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

Факультет електроніки
(повна назва інституту/факультету)

Кафедра акустичних та мультимедійних електронних систем
(повна назва кафедри)

«На правах рукопису»
УДК 621.397.63

«До захисту допущено»
Завідувач кафедри
_____ С.А. Найда
(ініціали, прізвище)
“_07_” __12_____2020 р.

Магістерська дисертація

зі спеціальності 171 «Електроніка»
(код і назва)

на тему: «Дослідження засобів об’єднання процедур захисту від завад та несанкціонованого доступу в інформаційних системах»

Виконав: студент II курсу, групи ДВ-92мп
(шифр групи)

Гаркуша Анатолій Євгенович _____
(прізвище, ім’я, по батькові) (підпис)

Керівник
професор, д.т.н., проф. Власюк Г.Г. _____
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Консультант
(науковий ступінь, вчене звання, , прізвище, ініціали) (підпис)

Рецензент професор кафедри ЕПС, д.т.н. Мельник І.В. _____
(посада,науковий ступінь,вчене звання, прізвище,ініціали) (підпис)

Засвідчую, що у цьому дипломному проекті немає запозичень з праць інших авторів без відповідних посилань.
Студент _____
(підпис)

Київ – 2020 року

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

Факультет електроніки

Кафедра акустичних та мультимедійних електронних систем

Рівень вищої освіти – другий (магістерський) за освітньо-професійною програмою

Спеціальність 171 «Електроніка» («Електронні системи мультимедіа та засоби Інтернету речей»)

ЗАТВЕРДЖУЮ
Завідувач кафедри
С.А. Найда
(ініціали, прізвище)
« 07» 12 2020 р.

ЗАВДАННЯ

на магістерську дисертацію студенту

Гаркуша Анатолій Євгенович
(прізвище, ім'я, по батькові)

1. Тема роботи — Дослідження засобів об'єднання процедур захисту від завад та несанкціонованого доступу в інформаційних системах
керівник роботи Власюк Ганна Григорівна д.т.н., проф.
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від «05» листопада 2020 р. № 3241-с

2. Строк подання студентом дисертації: 3 грудня 2020р

3. Об'єкт дослідження — дослідження щодо поєднання різноманітних систем захисту інформації

4. Предмет дослідження (Вихідні дані – для магістерської дисертації за освітньо-професійною програмою) — Процедури захисту інформації

5. Перелік завдань, які потрібно розробити :

1. Дослідити загрози інформації в інформаційних системах.

2. Проаналізувати кодування інформації та інформаційну безпеку.

3. Розглянути процедури шифрування як засіб захисту від небажаного доступу.

4. Провести об’єднання процедур кодування та шифрування..

6. Перелік графічного (ілюстративного) матеріалу 14 слайдів презентації - завади в каналах зв’язку, види спотворення інформації що передається, надлишковість інформації як метод захисту від спотвореннь, лінійне кодування, сучасні турбокоди, шифрування інформації від зловмисників, необхідність поєднання алгоритмів кодування та шифрування та розрахунок довжини блоку для процедур поєднання, визначення точок поєднання двох алгоритмів, аналіз сфер використання розробленого алгоритму

7. Орієнтовний перелік публікацій: III Всеукраїнська науково-технічна конференція, Сучасні технології кіно та аудіовізуальних систем – 9-10 грудня 2019, 64с.

8. Консультант розділів дисертації*

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

9. Дата видачі завдання 01.09.2020р.

Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітка
1	Написання першого розділу	11.10.2020	
2	Написання другого розділу	18.10.2020	
3	Написання третього розділу	15.11.2020	
4	Написання четвертого розділу	24.11.2020	
5	Підготовка та оформлення матеріалів матеріалів та пояснювальної записки	29.11.2020	
6	Підготовка та оформлення презентації для доповіді	04.12.2020	

Студент _____ Гаркуша А.Є.
(підпис) (ініціали, прізвище)

Керівник роботи
професор _____ Г.Г.Власюк
(підпис) (ініціали, прізвище)

УДК 621.397.63

РЕФЕРАТ

Гаркуша А.Є. Дослідження засобів об'єднання процедур захисту від завад та несанкціонованого доступу в інформаційних системах: магістерська дис. : 171 Електроніка. Київ, КПІ ім. Ігоря Сікорського, 2020. 85 с.

Атестаційна магістерська робота: 85 с., 16 рис., 3 табл., 14 джерел, 2 додатки.

КАНАЛИ ЗВ'ЯЗКУ, ІНФОРМАЦІЙНІ СИСТЕМИ, ПРОТОКОЛИ, ШИФРУВАННЯ ДАНИХ, КОДУВАННЯ ДАНИХ

Актуальність роботи. Шифрування даних вдосконалюються кожного дня та набувають неабиякого значення для якісного функціонування того чи іншого способу захисту будь-якої інформації від несанкціонованого доступу, перегляду, а також її використання, засновані на перетворенні даних в зашифрований формат.

Мета роботи. Розробити рекомендації для практичної реалізації поєднання процедур кодування та шифрування інформації під час її передачі. Об'єктом дослідження є поєднання процедур захисту інформації.

Для досягнення поставленої мети проведено аналіз сучасного стану захисту інформаційних систем, проаналізовані загрози та способи зменшення їх впливу. Досліджено сучасні архітектури систем захисту інформації. Розроблено модель поєднання процедур кодування та шифрування інформації під час її передавання по каналах зв'язку. Розроблено сценарії для використання сучасних алгоритмів шифрування.

Робота містить аналіз поєднання алгоритмів шифрування та кодування в межах одного алгоритму . Описано протоколи, які використовуються. Проведено порівняльний аналіз адаптації блоків інформації для різних процедур. Запропоновано алгоритм поєднання двох процедур кодування та двох процедур шифрування інформації. Отримані результати можуть бути використані для побудови сучасних ефективних систем безпечної передачі інформації.

SUMMARY

Certification master's thesis: 85 pages, 16 figures, 3 tables, 14 sources, 2 appendices.

The object of research is a combination of information protection procedures.

Purpose: to develop recommendations for the practical implementation of a combination of procedures for encoding and encrypting information during its transmission.

To achieve this goal, an analysis of the current state of protection of information systems, analyzed threats and ways to reduce their impact. Modern architectures of information security systems are studied. A model of combining procedures for encoding and encrypting information during its transmission over communication channels has been developed. Scripts have been developed for the use of modern encryption algorithms.

The paper contains an analysis of a combination of encryption and encryption algorithms within one algorithm. Describes the protocols used. A comparative analysis of the adaptation of information blocks for different procedures. An algorithm for combining two coding procedures and two information encryption procedures is proposed. The obtained results can be used to build modern effective systems for secure transmission of information.

ЗМІСТ

1 АНАЛІТИЧНИЙ ОГЛЯД

1.1 Аналіз проблематики захисту інформації. Джерела загроз інформаційним системам

Джерела загроз інформації можна розділити на три основні групи:

- обумовлені діями суб'єкта (антропогенні);
- обумовлені технічними засобами (техногенні джерела);
- обумовлені стихійними явищами.

Антропогенні джерела загроз можна розглядати як суб'єкти, що мають доступ (санкціонований або несанкціонований) до роботи з інформацією, що захищається. Антропогенні джерела загроз по відношенню до інформаційної системі є зовнішніми або внутрішніми. Зовнішні антропогенні джерела поділяються на випадкові та навмисні. Випадкові джерела приводять до наступних вразливостей: помилок при розробці інформаційної системи та її елементів, помилок в програмному забезпеченні, що відповідає за інформаційну безпеку, різного роду відмов притаманних інформаційній системі. До цих джерел відноситься персонал підрядників, персонал наглядових організацій і аварійних служб. [1]

Загрози від даних джерел, як правилу стаються по причині незнання чи недбалості. Часто проста цікавість може становити загрозу безпеці інформації. Навмисні джерела це корисливі дії порушників. Основна мета таких джерел – умисна дезорганізація роботи, вилучення інформаційних систем з ладу, спотворення інформації за рахунок проникнення в інформаційні ресурси підприємства шляхом несанкціонованого доступу. Внутрішні джерела безпеки представляють собою висококваліфікованих працівників в області інформаційних технологій, знайомих зі специфікою структурою інформаційних систем і принципами роботи засобів захисту інформації. Вони мають можливість

використовувати технічні засоби мережі та встановлене програмне забезпечення.

Серед внутрішніх джерел загроз особливе місце займають загрози у вигляді помилкових дій і порушень вимог експлуатаційної та іншої документації співробітниками установи. Специфічну групу внутрішніх антропогенних джерел безпеки складають особи, що мають психічні проблеми. Ці особи можуть бути з числа основного, допоміжного і технічного персоналу, чи представників служби захисту інформації. Дана група розглядається в складі перерахованих вище джерел загроз, але методи парирування загрозам для цієї групи можуть мати свої відмінності.

Зробимо аналіз можливих загроз:

- Навмисні дії осіб, що мають доступ до інформаційних систем, включно з користувачами та іншими співробітників, що реалізують загрози безпосередньо всередині підприємства (внутрішній порушник);
- Навмисні дії осіб, що не мають доступу до інформаційних систем підприємства і реалізують загрози з зовнішніх мереж зв'язку загального користування або мереж міжнародного інформаційного обміну;
- Загрози, пов'язані з навмисними діями осіб, що не мають доступу до інформаційних систем і реалізують загрози технічними каналами витоку інформації.

Техногенні джерела загроз безпосередньо залежать від властивостей техніки. Такі джерела бувають зовнішніми або внутрішніми. Зовнішні джерела це елементи інформаційних систем: засоби зв'язку та мережі інженерних комунікацій. До внутрішніх джерел можна віднести неякісну техніку та програми призначені для обробки та передавання інформації, а також різноманітні допоміжні засоби (відео нагляду, охорони, сигналізації, телефонії.

До стихійних джерел загроз вирізняються великою кількістю і мінливістю і є, як правило, зовнішніми по відношенню до інформаційних систем. Це в першу чергу різноманітні природні явища: пожежі, повені, землетруси, торнадо.

Спрогнозувати такі події важко. При настанні подібних подій порушується штатний режим функціонування самої інформаційної системи і засобів захисту, що потенційно може привести до порушення конфіденційності, цілісності, доступності та інших характеристик безпеки інформації. Як правило, захист від загроз, що виходять від техногенних та стихійних джерел загроз безпеки інформації, в основному регламентується інструкціями, розробленими і затвердженими оператором з урахуванням особливостей експлуатації інформаційних систем підприємства та діючої нормативної бази установи.

1.2 Процедури захисту інформації

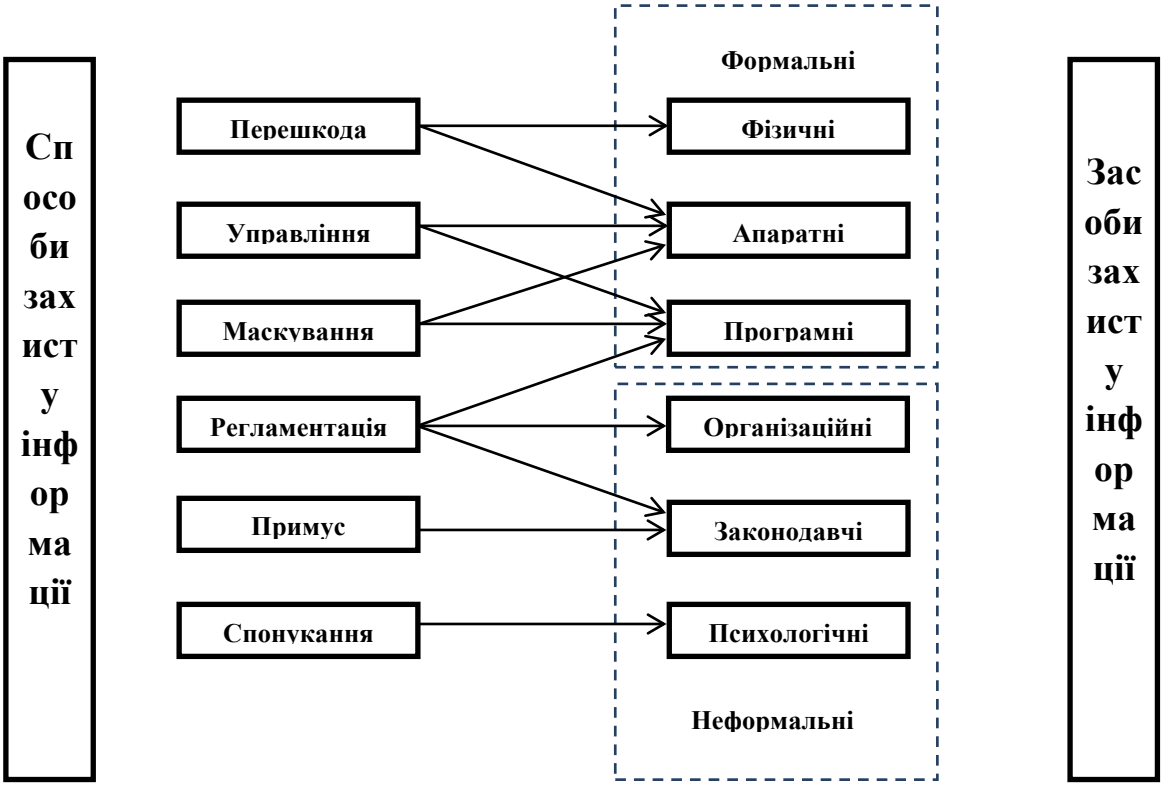


Рисунок 1.1 – Способи захисту інформації.

Методи захисту інформації (рисунок 1.1) поділяють на:

- Перешкода – створення для загрози бар'єрів, при подоланні яких зловмисник отримає складнощі.
- Управління – керування ресурсами захисту інформаційної системи.
- Маскування – умисне спотворення інформації, яке робить їх дуже важкою для розуміння. (наприклад шифрування).
- Регламентація – комплекс організаційних заходів, що створюють такі умови доступу до інформації, які затрудняють доступ сторонніх осіб.
- Примус – створенні умов, які змушують персонал дотримуватися порядку обробки інформації під загрозою відповідальності.

- Спонукання – створення умов, коли користувачі дотримуються правил роботи з інформацією. [2]

1.3 Засоби захисту інформації

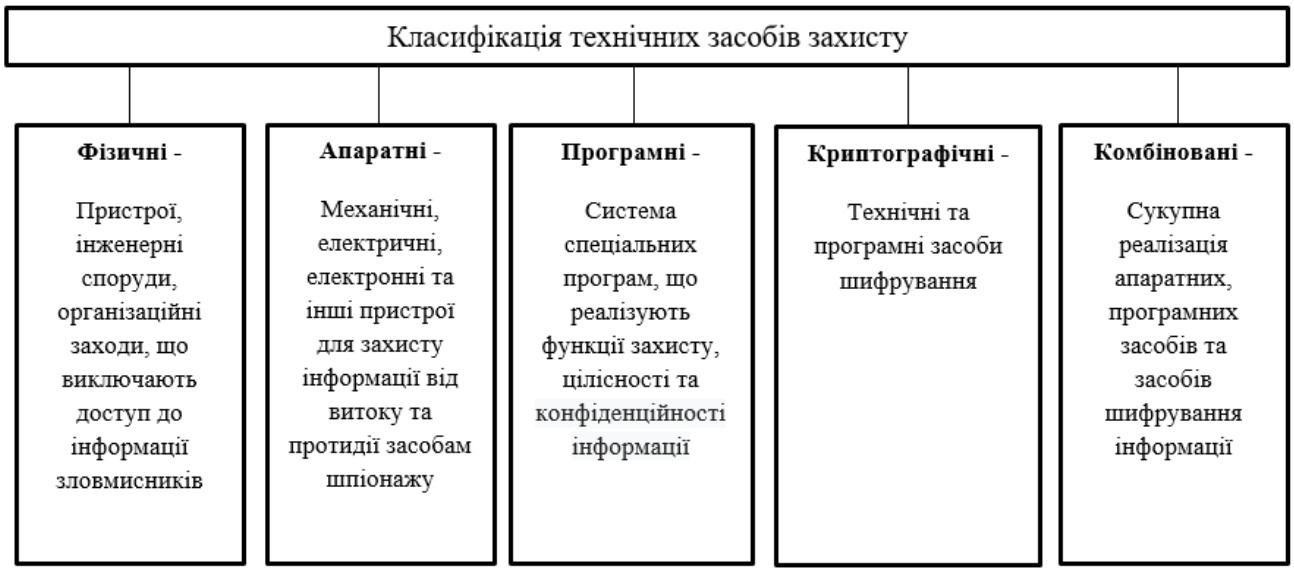


Рисунок 1.2 – Засоби захисту інформації.

