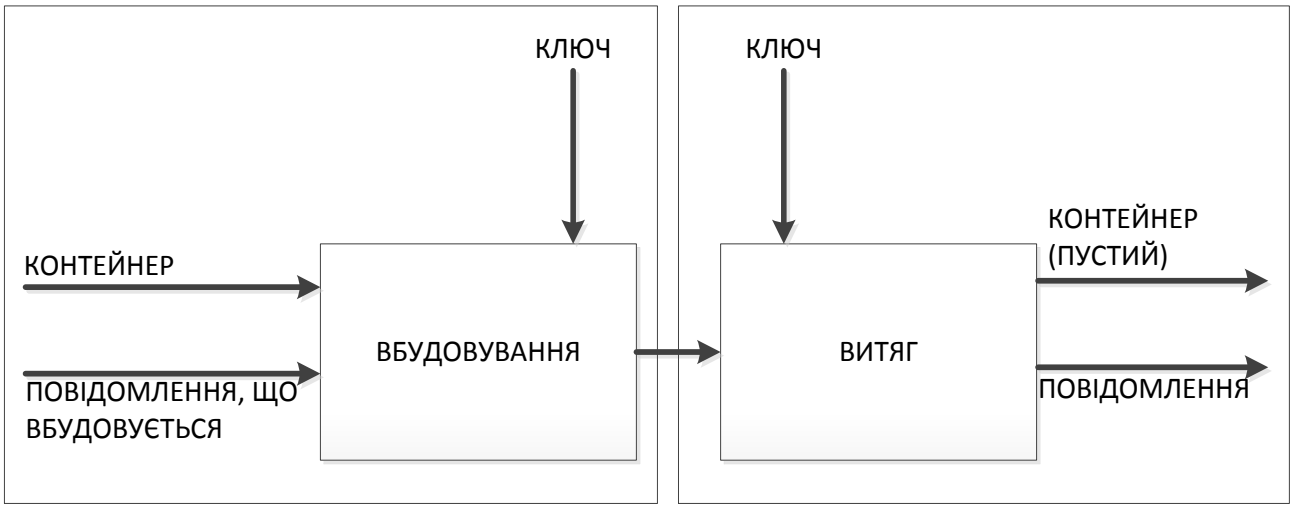


Рисунок 1.1 – Загальний вигляд стеганосистем

Із розвитком технологій також свій розвиток отримали, звісно, і методи приховування повідомлень. Основою задачею стеганографії є пошук контейнера для передачі секретної інформації. І чим більше він неявний, тим краще.

Загалом стеганографію можна поділити на три основних направлення:

– Класична стеганографія
– Комп'ютерна стеганографія
– Цифрова стеганогафія

Класична стеганографія це саме фізичні методи приховування, що використовувались ще до появи цифрової техніки і інформація передавалась в іншому форматі.

Комп'ютерна стеганографія – це один із напрямків класичної стеганографії, але з основою комп'ютерної техніки. Так з появою дисків для приховування почали використовувати частини, що не використовуються при записі основної інформації. Можливе також форматування дисків під розмір секторів, що відрізняється від розповсюдженого в операційних системах. Гарним прикладом є файлова система така як, наприклад, StegFS для Linux. Більш широка у використанні файлова система FAT32, що має стандартний розмір кластера 4 Кб. При цьому для зберігання файлу меншого розміру виділяється 4 Кб, а вільне місце в кластері можна використати, як контейнер для зберігання.

Цифрова стеганографія – це направлення, основане на приховуванні інформації в цифрові об'єкти, при цьому зберігаючи функціонал цих об'єктів.

Як правило, ці об'єкти є мультимедійими об'єктами. Внесення спотворень, що знаходяться нижче порога чуттєвості людини, не призводить до сильно помітних змін цих об'єктів. Тобто в якості носія прихованої інформації має бути об'єкт, який допускає спотворення інформації, але при цьому зберігається його функціональність.

В якості носія зазвичай використовуються файли зображень або звукові файли. Такі файли мають велику надлишковість і зазвичай мають великий розмір, що забезпечує достатньо кількості місця для приховування простого або

форматованого повідомлення. Приховуване повідомлення може бути простим набором чисел, зображенням, простим або зашифрованим текстом.

Рисунок 1.2 – Класифікація цифрової стеганографії

Більшість мультимедійних форматів мають поля розширення, які можуть заповнюватися користувацької інформацією або заповнені нулями – в останньому випадку їх також можна використовувати як контейнери для зберігання і передачі інформації. Але цей метод не тільки не забезпечує необхідного рівня захищеності, але і не дає можливості приховувати значні обсяги даних.

Для забезпечення більшої непомітності при розміщенні файлів у контейнер використовують алгоритми шифрування. Для більшої надійності і схожості оригіналу слід використовувати зображення з завадами в молодших розрядах. Такі зображення вже містять в собі випадкові завади, які додатково маскують факт вбудовування сторонньої інформації всередину файлу.

1.2 Мережева модель OSI

OSI – Open Systems Interconnection является еталонною моделлю стеку мережевих протоколів, завдяки якій мережеві пристрої можуть безпосередньо взаємодіяти один з одним. Модель визначає сім рівнів взаємодії систем. Кожен рівень виконує певні функції при взаємодії цього типу.

Рисунок 1.3 – Мережева модель OSI

Модель являє собою сім рівнів, причому один має апаратну реалізацію, п’ять - мають програмну реалізацію та один здійснює зв’язок між ними та функціонує і апаратно, і програмно.

Рівень додатків (Application layer) – це набір протоколів, при використанні котрих, користувачі мережі отримують доступ до ресурсів, а саме файлів, принтерів або вебсторінок, а також організує спільну роботу, таку, як електронна пошта. Спеціальні елементи рівня додатків забезпечують сервіс для більшості прикладних додатків, таких як програми пересилки файлів та емуляції терміналів. У моделі OSI додаток, якому потрібно виконати певне завдання, посилає певні дані у вигляді датаграми на прикладний рівень. Одна з основних завдань цього рівня – визначити, як саме потрібно обробляти запит прикладної програми, іншими словами, який вигляд повинен прийняти даний запит.

Представницький рівень (Presentation layer) – забезпечує, щоб інформація, яка передається рівнем додатків, була зрозуміла рівню додатків в іншій системі. У випадках необхідності рівень представлення в момент передачі виконує перетворення форматів даних в певний уніфікований формат представлення, а в момент прийому повідомлення виконує зворотне перетворення. Таким чином, рівні додатків можуть подолати деякі відмінності в представленні даних. Така ситуація може виникнути в мережі з не однотипними комп'ютерами, яким необхідно обмінюватися даними. Так, в полях баз даних інформація представлена у вигляді літер або цифр, але досить часто що і у вигляді графічного зображення. Обробляти ці дані потрібно за особливими правилами, такими як числа з плаваючою комою.

Рівень сеансів (Session layer) – забезпечує управління активною сесією для того, щоб визначати, яка зі сторін на даний момент є активною, а також надає засоби синхронізації. Останні дозволяють вставляти контрольні точки в достатньо довгі передачі, щоб у разі помилки можна було повернутися до останньої контрольної точки, замість того щоб починати передачу спочатку. На практиці деякі додатки використовують рівень сеансів, але загалом він рідко реалізується. Рівень сеансів керує передачею інформації між прикладними процесами, координує їх прийом, передачу та в наслідку видачу одного сеансу зв'язку. Крім того, рівень сеансів містить додатково функції управління паролями, управління діалогом, синхронізації та скасування зв'язку в робочій сесії передачі після збою

внаслідок помилок в нижчих рівнях. Функції рівня складаються в управлінні зв’язку між декількома додатками, що працюють на різних робочих станціях. Це відбувається у вигляді повністю структурованого діалогу. Серед цих функцій потрібно зазначити функції створення робочої сесії, управління передачею та прийомом пакетів з повідомленнями під час сесії та її завершення.

Транспортний рівень (Transport layer) – при передачі повідомлення пакети можуть бути спотворені, або загублені. Хоча деякі додатки мають власні засоби обробки помилок, в більшості, існують такі, які вважають за доцільніше - відразу мати справу з надійним з’єднанням. Робота транспортного рівня заключається на забезпеченні додаткам передачу даних з високим ступенем надійності, яка необхідна для їх нормальної роботи. Модель визначає п’ять класів сервісу, що може надавати транспортний рівень. Ці види сервісу, загалом, відрізняються якістю наданих послуг, а саме терміновістю, можливістю відновлення зв’язку, наявністю мультиплексування декількох з’єднань між різними протоколами через загальний транспортний протокол, а також здатність до виявлення та виправлення помилок передачі, конкретно - спотворень, втрати та дублювання пакетів. Транспортний рівень визначає адресацію додатків на фізичних пристроях в мережі. Цей рівень гарантує доставку блоків інформації адресатам та управляє цією доставкою. Його головним завданням є забезпечення ефективних, зручних та надійних форм передачі інформації між системами. Коли в процесі обробки є більш, ніж однин пакет - транспортний рівень контролює черговість їх проходження. Якщо проходить дублікат прийнятого раніше повідомлення, то даний рівень розпізнає його і, відповідно, ігнорує повідомлення.

Мережевий рівень (Network layer) – встановлює зв’язки в обчислювальних мережах між двома системами та забезпечує прокладку віртуального каналу між ними. Віртуальний канал являє собою функціонування декількох компонентів мережі, що створює взаємодіючим компонентами ілюзію створення між ними потрібного тракту. Крім цього, мережевий рівень, також, повідомляє транспортному рівню про наявність помилок. Протокольним блоком даних мережевого рівня являються пакети. У них містяться фрагменти повідомлення.

Мережевий рівень відповідає за їх адресацію у мережі та доставку відповідно. Процес знаходження найкращого шляху для передачі називають маршрутизацією, окрім того, ця задача є головним завданням мережевого рівня. Ця проблема ускладнюється ще і тим, що найкоротший шлях далеко не завжди є найкращий. Найчастіше основним критерієм при виборі маршруту є час передачі даних по цьому маршруту, що залежить від пропускної здатності каналу зв'язку та інтенсивності потоку трафіка, що може змінюватися з часом. Деякі алгоритми маршрутизації намагаються приймати рішення на основі зміни навантаження, а інші приймають рішення, грунтуючись на середніх показниках за деякий визначений час. Вибір маршруту може здійснюватися ще і за іншими критеріями, такими як, надійність передачі. Протокол канального рівня забезпечує доставку даних між будь-якими вузлами тільки в мережі з певною визначеною топологією. Це жорстке обмеження, що не дозволяє будувати мережі з розрізненою структурою, наприклад, мережі, що об'єднують кілька мереж підприємства в єдину мережу, або високонадійні мережі, що мають надлишкові зв'язку між вузлами. Таким чином, всередині мережі доставка даних регулюється канальним рівнем, а ось доставкою даних між мережами займається мережевий рівень. При організації доставки пакетів на мережевому рівні використовується поняття номер мережі. У цьому випадку адреса одержувача складається з номера мережі та номера комп'ютера в цій мережі. Мережі з'єднуються між собою спеціальними пристроями, званими маршрутизаторами. Маршрутизатор – це пристрій, який збирає інформацію про топологію між мережевих з'єднань та на її підставі пересилає пакети мережевого рівня в мережу призначення. Для передачі повідомлення від відправника, що знаходиться в одній мережі, отримувачу, що знаходиться в іншій мережі, потрібно здійснити деяку кількість транзитних передач між мережами, щоразу, вибираючи відповідний маршрут. Таким чином, маршрут це послідовність маршрутизаторів, через які проходить пакет. Мережевий рівень відповідає за розподіл користувачів на групи та маршрутизацію пакетів на основі перетворення MAC-адрес в мережеві адреси.

Мережевий рівень забезпечує також повну передачу пакетів на транспортний рівень.

Канальний рівень (Data-link layer) – одиницею передачі на цьому рівні інформації є кадри. Кадри є логічно організованими структурами, в які можна інкапсулювати дані. Завдання канального рівня передавача кадрів від мережевого рівня до фізичного. На фізичному рівні просто пересилаються електричні сигнали, що є бітами. При цьому не враховується, що в деяких мережах, в яких лінії зв'язку використовуються кількома парами взаємодіючих робочих станій, фізичне середовище передачі може бути зайнятим. адже основним завданням канального рівня є перевірка доступності середовища передачі. Іншим, не менш важливим завданням канального рівня є реалізація механізмів достовірного виявлення та корекції помилок. Канальний рівень, також, забезпечує коректність передачі кожного кадру, вміщуючи спеціальну послідовність біт, на початку та кінці кожного кадру, щоб відзначити його, а також визначає контрольну суму, підсумовуючи всі байти кадру певним способом та додаючи контрольну суму до кадру. Коли кадр приходить, одержувач також обчислює контрольну суму отриманих даних та порівнює результат з контрольною сумою з кадру. Якщо вони збігаються, тоді кадр вважається правильним та приймається. У разі якщо контрольні суми не збігаються, то фіксується помилка. Цей рівень повинен визначити, початок та закінчення блоку, а також звісно виявляти помилки передачі. На цьому ж рівні визначаються правила використання фізичного рівня вузлами мережі. Електричне уявлення даних розпізнаються на цьому та тільки на цьому рівні. Тут виявляються та виправляються помилки. Канальний рівень забезпечує створення, передачу та прийом кадрів даних. Цей рівень обслуговує запити мережевого рівня та використовує сервіс фізичного рівня для прийому та передачі пакетів. Специфікація IEEE 802.x ділить канальний рівень на два підрівні:

- Logical Link Control;
- Media Access Control.

Підрівень LLC забезпечує обслуговування мережного рівня життя та пов’язані з передачею та прийомом призначених для користувача повідомлень, та MAC контроль доступу до середовища. Підрівень MAC регулює доступ до поділюваного фізичного середовища та управляє доступом до каналу зв’язку. Підрівень LLC знаходиться вище підрівня MAC. Канальний рівень визначає доступ до середовища та управління передачею у вигляді процедури передачі даних по каналу. При великих обсягах переданих блоків даних канальний рівень ділить їх на кадри та передає кадри у вигляді послідовностей. При отриманні кадрів рівень формує з них передані блоки даних. Розмір блоку даних залежить від способу передачі, якості каналу, по якому він передається. У локальних мережах протоколи канального рівня використовуються комп’ютерами, мостами, комутаторами та маршрутизаторами. У комп’ютерах функції канального рівня реалізуються спільними зусиллями мережевих адаптерів та їх драйверів.

