НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

Навчально-науковий інститут телекомунікаційних систем

Кафедра телекомунікацій

«На правах рукопису»
УДК _____

«До захисту допущено»
Завідувач кафедри
_____ Сергій КРАВЧУК
«___»_____2022 р.

Магістерська дисертація

на здобуття ступеня магістра
за освітньо-професійною програмою «Інженерія та програмування інфокомунікацій»
зі спеціальності 172 «Телекомунікації та радіотехніка»

на тему: «Розвиток методів захисту телекомунікаційних та інформаційних систем»

Виконала:
студентка ІІ курсу, групи ТЗ-11мп
Нсер Анжела Махер _____

Керівник:
доцент кафедри ТК НН ІТС, к.т.н., с.н.с.
Міночкін Дмитро Анатолійович _____

Рецензент:
Доцент кафедри ІКТС, к.т.н., доцент
Кононова Ірина Віталіївна _____

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань.
Студент _____

Київ – 2022 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

Навчально-науковий інститут телекомунікаційних систем
Кафедра телекомунікацій

Рівень вищої освіти – другий (магістерський)

Спеціальність – 172 «Телекомунікації та радіотехніка»

Освітньо-професійна програма «Інженерія та програмування інфокомунікацій»

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ Сергій КРАВЧУК
«___»_____2022 р.

ЗАВДАННЯ
на магістерську дисертацію студентці

Нсер Анжелі Махер

1. Тема дисертації «Розвиток методів захисту телекомунікаційних та інформаційних систем», науковий керівник дисертації Міночкін Дмитро Анатолійович, к.т.н., с.н.с., затверджені наказом по університету від «28» жовтня 2022 р. № 3995-с.

2. Термін подання студентом дисертації 10.12.2022 р.

3. Об’єкт дослідження: комплексна система шифрування, реалізована за допомогою AES.

4. Предмет дослідження: є метод підвищення безпеки передачі даних за допомогою системи шифрування реалізованої за допомогою AES.

5. Перелік завдань, які потрібно розробити: схеми шифрування, що поєднує алгоритми AES і RSA; реалізацію алгоритму AES, комплексне застосування та порівняння алгоритму з іншими існуючими методами

6. Орієнтовний перелік ілюстративного матеріалу

- Слайд №1,2 – Тема, мета, актуальність роботи;
- Слайд №3-4 – Реалізація модифікованого алгоритму;
- Слайд №5-6 – Імітаційний тест та його результати;
- Слайд №7-8 – Порівняння з іншими методами та аналіз алгоритму;
- Слайд №9-10 – Висновок та публікації.

7. Орієнтовний перелік публікацій

- Нсер А.М. консульт. Рибак О.О. Аналіз протоколу Data Distribution Service (DDS) // Дванадцята міжнародна науково-технічна конференція студентів та аспірантів «ПЕРСПЕКТИВИ РОЗВИТКУ ІНФОРМАЦІЙНО-ТЕЛЕКОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ ТА СИСТЕМ» XII Міжнародна науково-технічна конференція студентів та аспірантів «Перспективи розвитку інформаційно-телекомунікаційних технологій та систем» ПРІТС 2020: Збірник тез конференції. К.: КПІ ім. Ігоря Сікорського, 2020 с. 359
- Нсер А.М., Міночкін Д.А. Огляд супутникового інтернету Starlink // XV Міжнародна науково-технічна конференція "Перспективи телекомунікацій" ПТ-2021: Збірник матеріалів конференції. К.: КПІ ім. Ігоря Сікорського, 2021 с. 347-349
- Нсер А.М., Міночкін Д.А. OPEN SOURCE INTELLIGENCE (OSINT) XIV Міжнародна науково-технічна конференція студентів та аспірантів «Перспективи розвитку інформаційно-телекомунікаційних технологій та систем» ПРІТС 2022: Збірник тез конференції. К.: КПІ ім. Ігоря Сікорського, 2022 с. 224-225
- Подана до опублікування: Нсер А.М., Міночкін Д.А. Програмні методи моніторингу мережевої безпеки // Збірник наукових праць Військового інституту Київського національного університету імені Тараса Шевченка. К.: ВІКНУ, 2022. № 77.

8. Дата видачі завдання “ 23 ” вересня 2021 р.

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1	Аналіз літератури за темою наукового пошуку	23.09.2021-15.10.2021	виконано
2	Підготовка першого розділу магістерської дисертації	16.10.2021-15.11.2021	виконано
3	Підготовка тез доповідей на наукову конференцію «ПТ-2022»	16.11.2021-31.12.2021	виконано
4	Дослідження методів забезпечення захисту технологій за допомогою методів шифрування	10.01.2022-28.02.2022	виконано
5	Підготовка тез доповідей на наукову конференцію «ПТ-2022»	01.03.2022-01.04.2022	виконано
6	Дослідження основних методів шифрування	02.04.2022-12.04.2022	виконано
7	Підготовка другого розділу магістерської дисертації	01.09.2022-20.09.2022	виконано
8	Підготовка до публікації наукової статті	21.09.2022-03.10.2022	виконано
9	Підготовка третього розділу магістерської дисертації	04.10.2022-18.10.2022	виконано
10	Підготовка четвертого розділу магістерської дисертації	19.10.2022-08.11.2022	виконано
11	Підготовка стартап проекту за магістерською дисертацією	09.11.2022-24.11.2022	виконано
12	Висновки по всім сферам дослідження, які були проведені.	25.11.2022-26.11.2022	виконано
13	Оформлення магістерської дисертації	27.11.2022-05.12.2022	виконано

Студент Анжела НСЕР

Науковий керівник дисертації Дмитро МІНОЧКІН

РЕФЕРАТ

Магістерська дисертація містить 95 сторінок, 12 рисунків та 11 таблиць. В роботі було використано 58 джерел.

Метою роботи є покращення методів безпеки передачі даних та їх стійкості до атак.

Об'єктом дослідження є комплексна система шифрування, реалізована за допомогою AES.

Предметом досліджень є метод підвищення безпеки передачі даних за допомогою системи шифрування реалізованої за допомогою AES.

Дана магістерська дисертація зосереджена на систематичному аналізі цих питань і підсумовує реалізацію алгоритму AES, комплексне застосування та порівняння алгоритму з іншими існуючими методами. Щоб проаналізувати продуктивність запропонованого алгоритму та повною мірою використати переваги алгоритму шифрування AES, необхідно зменшити круглий ключ та покращити розклад ключів, а також органічно інтегруватися з алгоритмом RSA. Для реалізації алгоритму використовується мова Java через її велику бібліотеку, потім, щоб показати ефективність запропонованого методу, ми порівнюємо різні параметри, такі як швидкість шифрування/дешифрування, ентропії та споживання пам’яті з класичним алгоритмом. Виходячи з результатів порівняння між AES і гібридним алгоритмом AES, запропонований алгоритм показує хорошу продуктивність і високий рівень безпеки. Тому його можна використовувати для керування ключами та функцій безпеки, зокрема для обміну конфіденційними файлами через незахищений канал..

Ключові слова: Шифрування, AES, RSA, алгоритм, безпека.

ABSTRACT

The master's thesis contains 95 pages, 12 figures and 11 tables. 58 sources were used in the work.

The purpose of the work is to improve data transmission security methods and their resistance to attacks.

The object of the study is a complex encryption system implemented using AES.

The subject of research is a method of increasing the security of data transmission using an encryption system implemented using AES.

This master's thesis focuses on a systematic analysis of these issues and summarizes the implementation of the AES algorithm, the complex application and comparison of the algorithm with other existing methods. In order to analyse the performance of the proposed algorithm and take full advantage of the AES encryption algorithm, it is necessary to reduce the circular key and improve the key distribution, and organically integrate with the RSA algorithm. Java language through its large library is used to implement the algorithm, then to show the effectiveness of the proposed method, we compare various parameters such as encryption/decryption speed, entropy and memory consumption with the classical algorithm. Based on the comparison results between AES and hybrid AES algorithm, the proposed algorithm shows good performance and high level of security. Therefore, it can be used for key management and security functions, including sharing sensitive files over an unsecured channel.

Keywords: Encryption, AES, RSA, algorithm, security.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ

AES	Advanced Encryption Standard
RSA	Rivest, Shamir и Adleman
DES	Data Encryption Standard
3DES	Triple Data Encryption Algorithm
SaaS	Software as a service
LAN	Локальні мережі (Local Area Network)
MAN	Столичні мережі (Metropolitan Area Network)
PAN	Персональна мережа (Personal Area Network)
WAN	Глобальна мережа (Wide Area Network)
HAN	Домашня мережа (Home Area Network)
CAN	Кампусна мережа (Campus Area Network)
RAN	Мережа радіодоступу (Radio Access Network)
VPN	Віртуальна приватна мережа (Virtual Private Network)
GAN	Глобальна мережа (Global Area Network)
OSI	Open Systems Interconnection
TCP/IP	Transmission Control Protocol/Internet Protocol
HTTPS	Secure HyperText Transmission Protocol
SMTP	Simple Mail Transfer Protocol
DNS	Domain Name System
UDP	User Datagram Protocol
FTP	Протокол передачі файлів (File Transfer Protocol)
DHCP	Протокол динамічної конфігурації хоста (Dynamic Host Configuration Protocol)
HDLC	High-Level Data Link Control
UDP	User Datagram Protocol

ВСТУП

Із швидким розвитком Інтернет-технологій і зростанням популярності електронної комерції технологія шифрування даних відіграє дуже важливу роль у безпеці даних. Інформаційна безпека має два аспекти: протокол безпеки та криптографічний алгоритм, і останній є основою технологією інформаційної безпеки. Алгоритм шифрування Advanced Encryption Standard (AES) є одним із найбільш часто використовуваних алгоритмів у алгоритмах симетричного шифрування. Такі алгоритми стикаються з проблемами при використанні в контексті функцій керування ключами та безпеки.

Дана магістерська дисертація зосереджена на систематичному аналізі цих питань і підсумовує реалізацію алгоритму AES, комплексне застосування та порівняння алгоритму з іншими існуючими методами. Щоб проаналізувати продуктивність запропонованого алгоритму та повною мірою використати переваги алгоритму шифрування AES, необхідно зменшити круглий ключ та покращити розклад ключів, а також органічно інтегруватися з алгоритмом RSA. Для реалізації алгоритму використовується мова Java через її велику бібліотеку, потім, щоб показати ефективність запропонованого методу, ми порівнюємо різні параметри, такі як швидкість шифрування/дешифрування, ентропії та споживання пам’яті з класичним алгоритмом. Виходячи з результатів порівняння між AES і гібридним алгоритмом AES, запропонований алгоритм показує хорошу продуктивність і високий рівень безпеки. Тому його можна використовувати для керування ключами та функцій безпеки, зокрема для обміну конфіденційними файлами через незахищений канал. Цей аналіз надає корисну довідкову інформацію для вибору різних алгоритмів шифрування відповідно до різних потреб бізнесу.

Метою роботи є покращення методів безпеки передачі даних та їх стійкості до атак.

Об'єктом дослідження є комплексна система шифрування, реалізована за допомогою AES.

Предметом досліджень є метод підвищення безпеки передачі даних за допомогою системи шифрування реалізованої за допомогою AES.

Новизна: запропонований метод підвищення безпеки технологій за допомогою схеми шифрування, що поєднує алгоритми AES і RSA, і використовує два фіксовані алгоритми шифрування для перевірки алгоритму шифрування. Це дозволить зменшити втрати інформації та кількість несанкціонованих доступів до елементів мережі, а отже це зменшення можливих збитків та вірогідності втрати контролю над пристроями та елементами мережі.

Структура роботи. Робота складається з реферату, змісту, переліку скорочень, вступу, п’яти розділів, висновків до кожного розділу, загальних висновків по роботі, списку використаних джерел та додатків.

РОЗДІЛ 1
ОСНОВНІ ПОНЯТТЯ ПРО ІНФОРМАЦІЙНІ СИСТЕМИ ТА МЕРЕЖІ

Інформаційна система — це інтегрований набір компонентів для збору, зберігання й обробки даних, а також для надання інформації, знань і цифрових продуктів [1]. Підприємства та інші організації розраховують на інформаційні системи, щоб здійснювати та керувати своїми операціями, взаємодіяти зі своїми клієнтами та постачальниками. Інформаційні системи використовуються для управління міжорганізаційними ланцюгами поставок та електронними ринками.

Оскільки інформаційні системи сприяли більш різноманітній людській діяльності, вони справляли глибокий вплив на суспільство. Ці системи пришвидшили темп повсякденної діяльності, дозволили людям розвивати та підтримувати нові та часто більш вигідні стосунки, вплинули на структуру та склад організацій, змінили тип продуктів, що купуються, і вплинули на характер роботи. Інформація та знання стали життєво важливими економічними ресурсами. Проте, разом із новими можливостями, залежність від інформаційних систем принесла нові загрози. Інтенсивні галузеві інновації та наукові дослідження постійно створюють нові можливості, одночасно прагнучи стримувати загрози.

1.1 Компоненти інформаційних систем

Основними компонентами інформаційних систем є комп’ютерне обладнання та програмне забезпечення, комунікації, бази даних і сховища даних, людські ресурси та процедури [2]. Апаратне забезпечення, програмне забезпечення та комунікації складають інформаційну технологію (IT), яка зараз укорінена в діяльності та управлінні організаціями.

1.1.1 Комп’ютерне обладнання

Термін апаратне забезпечення стосується машин і обладнання. У сучасній інформаційній системі до цієї категорії відноситься сам комп’ютер і все його допоміжне обладнання. Допоміжне обладнання включає пристрої введення та виведення, пристрої зберігання та пристрої зв’язку [3].

Сьогодні в усьому світі навіть найменші фірми, а також багато домогосподарств володіють або орендують комп'ютери. Люди можуть мати кілька комп’ютерів у вигляді смартфонів, планшетів та інших переносних пристроїв. Великі організації зазвичай використовують розподілені комп’ютерні системи, від потужних серверів з паралельною обробкою даних, розташованих у центрах обробки даних, до широко розкиданих персональних комп’ютерів і мобільних пристроїв, інтегрованих в організаційні інформаційні системи. Датчики стають все більш широко поширеними у фізичному та біологічному середовищі для збору даних і, у багатьох випадках, для здійснення контролю за допомогою пристроїв, відомих як приводи. Разом із периферійним обладнанням, пристрої введення-виведення та телекомунікаційне обладнання становлять апаратне забезпечення інформаційних систем. Все частіше комп’ютерні послуги та послуги зберігання надаються з хмари — зі спільних об’єктів, доступ до яких здійснюється через телекомунікаційні мережі.

1.1.2 Програмне забезпечення

Комп’ютерне програмне забезпечення поділяється на два великі класи: системне програмне забезпечення та прикладне програмне забезпечення[4]. Основним системним програмним забезпеченням є операційна система. Він керує обладнанням, даними та програмними файлами та іншими системними ресурсами та надає користувачеві засоби керування комп’ютером, як правило, через графічний інтерфейс користувача (GUI). Прикладне програмне забезпечення – це програми,

призначені для вирішення конкретних завдань користувачів. Додатки для смартфонів стали звичайним способом доступу людей до інформаційних систем. Великі фірми використовують ліцензовані програми, розроблені та підтримувані спеціалізованими компаніями, що займаються програмним забезпеченням, налаштовуючи їх відповідно до своїх конкретних потреб, а також розробляють інші програми власними силами або на сторонніх підприємствах. Компанії також можуть використовувати програми, які постачаються як software-as-a-service (SaaS). Software-as-a-service (SaaS) — це модель розповсюдження програмного забезпечення, за якої хмарний постачальник розміщує програми та робить їх доступними для кінцевих користувачів через Інтернет [5]. У цій моделі незалежний постачальник програмного забезпечення (ISV - independent software vendor) може укласти контракт зі стороннім постачальником хмарних технологій для розміщення програми.

1.1.3 Комунікації

Комунікації використовуються для з’єднання комп’ютерних систем і переносних пристроїв, а також для передачі інформації. З’єднання встановлюються через дротове або бездротове середовище. Дротові технології включають коаксіальний кабель і волоконну оптику. Бездротові технології, переважно засновані на передачі мікрохвиль і радіохвиль, також вони підтримують мобільні обчислення. Поширені інформаційні системи виникли з обчислювальними пристроями, вбудованими в багато різних фізичних об’єктів. Наприклад, такі датчики, як пристрої радіочастотної ідентифікації (RFID), можна приєднати до продуктів, що переміщуються через ланцюг поставок, щоб забезпечити відстеження їхнього розташування та моніторингу їх стану. Бездротові сенсорні мережі, інтегровані в Інтернет, можуть створювати величезні обсяги даних, які можна використовувати для підвищення продуктивності або моніторингу навколишнього середовища.

Залежно від потреб організації можливі різні конфігурації комп’ютерної мережі. Локальні мережі (LAN) об’єднують комп’ютери в певному місці, наприклад в офісній будівлі або академічному містечку. Столичні мережі (MAN) охоплюють обмежену густонаселену територію та є електронною інфраструктурою «розумного міста». Глобальні мережі (WAN) з’єднують широко розподілені центри обробки даних, якими часто керують різні організації. Однорангові мережі без централізованого контролю забезпечують широкий обмін вмістом. Інтернет – це сукупність мереж, що об’єднує мільярди комп’ютерів, розташованих на всіх континентах. Через мережу користувачі отримують доступ до інформаційних ресурсів, таких як великі бази даних, і до інших осіб, таких як колеги, клієнти, друзі або люди, які поділяють їхні професійні чи приватні інтереси.

Розгалужена мережева інфраструктура підтримує зростаючий перехід до хмарних обчислень із ресурсами інформаційної системи, спільними для кількох компаній, що забезпечує ефективність використання та свободу локалізації центрів обробки даних. Програмно визначена мережа забезпечує гнучкий контроль телекомунікаційних мереж за допомогою алгоритмів, які реагують на вимоги в реальному часі та доступність ресурсів.

1.1.4 Бази даних і сховища даних

Багато інформаційних систем — це головним чином засоби доставки даних, що зберігаються в базах даних. База даних — це набір взаємопов’язаних даних, організованих таким чином, що окремі записи або групи записів можна отримати відповідно до різних критеріїв. База даних — це систематизоване зібрання даних. Вони підтримують електронне зберігання та маніпулювання даними. Бази даних спрощують керування даними. [6]. Типовими прикладами баз даних є записи про співробітників і каталоги продукції. Бази даних підтримують операції та функції

управління підприємством. Сховища даних містять архівні дані, зібрані протягом тривалого часу, які можна отримати, щоб розробити та вивести на ринок нові продукти, краще обслуговувати існуючих клієнтів або охопити потенційних нових клієнтів. Будь-хто, хто коли-небудь купував щось за допомогою кредитної картки — особисто, поштою чи через Інтернет — включається до таких колекцій даних.

Масовий збір і обробка кількісних або структурованих даних, а також текстових даних, які часто збираються в Інтернеті, перетворилися на широку ініціативу, відому як «великі дані». Рішення, засновані на фактах, відображених у великих даних, можуть отримати багато переваг. Приклади включають доказову медицину, економію ресурсів у результаті уникнення марнотратства та рекомендації щодо нових продуктів на основі інтересів користувача. Великі дані створюють інноваційні бізнес-моделі. Наприклад, комерційна фірма збирає ціни на товари за допомогою краудсорсингу (збір від багатьох незалежних осіб) за допомогою смартфонів по всьому світу. Зведені дані надають завчасну інформацію про рух цін, забезпечуючи більш оперативне прийняття рішень, ніж це було можливо раніше.

Обробка текстових даних, таких як відгуки та думки, висловлені особами в соціальних мережах, блогах і на форумах, дозволяє автоматизовано аналізувати настрої для маркетингу, конкурентної розвідки, розробки нових продуктів та інших цілей прийняття рішень.

1.1.5 Людські ресурси та процедури

Кваліфіковані люди є життєво важливим компонентом будь-якої інформаційної системи. До технічного персоналу входять менеджери з розробки та операцій, бізнес-аналітики, системні аналітики та дизайнери, адміністратори баз даних, програмісти, спеціалісти з комп’ютерної безпеки та оператори комп’ютерів. Крім того, усі працівники організації повинні бути навчені якомога повніше використовувати можливості

інформаційних систем. Мільярди людей у всьому світі вивчають інформаційні системи, користуючись Інтернетом.

Процедури використання, експлуатації та підтримки інформаційної системи є частиною її документації. Наприклад, необхідно встановити процедури для виконання програми розрахунку заробітної плати, зокрема, коли її запускати, хто має право запускати її та хто має доступ до результатів. В ініціативі автономних обчислень центри обробки даних дедалі частіше працюють автоматично з процедурами, вбудованими в програмне забезпечення, яке контролює ці центри.