До засобів захисту інформації (рисунок 1.2) відносяться:

- Фізичні засоби – пристрої (електричні, механічні, електронні) та системи, які розраховані на автономну роботу, та працюють на пониження впливу різноманітних завад та дестабілізуючих чинників.
- Апаратні засоби – різноманітні електронні чи механічні пристрої, які вмонтовано апаратно в системи обробки, передачі чи зберігання даних чи приєднані до них з метою підвищення рівня захисту інформації.
- Програмні засоби – специфічні програмні модулі, що є частиною програмного забезпечення. Мета цих модулів – підвищення рівня захисту інформації. [3]

1.4 Програмні засоби захисту інформації

Рисунок 1.3 – Засоби програмного захисту.

Програмні засоби (рисунок 1.3) - це об'єктивні форми представлення сукупності даних і команд, призначених для функціонування комп'ютерів і комп'ютерних пристроїв з метою отримання певного результату, а також підготовлені і зафіксовані на фізичному носії матеріали, отримані в ході їх розробок, і породжувані ними аудіовізуальні відображення. [4]

Програмні засоби складаються з програм для ідентифікації користувачів, контролю доступу, кодування, декодування, шифрування та дешифрування інформації, видалення надмірної інформації (тимчасових файлів), проведення тестування систем захисту. Властивості програмних засобів це універсальність, гнучкість, висока надійність, простота установки та налаштування, здатність до модернізації. Недоліки програмного забезпечення це обмежена функціональність мережі, обмежене використання ресурсів серверів чи комп’ютерів, чутливість до випадкових або навмисним змін, залежність від апаратної частини серверів чи комп'ютерів.

Програмне забезпечення можна поділити за своїм функціоналом на:

- Архівації даних;
- Антивірусного захисту;
- Криптографічного захисту;

- Засоби управління доступом;
- Протоколювання і тестування.

Можливе також поєднання функцій:

- Захист баз даних;
- Захист операційних систем;
- Захист інформації.

Інколи резервування інформації потрібно робити за наявності низького рівня ресурсів розміщення баз даних, зокрема з персональних комп'ютерів чи ноутбуків. За таких умов бажано використовувати системи програмного резервування (архівації). Архівація це склеювання кількох файлів і навіть папок в спільний архівний файл, з одночасним зменшенням повного обсягу отриманих файлів зменшенням рівня надмірності інформації, але без її втрат. Тобто забезпечується можливість точного відтворення початкових файлів. Більшості систем архівації створена на використанні алгоритмів стиснення, запропонованих у другій половині 20 сторіччя.

Самі використовувані архівні формати:

- ZIP, 7Z, ARC (DOS і Windows);
- TAR (Unix та Linux);
- JAR, ARJ (Java ARchive);
- RAR (мультисистемний архіватор). [5]

Потрібно лише зробити вибір програми архіватора. При виборі враховуються наступні характеристики – швидкість процесу архівації та деархівації, рівень стиснення інформації, сумісності з різноманітними форматами даних, зручність графічного та консольного інтерфейсу, підтримка операційної системи. Більшість з існуючих архіваторів безкоштовні (Freeware), або умовно-безкоштовні (Shareware). Важливо мати графік архівації інформації, та дотримуватись його.

1.5 Антивірусні програми

Всі сучасні антивірусні програми можна розділити за принципом роботи і призначенням наступним чином:

1. Сканери;
2. Ревізори диска;
3. Вбудовані антивіруси;

Сканери це антивірусні програми які переглядають вміст файлів, розташованих на дисках комп'ютера, а також вміст оперативної пам'яті комп'ютера з метою пошуку вірусів. Сучасні антивірусні сканери шукають шкідливі програми не тільки по їх сигнатурам (тобто за послідовностей байтів даних, характерних для даних вірусів), але і застосовують витончені евристичні алгоритми.

Ревізори диска це такі антивірусні програми, що використовують у своїй роботі метод виявлення змін. У режимі попереднього сканування ревізор диска створює базу даних з контрольними сумами та іншою інформацією, що дозволяє згодом контролювати цілісність файлів. Кожен раз при завантаженні операційної системи або за явною запитом користувача ревізор виконує сканування диска, обчислюючи заново контрольну інформацію. Потім ця інформація звіряється з вмістом попередньо створеної бази даних.

Ревізори диска здатні виявляти зміни, внесені у файли комп'ютерними вірусами та іншими шкідливими програмами, а також користувачами.

Вбудовані антивіруси. У деяких випадках для захисту від вірусів і інших шкідливих програм необхідно використовувати спеціалізовані рішення. Це відбувається коли звичайні промислові антивіруси не в змозі забезпечити необхідний рівень захисту або коли їх застосування негативно позначається на продуктивності системи.

Вбудовані антивіруси здатні захистити спеціалізовані, унікальні і малопоширені інформаційні системи Застосування вбудованих антивірусів дозволяє посилити надійність антивірусного захисту. Крім того, якщо прикладна програма або інформаційна система зберігає дані в своєму внутрішньому форматі, вбудований модуль дозволить виконувати антивірусну перевірку цих даних.

1.6 Криптографія як програмний метод захисту інформаційних систем

Криптографія є однією з двох гілок загального наукового напрямку – криптології. Другою гілкою криптології є криптоаналіз. Цілі криптографії та криптоаналізу прямо протилежні. Криптографічні методи знайшли широке застосування в практичній інформатиці для вирішення численних проблем інформаційної безпеки. У проблематиці сучасної криптографії можна виділити наступні три типи основних завдання:

- забезпечення конфіденційності – захисту довіреної інформації від сторонніх осіб, та цілісності –захисту інформації від заміни або спотворення.
- створення умов для анонімності;
- забезпечення аутентифікації інформації та джерела повідомлення.

Перший тип завдань відноситься до захисту інформації від несанкціонованого доступу по секретному ключу. Доступ до інформації (інформаційних ресурсів) мають тільки власники ключа. Другий і третій типи завдань зобов'язані своєю постановкою масовому застосуванню електронних способів обробки і передачі інформації (банківська сфера, електронна комерція, канали міжособистісної комунікації та ін.).

Конфіденційність – властивість інформації бути відомою тільки допущеним, що пройшли авторизацію суб'єктам системи (користувачам,

програмам, процесам). Авторизація – надання суб'єктам доступу до об'єктів системи.

Аутентифікація – перевірка ідентифікації користувача, пристрою або іншого компонента в системі (зазвичай для прийняття рішення про дозвіл доступу до ресурсів системи). Приватним варіантом аутентифікації є встановлення приналежності повідомлення конкретному автору. Криптографічне перетворення складається з двох етапів: прямого і зворотного. Пряме перетворення називають шифруванням (encrypt), зворотне – дешифруванням (decrypt). Процес передачі шифрованих повідомлень ілюструє (рисунок 1.4) [6].

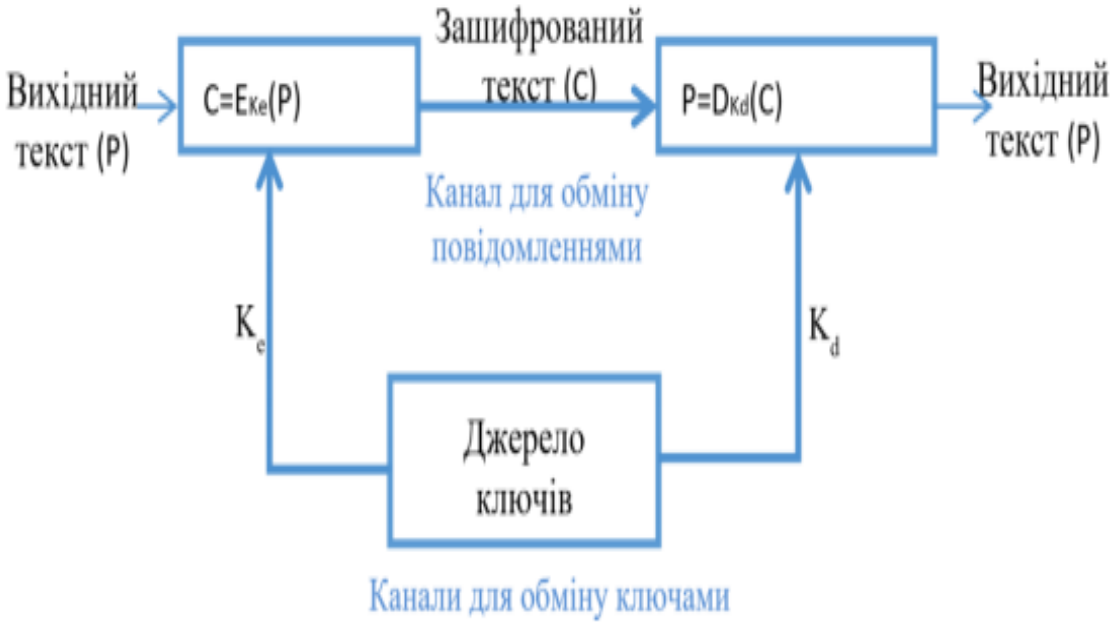


Рисунок 1.4 – Шифрування.

1.7 Надійність програмного забезпечення

Завдання та функції, які виконуються інформаційною системою, що працює в комп'ютерній мережі або поза нею, можливо виконати лише при поєднанні апаратних та програмних засобів. Отже при аналізі захищеності інформації необхідно враховувати єдину систему як апаратних так програмних засобів. Надійністю програмного забезпечення це його властивість відпрацьовувати свої специфічні завдання, не змінюючи характеристики в запроектованих межах в поточних умовах експлуатації.

Головні характеристики надійності програмного забезпечення це:

- безвідмовність;
- відновлюваність;
- коректність;
- стійкість.

Безвідмовність програм – це забезпечення тривалого в часі виконання функцій для обробки та захисту інформації в системі. Головною характеристикою є ймовірність стабільної та безвідмовної роботи програмного забезпечення в проектних умовах середовища на протязі тривалого періоду. Відмова програмного забезпечення це відхилення характеристик роботи від запроектованих вимог. Під зовнішнім середовищем слід розуміти характеристики вхідних даних та стан інформаційної системи. Період спостереження враховує весь час, що необхідний для виконання системою покладених функцій.

Безвідмовність програмного забезпечення можна оцінити середнім часом появи першої відмови під час неперервної роботи комплексу таких програм. При цьому мається на увазі, що всі апаратні засоби системи були в повністю працездатному у стані. Головна відмінність програмного забезпечення від апаратних засобів є те, що програми не зношуються і зламатися вони не можуть. Виходить, що характеристики функціонування програмного забезпечення

залежать тільки від якості його виготовлення, що зумовлене лише процесом розробки. Безвідмовність програм визначається його точністю і залежить від наявності чи відсутності в ньому помилок програмування, що були внесені під час його створення. Прояв помилок програм пов'язано ще і з тим, що в з часом можуть оброблятися дані, що до цього не зустрічалися в системі, які програма не в змозі правильно обробити. Тобто вхідні дані в значній мірі впливають на працездатність програмного забезпечення.

Іншою характеристикою є стійкість функціонування програмного забезпечення. Мається на увазі властивість програм зменшувати наслідки створених помилок і негативних впливів зовнішнього середовища чи протистояти їм. Стійкість забезпечується за допомогою введення різноманітних форм надмірності:

- дублюючі модулі програм;
- альтернативні програми для одних і тих же завдань;
- контроль за процесом виконання програм.

Відновлюваність це властивість програмного забезпечення, що характеризує можливість пристосовуватися до виявлення помилок і їх усунення.

Стійкість – властивість здійснювати необхідне перетворення інформації при збереженні вихідних рішень програми в межах допусків, встановлених специфікацією. Стійкість характеризує поведінку програми при впливі на неї таких чинників нестійкості, як помилки операторів, а також не виявлені помилки програми.

Коректність програмного забезпечення – властивість безпомилкової реалізації необхідного алгоритму при відсутності таких чинників, що заважають, як помилки вхідних даних, помилки операторів, збої і відмови.

У інтуїтивному значенні під коректністю розуміють властивості програми, що свідчать про відсутність в ній помилок, допущених розробником на різних етапах проектування (специфікації, проектування алгоритму і структур даних,

кодування). Коректність самої програми розуміють по відношенню до цілей, поставлених перед її розробкою.

В якості основної кількісної міри надійності що характеризує закономірність появи відмов у часу, прийнята ймовірність безвідмовної роботи. Імовірність безвідмовної роботи - це ймовірність того, що за певний час роботи і в заданих умовах експлуатації відмови не відбувається. Інший параметр це інтенсивність відмов, характерний графік якої зображено на рисунку 1.5. [7]

Основні показники надійності програмного забезпечення. Якщо розглядати відмови програмного забезпечення без урахування його відновлення, а також випадковий характер відмов у програмах, то основні показники надійності в цьому випадку є:

- ймовірність безвідмовної роботи програми p(t), що представляє собою ймовірність того, що помилки програми не виявляться в інтервалі часу (0, t);
- ймовірність відмови програми q(t) або ймовірність події відмови програм до моменту часу t;
- інтенсивність відмов програми λ (t);
- середнє напрацювання програми на відмову T, що є математичним очікуванням тимчасового інтервалу між послідовними відмовами.

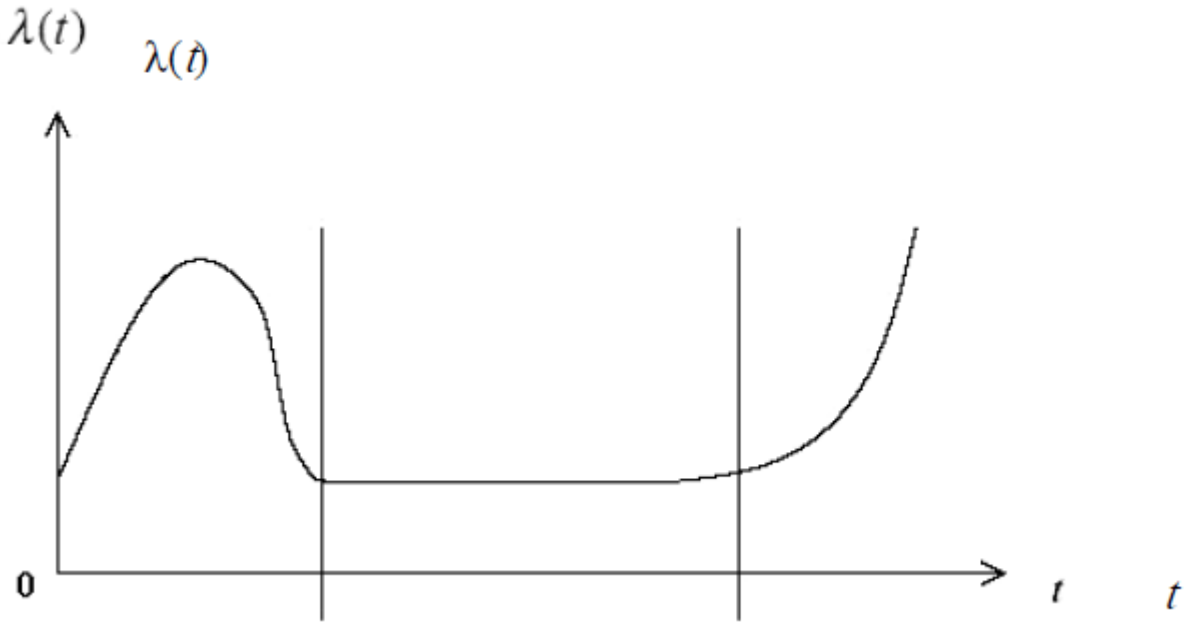


Рисунок 1.5 Залежність інтенсивності відмов від часу.

1.8 Завади в радіоканалах

В процесі проходження по каналу зв'язку сигнал піддається спотворенням. Необоротні спотворення форми сигналу в каналі є наслідком впливу перешкод. Завадами називаються будь-які випадкові впливи в каналі зв'язку на сигнал, що приводить до непоправного спотворення його форми.