Фізичний рівень (Phisical layer) – призначений для роботи з фізичними засобами з’єднання, що є сукупністю фізичного середовища, апаратних та програмних засобів, що забезпечує передачу сигналів між системами. Фізичне середовище являє собою матеріальну субстанцію, через яку здійснюється передача сигналів. Воно є основою, на якій будуються фізичні засоби з’єднання. Як фізичне середовище широко використовуються, в основному, метали, оптика та кварц. Фізичний рівень складається з підрівня стикування з середовищем та підрівня перетворення передачі. Перший з них забезпечує сполучення потоку даних з використовуваним фізичним каналом зв’язку. Другий здійснює перетворення, пов’язані з застосовуваними протоколами. Даний рівень забезпечує фізичну взаємодію з каналом передачі даних, а також описує процедури передачі сигналів в канал та отримання їх з каналу. На цьому рівні визначаються усі параметри для фізичної зв’язку в системах. Фізичний рівень отримує пакети даних від канального рівня та перетворює їх в оптичні або електричні сигнали, відповідні нулю та одиниці. Ці сигнали посилаються через середовище передачі на прийомний вузол. Механічні, електричні або оптичні властивості середовища передачі включають:

- Тип кабелів та роз’ємів
- Розпінування контактів в роз’ємах
- Схему кодування сигналів для 0 і 1 [5]

1.3 Стек протоколів TCP/IP

TCP/IP – це стандарт стеку протоколів, що використовується для передачі інформації в комп’ютерних мережах. OSI – є еталонною моделлю побудови мережі, TCP/IP – є реально працюючою технологією. Проте він реалізований на основі OSI, а також є найпопулярнішим на даний час стандартом для передачі даних. [6]

Рівень доступу до мережі (Network Acces layer) відповідає фізичному і канальному рівням моделі OSI. Цей рівень в не регламентується, але підтримує всі популярні стандарти фізичного і канального рівня локальних мереж:

- Ethernet;
- Token Ring;
- FDDI;
- Fast Ethernet.

А для глобальних мереж – протоколи peer-to-peer з’єднань SLIP і PPP, протоколи територіальних мереж з комутацією пакетів, таких як X.25, Frame relay. Розроблено також спеціальна специфікація, що визначає використання технології ATM, як транспорту канального рівня. Зазвичай при появі нової технології локальних або глобальних мереж вона швидко включається в стек TCP/IP за рахунок розробки відповідного RFC, що визначає метод інкапсуляції пакетів IP у її кадри.

Рівень мережі Інтернет (Internet) – це рівень взаємодії між локальними мережами, що займається передачею пакетів з використанням різних транспортних технологій локальних мереж і ліній спеціального зв’язку. В якості основного протоколу мережевого рівня, на основі моделі OSI, в стеку використовується протокол IP, що проектувався спочатку як протокол передачі

пакетів в мережах, що складаються з великої кількості локальних мереж, об’єднаних і локальними, і глобальними зв’язками.

Рисунок 1.4 – Структура стеку протоколів TCP/IP

.Тому протокол IP добре працює в мережах зі складною топологією, раціонально використовуючи наявність у них підсистем і ощадливо витрачаючи пропускну здатність низько швидкісних ліній зв’язку. Протокол IP є датаграмним протоколом, тобто він не дає гарантій доставки пакетів до вузла призначення, але намагається цього досягнути. До рівня взаємодії між мережами відносяться всі

протоколи, пов’язані із створенням та зміною таблиць маршрутизації, такі як протоколи маршрутизації RIP, OSPF та EIGRP, а також протокол управляючих та інформаційних повідомлень ICMP. Останній протокол призначений для обміну повідомленнями про помилки між маршрутизаторами мережі та вузлами, тобто джерелами пакетів. За допомогою спеціальних службових пакетів ICMP повідомляється про неможливість доставки пакета, про перевищення часу життя або тривалості збору пакета з фрагментів, про ненормальні величини параметрів, а також про зміну маршруту пересилання та типу обслуговування, про стан системи.

Транспортний рівень (Transport layer) може вирішувати проблему негарантованої доставки повідомлень, а також гарантувати правильну послідовність доставки даних. Транспортні протоколи визначають, для якого саме додатка призначені ці дані. Протоколи автоматичної маршрутизації, логічно представлені на цьому рівні, оскільки працюють поверх IP, насправді є частиною протоколів мережевого рівня, такого як OSPF. TCP є гарантованим транспортним механізмом з встановленням попереднього з’єднання, що надає додаткам надійну передачу потоку даних, що дає впевненість у надійності та безпомилковості отриманих даних, запитує дані повторно в разі втрати та усуває дублювання даних. TCP дозволяє регулювати навантаження на мережу, а також зменшувати час очікування даних під час передачі на великі відстані. Більш того, TCP гарантує, що отримані дані були відправлені точно в такій же послідовності. У цьому його головна відмінність від UDP. UDP – протокол передачі датаграм без встановлення з’єднання. Також його називають протоколом ненадійним, в розумінні неможливості упевнитися в точній доставці повідомлення одержувачу, а також можливої зміни порядку доставки пакетів. У додатках, що вимагають гарантованої передачі даних, використовується протокол TCP. UDP зазвичай використовується в таких додатках, де присутнє потокове відео та комп’ютерні ігри, де допускається втрата пакетів, а повторний запит проблематичний або невиправданий, або в додатках, як запити до DNS, де створення з’єднання займає

більше ресурсів, ніж повторна відправка. TCP та UDP використовують для визначення протоколу верхнього рівня число, що називається портом.

Рівень додатків (Aplication layer) – реалізовує верхні три рівня моделі OSI. За довгі роки використання в мережах різних країн та організацій стек TCP/IP нагромадив велику кількість протоколів та сервісів прикладного рівня. До них відносяться такі широко використовувані протоколи, як протокол копіювання файлів FTP, протокол емуляції терміналу telnet, поштовий протокол SMTP, використовуваний в електронній пошті мережі Internet, гіпертекстові сервіси доступу до вилученої інформації, такі як WWW та багато інших. Зупинимося дещо докладніше на деяких з них. Протокол передачі файлів FTP реалізує віддалений доступ до файлу. Для того, щоб забезпечити надійну передачу, FTP використовує в якості транспорту протокол TCP. Крім пересилки файлів протокол FTP має й інші можливості такі, як користувачеві надається можливість інтерактивної роботи з віддаленим комп'ютером, наприклад, він може вилучати вміст його каталогів. Окрім того, FTP виконує аутентифікацію користувачів. Перш, ніж отримати доступ до файлу, відповідно до протоколу користувачі повинні повідомити своє ім'я та пароль. Для доступу до публічних каталогів FTP-архівів Internet парольна аутентифікації не потрібно, та її обходять за рахунок використання для такого доступу заздалегідь зумовленого анонімного імені користувача. Протокол FTP пропонує найбільш широкий асортимент послуг для роботи з файлами, проте він є найскладнішим для програмування. Додатки, яким не потрібні всі можливості FTP, можуть використовувати інший, більш економний протокол найпростіший протокол пересилки файлів TFTP. Цей протокол реалізує тільки передачу файлів, причому як транспорт використовується більш простий, ніж TCP, протокол UDP. Протокол емуляції терміналу telnet забезпечує передачу потоку байтів між процесами, а також між процесом та терміналом. Найбільш часто цей протокол використовується для емуляції терміналу віддаленого комп'ютера. При використанні сервісу telnet користувач фактично керує віддаленим комп'ютером так само, як та локальний користувач, адже такий вид доступу вимагає гарного захисту. адже сервери telnet

завжди використовують як мінімум аутентифікацію за паролем, а іноді та більш потужні засоби захисту. Протокол SNMP використовується для організації мережевого управління. Спочатку протокол SNMP був розроблений для віддаленого контролю та управління маршрутизаторами Internet, які традиційно часто називають також шлюзами. Окрім того існує ще велика кількість протоколів що використовують і протокол із встановленням з’єднання і протокол без встановлення з’єднання, тобто TCP та UDP [7]

1.4 Протокольні блоки даних

При переході між рівнями в стеці протоколів відбувається інкапсуляція при переході на нижчі рівні і декапсуляція при переході на верхні рівні. Відповідно кожному рівні додається або вилучається заголовок цього рівня. Заголовки несуть службову інформацію, яка дозволяє доставити повідомлення в місце призначення або повідомить що це неможливо.

1.4.1 Транспортний рівень

Транспортний рівень забезпечує і контроль, що необхідний для збору сегментованих даних. Він відслідковує комунікацію між додатками, сегментує дані і керує кожним фрагментом.

TCP надає послуги з доставки даних, перебуваючи в транспортному рівні моделей OSI і TCP/IP. TCP вимагає встановленого з’єднання до початку передачі даних, адже його називають підпротоколом, орієнтованим на з’єднання. TCP забезпечує забезпечення встановлення з'єднання перед передачею.

В залежності від того, які вимоги до передачі інформації використовується ненадійний протокол передачі даних – UDP, або надійний протокол передачі – TCP. Відповідно заголовок транспортного рівня для них відрізняється.

Порт відправника – ідентифікує додаток на робочій станції, що створює запит на обмін інформацією з іншим додатком, займає 16 біт,.

Порт отримувача – ідентифікує додаток на робочій станції, до якого надходить запит на обмін інформацією з іншим додатком, займає 16 біт.

Рисунок 1.5 – Структура заголовку UDP

Довжина дейтаграми – визначає довжину сформованого сегмента разом із заголовком.

Рисунок 1.6 – Структура заголовку TCP

Контрольна сума – використовується для перевірки на наявність помилок в сегменті.

Порядковий номер – кожен байт в відправленому сегменті нумерується для правильного збору сегментів в єдиний потік. Крім того, за допомогою номера контролюється весь потік даних і виправляються пошкоджені сегменти.

Номер підтвердження – потрібен для того, щоб переконатися, що всі передані сегменти досягли адресата.

Прапори – спеціальні біти, що вказують на стан активної сесії. Наприклад, це можуть бути запит на встановлення з'єднання або запит на розрив з'єднання.

Розмір вікна – визначає кількість байт що можливі для відправки за один раз.

Приймаючий вузол відправить підтвердження тільки для останнього байта в сегменті. Таким чином не потрібно відправляти підтвердження за кожен байт, економлячи тим самим час і канал зв'язку. [8]

1.4.2 Мережевий рівень

Більшість мережевих технологій передачі даних використовують пакети для передачі даних від вихідного пристрою до пристрою призначення. Протокол IP не є винятком. IP-пакети - це найважливіші та основні компоненти протоколу. Вони є структурами, які несуть дані під час передачі. У них також є заголовок, який містить інформацію, яка допомагає їм знайти свій шлях і зібрати після передачі.

Дві основні функції протоколу IP - маршрутизація та адресація. Для маршрутизації пакетів до та з машин у мережі IP використовує IP-адреси, які переносяться у пакетах.

На мережевому рівні утворюються пакети. IP-пакет складається із заголовка й поля даних. Заголовок, як правило, має довжину 20 байт.

Номер версії – займає 4 біти, вказує версію протоколу IP. Зараз використовується версія 4, та готується перехід на версію 6.

Довжина заголовка IP-пакета займає 4 біта та вказує значення довжини заголовка, вимірюване в тридцяти двох бітових словах. Зазвичай заголовок має

довжину в 20 байт, але при можливому збільшенні обсягу службової інформації ця довжина може бути збільшена за рахунок додаткових байт у полі Опції.

Рисунок 1.7 – Структура IP-пакета

Тип сервісу займає один байт та задає пріоритетність пакета та вид критерію вибору маршруту. Перші три біти цього поля утворять підполе пріоритету пакета. Пріоритет може мати значення від найнижчого – 0, до найвищого – 7. Маршрутизатори та комп’ютери можуть брати до уваги пріоритет пакета та обробляти більш важливі пакети в першу чергу. Поле Тип сервісу містить також три біти, що визначають критерій вибору маршруту. Реально вибір здійснюється між трьома альтернативами: малою затримкою, високою вірогідністю та високою пропускною здатністю. Встановлений біт D говорить про те, що маршрут повинен вибиратися для мінімізації затримки доставки даного пакета, біт T – максимізація пропускної здатності, а біт R – максимізація надійності доставки. У багатьох мережах поліпшення одного із цих параметрів пов’язане з погіршенням іншого, крім того, обробка кожного з них вимагає додаткових обчислювальних витрат. адже рідко, є потрібним установлювати

одночасно хоча б два із цих трьох критеріїв вибору маршруту. Зарезервовані біти мають нульове значення.

Загальна довжина займає 2 байти та означає загальну довжину пакета з урахуванням заголовка та поля даних. Максимальна довжина пакета обмежена розрядністю поля, що визначає цю величину, та становить 65 535 байт, однак у більшості комп'ютерів та мереж настільки великі пакети не використовуються. При передачі по мережах різного типу довжина пакета вибирається з урахуванням максимальної довжини пакета протоколу нижнього рівня, що несе IP-пакети. Якщо це кадри Ethernet, то вибираються пакети з максимальною довжиною в 1 500 байт, що вміщаються в поле даних кадру Ethernet. У стандарті передбачається, що всі хости повинні бути готові приймати пакети аж до 576 байт довжиною. Хостам рекомендується відправляти пакети розміром більш ніж 576 байт, тільки якщо вони впевнені, що приймаючий хост або проміжна мережа готові обслуговувати пакети такого розміру.

Ідентифікатор пакета займає 2 байти та використовується для розпізнавання пакетів, що утворилися шляхом фрагментації вихідного пакета. Всі фрагменти повинні мати однакове значення цього поля.

Поле прапорів займає 3 біти та містить налаштування, пов'язані із фрагментацією. Встановлений біт DF забороняє маршрутизатору фрагментувати даний пакет, а встановлений біт MF говорить про те, що даний пакет є проміжним фрагментом. Біт, що залишився, зарезервований.

Зсув фрагмента займає 13 біт та задає зсув у байтах поля даних цього пакета від початку загального поля даних вихідного пакета, що був фрагментований. Використовується при зборці або розбиранні фрагментів пакетів при передачах їх між мережами з різними величинами MTU. Зсув повинен бути кратним 8 байт.