1.2 Типи мереж за просторовим охопленням

Комп'ютерна мережа — це набір комп'ютерів, які спільно використовують ресурси, розташовані на вузлах мережі або надані ними. Комп'ютери використовують загальні протоколи зв'язку через цифрові з'єднання для зв'язку один з одним. Ці взаємозв'язки складаються з технологій телекомунікаційних мереж, заснованих на фізично дротових, оптичних і бездротових радіочастотних методах, які можуть бути організовані в різноманітних мережевих топологіях.

Вузли комп'ютерної мережі можуть включати персональні комп'ютери, сервери, мережеве обладнання або інші спеціалізовані хости чи хости загального призначення. Вони ідентифікуються мережевими адресами та можуть мати імена хостів. Імена хостів служать мітками для вузлів, які рідко змінюються після початкового призначення. Мережні адреси служать для визначення місцезнаходження та ідентифікації вузлів за протоколами зв'язку, такими як Інтернет-протокол.

Комп'ютерні мережі можна класифікувати за багатьма критеріями, включаючи середовище передачі, що використовується для передачі сигналів, пропускну здатність, протоколи зв'язку для організації мережевого трафіку, розмір мережі, топологію, механізм керування трафіком та організаційні наміри.

Мережі можуть характеризуватися багатьма властивостями або особливостями, такими як фізична ємність, організаційне призначення, авторизація користувачів, права доступу та інші. Також відмінним методом класифікації є фізичний протяжність або географічний масштаб. На рисунку 1 визначення класифікація мереж за просторовим охопленням.

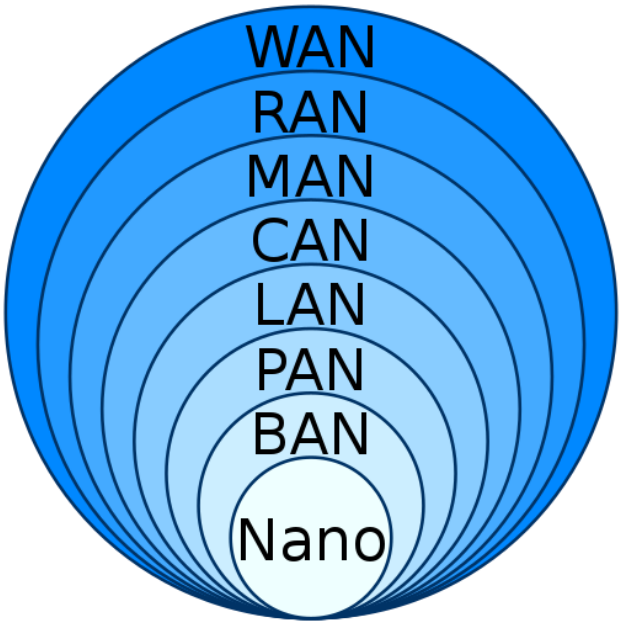


Рис.1.1 Класифікація мереж за просторовим охопленням

Мережа на тілі (BAN - body area network)— це бездротова мережа обчислювальних пристроїв, які можна носити [7]. Пристрої BAN можуть бути вбудовані в тіло як імплантати, можуть бути встановлені на поверхні тіла у фіксованому положенні або можуть супроводжуватися пристроями, які люди можуть носити в різних положеннях, наприклад як у кишенях одягу, в руках або в різних сумках [8].

Нанорозмірна мережа має ключові компоненти, реалізовані на нанорозмірі, включаючи носії повідомлень, і використовує фізичні принципи, які відрізняються від механізмів комунікації на макророзмірі. Нанорозмірна комунікація поширює комунікацію на дуже маленькі датчики та виконавчі механізми, що знаходяться в біологічних системах, а також працює в середовищах, які були б надто жорсткими для інших методів комунікації [9].

Персональна мережа (PAN - personal area network) — це комп’ютерна мережа, яка використовується для зв’язку між комп’ютерами та різними інформаційно-технологічними пристроями, розташованими поблизу однієї особи. Деякими прикладами пристроїв, які використовуються в PAN, є персональні комп’ютери, принтери, телефони, сканери та ігрові консолі. PAN може включати дротові та бездротові пристрої. Радіус дії PAN зазвичай досягає 10 метрів [10]. Дротова PAN зазвичай складається з підключеннями USB і FireWire, тоді як такі технології, як Bluetooth і інфрачервоний зв’язок, зазвичай утворюють бездротову PAN.

Локальна мережа (LAN - local area network) — це мережа, яка об’єднує комп’ютери та пристрої в обмеженій географічній зоні, наприклад у домі, школі, офісній будівлі або групі будівель, розташованих близько. Дротові локальні мережі найчастіше базуються на технології Ethernet. Інші мережеві технології, такі як ITU-T G.hn, також забезпечують спосіб створення дротової локальної мережі за допомогою існуючої проводки, такої як коаксіальні кабелі, телефонні лінії та лінії електропередач [11].

LAN можна підключити до глобальної мережі (WAN - wide area network) за допомогою маршрутизатора. Визначальними характеристиками локальної мережі, на відміну від глобальної мережі, є вищі швидкості передачі даних, обмежений географічний діапазон і відсутність опори на орендовані лінії для забезпечення з’єднання. Поточні технології Ethernet або інші технології локальної мережі IEEE 802.3 працюють при передачі даних. швидкості до та понад 100 Гбіт/с [12], стандартизовані IEEE у 2010 році.

Домашня мережа (HAN - home area network) — це домашня локальна мережа, яка використовується для зв’язку між цифровими пристроями, зазвичай розгорнутими вдома, як правило, невеликою кількістю

персональних комп’ютерів і аксесуарів, таких як принтери та мобільні комп’ютерні пристрої. Важливою функцією є спільне використання доступу до Інтернету, часто широкосмугового доступу через кабельний доступ до Інтернету або постачальника цифрової абонентської лінії (DSL).

Кампусна мережа (CAN - campus area network) складається із з’єднання локальних мереж у межах обмеженої географічної області. Мережеве обладнання (комутатори, маршрутизатори) і засоби передачі (оптичне волокно, кабелі Cat5 тощо) майже повністю належать орендареві або власнику кампусу (підприємство, університет, уряд тощо).

Магістральна мережа є частиною інфраструктури комп’ютерної мережі, яка забезпечує шлях для обміну інформацією між різними локальними мережами або підмережами. Магістраль може об’єднувати різноманітні мережі в одній будівлі, у різних будівлях або на великій території. Під час проектування магістралі мережі продуктивність мережі та перевантаженість мережі є критичними факторами, які слід брати до уваги. Зазвичай пропускна здатність магістральної мережі більша, ніж пропускна здатність окремих мереж, підключених до неї.

Столична мережа (MAN - metropolitan area network) — це комп’ютерна мережа, яка об’єднує користувачів із комп’ютерними ресурсами в географічному регіоні розміром із міську область. Термін MAN застосовується до з’єднання локальних мереж (LAN) у місті в єдину більшу мережу, яка також може запропонувати ефективне підключення до глобальної мережі. Термін також використовується для опису взаємозв’язку кількох локальних мереж у міському районі за допомогою з’єднань «точка-точка» між ними [13].

Мережа радіодоступу (RAN - radio access network) є частиною мобільної телекомунікаційної системи. Він реалізує технологію радіодоступу. Концептуально він знаходиться між таким пристроєм, як мобільний телефон, комп’ютер або будь-яка дистанційно керована

машина, і забезпечує з’єднання з його базовою мережею (CN). Залежно від стандарту, мобільні телефони та інші бездротові підключені пристрої по-різному відомі як обладнання користувача (UE), термінальне обладнання, мобільна станція (MS) тощо.

Глобальна мережа (WAN - wide area network) — це комп’ютерна мережа, яка охоплює велику географічну територію, наприклад місто, країну, або охоплює навіть міжконтинентальні відстані. WAN використовує канал зв’язку, який поєднує багато типів носіїв, таких як телефонні лінії, кабелі та радіохвилі. Глобальна мережа часто використовує засоби передачі, що надаються звичайними операторами, наприклад телефонними компаніями. Технології WAN зазвичай функціонують на трьох нижніх рівнях еталонної моделі OSI: фізичному рівні, рівні каналу даних і мережевому рівні [14].

Корпоративна приватна мережа — це мережа, яку будує одна організація для з’єднання своїх офісів (наприклад, виробничих ділянок, головних офісів, віддалених офісів, магазинів), щоб вони могли спільно використовувати комп’ютерні ресурси.

Віртуальна приватна мережа (VPN - virtual private network) розширює приватну мережу через загальнодоступну мережу та дозволяє користувачам надсилати й отримувати дані через спільні чи загальнодоступні мережі так, якби їхні комп’ютерні пристрої були безпосередньо підключені до приватної мережі [15]. Переваги VPN включають збільшення функціональності, безпеки та керування приватною мережею. Він забезпечує доступ до ресурсів, недоступних у загальнодоступній мережі, і зазвичай використовується для віддалених працівників.

Глобальна мережа (GAN - global area network) — це мережа, яка використовується для підтримки мобільного зв’язку в довільній кількості бездротових локальних мереж, зон покриття супутників тощо. Ключовою

проблемою мобільного зв’язку є передача зв’язку користувача з однієї локальної зони покриття в іншу.

Висновок

У розділі описані інформаційні системи та їх типи. Інформаційна система це інтегрований набір компонентів для збору, зберігання й обробки даних, а також для доставки інформації, знань і цифрових продуктів. Основними компонентами інформаційних систем є комп’ютерне обладнання та програмне забезпечення, телекомунікації, бази даних і сховища даних, людські ресурси та процедури.

Також була наведена класифікація та визначення мереж за просторовим охопленням. А саме класифікують наступні мережі: мережа на тілі (BAN - body area network), нанорозмірна мережа, персональна мережа (PAN - personal area network), локальна мережа (LAN - local area network), домашня мережа (HAN - home area network), кампусна мережа (CAN - campus area network), столична мережа (MAN - metropolitan area network), мережа радіодоступу (RAN - radio access network), глобальна мережа (WAN - wide area network), корпоративна приватна мережа), віртуальна приватна мережа (VPN - virtual private network) та глобальна мережа (GAN - global area network). Дані мережі мають різний принцип роботи, зону покриття, кількість хостів та/або локальних мереж.

РОЗДІЛ 2
ВІДОМОСТІ ПРО ПРИНЦИП РОБОТИ ТА АРХІТЕКТУРУ МЕРЕЖ

2.1 Архітектура мереж та протоколи

Мережа – це два або більше комп'ютерів, з'єднаних між собою для обміну інформацією. Зазвичай кожен підключений пристрій називають вузлом, щоб цей опис можна було застосувати до ширшого кола пристроїв.

2.1.1 Види мережевих архітектур

Існує безліч способів розробити мережеву архітектуру, але більшість з них належить до одного з двох типів. Це однорангові та клієнт/сервер архітектури [16].

У одноранговій моделі всі пристрої в мережі мають однакові обов’язки та привілеї один перед одним. Це означає, що завдання рівномірно розподіляються по всій мережі. Файли на одному комп’ютері можна спільно використовувати з будь-яким іншим комп’ютером, роблячи кожен вузол мережевим накопичувачем. Такі ресурси, як принтер, підключений до одного пристрою, також видимі для всіх інших пристроїв у мережі.

В архітектурі клієнт/сервер усі пристрої в мережі, які називаються «клієнтами», підключені до центрального вузла, який називається «сервер». Сервер виконує основну частину мережевих операцій – зберігання даних, обробку клієнтських запитів, кібербезпеку та контроль доступу.

Більшість великих мереж, таких як WAN, часто використовують модель клієнт/сервер. У цьому випадку клієнтським пристроєм є ваш комп’ютер або смартфон. Клієнт/сервер також є кращою архітектурою корпоративної мережі.

Існує також гібридна архітектура під назвою периферійні обчислення, яка стає все більш популярною в Інтернеті речей (IoT). Це схоже на архітектуру клієнт/сервер. Однак замість того, щоб сервер відповідав за всі завдання зберігання та обробки, деякі з них делегуються комп'ютерами, розташованими ближче до клієнтської машини, які називаються периферійними пристроями.

2.1.2 Складові блоки мережевої архітектури

Проектування будь-якої архітектури цифрової мережі передбачає оптимізацію її складових блоків. До них належать апаратне забезпечення, носії передачі, протоколи та топологія.

Апаратне забезпечення – це обладнання, яке утворює компоненти мережі, наприклад пристрої користувача (ноутбуки, комп'ютери, мобільні телефони), маршрутизатори, сервери та шлюзи. Таким чином, у певному сенсі мета будь-якої мережевої архітектури полягає в тому, щоб знайти найефективніший спосіб передачі даних від однієї апаратної точки до іншої.

Носії передачі – це фізичні з'єднання між апаратними пристроями в мережі. Різні носії мають різні властивості, які визначають швидкість переміщення даних від однієї точки до іншої.

Вони бувають двох видів: дротові та бездротові. Дротові носії включають фізичні кабелі для підключення. Приклади включають коаксіальний і волоконно-оптичний. Бездротові медіа, з іншого боку, покладаються на мікрохвильові або радіосигнали. Найпопулярнішими прикладами є Wi-Fi і стільниковий зв'язок.

Протоколи – це набір правил або процедур для передачі даних між електронними пристроями, наприклад комп'ютерами. Для того, щоб комп'ютери могли обмінюватися інформацією, має існувати домовленість про те, як буде структурована інформація та як кожна сторона надсилатиме та отримуватиме її. Без протоколу комп'ютер, який передає, наприклад, міг

би надсилати свої дані у 8-бітних пакетах, тоді як комп'ютер-одержувач міг би очікувати даних у 16-бітних пакетах. Протоколи розробляються міжнародними або галузевими організаціями. Найважливішим комп'ютерним протоколом є OSI (Open Systems Interconnection), набір інструкцій для реалізації мережевого зв'язку між комп'ютерами. Серед найважливіших наборів Інтернет-протоколів є TCP/IP (Transmission Control Protocol/Internet Protocol), HTTPS (Secure HyperText Transmission Protocol), SMTP (Simple Mail Transfer Protocol) і DNS (Domain Name System) [17].

Топологія - це структура мережі. Це важливо, оскільки такі фактори, як відстань між мережевими пристроями, впливатимуть на те, як швидко дані можуть досягати місця призначення, впливаючи на продуктивність. Існують різні топології мереж, кожна з яких має сильні та слабкі сторони [18].

Наприклад, зіркоподібна топологія описує макет, у якому всі пристрої в мережі підключені до центрального концентратора. Перевагою такої схеми є те, що можна легко підключати пристрої до мережі. Однак, якщо центральний концентратор виходить з ладу, вся мережа виходить з ладу.

З іншого боку, топологія шина - це те, де всі мережеві пристрої підключені до одного шляху, який називається шиною. Шина діє як магістраль, яка переносить дані від однієї частини мережі до іншої. Незважаючи на те, що він дешевий і простий у застосуванні, його продуктивність має тенденцію до сповільнення, коли до мережі додається більше пристроїв. Сьогодні більшість мережевих архітектур використовують гібридну топологію, поєднуючи різні топології, щоб компенсувати слабкі місця.

2.1.3 Переваги та недоліки різних видів мережевих архітектур

Різні мережеві архітектури мають свої плюси та мінуси; і знання їх є ключем до вибору правильної архітектури згідно різних потреб та вимог [19].

Однорангові моделі часто недорогі та прості в установці, оскільки не потрібно інвестувати в потужний сервер. Теоретично все, що вам потрібно, це мережеві кабелі або маршрутизатор. Він також досить міцний, якщо один комп'ютер виходить з ладу, мережа залишається працювати. Розподілений характер також зменшує або принаймні розподіляє навантаження на мережу, щоб запобігти перевантаженням.

Однак одноранговими моделями важче керувати. Оскільки централізованого концентратора немає, вам потрібно буде налаштувати кожен комп'ютер окремо. Таким чином, однорангові мережі також менш безпечні. Для викрадення мережі достатньо одного зламаного комп'ютера.

Моделями клієнт/сервер, з іншого боку, легше керувати, оскільки вони мають централізований підхід. Ви можете налаштувати права доступу, брандмауери та проксі-сервери, щоб підвищити безпеку мережі. Таким чином, налаштування клієнт/сервер найкраще підходить для великих мереж на великій відстані.

Недоліком цього підходу є те, що архітектура клієнт/сервер є дорожчою для налаштування, оскільки вам потрібен потужний сервер, щоб справлятися з навантаженням мережі. Крім того, для керування сервером потрібен спеціальний адміністратор.

Але найбільший недолік моделі клієнт/сервер полягає в тому, що сервер є слабкою ланкою. Якщо сервер вимикається, вся мережа вимикається. Таким чином, безпека часто є найнадійнішою на сервері та поблизу нього.

2.2 Набір Інтернет протоколів

Набір Інтернет протоколів, широко відомий як TCP/IP, є основою для організації набору протоколів зв’язку, що використовуються в Інтернеті та подібних комп’ютерних мережах відповідно до функціональних критеріїв. Основними протоколами в наборі є Transmission Control Protocol (TCP), User Datagram Protocol (UDP) і Internet Protocol (IP) [20].

TCP/IP визначає спосіб обміну даними через Інтернет, забезпечуючи наскрізний зв’язок, який визначає, як їх потрібно розбивати на пакети, адресувати, передавати, маршрутизувати та отримувати в пункті призначення. Протокол TCP/IP не потребує централізованого керування та створений для забезпечення надійності мереж із можливістю автоматичного відновлення після збою будь-якого пристрою в мережі.

Два основних протоколи виконують певні функції. TCP визначає, як програми можуть створювати канали зв'язку в мережі. Він також керує тим, як повідомлення збирається в менші пакети, перш ніж вони потім будуть передані через Інтернет і повторно зібрані в правильному порядку за адресою призначення.

IP визначає, як адресувати та маршрутизувати кожен пакет, щоб переконатися, що він досягає правильного пункту призначення. Кожен шлюзовий комп’ютер у мережі перевіряє цю IP-адресу, щоб визначити, куди пересилати повідомлення.

Маска підмережі повідомляє комп’ютеру чи іншому мережевому пристрою, яка частина IP-адреси використовується для представлення мережі, а яка – для представлення хостів або інших комп’ютерів у мережі.

Трансляція мережевих адрес (NAT) — це віртуалізація IP-адрес. NAT допомагає підвищити безпеку та зменшити кількість IP-адрес, необхідних організації [21].

Ранній архітектурний документ, RFC 1122 під назвою «Вимоги до хосту» структурований у параграфах, які стосуються рівнів, але документ

посилається на багато інших архітектурних принципів і не наголошує на рівнях. Він узагальнено визначає чотирирівневу модель, яка складається з прикладного, транспортного, мережевого та канального рівнях[22].

2.2.1 Прикладний рівень

Прикладний рівень включає протоколи, які використовуються більшістю додатків для надання послуг користувачам або обміну даними додатків через мережеві з'єднання, встановлені протоколами нижчого рівня. Це може включати деякі основні служби підтримки мережі, такі як протоколи маршрутизації та конфігурація хоста. Приклади протоколів прикладного рівня включають протокол передачі гіпертексту (HTTP - Hypertext Transfer Protocol), протокол передачі файлів (FTP - File Transfer Protocol), простий протокол передачі пошти (SMTP - Simple Mail Transfer Protocol) і протокол динамічної конфігурації хоста (DHCP - Dynamic Host Configuration Protocol) [23]. Дані, закодовані відповідно до протоколів прикладного рівня, інкапсулюються в одиниці протоколу транспортного рівня, які, у свою чергу, використовують протоколи нижчого рівня для фактичної передачі даних.

Модель TCP/IP не враховує особливості форматування та представлення даних і не визначає додаткових рівнів між прикладним і транспортним рівнями, як у моделі OSI (рівні представлення та сеансу). Відповідно до моделі TCP/IP, такі функції є сферою бібліотек та інтерфейсів прикладного програмування.

Протоколи прикладного рівня часто пов'язані з конкретними клієнт-серверними програмами, а звичайні служби мають добре відомі номери портів, зарезервовані Управлінням розподілення номерів в Інтернеті (IANA - Internet Assigned Numbers Authority). Наприклад, HTTP використовує порт сервера 80, а Telnet — порт сервера 23. Клієнти, які підключаються до служби, зазвичай використовують тимчасові порти,

тобто номери портів, призначені лише на час транзакції випадковим чином або з певного діапазону, налаштованого в додаток.