Джерелами адитивних перешкод є фізичні явища здатні спотворити форму корисного сигналу. Серед джерел перешкод слід зазначити атмосферні (пов'язані з грозовими явищами), індустріальні (випромінювання електричних промислових і медичних приладів, систем автомобільного запалювання і т. п.), Космічні (випромінювання космічних об'єктів), перешкоди від сторонніх радіостанцій і т. п. У будь-якому каналі зв'язку типовими є перешкоди флуктуаційного характеру, пов'язані з електричними коливаннями шумового характеру, що виникають внаслідок електричних збурень на рівні молекулярних

і атомарних структур фізичних компонент функціональних блоків системи зв'язку.

За характером процесів адитивні перешкоди можна розділити:

на гладкі, безперервні, широкосмугові по спектру частот (теплові, флуктуаційні шуми);

імпульсні (хаотичні послідовності імпульсів) – перешкоди у вигляді одиночних імпульсів, що випливають один за іншим через такі проміжки часу. Перехідні процеси в каналі від одного імпульсу встигають практично завершитися до моменту приходу наступного імпульсу;

зосереджені по спектру випромінювань – сигнали сторонніх радіостанцій, які називають структурно-детермінованими;

різного роду прицільні перешкоди – перешкоди, створювані супротивником.

Мультиплікативні перешкоди найчастіше породжуються явищами, пов'язаними з особливими умовами поширення радіохвиль в атмосфері. Випадкові зміни навколишнього середовища (тропосфери, іоносфери) призводять до флуктуацій амплітуд і фаз канальних сигналів, багато променевість радіосигналів, що приходять в точку прийому.

У радіолокації і радіонавігації перешкоди прийнято ділити на активні – перешкоди від різних сторонніх джерел і пасивні перешкоди, що виникають в результаті віддзеркалення сигналів від сторонніх об'єктів. Крім того, розрізняють навмисні спеціально організовані противником і ненавмисні. Розглянуті вище шумові, індустріальні та взаємні перешкоди відносяться до активних та ненавмисних. Прицільні або навмисні перешкоди створюються противником за допомогою спеціальних засобів радіопротидії. Вони також можуть мати характер активних перешкод, створюваних радіопередавачами протидії, або пасивних перешкод, що виникають в результаті віддзеркалення від штучних об'єктів (до них можна віднести дипольні відбивачі, помилкові цілі, розкидану в повітрі металеву фольгу і ін.).

1.9 Завади в кабельних каналах

Структуровані кабельні системи (СКС), складають основу локальних мереж офісів і телекомунікаційну інфраструктуру будівель. Мережева завантаженість зростає на порядок кожні п'ять років, що вимагає заміни категорій СКС, які були обрані під поточні протоколи.

Головна завада для проходження сигналів к кабелях – між кабельні наводки.

Кабельний джгут або паралельно прокладені неекрановані кабелі можуть створювати неприпустимий рівень перешкод роботі високо швидкісних протоколів. Для обмеження їх впливу може знадобитися посилення специфікацій стандартів, зміна правил прокладки кабелів або прийняття інших заходів.

Між кабельні наводки можна визначити як небажані електромагнітні сигнали від прокладених в джгутах суміжних кабелів. Вимірюються як відношення сигналу, що подається на активну виту пару одного кабелю, до сигналу, наведеного в контрольній парі іншого кабелю.

Найбільший рівень наведень виникає між крученими парами, які мають однаковий крок скрутки. Всі провідники мають кольорове маркування, тому можна говорити, наприклад, про наводки між синіми або коричневими парами.

Якщо в джгуті більше двох кабелів, з'являється ефект сумарних наведень. Крім рівно крокових, і інші пари чинять негативний вплив.

Як і для перешкод між парами, між кабельні наведення можуть бути двох направленими і односпрямованим.

Сусідні кабелі вносять додаткові перешкоди, це знижує відношення сигнал-шум каналу і може вплинути на роботу протоколів. В результаті зменшуються динамічний і частотний діапазони каналу.

Важливо врахувати, що навіть після монтажу системи ефект між кабельних наведень неможливо виявити за допомогою стандартних вимірів.

Отже, для усунення проблеми доведеться посилювати параметри стандартів або вживати інші заходи.

Для мінімізації між кабельних наведень неекранованих кабелів їх слід розташовувати вільно і не паралельно. Заповнення коробів не повинно перевищувати 40%.

Поширена в даний час практика фіксації кабелів стяжками може бути скасована. Для вертикальних каналів ці рекомендації буде особливо складно виконати, оскільки фіксація в даному випадку запобігає надмірним поздовжнім навантаженням, що призводять до розтягування кабелів і погіршення їх параметрів.

Якщо всі ці заходи виявляться неефективним, залишається зменшення довжини каналів, яке дозволить досягти заданого відносини сигнал-шум шляхом зменшення загасання.

1.10 Ймовірність помилки в цифрових каналах

Головним параметром оцінки якості сигналу, що передається є співвідношення рівня сигналу до шуму (S/N). Де S –потужність сигналу а N-шуму. У дискретних каналах зв'язку прийнято використовувати відношення нормованих параметрів сигналу та шуму - E_b/N_o, де E_b – енергія одного біта сигналу. Нормований параметр це потужність сигналу помножена на час (T_b) за який передається один сигнальний біт. N_o – описує нормовану потужність шуму, яка виражається діленням потужності шуму N на частотну ширину W.

Час передачі біта обернено пропорційний швидкості передачі. $T_b=1/R$, де R – бітова швидкість передачі інформації.

$$\frac{E_b}{N_0} = \frac{S}{N}\left(\frac{W}{R}\right) ;$$

Головним критерієм оцінки якості в цифрових системахо зв'язку є характеристика ймовірності появи помилки (P_B) від велиичини E_b/N_o. На рис. 1.6 зображено характерна залежність цього критерія (для $E_b/N_o \geq X_0$, $P_B \leq P_0$). Безрозмірне відношення E_b/N_o – це стандартна якісна міра систем цифрового зв'язку. Висновок - критерій E_b/N_o є метрикою порівняння якості різних систем передачі інформації. [8]

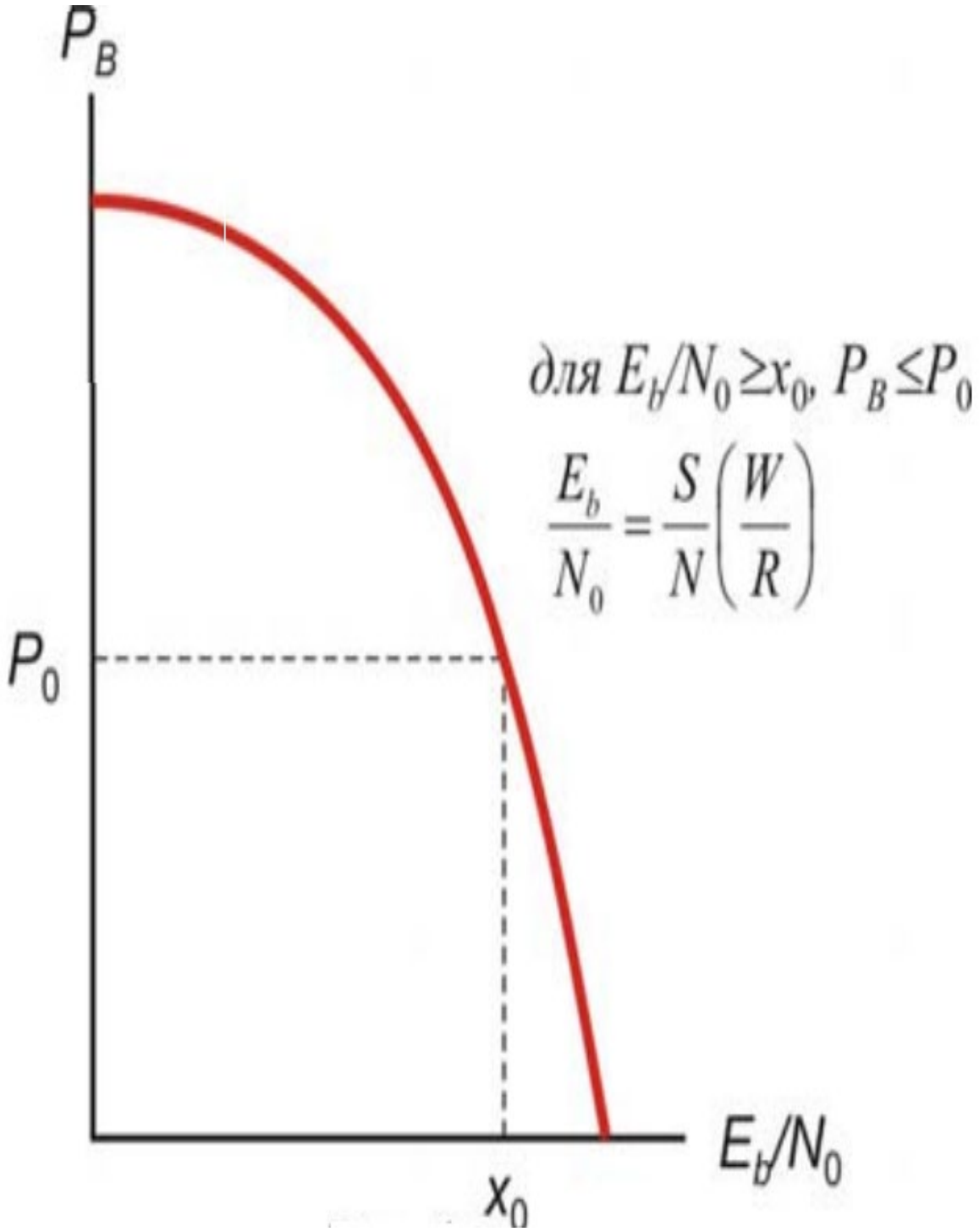


Рисунок 1.6 Ймовірність помилки в цифровому каналі.

2. КОДУВАННЯ ІНФОРМАЦІЇ ТА ІНФОРМАЦІЙНА БЕЗПЕКА

Завадостійке кодування для цифрового каналу з завадами побудоване на наступному твердженні: при проходженні нулів та одиниць з швидкістю меншою за пропускну здатність канал, можна знайти код, який зменшить ймовірність помилки до потрібної величини.

Це можливо за рахунок збільшення надмірності. Використовується такий код передачі інформації, у якого задіяні не всі можливі комбінації, а лише частка з них. Це підвищує стійкість перед завадами каналу. Коригувальні властивості кодів залежать від правил побудови цих кодів і параметрів коду (тривалості символів, числа розрядів, надмірності і ін.).

В даний час найбільша увага приділяється двійковим рівномірним коригуючим кодам. Вони мають хороші коригуючі властивості і їх реалізація порівняно проста.

Найбільш часто застосовуються блокові коди. При передачі інформації блоковими кодами ця інформація подається у виді окремих блоків однакової довжини. Кодування та декодування кожного з блоків виконується окремо.

Рівномірним називають такий блоковий код у якого кількість біт є рівною для всіх символів повідомлення.

Розрізняють роздільні і нероздільні блокові коди.

При кодуванні роздільними кодами кодові послідовності будуються з двох частин: інформаційної та перевірочної. Інформаційні та перевірочні біти у кодових комбінаціях роздільного коду знаходяться на тих же позиціях.

При кодуванні нероздільними кодами розділити символи послідовності на інформаційні та перевірочні неможливо.

Безперервними називаються такі коди, в яких введення надлишкових символів в кодовані послідовності інформаційних символів здійснюється безперервно, без поділу її на незалежні блоки. Безперервні коди також можуть бути роздільними і нероздільними.

В даний час незалежно один від одного існують і розвиваються дві специфічні гілки перетворення двійковій інформації:

- кодування;
- криптографічне шифрування.

Ці гілки базуються на різних підходах: кодування вживає, головним чином,

чисто алгебраїчний підхід. Шифрування в залежності від класу процедур – різні методи – від комбінаторних до таких, як еліптичні функції, арифметику в залишкових класах, обчислення в полі Галуа і ін. Як при кодуванні, так і при шифрування мова йде про процедури (алгоритми, правила, формули перетворення однієї двійкової комбінації X в іншу Y, тобто

$$Y = F(X).$$

Для опису такої процедури використовується матричний підхід. Двійкові послідовності представляються у вигляді поліномів, а окремі кроки (раунди) шифрування – матричними операціями. Елементами матриць виступають окремі байти інформаційної послідовності або ключа, що дозволяє суттєво полегшити програмну реалізацію відповідних процедур.

У каналі зв'язку повідомлення, складене з символів (букв) однієї мови (наприклад української) може перетворюватися в символи іншого алфавіту. Правило, яке описує однозначна відповідність букв різних алфавітів при такому перетворенні, називається кодом, а сам процес перетворення – кодуванням. Зворотне перетворення називають декодуванням. На принципах кодування інформації засновано більшість комп'ютерних операцій: запис символів алфавіту за допомогою таблиць кодування, запис чисел в двійковому коді, перевірка правильності обчислень, шифрування і дешифрування інформації. Використання принципів кодування для перевірки обчислень і шифрування інформації, а також кодування доступу в комп'ютерну систему є основою інформаційної безпеки.

2.1 Коди з корекцією помилок.

Коригуючими називаються коди які дозволяють виявляти і виправляти помилки. Ідею коригувальних кодів можна уявити за допомогою N-мірного куба. Візьмемо тривимірний куб (рисунок 2.1), довжина ребер в якому дорівнює одиниці. Вершини такого куба відображають двійкові коди. Мінімальна відстань між вершинами визначається мінімальною кількістю ребер, що знаходяться між вершинами. Ця відстань називається кодовою (або хеммінговим) і позначається буквою d. [9]

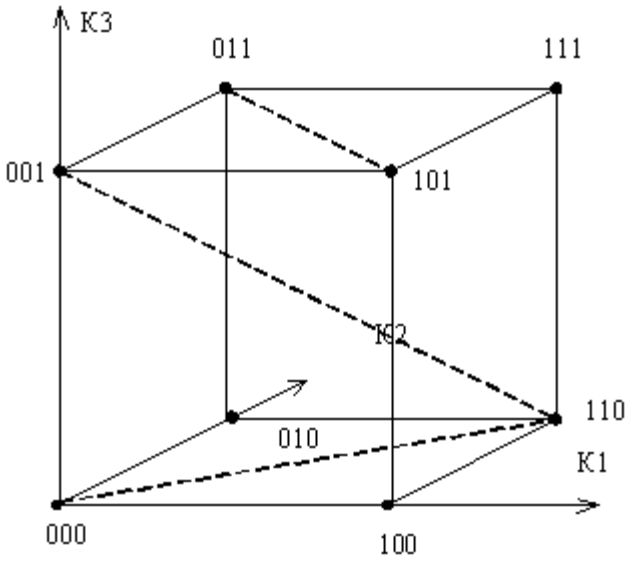


Рис.2.1. Двійковий код в вигляді куба.

Кодова відстань – це найменш можлива кількість елементів, в яких кодові комбінації відрізняються одна від одної. Скористаємся прикладом визначення кодової відстані. Достатньо по бітно додати кодові комбінації по модулю 2.

$$10110101111 + 11001010101 = 01111111010$$

Отже отримали результат відстань складає d = 8.

Для коду з 3 символами маємо вісім можливих кодових комбінацій, що можна розмістити на вершинах тривимірного куба. У такого коду відстань d = 1, і для передачі використовуються всі вісім кодових комбінацій: 000, 001, ..., 111. Такий код не є завадостійким і не може виправити помилку.

Виберемо комбінації з кодовою відстанню d = 2, наприклад: 000, 110, 101, 011. З таким параметром код дозволить просто виявляти одинарні помилки. Ці комбінації є дозволеними, тими що призначені для інформації. Всі інші комбінації: 001, 010, 100, 111 – є забороненими.

Будь-які одинарні помилки призводить до того, що дозволена комбінація переходить в найближчу заборонену комбінацію. Отримавши заборонену комбінацію, ми виявимо помилку.

Беремо d = 3.

Приклад: Дозволені: 000, 111. Заборонені: 001, 010, 011, 100, 101, 110.

Такий код може виправити одну одиночну помилку або виявити дві помилки. Таким чином, збільшуючи кодову відстань можна збільшити стійкість коду. У загальному випадку кодова відстань визначається за формулою d = t + l + 1, де t – число допустимих помилок, l – число виявлених помилок. Зазвичай l > t.

В основному коригувальні коди є лінійними. У яких контрольні символи утворюються шляхом лінійного поєднання інформаційних символів. До того ж, коригувальні коди є груповими – такими, що мають одну головну операцію. Необхідна умова замкнутості – результат додавання елементів групи також належить цій групі. Число розрядів в групі є сталим. Цій умові задовольняє операція порозрядного додавання по модулю 2. У групі, також, має бути присутнім нульовий елемент.