Час життя займає 1 байт та означає граничний строк, протягом якого пакет може переміщатися по мережі. Час життя даного пакета виміряється в секундах та задається джерелом передачі. На маршрутизаторах та в інших вузлах мережі після закінчення кожної секунди з поточного часу життя віднімається одиниця; одиниця віднімається та у адже випадку, коли час затримки менше секунди.

Оскільки сучасні маршрутизатори рідко обробляють пакет довше, ніж за одну секунду, то час життя можна вважати рівним максимальному числу вузлів, які дозволено пройти даному пакету до того, як він досягне місця призначення. Якщо параметр часу життя стане нульовим до того, як пакет досягне одержувача, цей пакет буде знищений. Час життя можна розглядати як годинний механізм самознищення. Значення цього поля змінюється при обробці заголовка IP-пакета.

Ідентифікатор протоколу верхнього рівня займає один байт та вказує, якому протоколу верхнього рівня належить інформація, розміщена в полі даних пакета. Значення ідентифікаторів для різних протоколів приводяться в документі RFC.

Контрольна сума займає 2 байти та розраховується тільки по заголовку. Оскільки деякі поля заголовка міняють своє значення в процесі передачі пакета по мережі, контрольна сума перевіряється та повторно розраховується при кожній обробці IP-заголовка. Контрольна сума – 16 біт підраховується, як доповнення до суми всіх шістнадцяти бітових слів заголовка. При обчисленні контрольної суми значення самого поля «контрольна сума» встановлюється в нуль. Якщо контрольна сума невірна, то пакет буде відкинутий, як тільки помилка буде виявлена.

IP-адреса джерела та IP-адреса призначення мають довжину 32 біта.

Поле опції є необов’язковим та використається звичайно тільки при налагодженні мережі. Механізм опцій надає функції керування, які необхідні або просто корисні при певних ситуаціях, однак він не потрібний при звичайних комунікаціях. Це поле складається з декількох полів, кожне з яких може бути одним з восьми визначених типів. У цих полях можна вказувати точний маршрут проходження маршрутизаторів, реєструвати прохідні пакетом маршрутизатори, поміщати дані системи безпеки, а також тимчасові оцінки. Адже число полів може бути довільним, то наприкінці поля опції повинне бути додане трохи байт для вирівнювання заголовка пакета по 32-бітовій границі.

Поле Вирівнювання використовується для того, щоб переконатися в тому, що IP-заголовок закінчується на 32-бітній границі. Вирівнювання виконується нулями. [9]

1.4.2 Канальний рівень

Вже відомо, що інкапсульовані дані, визначені рівнем мережевого доступу, називаються кадром Ethernet. Кадр Ethernet починається із заголовка, який містить серед інших даних вихідний та кінцевий MAC-адреси. Середня частина кадру – це фактичні дані. Кадр закінчується полем під назвою послідовності перевірки кадру Структура кадру Ethernet визначена в стандарті IEEE 802.3.

На канальному рівні інкапсуляція відбувається у кадр Ethernet, що дозволяє передавати повідомлення в межах однієї локальної мережі.

Поле преамбули – сім байт, що використовується як синхронізуючі, і мають значення – 10101010.

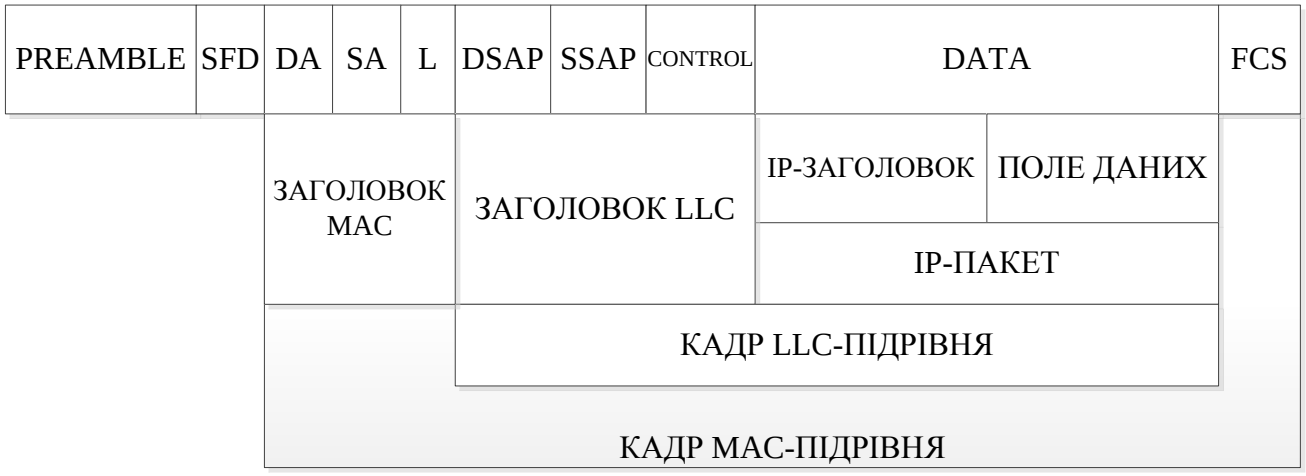


Рисунок 1.8 – Структура кадру Ethernet

Початковий обмежувач кадру – один байт, що має формат – 10101011

Адреса призначення – має довжину 6 байт. Перший біт старшого байта ознака того, чи є адреса індивідуальним або груповим. Групова адреса призначена для всіх вузлів мережі або групі вузлів, які налаштовані, як члени однієї групи. В випадку, якщо адреса складається з всіх одиниць, то він призначений всіх вузлів

мережі і називається широкомовним. Другий біт першого байта визначає спосіб призначення адреси нуль – централізовано, одиниця – локально. На практиці адреси майже завжди призначаються централізовано комітетом IEEE, що розподіляє між виробниками техніка унікальні ідентифікатори. Цей ідентифікатор міститься в трьох старших байта адреси, три молодших байта назначаються виробниками обладнання.

Адреса джерела – так само має довжину 6 байт, при цьому перший біт дорівнює 0.

Довжина – визначає довжину поля даних в кадрі, та має розмір в 2 байта,.

Поле даних може містити до 1500 байт, але за стандартом Ethernet мінімальна довжина кадру встановлена 46 байт, але якщо розмір поля даних є меншим 46 байт використовується наступне поле – поле заповнення. [10]

2 АНАЛІЗ МОЖЛИВИХ РІШЕНЬ ЗАДАЧІ ТА ОБГРУНТУВАННЯ ПОДАЛЬШОГО НАПРЯМУ

2.1 Аналіз полів протокольних блоків даних

Основною задачею дослідження є аналіз полів на можливість використання їх у якості контейнерів для передачі прихованих повідомлень. Також потрібно оцінити чи є можливість їх використання не пошкодивши функціонал блоку даних.

Окрім того потрібно знайти можливі методи одночасної передачі в декількох полях, для підвищення якості та об’ємів інформації що буде передаватись.

2.1.1 Заголовок канального рівня

За стандартом IEEE 802.3 мінімальний розмір кадру Ethernet складає 64 байта, а максимальний 1518 байт.

Поле преамбули являє собою сім синхронізуючих байт 10101010, а поле початкового обмежувачу кадру один байт 10101011, але оскільки вони незмінні їх не можна використовувати в якості контейнера.

Адресу призначення також не можна модифікувати оскільки в ній буде вказано адресу, де буде знаходитись приймач повідомлення.

Адреса джерела може модифікуватись, але це має деякі особливості. По-перше MAC-адреси мають певну структуру не є випадковими, адже при перехоплені трафіку підміна адрес буде помітна. По-друге буде втрачено будь-яку можливість підтвердження доставки повідомлення.

Поля довжина і контрольна сума не можуть використовуватись адже їх зміна може призвести до того, що кадр буде вважатись пошкодженим і буде відкинутий.

Отже, можна зробити висновок, що за допомогою кадру Ethernet можна передати максимум 6 байт інформації.

І крім цього, не слід забувати, що кадри передаються на канальному рівні і адже адреса джерела буде змінена при проходженні через перший комутатор. адже використання кадрів Ethernet для передачі прихованого повідомлення не є можливим.

2.1.2 Заголовок мережевого рівня

На відміну від кадрів пакети можуть передаватись на достатньо великі відстані без зміни, адже є непоганим варіантом для контейнера.

Поля номер версії і довжина заголовка є службовими і їх зміна призведе до порушення проходження пакету по мережі, що не дає змоги використовувати їх.

Тип сервісу займає один байт. Пріоритет пакету встановлюється першими трьома бітами. Окрім нього є біти затримки, пропускної здатності і надійності. Всі вони можуть використовуються для конфігурування проходження пакету по мережі. Потенційно їх можна змінити, але це може призвести до того, що при передачі потоку пакети можуть дійти до приймача в неправильному порядку. Але ця проблема може бути вирішена встановленням затримки між відправленням пакетів. Два останніх біта є зарезервованими і можуть використовуватись довільно.

Загальна довжина не піддається модифікації у зв’язку з тим, що від нього залежить проходження пакета.

Ідентифікатор пакета використовується, щоб допомогти зібрати пакет з декількох можливих фрагментів. Коли дані надсилаються по мережі, вони розбиваються на невеликі розділи, які обгорнуті цими пакетами. Мережі IP, такі як Інтернет, як правило, не захищені, адже пакети можуть бути втрачені, затримані і можуть прийти в неправильному порядку. Після того, як вони прибули до пункту призначення, ідентифікаційний тег допомагає ідентифікувати пакет і зібрати дані назад у початковий вигляд.

Поле прапорів займає 3 біти й містить ознаки, пов’язані із фрагментацією. При її відсутності три біта можуть використовуватись довільно.

Зсув фрагмента займає 13 біт і може використовуватись для передачі прихованого повідомлення при встановленому прапорі на заборону фрагментації.

Час життя займає 1 байт і число, яке вказує, скільки переходів пакет може зробити, перш ніж він загине. Зазвичай на кожному маршрутизаторі аналізується пакет і на основі інформації, присутньої в цьому маршрутизаторі на інших сусідніх маршрутизаторах, робиться вибір, який маршрут найкращий. Потім пакет пересилається до наступного маршрутизатора. У цій конфігурації пакет може цілком обійтись. Існує також затоплення як інший метод, який передбачає надсилання копії пакету до кожного сусіднього маршрутизатора; тоді лише цільова машина споживає пакет. Інші пакети продовжуватимуть роумінг. TTL - це число, як правило, 255, яке зменшується щоразу, коли пакет передає маршрутизатор. Таким чином, надлишки пакетів в кінцевому рахунку загинуть, як тільки TTL досягне нуля.

Поле контрольна сума – використовується для виявлення та виправлення помилок під час передачі пакету. Дані в пакеті подаються в математичний алгоритм, що призводить до суми, яка надсилається разом з даними в пакеті. Після отримання ця сума знову обчислюється за допомогою того ж алгоритму. Якщо вона збігається з початковою сумою, дані є хорошими, в іншому випадку вони вважаються пошкодженими, а пакет відкидається.

Ідентифікатор протоколу верхнього рівня, контрольна сума, IP-адреса джерела і IP-адреса призначення не можуть використовуватись для передачі.

Поле вирівнювання використовується для вирівнювання по 32-х бітній границі. Тобто максимальна довжина вбудованого повідомлення може бути рівна 31 біту.

На основі аналізу було визначено, які поля можуть бути використані в якості контейнерів і оформлені у вигляді таблиці.

У таблиці 2.1 наведено загальну кількість усіх можливих розмірів контейнера. Для більшої доцільності слід використовувати деякі з них одночасно. Але слід врахувати, що не всі з них можна використовувати одночасно.

Таблиця 2.1 – Порівняння розмірів контейнерів

Контейнер	Можлива кількість інформації, що передається.
Зарезервовані біти в полі тип сервісу	2 біта
Поле тип сервісу	8 біт
Поле ідентифікатор пакета	16 біт
Поле прапорів	3 біта
Поле прапорів без прапору заборони фрагментації	2 біта
Поле зсув фрагмента	13 біт
Поле час життя	8 біт
Вирівнювання	31 біт

Для максимально швидкої передачі поле тип сервісу не можна заміняти. адже буде використано тільки 2 зарезервованих біта. Ідентифікатор пакета може використовуватись при відсутності фрагментації, тобто коли не використовується поле прапорів, а саме перший його біт. Тоді зі згаданим вище отримуємо 36 біт. Поле зсув фрагмента використовується лише при фрагментації, а поле пов’язане з нею вже встановлено адже використовуємо його безперешкодно. Отримуємо 49 біт. Поле час життя враховувати не буде у всіх комбінаціях, оскільки надійність використання цього поля, як контейнера, дуже низька. Поле вирівнювання приймаємо за максимально велике можливе значення, тобто 31 біт. Загалом отримуємо 80 біт.

При збереженні швидкості передачі, але без заборони фрагментації, можливість використовуватись поле ідентифікатора пакета можливе. Проте ми втрачаємо можливість використовувати поле зсуву фрагменту, отримуючи можливість використовувати поле прапорів повністю. адже в сумі отримуємо 68 біт.