На прикладному рівні модель TCP/IP розрізняє протоколи користувача та протоколи підтримки [24]. Протоколи підтримки надають послуги системі мережевої інфраструктури. Користувальницькі протоколи використовуються для реальних програм користувача. Наприклад, FTP є протоколом користувача, а DNS є протоколом підтримки.

Хоча програми зазвичай знають ключові якості з'єднання транспортного рівня, такі як IP-адреси кінцевих точок і номери портів, протоколи прикладного рівня зазвичай сприймають протоколи транспортного рівня (і нижчих) як чорні ящики, які забезпечують стабільне мережеве з'єднання, через яке можна спілкуватися. Маршрутизатори та комутатори зазвичай не перевіряють інкапсульований трафік, скоріше вони просто забезпечують його канал. Однак деякі брандмауери та програми для обмеження смуги пропускання використовують глибоку перевірку пакетів для інтерпретації даних програми. Прикладом є протокол резервування ресурсів (RSVP). Додаткам, на які впливає NAT, також іноді необхідно враховувати корисне навантаження програми.

2.2.2 Транспортний рівень

Транспортний рівень встановлює основні канали даних, які додатки використовують для обміну даними для конкретного завдання. Рівень встановлює з'єднання хост-хост у формі наскрізних служб передачі повідомлень, які не залежать від основної мережі та не залежать від структури даних користувача та логістики обміну інформацією. Підключення на транспортному рівні можна класифікувати як орієнтоване на підключення, реалізоване в TCP, або без підключення, реалізоване в UDP. Протоколи на цьому рівні можуть забезпечувати контроль помилок, сегментацію, контроль потоку, контроль перевантаження та адресацію додатків (номера портів).

З метою забезпечення специфічних для процесу каналів передачі для додатків рівень встановлює концепцію мережевого порту. Це пронумерована логічна конструкція, виділена спеціально для кожного каналу зв'язку, який потрібен програмі. Для багатьох типів служб ці номери портів стандартизовані, щоб клієнтські комп'ютери могли звертатися до певних служб комп'ютера-сервера без залучення служб виявлення служб або служб каталогів.

Новітній протокол передачі керування потоком (SCTP - Stream Control Transmission Protocol) також є надійним транспортним механізмом, орієнтованим на підключення. Він орієнтований на потік повідомлень, а не на потік байтів, як TCP, і забезпечує кілька потоків, мультиплексованих через одне з'єднання [25]. Він також забезпечує підтримку багатоадресного підключення, у якому кінець з'єднання може бути представлений декількома IP-адресами (що представляють кілька фізичних інтерфейсів), так що в разі збою одного з'єднання підключення не переривається.

Надійність також може бути досягнута за допомогою IP через надійний протокол передачі даних, такий як High-Level Data Link Control (HDLC).

Протокол дейтаграм користувача (UDP - User Datagram Protocol) — це протокол дейтаграм без підключення. Надійність протоколу вирішується шляхом виявлення помилок за допомогою алгоритму контрольної суми. UDP зазвичай використовується для таких додатків, як потокове передавання медіа (аудіо, відео, голос через IP тощо), де своєчасне надходження важливіше, ніж надійність, або для простих програм запитів/відповідей, таких як пошук DNS. Транспортний протокол реального часу (RTP - Real-time Transport Protocol) — це протокол дейтаграм, який використовується через UDP і призначений для передачі даних у реальному часі [26].

Транспортний рівень моделі TCP/IP або рівень хост-хост приблизно відповідає четвертому рівню моделі OSI, який також називають транспортним рівнем.

2.2.3 Мережевий рівень

Мережевий рівень, який також називають Інтернет-рівнем, працює з пакетами та з'єднує незалежні мережі для транспортування пакетів через межі мережі. Протоколами мережевого рівня є IP та протокол керування повідомленнями Інтернету (ICMP - Internet Control Message Protocol), який використовується для звітування про помилки.

Робота в Інтернеті вимагає надсилання даних із вихідної мережі в мережу призначення. Цей процес називається маршрутизацією та підтримується адресацією та ідентифікацією хостів за допомогою ієрархічної системи адресації IP. Інтернет-рівень забезпечує можливість передачі дейтаграм між хостами, розташованими в потенційно різних мережах IP, пересилаючи дейтаграми на відповідний наступний маршрутизатор для подальшої ретрансляції до місця призначення. Інтернет-рівень відповідає за надсилання пакетів через потенційно кілька мереж. Завдяки цій функціональності Інтернет-рівень робить можливим міжмережевий зв'язок, взаємодію різних IP-мереж, і, по суті, створює Інтернет. Інтернет-рівень не розрізняє різні протоколи транспортного рівня. Він передає дані для різних протоколів верхнього рівня. Кожен із цих протоколів ідентифікується унікальним номером протоколу.

Інтернет-протокол є основним компонентом Інтернет-рівня, і він визначає дві системи адресації для ідентифікації мережевих хостів і їх розташування в мережі. Оригінальною адресною системою ARPANET і її наступника, Інтернету, є Інтернет-протокол версії 4 (IPv4). Він використовує 32-розрядну IP-адресу і тому здатний ідентифікувати приблизно чотири мільярди хостів. Це обмеження було усунено в 1998 році стандартизацією Інтернет-протоколу версії 6 (IPv6), який використовує

128-бітні адреси. Виробнича реалізація IPv6 з’явилася приблизно в 2006 році.

2.2.4 Канальний рівень

Рівень мережевого інтерфейсу або рівень каналу даних складається з протоколів, які працюють лише на каналі - мережевий компонент, який з'єднує вузли або хости в мережі. Протоколи цього найнижчого рівня включають Ethernet для локальних мереж і протокол розпізнавання адрес.

Протоколи канального рівня працюють у межах з’єднання локальної мережі, до якого приєднаний хост. Канал включає всі хости, доступні без перетину маршрутизатора. Тому розмір каналу визначається структурою мережевого обладнання. В принципі, TCP/IP розроблено таким чином, щоб бути незалежним від апаратного забезпечення та може бути реалізовано поверх практично будь-якої технології канального рівня. Це стосується не лише апаратних реалізацій, але й віртуальних рівнів зв’язку, таких як віртуальні приватні мережі та мережеві тунелі.

Канальний рівень використовується для переміщення пакетів між інтерфейсами Інтернет-рівня двох різних хостів на одному каналі. Процесами передачі і прийому пакетів по каналу можна керувати в драйвері пристрою для мережевої карти, а також в мікропрограмі або спеціалізованих чіпсетах. Вони виконують такі функції, як формування кадрів, для підготовки пакетів рівня Інтернету до передачі та, нарешті, передають кадри на фізичний рівень і через середовище передачі. Модель TCP/IP включає специфікації для перетворення методів мережевої адресації, що використовуються в Інтернет-протоколі, на адреси канального рівня, такі як адреси керування доступом до середовища (MAC).

2.3 Інкапсуляція даних

Інкапсуляція - це метод побудови модульних мережевих протоколів, при якому логічно незалежні функції мережі абстрагуються від механізмів,

що знаходяться нижче, шляхом включення або інкапсулювання цих механізмів в більш високорівневі об'єкти [27].

2.3.1 Рівні при інкапсуляції даних

На кожному рівні блок даних протоколу містить корисні дані, що передаються. Зазвичай до корисних даних додається заголовок, що містить необхідну інформацію для передачі корисного навантаження даних, таку як адреси вузлів джерела і призначення в мережі.

На рис.2.1 показано, як блоки даних протоколу розміщуються в IPS.

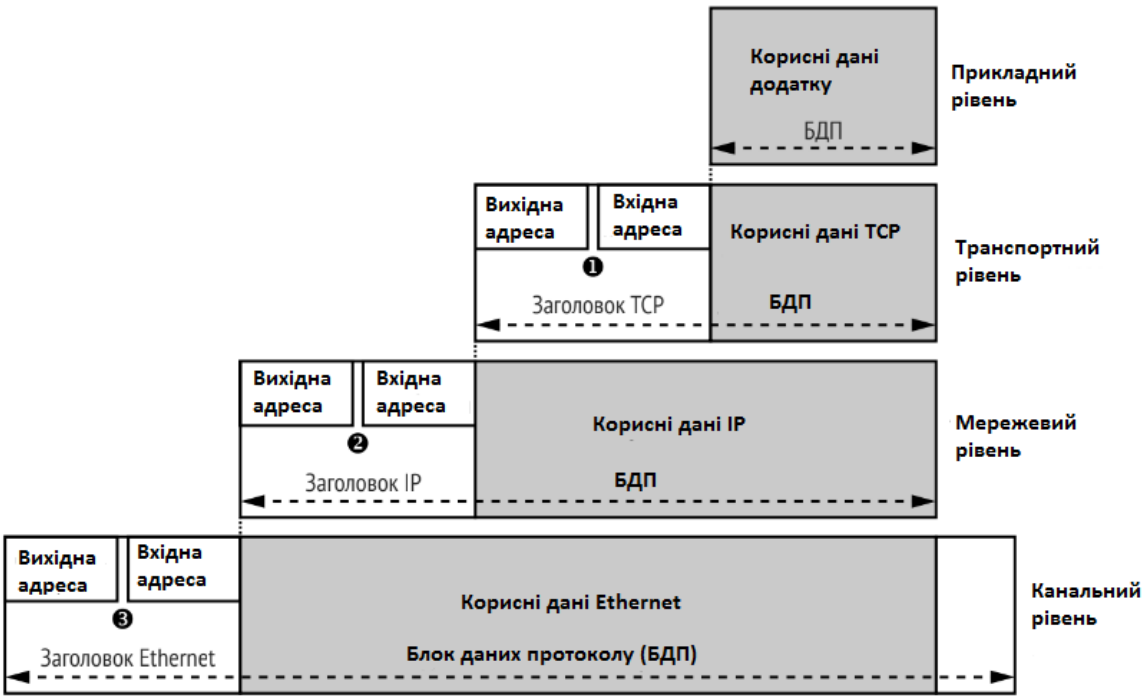


Рис.2.1 Інкапсуляція даних IPS

Заголовок TCP містить номер вихідного порту та порту призначення 1. Ці номери портів дозволяють одному вузлу мати декілька унікальних мережних з'єднань. Номери портів для протоколу TCP (і UDP) знаходяться в діапазоні від 0 до 65535.

Більшість номерів портів надаються новим з'єднанням у міру потреби, але є й особливі випадки, наприклад порт 80 для HTTP. Корисні дані та заголовок TCP зазвичай називаються сегментом, тоді як корисні дані та заголовок UDP – дейтаграмою.

Протокол IP використовує адресу джерела та адресу призначення 2. Адреса призначення (вхідна адреса) дозволяє надсилати дані на певний вузол у мережі. Адреса джерела дозволяє одержувачу даних дізнатися, який вузол надіслав дані, і дає можливість одержувачу відповісти відправнику. IPv4 використовує 32-бітові адреси, які зазвичай записуються у вигляді чотирьох чисел, розділених точками, наприклад, 192.168.10.1. IPv6 використовує 128-бітові адреси, тому що 32-бітових адрес недостатньо для кількості вузлів у сучасних мережах.

Адреси IPv6 зазвичай записуються у вигляді шістнадцяткових чисел, розділених двокрапками, наприклад fe80:0000:0000:0000:897b:581e:44b0:2057. Довгі рядки з 0000 можна записувати з використанням символу подвійного двокрапки. Наприклад, попередню IPv6-адресу можна також записати як fe80::897b:581e:44b0:2057. Корисні дані та заголовок протоколу IP зазвичай називаються пакетом.

Ethernet також містить адреси джерела та призначення 3. Ethernet використовує 64-бітове значення, яке називають MAC-адреса. Як правило, MAC-адреса встановлюється під час виготовлення адаптера Ethernet. Зазвичай ці адреси записуються у вигляді серії шістнадцяткових чисел, розділених дефісом або двокрапкою, наприклад 0A-00-27-00-00-0E. Корисні дані Ethernet, включаючи заголовок та кінцевик, зазвичай називаються кадром.

2.3.2 Передача даних

На рис. 2.2 показана проста мережа Ethernet із трьома вузлами. У цьому прикладі вузол з IP-адресою 192.1.1.101 (1) хоче надіслати дані протоколу IP на вузол 2 з IP-адресою 192.1.1.50. (Комутатор 3 пересилає кадри Ethernet між усіма вузлами в мережі. Комутатору не потрібна IP-адреса, тому що він працює тільки на канальному рівні.) Ось що відбувається при передачі даних між двома вузлами [28].

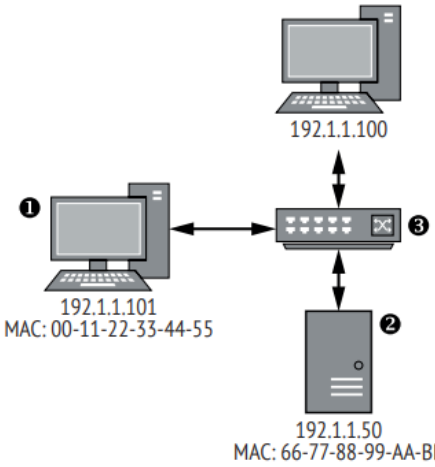


Рис.2.2 Проста мережа Ethernet

1. Вузол мережного стека операційної системи 1 інкапсулює дані прикладного та транспортного рівнів та створює IP-пакет з адресою відправника 192.1.1.101 та адресою призначення 192.1.1.50.

2. На цьому етапі операційна система може інкапсулювати IP-дані як кадр Ethernet, але вона може не знати MAC-адресу цільового вузла. Вона може запросити MAC-адресу для певної IP-адреси за допомогою протоколу ARP, який надсилає запит усім вузлам у мережі, щоб знайти MAC-адресу для IP-адреси призначення.

3. Як тільки вузол 1 отримує ARP-відповідь, він може побудувати кадр, задаючи в якості адреси відправника локальну MAC-адресу 00-11-22-33-44-55 та адресу призначення 66-77-88-99-AA-BB . Новий кадр передається по мережі та приймається комутатором 3.

4. Комутатор пересилає кадр на вузол призначення, який розпаковує IP-пакет та перевіряє відповідність IP-адреси призначення. Потім корисні дані IP витягуються і передаються вгору по стеку для прийому додатком, що очікує.

2.3.3 Мережева маршрутизація

Ethernet вимагає, щоб всі вузли були безпосередньо підключені до однієї локальної мережі. Ця вимога є серйозним обмеженням для по-справжньому глобальної мережі, оскільки фізично з'єднувати вузли один з

одним неможливо [29]. Замість того щоб вимагати прямого підключення всіх вузлів, адреси відправника та одержувача дозволяють маршрутизувати дані по різних мережах до тих пір, поки ті не досягнуть потрібного вузла призначення, як показано на рис. 1.6.

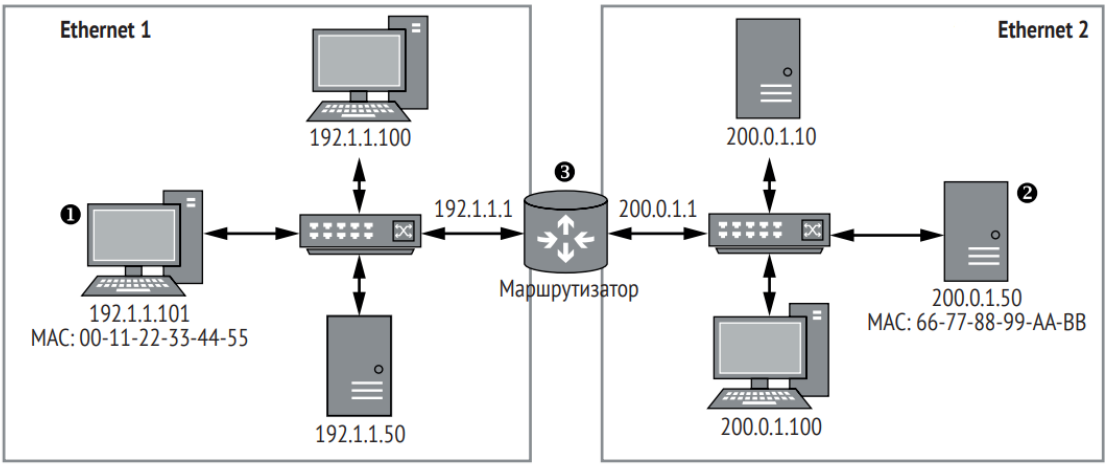


Рис.2.3 Приклад маршрутизованої мережі, яка з'єднує дві мережі Ethernet

На малюнку видно дві мережі Ethernet, кожна з яких має окремі діапазони IP-адрес. Наступний опис пояснює, як IP використовує цю модель для відправки даних від вузла 1 в мережі 1 до вузла 2 мережі 2.

1. Вузол мережного стека операційної системи 1 інкапсулює дані прикладного та транспортного рівнів та створює IP-пакет з адресою відправника 192.1.1.101 та адресою одержувача 200.0.1.50.

2. Мережному стеку необхідно надіслати кадр Ethernet, але оскільки IP-адреса призначення не існує в жодній мережі Ethernet, до якої підключено вузол, стек звертається до таблиці маршрутизації операційної системи. У цьому прикладі таблиця маршрутизації містить запис IP-адреси 200.0.1.50. Запис вказує на те, що маршрутизатор 3 на IP-адресі 192.1.1.1 знає, як дістатися цієї адреси призначення.

3. Операційна система використовує протокол ARP для пошуку MAC-адреси маршрутизатора за адресою 192.1.1.1, а вихідний IP-пакет інкапсулюється в Ethernet кадр з цією MAC-адресою.

4. Маршрутизатор отримує кадр Ethernet та розпаковує IP-пакет. Коли маршрутизатор перевіряє IP-адресу призначення, він визначає, що IP-пакет призначений не для маршрутизатора, а іншого вузла в іншій підключеної мережі. Маршрутизатор шукає MAC-адресу 200.0.1.50, інкапсулює вихідний IP-пакет у новий кадр Ethernet і відправляє його до мережі 2.

5. Вузол призначення отримує кадр Ethernet, розпаковує IP-пакет та обробляє його вміст.

Цей процес маршрутизації може повторюватися кілька разів. Наприклад, якщо маршрутизатор не був підключений до мережі, що містить вузол 200.0.1.50 безпосередньо, він звірився б зі своєю таблицею маршрутизації і визначив наступний маршрутизатор, якому він міг би відправити IP-пакет.

Очевидно, що для кожного вузла мережі було б непрактично з'ясовувати, як дістатися іншого вузла в інтернеті. Якщо для пункту призначення немає явного запису маршрутизації, операційна система надає запис у таблиці маршрутизації за промовчанням, яка називається стандартним шлюзом. Вона містить IP-адресу маршрутизатора, яка може пересилати IP-пакети за призначенням.

Висновок

У цьому розділі було представлено короткий огляд основ мереж, їх роботу та архітектуру.

Було визначено, що існує безліч способів розробити мережеву архітектуру, але більшість з них належить до одного з двох типів. Це однорангові та клієнт/сервер архітектури.

Проектування будь-якої архітектури цифрової мережі передбачає оптимізацію її складових блоків. До них належать апаратне забезпечення, носії передачі, протоколи та топологія.

Різні мережеві архітектури мають свої плюси та мінуси; і знання їх є ключем до вибору правильної архітектури згідно різних потреб та вимог.

Набір Інтернет протоколів, широко відомий як TCP/IP, є основою для організації набору протоколів зв’язку, що використовуються в Інтернеті та подібних комп’ютерних мережах відповідно до функціональних критеріїв. Основними протоколами в наборі є Transmission Control Protocol (TCP), User Datagram Protocol (UDP) і Internet Protocol (IP)

Ранній архітектурний документ, RFC 1122 під назвою «Вимоги до хосту» структурований у параграфах, які стосуються рівнів, але документ посилається на багато інших архітектурних принципів і не наголошує на рівнях. Він узагальнено визначає чотирирівневу модель, яка складається з прикладного, транспортного, мережевого та канального рівнях

Було розглянуто IPS, включаючи протоколи, які використовуються в реальних мережах, та показано як дані передаються між вузлами локальної мережі, а також у віддалених мережах за допомогою маршрутизації.

РОЗДІЛ 3
КРИПТОГРАФІЧНІ МЕТОДИ ЗАХИСТУ ТА ШИФРУВАННЯ

3.1 Поняття криптографічного шифрування

Шифрування даних — це метод захисту даних шляхом їх кодування таким чином, що їх може розшифрувати або отримати доступ до них тільки особа, яка має правильний ключ шифрування. Коли фізична чи юридична особа отримує доступ до зашифрованих даних без дозволу, вони виглядають зашифрованими або нечитаними.

Шифрування даних — це процес перетворення даних з формату, що читається, в зашифрований фрагмент інформації. Це зроблено для того, щоб сторонні не могли прочитати конфіденційні дані у дорозі. Шифрування може застосовуватися до документів, файлів, повідомлень або будь-якої іншої форми мережі.