Приклади:

1101, 1110, 0111, 1011 – не є групою, тому що немає нульового елемента.

0000 1101, 1110, 0111 – також не група, не виконана умова замкнутості: (1101 + 1110 = 0011).

000, 001, 010, 011, 100, 101, 110, 111 – група.

000, 001, 010, 111 – підгрупа.

Коригувальні коди формуються шляхом додавання до початкової інформаційної комбінації довжиною=m перевірочних символів довжиною=k. У підсумку передаються n = m + k символів. При цьому коригувальні коди називаються (n, m) кодами. Як можна визначити необхідну кількість контрольних символів? Для побудови коду здатного виявляти і виправляти одиночну помилку кількість інформаційних символів буде: [10]

$$m = n - k = n - \log(n + 1).$$

2.2 Канали з помилкою

Звідки взагалі беруться помилки, які ми збираємося виправляти. Припустимо маємо завдання. Потрібно передати кілька блоків даних, кожен з яких кодується ланцюжком двійкових цифр. Отримана послідовність нулів і одиниць передається через канал зв'язку. Але так склалося, що реальні канали зв'язку часто схильні до помилок. Взагалі кажучи, помилки можуть бути різних видів – може з'явитися зайва цифра або якась прірва. Але ми будемо розглядати тільки ситуації, коли в каналі можливі лише заміни нуля на одиницю і навпаки. Причому знову ж для простоти будемо вважати такі заміни рівно ймовірними.

Помилка – це малоймовірна подія (а інакше навіщо нам такий канал взагалі, де одні помилки?), А значить, ймовірність двох помилок менше, а трьох уже зовсім мала. Ми можемо вибрати для себе деяку прийнятну величину ймовірності. Це дозволить нам сказати, що в каналі можливо не більше дозволеного числа помилок. Це буде характеристикою каналу зв'язку.

При передачі по каналу зв'язку інформація піддається впливу різного роду перешкод: широко відомі флуктуаційні, гармонійні і імпульсні перешкоди.

Флуктуаційна завада це напруга, мінлива у часі випадковим чином. Причина появи її – теплові шуми лінії чи елементів апаратури. Гармонійна

перешкода наближено описується синусоїдальним коливанням. Ці перешкоди виникають, як правило, в самій апаратурі через паразитне проникнення в канал різних несучих коливань.

Імпульсна завада – пікове значення якої можна порівняти з амплітудою корисного сигналу або перевищує її. Імпульсні перешкоди зазвичай з'являються пачками, по кілька перешкод в пачці. Характер процесу появи пачок в часі і окремих перешкод всередині однієї пачки істотно змінюється від каналу до каналу і навіть в одному каналі в різні періоди часу .

В результаті дії перешкод в каналі зв'язку інформація, передана по цьому каналу, спотворюється, прийняте повідомлення буде відрізнятися від переданого, повідомлення приймається з помилкою. Виникнення помилок – випадковий процес, і передбачити появу їх заздалегідь, до експерименту, можна тільки статистично, вказуючи ймовірність того, що помилка або станеться, або ні. При цьому ймовірність помилки може надаватися незалежно від значення переданого елементарного символу 0 або 1. Серед неправильно прийнятих сигналів однаково часто зустрічаються як 1, так і 0. Канал зв'язку з такими помилками називається симетричним каналом.

2.3 Поняття про інформаційну надлишковість.

Надлишковість системи – перевищення будь-яких параметрів системи в порівнянні з їх мінімальними значеннями, необхідними для вирішення певної задачі.

При розгляді системи на технічному рівні розрізняють:

- Сигнальну – перевищення обсягу сигналів;
- Структурну – пов'язану з мірою складності системи; На абстрактному рівні виділяють:
- Інформаційну, коли система здатна переробляти більше, ніж необхідно, кількість інформації;

- Алгоритмічну, що виражається в надмірній складності алгоритму функціонування системи.

Крім цього, розрізняють штучну і природну надмірності інформаційної системи. Скорочення природної інформаційної надмірності дозволяє спростити систему. Введення штучної надмірності в систему підвищує її стійкість (точність), надійність. Покращує функціональні характеристики системи, але збільшує узагальнені витрати на її створення. Для оцінки виграшу, який дає введення надмірності, використовується критерій функціональної ефективності системи, що зіставляє як ступінь поліпшення функціональних параметрів системи, так і збільшення узагальнених витрат.

Стиснення даних – це спосіб зменшити або усунути небажану надмірність, в той час як контрольні суми – це спосіб додавання бажаної надмірності з метою виявлення помилок при обміні даними по каналу обмеженої місткості.

Традиційно поняття інформаційної надлишковості (ІН) найчастіше пов'язують із використанням завадо захищених кодів для передачі і зберігання інформації. За К. Шенноном рівень інформаційної надлишковості (J) визначається відносним перевищенням максимально можливої ентропії H_{max} над реальною ентропією H_r конкретного джерела інформації при використанні певного способу кодування.

$$J = 1 - \frac{H_r}{H_{max}} = H_{max} - H_r$$

де $H_{max} = \log_2 N$

$$H_r = \sum_{i=1}^{N} p_i \log p_i$$

N – кількість можливих повідомлень, p_i – ймовірність помилки і-го повідомлення. [7]

2.4. Групові коди.

Лінійним називається код, в якому перевірочні символи являють собою лінійні комбінації інформаційних. Груповим називається код, який використовує алгебраїчні операції додавання по модулю два.

Характеристика лінійного коду: сума (по модулю 2) векторів лінійного коду дає новий вектор, що також відноситься до цьому коду. Характеристика групового коду: мінімальна дистанція між кодовими векторами дорівнює мінімальній вазі ненульових векторів коду. Вага кодового вектора рівна числу одиниць в кодової комбінації.

Групові коди зазвичай задають за допомогою матриць, розмірності k і n.

m – кількість інформаційних біт, k – кількість перевірочних біт.

$$n = k + m.$$

Будь-яке кодове слово V групового коду (n, k) можна отримати множенням вектора U, що представляє інформаційне слово, на твірну матрицю G розмірності [k,n]:

$$V = U \times G$$

Прийняте слово можна перевірити на відсутність помилок множенням його на транспоновану перевірочну матрицю H. Якщо слово прийнято без помилок, результат множення нульовий: $VH^T = 0$. Перевірочна матриця пов'язана з твірною матрицею співвідношенням

$$G \times H^T = 0$$

Твірна матриця вибирається неоднозначно. Для спрощення кодування і декодування зручно використовувати твірну матрицю, складену з двох матриць: одиничної матриці розмірності [k,k] і дописує справа матриці-доповнення, або контрольної під матриці, розмірності [k,m].

2.5 Код Хеммінга

Код Хеммінга, що є груповим (n, m) кодом з мінімальною відстанню d = 3 дозволяє виявляти і виправляти одноразові помилки. Число контрольних символів k = n - m можна визначити за нерівністю Хеммінга для одноразової помилки.

Побудова кодів Хеммінга. Будуємо послідовності коду Хеммінга. Членами цих послідовностей є числа 1, 2, 4, 8, 16, ..., 2^{k-1} тобто ступеня двійки, причому $2^{k-1} <= n$, а $2^k > = n + 1$. Члени b_i набору b_1 ... b_n, у яких індекс і належить множині (1, 2, 4, 8, 16, ..., 2^{k-1}), називаються контрольними членами, інші – інформаційними членами. Легко бачити, що контрольних членів буде k, а інформаційних n - k = m. Сформулюємо тепер правило побудови набору b_1 ... b_n за шаблоном a_1 ... a_m. Спочатку визначаються інформаційні члени

$b_3 = a_1$, $b_5 = a_2$, $b_6 = a_3$

Таким чином, набор інформаційних членів, розміщених початковим порядком, збігається с набором $a_1 \ldots a_k$. Обчислюємо контрольні біти

$$b_1 = (b_3 + b_5 + b_7 + \ldots) \bmod 2,$$

$$b_2 = (b_3 + b_6 + b_7 + \ldots) \bmod 2,$$

$$\ldots..$$

Приклад:

Нехай m = 4. Тоді n = 7 і k = 3. Відповідно до етапу 1 отримуємо самокорегуючий код. Результат представлений в таблиці, в якій контрольні члени позначені зірочкою.

У цій таблиці спочатку в стовпці з номерами 3, 5, 6 і 7 (інформаційні члени) вписуються зверху вниз набори 0000 ..., 1111. Потім за формулами заповнюються стовпці з номерами 1, 2 і 4.

$$b_1 = (b_3 + b_5 + b_7) \bmod 2,$$
$$b_2 = (b_3 + b_6 + b_7) \bmod 2,$$
$$\ldots\ldots.$$

Нехай b = 0110011 і в ньому джерело перешкод спотворив 5-й член (S = 5). Тоді c = 0110111. Обчислимо номер члена, в якому сталася помилка. Ми маємо

$$T_1 = (c_1 + c_3 + c_5 + c_7) \bmod 2 = (0 + 1 + 1 + 1) \bmod 2 = 1,$$
$$T_2 = (c_2 + c_3 + c_6 + c_7) \bmod 2 = (1 + 1 + 1 + 1) \bmod 2 = 0,$$
$$T_4 = (c_4 + c_5 + c_6 + c_7) \bmod 2 = (0 + 1 + 1 + 1) \bmod 2 = 1;$$

Отже, T = 101.

2.6. Циклічні коди

Узагальненням коду Хеммінга є циклічні коди. Це коди з широким вибором довжини і можливостей виправлення помилок. Циклічні коди характеризуються поліномом g(x) ступеня n-m,

$$g(x) = 1 + g_1x + g_2x^2 + ... + g_{n-m}\, x^{n-m}$$

g(x) – твірний многочлен циклічного коду. Якщо многочлен g(x) ступеня n-m є дільником многочлена x^n+1, то код, який використовує g(x) є лінійним циклічним (n, m) кодом. Число циклічних n-розрядних кодів дорівнює числу дільників багаточлена x^n+1. Розглянемо алгебру циклічних кодів. Припустимо, необхідно перемножити три многочлена

$$(x^3 + x^2 + 1) \bullet (x^3 + x + 1) \bullet (x + 1).$$

Дії виробляються також як і в звичайній алгебрі, тільки складання проводиться по модулю 2.

x^3+x^2+1
x^3+x+1
x^3+x^2+0+1
x^4+x^3+0+x
$x^6+x^5+0+x^3$
$x^6+x^5+x^4+x^3+x^2+x+1$
$x+1$
$x^6+x^5+x^4+x^3+x^2+x+1$
$x^7+x^6+x^5+x^4+x^3+x^2+x$
$x^7+0+0+0+0+0+0+1=x^7+1$

1101 * 1011
1101
1101
1111111*11
1111111
10000001 = x^7+1

Рисунок 2.3 – Розрахунок циклічного коду.

При розподілі операція віднімання замінюється операцією додавання по модулю 2. Наприклад, необхідно розділити многочлен сьомий ступеня на многочлен третього ступеня

$$(x^7 + x^5 + x^4 + x + 1) / (x^3 + x^2 + 1).$$

Операція ділення може бути проведена у вигляді многочленів або у вигляді двійкових кодів.

$x^7+0+x^5+x^4+0+0+x+1$ | x^3+x^2+1
$x^7+x^6+0+x^4$ | x^4+x^3+1
x^6+x^5+0+0
x^6+x^5+0+x
$x^3+0+x+1$
x^3+x^2+0+1
x^2+x

10110011 | 1101
1101 | 11001
1100
1101
1011
1101
110

Рисунок 2.4 – Розрахунок циклічного коду.

Циклічний код одержують у такий спосіб: заданий многочлен h(x) (число, яке треба закодувати) спочатку множиться на одночлен x^{n-m}, а потім ділиться на твірний многочлен g(x).

В результаті отримаємо:

$$\frac{h(x)x^{n-m}}{g(x)} = Q(x) + \frac{R(x)}{g(x)}, \text{ або}$$

$$F(x) = Q(x)g(x) = x^{n-m}h(x) + R(x)$$

Таким чином, циклічний код можна побудувати множенням кодової комбінації h(x) на одночлен x^{n-m} і додаванням до результату залишку R(x). При декодуванні, отриману закодовану комбінацію треба розділити на g(x). Наявність залишку ділення вказує на наявність помилки. За заданою довжиною кодової комбінації n визначають необхідне число контрольних символів.

$$n = m + k$$

де, m – кількість інформаційних символів, k – кількість контрольних символів.

Співвідношення значень n, m, k можна визначити по таблиці

Таблиця 2.2 – Співвідношення значень n, m, k.

n	3	5	6	7	9…15	17…31	33…63	65…127
m	1	2	3	4	5…11	12…26	27…57	28…120
k	2	3	3	3	4	5	6	7

Потім вибирають найкоротший многочлен зі ступенем, що дорівнює числу контрольних символів; його і приймають за твірний поліном. Приклад: Нехай потрібно закодувати комбінацію 1101, що відповідає $h(x) = x^3 + x^2 + 1$. За формулою визначаємо число контрольованих символів k=3.

Візьмемо многочлен $g(x) = x^3 + x + 1$, тобто 1011.

$$h(x) \bullet x^k = (x^3 + x^2 + 1) \bullet x^3 = x^6 + x^5 + x3 \Rightarrow 11010000$$

Розділимо отримане на твірний поліном g(x)

$$\frac{h(x)x^k}{g(x)} = \frac{x^6 + x^5 + x^3}{x^3 + x + 1} = x^3 + x^2 + x + 1 \frac{1}{x^3 + x + 1} \Rightarrow 1111 + \frac{001}{1011}$$

При діленні необхідно враховувати, що віднімання проводиться по модулю 2. Залишок підсумовуємо з $h(x)x^k$. В результаті отримаємо закодоване повідомлення:

$$F(x) = (x^3 + x^2 + 1) \bullet (x^3 + x + 1) = (x^3 + x^2 + 1) \bullet x^3 + 1 => 1101001$$

В отриманій кодової комбінації циклічного коду інформаційні символи h(x) = 1101, а контрольні k(x) = 001. Закодоване повідомлення ділиться на твірний поліном g(x) без залишку. Повідомлення, яке закодовано, є однією з комбінацій 4-розрядного коду, так як вся група повідомлень містить N = 16 повідомлень. Це означає, що якщо всі повідомлення передаються в закодованому вигляді, то кожне з них необхідно кодувати так само, як і комбінацію h(x) = 1101. Однак виконувати додаткові 15 розрахунків (а в загальному випадку 2^{n-m}-1 розрахунок) немає необхідності . Це можна зробити простіше, шляхом складання твірної матриці. Твірна матриця складається на основі одиничної транспонованої матриці, до якої справа дописується матриця доповнень:

$$H_{n,k} = \| I_k, C_{n,r} \|$$

Матриця доповнень виходить із залишків від ділення одиниці з нулями на твірний многочлен g (x). Комбінації одиниць з нулями є вектори помилок: 00 ... 01, 00 ... 10, 10 ... 00. Кожному вектору помилок буде відповідати свій залишок (розпізнавальний знак):

$$H_{7,4} = \left\| \begin{array}{cccc|ccc} 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 1 \end{array} \right\|$$

Отримано 4 комбінації циклічного коду, тобто стільки, скільки інформаційних розрядів, а так як в 4-розрядному двійковому коді всього 16 комбінацій, то решта 11 ненульових комбінацій знаходяться підсумовуванням за

модулем 2 всіляких поєднань рядків твірної матриці. Наприклад, необхідно з вихідних кодів 1101 і 1010 отримати циклічні коди. Вони виходять підсумовуванням відповідних рядків твірної матриці:

$$1 + 3 + 4 = 1101001;$$

$$2 + 4 = 1010011.$$

Розглянемо алгоритм декодування інформації при використанні циклічного коду. Перший крок це поліномне ділення прийнятої інформаційної комбінації на твірний поліном. У випадку рівності залишку нулю можна стверджувати, що інформацію отримано без помилок. Ненульовий залишок ділення - прийнята комбінація містить помилки. Порівнюючи ненульовий залишок з рядками перевірочної матриці знаходять номер біта з помилкою.

Загальний виправлення помилок наступний:

Комбінацію отриманих бітів поліномно ділимо на твірний многочлен. Якщо ступень поліному залишку менше кількості допустимих помилок то достатньо отриману комбінацію додати операцією HOR до залишку ділення. Результат додавання і буде правильною комбінацією.