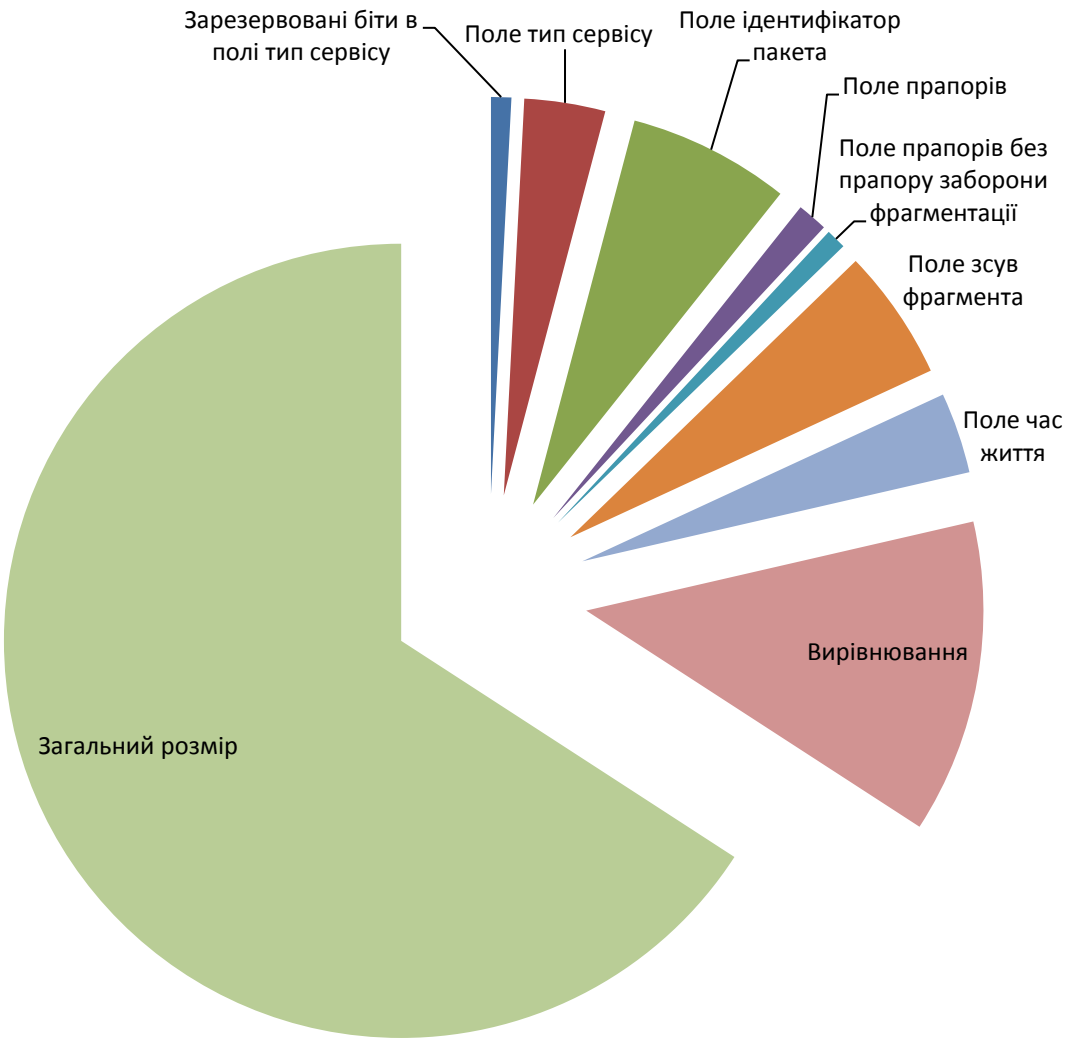


Рисунок 2.1 – Порівняння розміру контейнерів із загальним розміром заголовку

При затримках між відправкою пакетів і при забороні фрагментації ми додатково отримаємо використання поля тип сервісу повністю. Загалом отримаємо 86 біт.

При затримках між відправкою пакетів і при дозволеній фрагментації, по аналогії з попередніми методами виходить 74 біта.

Таблиця 2.2 – Максимальна кількість переданої інформації

Метод передачі	Максимальний об’єм інформації, можливої для передачі в одному пакеті
Швидкий метод із забороною фрагментації	80 біт
Швидкий метод без заборони фрагментації	68 біт
Метод із затримками відправки пакетів та з забороною фрагментації	86 біт
Метод із затримками відправки пакетів та без заборони фрагментації	74 біта

Порівнявши ці методи можна сказати, що найбільший об’єм інформації в одному пакеті можливо передати за допомогою методу із затримками відправки пакетів та з забороною фрагментації.

2.1.3 Заголовок транспортного рівня

Транспортний рівень відповідає за доставку інформації адресату. Існує два протоколи передачі з підтвердженням доставки і без. Оскільки UDP не може гарантувати надійної доставки повідомлення, передача прихованого повідомлення за допомогою даного протоколу не є доцільним.

Тому в якості контейнера буде використовуватись TCP-заголовок.

Насамперед потрібно визначити, які з полів не можливо використовувати без порушення нормальної передачі. Оскільки на шляху проходження повідомлення можуть блокуватись деякі порти, перед передачею потрібно

перевірити які порти відкриті на місці призначення. адже порт отримувача змінювати не бажано.

Порядковий номер використовується для збору сегментів у єдине ціле і його використання не є можливим оскільки порушення цього поля не дозволить зібрати повідомлення в потік. Але збір сегментів в потік не є обов'язковим, оскільки приховане повідомлення може бути вилучено раніше, адже його можна модифікувати при умові, що пакет буде опрацьовуватись приймачем і його доставка до певного додатку не є обов'язковою.

Номер підтвердження можливий для використання в якості контейнера, якщо ініціюється передача потоку в певну кількість сегментів і підтвердження відправляється один раз після завершення передачі потоку.

Довжина заголовку і прапори не використовуються оскільки при їх зміні пакети будуть відкидатися як пошкоджені. Зміна інших полів таких, як розмір вікна, контрольна сума і вказівник важливості можуть порушити передачу сегмента і адже їх зміна не є можливою.

Тому, можна зробити висновок, що без додаткових маніпуляцій передача прихованого повідомлення на мережевому рівні можлива тільки при використанні поле номер підтвердження, як контейнер. А воно має довжину 32 біта.

Також при умові прийому повідомлення на рівні пакету можна використовувати поле порядковий номер. А це ще 32 біта.

Загалом отримуємо 8 байт.

2.1.4 Порівняння можливих методів передачі прихованого повідомлення

Для порівняння швидкості передачі буде використане відношення кількості прихованої інформації, що передається, до кількості інформації, що потрібно передати загалом. (2.1)

$$k = \frac{N_{\text{кн}}}{N_{\text{заг}}} \quad , \tag{2.1}$$

де k – коефіцієнт швидкості передачі,

$N_{кн}$ – кількість корисного навантаження,

$N_{заг}$ – загальний об’єм блоку даних.

А також кількість пакетів, що потрібно відправити для передачі 1 кілобайта корисного повідомлення. (2.2)

$$N_{п} = \left\lceil \frac{1\ кБайт}{N_{кн}} \right\rceil \quad (2.2)$$

де $N_{п}$ – кількість блоків даних,

$N_{кн}$ – кількість корисного навантаження.

Виконавши аналіз можливого використання полів заголовку канального рівня, зроблено висновок, що максимальна довжина прихованого повідомлення, що може передатись на канальному рівні без порушення нормальної передачі кадрів – 6 байт. Мінімальна довжина кадру Ethernet – 64 байта.

Коефіцієнт швидкості передачі:

$$k = \frac{6\ байт}{64\ байт} = 0{,}09735$$

$$N_{п} = \left\lceil \frac{1024\ байта}{6\ байт} \right\rceil = 171$$

Окрім того, передача кадрами можлива тільки в межах однієї локальної мережі, що робить неможливим передачу прихованого повідомлення за її межі.

Виконавши аналіз полів заголовку мережевого рівня було виділено чотири можливих метода передачі інформації. Крім того на мережевому рівні є можливість передавати інформацію в інші мережі. Мінімальна довжина пакету – 20 байт, тобто 160 біт.

Для швидкого методу із забороною фрагментацією:

$$k = \frac{80\ біт}{160\ біт} = 0{,}5$$

$$N_{\text{п}} = \left\lceil \frac{8192 \text{ біта}}{80 \text{ біт}} \right\rceil = 103$$

Для швидкого методу без заборони фрагментації:

$$k = \frac{68 \text{ біт}}{160 \text{ біт}} = 0{,}425$$

$$N_{\text{п}} = \left\lceil \frac{8192 \text{ біта}}{68 \text{ біт}} \right\rceil = 121$$

Для методу із затримками та з забороною фрагментації:

$$k = \frac{86 \text{ біт}}{160 \text{ біт}} = 0{,}5375$$

$$N_{\text{п}} = \left\lceil \frac{8192 \text{ біта}}{86 \text{ біт}} \right\rceil = 96$$

Для метод із затримками відправки пакетів та без заборони фрагментації:

$$k = \frac{74 \text{ біт}}{160 \text{ біт}} = 0{,}4625$$

$$N_{\text{п}} = \left\lceil \frac{8192 \text{ біта}}{74 \text{ біт}} \right\rceil = 111$$

Провівши аналіз полів заголовку транспортного рівня, було визначено, що передача за допомогою протоколу UDP є недоцільною, оскільки немає гарантії доставки повідомлення. А в протоколі TCP можливе використання 8 байт. Коефіцієнт швидкості передачі:

$$k = \frac{8 \text{ байт}}{64 \text{ байт}} = 0{,}125$$

$$N_{\text{п}} = \left\lceil \frac{1024 \text{ байта}}{8 \text{ байт}} \right\rceil = 128$$

Отримані дані було проаналізовано і систематизовано. Результат наведено у таблиці 2.3

Таблиця 2.3 – Порівняння методів передачі прихованої

Метод передачі	k	$N_{\text{п}}$
Заголовок канального рівня	0,09735	171
Заголовок мережевого рівня. Швидкий методу із забороною фрагментації	0,5	103
Заголовок мережевого рівня. Швидкий методу без заборони фрагментації	0,425	121
Заголовок мережевого рівня. Метод із затримками та з забороною фрагментації	0,5375	96
Заголовок мережевого рівня. Метод із затримками відправки пакетів та без заборони фрагментації	0,4625	111
Заголовок транспортного рівня	0,125	128

За таблицею 2.3 видно, що найменша кількість пакетів при передачі потребує метод із затримками та з забороною фрагментації.

Окрім того, не слід забувати, що однією з найважливіших задач стеганогріафії є приховування передачі інформації. Вище перераховані методи дозволяють передавати достатньо великі об’єми інформації, але при цьому при перехоплені пакету можна побачити що він модифікований. адже для більшої таємності слід використовувати лише ті поля, які створюються рандомно або псевдорандомно. Так в заголовку мережевого рівня це поле ідентифікатор пакета, оскільки воно використовується для збору пакету з фрагментів при фрагментації.

В заголовку транспортного рівня це поле порядкового номеру. Оскільки ці два контейнера можна використовувати одночасно отримуємо 6 байт. Визначимо для них коефіцієнт передачі і кількість пакетів для передачі одного кілобайта інформації:

$$k = \frac{6\ \text{байт}}{64\ \text{байт}} = 0{,}09735$$

$$N_{\text{п}} = \left\lceil \frac{1024\ \text{байта}}{6\ \text{байт}} \right\rceil = 171$$

Як бачимо, даний метод поступається перед всіма методами по швидкості, окрім методу передачі з використанням заголовку канального рівня. Але, тим не менше, він є максимально захищеним від перехоплення.

2.2 Додаткові можливості стеку протоколів TCP/IP

Як було вказано вище, однією з проблем при передачі прихованого повідомлення чи дійде воно до місця призначення. І адже є доцільним розглянути можливість більш достовірні методи передачі.

2.2.1 Потреба у фрагментації

У звіті TCPdump можливе повідомлення DF, що означає те, що був встановлений прапор який забороняє фрагментацію конкретної датаграми. Якщо датаграма з встановленим прапором Don’t Fragment має пройти через мережу в якій потрібна фрагментація, маршрутизатор виявляє проблему, відкидає пакет і надсилає відправнику ICMP-повідомлення про помилку.

Це повідомлення містить значення максимального розміру даних, що передаються, в мережі куди передається повідомлення.

Особливістю цього повідомлення є те, що воно має максимальний пріоритет і оскільки є службовим, шанс того, що його заблокує брандмауер дуже низька.

адже це повідомлення підходить як контейнер для передачі прихованого повідомлення.

2.2.2 Сканування портів

Сканування портів дозволяє отримати інформацію про робочі станції, включаючи імена пристроїв, IP-адреси, операційні системи, програмне забезпечення та служби, імена користувачів, групи та відкриті порти.

Існує чотири різних підходи до сканування мережевих портів:

- Горизонтальне сканування
- Вертикальне сканування
- Розподілене вертикальне сканування
- Розподілене горизонтальне сканування

При горизонтальному скануванні переглядається один і той же порт на декількох комп’ютерах, тобто на декількох IP-адресах. Атакуючий прагне знайти унікальні пристрої, що мають певні відкриті сервіси. Таким чином скануються певні порти на всіх пристроях, з різними IP-адресами в межах певного діапазону. Горизонтальне сканування є найбільш часто використовуваний в даний момент типом сканування портів.

Вертикальним скануванням називають процес, при якому сканується кілька портів на одному комп’ютері, тобто одна IP-адреса.

При розподіленому вертикальному скануванні кілька джерел послідовно сканують кілька портів на одному IP-адресу.

При розподіленому горизонтальному скануванні кілька джерел сканують один і той же порт на декількох IP-адресах послідовним чином. Під час розподіленого сканування часто змінюються IP-адреси, що робить їх виявлення досить складним завданням.

Сканування портів є досить непоганим методом передачі прихованого повідомлення, оскільки повідомлення є службовими і не блокуються пристроями.

А також з урахуванням кількості повідомлень при одному скануванні можна передавати великий об’єм інформації.

3 РЕАЛІЗАЦІЯ МЕТОДУ ПЕРЕДАЧІ ПРИХОВАНОГО ПОВІДОМЛЕННЯ

3.1 Огляд існуючих програмних продуктів та вибір основи для розробки

Стеганографія є дуже старим напрямком і зародилася ще до нашої ери. адже при появі будь-якої технології, її намагаються використати для передачі прихованих повідомлень. Звісно мережеві технології ця участь не минула.

Однією з найвідоміших програм, що використовує стеганографію Steganos Privacy Suite. Вона використовує приховані повідомлення при передачі даних через електронну пошту та при передачі файлів на віртуальний жорсткий диск.

Крім того існує чимало інших продуктів, що користуються методами стеганографії:

- Mr. Crypto
- Secret Letter
- appendX v0.4
- SecurEngine Professional
- bmpPacker
- Stegdetected
- Steghide
- F5
- Hide and Seek
- MP3Stego
- gifshuffle

Але всі вище перераховані програми не використовують протокольні блоки даних.

Окремо потрібно виділити один з методів прихованої передачі інформації, що по суті є стеганографією, що називається ICMP-тунель. Основною ідеєю цього метода є обмін звичайними службовими повідомленнями, такими як ping або

traceroute. Але окрім того в повідомлення іде вбудовування прихованого повідомлення.