Щоб зберегти цілісність даних, шифрування є життєво важливим інструментом, значення якого неможливо переоцінити. Майже все, що ми бачимо в Інтернеті, пройшло через певний рівень шифрування, будь то веб-сайти або програми [30]. Ручне шифрування використовувалося з римських часів, але цей термін став асоціюватися з маскуванням інформації за допомогою електронних комп’ютерів. Шифрування є базовим процесом криптології.

Комп’ютери шифрують дані шляхом застосування алгоритму, тобто набору процедур або інструкцій для виконання певного завдання, до блоку даних. Персональний ключ шифрування або ім’я, відоме лише передавачу повідомлення та його одержувачу, використовується для керування шифруванням даних алгоритмом, таким чином створюючи унікальний зашифрований текст, який можна розшифрувати лише за допомогою ключа.

До сучасності криптографія стосувалася майже виключно «шифрування», яке є процесом перетворення звичайної інформації

(відкритий текст) у незрозумілу форму (зашифрований текст) [31]. Дешифрування відбувається навпаки, іншими словами, перехід від незрозумілого зашифрованого тексту назад до відкритого тексту. У формальних математичних термінах «криптосистема» — це впорядкований список елементів кінцевих можливих відкритих текстів, кінцевих можливих зашифрованих текстів, кінцевих можливих ключів, а також алгоритмів шифрування та дешифрування, які відповідають кожному ключу. Ключі важливі як формально, так і на практиці, оскільки шифри без змінних ключів можуть бути просто зламані, лише знаючи використовуваний шифр, і тому марні для більшості цілей. Історично склалося так, що шифри часто використовувалися безпосередньо для шифрування або дешифрування без додаткових процедур, таких як автентифікація або перевірка цілісності.

Терміни, які найчастіше плутають і неправильно використовують у лексиконі криптології, це код і шифр. Навіть експерти іноді вживають ці терміни як синоніми.

Код — це просто незмінне правило для заміни фрагмента інформації (наприклад, літери, слова або фрази) іншим об’єктом, але не обов’язково такого ж типу. Відомим прикладом є азбука Морзе, яка замінює буквено-цифрові символи візерунками з крапок і тире. Ймовірно, найвідомішим кодом, який використовується сьогодні, є Американський стандартний код для обміну інформацією (ASCII). Використовується в усіх персональних комп’ютерах і терміналах, він представляє 128 символів (і такі операції, як зворотний пробіл і повернення каретки) у формі семирозрядних двійкових чисел, тобто як рядок із семи 1 і 0. У ASCII нижня буква a завжди дорівнює 1100001, велика A завжди 1000001 і так далі. Акроніми також широко відомі і вживані коди, як, наприклад, Y2K (для «2000 року») і COD (що означає «готівкою при доставці»). Іноді таке кодове слово досягає незалежного існування (і значення), тоді як оригінальна еквівалентна фраза

забувається або, принаймні, більше не має точного значення, приписуваного кодовому слову.

Шифри, як і у випадку з кодами, також замінюють частину інформації (елемент відкритого тексту, який може складатися з літери, слова або рядка символів) іншим об’єктом. Різниця полягає в тому, що заміна виконується згідно з правилом, визначеним секретним ключем, відомим лише передавачу та законному одержувачу в очікуванні, що стороння особа, яка не знає ключа, не зможе інвертувати заміну, щоб розшифрувати шифр. У минулому різниця між кодами та шифрами була незначною. Однак у сучасних комунікаціях інформація часто кодується і шифрується, тому важливо розуміти різницю. Лінія супутникового зв’язку, наприклад, може кодувати інформацію символами ASCII, якщо вона є текстовою, або імпульсно-кодовою модуляцією та оцифровувати її у двійковій десятковій формі (BCD), якщо це аналоговий сигнал, наприклад мова. Отримані закодовані дані потім шифруються в шифри за допомогою стандарту шифрування даних або розширеного стандарту шифрування (DES або AES). Нарешті, сам отриманий потік шифрів кодується знову, використовуючи коди з виправленням помилок для передачі від наземної станції до орбітального супутника, а звідти назад до іншої наземної станції. Потім ці операції скасовуються в зворотному порядку призначеним отримувачем для відновлення вихідної інформації [32].

Криптоаналіз — це термін, який використовується для вивчення методів отримання значення зашифрованої інформації без доступу до ключа, який зазвичай необхідний для цього; тобто це вивчення того, як «зламувати» алгоритми шифрування або їх реалізації.

Деякі використовують терміни «криптографія» та «криптологія» як синоніми в англійській мові, тоді як інші використовують «криптографія» для конкретного позначення використання та практики криптографічних методів, а «криптологія» для позначення комбінованого вивчення

криптографії та криптоаналізу.Англійська є більш гнучкою, ніж деякі інші мови, у яких «криптологія» (розроблена криптологами) завжди вживається у другому значенні, наведеному вище.

Вивчення характеристик мов, які мають певне застосування в криптографії чи криптології (наприклад, частотні дані, буквосполучення, універсальні шаблони тощо), називається криптолінгвістикою.

За допомогою шифрування забезпечуються три стани безпеки інформації:

1. Конфіденційність.

Шифрування використовується для приховання інформації від неавторизованих користувачів під час передачі або зберігання.

2. Цілісність.

Шифрування використовується для запобігання зміні інформації під час передачі або зберігання.

3. Ідентифікованість.

Шифрування використовується для аутентифікації джерела інформації та запобігання відмови відправника інформації від того факту, що дані були надіслані саме їм.

Для того, щоб прочитати зашифровану інформацію, стороні, що приймає, необхідні ключ і дешифратор (пристрій, що реалізує алгоритм розшифровування). Ідея шифрування полягає в тому, що зловмисник, перехопивши зашифровані дані і не маючи до них ключа, не може ні прочитати, ні змінити інформацію, що передається. Крім того, у сучасних криптосистемах (з відкритим ключем) для шифрування та розшифрування даних можуть використовуватись різні ключі. Однак з розвитком криптоаналізу з'явилися методики, що дозволяють дешифрувати закритий текст без ключа. Вони ґрунтуються на математичному аналізі переданих даних.

3.2 Типи криптосистем

Існує два основних типи криптосистем: симетричні та асиметричні. У симетричних системах, єдиних відомих до 1970-х років, той самий секретний ключ шифрує та розшифровує повідомлення. Маніпулювання даними в симетричних системах відбувається значно швидше, ніж в асиметричних системах. Асиметричні системи використовують «відкритий ключ» для шифрування повідомлення та відповідний «приватний ключ» для його дешифрування. Перевага асиметричних систем полягає в тому, що відкритий ключ можна вільно публікувати, що дозволяє сторонам встановлювати безпечний зв’язок без спільного секретного ключа. На практиці асиметричні системи використовуються для того, щоб спочатку обмінюватися секретним ключем, а потім захищати зв’язок через більш ефективну симетричну систему, використовуючи цей ключ. Прикладами асиметричних систем є обмін ключами Діффі–Хеллмана, RSA (Рівест–Шамір–Адлеман), ECC (криптографія на еліптичній кривій) і постквантова криптографія. Захищені симетричні алгоритми включають широко використовуваний AES (Advanced Encryption Standard), який замінив старий DES (Data Encryption Standard).

3.2.1 Симетричні криптосистеми

Алгоритми із симетричним ключем — це алгоритми для криптографії, які використовують однакові криптографічні ключі як для шифрування відкритого тексту, так і для розшифровки зашифрованого тексту. Ключі можуть бути ідентичними, або між двома ключами може бути проста трансформація. На практиці ключі являють собою спільний секрет між двома або більше сторонами, який можна використовувати для підтримки приватного інформаційного зв’язку. Вимога, щоб обидві сторони мали доступ до секретного ключа, є одним із головних недоліків шифрування з симетричним ключем порівняно з шифруванням із відкритим ключем (також відомим як шифрування з асиметричним

ключем). Однак алгоритми шифрування із симетричним ключем зазвичай кращі для масового шифрування. Вони мають менший розмір ключа, що означає менше місця для зберігання та швидшу передачу. Через це шифрування з асиметричним ключем часто використовується для обміну секретного ключа на шифрування з симетричним ключем [33].

До винаходу схеми асиметричного шифрування єдиним способом, що існував, було симетричне шифрування. Ключ алгоритму повинен зберігатися в таємниці обома сторонами, повинні здійснюватися заходи щодо захисту доступу до каналу, по всьому шляху прямування криптограми, або сторонами взаємодії за допомогою криптооб'єктів, повідомлень, якщо цей канал взаємодії під грифом «Не для використання третіми особами». Алгоритм шифрування вибирається сторонами на початок обміну повідомленнями.

В даний час симетричні шифри поділяються на наступні типи:

1. Блокові шифри – вони обробляють інформацію блоками певної довжини (зазвичай 64, 128 біт), застосовуючи до блоку ключ у встановленому порядку, як правило, кількома циклами перемішування та підстановки, які називають раундами. Результатом повторення раундів є лавинний ефект — втрата відповідності бітів, що наростає, між блоками відкритих і зашифрованих даних.

2. Потокові шифри, у яких шифрування проводиться над кожним бітом чи байтом вихідного (відкритого) тексту з допомогою гамування. Поточний шифр може бути легко створений на основі блокового, запущеного у спеціальному режимі.

Більшість симетричних шифрів використовують складну комбінацію великої кількості підстановок та перестановок. Багато таких шифрів виконуються в кілька (іноді до 80) проходів, використовуючи кожному проході «ключ проходу». Безліч «ключів проходу» всім проходів називається «розкладом ключів» (key schedule). Як правило, воно

створюється з ключа виконанням над ним деяких операцій, у тому числі перестановок та підстановок.

Типовим способом побудови алгоритмів симетричного шифрування є мережа Фейстеля. Алгоритм будує схему шифрування з урахуванням функції F(D, K), де D — порція даних розміром удвічі менше блоку шифрування, а K — «ключ проходу» даного проходу. Від функції не вимагається оборотність — зворотна функція може бути невідома. Переваги мережі Фейстеля — майже повний збіг дешифрування з шифруванням (єдина відмінність — зворотний порядок ключів проходу в розкладі), що значно полегшує апаратну реалізацію.

Операція перестановки перемішує біти повідомлення за законом. У апаратних реалізаціях вона очевидно реалізується як переплутування провідників. Саме операції перестановки дають можливість досягнення ефекту лавини.

Операції підстановки виконуються як заміна значення певної частини повідомлення (часто 4, 6 чи 8 біт) на стандартне, жорстко вбудоване в алгоритм інше число шляхом звернення до константного масиву. Операція підстановки вносить у алгоритм нелінійність.

Найчастіше стійкість алгоритму, особливо до диференціального криптоаналізу, залежить від вибору значень таблицях підстановки (S-блоках). Як мінімум вважається небажаною наявність нерухомих елементів $S(x) = x$, а також відсутність впливу якогось біта вхідного байта на якийсь біт результату - тобто випадки, коли біт результату однаковий для всіх пар вхідних слів, що відрізняються тільки в цьому біті.

Симетричні шифри історично були сприйнятливі до атак на відомий відкритий текст, диференціального та лінійного криптоаналізу. Ретельна побудова функцій для кожного раунду може значно зменшити шанси на успішну атаку. Також можна збільшити довжину ключа або раундів у процесі шифрування для кращого захисту від атаки. Це, однак, має

тенденцію до збільшення потужності обробки та зниження швидкості, з якою виконується процес, через кількість операцій, які система повинна виконати.

Більшість сучасних алгоритмів із симетричним ключем, здається, стійкі до загрози постквантової криптографії. Квантові комп'ютери експоненціально збільшать швидкість, з якою ці шифри можуть бути декодовані; зокрема, алгоритм Гровера бере квадратний корінь із часу, який традиційно потрібен для атаки грубою силою, хоча ці вразливості можна компенсувати подвоєнням довжини ключа. Наприклад, 128-розрядний шифр AES не буде захищений від такої атаки, оскільки він скоротить час, необхідний для тестування всіх можливих ітерацій, з понад 10 квінтильйонів років до приблизно шести місяців. Навпаки, для декодування 256-бітного шифру AES квантовому комп'ютеру знадобиться стільки ж часу, скільки звичайному комп'ютеру для декодування 128-бітного шифру AES. З цієї причини AES-256 вважається «квантово стійким».

Алгоритми із симетричним ключем вимагають, щоб як відправник, так і одержувач повідомлення мали однаковий секретний ключ. Усі ранні криптографічні системи вимагали від відправника або одержувача якимось чином отримати копію цього секретного ключа через фізично безпечний канал.

Майже всі сучасні криптографічні системи все ще використовують внутрішні алгоритми симетричного ключа для шифрування основної маси повідомлень, але вони усувають потребу у фізично захищеному каналі за допомогою обміну ключами Діффі–Хеллмана або іншого протоколу відкритого ключа для безпечного узгодження свіжий новий секретний ключ для кожного сеансу/розмови (пряма секретність).

3.2.2 Асиметричні криптосистеми

Криптографія з відкритим ключем, або асиметрична криптографія, є областю криптографічних систем, які використовують пари пов’язаних ключів. Кожна пара ключів складається з відкритого ключа та відповідного закритого ключа [34]. Пари ключів генеруються за допомогою криптографічних алгоритмів на основі математичних задач, які називаються односторонніми функціями. Безпека криптографії з відкритим ключем залежить від збереження секретного ключа; відкритий ключ можна відкрито поширювати без шкоди для безпеки.

У системі шифрування з відкритим ключем будь-хто, хто має відкритий ключ, може зашифрувати повідомлення, одержавши зашифрований текст, але лише ті, хто знає відповідний закритий ключ, можуть розшифрувати зашифрований текст, щоб отримати вихідне повідомлення.

Наприклад, журналіст може опублікувати відкритий ключ пари ключів шифрування на веб-сайті, щоб джерела могли надсилати секретні повідомлення до організації новин у зашифрованому тексті. Лише журналіст, якому відомий відповідний приватний ключ, може розшифрувати зашифровані тексти, щоб отримати повідомлення джерел — підслухувач, який читає електронну пошту на шляху до журналіста, не може розшифрувати зашифровані тексти. Однак шифрування з відкритим ключем не приховує метаданих, наприклад, який комп’ютер використовував джерело для надсилання повідомлення, коли вони його надіслали чи скільки воно триває. Шифрування з відкритим ключем також нічого не повідомляє одержувачу про те, хто надіслав повідомлення — воно просто приховує вміст повідомлення в зашифрованому тексті, який можна розшифрувати лише за допомогою закритого ключа.

У системі цифрового підпису відправник може використовувати закритий ключ разом із повідомленням для створення підпису. Будь-хто з

відповідним відкритим ключем може перевірити, чи збігається підпис із повідомленням, але фальсифікатор, який не знає закритого ключа, не може знайти будь-яку пару повідомлення/підпис, яка б пройшла перевірку за допомогою відкритого ключа.

Наприклад, видавець програмного забезпечення може створити пару ключів підпису та включити відкритий ключ у програмне забезпечення, встановлене на комп'ютерах. Пізніше видавець може розповсюдити оновлення програмного забезпечення, підписане за допомогою закритого ключа, і будь-який комп'ютер, який отримує оновлення, може підтвердити його справжність, перевіривши підпис за допомогою відкритого ключа. Поки видавець програмного забезпечення зберігає закритий ключ у таємниці, навіть якщо фальсифікатор може поширювати шкідливі оновлення на комп'ютери, він не зможе переконати комп'ютери, що будь-які шкідливі оновлення є справжніми.

Алгоритми відкритого ключа є фундаментальними примітивами безпеки в сучасних криптосистемах, включаючи додатки та протоколи, які пропонують гарантії конфіденційності, автентичності та незаперечності електронних комунікацій і зберігання даних. Вони лежать в основі багатьох Інтернет-стандартів, таких як Transport Layer Security (TLS), SSH, S/MIME та PGP. Деякі алгоритми відкритих ключів забезпечують розподіл ключів і секретність (наприклад, обмін ключами Діффі–Хеллмана), деякі забезпечують цифрові підписи (наприклад, алгоритм цифрового підпису), а деякі забезпечують і те, і інше (наприклад, RSA). Порівняно з симетричним шифруванням, асиметричне шифрування повільніше, ніж симетричне шифрування, воно може бути занадто повільне для багатьох цілей [35]. Сучасні криптосистеми (такі як TLS, Secure Shell) використовують як симетричне шифрування, так і асиметричне шифрування, часто за допомогою асиметричного шифрування для

безпечного обміну секретним ключем, який потім використовується для симетричного шифрування.

До середини 1970-х років усі системи шифрування використовували алгоритми з симетричним ключем, у яких один і той самий криптографічний ключ використовується разом із базовим алгоритмом як відправником, так і одержувачем, які мають зберігати його в таємниці. Необхідно, щоб ключ у кожній такій системі обмінювався між сторонами, що спілкуються, певним захищеним способом перед будь-яким використанням системи – наприклад, через захищений канал. Ця вимога ніколи не є тривіальною і дуже швидко стає некерованою, коли кількість учасників збільшується, або коли захищені канали недоступні, або коли ключі часто змінюються (це є розумна криптографічна практика). Зокрема, якщо повідомлення призначені для захисту від інших користувачів, для кожної можливої пари користувачів потрібен окремий ключ.

Два найвідоміших використання криптографії з відкритим ключем:

1. Шифрування з відкритим ключем, при якому повідомлення шифрується відкритим ключем одержувача. Для правильно обраних і використовуваних алгоритмів повідомлення не можуть бути розшифровані на практиці будь-ким, хто не володіє відповідним закритим ключем, який, таким чином, вважається власником цього ключа, а отже, особою, пов’язаною з відкритим ключем. Це можна використовувати для забезпечення конфіденційності повідомлення.[36]

2. Цифрові підписи, у яких повідомлення підписується закритим ключем відправника та може бути перевірене будь-ким, хто має доступ до відкритого ключа відправника. Ця перевірка доводить, що відправник мав доступ до закритого ключа, а отже, ймовірно, це особа, пов’язана з відкритим ключем. Це також доводить, що підпис було підготовлено саме для цього повідомлення, оскільки перевірка буде невдалою для будь-якого

іншого повідомлення, яке можна створити без використання закритого ключа.

Одним із важливих питань є впевненість/доказ того, що конкретний відкритий ключ є автентичним, тобто що він правильний і належить особі чи організації, і не був підроблений чи замінений якоюсь (можливо, зловмисною) третьою стороною. Є кілька можливих підходів, зокрема:

1. Інфраструктура відкритих ключів (PKI), у якій одна або кілька третіх сторін, відомих як органи сертифікації, засвідчують право власності на пари ключів. Це означає, що система PKI (програмне забезпечення, апаратне забезпечення та керування) є надійною для всіх учасників.

2. «Мережа довіри», яка децентралізує автентифікацію за допомогою індивідуальних підтверджень посилань між користувачем і відкритим ключем, що належить цьому користувачу. PGP використовує цей підхід на додаток до пошуку в системі доменних імен (DNS). Система DKIM для цифрового підпису електронних листів також використовує цей підхід.

Асиметричне шифрування з відкритим ключем базується на таких принципах:

1. Можна згенерувати кілька дуже великих чисел (відкритий ключ та закритий ключ) так, щоб, знаючи відкритий ключ, не можна було обчислити закритий ключ за розумний термін. У цьому механізм генерації є загальновідомим.

2. Є надійні методи шифрування, що дозволяють зашифрувати повідомлення відкритим ключем так, щоб розшифрувати його можна було лише закритим ключем. Механізм шифрування загальновідомий.

3. Власник двох ключів нікому не повідомляє закритого ключа, але передає відкритий ключ контрагентам або робить його загальновідомим.

Якщо потрібно передати зашифроване повідомлення власнику ключів, відправник повинен отримати відкритий ключ. Відправник шифрує своє повідомлення відкритим ключем одержувача та передає його одержувачу (власнику ключів) відкритими каналами. При цьому розшифрувати повідомлення не може ніхто, окрім власника закритого ключа.

В результаті можна забезпечити надійне шифрування повідомлень, зберігаючи ключ розшифровки секретним для всіх навіть для відправників повідомлень.

Цей принцип можна пояснити через побутову аналогію "замок - ключ від замку" для відправки посилки. У учасника А є особистий замок та ключ від нього. Якщо учасник А хоче отримати секретну посилку від учасника Б, він публічно передає йому свій замок. Учасник Б засуває замок на секретній посилці та відправляє її учаснику А. Отримавши посилку, учасник А відкриває ключем замок і отримує посилку.