Якщо ступень поліному залишку більше кількості допустимих помилок то перед діленням на твірний поліном проводиться побітний зсув інформації вліво. Потім проводиться ділення на твірний поліном. Якщо отримали залишок потрібного ступеня, то маємо можливість виправити помилку додаванням (HOR).

Робимо зсув уже виправленого блока на біт вправо, і дальші по циклу аж до виправлення всіх помилок. [10]

2.7 Коди, найбільш використовувані в сучасних телекомунікаційних мережах

Перспективним напрямком сучасного завадо захисного кодування є використання турбокодів. Турбокоди утворюються при паралельному каскадуванні кількох простих компонентних кодів, що розділені пристроєм перемежувачем. По цій причині часто такі турбокоди називають паралельними каскадними згорточними кодами. Якщо в ролі компонентних кодів використовуються стандартні блокові коди (наприклад коди Хеммінга, циклічні коди, і т.п.), то такі коди називають паралельними каскадними блоковими кодами.

У процесі кодування інформаційна послідовність ділиться на блоки заданої довжини символів. Після цього сформована послідовність поступає на систематичний вхід кодера, а також паралельно на декілька гілок, що складаються з послідовних з'єднань перемежувача і компонентного кодера. Автори турбокодів, як компонентних запропонували використовувати рекурсивні систематичні згорточні коди. Саме використання систематичних згорточних кодів (РССК) при інших рівних умовах гарантує турбокоду найкращі характеристики.

Наступною важливою ланкою турбокодерів є перемежувач. За рахунок перемежувача процес формування кодових комбінацій турбокоду досить близький до випадкового. Тому турбокод з великим розміром блоку можна характеризувати як довгий випадковий код, а саме такі коди і потрібні для передачі інформації зі швидкостями, максимально близькими до пропускної здатності каналу зв'язку.

Висока ефективність використання турбоходів багато в чому зобов'язана розробленим для них алгоритмам декодування. У першу чергу відзначимо, що в основі декодування будь-яких коригувальних кодів лежить порівняння імовірнісних характеристик різних кодових слів.

Незабаром після винаходу турбокодів міжнародні організації по стандартизації засобів телекомунікацій почали проводити роботи по стандартизації параметрів турбокодів як методів канального кодування в системах передачі телеметричної інформації з космічних апаратів.

Для кодування в системах рухомого радіозв'язку третього покоління визначені наступні параметри турбокодів: швидкість коду 1/3, довжина інформаційного блоку (40 ... 5114) символів. [11]

Сигнально-кодові конструкції. Сигнально-кодові конструкції широко використовуються в структурах модемів для комп'ютерного обміну по провідним каналам тональної частоти в мережах Інтернет.

3 ПРОЦЕДУРИ ШИФРУВАННЯ ЯК ЗАСІБ ЗАХИСТУ ВІД НЕБАЖАНОГО ДОСТУПУ.

Шифрування – це засіб забезпечення конфіденційності даних, що зберігаються в пам'яті комп'ютера або передаються по дротовій чи бездротовій мережі.

Шифрування є наріжним каменем усіх служб інформаційної безпеки, будь то система аутентифікації або авторизації, захищений канал або засоби безпечного зберігання даних.

Алгоритм шифрування, який перетворює інформацію, що передається, з початкового простого для розуміння виду в зашифрований, має бути доповнений алгоритмом дешифрування, який застосовується да шифрованої інформації та повертає її в початковий вид.

Пара процедур – шифрування і дешифрування – називається криптосистемою. Зазвичай криптосистема передбачає наявність спеціального параметра – секретного ключа. Криптосистема вважається розкритою, якщо знайдено алгоритм дешифрування, що дає можливість підібрати ключ за невеликий час. Складність алгоритму розкриття є однією з важливих характеристик криптосистеми і називається крипто стійкістю.

Шифрування даних – це методи захисту будь-якої інформації від несанкціонованого доступу, перегляду, а також її використання, засновані на перетворенні даних в зашифрований формат.

Залежно від використовуваного алгоритму шифрування даних, методи перетворення поділяються за гарантованої або тимчасової крипто стійкості.

Шифрування даних, в залежності від структури ключів використовуваних при шифруванні діляться на:

Симетричне шифрування: сторонній особі може бути відомий алгоритм шифрування, але невідома невелика частина секретної інформації – ключа, однакового для відправника і одержувача повідомлення.

Асиметричне шифрування: сторонній особі може бути відомий алгоритм шифрування, і, можливо, відкритий ключ, але невідомий закритий ключ, відомий тільки одержувачу повідомлення.

Постулатом для симетричних криптосистем є таємність ключа. Симетричні крипто схеми в даний час прийнято поділяти на блокові і потокові.

Блокові криптосистеми розбивають текст повідомлення (файлу, документа і т.п.) на окремі блоки і потім здійснюють перетворення цих блоків з використанням ключа.

Потокові криптосистеми працюють трохи інакше. На основі ключа системи виробляють якусь послідовність – так звана вихідна гамма, яка потім накладається на текст повідомлення. Таким чином, перетворення тексту здійснюється як би потоком в міру вироблення гами. Як правило, використовується для потреб військових, шифрування в засобах зв'язку і т.п.

Однак не слід вважати цей поділ закостенілим. Так, наприклад, при використанні деяких хитрощів отримують з блочного шифру – потоковий і навпаки. А, наприклад, блоковий шифр з розміром вихідного блоку 8 біт (один символ) можна вважати потоковим.

3.1 Криптосистема DES та AES

DES (Data Encryption Standard) був розроблений ще в 1976 і над широко застосування. Шифрування за цим алгоритмом виконується над блоками. Перший крок це в разі наявності аналогової інформації – її перетворення в цифрову. Можна використати любий з відомих математичних алгоритмів оцифровки аналогового сигналу. На вхід блоку шифрування подають пакет даних. Розмір пакета незмінний і складає 64 біта. Цей блок ділиться на два рівних пів блока по 32 біта (L - ліву та R- праву). Під час шифрування пів блоки постійно змінюють свою позицію. Крім того циклічно права частина замінюється на нову, що отримана в результаті математичних дій над

попередніми частинами. Головним в шифруванні є певна послідовність операцій заміни та перестановки, що виконується циклічно над одною з частин. Ключ шифрування задає цю послідовність. Довжина ключа становить 64 біта, з яких 56 визначають правила шифрування. Інші 8 є перевірочними для перевірки правильності передавання ключа на приймач.

Слід зазначити, що зафіксовані вдалі результати по злому ключа шифрування, а отже і доступу до зашифрованої інформації. Але ці випадки поодинокі. Незважаючи на це процедура шифрування DES продовжують використовувати в інформаційних системах. Постійно проводять модернізації цього алгоритму шифрування. Наприклад збільшення довжини ключа з 64 до 112 біт. Це так званий потрійний DES. Слід зазначити що при цьому падає продуктивність.

Отже алгоритм DES має наступні характеристики

- довжина блоку – 64 біти;
- кількість раундів – 16;
- розмір ключа – 56 бітів;
- розмір кожного з під ключів $k_1, k_2, \ldots, k_{16}$ – 48 бітів. [6]

Advanced Encryption Standard (AES) – симетричний алгоритм блочного шифрування. Цей алгоритм витісняє вже застарілий Data Encryption Standard (DES), що не може більше повноцінно захищати мережі, що ускладнилися в наш час. Цей алгоритм, крім абревіатури AES, іноді називають ще Rijndael – це анаграма з частин імен бельгійських програмістів Joan Daemen і Vinent Rijmen, які розробили AES. AES і Rijndael – це трохи різні алгоритми шифрування, оскільки AES має сталий розмір блоку в 128 біт і сталі розміри розміри ключів в 128, 192 і 256 біт, в той час як для Rijndael можуть бути задані розміри блоку і ключа будь якого розміру, від мінімуму в 32 біт до максимуму в 256 біт.

AES повинен постійно перевірятися і поліпшуватися, з метою надійного зберігання зашифрованих даних. Інформація повинна бути захищена за допомогою AES з довжиною ключів 128, 192 і 256 біт. Для інформації, визначеної як особливо секретна, ця довжина становить 192 або 256 біт. Суть AES в тому, що будь-яка спроба отримати доступ до захищених даних – тобто підбір всіх можливих ключів – в часі буде тривати тисячі років.

Advanced Encryption Standard ключ довжиною в 128 біт є достатньо надійним захисти інформацію проти атаки, тобто з суто математичної точки зору підібрати один правильний пароль з усіх можливих є важко майже неможливим завданням.

3.2 Алгоритм шифрування криптосистеми

Структура алгоритму DES показана на рис. 3.1.

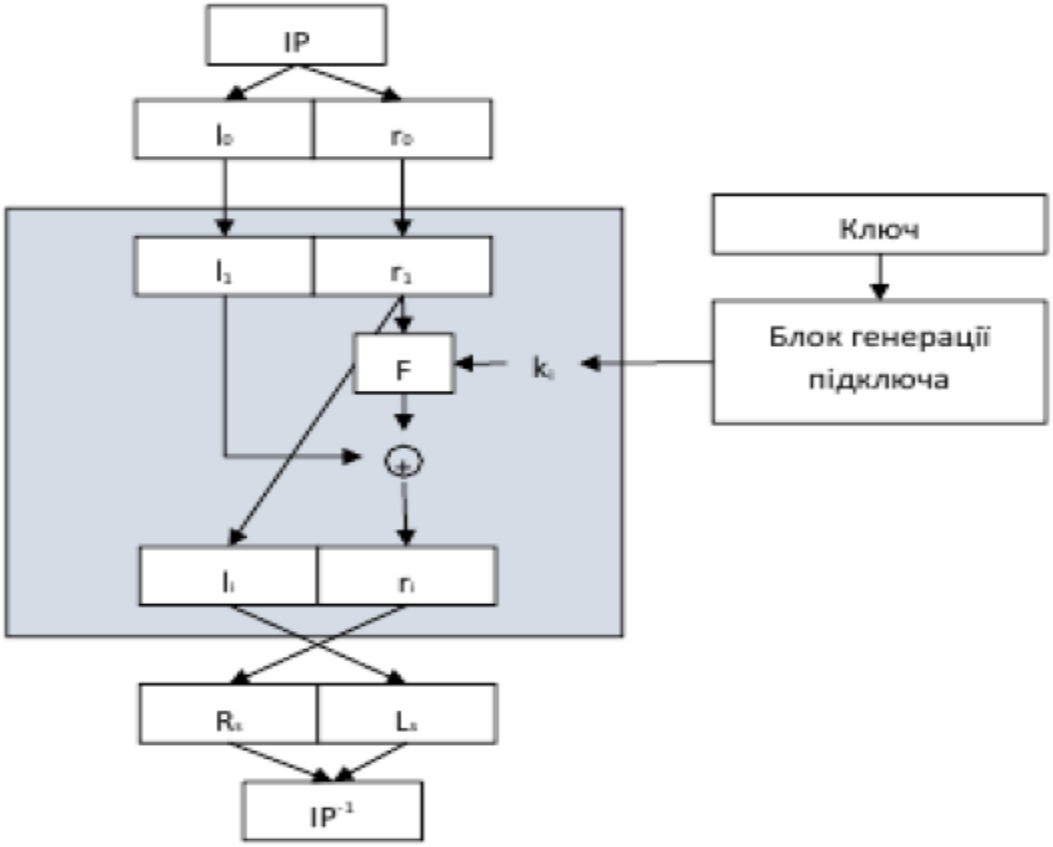


Рисунок 3.1 – Структура алгоритму DES.

Алгоритм DES (рисунок 3.1) описується за допомогою трьох етапів:

Спочатку вхідний блок, довжина якого 64 біта ділиться на дві частини (IP - l_0, r_0),

де IP - вхідний блок,

l_0 та r_0 відповідно лівий та правий 32 бітні пів блоки.

Наступним етапом алгоритму є 16 раундів операцій:

1. За допомогою функції F змінюємо правий пів блок з застосування поточного ключа, який становить 48 біт та обирається з повного ключа

(вихід блоку генерації ключа). Над правим пів блоком проводим операції перестановки та заміни за допомогою функції F та отриманого ключа.

2. Отриманий результат додаємо операцією XOR до лівого пів блоку, а суму записується на місце лівого пів блоку. Відповідно лівий пів блок записується на місце правого

3. Виконується 16 циклів таких операцій.

Цикл можна описати наступним виразом:

$$l_i = r_i \text{ , } r_i = l_i \, F(r_i\text{-}1, k_i)$$

де k_i – 48 бітний ключ циклу, підрядок 56-бітного вихідного ключа, F – функція шифрування.

Ці операції перестановки забезпечують значний рівень перемішування інформаційного повідомлення. [6]

3.3 Характерні параметри блокових шифрів

Алгоритми шифрування DES та AES відносяться до блочних.

Довжина блока алгоритму DES складає 64 біта. Під час виконання алгоритму блок ділиться на 2 під блока по 32 біта кожний. Над цими під блоками виконується 16 кіл циклу простих двійкових арифметичних дій з ключем. Довжина початкового ключа складає 64 біт. З яких 56 біт є інформаційними. Інші 8 біт – перевірка на парність (кодування ключа). Під час кожного кола циклу з 56 бітного ключа вибирається 48 біт на виконання математичних операцій з блоком даних. Для наступного кола циклу проходить зміщення ключа і обирається інший блок ключа. Під час шифрування довжина блоку не змінюється. Довжина зашифрованих даних складає 64 біта.

Під час виконання алгоритму блок формує матрицю з 4 рядками та 32 колонками, по 32 біта в кожній колонці. Над цими під блоками виконується 10, 12 або 14 (в залежності від довжини ключа 128, 192 або 256 біт) кіл циклу

простих двійкових арифметичних дій з ключем, зміщення в рядку або заміна колонок. Довжина ключа складає 128, 192 або 256 біт. Довший ключ – більше кіл циклу. Під час шифрування довжина блоку не змінюється. Довжина зашифрованих даних складає 128 біта.

Таблиця 4.1 Порівняльні характеристики процедур шифрування DES та AES

Назва	Довжина вхідного блоку, біт	Кількість та розмір під блоків	Довжина ключа,біт	Кількість циклів шифрування	Довжина вихідного блоку, біт
DES	64	2 по 32 біта	56	16	64
AES	128	4 по 32 біта	128	10	128
			192	12	
			256	14	

3.4 Стійкість шифрів до злому

Проаналізуємо стійкість шифрів до злому. Уже в 1987 році був розроблений алгоритм обчислення ключа DES – метод Девіса (Davies), заснований на специфічних властивостях таблиць замін DES. Посилений метод дозволяє обчислити 6 бітів ключа DES (решта 50 бітів – повним перебором можливих варіантів) при наявності 250 пар відомих відкритих текстів або обчислити 24 біта ключа при наявності 2 пар.

Надалі ці атаки були посилені (наприклад, атака лінійним криптоаналізом при наявності 243 пар відомих відкритих текстів замість 247), з'являлися також нові види атак на DES (наприклад, атака, що дозволяє обчислити ключ високоточним аналізом апаратного шифратора і подальшим аналізом помилок шифрування). Однак варто сказати, що всі ці атаки вимагають наявності величезної кількості пар «відкритий текст – шифртекст», отримання яких на практиці є настільки трудомісткою операцією, що найбільш простий атакою на DES вважається повний перебір можливих варіантів ключа шифрування. Крім того, практично відразу після появи DES були виявлені наступні проблеми з ключами шифрування DES. [12]

- 4 ключа з можливих 256 ключів алгоритму є слабкими (тобто не забезпечують необхідної стійкості при зашифровування). Це ключі, в яких всі біти будь-якої з половин розширеного ключа є нульовими або одиничними. В цьому випадку всі ключі раундів будуть однаковими.
- 6 пар ключів є еквівалентними (тобто інформація, зашифрована одним ключем з пари, розшифровується іншим ключем тієї ж пари), наприклад, пара ключів E0FEE0FEF1FEF1FE і FEE0FEE0FEE1 FEE 1 (шістьнадцяткові значення). Процедура розширення такого ключа замість 16 ключів раундів виробляє всього 2 різних ключа.

- 48 ключів є «можливо слабкими». Можливо слабкі ключі при їх розширенні дають тільки 4 різних ключа раундів, кожен з яких використовується при шифруванні по 4 рази.

Ймовірність вибору слабого ключа

$$P = \frac{64}{2^{56}} = 8.8 \times 10^{-16}$$

Головна перевага методу DES – простота реалізації. Недолік – відносно слабка стійкість проти кваліфікованих крипто аналітиків.