Також слід згадати програму Covert_TCP, що є найбільш відомою програмою, яка використовує стеганографію. Більше того вона використовує стеганографію з використанням заголовків протокольних блоків даних, а саме заголовку транспортного рівня. Вона вбудовує в заголовок TCP повідомлення в поле порядковий номер.

За основу розробки програми буде обрана саме ця програма, оскільки вона реалізує частину методу, який був обраний для реалізації.

3.2 Розробка програмної реалізації методу

Реалізація методу передачі прихованого повідомлення складається з двох програм. Перша реалізовує серверну частину, що вбудовує повідомлення в заголовки і передає його на певну адресу. Друга частина реалізовує клієнтську частину, що прослуховує мережу і коли визначає пакети з прихованим повідомленням, то вилучає приховане повідомлення і виводить його на монітор.

3.2.1 Розробка серверної частини

Серверна частина програми повинна реалізовувати механізм створення IP-пакету та TCP-сегменту, їх забезпечення повного функціонування та максимально високу вірогідність доставки до отримувача, а саме клієнтської частини програми. Окрім повинно реалізовуватись вбудовування прихованого повідомлення в певні поля заголовків протокольних блоків даних IP-пакету та TCP-сегменту відповідно.

Оскільки TCP-сегмент буде сформований програмою, потрібно додатково вирахувати контрольну суму. Це потрібно для того, щоб сегмент не був відкинутим через неправильну хеш-суму.

Для цього потрібно створити структуру псевдозаголовка для того щоб сегмент був повноцінним:

```
struct pseudo_header {
   u_int32_t source_address;
   u_int32_t dest_address;
   u_int8_t placeholder;
   u_int8_t protocol;
   u_int16_t tcp_length;
};
```

Окремо створимо функцію, що буде вираховувати контрольну суму:

```
unsigned short csum(unsigned short *ptr,int nbytes) {
   register long sum;
   unsigned short oddbyte;
   register short answer;
   sum=0;
   while(nbytes>1) {
        sum+=*ptr++;
        nbytes-=2;
   }
   if(nbytes==1) {
        oddbyte=0;
        *((u_char*)&oddbyte)=*(u_char*)ptr;
        sum+=oddbyte;
   }
   sum = (sum>>16)+(sum & 0xffff);
   sum = sum + (sum>>16);
   answer=(short)~sum; return(answer);
}
```

Повідомлення, що буде передаватись, вводиться з клавіатури:

```
char payload[1024];
fgets(payload, 1024, stdin);
int length = strlen(payload);
Далі потрібно провести нуль-термінування введеної строки:

if (length > 0 && payload[strlen (payload) - 1] == ‘\n’)
 payload[strlen (payload) - 1] = ‘\0’;
```

Вираховуємо кількість пакетів, що потрібні для передачі введеного з клавіатури повідомлення:

```
int n = (length + 5)/6;
```

Далі запускається цикл на кількість ітерацій, таку як кількість пакетів потрібну для передачі.

Створюється сокет:

```
int s = socket (PF_INET, SOCK_RAW, IPPROTO_TCP);
if(s == -1) {
     perror("Failed to create socket"); exit(1);
}
```

Створюється пакет представлений побітово і ініціалізуємо його нулями:

```
char datagram[4096] , source_ip[32] , *pseudogram;
memset (datagram, 0, 4096);
```

Створюється структури для IP-заголовку та TCP-заголовку:

```
struct iphdr *iph = (struct iphdr *) datagram;
struct tcphdr *tcph =
(struct tcphdr *) (datagram + sizeof (struct iphdr));
struct sockaddr_in sin;
struct pseudo_header psh;
```

Наступним кроком переходимо до заповнення полів заголовку. Почнемо зі встановлення довжини, що може бути мінімально 5:

```
iph->ihl = 5;
```

Далі встановлюється версія протоколу:

```
iph->version = 4;
```

Встановлюються пріоритет. В даному випадку він є не важливим:

```
iph->tos = 0;
```

Визначається загальна довжина пакета:

```
iph->tot_len = sizeof (struct iphdr) + sizeof
 (struct tcphdr);
```

Додаємо в заголовок першу частину прихованого повідомлення:

```
iph->id = (6*i < length ? payload[6*i] << 8 : 0) +
 (6*i + 1 < length ? payload[6*i + 1] : 0);
```

Оскільки поле ID при нульовому значені замінюється на випадкове перевіряємо його на нуль, і якщо воно нульове заміняємо на одиницю:

```
if (iph->id == 0)
iph->id = 1;
```

Заповнюємо інші службові поля:

```
iph->frag_off = 0;
iph->ttl = 64;
iph->protocol = IPPROTO_TCP;
iph->check = 0;
iph->saddr = inet_addr ( source_ip );
iph->daddr = sin.sin_addr.s_addr;
```

Визначаємо контрольну суму заголовка:

```
iph->check = csum ((unsigned short *) datagram,
 iph->tot_len);
```

Далі переходимо до створення TCP:

```
tcph->doff = 5;
tcph->fin=0;
tcph->syn=1;
tcph->rst=0;
tcph->psh=0;
tcph->ack=0;
```

```
tcph->urg=0;
tcph->window = htons (5840);
tcph->check = 0;
tcph->urg_ptr = 0;
```

Наступним кроком вичислюємо контрольну суму:

```
psh.source_address = inet_addr( source_ip );
psh.dest_address = sin.sin_addr.s_addr;
psh.placeholder = 0;
psh.protocol = IPPROTO_TCP;
psh.tcp_length = 0;
int psize = sizeof(struct pseudo_header) +
 sizeof(struct tcphdr);
pseudogram = (char*)malloc(psize);
memcpy(pseudogram , (char*) &psh ,
 sizeof (struct pseudo_header));
memcpy(pseudogram + sizeof(struct pseudo_header) ,
 tcph, sizeof(struct tcphdr));
tcph->check = csum( (unsigned short*)
 pseudogram , psize);
free(pseudogram);
```

Далі сформований пакет надсилається:

```
if (sendto (s, datagram, iph->tot_len , 0,
 (struct sockaddr *) &sin, sizeof (sin)) < 0) {
    perror("sendto failed");
}
```

```
else {
     printf ("Packet sent. \"" );
     for (j = 0; j < 6; ++j)
     if (6*i + j < length)
          printf("%c", payload[6*i + j]);
     puts("\"");
}
```

3.3.2 Клієнтська частина

Клієнтська частина програми повинна реалізовувати механізм прослуховування мережі та ідентифікації пакетів із вбудованими прихованими повідомлення. Подальше вилучення прихованого повідомлення з службових полів заголовків протокольних блоків даних а саме IP-пакету та TCP-сегменту. Збір декапсульованого повідомлення в одне ціле та виведення його на екран.

Для того щоб приймати повідомлення потрібно створити буфер у якому буде зберігатись декапсульоване приховане повідомлення:

```
char global_buffer[1024];
```

Також проводимо ініціалізацію змінних для адрес відправника та отримувача, а також сонету.

```
int sock_raw;
int global_n = 0;
char * src_addr, *dst_addr;
```

Для початку потрібно дізнатись адреси відправника і отримувача:

```
if (argc < 3) {
```

```
        puts("Enter source and destination ip");
        return 1;
    }
    src_addr = argv[1];
    dst_addr = argv[2];
```

Створення змінних потрібних для отримання повідомлення та створення сокету, що буде прослуховувати мережу:

```
    int saddr_size , data_size;
    struct sockaddr saddr;
    unsigned char *buffer = (unsigned char *)malloc(65536);
    sock_raw = socket(AF_INET , SOCK_RAW , IPPROTO_TCP);
    if(sock_raw < 0) {
        printf("Socket Error\n");
        return 1;
    }
```

Отримання пакета:

```
    saddr_size = sizeof saddr;
    data_size = recvfrom(sock_raw , buffer , 65536, 0, &saddr,
     &saddr_size);
    if(data_size <0 ) {
        printf("Recvfrom error , failed to get packets\n");
        return 1;
    }
```

Отримання вказівника на заголовок пакету і перевірка чи є протоколом TCP:

```
struct iphdr *iph = (struct iphdr*)buffer;
if (iph->protocol == IPPROTO_TCP) {
   handleMessage(buffer);
}
```

Якщо це TCP пакет відбувається його обробка. Для початку створюються елементи структури заголовків:

```
struct iphdr *iph = (struct iphdr *)Buffer;
struct tcphdr *tcph = (struct tcphdr *)
 (Buffer + sizeof (struct iphdr));
struct sockaddr_in source,dest;
```

Отримання адрес відправника і отримувача:

```
memset(&source, 0, sizeof(source));
source.sin_addr.s_addr = iph->saddr;
memset(&dest, 0, sizeof(dest));
dest.sin_addr.s_addr = iph->daddr;
```

Якщо адреса відправника співпадає з заданою відбувається декапсуляція першої частини прихованого повідомлення:

```
if (source.sin_addr.s_addr == inet_addr(src_addr)
 && dest.sin_addr.s_addr == inet_addr(dst_addr)) {
   setvbuf (stdout, NULL, _IONBF, 0);
   char payload[6];
   payload[0] = iph->id >> 8;
```

```
payload[1] = iph->id & ((1 << 8) - 1);
```

Декапсуляція другої частини прихованого повідомлення:

```
for (i = 0; i < 4; ++i) {
     payload[i + 2] = (tcph->seq >> i*8) & ((1 << 8 ) -
1);
}
```

Копіюємо отримані частини прихованого повідомлення в створений буфер:

```
for (i = 0; i < 6; ++i) {
     global_buffer[global_n++] = payload[i];
}
```

Отримана частина повідомлення виводиться на екран та перевірка на те чи є ця частина кінцем повідомлення:

```
for (i = 0; i < 6; ++i) {
     if (payload[i])
          printf("%c", payload[i]);
     else {
          puts("");
          global_n = 0;
          break;
     }
}
```

Відключення сокета:

```
close(sock_raw);
```

3.3.3 Принцип роботи

Спочатку запускається клієнтська частина. В неї вводяться адреси відправника і отримувача. Запускається сокет, що прослуховує мережу. Далі запускається серверна частина. В неї вводиться приховане повідомлення для інкапсуляції. Визначається кількість пакетів, що потрібна для передачі прихованого повідомлення. Приховане повідомлення ділиться на частини і надсилається за адресою призначення. Клієнтська програма ідентифікує пакети з прихованим повідомленням і декапсулює його з пакетів. Декапсульована частина повідомлення виводиться на екран і відбувається перевірка чи є це кінцем прихованого повідомлення.

4 ЕКОНОМІЧНА ЧАСТИНА

4.1 Оцінювання комерційного потенціалу розробки

Метою проведення технологічного аудиту є оцінювання комерційного потенціалу розробки, створеної в результаті науково-технічної діяльності [12].

Результатом магістерської кваліфікаційної роботи Стеганографічний метод передачі інформації в заголовках протокольних блоків даних є розробка програмного забезпечення, що дозволяє передавати приховані повідомлення в локальних та глобальних мережах. Для проведення технологічного аудиту залучено 3 незалежних експертів: Колесник Ірина Сергіївна (к.т.н., доцент каф. обчислювальної техніки ВНТУ), Снігур Анатолій Васильович (к.т.н., доцент каф. обчислювальної техніки ВНТУ) та Захарченко Сергій Михайлович (к.т.н., доцент каф. обчислювальної техніки ВНТУ).

Оцінювання комерційного потенціалу буде здійснене за критеріями, що наведені в таблиці 4.1.

Таблиця 4.1 - Критерії оцінювання комерційного потенціалу розробки бальна оцінка

Критерії оцінювання та бали (за 5-ти бальною шкалою)					
Кри-терій	0	1	2	3	4
Технічна здійсненність концепції:					
1	Достовірність концепції не підтверджена	Концепція підтверджена експертними висновками	Концепція підтверджена розрахунками	Концепція перевірена на практиці	Перевірено роботоздатність продукту в реальних умовах
Ринкові переваги (недоліки):					
2	Багато аналогів на малому ринку	Мало аналогів на малому ринку	Кілька аналогів на великому ринку	Один аналог на великому ринку	Продукт не має аналогів на великому ринку
3	Ціна продукту значно вища за ціни аналогів	Ціна продукту дещо вища за ціни аналогів	Ціна продукту приблизно дорівнює цінам аналогів	Ціна продукту дещо нижче за ціни аналогів	Ціна продукту значно нижче за ціни аналогів

Продовження таблиці 4.1

Критерії оцінювання та бали (за 5-ти бальною шкалою)					
Кри-тер.	0	1	2	3	4
4	Технічні та споживчі властивості продукту значно гірші, ніж в аналогів	Технічні та споживчі властивості продукту трохи гірші, ніж в аналогів	Технічні та споживчі властивості продукту на рівні аналогів	Технічні та споживчі властивості продукту трохи кращі, ніж в аналогів	Технічні та споживчі властивості продукту значно кращі, ніж в аналогів
5	Експлуатаційні витрати значно вищі, ніж в аналогів	Експлуатаційні витрати дещо вищі, ніж в аналогів	Експлуатаційні витрати на рівні експлуатаційних витрат аналогів	Експлуатаційні витрати трохи нижчі, ніж в аналогів	Експлуатаційні витрати значно нижчі, ніж в аналогів
Ринкові перспективи					
6	Ринок малий і не має позитивної динаміки	Ринок малий, але має позитивну динаміку	Середній ринок з позитивною динамікою	Великий стабільний ринок	Великий ринок з позитивною динамікою
7	Активна конкуренція великих компаній на ринку	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкурентів немає
Практична здійсненність					
8	Відсутні фахівці як з технічної, так і з комерційної реалізації ідеї	Необхідно наймати фахівців або витрачати значні кошти та час на навчання наявних фахівців	Необхідне незначне навчання фахівців та збільшення їх штату	Необхідне незначне навчання фахівців	Є фахівці з питань як з технічної, так і з комерційної реалізації ідеї
9	Потрібні значні фінансові ресурси, які відсутні. Джерела фінансування ідеї відсутні	Потрібні незначні фінансові ресурси. Джерела фінансування відсутні	Потрібні значні фінансові ресурси. Джерела фінансування є	Потрібні незначні фінансові ресурси. Джерела фінансування є	Не потребує додаткового фінансування
10	Необхідна розробка нових матеріалів	Потрібні матеріали, що використовуються у військово-промисловому комплексі	Потрібні дорогі матеріали	Потрібні досяжні та дешеві матеріали	Всі матеріали для реалізації ідеї відомі та давно використовуються у виробництві

Продовження таблиці 4.1

11	Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років. Термін окупності інвестицій більше 10-ти років	Термін реалізації ідеї від 3-х до 5-ти років. Термін окупності інвестицій більше 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій від 3-х до 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій менше 3-х років
12	Необхідна розробка регламентних документів та отримання великої кількості дозвільних документів на виробництво та реалізацію продукту	Необхідно отримання великої кількості дозвільних документів на виробництво та реалізацію продукту, що вимагає значних коштів та часу	Процедура отримання дозвільних документів для виробництва та реалізації продукту вимагає незначних коштів та часу	Необхідно тільки пові-домлення відповідним органам про виробництво та реалізацію продукту	Відсутні будь-які регламентні обмеження на виробництво та реалізацію продукту

Результати оцінювання комерційного потенціалу експертами розробки зведено в таблицю 4.2.