Знання про передачу замку та перехоплення посилки нічого не дадуть потенційному зловмиснику: ключ від замку є тільки в учасника А, тому посилка не може бути розкрита.

Оскільки алгоритми з асиметричним ключем майже завжди потребують набагато більше обчислень, ніж симетричні, зазвичай використовують загальнодоступний/приватний асиметричний алгоритм обміну ключами для шифрування та обміну симетричним ключем, який потім використовується криптографією з симетричним ключем для передачі даних за допомогою спільного симетричного ключа для алгоритму шифрування з симетричним ключем. PGP, SSH і сімейство схем SSL/TLS використовують цю процедуру, тому їх називають гібридними криптосистемами. Початковий обмін ключами на основі асиметричної криптографії для передачі згенерованого сервером симетричного ключа від сервера до клієнта має перевагу, оскільки не вимагає попереднього

спільного використання симетричного ключа вручну, наприклад, на друкованому папері чи дисках, що транспортуються кур’єром, а також забезпечує вищу пропускну здатностість криптографії з симетричним ключем порівняно з криптографією з асиметричним ключем для решти спільного з’єднання.

3.3 Сучасні алгоритми шифрування

3.3.1 Алгоритм RSA

Алгоритм шифрування з відкритим ключем RSA названий на честь трьох винахідників – Рона Рівеста (Ron Rivest), Аді Шаміра (Adi Shamir) та Леонарда Едлмана (Leonard Adleman). Вони розробили алгоритм шифрування RSA у 1977 році та заснували RSA Data Security у 1982 році. З того часу компанія пройшла кілька рук, у тому числі EMC і тепер належить Dell Technologies.

Власники бізнесу вимагають гнучкості та масштабованості у своїх зусиллях щодо зміцнення довіри та захисту інтернет-сайтів та транзакцій від хакерів. Хакери постійно розробляють складніші методи для порушення безпеки і заподіяння шкоди бізнесу або його клієнтам. Відповідальні власники бізнесу давно знають, що потрібно захищати свою присутність в Інтернеті за допомогою SSL сертифікатів, наданих довіреною сторонньою сертифікацією. Використання SSL-сертифіката дозволяє автентифікувати веб-сервер та передавати конфіденційну інформацію. SSL сертифікати традиційно покладалися на шифрування з використанням відкритих і закритих ключів на основі алгоритму RSA. Поки ці ключі залишаються безпечними, зростаючими загрозами з боку все більш потужних комп'ютерів спонукали Національний інститут стандартів і технологій (NIST), серед іншого, закликати додаткове посилення онлайн-шифрування. Наразі компанії мають можливість вибирати між сертифікатами, які забезпечують захист на основі алгоритму RSA, за двома альтернативними алгоритмами, ECC та DSA або для створення

сертифікатів для всіх трьох, встановлених на сервері. Така гнучкість дозволяє власникам бізнесу надавати ширший спектр варіантів шифрування для різних обставин, інфраструктури та груп клієнтів чи партнерів.

Transport Layer Security (TLS) та його попередники протоколів Secure Socket Layer (SSL) залишаються галузевим стандартом для перевірки автентичності веб-сайту та захисту інформації. Використовуваний веб-сайтами та браузерами, TLS дозволяє здійснювати автентифікацію, стиснення та шифрування даних між клієнтом (кінцевим користувачем) та сервером, гарантуючи, що хакери не можуть отримати доступ до даних під час їх надсилання.

Користувачі знають, що вони звертаються до веб-сайту або сторінки, захищеної TLS, коли "http" в адресному рядку замінюється "https", і в рядку стану з'являється невеликий замок. Такі захищені сторінки вимагають використання SSL-сертифікації для увімкнення шифрування інформації, яка передається на або із захищеного веб-сервера. Переважна більшість сертифікатів SSL сьогодні покладаються на ключі, згенеровані та підписані за допомогою алгоритму RSA.

Алгоритм RSA залишається ефективним варіантом шифрування. Тим не менш, довжина ключів продовжуватиме зростати експоненційно. Інтернет-спільноти відзначили здатність хакерів використовувати потужні комп'ютери для потенційного злому ключів, що наближаються до 1024 біт. Тому NIST рекомендував, щоб до кінця 2013 року сертифікаційні центри не видавали жодних нових сертифікатів SSL/TLS з розмірами відкритого ключа RSA розміром менше 2048 біт. У той же час альтернативні алгоритми шифрування та підписання були прийняті федеральним урядом, який випустив керівні принципи, що базуються на криптографії з еліптичною кривою (ECC) та алгоритмах цифрового підпису (DSA). Сертифікати, підписані з алгоритмом RSA, широко використовуються

протягом багатьох років, але гнучкість алгоритмів, заснована на рекомендаціях NIST, дозволяє підприємствам вибирати сертифікати, підписані трьома різними алгоритмами: RSA, DSA та ECC. Конструкція TLS дозволяє різним алгоритмам працювати або самостійно, або пліч-о-пліч, тому з гнучкістю алгоритму власники бізнесу можуть вибирати алгоритм відкритого ключа або комбінацію алгоритмів, які найкраще підходять для їхньої присутності в Інтернеті та інфраструктури [37].

Алгоритм шифрування являє собою математичну процедуру або набір кроків для кодування даних. RSA є найбільш широко використовуваним алгоритмом шифрування сьогодні. ECC - це новий алгоритм шифрування з меншими розмірами для сучасних та мобільних додатків. DSA зазвичай використовується державними підрядниками та субпідрядниками.

RSA відноситься до так званих асиметричних алгоритмів, у яких ключ шифрування не збігається з ключем дешифрування. Один із ключів доступний усім (так робиться спеціально) і називається відкритим ключем, інший зберігається тільки у його господаря і невідомий нікому іншому. За допомогою одного ключа можна виконувати операції лише в один бік. Якщо повідомлення зашифровано за допомогою одного ключа, розшифрувати його можна тільки за допомогою іншого. Маючи один із ключів неможливо (дуже складно) знайти інший ключ, якщо розрядність ключа висока.

В основі RSA лежить завдання факторизації добутку двох простих великих чисел. Для шифрування використовується проста операція зведення в ступінь за модулем N. Для розшифрування необхідно обчислити функцію Ейлера від числа N, для цього необхідно знати розкладання числа n на прості множники (у цьому і полягає задача факторизації). У RSA відкритий і закритий ключ складається з кількох чисел. Закритий ключ зберігається в секреті, а відкритий ключ

повідомляється іншому учаснику або десь публікується. Генерація проводиться за такою схемою:

1. Вибираються два простих числа p і q (p не дорівнює q).
2. Обчислюється модуль N = p * q.
3. Обчислюється значення функції Ейлера від модуля N: $F(N)=(p-1)(q-1)$.
4. Вибирається число e, зване відкритою експонентою, число e повинне лежати в інтервалі 1
5. Обчислюється число d, так звана секретна експонента, яка обчислюється за формулою $d*e=1(\mathrm{mod}F(N))$, тобто є мультиплікативно зворотньою до e за модулем F(N).

Підсумовуючи отримуємо наступне:

Пара (e, N) – відкритий ключ.

Пара (d, N) – закритий ключ.

Багато протоколів, зокрема Secure Shell (SSH), OpenPGP, S/MIME та SSL/TLS, покладаються на RSA для шифрування та функцій цифрового підпису. Він також використовується в програмному забезпеченні – очевидним прикладом є браузери, оскільки їм потрібно встановити безпечне з’єднання через незахищену мережу, як-от Інтернет, або перевірити цифровий підпис. Перевірка підпису RSA є однією з операцій, які найчастіше виконуються в мережевих системах [38].

Безпека RSA залежить від обчислювальної складності розкладання великих цілих чисел. Зі збільшенням обчислювальної потужності та відкриттям ефективніших алгоритмів розкладання на множники зростає й здатність розкладати на множники все більші й більші числа.

Міцність шифрування безпосередньо пов’язана з розміром ключа. Подвоєння довжини ключа може забезпечити експоненціальне збільшення міцності, хоча це погіршує продуктивність. Ключі RSA зазвичай мають довжину 1024 або 2048 біт, але експерти вважають, що 1024-бітні ключі

більше не є повністю захищеними від усіх атак. Ось чому уряд і деякі галузі переходять на мінімальну довжину ключа 2048 біт.

За винятком непередбаченого прориву в квантових обчисленнях, мине багато років, перш ніж знадобляться довші ключі, але криптографія на основі еліптичної кривої (ECC) набуває прихильності багатьох експертів з безпеки як альтернатива RSA для реалізації криптографії з відкритим ключем. Він може створювати швидші, менші та ефективніші криптографічні ключі.

Сучасне апаратне та програмне забезпечення готове до ECC, і його популярність, ймовірно, зростатиме. Він може забезпечити еквівалентну безпеку з меншою обчислювальною потужністю та використанням ресурсу акумулятора, що робить його більш придатним для мобільних додатків, ніж RSA.

3.3.2 Алгоритм AES

Advanced Encryption Standard (AES), також відомий як Rijndael (Рейндал) – симетричний алгоритм блокового шифрування, прийнятий як стандарт шифрування урядом США за результатами конкурсу AES у 1997 році. Специфікація цього шифру була опублікована 26 листопада 2001, а 26 травня 2002 він був оголошений стандартом шифрування. За статистикою на 2009 рік він був одним із найпоширеніших алгоритмів симетричного шифрування [39].

Для шифрування в алгоритмі AES застосовуються такі процедури перетворення даних:

1. ExpandKey — Обчислення ключів для всіх раундів.
2. SubBytes — процедура заміни байтів відповідно до таблиці заміни S-box. При цьому, значення байта заміни визначається за допомогою байта, що замінюється наступним чином: перші чотири біти замінної комірки визначають номер рядка таблиці заміни S-box, а останні

чотири - номер стовпця. У разі розшифрування використовується зворотна таблиця заміни (Inverted S-box) відповідно.

3. ShiftRows - процедура циклічного зсуву в рядах блоку стану (State). Для першого ряду зсув не виконується, для другого зсув становить 1 осередок, для третього - 2 осередки, для четвертого - 3 осередки. При цьому, при шифруванні зсув виконується вліво, для розшифрування - праворуч.

4. MixColumns - Змішування даних усередині кожного стовпця форми;

5. AddRoundKey — процедура логічного додавання по модулю 2 стовпця блоку стану (State) зі стовпцем блоку поточного раундового ключа. Ця процедура виконується однаково при шифруванні, так і при розшифруванні (що очевидно випливає з механізму роботи операції XOR).

6. RotWord — процедура циклічного зсуву на одну комірку вгору першого стовпця поточного блоку раундового ключа. Оскільки як для шифрування, так і для розшифрування використовується той самий ключ, то генерація (у специфікації, розширення) ключа виконується єдиним способом (тобто дана процедура теж не має зворотної собі).

Порядок виконання процедур 2 та 3 можна поміняти місцями через випередження цих операцій.

Процедури 4 і 5 також можна виконувати в різному порядку, але при цьому змінюється кількість їх викликів, оскільки MixColumns(AddRoundKey(A, B)) = AddRoundKey(MixColumns(A), MixColumns(B)).

Шифрування здійснюється за алгоритмом, наведеним на Рис.3.1.

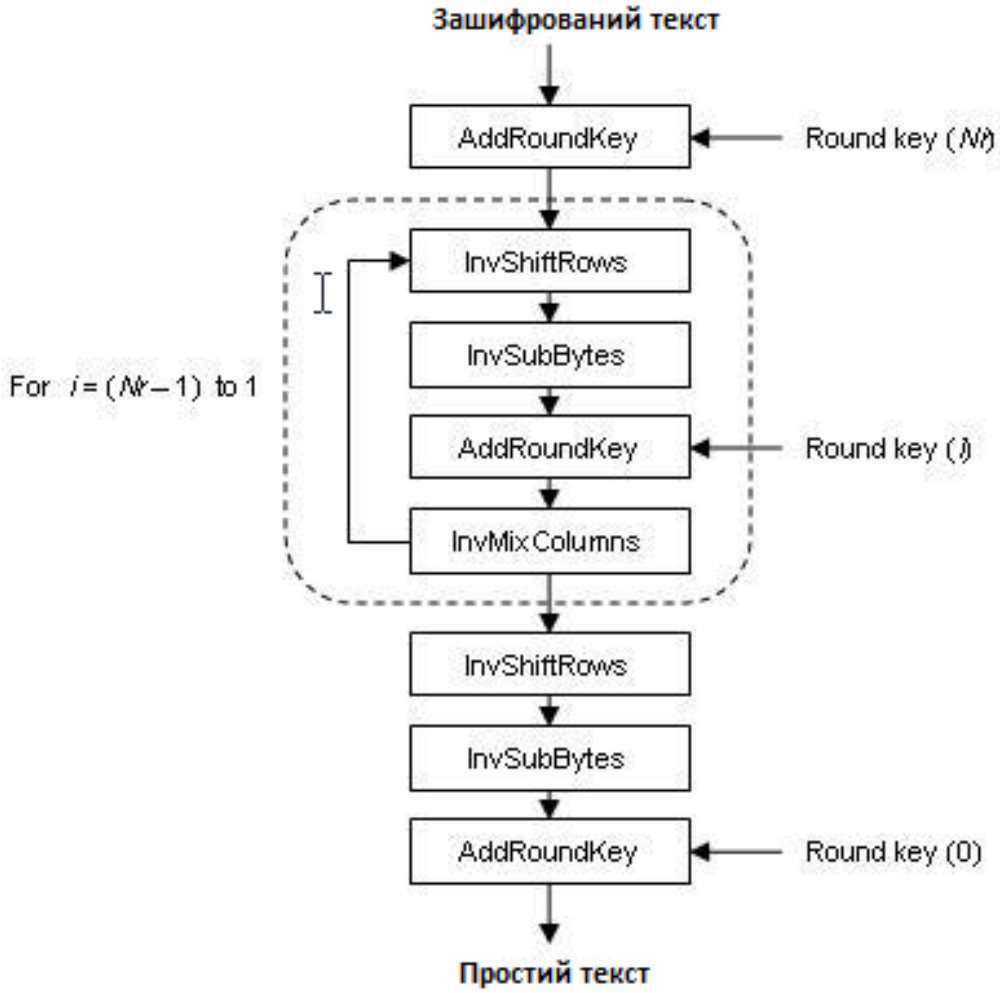


Рис.3.1 Алгоритм шифрування AES

Процедура шифрування є послідовним виконанням описаних вище допоміжних процедур над матрицею даних таку кількість раундів, яка визначена розміром ключа. Нижче розглянемо повний цикл шифрування для 128-бітного ключа.

Як можна помітити на схемі, спочатку відбувається фаза ініціалізації, під час якої ми використовуємо процедуру AddRoundKey() на блок стану (State), як раундовий ключ тут використовується 128-бітний ключ, заданий користувачем. Далі блок стану проходить цикл з 9 раундів, де послідовно над ним виконуються процедури SubBytes(), ShiftRows(), MixColumns() і AddRoundKey() (з використанням ключа, згенерованого на основі ключа попереднього раунду). У фінальній стадії ми виконуємо

неповний раунд, який складається з процедур SubBytes(), ShiftRows() та AddRoundKey(), і на виході отримуємо зашифроване повідомлення.

Процедура розшифрування в даному випадку виглядає так само, тільки всі дії виконуються в зворотному порядку.

У сучасній криптографії AES широко поширений і підтримується як апаратним, так і програмним забезпеченням. До теперішнього часу не було виявлено жодних практичних криптоаналітичних атак проти AES. Крім того, AES має вбудовану гнучкість довжини ключа, що забезпечує певний ступінь «захищеності від майбутнього» від прогресу в можливості виконувати вичерпний пошук ключів.

3.3.3 Алгоритм Triple DES

Triple DES (3DES), офіційна назва Triple Data Encryption Algorithm (TDEA або Triple DEA) – симетричний блоковий шифр, створений Уітфілдом Діффі, Мартіном Хеллманом та Уолтом Тачманном у 1978 році на основі алгоритму DES. Швидкість роботи 3DES в 3 рази нижче, ніж у DES, але криптостійкість набагато вища - час, необхідний для криптоаналізу 3DES, може бути в мільярд разів більше, ніж час, необхідний для розкриття DES. Замість офіційної назви постачальники, користувачі та розробники криптосистем використовують термін "3DES".

Попередни шифру 3DES – DES – був створений в далекому 1975 році. Проте, разом із розвитком технологій захисту інформації, розвивалися й технології злому. Саме швидкість підбору ключів, зашифрованих за допомогою DES стала основною проблемою цього методу шифрування. Але винахід нового алгоритму шифрування вимагає багато грошей і часу не тільки на саму розробку, але і на те, щоб змінити існуючу структуру безпеки багатьох світових компаній, які вже використовували DES. Незабаром прагматичне рішення було знайдено. Пропонувалося змінити не сам шифр, а спосіб його застосування. Саме тоді виник 3DES.

Існує 2 основних варіанти шифрування алгоритмом 3DES: 3-key Triple DES та 2-key Triple DES. Як видно з назви, важлива відмінність цих методів - кількість ключів (три і два відповідно). У свою чергу, кожен з цих алгоритмів має по 2 різні типи: EEE (encryption-encryption-encryption) та EDE (encryption-decryption-encryption) шифрування:

1. DES-EEE3: Вихідний текст тричі шифрується, використовуючи різні ключі.
2. DES-EDE3: Початковий текст шифрується, потім дешифрується (вже іншим ключем), а потім знову шифрується (третім ключем). Наочне уявлення можна побачити нижче.
3. DES-EEE2: Вихідний текст тричі шифрується, однак ключі на першому та останньому кроці однакові.
4. DES-EDE2: Вихідний текст шифрується, потім дешифрується (іншим ключем), потім знову шифрується (ключом, який використовується при першому шифруванні).

На практиці, використовується найбільш тип 3DES шифрування - DES-EDE3.

При шифруванні ключі можна вибрати кількома способами. Нижче, в порядку зниження криптостійкості, представлені такі методи:

1. Усі ключі незалежні, тобто. різні.
2. Ключі, які використовуються при першому та останньому шифруванні однакові, другий ключ відрізняється від них.
3. Усі ключі однакові.

Перший варіант є стійким, так як 3 рази повторює шифрування різними ключами. Якщо при шифруванні DES довжина ключа 56 біт (у кожному байті використовується 7 біт замість 8), то при шифруванні 3DES з різними ключами, довжина підсумкового ключа збільшується втричі - 168 біт.

Другий варіант менш надійний – довжина його ключа лише 112 біт. Однак, цей варіант шифрування більш надійний, ніж звичайне подвійне шифрування за допомогою DES: він захищає від атак "зустріч посередині", адже замість двох послідовних шифрувань також має у своєму складі дешифрування, яке ускладнює складання таблиць і знаходження однакових значень у цих таблицях (основні дії за такого типу атаки).

Третій варіант має таку ж криптостійкість, як і сам DES – довжина ключа 56 біт.

Кожен ключ DES зберігається і передається як 8 байтів, кожен байт - з непарним паритетом, так що повний набір ключів займе 24 байти у першому варіанті, 16 у другому та 8 у третьому.

Шифрування кількох блоків зазвичай відбувається з використанням одного з режимів шифрування, які не залежать від алгоритму шифрування кожного конкретного блоку. Проте, деякі нормативні документи накладають обмеження використання деяких режимів шифрування. Наприклад, згідно з ANS X9.52, при використанні режиму зчеплення блоків вектор ініціалізації повинен змінюватися щоразу. Інші нормативні документи не накладають жодних обмежень, розглядаючи шифрування 3DES як шифрування одного блоку. Якщо порівнювати криптостійкість режимів шифрування, режим зчеплення блоків є найбезпечнішим за рахунок використання додаткового вектора ініціалізації. Однак, режим електронної кодової книги (Triple ECB), використовується частіше, завдяки своїй швидкості та можливості розпаралелювання.

3DES з трьома ключами реалізований у багатьох додатках, орієнтованих працювати з Інтернет, зокрема PGP і S/mime. Потрійний DES є досить популярною альтернативою DES і використовується при керуванні ключами у стандартах ANSI X9.17 та ISO 8732 та в PEM (Privacy Enhanced Mail). Індустрія електронних платежів використовує 3DES і

продовжує активно розробляти та публікувати стандарти, засновані на ньому (наприклад, EMV).

Незважаючи на те, що фахівці стверджують, що алгоритм 3DES-EDE з трьома різними ключами залишиться надійним до 2030 року, 3DES стає менш популярним: на зміну йому приходить новий алгоритм AES Rijndael. Rijndael, реалізований програмно, працює у шість разів швидше. Тому 3DES найбільше підходить для апаратних реалізацій. Багато систем безпеки продовжують підтримувати як 3DES, так і AES. Хоча 3DES може підтримуватись для зворотної сумісності, він більше не рекомендований для використання [40].