Також основними недоліками симетричних методів є необхідність організації закритого каналу для передачі ключа. Тобто зловмисник має можливість отримати ключ.

Слабкість широкого в використанні алгоритму шифрування AES була проаналізована дослідниками Alex Biryuko, Orr Dunkelman, Nathan Keller та інш. Вони показали, що 256-бітна версія AES сприйнятлива до серії так званих related-key атак (атак з пов'язаними ключами), що істотно знижують необхідний для пошуку ключа час. Одна з технік, яка використовується проти 11-раундової версії алгоритму, може бути завершена за 2^{70} операцій. Інша техніка використовує тільки два пов'язаних ключа для злому повного ключа 9-раундової версії за 2^{39} операцій, що значно швидше в порівнянні з показником 2^{120} для кращого попереднього механізму атаки. Третя техніка зламує 10-раундову версію за час виконання 2^{45} операцій. Як і попередні техніки атак на ключ, останні запропоновані методики здебільшого непрактичні для швидкого розкриття потрібних даних.

Проте, цінність досліджень стійкості алгоритму визначається його широким розповсюдженням в шифруванні чутливої до розкриття інформації та каналів її передачі. AES лежить в основі кількох алгоритмів-кандидатів на новий алгоритм хешування SHA-3, який повинен бути прийнятий американським Національним інститутом стандартів і технологій (US National Institute of Standards and Technology).

Дослідження дали і несподіваний ефект: виявилося, що 256-бітний AES менш стійкий, ніж 192-бітний. Атаки з пов'язаними ключами практично не працюють з AES-192 або AES-128. Робота заснована на попередніх дослідженнях, описувала шляхи отримання деталей про ключах за допомогою використання техніки "бумеранг-атак". Вони також практично непридатні, але просування в методиках свідчить про слабшає стійкості стандарту шифрування AES

3.5 Практичне використання шифрування

Розглянемо ситуації коли потрібно зашифрувати певний файл або папку. Наприклад, якщо дані передаються по відкритих каналах або зберігаються на зовнішньому носії. Для подібних завдань цілком підходить OpenSSL – надійне крос платформенне рішення.

OpenSSL підтримує різні алгоритми шифрування, плюс він за замовчуванням встановлений в багатьох операційних системах, а установка на інші не складе труднощів. OpenSSL це мультисистемний програмний пакет, що базується на двох методах шифрування (симетричному та асиметричному.

Більш простим та вживаним є симетричний метод шифрування даних за допомогою OpenSSL. Приведемо приклад шифрування документа файл doc.tar.gz на платформі linux з алгоритмом AES з довжиною ключа 256 біт:

```
openssl enc -aes-256-cbc -salt -in doc.tar.gz -out doc.tar.gz.a
openssl enc -d -ades-256-cbc -in doc.tar.gz.a -out doc.tar.gz
```

При виконанні шифрування буде запропоновано вибрати пароль. Цей пароль буде використаний при розшифруванні. [13]

OpenSSl використовується не лише для шифрування файлів. Цей програмний пакет можна також використати для побудови ключів-сертифікатів необхідних для роботи інших програм. Наприклад інтернет сайту.

3.6 Шифрування в інтернеті

Для шифрування обміну інформації в інтернет використовуються наступні алгоритми шифрування: SSl, TSL.

Для забезпечення безпеки передачі даних через інтернет використовується протокол SSL. Зрозуміти, що дані передаються в шифрованому виді можна поглянувши в адресну строчку браузера. HTTPS – означає що дані передаються в шифрованому виді. HTTP – не шифрованому. Шифрування даних в інтернеті згають ,що зловмисник який спробує отримати ці дані та прочитати її побачить лише незрозумілий набір символів. Цей набір майже неможливо розшифрувати. В наш час це важливо тому, що більшість людей оплачуєте покупку в інтернет магазинах в режимі онлайн. Всі такі сервіси зобов’язані зобов'язані встановити ssl сертифікат для сайтах для оплати покупок чи послуг.

SSL виконує процедуру перевірки автентичності, щоб мати гарантію того, що обидва користувачі ()покупець та продавець) дійсно є тими, ким вони себе ідентифікують.

SSL серед іншого забезпечує цілісність даних, тобто, забезпечується незмінність даних на шляху між користувачами. SSL постійно модифікується. Кожна нова версія більш безпечна чим попередня.

Одне із оновлень SSL стали називати TLS (transport layer security). TLS є прямим нащадком свого попередника SLS. TLS також постійно оновлюється. Відмінності між остаточною версією SSL (3.0) і першою версією TLS не є радикальними, зміна імені було застосовано для позначення зміни власника.

TLS - це сучасний протокол шифрування даних в інтернеті. Багато людей часто продовжують називати його "SSL-шифруванням". Це може бути

джерелом плутанини для споживачів, які купують рішення для забезпечення безпеки. Але можна сказати, що в наш час майже скрізь використовується TLS.

На інтернет сервері (наприклад Apache) TLS шифрування виконується на основі 4 файлів. Два з них необхідні для генерування ключа. Інші два є самим ключам. Перший – ключ організації, що видала сертифікат, другий саме сертифікат сайту. Сертифікат сайту може мати кілька рівнів безпеки, підтримувати роботі сайту з суб доменами чи ні. Все це впливає на ціну таких сертифікатів. На міжнародному ринку безліч компаній, що надають послуги з генерації файлів сертифікатів для адміністраторів сайтів. [14]

Особисто я маю досвід налаштування власного сайту для підтримки роботи в режимі шифрування TLS. Чотири необхідних файла було замовлено на ukrnames.com.

esites.pp.ua.key

esitespp.ua.csr

esutes.pp.ua.crt

comodo.crt

esites.pp.ua.key має наступний вміст:

----BEGIN PRIVATE KEY-----
MIIEvwIBADANBgkqhkiG9w0BAQEFAASCBKkwggSlAgEAAoIBAQDX8Dz69Z93ltTk
yujqLMOAPAJbV6MQx2+qRjM0tAqTdsJlkXbBrwL0Q2bD5ywdBQZYCcvXIq6WDGPa
vGCyZ/WRyLNRvjs/0SDq11tvU5BiODlA7fQ/iL8r6v+NziGe5xc3bcNkfCHIbIDM
vshGGWC60wZ1ZjGfmIQMc7olpypQU+ZXKH/6PphY+c2DbAUTLJVz931JFkMXHpJs
D2mkmaxRIgpjp1xuxOqYB7dbTwjSnmByFo/jsjiQjgOKmypZU0fZiOfpQDjXszdw
742tSQTRxeqA2TdPqMXJcfYg3vkykwcjEKkNRiWTw2/txK9RfanLIJ5dHxUuDyqw
MTWuNMmLAgMBAAECggEBAJm3peFMipl6INFB9K4G8aDOlyro5ubh4gbHdFo3NDQn
XVkqROy5phO2C2i3nrqFwi9XwejDAp9D8K1cXkWCJdkCk0xEbC8QsysZ5TgJt4WD
2pXzDmVP9i7QWd1beKmCfg9yLp7Qksf3HXUi9rSYCjjUUguulPK0goPOuhtltYR5
9nyb2rayeC8pJjuRfQDmTpCMbhGCNnAlKI5HYX7y5Ekjp/j/SIiusgvLP03z2IAE
w7QFusg6Rm1j3v8hvcmMT4hxF8sWHl119F1OYIBnr0S1VoW86Hys4/spLA9xPaZ6
+vh/Cjx233DMxFLAoeschM23NplJe8Do3imTz+VyrWkCgYEA7tHpaTUxL1BhUcFS
epk8GRBE/6Ybajwmto4SZ5mycGRk7NqkwkH8J9ra0tucgzjbLIDfToFVeSjJovRw
xYXuLsOxtVc/3BJVhl0dHfWMy5A/TIDjicU4/7HHkVLwZtMxv1689vCgYONrG84K
Sv/RhHi8UZiyxNsuw9hRo3VUqvcCgYEA53jxJpeWw5qVgGyJmZB3+IZAPGG6hP8C
2gKRMFfBrXUPgcNVYCWOYIkBe1mEB+MIg8iwiYmJNTWAaMHXM3XRr6c9fxLVTm9V
gW35gdaKQJ4na/+7ema2SxJobInFsjpeu1PIK80sjzSqR/SoibYHFQdY22m2NK/7
ZiII3FUe/Q0CgYEA6/wiT6KPGnir4bRNAvYeC4onFBMYLe6INoXGq/aU+zhI/j7C

5W18dAlvh0IYsxXghWZGwnVMd2+VZ19xe8zu4ejf1hgD8ztupVmnJu53lMLRuY/2
xzFKNGeAkxpIkOKJ2tcLlFuLrpGqSgxb4WZNqz/+ngZsR9dpq346lXGgG1sCgYEA
l6aUD923B9YqZqY1jGerQibqxOSeZ+4PNzgBWXPVT0hnbXPFZqWUX0jDsMXYC7bB
6F0+1A1I2JSB2aBN1sg1JejjEHMNT0L/+TlNA7TEhUtYtpgogyvx+MNJqPqmjHtT
f1WHyrK0o+o0Ocpf9UFN6IgUvkPG/UPi6GoZ9gjKsa0CgYABGCIUkYMFvuH59r7b
jq2YyBIxUi/Jvnga4WKYFPFxsC9fuqpj0xqvDWP99vpvNO4/GYqtSaOrV/U3T5EL
GLGG1+7DDvPiIxsiC5isZ37RWJdrxZcBg67a9cJAKskWfryAGO1Ea6KYZYiL/Ej+
klrkT5NsPlmsUSpn8o3zcdAtpg==
-----END PRIVATE KEY-----

esites.pp.ua.csr має наступний вміст:

----BEGIN CERTIFICATE REQUEST-----
MIIC5jCCAc4CAQAwgaAxCzAJBgNVBAYTAlVBMRkwFwYDVQQIDBDQmtC40LXQstGB
0LrQsNGPMRkwFwYDVQQHDBDQo9C60YDQsNC40L3QutCwMRAwDgYDVQQKDAdlLXNp
dGVzMQswCQYDVQQLDAJJVDEWMBQGA1UEAwwNZS1zaXRlcy5pbi51YTEkMCIGCSqG
SIb3DQEJARYVdG9saWtsZW9uNDhAZ21haWwuY29tMIIBIjANBgkqhkiG9w0BAQEF
AAOCAQ8AMIIBCgKCAQEA1/A8+vWfd5bU5Mro6izDgDwCW1ejEMdvqkYzNLQKk3bC
ZZF2wa8C9ENmw+csHQUGWAnL1yKulgxj2rxgsmf1kcizUb47P9Eg6tdbb1OQYjg5
QO30P4i/K+r/jc4hnucXN23DZHwhyGyAzL7IRhlgutMGdWYxn5iEDHO6JacqUFPm
Vyh/+j6YWPnNg2wFEyyVc/d9SRZDFx6SbA9ppJmsUSIKY6dcbsTqmAe3W08I0p5g
chaP47I4kI4DipsqWVNH2Yjn6UA417M3cO+NrUkE0cXqgNk3T6jFyXH2IN75MpMH
IxCpDUYlk8Nv7cSvUX2pyyCeXR8VLg8qsDE1rjTJiwIDAQABoAAwDQYJKoZIhvcN
AQELBQADggEBAJTGu+dUSlKqvrmeDD8bthowLQ2Dr/EH7/Z456a6LomW+pBvlgpW
41szDsw+h1JinvR4V5VH4fUXeH3exgWwwlEAm+n/zQ0/SyMyqLeKw7gmh6ZH/tBk
hiWPDfdNlYMA3G+9sqo/peJE9JhBGaseZyh9TxYtm6FkRYViqGLBJbkj1vNACpF9
SdJNi1ijebUX+tKbudB9umhwbxistKfUHhVXoc3xYb7dd/jXMVp4dcxffdnyxsgy
3qaxewitMUuFFWMyn99JEIHtdDbNhccFDNaQMbfhyCvELrlihX1XOSRbH2dWCCdb
G20dDc7Y3Fe79P8YQuVVGxdXVJzzSSUZQPQ=
-----END CERTIFICATE REQUEST-----

esites.pp.ua.crt має наступний вміст:

-----BEGIN CERTIFICATE-----
MIIGQjCCBSqgAwIBAgIQJSAvGdisbbQGl5Jg230i7TANBgkqhkiG9w0BAQsFADCB
jzELMAkGA1UEBhMCR0IxGzAZBgNVBAgTEkdyZWF0ZXIgTWFuY2hlc3RlcjEQMA4G
A1UEBxMHU2FsZm9yZDEYMBYGA1UEChMPU2VjdGlnbyBMaW1pdGVkMTcwNQYDVQQD
Ey5TZWN0aWdvIFJTQSBEb21haW4gVmFsaWRhdGlvbiBTZWN1cmUgU2VydmVyIENB
MB4XDTE5MDkyODAwMDAwMFoXDTE5MTIyNzIzNTk1OVowgZwxITAfBgNVBAsTGERv
bWFpbiBDb250cm9sIFZhbGlkYXRlZDFCMEAGA1UECxM5UHJvdmlkZWQgYnkgQ2Vu
dGVyIG9mIFVrcmFpbmlhbiBJbnRlcm5ldCBOYW1lcyAoVUtSTkFNRVMpMRswGQYD
VQQLExJVa3JuYW1lcyBUcmlhbCBTU0wxFjAUBgNVBAMTDWUtc2l0ZXMuaW4udWEw
ggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQDX8Dz69Z93ltTkyujqLMOA
PAJbV6MQx2+qRjM0tAqTdsJlkXbBrwL0Q2bD5ywdBQZYCcvXIq6WDGPavGCyZ/WR
yLNRvjs/0SDq11tvU5BiODlA7fQ/iL8r6v+NziGe5xc3bcNkfCHIbIDMvshGGWC6
0wZ1ZjGfmIQMc7olpypQU+ZXKH/6PphY+c2DbAUTLJVz931JFkMXHpJsD2mkmaxR
Igpjp1xuxOqYB7dbTwjSnmByFo/jsjiQjgOKmypZU0fZiOfpQDjXszdw742tSQTR
xeqA2TdPqMXJcfYg3vkykwcjEKkNRiWTw2/txK9RfanLIJ5dHxUuDyqwMTWuNMmL
AgMBAAGjggKJMIIChTAfBgNVHSMEGDAWgBSNjF7EVK2K4Xfpm/mbBeG4AY1h4TAd
BgNVHQ4EFgQURipeC7Koh72SNtuf/78Ykl4gwZYwDgYDVR0PAQH/BAQDAgWgMAwG
A1UdEwEB/wQCMAAwHQYDVR0lBBYwFAYIKwYBBQUHAwEGCCsGAQUFBwMCMEkGA1Ud
IARCMEAwNAYLKwYBBAGyMQECAgcwJTAjBggrBgEFBQcCARYXaHR0cHM6Ly9zZWN0
aWdvLmNvbS9DUFMwCAYGZ4EMAQIBMIGEBggrBgEFBQcBAQR4MHYwTwYIKwYBBQUH
MAKGQ2h0dHA6Ly9jcnQuc2VjdGlnby5jb20vU2VjdGlnb1JTQURvbWFpblZhbGlk
YXRpb25TZWN1cmVTZXJ2ZXJDQS5jcnQwIwYIKwYBBQUHMAGGF2h0dHA6Ly9vY3Nw
LnNlY3RpZ28uY29tMCsGA1UdEQQkMCKCDWUtc2l0ZXMuaW4udWGCEXd3dy5lLXNp
dGVzLmluLnVhMIIBBQYKKwYBBAHWeQIEAgSB9gSB8wDxAHcAY/Lbzeg7zCzPC3KE
J1drM6SNYXePvXWmOLHHaFRL2I0AAAFteHrWEAAABAMASDBGAiEArx0Zmtfvug/Q
SGSm4Uz72IM3Ho65rOJRPnqr6VWLXFkCIQD1QGX9d1Uud5LQYXTsYCiueMIgnvo1
rxwcnQjoJCJ/FAB2AHR+2oMxrTMQkSGcziVPQnDCv/1eQiAIxjc1eeYQe8xWAAAB
bXh61i4AAAQDAEcwRQIgPRomEwymNkP/u2HV6nXWjmTL9YE+7o+sV9Mo+v+2JaAC
IQD/MDXx172G1gNX2/aOkiLflBI9UIzrRE3iaYGMKxnV4TANBgkqhkiG9w0BAQsF
AAOCAQEART3gQDMdT9LuBbEAgerzhSgQEEEEvoFfdcBT3sgskE+oiOadIRGsOldD
VbnRPVU2bB4luPjdXJx1M9lc9t2Nn0u62SOAzCNyUESiasRC+RwO7LSHMZrdXl5Z
me0vmtxufP5X0k9CtIFBE6DiT74Vr9IZ9mUvmNz80g/F9Y2vzgnIDE4e/ZoMUWJ9
av2H6d+ByFTx1uxADOIMbwMozeqta1dG744RUjVE+fosrBWwcropPyiUg6LFQf02
oqKD32sFp0+eVOK/y3lEiNLTzWajCOI1SVyQJzTx0e5uIEzdQGXkuNhVhy67vSNg
sDYJaA4q2sCk259HmEKnwpk1C1CYBg==
-----END CERTIFICATE-----

Слід зазначити, що сертифікати для сайтів інтернет дійсні лише певний період часу. Зазвичай такі сертифікати працюють на протязі 12 або 24 місяців. Потім потрібне їх поновлення. Якщо сертифікат затермінований, то сайт покаже помилку з можливістю і запропонує перейти в небезпечний режим передачі даних.