Таблиця 4.2 - Результати оцінювання комерційного потенціалу розробки

Критерії	Прізвище, ініціали, посада експерта		
	1 – Колесник	2 – Снігур	3 – Захарченко
	Бали, виставлені експертами:		
1	3	3	3
Ринкові переваги (недоліки):			
2	3	2	4
3	2	3	3
4	4	4	3
5	3	4	3
Ринкові перспективи			
6	3	3	2
7	2	4	3
Практична здійсненність			
8	4	3	3
9	3	2	3
10	3	2	2
11	3	3	3
12	3	3	3
Сума балів	$СБ_1$=36	$СБ_2$=36	$СБ_3$=35
Середньоарифметична сума балів $\overline{СБ}$		35,5	

За даними таблиці 4.2 можна зробити висновок, щодо рівня комерційного потенціалу розробки. Зважимо на результат й порівняємо його з рівнями комерційного потенціалу розробки, що представлено в таблиці 4.3.

Таблиця 4.3 – Рівні комерційного потенціалу розробки

Середньоарифметична сума балів $\overline{СБ}$, розрахована на основі висновків експертів	Рівень комерційного потенціалу розробки
0 – 10	Низький
11 – 20	Нижче середнього
21 – 30	Середній
31 – 40	Вище середнього
41 – 48	Високий

Рівень комерційного потенціалу розробки, становить 35,5 балів, що відповідає рівню «вище середнього».

Актуальність розробки полягає в тому що захист інформації один із пріоритетних напрямків розвитку сучасних інформаційних та мережевих технологій. В якості аналога для розробки було обрано стеганографічний додаток Covert_TCP. Основними недоліками аналогу є те, що він дозволяє передавати приховане повідомлення лише на транспортному рівні, що не дає змогу передавати великі об’єми даних.

У таблиці 4.4 наведені основні технічні показники аналога і нового програмного продукту

Таблиця 4.4 - Основні технічні показники аналога і нового програмного продукту

Показники	Аналог	Нова розробка
Функціональність	2	4
Надійність	4	4
Сумісність	2	2
Супровід	4	4
Економія ресурсів і часу	2	4

Основною перевагою даного продукту є збільшення об’єму інформації що можна передати за одну ітерацію в два раза. Цей продукт є модернізацією незалежного прикладного додатку, а саме додатку Covert_TCP.

Наш продукт може бути використаний тільки в доволі вузькій сфері реалізації, а точніше лише для передачі невеликих об’ємів даних. Продукт має задовольнити потреби в приховуванні факту передачі повідомлень невеликого об’єму.

В подальшому дана робробка може бути використана для передачі корпораціями електронних ключів та іших секретних повідомлень невеликого об’єму.

4.2 Прогнозування витрат на виконання наукової роботи та впровадження результатів.

Проведемо прогнозування витрат на виконання науково-дослідної, дослідно-конструкторської та конструкторсько-технологічної роботи для розробки програмного забезпечення, яке складається з таких етапів:

1-й етап: розрахунок витрат, які безпосередньо стосуються виконавців даного розділу роботи;

2-й етап: розрахунок загальних витрат на виконання даної роботи;

3-й етап: прогнозування загальних витрат на виконання та впровадження результатів даної роботи.

Виконаємо розрахунок витрат приймаючи до уваги те, що розробкою займався один розробник програмного забезпечення.

1. Основна заробітна розробника-дослідника $З_о$:

$$З_о = \frac{М}{Т_р} \cdot t \ [грн], \qquad (4.1)$$

де М – місячний посадовий оклад – 10000 грн;

$Т_р$ – число робочих днів в місяці; приблизно $Т_р$ = (22) дні;

t – число робочих днів роботи розробника-дослідника - 66.

$$З = 150000/22 \cdot 66 = 45000 \text{ (грн)}.$$

2. Додаткова заробітна плата $З_д$ розробника розраховується як 10% від основної заробітної плати:

$$З_д = 0,10 \cdot 45000,00 = 4500,00 \text{ (грн)}.$$

3. Нарахування на заробітну плату $Н_{зп}$ розробника становить:

$$H_{зп} = (З_о + З_д) \cdot \frac{\beta}{100} \text{ [грн]}, \quad (4.2)$$

де $З_о$ – основна заробітна плата розробника;

$З_д$ – додаткова заробітна плата розробника;

β – ставка єдиного внеску на загальнообов'язкове державне соціальне страхування – 22%.

$$Н_{зп} = (45000,00 + 4500,00) \cdot 0,22 = 10890,00 \text{ (грн)}.$$

Амортизація обладнання, комп'ютерів та приміщень, які використовувались під час виконання даного етапу роботи. Дані відрахування розраховують по кожному виду обладнання, приміщенням тощо.

У спрощеному вигляді амортизаційні відрахування А в цілому розраховуємо за формулою:

$$А = \frac{Ц \cdot Т}{12 \cdot Т_В} \text{ [грн]}, \quad (4.3)$$

де Ц – загальна балансова вартість обладнання, приміщення тощо, грн;

Т – фактична тривалість використання, міс;

$Т_В$ – термін використання обладнання, приміщень тощо, роки.

Розробка програмного забезпечення проводилася протягом 3 місяців.

Зроблені розрахунки зведено до таблиці 4.5.

Таблиця 4.5 – Амортизаційні відрахування

Найменування	Балансова вартість, грн	Термін використання, роки	Фактична тривалість використання, міс	Величина амортизаційних відрахувань, грн
Офісне приміщення	100000	25	3	100,00
Ноутбук	15000	5	3	750,00
Всього				850,00

Інформацію про матеріали, що використовуються при розробці даного інноваційного продукту внесено до таблиці 4.6.

Таблиця 4.6 – Матеріали, що використовуються при розробці продукту

Найменування матеріалу	Ціна за одиницю, грн.	Витрачено, шт.	Вартість витраченого матеріалу, грн
Папір (пачка)	98,00	1	98,00
Канцтовари	12,00	4	48,00
Всього			146,00

Під час розробки програмного продукту використовувались лише безкоштовні програмні засоби.

Витрати на енергію визначаються на основі витрат на одиницю продукції та тарифів на енергію за допомогою формули 4.2:

$$B_e = B \cdot П \cdot Ф \cdot K_n \, [грн], \qquad (4.4)$$

де В – вартість 1кВт електроенергії;

П – установлена потужність обладнання, кВт;

Ф – фактична кількість годин роботи комп’ютера при створенні програмного продукту, годин;

$К_п$ – коефіцієнт використання потужності .

Отже, витрати на енергію становлять:

$$B_e = 1{,}88 \cdot 0{,}5 \cdot 528 \cdot 0{,}4 = 198{,}52\ (грн).$$

Також потрібно врахувати витрати на доступ до мережі Інтернет, що використовувався під час виконання роботи.

Витрати за доступ до Інтернет можна розрухувати за формулою:

$$B_{дi} = Ц_{дi} \cdot T\ [грн], \qquad (4.5)$$

де $Ц_{ді}$ – це ціна доступу за місяць;

Т – кількість місяців використання доступу до мережі.

Отже, витрати на доступ до мережі Інтернет становлять:

$$В_{ді} = 150 \cdot 3 = 450{,}00\ (грн).$$

Інші витрати $В_{ін}$ охоплюють: витрати на управління організацією, оплату службових відряджень, витрати на утримання, ремонт та експлуатацію основних засобів, витрати на опалення, освітлення, водопостачання, охорону праці тощо. Інші витрати $І_в$ можна прийняти як 100% від суми основної заробітної плати розробника:

$$В_{ін} = 1 \cdot 45000{,}00 = 45000{,}00\ (грн).$$

В результаті сума усіх витрат, що вказані вище дає витрати на виконання даного етапу роботи В:

$$В = З_о + З_д + Н_{зп} + А + В_{мат} + В_е + В_{ді}\ [грн],$$

$$В = 45000{,}00 + 4500{,}00 + 10890 + 850 + 146 + 198{,}25 + 450 + 45000 = 107034{,}50\ (грн).$$

2-й етап. Розрахунок загальних витрат на виконання даної роботи. Загальна вартість всієї наукової роботи визначається за $В_{заг}$ формулою:

$$B_{заг} = \frac{B}{\alpha} \ [грн], \tag{4.6}$$

де α – частка витрат, які безпосередньо здійснює виконавець даного етапу роботи, у відн. одиницях.

Так, як над роботою задіяна одна людина, якою виконується уся робота, то α становить 1. Підставивши дані у формулу, отримуємо:

$$В_{заг} = 107034{,}50 \text{ (грн)}.$$

3-й етап. Прогнозування загальних витрат на виконання та впровадження результатів виконаної роботи. Прогнозування загальних витрат ЗВ на виконання та впровадження результатів виконаної роботи здійснюється за формулою:

$$ЗВ = \frac{B_{заг}}{\beta} \ [грн], \tag{4.7}$$

де β – коефіцієнт, який характеризує етап (стадію) виконання даної роботи. Так, якщо розробка знаходиться:

- на стадії науково-дослідних робіт, то $\beta \approx 0{,}1$;
- на стадії технічного проектування, то $\beta \approx 0{,}2$;
- на стадії розробки конструкторської документації, то $\beta \approx 0{,}3$;
- на стадії розробки технологій, то $\beta \approx 0{,}4$;
- на стадії розробки дослідного зразка, то $\beta \approx 0{,}5$;
- на стадії розробки промислового зразка, $\beta \approx 0{,}7$;
- на стадії впровадження, то $\beta \approx 0{,}9$.

$$ЗВ = 107034{,}50/0{,}7 = 152906{,}42 \text{ (грн)}.$$

4.3 Прогнозування комерційних ефектів від реалізації результатів розробки

Спробуємо кількісно спрогнозувати, яку вигоду, можна отримати у майбутньому від впровадження результатів виконаної наукової роботи. Зрозуміло, що всі зроблені розрахунки будуть приблизними і не передбачають деталізації.

В умовах ринку узагальнюючим позитивним результатом, що його отримує підприємець від впровадження результатів нової розробки, є збільшення чистого прибутку. Зростання чистого прибутку спробуємо оцінити у теперішній вартості грошей.

Зростання чистого прибутку забезпечить надходження додаткових коштів, які дозволять покращити фінансові результати діяльності та виплатити кредити (якщо вони потрібні для впровадження результатів розробки).

Оцінити збільшення чистого прибутку $\Delta П_i$ для кожного із років, протягом яких очікується отримання позитивних результатів від впровадження розробки можливо використовуючи формулу:

$$\Delta П_i = \sum_i^n \left(\Delta П_o \cdot N + Ц_o \cdot \Delta N\right) \cdot \lambda \cdot \rho \cdot \left(1 - \frac{\upsilon}{100}\right) [грн], \qquad (4.8)$$

де $\Delta По$ – покращення основного оціночного показника від впровадження результатів розробки у даному році. Зазвичай таким показником може бути ціна одиниці нової розробки;

N – основний кількісний показник, який визначає діяльність підприємства у даному році до впровадження результатів наукової розробки;

ΔN – покращення основного кількісного показника діяльності підприємства від впровадження результатів розробки;

$Ц_o$ – основний оціночний показник, який визначає діяльність підприємства у даному році після впровадження результатів наукової розробки;

n – кількість років, протягом яких очікується отримання позитивних результатів від впровадження розробки;

λ – коефіцієнт, який враховує сплату податку на додану вартість.

ρ – коефіцієнт, який враховує рентабельність продукту. Рекомендується приймати ρ = 0,2...0,3;

υ – ставка податку на прибуток (18%).

В результаті впровадження результатів наукової розробки покращується якість продукту, що дозволяє підвищити ціну його реалізації на 500 грн. Кількість одиниць реалізованої продукції також збільшиться: протягом першого року – на 1500 шт., протягом другого року – ще на 2000 шт., протягом третього року – ще на 2000 шт.

Орієнтовно: реалізація продукції до впровадження результатів наукової розробки складала 1 шт., а її ціна – 500 грн.

Спрогнозуємо збільшення чистого прибутку підприємства від впровадження результатів наукової розробки у кожному році відносно базового.

Збільшення чистого прибутку підприємства ΔПі протягом першого року складе:

$$\Delta П_1 = [500 \cdot 1 + (500 + 500) \cdot 1500] \cdot 0{,}88 \cdot 0{,}25 \cdot \left(1 - \frac{18}{100}\right) = 270690{,}00\ (грн).$$

Збільшення чистого прибутку підприємства ΔПі протягом другого року (відносно базового року, тобто року до впровадження результатів наукової розробки) складе:

$$\Delta П_2 = [500 \cdot 1 + (500 + 500) \cdot (1500 + 2000)] \cdot 0{,}88 \cdot 0{,}25 \cdot \left(1 - \frac{18}{100}\right) = 631490{,}00\ (грн).$$

Збільшення чистого прибутку підприємства ΔПі протягом третього року (відносно базового року, тобто року до впровадження результатів наукової розробки) складе:

$$\Delta П_3 = [500 \cdot 1 + (500 + 500) \cdot (1500 + 2000 + 2000)] \cdot 0{,}88 \cdot 0{,}25 \cdot \left(1 - \frac{18}{100}\right) = 992290{,}00\ (грн).$$

4.4 Розрахунок ефективності вкладених інвестицій та періоду їх окупності

Основними показниками, які визначають доцільність фінансування наукової розробки інвестором, є абсолютна і відносна ефективність вкладених інвестицій та термін їх окупності.