Висновок

В даному розділі було розглянуто криптографічні методи захисту та шифрування. А саме поняття криптографічного шифрування, типи криптосистем та сучасні алгоритми шифрування. Шифрування даних — це метод захисту даних шляхом їх кодування таким чином, що їх може розшифрувати або отримати доступ до них тільки особа, яка має правильний ключ шифрування.

Існує два основних типи криптосистем: симетричні та асиметричні. У симетричних системах, той самий секретний ключ шифрує та розшифровує повідомлення. Маніпулювання даними в симетричних системах відбувається значно швидше, ніж в асиметричних системах. Асиметричні системи використовують «відкритий ключ» для шифрування повідомлення та відповідний «приватний ключ» для його дешифрування. Перевага асиметричних систем полягає в тому, що відкритий ключ можна вільно публікувати, що дозволяє сторонам встановлювати безпечний зв’язок без спільного секретного ключа.

РОЗДІЛ 4
КОМПЛЕКСНА СИСТЕМА ШИФРУВАННЯ, РЕАЛІЗОВАНА ЗА ДОПОМОГОЮ AES

4.1 Аналіз останніх досліджень

У криптографії існує багато типів алгоритмів шифрування. Загалом їх можна розділити на три типи: симетричний алгоритм шифрування, асиметричний алгоритм шифрування та алгоритм шифрування з одним записом.

Використовуються різні алгоритми шифрування щодо безпеки, ефективності шифрування, складності реалізації, є великі відмінності в найкращих місцях для використання. Серед них симетричний алгоритм шифрування використовує той самий ключ для шифрування та дешифрування. Загальні алгоритми включають IDEA, DESX, RC4, RC5, RC6, DES, 3DES і AES.

Алгоритми симетричного шифрування часто використовуються в ситуаціях, коли шифруються великі обсяги даних або дані часто надсилаються. Асиметричний алгоритм шифрування означає, що для шифрування та дешифрування використовуються різні відкритий і закритий ключі.

Іноді асиметричне шифрування також називають шифруванням з відкритим ключем. Алгоритм також оборотний. Загальні алгоритми включають RSA, DSA, ECC, Diffie-Hellman та El. Gamal та ін. Алгоритми асиметричного шифрування зазвичай використовуються для шифрування відкритим ключем, дешифрування закритого ключа, підпису закритого ключа, перевірки відкритого ключа, шифрування невеликої кількості конфіденційної інформації, цифрового підпису тощо.

Алгоритм шифрування з одним входом і алгоритм хеш-шифрування є односпрямованими незворотними алгоритмами. Зашифровані дані неможливо розшифрувати. Загальні алгоритми включають MD5 і SHA.

Алгоритм зазвичай використовується для зберігання невідновлюваних паролів, перевірок цілісності інформації тощо, таких як перевірка файлів, цифровий підпис, протокол автентифікації тощо. Як зашифрувати критично важливу інформацію стало центром уваги в галузі IT та створення мереж [41]. Тим не менш, криптоаналіз AES не припинився, і кілька дослідників досліджують нові методи, які дозволять нам досягти конкурентоспроможності.

Робота, запропонована Ріною Мехла та Харлін Каур [42], зосереджена на модифікації розширення ключа та перетворення рядка зсуву AES, щоб зробити відповідний алгоритм високостійким до атак. Запропонований ними метод також скоротив час, що витрачається на шифрування зображень, і забезпечує кращий вихід, ніж AES.

У 2016 році Смалюкас і Гітіс Вайцекаускас [43] переглянули та вдосконалили алгоритм AES, щоб зменшити обчислення алгоритму та покращити передачу даних. Запропонована ними техніка використовує генератор бітів парності для представлення високого рівня захисту та створення високої передачі даних без використання Mixcolumns.

Дімас Натанаель, Фейсал, Деві Суріані [44] запропонували реалізацію алгоритму ECC для захисту текстових повідомлень у мобільних повідомленнях. Вони дотримуються підходу, запропонованого Сінгхом, для створення програми чату Android із наскрізним шифруванням на пристрої. Вони також пропонують свої програми для чату показуючи експериментальні результати, такі як точність отриманого текстового повідомлення, середній час шифрування та час дешифрування [45]. Вони включають інформацію про випадкові перешкоди для покращення захисту даних.

Ченг Тан, Ксіоян Денг, Люїн Цанг [46] знайшли 5 широко використовуваних блокових шифрів, AES DES, 3DES, RC5 та

ідентифікацію Blowfish. Після перевірки файлів Cipherteext рівень ідентифікації перевищує 97%.

4.2 Модифікація алгоритму AES

В даній модифікації алгоритму Розширеного стандарту шифрування (AES) не було змінено його основу. Тим не менш, криптоаналіз AES не зупинився, і багато дослідників шукають нові підходи, які дозволять нам досягти конкурентоспроможності. Забезпечити високу ефективність і гнучкий алгоритм для різних бізнес-потреб (електронна комерція, електронна пошта, банківська картка і т.д.).

У даному дослідженні було внесено наступні дві модифікації оригінального алгоритму AES:

1. Додавання нового ключа: функції виконуються, як показано на рис.4.1, у запропонованому алгоритмі AES, і вводиться додатковий ключ (модифікований ключ). Спочатку генерується ключ, а потім шифрується відкритим ключем.

Доки не виконається етап розширення ключа процесу шифрування, додатковий ключ спочатку оброблятиметься операцією XOR із звичайним текстом. XOR викликає цей процес InitialAddRoundKey. Новий вихідний результат операції InitialAddRoundKey використовується як відкритий текст для наступних кроків. Перед цим звичайний ключ витрачається на створення підключів;

Операція XOR або Виняткова диз'юнкція — це логічна та бітова операція, що набуває значення «істина» тоді й лише тоді, коли значення «істина» має суто один з її операндів. Виняткова диз'юнкція є запереченням логічної еквівалентності. У випадку двох змінних результат виконання операції є істинним тоді й тільки тоді, якщо лише один з аргументів є істинним. Для функції трьох і більше змінних результат виконання операції буде істинним тільки тоді, коли аргументів, рівних 1, на заданому наборі буде непарна кількість. Така операція природним

чином виникає в кільці лишків за модулем 2, звідки й походить назва операції.

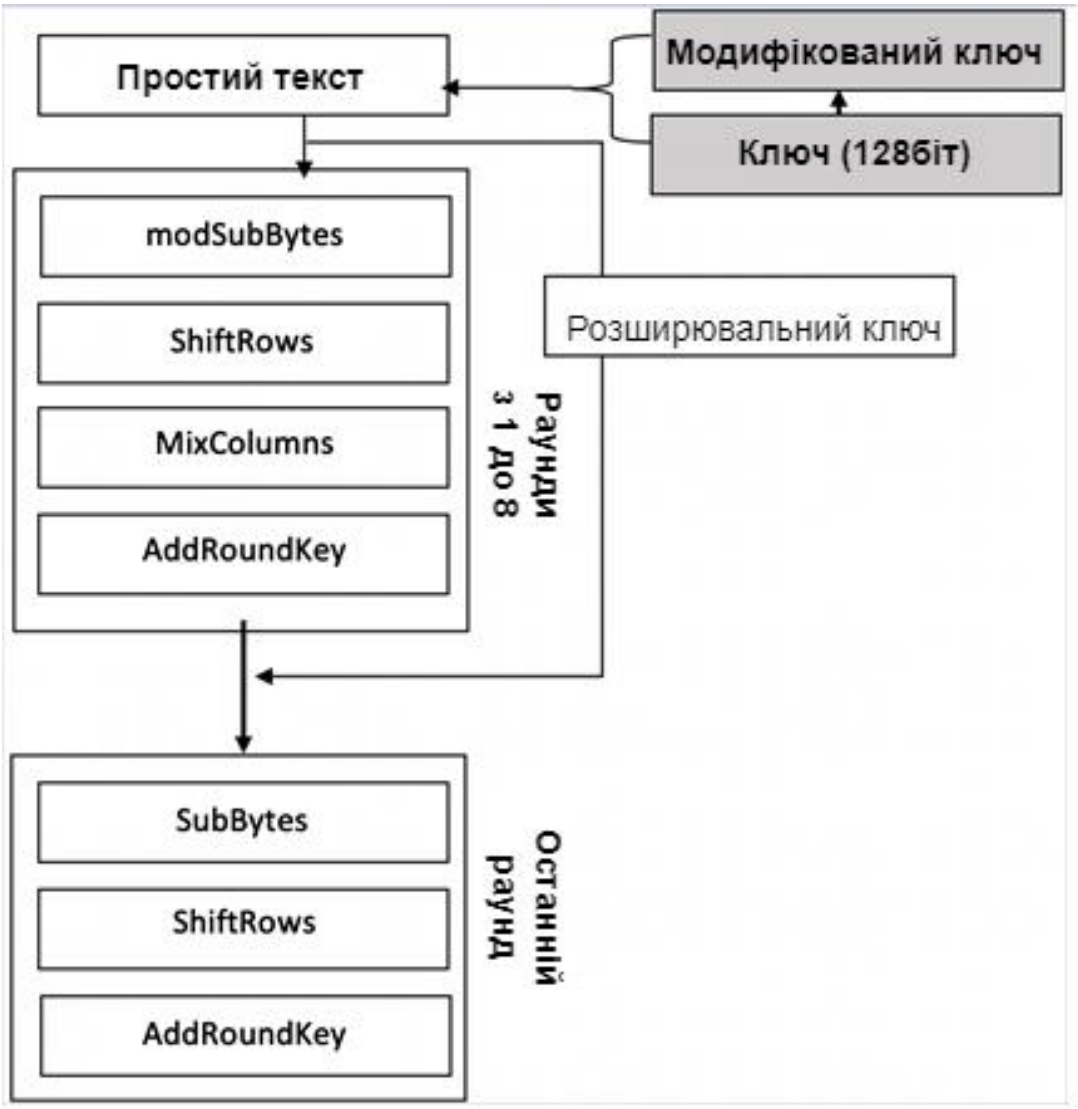


Рис.4.1 Блок-схема шифрування AES

Додавання за модулем 2 слід відрізняти від простого додавання булевих операндів, яке відповідає звичайному логічному «або», тобто логічній диз'юнкції.

Відповідною операцією в теорії множин є симетрична різниця множин істинності операндів.

2. Реконфігурація у функції SubBytes: замість існуючої операції SubBytes ми вставили нову операцію в початкову операцію SubBytes під назвою Modified Transport. І ми оновлюємо SubBytes до цієї функції як

ModSubBytes. Далі дані в процесі ModSubBytes передаються до того, як значення S-Box були замінені. Масив стану ділиться на дві половини (по 4 біти кожна) кожна частина масиву стану (значення 8 біт) і передається або обмінюється для досягнення нового значення стану в процесі транспортування.

4.3 Методика проектування комплексної системи шифрування

Для забезпечення конфіденційності інформації використовуються як методи симетричного, так і асиметричного шифрування [47] [48]. У симетричному шифруванні використовується єдиний ключ, і для безпечного спілкування всі особи, які отримають повідомлення, повинні мати цей секретний ключ.

Асиметричне шифрування використовує пару ключів: закритий і відкритий ключ. Технологія асиметричного шифрування може забезпечити безпеку та неспростовність за допомогою цифрового підпису. Управління ключами AES набагато складніше, ніж RSA. Якщо нам потрібно зашифрувати великі обсяги даних, AES є хорошим вибором, який може ефективно покращити швидкість шифрування та дешифрування.

У сучасній криптографії AES широко поширений і підтримується як апаратним, так і програмним забезпеченням. До теперішнього часу не було виявлено жодних практичних криптоаналітичних атак проти AES. Крім того, AES має вбудовану гнучкість довжини ключа, що забезпечує певний ступінь «захищеності від майбутнього» від прогресу в можливості виконувати вичерпний пошук ключів.

Однак, безпека AES гарантується лише в тому випадку, якщо вона правильно реалізована та використовується добре керування ключами.

Електронні підписи вимагають використання асиметричних алгоритмів. Алгоритми шифрування AES не можуть реалізувати підписи. RSA можна використовувати для реалізації електронних підписів. Ці два

типи алгоритмів шифрування мають свої переваги та недоліки [49]. Щоб повною мірою розкрити переваги кожного алгоритму, алгоритми шифрування AES і RSA можна комплексно використовувати в процесі використання.

Оскільки алгоритм RSA має високий рівень безпеки, але повільну швидкість шифрування, його можна використовувати для шифрування. Ключ AES надсилає зашифрований ключ AES іншій стороні. Після отримання ключа AES інша сторона розшифровує ключ AES за допомогою RSA, а потім розшифровує отримані дані за допомогою ключа AES. Таким чином, оскільки довжина ключових даних AES, як правило, мала, ефективність шифрування та дешифрування є високою, і велика кількість шифрованих даних використовує алгоритм шифрування AES, а швидкість шифрування та дешифрування самого AES відносно висока, і апаратне забезпечення може ще більше покращити шифрування. Швидкість дешифрування [50] [51], інтегрований процес шифрування (як показано на Рис.4.2)

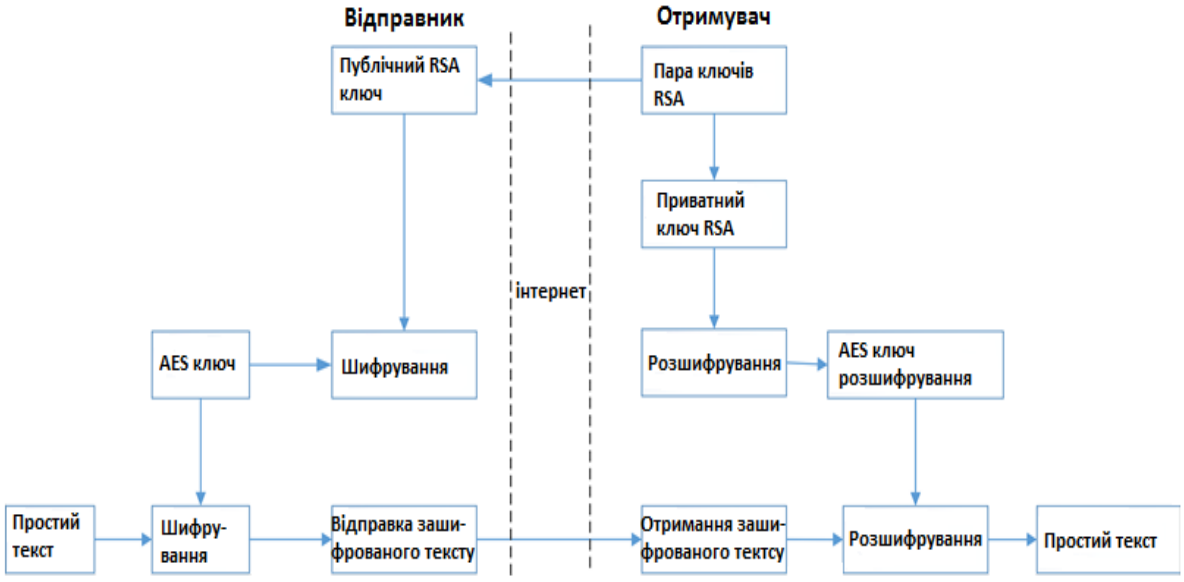


Рис.4.2 Інтегрований процес застосування шифрування

Процес реалізації алгоритму шифрування виглядає наступним чином:

1. Одержувач створює відкритий ключ RSA та закритий ключ (пару ключів), отримувач зберігає закритий ключ і надсилає відкритий ключ RSA відправнику даних через Інтернет;

2. Відправник даних створює розширений ключ AES, шифрує ключ AES відкритим ключем RSA, надісланим одержувачем, і шифрує дані відкритого тексту для надсилання створеним ключем AES;

3. Після отримання зашифрованого тексту та зашифрованого ключа AES приймач розшифровує ключ AES за допомогою закритого ключа RSA, збереженого отримувачем, а потім розшифровує отримані дані зашифрованого тексту за допомогою ключа, щоб отримати дані відкритого тексту.

У фактичному процесі застосування, якщо сторони обміну даними часто надсилають один одному великий обсяг даних, комплексну схему роботи шифрування можна додатково оптимізувати, тобто:

1. Обмін ключами AES виконується RSA через регулярні проміжки часу;

2. Після обміну ключем AES дві сторони надсилають дані за допомогою ключа AES ключа іншої сторони.

Специфічний процес полягає в тому, що і відправник, і одержувач використовують RSA для створення пари паролів і надсилають згенерований відкритий ключ іншій стороні. Кожна сторона генерує ключ AES, шифрує AES, створений відкритим ключем іншої сторони, і надсилає його іншій стороні. Після того, як шифрований текст пароля AES зашифровано відкритим ключем RSA, пароль AES розшифровується за допомогою відповідного закритого ключа RSA та зберігається. Протягом певного періоду часу обидві сторони використовують AES іншої сторони для шифрування та надсилання, обидві сторони отримують зашифрований

текст одна одної, а потім використовують збережений ключ AES для розшифровки та отримання відкритого тексту. При застосуванні цієї схеми ключ AES необхідно періодично реконструювати та замінювати, а ймовірність витоку пароля можна значно зменшити шляхом періодичної заміни пароля.

4.4 Симуляційний тест і результати

Система реалізує схему шифрування, що поєднує алгоритми AES і RSA, і використовує два фіксовані алгоритми шифрування для перевірки алгоритму шифрування. Час, який використовується для шифрування та дешифрування AES, в основному однаковий. Час, який використовується для шифрування відкритим ключем RSA, і час шифрування AES в основному однаковий. Різниця невелика, але час, потрібний для дешифрування за допомогою закритого ключа RSA є більшим, результат тесту на рис.4.3.

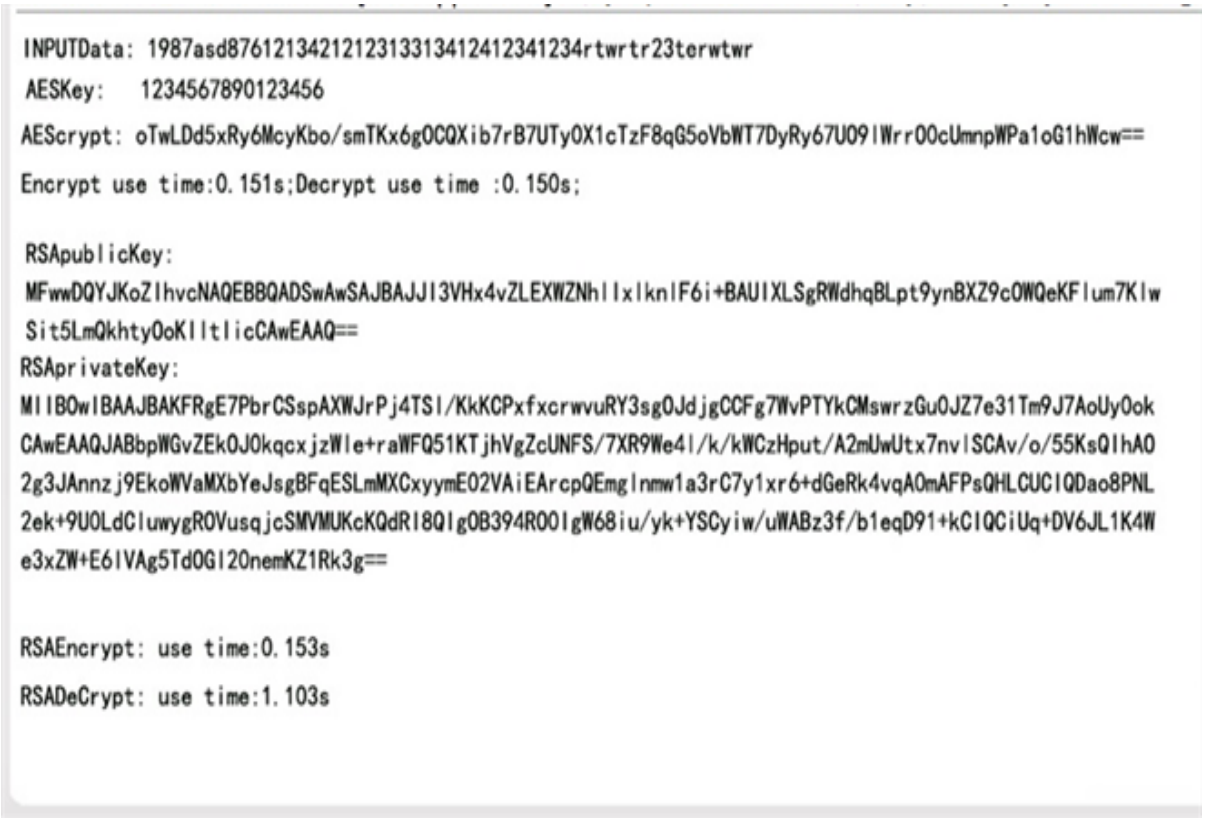


Рис.4.3 Результати тесту

Щоб перевірити продуктивність, шляхом передачі електронного контрактного документа розміром 11 М для перевірки симуляції передачі, 128-бітний ключ шифрування спочатку генерується на відправнику, і ключ шифрується за допомогою асиметричного алгоритму RSA, а потім надсилається одержувачу для прийому. Після отримання шифртексту контракту та симетричного ключа сторона успішно завершила розшифровку контракту, а комбінація розшифровки займає близько 20 секунд. Для шифрування контракту використовується асиметричний алгоритм шифрування. Після відправлення він знову розшифровується. Процес дешифрування займає близько 180 секунд. Таким чином, ефективність використання гібридних алгоритмів шифрування набагато краща, ніж у асиметричних алгоритмів шифрування. Імітаційний тест (як показано в таблиці 4.1 і рисунку 4.4).