Потрібно постійно слідкувати за сертифікатами сайту і періодично їх поновлювати. Крім того браузер показує користувачу рівень надійності сертифіката, і в разі ненадійного або слабо надійного сертифіката користувач може легко залишити ваш сайт. Отже ненадійний сертифікат приводить до втрати користувачів і клієнтів.

Крім того відому пошукові системи не дають високу позицію в пошукових запитах для сайтів з відсутніми сертифікатами.

На рисунку 3.2 зображено представлення сертифікату сайту на стороні браузера користувача.

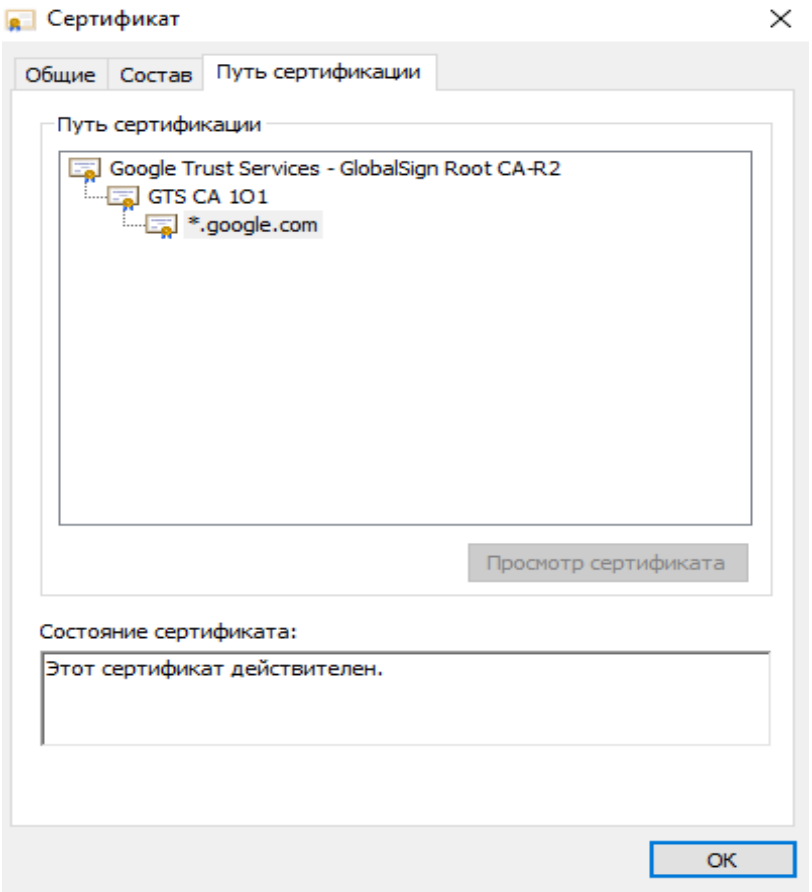


Рисунок 3.2 Приклад сертифікату сайту

Сертифікат вбудовуються в конфігураційні файли веб сервера. Наприклад apache2. Конфігурація веб сервера показана на рисунку 3.3.

```
SSLRandomSeed startup builtin
SSLRandomSeed connect builtin
        AddType application/x-x509-ca-cert .crt
        AddType application/x-pkcs7-crl    .crl
        SSLPassPhraseDialog  builtin
        SSLSessionCacheTimeout  300
        SSLProtocol All -SSLv2 -SSLv3 -TLSv1
        SSLHonorCipherOrder on
        SSLCipherSuite HIGH:!MEDIUM:!LOW:!aNULL:!PSK:!RC4:!MD5:!DES:!3DES
        ServerTokens Min

        <VirtualHost sip.e-sites.in.ua:443>
                DocumentRoot "/var/www/sip.esites.pp.ua"
                ServerName sip.esites.pp.ua
                UseCanonicalName On
                ServerAdmin adminc@esites.pp.ua
                ErrorLog ${APACHE_LOG_DIR}/error.log
                TransferLog ${APACHE_LOG_DIR}/httpd-access.log
                SSLEngine on
                SSLHonorCipherOrder on
                SSLCipherSuite EECDH+AESGCM:EDH+AESGCM:AES256+EECDH:AES256+EDH:ECDHE-RSA-AES128-GCM-SHA384:ECDHE-RSA-AES128-GCM-SHA256:
                ECDHE-RSA-AES128-GCM-SHA128:DHE-RSA-AES128-GCM-SHA384:DHE-RSA-AES128-GCM-SHA256:DHE-RSA-AES128-GCM-SHA128:ECDHE-RSA-AES128-SHA384:
                ECDHE-RSA-AES128-SHA128:ECDHE-RSA-AES128-SHA:ECDHE-RSA-AES128-SHA:DHE-RSA-AES128-SHA128:DHE-RSA-AES128-SHA128:DHE-RSA-AES128-SHA:
                DHE-RSA-AES128-SHA:ECDHE-RSA-DES-CBC3-SHA:EDH-RSA-DES-CBC3-SHA:AES128-GCM-SHA384:AES128-GCM-SHA128:AES128-SHA128:AES128-SHA128:
                AES128-SHA:AES128-SHA:DES-CBC3-SHA:HIGH:!aNULL:!eNULL:!EXPORT:!DES:!MD5:!PSK:!RC4:!3DES
                SSLCertificateFile /etc/ssl/esites/sip.esites.pp.ua.crt
                SSLCertificateKeyFile /etc/ssl/esites/sip.esites.pp.ua.key
                SSLCertificateChainFile /etc/ssl/esites/sip.comodo.crt
        <Directory /var/www/sip.e-sites.in.ua>
                Include /var/www/sip.esites.pp.ua/.htaccess
                 Options Indexes FollowSymLinks MultiViews  ExecCGI
                AddHandler cgi-script .cgi .pl .py
                AllowOverride None
                Order allow,deny
                allow from all
        </Directory>
                <Files ~ "\.(cgi|shtml|phtml|php3?)$">
                    SSLOptions +StdEnvVars
                </Files>

                SetEnvIf User-Agent ".*MSIE.*" \
                         nokeepalive ssl-unclean-shutdown \
                         downgrade-1.0 force-response-1.0
                CustomLog /var/log/httpd-ssl_request.log \
                          "%t %h %{SSL_PROTOCOL}x %{SSL_CIPHER}x \"%r\" %b"
        RewriteEngine on
        RewriteCond %{REQUEST_METHOD} ^(TRACE|TRACK)
        RewriteRule .* - [F]
        TraceEnable off

        </VirtualHost>
```

Рисунок 3.3 Конфігурація веб сервера

4 МОЖЛИВОСТІ ОБ'ЄДНАННЯ ПРОЦЕДУР КОДУВАННЯ ТА ШИФРУВАННЯ.

Кодування інформаційного сигналу використовується для зменшення впливу завад на шляху проходження сигналу. В той час інформація під час проходження сигналу може бути викрадена зловмисниками. Виникає необхідність її шифрування.

Оптимальним буде сумісне використання процедур кодування та шифрування інформації.

І шифрування і кодування це прості математичні дії над двійковими числами, що повторюються багаторазово за визначеним алгоритмом. Найпоширеніша математична операція, що використовується – виключне АЛЕ (XOR).

Розглянемо варіанти поєднання алгоритмів.

Послідовне поєднання алгоритмів. Алгоритми використовуються один за одним. Наприклад спочатку сигнал шифрується від зловмисників, а потім кодується для зменшення впливу завад. Можливий варіант більш складного поєднання алгоритмів.

Передусім зауважимо, що загрози несанкціонованого доступу та вплив завад виникають саме при передаванні інформації відкритими для стороннього втручання каналами передачі або зберігання інформації. Тому порядок застосування зазначених процедур логічно було б вибрати таким, щоб кодування виконувалось безпосередньо перед передаванням інформації по каналу, а декодування (виявлення або виправлення помилок) зразу ж після приймання інформації приймачем. Але з технічних причин для вирівнювання форматів повідомлення зручно спочатку провести кодування блоку повідомлення, а лише після цього – його шифрування.

Пропонована об’єднана процедура інформаційного обміну може бути представлена такими кроками.

1. Повідомлення, яке потрібно передати, ділимо на блоки довжиною m=48.
2. Ділимо повідомлення на 2 по 24 біта кожне.
3. Проводимо кодування кожного блоку шляхом обчислення (29-24)=5 перевірочних бітів та додавання їх до інформаційних символів.
4. Додаємо по 3 випадкових баластних біта до кожного напів повідомлень.
5. Проводимо шифрування отриманого блоку повідомлення та передаємо його у такому вигляді в канал передачі телекомунікаційними засобами.
6. Після отримання зашифрованого повідомлення та його дешифрування баластні біти відкидається.
7. Проводиться процедура декодування. Вибраний код дозволяє виправляти 1-кратні помилки в блоках повідомлення або виявляти 2-кратні.

4.1 Вибір та обґрунтування довжини блоків при суміщенні процедур.

Ключовим фактором при кодуванні та шифруванні інформації є довжина інформаційного блоку над яким одночасно виконується математичні операції. Довжина блоку для деяких алгоритмів є стандартною, для інших може змінюватись.

Для обчислення блоку сигналу кодування використаємо нерівність Хеммінга

$$k \geq \log_2(k + m + 1)$$

Де k – кількість перевірочних біт , а m – кількість інформаційних біт

n=m+k – загальна кількість біт що передаються

Згідно цієї формули для виправлення 1 помилки при інформаційному пакеті в 7 біт необхідно добавити 3 контрольних біта. При 15 бітах – 4 контрольних. При 31 бітах – 5 контрольних. При 63 бітах – 6 контрольних.

В вихідному сигналі порядок контрольних біт будуть 1,2,4,8,16,32,64, …

При виборі довжині блока маємо врахувати наступне. Якщо кодування виконується до шифрування, то матимемо рівність n=d, n=m+k

d – довжина блоку, що піддається алгоритму шифрування. У випадку якщо кодування виконується після операції шифрування матимемо m=d.

Обираємо наступний порядок.

1. Кодування сигналу.
2. Шифрування сигналу.

Довжина блоку шифрованого сигналу буде складати d=64 біта. Під час шифрування блок ділиться на два під блоки по 32 біта кожний. Якщо виконувати кодування на початковому етапі шифрування то матимемо додаткових 5 бітів і 26 інформаційних. Загальна довжина кодованого сигналу складе 32 біт. Порядкові номера контрольних бітів відповідно будуть (1,2,4,8,16).

Матриця перетворення митиме розмірність [5,31]

$$G\,[5,31] = \begin{vmatrix} 10101010101010101010101010101 \\ 01100110011001100110011001100 11 \\ 00011110000111100001111000011 11 \\ 00000001111111100000000111111 11 \\ 00000000000000011111111111111 11 \end{vmatrix}$$

4.2 Реалізація алгоритму об’єднання код Хеммінга + шифр DES

Реалізуємо поєднання процедур шифрування методом DES то кодування алгоритмом Хемінга. Розмір інформаційного блока шифрування методом DES складає 64 біта. Під час шифрування блок ділиться на два під блоки розміром 32 біта кожний.

Розглянемо можливість кодування під час виконання алгоритму шифрування. Алгоритм DES виконує 16 однотипних математичних алгоритмів над блоками фіксованої довжини в 32 біта. Оскільки кодування Хеммінга збільшує довжину блоку під час кожного його застосування то можливості одночасного використання алгоритмів можливе лише на початковому або кінцевому етапі виконання алгоритму шифрування. Циклічність та фіксована довжина блоків під час шифрування зазначеним методом не дають можливості використовувати кодування під час виконання циклу шифрування.

При кодуванні блока в 31 біта матимемо 5 перевірочних біт. Отже 26 біт можуть бути інформаційними. Оскільки всього повний блок складається з двох пів блоків то матимемо 52 біта на вході процедури шифрування. 52 біта складає 6.5 байта. Не ціла кількість байтів часто не є зручною для апаратної та програмної реалізації.

Зробимо наступне. Для подолання розбіжності в пів байта між довжиною блока для шифрування в DES n=64 та довжиною отриманої кодової комбінації коду Хеммінга n=32 найпростіше при шифруванні використовувати блоки на 2 біта менші, а ці біти штучно додавати до блока у вигляді баласту, наприклад, завжди випадковим чином у двох молодших розрядах блоку. Після такого «вирівнювання» до 6 байт (48 біт) інформаційних двійкових символів поділених на 2 пів блоки по 24 біта спочатку додаємо добавляємо по 5 перевірочних біт в розряди 1, 2, 4, 8, 16 згідно алгоритму кодування Хеммінга а потім по 3 випадкових баластних біта в молодші розряди. Отримані два 32 бітових пів

блока подається в цикл на шифрування за стандартною для DES процедурою. Зрозуміло, після дешифрування цей біт як баластний потрібно буде відкинути.

Подібне використання бітів баласту дозволило вирівняти довжини блоків кодування та шифрування в бітах та байтах відповідно.

Алгоритм поєднання кодування і шифрування буди наступним.

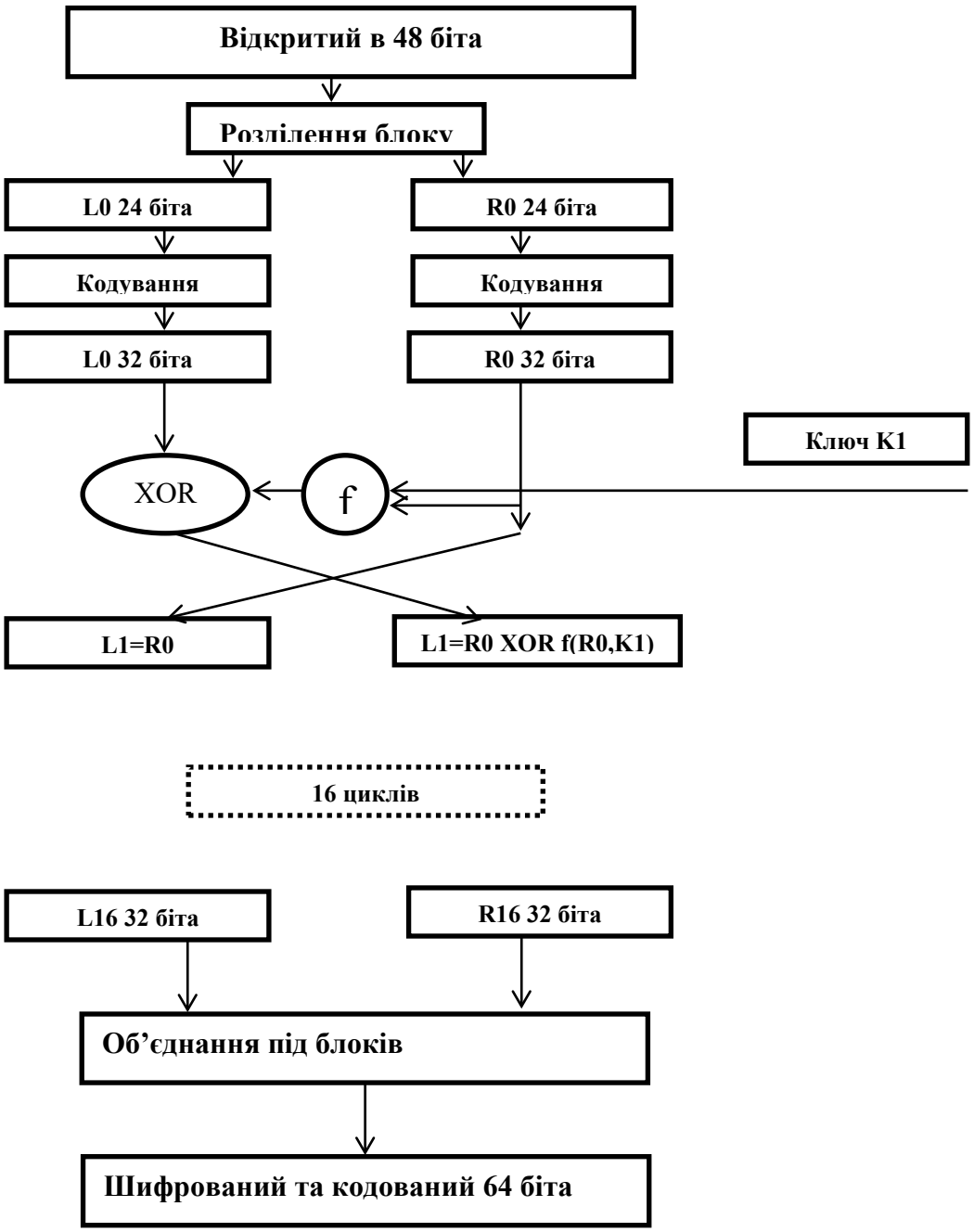


Рис 4.1 Алгоритм поєднання шифрування DES та кодування кодом Хеммінга.