Розрахунок ефективності вкладених інвестицій передбачає проведення таких робіт:

1-й крок. Розраховується теперішня вартість інвестицій PV, що вкладаються в наукову розробку. Такою вартістю можемо вважати прогнозовану величину загальних витрат ЗВ на виконання та впровадження результатів НДДКР, розраховану за формулою, тобто будемо вважати, що ЗВ = PV=152906,42 грн.

2-й крок. Розраховується очікуване збільшення прибутку $\Delta П_i$, що його отримає підприємство (організація) від впровадження результатів наукової розробки, для кожного із років, починаючи з першого року впровадження. Таке збільшення прибутку також було розраховане нами раніше.

3-й крок. Будуємо вісь часу, на яку наносимо всі платежі (інвестиції та прибутки), що мають місце під час виконання науково-дослідної роботи та впровадження її результатів.

Платежі показуємо у ті терміни, коли вони здійснюються.

Припустимо, що загальні витрати ЗВ на виконання та впровадження результатів НДДКР (або теперішня вартість інвестицій PV) дорівнює 152906,42 грн. Результати вкладених у наукову розробку інвестицій почнуть виявлятись протягом трьох років. У першому році підприємство отримає збільшення чистого прибутку на 270690,00 грн відносно базового року, у другому році – збільшення чистого прибутку на 631490,00 грн (відносно базового року), у третьому році – збільшення чистого прибутку на 992290,00 грн (відносно базового року).

Тоді рух платежів (інвестицій та додаткових прибутків) буде мати вигляд, наведений на рис. 4.1.

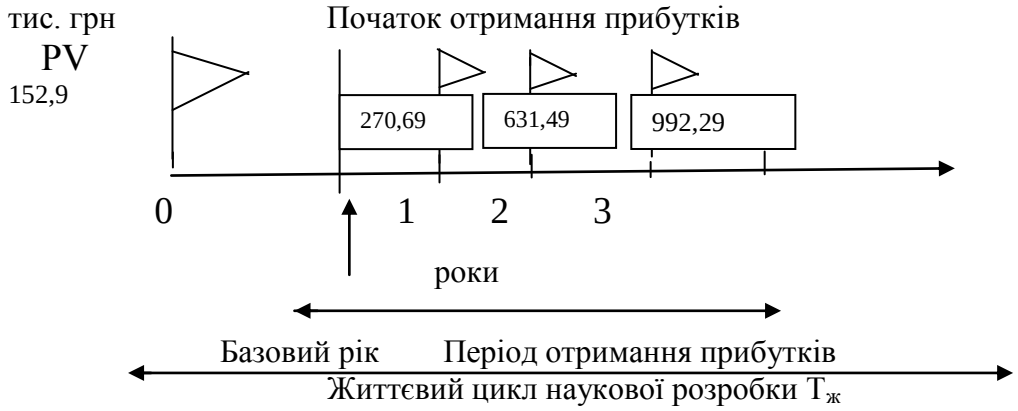


Рисунок 4.1 – Вісь часу з фіксацією платежів, що мають місце під час розробки та впровадження результатів НДДКР

4-й крок. Розраховується абсолютна ефективність вкладених інвестицій $Е_{абс}$. Для цього використаємо формулу:

$$Е_{абс} = (ПП – PV), \qquad (4.9)$$

де ПП – приведена вартість всіх чистих прибутків, що їх отримає підприємство (організація) від реалізації результатів наукової розробки, грн.;

PV – теперішня вартість інвестицій PV = ЗВ, грн.

У свою чергу, приведена вартість всіх чистих прибутків ПП розраховується за формулою:

$$ПП = \sum_{1}^{т} \frac{\Delta П_i}{(1+\tau)^t}, \qquad (4.10)$$

де $\Delta П_i$ – збільшення чистого прибутку у кожному із років, протягом яких виявляються результати виконаної та впровадженої НДДКР, грн;

т – період часу, протягом якого виявляються результати впровадженої НДДКР, роки;

τ – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні; для України цей показник знаходиться на рівні 0,1;

t – період часу (в роках) від моменту отримання чистого прибутку до точки „0”.

Якщо $Е_{абс} > 0$, то результат від проведення наукових досліджень та їх впровадження принесе прибуток, але це також ще не свідчить про те, що інвестор буде зацікавлений у фінансуванні даного проекту (роботи).

Розрахуємо абсолютну ефективність інвестицій, вкладених у реалізацію проекту. Ставка дисконтування τ дорівнює 0,1. Отримаємо:

$$ПП = \frac{270690,00}{(1+0,1)^1} + \frac{631490,00}{(1+0,1)^2} + \frac{992290,00}{(1+0,1)^3} = 1445721,80\ (грн).$$

Тоді,

$$Е_{абс} = 1445721,80 – 152906,42 = 1430515,38\ (грн).$$

Оскільки $Е_{абс} > 0$, то вкладання коштів на виконання та впровадження результатів НДДКР є доцільним.

5-й крок. Розраховуємо відносну (щорічну) ефективність вкладених в наукову розробку інвестицій $Е_в$. Для цього використовуємо формулу:

$$Е_в = \sqrt[Т_ж]{1 + \frac{Е_{абс}}{PV}} - 1, \qquad (4.11)$$

де $Е_{абс}$ – абсолютна ефективність вкладених інвестицій, грн;

PV – теперішня вартість інвестицій PV = ЗВ, грн;

$Т_ж$ – життєвий цикл наукової розробки, роки.

Далі, розрахована величина $Е_в$ порівнюється з мінімальною (бар'єрною) ставкою дисконтування τмін, яка визначає ту мінімальну дохідність, нижче за яку інвестиції вкладатися не будуть.

У загальному вигляді мінімальна (бар'єрна) ставка дисконтування τмін визначається за формулою:

$$\tau = d + f, \quad (4.12)$$

де d – середньозважена ставка за депозитними операціями в комерційних банках; d = 0,2;

f – показник, що характеризує ризикованість вкладень; зазвичай, величина f = 0,05.

Якщо величина $E_в > \tau$ мін, то інвестор може бути зацікавлений у фінансуванні даної наукової розробки. В іншому випадку фінансування наукової розробки здійснюватися не буде. Спочатку спрогнозуємо величину τ мін. Припустимо, що за даних умов τ мін = 0,2 + 0,05 = 0,25. Тоді відносна (щорічна) ефективність вкладних інвестицій в проведення наукових досліджень та впровадження їх результатів складе:

$$E_в = \sqrt[3]{1 + \frac{1430515,38}{152906,42}} - 1 = \sqrt[3]{10,35} - 1 = 1,17 \text{ або } 117\%$$

Оскільки $E_в$ = 117 % > τ мін = 0,25 = 25%, то у інвестора буде зацікавленість вкладати гроші в дану наукову розробку.

6-й крок. Розраховуємо термін окупності вкладених у реалізацію наукового проекту інвестицій. Термін окупності вкладених у реалізацію наукового проекту інвестицій $T_{ок}$ можна розрахувати за формулою:

$$T_{ок} = \frac{1}{E_в} \ [грн]. \quad (4.13)$$

Для розробки термін окупності вкладених у реалізацію проекту інвестицій $T_{ок}$ складе:

$$T_{ок} = \frac{1}{1,17} = 0,85 \text{ (року)},$$

що свідчить про доцільність фінансування даної наукової розробки.

4.5 Висновки

В даному розділі було виконано оцінювання комерційного потенціалу розробки. Проведено технологічний аудит з залученням трьох незалежних експертів. Визначено, що рівень комерційного потенціалу розробки вище середнього.

Аналіз комерційного потенціалу розробки показав, що програмний продукт за своїми характеристиками випереджає аналогічні програмні продукти і є перспективною розробкою. Він має кращі функціональні показники, а тому є конкурентоспроможним товаром на ринку. Існуючі переваги нової розробки дозволять швидко її поширити та популяризувати.

Згідно із розрахунками всіх статей витрат на виконання науково-дослідної, дослідно-конструкторської та конструкторсько-технологічної роботи загальні витрати на розробку складають 152906,42 грн.

Розрахована абсолютна ефективність вкладених інвестицій в сумі 1430515,38 грн свідчить про отримання прибутку інвестором від комерціалізації програмного продукту.

Щорічна ефективність вкладених в наукову розробку інвестицій складає 117 %, що вище за мінімальну бар'єрну ставку дисконтування, яка складає 25%. Це означає потенційну зацікавленість інвесторів у фінансуванні розробки.

Термін окупності вкладених у реалізацію проекту інвестицій становить 0,85 року, що також свідчить про доцільність фінансування нової розробки.

ВИСНОВКИ

Проаналізовано сучасний стан технологій захисту інформації, зокрема, методи приховування факту передачі інформації – стеганографію, в результаті чого було визначено, що розвиток у напряму стеганографії в протокольних блоках даних є доцільним.

Розглянуто рівні моделі OSI та стеку протоколів TCP/IP, а саме протокольні блоки даних канального, мережевого та транспортного рівнів. Проаналізовано службові поля вище згаданих протокольних блоків даних на можливість використання їх в якості контейнерів для передачі прихованого повідомлення.

Визначено всі можливі поля для передачі прихованого повідомлення, та на основі отриманих даних були виділені основні методи передачі прихованої інформації в заголовках протокольних блоків даних. Також було визначено швидкість передачі прихованого повідомлення відносно загального об’єму та коефіцієнт корисного навантаження від загального об’єму блоку даних.

На основі отриманого результату було визначено найдоцільніший метод передачі повідомлення.

Обраний метод був реалізований у вигляді двох програм, причому одна реалізовує серверну частину, що інкапсулює приховане повідомлення в заголовок та передає пакет, та клієнтську частину, що прослуховує мережу на наявність пакетів з прихованим повідомлення і декапсулює його у випадку знаходження.

На основі отриманих результатів, можна зробити висновок, що подальший розвиток даного напряму є перспективним, оскільки описані методи стеганографії в об’єднані з іншими методами захисту інформації, достатньо сильно підвищують ефективність захисту.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Технологии и средства связи [Електронний ресурс]. – Режим доступу: http://tssonline.ru/articles2/fix-corp/rost-obema-informatsii--realii-tsifrovoy-vselennoy - Рост объема информации – реалии цифровой вселенной - Назва з екрану.

2. Науково-технічна конференція ВНТУ [Електронний ресурс]. – Режим доступу: https://conferences.vntu.edu.ua/index.php/all-fitki/all-fitki-2019/schedConf/presentations - Науково-технічна конференція факультету інформаційних технологій і комп'ютерної інженерії - Назва з екрану.

3. Бауэр Ф. Расшифрованные секреты. Методы и принципы криптологии / Бауэр Ф. –М.: Мир,2007.– 550 с.

4. Bauman National Library [Електронний ресурс]. – Режим доступу: https://ru.bmstu.wiki/Стеганография - Стеганография - Назва з екрану.

5. Олифер В.Г. Компьютерные сети. Принципы, технологии, протоколы: [учебник для ВУЗов]/ В.Г. Олифер, Н.А. Олифер. – С-Пт.: Питер, 2010. – 944 с. – ISBN 978-5-49807-389-7.

6. Куроуз Дж. Компьютерные сети. Многоуровневая архитектура Интернета/ Дж. Куроуз, К.Росс. –С-Пт.: Питер,2004.– 765 с. – ISBN 5-8046-0093-1.

7. Таненбаум Э. Компьютерные сети/ Э. Таненбаум – С-Пт.: Питер, 2003. – 992 с. – ISBN 5-318-00492-X

8. FIBERBIT [Електронний ресурс]. – Режим доступу: http://fiberbit.com.tw/tcp-transmission-control-protocol-segments-and-fields/ - TCP (Transmission Control Protocol) Segments and Fields - Назва з екрану.

9. Lifewire [Електронний ресурс]. – Режим доступу: https://www.lifewire.com/structure-of-ip-packet-3426715 - The Structure of an IP Packet - Назва з екрану.

10. Study CCNA [Електронний ресурс]. – Режим доступу: https://study-ccna.com/ethernet-frame/ - Ethernet frame - Назва з екрану.

11. Навчальні матеріали онлайн [Електронний ресурс]. – Режим доступу: http://pidruchniki.com/18421120/ekonomika/klasifikatsiya_kompyuternih_merezh - Класифікація комп’ютерних мереж - Назва з екрану.

12. Методичні вказівки до виконання студентами-магістрантами економічної частини магістерських кваліфікаційних робіт / Уклад. В. О. Козловський – Вінниця: ВНТУ, 2012. – 22 с.

Додаток А

Міністерство освіти і науки України

Вінницький національний технічний університет

Факультет інформаційних технологій та комп’ютерної інженерії

Кафедра обчислювальної техніки

ЗАТВЕРДЖУЮ

Завідувач кафедри ОТ

_____

«____» _____ 20___ року

ТЕХНІЧНЕ ЗАВДАННЯ

на виконання магістерської дипломної роботи

«СТЕГАНОГРАФІЧНИЙ МЕТОД ПЕРЕДАЧІ ІНФОРМАЦІЇ В ЗАГОЛОВКАХ ПРОТОКОЛЬНИХ БЛОКІВ ДАНИХ

»

08-023.МКР.011.00.000.ТЗ

Виконав: студент 2 курсу, групи 1КІ-18м

зі спеціальності:

123 «Комп’ютерна інженерія»
(шифр і назва напряму підготовки)

Моторнюк Д.А.
(прізвище та ініціали)

Керівник Захарченко С.М.
(прізвище та ініціали)

м. Вінниця – 2019 р.

1. Підстава для виконання дипломної роботи (ДР)

1.1 Предметом роботи даної програми є метод стеганографії, а саме вбудовування прихованого повідомлення в поля заголовків протокольних блоків даних, для передачі і його подальшого вилучення.

1.2 Наказ про затвердження теми дипломної роботи.

2. Мета і призначення ДР

Підсумком виконання даної дипломної роботи стала розробка методу вбудовування прихованого повідомлення в поля заголовків протокольних блоків даних мережевого та транспортного рівнів. Даний метод дозволяє збільшити кількість інформації, що передається, вдвічі, зберігаючи при цьому рівень секретності.