Таблиця 4.1

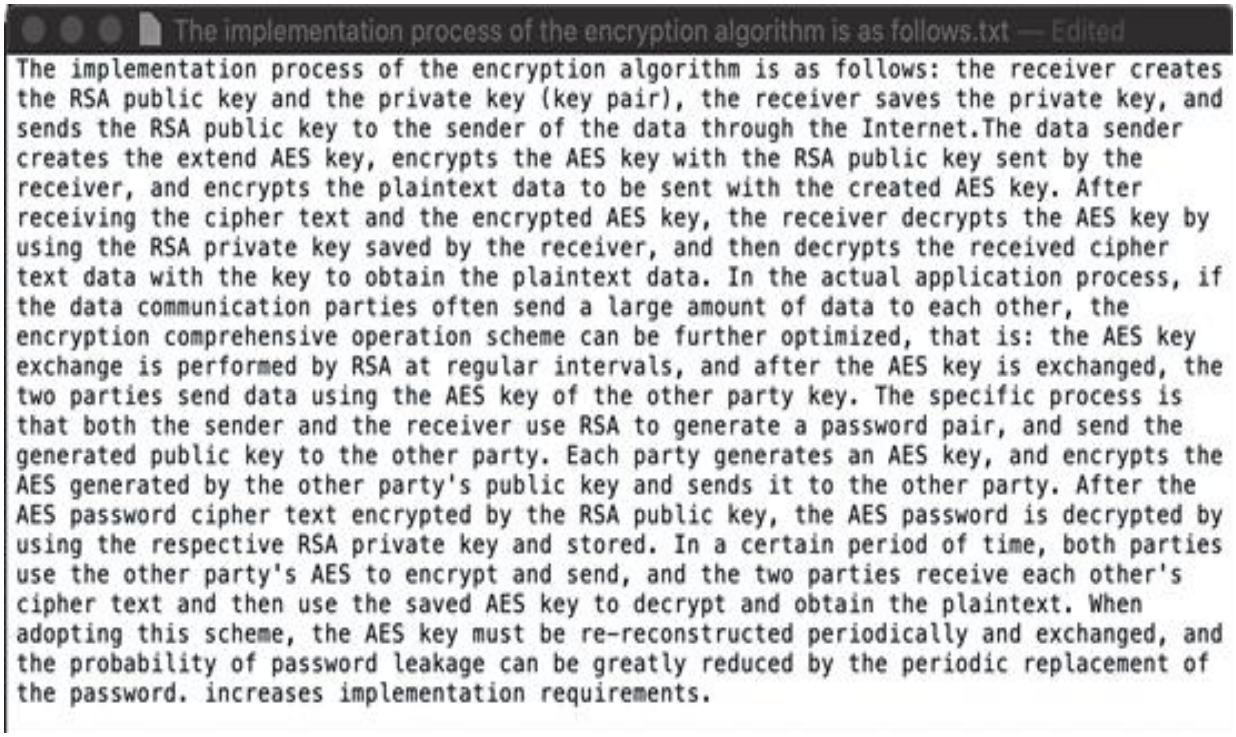
ТИП	Об'єм даних	Зашифровані дані (M)	Час шифрування (s)	Час розшифрування (s)
AES	1	2.034	0.153	0.305
	5	10.172	0.763	1.525
	10	101.724	3.813	15.253
	20	2034.472	19.067	305.064
RSA	1	1.001	2777.778	229,166.667
	5	5.007	13,888.889	1,145,833.333
	10	25.035	69,444.444	11,458,333.333
	20	125.174	347,222.222	229,166,666.667
Гібридне шифрування	1	2.134	0.183	0.405
	5	11.172	0.963	1.895
	10	109.724	4.213	16.153
	20	2087.472	19.867	315.064

Рис.4.4 Порівняння часу шифрування та дешифрування стандартного AES та гібридного алгоритму.

Експериментально доведено, що складний алгоритм шифрування за швидкістю роботи близький до алгоритму шифрування з симетричним ключем. Згідно з тестовими даними, складний алгоритм шифрування лише на 0,8 секунди довший, ніж алгоритм симетричного шифрування для 20 млн даних. Для такого роду програм шифрування алгоритм може відповідати щоденному використанню великої кількості програм.

За допомогою експериментальних випробувань складний алгоритм шифрування базується на безпеці. Складний алгоритм шифрування (гібридне шифрування) використовує AES для основних даних. З точки зору безпеки даних, це безпечніше, а ключ шифрування AES здійснюється через RSA. Зашифровано, тому зашифровані дані більш безпечні, ніж AES.

Далі ми порівнюємо підхід нашого методу з продуктивністю інших методів, для яких ми використовуємо файл text.txt (Рис.4.5) із 409 символами розміром 4 Кб.


The implementation process of the encryption algorithm is as follows.txt — Edited

The implementation process of the encryption algorithm is as follows: the receiver creates the RSA public key and the private key (key pair), the receiver saves the private key, and sends the RSA public key to the sender of the data through the Internet.The data sender creates the extend AES key, encrypts the AES key with the RSA public key sent by the receiver, and encrypts the plaintext data to be sent with the created AES key. After receiving the cipher text and the encrypted AES key, the receiver decrypts the AES key by using the RSA private key saved by the receiver, and then decrypts the received cipher text data with the key to obtain the plaintext data. In the actual application process, if the data communication parties often send a large amount of data to each other, the encryption comprehensive operation scheme can be further optimized, that is: the AES key exchange is performed by RSA at regular intervals, and after the AES key is exchanged, the two parties send data using the AES key of the other party key. The specific process is that both the sender and the receiver use RSA to generate a password pair, and send the generated public key to the other party. Each party generates an AES key, and encrypts the AES generated by the other party's public key and sends it to the other party. After the AES password cipher text encrypted by the RSA public key, the AES password is decrypted by using the respective RSA private key and stored. In a certain period of time, both parties use the other party's AES to encrypt and send, and the two parties receive each other's cipher text and then use the saved AES key to decrypt and obtain the plaintext. When adopting this scheme, the AES key must be re-reconstructed periodically and exchanged, and the probability of password leakage can be greatly reduced by the periodic replacement of the password. increases implementation requirements.

Рис.4.5 Файл TEXT.txt використовується для порівняння кількох підходів.

Результати аналізу наведено в таблиці 2. Порівняно з Віджілою та Мунес Вараном [52], наш підхід виявляється кращим, а також перевершує підхід Дімаса Натаніеля, Файсала, Деві Суріана, [45], коли йдеться про шифрування, але нам все одно потрібно покращити нашу продуктивність, оптимізувавши розшифровку 0.312.

Таблиця 4.2

Методи	Зашифрований текст(кБ)	Час шифрування(ч)	Час розшифрування(s)
Віджіла, Мунес [12]	459.118	34700,00	0.83
Дімас, Файсал, Суріан[5]	15.587	0.263	0.206
Запропонований	17.966	0.258	0.312

4.5 Аналіз алгоритму

Для об’єктивного аналізу алгоритму потрібно зрівняти різні параметри, такі як пам’ять, ентропія, проблеми керування ключами та вичерпного пошуку (метод грубої сили).

1. Для реалізації різні методи шифрування вимагають різних розмірів пам'яті. Необхідна пам'ять залежить від кількості операцій, які повинен виконати алгоритм, використовуваного розміру ключа, векторів, що використовуються для ініціалізації, і типів операцій. Використана пам'ять програми впливає на витрати [53] [54]. Важливо, щоб необхідна пам'ять була якомога меншою.

Рисунок 4.6 і таблиця 4.3 відображають обсяг пам’яті, який використовується для визначених алгоритмів для операцій пристрою. Поточний AES споживає найменшу кількість пам’яті, тоді як RSA споживає максимальну кількість пам’яті на робочий пристрій.

Гібридний алгоритм потребує середнього розміру пам’яті та демонструє невелику різницю порівняно з AES, але кращий результат, ніж RSA. Отже, якщо для будь-якої програми вимогою є найменший розмір пам’яті, AES є найкращим вибором, але з точки зору безпеки гібридне шифрування ефективніше.

Таблиця 4.3

Алгоритм	Використання пам'яті(KB)
AES	14,70
RSA	31,50
Гібридне шифрування	17,76

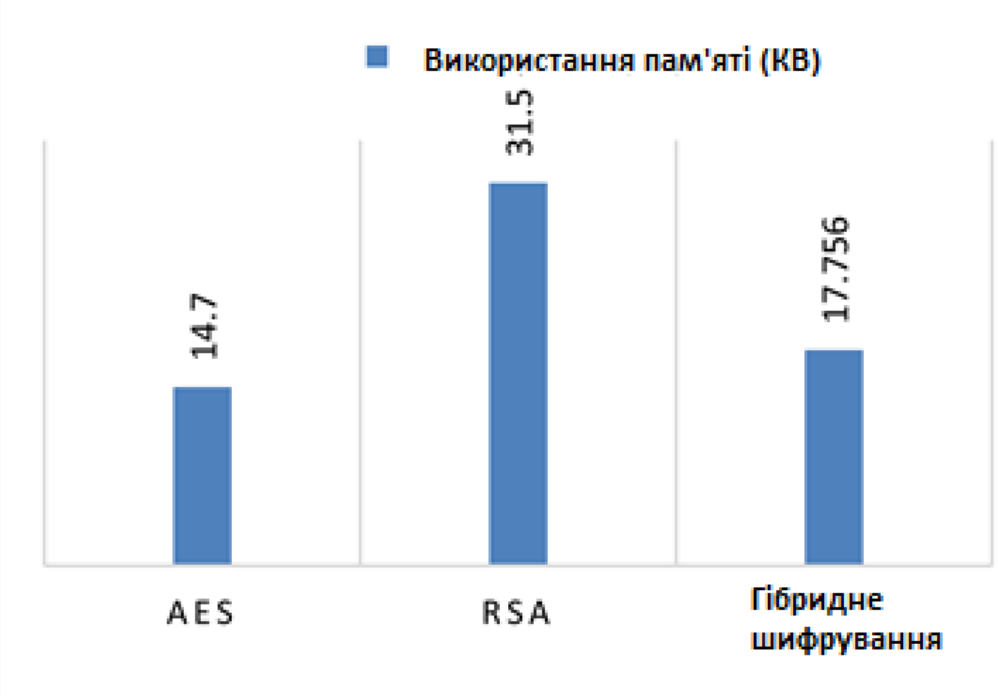


Рис.4.6 Порівняння споживання пам'яті.

2. Порівняння ентропіїї - популярна і класична міра невизначеності в теорії пізнання була описана в 1948 році (Shannon, 1948) [55]. Шеннон припустив, що ентропія H(X) може бути визначена середньою кількістю інформації дискретної випадкової величини X.

$$H\ (X) = \sum_{i=1}^{n} p(x_i) \log_2 p(x_i) \qquad (4.1)$$

на таких умовах:

- X складається зі скінченного простору вибірки $x_1, x_2, x_3, \ldots, x_n$;
- P(xi) розподіл ймовірностей, $x_i > X$;
- $\sum_i^n 1\ p(x_i)$

На рисунку 4.7 показано, що гібридне кодування усереднює максимальну середню ентропію на байт кодування. Ентропія - це випадкова міра інформації. Криптографічні алгоритми роблять випадковість невід'ємною та бажаною властивістю. Гібридне шифрування забезпечує високу випадковість вихідних даних, що робить дані менш сприйнятливими до атак.

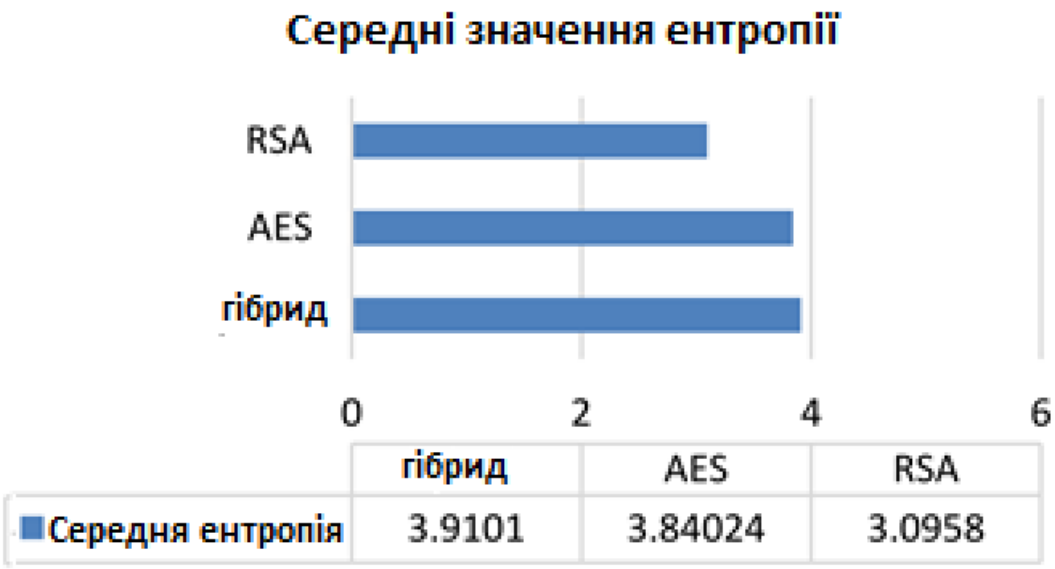


Рис.4.7 Порівняння ентропії на байт із наявним результатом.

3. Теоретичний аналіз проблеми керування ключами вказує на те, що запропонований алгоритм долає проблеми безпеки, проблеми керування ключами. Керування ключами легше за допомогою залучення надійної третьої сторони [56]. Будь-які два комп'ютери не потребують спільного використання ключа. Тому у всій мережі з n комп'ютерів буде доступно лише n ключів. Принцип криптосистеми з відкритим ключем полягає в тому, що ключ шифрування та ключ дешифрування розділені. Кожен може зробити власні створені ключі шифрування та алгоритми загальнодоступними, а секретні лише ключі розшифровки. Кожен, хто використовує цей ключ шифрування та алгоритм для надсилання зашифрованої інформації користувачеві, може відновити її. Перевага криптографії з відкритим ключем полягає в тому, що шлях передачі ключа не повинен бути занадто високим, що значно спрощує керування ключами.

4. AES вже є безпечним алгоритмом, який виходить за рамки криптоаналізу. Хакери часто прагнуть знайти ключ шифру криптоаналізу, який можна використовувати для декодування зашифрованого тексту [57] [58]. Найбільше, що криптоаналіз в теорії - це груба сила, яку можна використовувати проти всіх криптографічних алгоритмів. Під час атаки

грубою силою хакери шукають ключ шифру для всіх можливих комбінацій ключів. Вони перевіряють кожну можливу комбінацію клавіш і виконують дешифрування сліду, щоб перевірити, чи це правильний ключ. Питання тепер полягає в тому, скільки часу потрібно грубій силі, щоб знайти справжній ключ? Час для атаки грубою силою залежить від розміру ключа. Це можна знайти дуже легко, якщо розмір ключа невеликий. Але якщо розмір ключа довший, може знадобитися досить багато часу, щоб знайти справжній ключ.

Висновок

У цьому розділі було запропоновано вирішити проблему складного керування ключами алгоритму AES і низької ефективності RSA шляхом скорочення раунду виконання та модифікації початкового ключа. Було введено складний алгоритм шифрування, описано модифікацію алгоритму AES, представлено комбінацію алгоритму RSA та його методологію. Також було продемонстровано моделювання та аналіз результатів. Використовуючи гібридне шифрування, швидкість шифрування значно покращується, а рівень безпеки покращується.

Система шифрування, що поєднує алгоритми AES і RSA, повністю використовує переваги симетричного та асиметричного ключів. Сеансовий ключ, який використовується у файлі, зашифровано за допомогою RSA, а шифрування файлу даних зашифровано за допомогою AES. Ефективність обробки шифрування системи висока. Проаналізовано та узагальнено алгоритми шифрування, які зазвичай використовуються в криптографії. Алгоритм шифрування AES реалізований на основі мови JAVA. Алгоритм упакований і розроблений для змішаного використання алгоритмів шифрування AES і RSA. Також продемонстровано системний аналіз поширеного алгоритму шифрування. Завдяки системній розробці алгоритму шифрування він може допомогти відповідному користувачеві.

РОЗДІЛ 5
РОЗРОБКА СТАРТАП ПРОЕКТУ

5.1 Опис ідеї проекту

У цьому розділі мною запропоновано інноваційний підхід до підвищення безпеки сайтів та інших суміжних технологій за допомогою комплексної системи шифрування, яка реалізована за допомогою AES. Мій проект направлений на збільшення можливостей протистояння та запобігання атакам на технології, які створюють різноманітні компанії, а також збільшення рівня довіри та лояльності користувачів.

Таблиця 5.1

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Метод підвищення безпеки технологій передавання даних, що дає можливість марнувати майнові та немайнові ресурси атакуючих, а також збільшує відсоток задоволених клієнтів.	1. Підвищення рівня довіри клієнтів. 2. Зменшення кількості атак на технології передавання даних. 3. Покращення рівня конфіденційності та збереження даних.	Компанії що займаються розгортанням та обслуговуванням сайтів та інших технологій зможуть зменшити неефективні витрати при ліквідаціях результатів атак, яких можливо уникнути та сконцентрувати увагу на атаках націлених на аутентифікацію, що використовують неусунені вразливості сервісів.

5.2 Технологічний аудит ідеї проекту

Система реалізує схему шифрування, що поєднує алгоритми AES і RSA, і використовує два фіксовані алгоритми шифрування для перевірки алгоритму шифрування.

Система шифрування, що поєднує алгоритми AES і RSA, повністю використовує переваги симетричного та асиметричного ключів. Сеансовий ключ, який використовується у файлі, зашифровано за допомогою RSA, а шифрування файлу даних зашифровано за допомогою AES.

Використовується для локальних комп'ютерних мереж та сенсорних мереж передачі інформації, створення телекомунікаційної мережі безпроводового космічного радіозв'язку та радіозв'язку між аеростратосферними телекомунікаційними платформами.

Таблиця 5.2

№	Ідея проекту	Технології реалізації	Наявність технологій	Доступність технологій
1	Комплексна система шифрування, реалізована за допомогою AES	Технології алгоритмів шифрування за допомогою RSA та AES.	Залежить від компанії, яка розгортає	Є доступними та безкоштовними для використання
2		Використання власного обладнання	Залежить від компанії, яка розгортає	Доступний, необхідна купівля обладнання
Обрана технологія реалізації проекту: технології алгоритмів шифрування за допомогою RSA та AES.				

5.3 Аналіз ринкових можливостей запуску стартап-проекту

При дослідженні ринкових можливостей, в першу чергу проведений аналіз попиту: наявність попиту, обсяг, динаміка розвитку ринку. Дані наведені у таблиці нижче.

Таблиця 5.3

№	Показники стану ринку (найменування)	Характеристика
1	Кількість головних гравців, од	3
2	Загальний обсяг продаж	?
3	Динаміка ринку (якісна оцінка)	Зростає
4	Наявність обмежень для входу	Немає
5	Специфічні вимоги для стандартизації, специфікації	ISO/IEC 18033-3, IEEE 1363
6	Середня норма рентабельності в галузі (або по ринку), %	?

Виходячи із отриманих даних та необхідності підвищення безпеки у технологіях передачі даних можна зробити висновок, що ринок є привабливим для входження.

Наступний крок - визначення потенційних груп клієнтів, їх характеристики та орієнтовний перелік вимог до товару (послуги) для кожної групи, представлені в наступній таблиці.