Основними перевагами є:

– Основні переваги – це безкоштовність і можливість передавати приховані повідомлення в локальних та глобальних мережах.

– Використання полів що формуються псевдовипадково і тому не впливають на функціонування блоку даних.

Основними недоліками є:

– Єдиним недоліком є те, що через відсутність шифрування при перехоплені дані можуть бути вкрадені.

3. Вихідні дані для виконання ДР

Список технічної літератури, аналіз, вивчення та дослідження процесів захисту інформації та стеганографії, технічне завдання на магістерську роботу.

4. Матеріали, що подаються до захисту ДР

Пояснювальна записка ДР, графічні і ілюстративні матеріали, протокол попереднього захисту ДР на кафедрі, відгук наукового керівника, відзив

рецензента, протоколи складання державних екзаменів, анотації до ДР українською та іноземною мовами.

5. Техніко-економічні показники

5.1 Витрати на програмні засоби, що використовуються в ході даної розробки, повинні бути мінімальними.

5.2 Лімітна ціна на програмне забезпечення не повинна перевищувати ціну аналога.

6. Порядок контролю виконання та захисту ДР

6.1.Робота виконується в три етапи, таблиця 6.1.

Таблиця 6.1 – Етапи виконання роботи.

Етап	Зміст	Початок	Кінець	Результат
1	Інформаційний пошук та огляд літературних джерел. Розробка методу та програмної реалізації стеганографічного методу			Розділи 1 та 3.
2	Техніко-економічне обґрунтування теми дипломної роботи, розробка програмного засобу.			Чернетки матеріалів 1 розділу. Попередній захист.
3	Підготовка матеріалів пояснювальної записки.			Пояснювальна записка.

7. Загальний алгоритм роботи програми є таким:

1) Формуються псевдозаголовки мережевого та транспортного рівнів
2) Відбувається отримання повідомлення
3) Визначення кількості ітерацій для пересилки повідомлення
4) Вбудовування частини повідомлення в заголовок
5) Відправка блоку даних
6) Якщо повідомлення передане повністю пункт закінчити роботу, якщо ні перехід до пункту 4

8. Порядок контролю та прийому

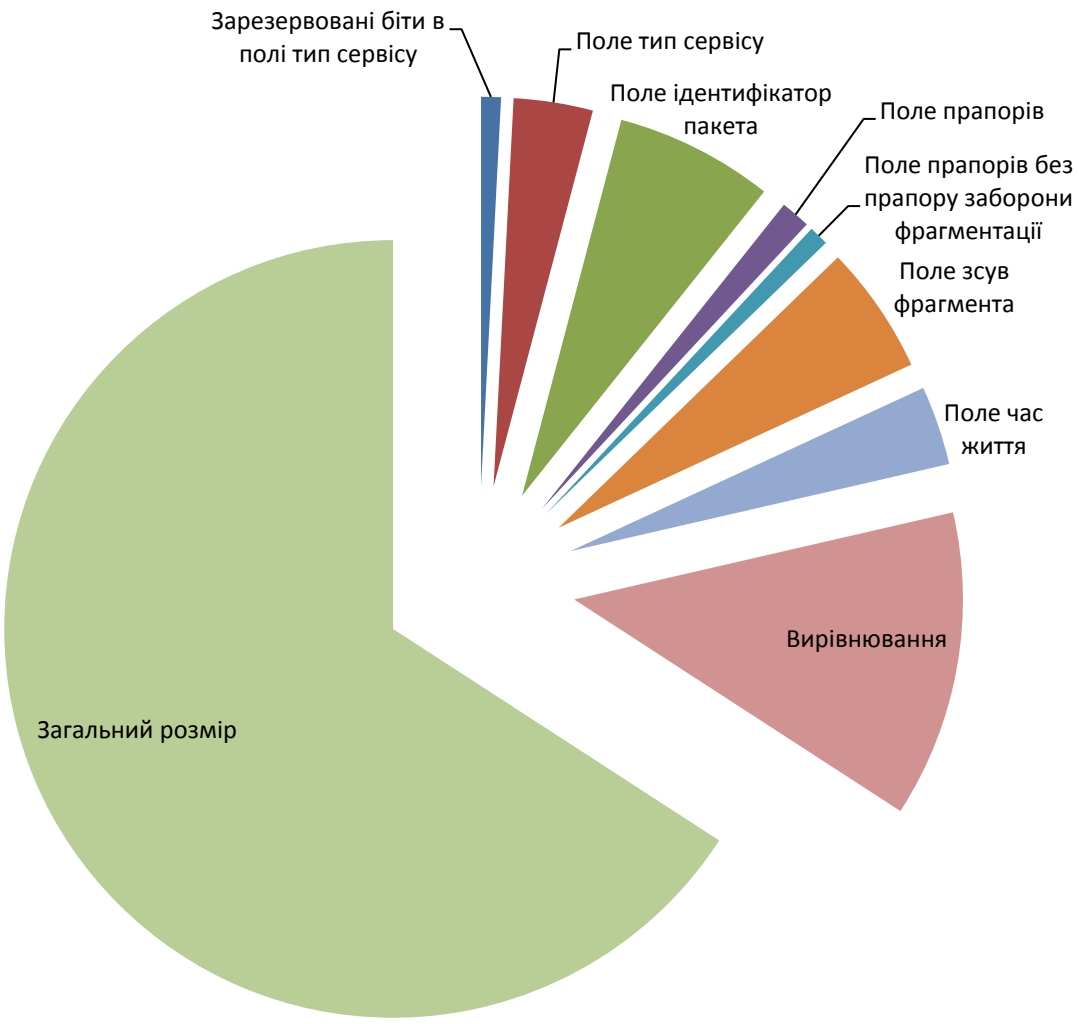
8.1 До приймання дипломної роботи надається:

- пояснювальна записка з відповідними узгодженнями;
- презентація;
- відгук керівника роботи та рецензента;

Технічне завдання до виконання прийняв _____ Моторнюк Д.А.

Додаток Б

Порівняння розміру контейнерів із загальним розміром заголовку



Додаток В

Порівняння методів передачі прихованого повідомлення

Метод передачі	k	$N_{\text{п}}$
Заголовок канального рівня	0,09735	171
Заголовок мережевого рівня. Швидкий методу із забороною фрагментації	0,5	103
Заголовок мережевого рівня. Швидкий методу без заборони фрагментації	0,425	121
Заголовок мережевого рівня. Метод із затримками та з забороною фрагментації	0,5375	96
Заголовок мережевого рівня. Метод із затримками відправки пакетів та без заборони фрагментації	0,4625	111
Заголовок транспортного рівня	0,125	128

Додаток Г

Лістинг серверної частини

```
#include<stdio.h>
#include<string.h>
#include<sys/socket.h>
#include<stdlib.h>
#include<errno.h>
#include<netinet/tcp.h>
#include<netinet/ip.h>
#include <unistd.h>
struct pseudo_header {
     u_int32_t source_address;
     u_int32_t dest_address;
     u_int8_t placeholder;
     u_int8_t protocol;
     u_int16_t tcp_length;
};
unsigned short csum(unsigned short *ptr,int nbytes) {
     register long sum;
     unsigned short oddbyte;
     register short answer;
     sum=0;
     while(nbytes>1) {
          sum+=*ptr++; nbytes-=2;
     }
     if(nbytes==1) {
          oddbyte=0;
          *((u_char*)&oddbyte)=*(u_char*)ptr;
          sum+=oddbyte;
     }
```

```
    sum = (sum>>16)+(sum & 0xffff);
    sum = sum + (sum>>16);
    answer=(short)~sum;
    return(answer);
}
int main (int argc, char* argv[]) {
srand(time(NULL));
if (argc < 3) {
    puts("Введіть адреси відправника та призначення");
    return 1;
}
while (1) {
    puts("Enter payload:");
    char payload[1024];
    fgets(payload, 1024, stdin);
    int length = strlen(payload);
    if (length > 0 && payload[strlen (payload) - 1] == '\n')
    payload[strlen
      (payload) - 1] = '\0';
    if (!length) break;
    int n = (length + 5)/6;
    int i;
    for (i = 0; i < n; ++i) {
        usleep(10000);
        int s = socket (PF_INET, SOCK_RAW, IPPROTO_TCP);
        if(s == -1) {
            perror("Failed to create socket");
            exit(1);
        }
        char datagram[4096] , source_ip[32] , *pseudogram;
```

```
memset (datagram, 0, 4096);
struct iphdr *iph = (struct iphdr *) datagram;
struct tcphdr *tcph = (struct tcphdr *) (datagram +
sizeof (struct iphdr));
struct sockaddr_in sin;
struct pseudo_header psh;
strcpy(source_ip , argv[1]);
sin.sin_family = AF_INET;
sin.sin_port = htons(80);
sin.sin_addr.s_addr = inet_addr (argv[2]);
iph->ihl = 5;
iph->version = 4;
iph->tos = 0;
iph->tot_len = sizeof (struct iphdr) + sizeof
(struct tcphdr);
iph->id = (6*i < length ? payload[6*i] << 8 : 0) +
(6*i + 1 < length ? payload[6*i + 1] : 0);
if (iph->id == 0) iph->id = 1;
iph->frag_off = 0;
iph->ttl = 64;
iph->protocol = IPPROTO_TCP;
iph->check = 0;
iph->saddr = inet_addr ( source_ip );
iph->daddr = sin.sin_addr.s_addr;
iph->check = csum ((unsigned short *) datagram, iph-
>tot_len);
tcph->source = htons (20);
tcph->dest = htons (rand() % 10000);
tcph->ack_seq = 0;
tcph->seq = 0;
```

```
int j;
for (j = 0; j < 4; ++j)
     tcph->seq += (6*i + 2 + j < length ?
payload[6*i + 2 + j] : 0)
       << 8*j;
tcph->doff = 5;
tcph->fin=0;
tcph->syn=1;
tcph->rst=0;
tcph->psh=0;
tcph->ack=0;
tcph->urg=0;
tcph->window = htons (5840);
tcph->check = 0;
tcph->urg_ptr = 0;
psh.source_address = inet_addr( source_ip );
psh.dest_address = sin.sin_addr.s_addr;
psh.placeholder = 0;
psh.protocol = IPPROTO_TCP;
psh.tcp_length = 0;
int psize = sizeof(struct pseudo_header) +
sizeof(struct tcphdr);
pseudogram = (char*)malloc(psize);
memcpy(pseudogram , (char*) &psh , sizeof (struct
pseudo_header));
memcpy(pseudogram + sizeof(struct pseudo_header) ,
tcph, sizeof(struct tcphdr));
tcph->check = csum( (unsigned short*) pseudogram ,
psize);
free(pseudogram);
```

```
        int one = 1;
        const int *val = &one;
        if (setsockopt (s, IPPROTO_IP, IP_HDRINCL, val,
        sizeof (one)) < 0) {
            perror("Error setting IP_HDRINCL");
            exit(0);
        }
        if (sendto (s, datagram, iph->tot_len , 0, (struct
        sockaddr *) &sin, sizeof (sin)) < 0) {
            perror("sendto failed");
        }
        else {
        printf ("Packet sent. \"" );
        for (j = 0; j < 6; ++j)
            if (6*i + j < length)
                printf("%c", payload[6*i + j]); puts("\"");
        }
    }
}
return 0;
}
```

Додаток Д

Лістинг клієнтської частини

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#include<netinet/ip_icmp.h>
#include<netinet/udp.h>
#include<netinet/tcp.h>
#include<netinet/ip.h>
#include<sys/socket.h>
#include<arpa/inet.h>
void processPacket(unsigned char*);
void handleMessage(unsigned char*);
int sock_raw;
char global_buffer[1024];
int global_n = 0;
char * src_addr, *dst_addr;
int main(int argc, char* argv[]) {
    if (argc < 3) {
        puts("Enter source and destination ip");
        return 1;
    }
    src_addr = argv[1];
    dst_addr = argv[2];
    int saddr_size , data_size;
    struct sockaddr saddr;
    unsigned char *buffer = (unsigned char *)malloc(65536);
    puts("Starting...");
    sock_raw = socket(AF_INET , SOCK_RAW , IPPROTO_TCP);
    if(sock_raw < 0) {
```

```
            printf("Socket Error\n");
            return 1;
        }
        while(1) {
            saddr_size = sizeof saddr;
            data_size = recvfrom(sock_raw , buffer , 65536 , 0 ,
        &saddr ,
             &saddr_size);
            if(data_size <0 ) {
                printf("Recvfrom error , failed to get
            packets\n");
                return 1;
            }
            processPacket(buffer);
    }
    close(sock_raw);
    printf("Finished");
    return 0;
    }
    void processPacket(unsigned char* buffer) {
        struct iphdr *iph = (struct iphdr*)buffer;
        if (iph->protocol == IPPROTO_TCP) {
            handleMessage(buffer);
        }
    }
    void handleMessage(unsigned char *Buffer) {
        int i;
        struct iphdr *iph = (struct iphdr *)Buffer;
        struct tcphdr *tcph = (struct tcphdr *) (Buffer + sizeof
    (struct
```

```
        iphdr));
        struct sockaddr_in source,dest;
        memset(&source, 0, sizeof(source));
        source.sin_addr.s_addr = iph->saddr;
        memset(&dest, 0, sizeof(dest));
        dest.sin_addr.s_addr = iph->daddr;
        if (source.sin_addr.s_addr == inet_addr(src_addr) &&
        dest.sin_addr.s_addr == inet_addr(dst_addr)) {
             setvbuf (stdout, NULL, _IONBF, 0);
             char payload[6];
             payload[0] = iph->id >> 8;
             payload[1] = iph->id & ((1 << 8) - 1);
             for (i = 0; i < 4; ++i) {
                  payload[i + 2] = (tcph->seq >> i*8) & ((1 << 8
             ) - 1);
             }
             for (i = 0; i < 6; ++i) {
                  global_buffer[global_n++] = payload[i];
             }
             for (i = 0; i < 6; ++i) {
                  if (payload[i])
                       printf("%c", payload[i]);
                  else {
                       puts("");
                       global_n = 0;
                       break;
                  }
             }
        }
}
```