Таблиця 5.4

№	Потреба, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару (послуги)
1	Лояльність клієнтів	Компанії що розгортають мережі та пропонують сервіси	Компанії можуть самі обрати як саме використати даний алгоритм та на якому обладнанні буде реалізовуватися даний метод	Впровадження алгоритму для підвищення безпеки технологій
				Використання зібраних алгоритмів атак для більш швидкого запобігання майбутнім атакам

Далі проведемо аналіз ринкового середовища – визначення факторів, які сприяють ринковому впровадженню проекту та які йому перешкоджають.

Таблиця 5.5

№ п/п	Фактор	Зміст загрози	Можлива реакція компанії
1	Необхідність введення нового типу аутентифікації	Компанії, які вже використовують інакший тип аутентифікації, який їх влаштовує не будуть бачити необхідності змінювати її на всіх пристроях, так як це призведе до значних затрат.	Запропонований тип аутентифікації має менші ризики, а отже можливі збитки будуть меншими.
2	Введення нових методів шифрування	Компанії, які вже використовують інші методи шифрування, які їх влаштовують не будуть бачити необхідності змінювати їх, так як це призведе до значних затрат.	Запровадження нових методів шифрування, так як вони мають більшу захищеність.

5.4 Розроблення ринкової стратегії проекту

Перший крок розроблення ринкової стратегії передбачає визначення стратегії охоплення ринку (опис цільових груп потенційних споживачів)

Таблиця 5.6

№	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
1	Компанії оператори	Переважно готові	Дуже високий	Низька	Легко
2	Розробники сайтів	Переважно готові	Дуже високий	Низька	Легко
Цільовими групами обрано компанії операторів та вендорів, які зацікавлені у підвищенні безпеки технологій за допомогою використання схеми шифрування, що поєднує алгоритми AES і RSA.					

За результатами аналізу потенційних груп споживачів (сегментів) визначено стратегію охоплення ринку – масовий маркетинг. Тому, що компанія працює із всім ринком, пропонуючи стандартизовану програму (включно із характеристиками товару/послуги).

5.5 Розроблення маркетингової програми стартап-проекту

Маркетингова програма – це запорука вдалого запуску продукту і комплекс взаємопов'язаних завдань, заходів соціального, економічного, науково-технічного, виробничого, організаційного характеру з визначенням ресурсів, що використовуються, а також джерел одержання цих ресурсів. Він намічений для планомірного здійснення, об'єднаний єдиною метою та залежний від певних строків. Маркетингова програма передбачає планування конкретних дій з реалізації маркетингових

стратегій. В ній оптимально поєднуються інструменти маркетингу з урахуванням конкретного періоду дії плану і відповідного фінансового забезпечення.

Першим кроком є формування маркетингової концепції товару, який отримає споживач. Для цього потрібно підсумувати результати попереднього аналізу конкурентоспроможності товару.

Таблиця 5.7

№	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентом
1	Забезпечення кращої безпеки передачі даних	Використання схеми шифрування, що поєднує алгоритми AES і RSA.	Покращення безпеки за допомогою модифікованого алгоритму шифрування.
2	Анулювання сертифікатів користувачів, які використовують вразливості сервісів	Отримання інформації про небезпечність вразливостей, що ліквідуються з можливістю зменшити їх шкоду	Акцентування на необхідності поставити певним типам вразливостей більший пріоритет

Останньою складової програми розроблення концепції маркетингових комунікацій, що спирається на попередньо обрану основу для позиціонування, визначену специфіку поведінки клієнтів

Таблиця 5.8

№ п/п	Специфіка поведінки цільових клієнтів	Канали комунікацій	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення
1	Відкриті до нового, можлива потреба в даних тестування	Корпоративна пошта та веб-застосунки, професійні соціальні мережі	Доступність, якість, швидкість, прозорість. Можливість покращення безпеки передачі даних	Акцентування на новизні продукту та відкритість до тестування

У концепції маркетингових комунікацій компанія ретельно координує роботу каналів комунікації, рекламу в професійних соціальних мережах, продаж продукту, стимулювання збуту. Слідкує за змінами у поведінці та потребах цільових споживачів

Висновок

В даному розділі був проведений маркетинговий аналіз перспектив реалізації схеми шифрування, що поєднує алгоритми AES і RSA, і використовує два фіксовані алгоритми шифрування для перевірки алгоритму шифрування. Також було проведене оцінювання можливостей її ринкового впровадження.

В результаті дослідження визначено, що існує можливість ринкової комерціалізації проекту в першу чергу завдяки комбінованому підходу до вирішення проблем. Такий підхід забезпечує велику гнучкість під час оптимізації та дозволяє забезпечити кращу безпеку даних.

Проведений аналіз підтверджує, що подальша імплементація проекту є доцільною.

ЗАГАЛЬНІ ВИСНОВКИ ПО РОБОТІ

У першому розділі описані інформаційні системи та їх типи. Інформаційна система це інтегрований набір компонентів для збору, зберігання й обробки даних, а також для доставки інформації, знань і цифрових продуктів. Основними компонентами інформаційних систем є комп'ютерне обладнання та програмне забезпечення, телекомунікації, бази даних і сховища даних, людські ресурси та процедури.

Також була наведена класифікація та визначення мереж за просторовим охопленням. А саме класифікують наступні мережі: мережа на тілі (BAN - body area network), нанорозмірна мережа, персональна мережа (PAN - personal area network), локальна мережа (LAN - local area network), домашня мережа (HAN - home area network), кампусна мережа (CAN - campus area network), столична мережа (MAN - metropolitan area network), мережа радіодоступу (RAN - radio access network), глобальна мережа (WAN - wide area network), корпоративна приватна мережа), віртуальна приватна мережа (VPN - virtual private network) та глобальна мережа (GAN - global area network). Дані мережі мають різний принцип роботи, зону покриття, кількість хостів та/або локальних мереж.

У другому розділі було представлено короткий огляд основ мереж, їх роботу та архітектуру. Було визначено, що існує безліч способів розробити мережеву архітектуру, але більшість з них належить до одного з двох типів. Це однорангові та клієнт/сервер архітектури. Проектування будь-якої архітектури цифрової мережі передбачає оптимізацію її складових блоків. До них належать апаратне забезпечення, носії передачі, протоколи та топологія. Різні мережеві архітектури мають свої плюси та мінуси; і знання їх є ключем до вибору правильної архітектури згідно різних потреб та вимог.

Набір Інтернет протоколів, широко відомий як TCP/IP, є основою для організації набору протоколів зв'язку, що використовуються в Інтернеті та

подібних комп’ютерних мережах відповідно до функціональних критеріїв. Основними протоколами в наборі є Transmission Control Protocol (TCP), User Datagram Protocol (UDP) і Internet Protocol (IP)

Ранній архітектурний документ, RFC 1122 під назвою «Вимоги до хосту» структурований у параграфах, які стосуються рівнів, але документ посилається на багато інших архітектурних принципів і не наголошує на рівнях. Він узагальнено визначає чотирирівневу модель, яка складається з прикладного, транспортного, мережевого та канального рівнях

Було розглянуто IPS, включаючи протоколи, які використовуються в реальних мережах, та показано як дані передаються між вузлами локальної мережі, а також у віддалених мережах за допомогою маршрутизації.

В третьому розділі було розглянуто криптографічні методи захисту та шифрування. А саме поняття криптографічного шифрування, типи криптосистем та сучасні алгоритми шифрування.

Існує два основних типи криптосистем: симетричні та асиметричні. У симетричних системах, той самий секретний ключ шифрує та розшифровує повідомлення. Маніпулювання даними в симетричних системах відбувається значно швидше, ніж в асиметричних системах. Асиметричні системи використовують «відкритий ключ» для шифрування повідомлення та відповідний «приватний ключ» для його дешифрування. Перевага асиметричних систем полягає в тому, що відкритий ключ можна вільно публікувати, що дозволяє сторонам встановлювати безпечний зв’язок без спільного секретного ключа. На практиці асиметричні системи використовуються для того, щоб спочатку обмінюватися секретним ключем, а потім захищати зв’язок через більш ефективну симетричну систему, використовуючи цей ключ.

У четвертому розділі було запропоновано вирішити проблему складного керування ключами алгоритму AES і низької ефективності RSA шляхом скорочення раунду виконання та модифікації початкового ключа.

Було введено складний алгоритм шифрування, описано модифікацію алгоритму AES, представлено комбінацію алгоритму RSA та його методологію. Також було продемонстровано моделювання та аналіз результатів. Використовуючи гібридне шифрування, швидкість шифрування значно покращується, а рівень безпеки покращується.

Система шифрування, що поєднує алгоритми AES і RSA, повністю використовує переваги симетричного та асиметричного ключів. Сеансовий ключ, який використовується у файлі, зашифровано за допомогою RSA, а шифрування файлу даних зашифровано за допомогою AES. Проаналізовано та узагальнено алгоритми шифрування, які зазвичай використовуються в криптографії. Алгоритм шифрування AES реалізований на основі мови JAVA. Алгоритм упакований і розроблений для змішаного використання алгоритмів шифрування AES і RSA. Завдяки системній розробці алгоритму шифрування він може допомогти відповідному користувачеві.

В п’ятому розділі був проведений маркетинговий аналіз перспектив реалізації схеми шифрування, що поєднує алгоритми AES і RSA, і використовує два фіксовані алгоритми шифрування для перевірки алгоритму шифрування. Також було проведене оцінювання можливостей її ринкового впровадження.

В результаті дослідження визначено, що існує можливість ринкової комерціалізації проекту в першу чергу завдяки комбінованому підходу до вирішення проблем. Такий підхід забезпечує велику гнучкість під час оптимізації та дозволяє забезпечити кращу безпеку даних.

Проведений аналіз підтверджує, що подальша імплементація проекту є доцільною.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Zwass, V. (2022, August 24). Information system. Encyclopedia Britannica. https://www.britannica.com/topic/information-system

2. Stair, Ralph (2020). Principles of Information Systems. George Reynolds (14th ed.). Mason, OH: Cengage. ISBN 978-0-357-11252-6. OCLC 1305839544.

3. Awati, R., & Rosencrance, L. (2021, October 6). computer hardware. Networking. https://www.techtarget.com/searchnetworking/definition/hardware

4. Computer Hope. (2021, August 16). What is Software? https://www.computerhope.com/jargon/s/software.htm

5. Chai, W., & Casey, K. (2022, October 24). Software as a Service (SaaS). Cloud Computing. https://www.techtarget.com/searchcloudcomputing/definition/Software-as-a-Service

6. Peterson, R. (2022, October 1). What is a Database? Definition, Meaning, Types with Example. Guru99. https://www.guru99.com/introduction-to-database-sql.html

7. Ullah, S.; Higgins, H.; Braem, B.; Latre, B.; Blondia, C.; Moerman, I.; Saleem, S.; Rahman, Z.; Kwak, K. S. (2012). "A Comprehensive Survey of Wireless Body Area Networks: On PHY, MAC, and Network Layers Solutions". Journal of Medical Systems. 36 (3): 1065–1094. doi:10.1007/s10916-010-9571-3. hdl:1854/LU-3234782. PMID 20721685. S2CID 7988320.

8. Poslad, Stefan (2009). Ubiquitous Computing Smart Devices, Smart Environments and Smart Interaction. Wiley. ISBN 978-0-470-03560-3. Archived from the original on 2012-02-15.

9. Bush, S. F. (2010). Nanoscale Communication Networks. Artech House. ISBN 978-1-60807-003-9.

10. Yasar, K. (2022, October 3). personal area network (PAN). Mobile Computing. https://www.techtarget.com/searchmobilecomputing/definition/personal-area-network

11. What Is a LAN? (2022, September 16). Cisco. https://www.cisco.com/c/en/us/products/switches/what-is-a-lan-local-area-network.html

12. IEEE P802.3ba 40Gb/s and 100Gb/s Ethernet Task Force. (n.d.). https://www.ieee802.org/3/ba/

13. Wright, G. (2021, February 24). metropolitan area network (MAN). Networking. https://www.techtarget.com/searchnetworking/definition/metropolitan-area-network-MAN

14. What Is a Wide Area Network (WAN) and How Does It Work? (2021, June 22). Lifewire. https://www.lifewire.com/wide-area-network-816383

15. What Is a VPN? - Virtual Private Network. (2022, October 25). Cisco. https://www.cisco.com/c/en/us/products/security/vpn-endpoint-security-clients/what-is-vpn.html

16. Fusion Connect. (2021, September 13). What is Network Architecture? And, How Does It Work? https://www.fusionconnect.com/blog/what-is-network-architecture

17. The Editors of Encyclopaedia Britannica. (2009, March 13). Protocol | Definition, Examples, & Facts. Encyclopedia Britannica. https://www.britannica.com/technology/protocol-computer-science

18. Grant, T. J., Janssen, R. H. P., & Monsuur, H. (2014). Network Topology in Command and Control: Organization, Operation, and Evolution. ISBN 9781466660595.

19. Advantages and Disadvantages of Network Architecture. (n.d.). Huawei Enterprise Support Community.

https://forum.huawei.com/enterprise/en/advantages-and-disadvantages-of-network-architecture/thread/829661-100181

20. TCP/IP Internet Protocol. (2020, November 24). LivingInternet. https://www.livinginternet.com/i/ii_tcpip.htm

21. Shacklett, M. E., Novotny, A., & Gerwig, K. (2021, July 13). TCP/IP. Networking. https://www.techtarget.com/searchnetworking/definition/TCP-IP

22. RFC 1958 - Architectural Principles of the Internet. (n.d.). https://datatracker.ietf.org/doc/html/rfc1958

23. TCP/IP Illustrated, Volume 1: The Protocols, Addison-Wesley, 1994, ISBN 0-201-63346-9

24. Internet Protocol Suite". Requirements for Internet Hosts – Communication Layers. 1989. doi:10.17487/RFC1122. RFC 1122.

25. SCTP (Stream Control Transmission Protocol) – the reliable, message-oriented transport protocol. (2019, January 2). IONOS Digital Guide. https://www.ionos.com/digitalguide/server/know-how/sctp/

26. GeeksforGeeks. (2020, June 24). Real Time Transport Protocol (RTP). https://www.geeksforgeeks.org/real-time-transport-protocol-rtp/

27. Jacobs, D. (2022, January 26). Intro to encapsulation and decapsulation in networking. Networking. https://www.techtarget.com/searchnetworking/tip/Intro-to-encapsulation-and-decapsulation-in-networking

28. Forshaw, J. (2017). Attacking Network Protocols: A Hacker's Guide to Capture, Analysis, and Exploitation (1st ed.). No Starch Press. ISBN 978-1-59327-750-5

29. Carrol, J. D. J. &. (2022). Routing TCP/IP, Volume 1 (2nd Edition) (2nd ed.). PEARSON INDIA. ISBN 8131700429

30. Simplilearn. (2022, November 11). What Is Data Encryption: Types, Algorithms, Techniques and Methods. Simplilearn.com. https://www.simplilearn.com/data-encryption-methods-article

31. Kahn, D. (1996). The Codebreakers: The Comprehensive History of Secret Communication from Ancient Times to the Internet. Scribner/Macmillan Library Reference USA/Simon & Schuster, Inc.

32. Simmons, G. J. (1999, July 26). Cryptology | Definition, Examples, History, & Facts. Encyclopedia Britannica. https://www.britannica.com/topic/cryptology

33. Johnson, L. (2016). Security Controls Evaluation, Testing, and Assessment Handbook. Syngress Publishing. ISBN 9780128023242

34. R. Shirey (August 2007). Internet Security Glossary, Version 2. Network Working Group. doi:10.17487/RFC4949

35. Alvarez, Rafael; Caballero-Gil, Cándido; Santonja, Juan; Zamora, Antonio (27 June 2017). "Algorithms for Lightweight Key Exchange". Sensors. 17 (7): 1517. doi:10.3390/s17071517

36. Asymmetric encryption. (2022, May 31). IONOS Digital Guide. https://www.ionos.com/digitalguide/server/security/public-key-encryption/

37. Keyfactor. (2021, August 5). Types of Encryption Algorithms + Pros and Cons for Each –. https://www.keyfactor.com/resources/types-of-encryption-algorithmns/

38. Cobb, M. (2021, November 4). RSA algorithm (Rivest-Shamir-Adleman). Security. https://www.techtarget.com/searchsecurity/definition/RSA

39. Daniel, B. (2022, March 24). What Is AES Encryption? [The Definitive Q&A Guide]. https://www.trentonsystems.com/blog/aes-encryption-your-faqs-answered

40. Saha, P. (2022, August 17). Why 3DES or Triple DES is Officially Being Retired. Encryption Consulting. https://www.encryptionconsulting.com/why-3des-or-triple-des-is-officially-being-retired/

41. Odeh, A., Masadeh, S.R. and Azzazi, A. (2015) A Performance Evaluation of Common Encryption Techniques with Secure Watermark System (SWS). International Journal of Network Security & Its Applications, https://doi.org/10.5121/ijnsa.2015.7303

42. Kazlauskas, K., Smaliukas, R. and Vaicekauskas, G. (2016) A Novel Method to Design S-Boxes Based on Key-Dependent Permutation Schemes and Its Quality Analysis. International Journal of Advanced Computer Science and Applications, https://doi.org/10.14569/IJACSA.2016.070412

43. Priyadarshini, P., Narayankar, P., Narayan, D.G. and Meena, S.M. (2016) A Comprehensive Evaluation of Cryptographic Algorithms: DES, 3DES, AES, RSA and Blowfish. Procedia Computer Science, 78, 617-624. https://doi.org/10.1016/j.procs.2016.02.108

44. Ali, A.H. and Sagheer, A.M. (2017) Design of an Android Application for Secure Chatting. International Journal Computer Network and Information Security, 9, 29-35. https://doi.org/10.5815/ijcnis.2017.02.04

45. Dimas, N., Faisal and Suryani, D. (2018) Text Encryption in Android Chat Applications using Elliptical Curve Cryptography (ECC). Procedia Computer Science, 135, 283-291. https://doi.org/10.1016/j.procs.2018.08.176

46. Teng, L., Li, H., Yin, S. and Sun, Y. (2019) A Modified Advanced Encryption Standard for Data Security. International Journal of Network Security, 22, 112-117.

47. Tan, C., Deng, X. and Zhang, L. (2018) Identification of Block Ciphers under CBC Mode. Procedia Computer Science, 131, 65-71. https://doi.org/10.1016/j.procs.2018.04.186

48. Abdel-hafeez, S., Sawalmeh, A. and Bataineh, S. (2017) High Performance AES Design using Pipelining Structure over GF((24)2). 2007 IEEE International Conference on Signal Processing and Communications, Dubai, 24-27 November 2007, 716-719. https://doi.org/10.1109/ICSPC.2007.4728419

49. Cao, T. (2016) Design and Implementation of Encryption System Based on AES. Software Development and Application, 21, 141-142.

50. Indra Sena, R.M. and Siva Kumar, A.P. (2016) Secured Data Transmission Using Wavelet Based Steganography and Cryptography by Using AES Algorithm. Procedia Computer Science, 85, 62-69. https://doi.org/10.1016/j.procs.2016.05.177

51. Yang, B. and Bo, L. (2018) Complex Encryption Computer System Realized by AES World Smart Home. Information System Engineering, No. 12, 31-35.

52. Vigila, S.M.C. and Muneeswaran, K. (2009) Implementation of Text Based Cryptosystem Using Elliptic Curve Cryptography. 2009 1st International Conference on Advanced Computing, Chennai, 13-15 December 2009, 82-85. https://doi.org/10.1109/ICADVC.2009.5378025

53. Xia, G. (2019) Technical Browsing of Complex Encryption Computer System Realized by AES. Automotive Application, No. 6, 26-28.

54. Zhang, M. (2019) Alarm System of Complex Encryption Computer System Implemented by AES. Inner Mongolia University of Science and Technology, Baotou, 21-22.

55. Neenu, S. and Bonifus, P.L. (2016) Design of AES Architecture with Area and Speed Tradeoff. Procedia Technology, 24, 1135-1140. https://doi.org/10.1016/j.protcy.2016.05.066

56. Jie, K. and Liu, Y. (2015) Analysis of Data Encryption Algorithms. China Science and Technology, 18, 33-34.

57. Rakesh, K. and Geetu, M. (2015) A Novel Framework for Secure File Transmission Using Modified Aes and md5 Algorithms. International Journal of Information and Computer Security, 7, Article No. 91. https://doi.org/10.1504/IJICS.2015.073012

58. Berry, R., Berry, K. and Kumar, A. (2016) Review on Network Security and Cryptography. International Journal of Innovative Research in Technology, 3, 44-45.