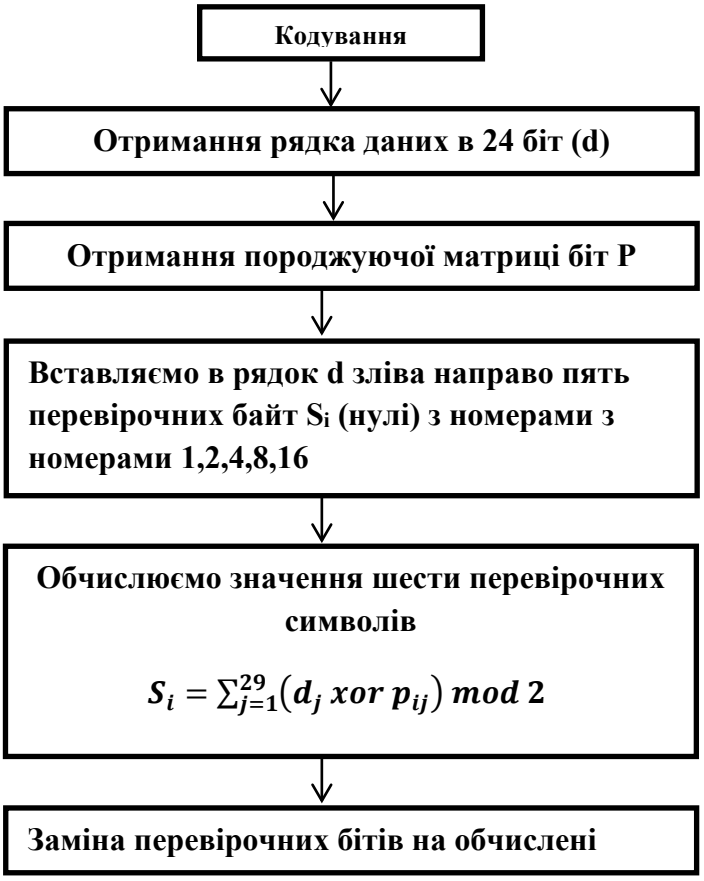


Рис 4.2 Процедура коду Хеммінга.

Твірна матриця

$$G\,[5,29] = \begin{vmatrix} 10101010101010101010101010101 \\ 01100110011001100110011001100 \\ 00011110000111100001111000011 \\ 00000001111111100000000111111 \\ 00000000000000011111111111111 \end{vmatrix}$$

Зміна ключа під час виконання циклу відбувається за стандартним алгоритмом DES. Тобто шляхом зсуву бітів ключа з повного 56 бітного ключа обираються 48 біт необхідних для виконання кожного циклу.

Декодування і виправлення помилок.

Тепер, припустимо, ми отримали закодоване першою частиною алгоритму повідомлення, але воно прийшло до нас з помилкою. Наприклад ми отримали 32 бітний блок з неправильним 11 бітом (11-ий біт передався неправильно).

Вся частина алгоритму декодування полягає в тому, що необхідно заново обчислити всі контрольні біти (так само як і в першій частині) і порівняти їх з контрольними бітами, які ми отримали. Так, порахувавши контрольні біти з неправильним 11-м бітом ми отримаємо таку картину.

Ми бачимо, що контрольні біти під номерами: 1, 2, 8 не збігаються з такими ж контрольними бітами, які ми отримали, а біти 4, 16 збігаються. Тепер просто склавши номера позицій неправильних контрольних біт (1 + 2 + 8 = 11) ми отримуємо позицію помилкового біта. Тепер просто інвертувати його і відкинувши контрольні біти, ми отримаємо вихідне повідомлення в первозданному вигляді.

Тобто сума порядкових номерів вирахуваних заново з отриманого повідомлення контрольних бітів, що не співпали з отриманими контрольними бітами і буде позицією помилки в інформаційному повідомленні.

Тобто алгоритм декодування нічим не відрізняється від алгоритму кодування. Помилку в інформації виявляємо порівнянням контрольних бітів вирахуваних на стороні передатчика та на стороні приймача.

4.3 Розробка практичної реалізації алгоритму об’єднання код циклічний код + шифр AES.

Розробку алгоритму поєднання цих процедур захисту почнемо з процедури шифрування. Стандартна довжина блоку шифрування цього алгоритму складає 128 біт. Під час виконання цього алгоритму з блоку нешифрованих даних формується матриця 4 на 4 байта, або 4 на 32 біта. Маємо матрицю з 4 рядків та 32 колонок одиниць чи нулів. Далі згідно алгоритму AES над цією матрицею виконується циклічно 10, 12 або 14 математичних операцій чи операцій перестановки біт згідно з ключем, розмір якого може бути 128, 192 або 256 біт відповідно. Під час проходження циклу розмір матриці та ключа залишається фіксованим. Оскілки алгоритм кодування збільшує розмір блоку, то єдина можливість його використання під час поєднання процедур це час формування матриці.

Використаємо циклічний код. Для циклічного коду кількість перевірочних символів у блоці довжиною 32 біта буде 5 перевірочних біт. Якщо в коді Хеммінга місця перевірочних символів в коді були (1,2,4,8,16) то в циклічному коді ці символи будуть в кінці блоку.

Використаємо 26 інформаційних символі та 5 перевірочних. Загальна довжина блоку 31 біта. Біт з номером 32 буде баластним який обирається випадковим чином а потім відкидається.

Для роботи алгоритму циклічного кодування вибираємо простий многочлен 5 порядку.

Це буде твірний многочлен:

$$g(x)=x^5+x^2+1.$$

У циклічному кодуванні важливим є використання твірного многочлена. Це такий многочлен правильного порядку, що не може бути розкладений на добуток многочленів меншої розмірності. Існують таблиці таких многочленів.

При циклічному інформаційного кодуванні блоку з 26 біт алгоритм створить відповідний многочлен 25 порядку.

Наприклад для блоку з 26 біт 00000000000000000000001001 буде утворено наступний многочлен:

$$m(x)=x^{25}+x^{23}+1$$

Множимо цей многочлен на x^5 відповідно до алгоритму циклічного кодування.

$$n(x)= (x^{25}+x^{23}+1) * x^5 = x^{30}+x^{28}+ x^5$$

Отримали блок 00000000000000000000001001000000

Ділимо цей многочлен на твірний (100101)

$$k(x)=(x^{30}+x^{28}+ x^5) /(x^5+x^2+1)$$

$$\frac{00000000000000000000001001000000}{100101}$$

Остача від такого ділення буде 01010

Отже отримали наступну кодовану комбінацію

00000000000000000000001001010100 із 32 біта (останній біт баластовий).

Матимемо наступний алгоритм поєднання відповідних процедур шифрування та кодування. З інформаційного потоку виділяємо почергово по 13 байт, що відповідає 104 бітам. З цих 104 біт формуємо матрицю з 4 рядків по 26 біт в кожному. Кожний рядок матриці кодуємо циклічним кодом. Для цього виконуємо поліномне множення рядка матриці на x^5 а потім ділення на твірний поліном x^5+x^2+1.

Цими математичними діями ми отримуємо контрольні 5 біт коду, що дає можливість побудувати рядок довжиною в 32 біта (1 біт баластний). Тобто ми отримали нову матрицю в 4 рядка і довжиною 32 біта кожний. Цю матрицю і подаємо на початок циклу шифрування за допомогою ключа. В кінці циклу шифрування отримаємо уже зашифрований блок даних довжиною в 128 біт.

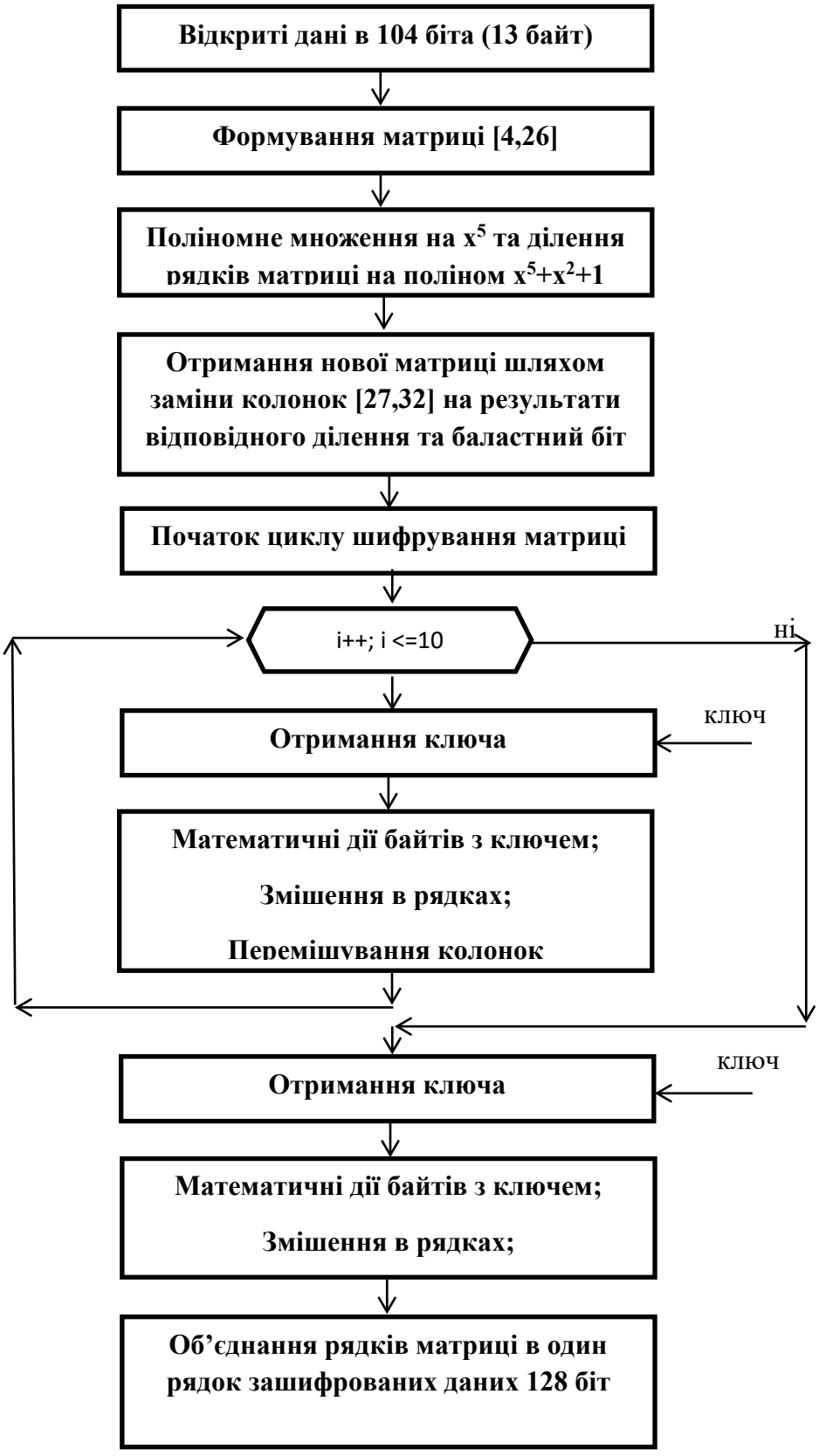


Рис 4.3 Алгоритм поєднання шифрування AES та кодування циклічним кодом.

На відміну від попереднього алгоритму в алгоритмі поєднання процедур кодування циклічним кодом та шифрування AES неузгодженості в розмірності бітів та байтів не виникає. 13 (104 біта) байт інформації під час кодування перетворюються в 16 байт (128 біт) закодованої інформації. Цей блок далі шифрується не змінюючи своєї довжини.

В результаті поєднання процедур шифрування та кодування ми зменшуємо кількість математичних операцій під час передавання інформації, db результаті чого зменшується час на підготовку блоків інформації до передачі по каналам зв'язку. Також підвищується надійність збереження інформації, як від впливу різноманітних чинників на сигнал так і від доступу до інформації зловмисників.

4.4 Аналіз поєднання процедур кодування та шифрування

Поєднання процедур захисту інформації під час ii передавання по провідним та безпровідним каналам зв'язку.

Інформація по каналах зв'язку передається в вигляді електромагнітних сигналів. Канали зв'язку не ідеальні. На них постійно діють різноманітні завади, що спотворюють електромагнітні сигнали які передаються. В результаті цих завад інформація на прийомі може відрізнятись від переданої інформації. Найпоширеніше спотворення – це зміна 1 на 0 або 0 на 1 в двійковому потоці символів, без зміни їх кількості.

Методи боротьби з спотворенням інформації це її кодування. Кодування використовує принцип надлишковості. Тобто додаються додаткові перевірочні символи. Як правило перевірочними символами є символи з номерами 1, 2, 4, 8, 16, 64 ,128, 256 , … (тобто номера ступені 2). Ці символи можуть бути на цих позиціях або в кінці. Така кількість перевірочних символів дозволяє виправити 1 помилку.

Інформація, що передається може бути прочитана або замінена зловмисниками. Щоб уникнути цього інформація шифрується. На відміну від

кодування шифрування не змінює розмір інформаційних блоків. Шифрування та дешифрування виконуються за правилами, що зберігаються в спеціальному файлі. При цьому проходить перемішування символів та різноманітні математичні дії над ними. Цей спеціальний файл називають ключем. Підібрати ключ для дешифровки вкраденої інформації майже не можливо.

Виникає необхідність поєднання процедур кодування та шифрування інформації в межах одного алгоритму. Мета цієї роботи проаналізувати можливості та запропонувати алгоритми поєднання процедур кодування та шифрування.

Одночасне кодування та шифрування сигналу убезпечить інформацію від завад в каналі зв’язку та від зловмисників. До того ж таке поєднання процедур в межах одного алгоритму скоротить кількість математичних дій, а отже збільшить швидкість операцій підготовки інформації до передавання.

Було розглянуто можливості поєднання алгоритмів процедур кодування та шифрування на прикладі самих вживаних процедур.

1. Кодування Хеммінга та шифрування DES.
2. Циклічне кодування та шифрування AES.

Для поєднання процедур шифрування я провів аналіз роботи окремих алгоритмів, розрахував довжину вхідного інформаційного блоку, з урахуванням умов цілісності блоку в бітах та байтах. Для виконання цією умови використав баластні біти. Точку поєднання алгоритмів процедур вибрано з урахуванням:

- Циклічності процедур шифрування;
- Збільшення довжини блоку при виконанні процедури кодування;
- Формування інформаційної матриці з блоку даних в алгоритмі шифрування.

Запропонований об’єднаний алгоритм матиме переваги обох алгоритмів. При цьому він матиме меншу кількість математичних операцій чим при послідовному виконанні алгоритмів кодування та шифрування.

Слід також розглянути сфери застосування запропонованого алгоритму. В наш час швидкими темпами розвивається ринок таких послуг як: відео нагляд, охоронні системи, інтелектуальний будинок. Як правило ці системи потребують дистанційне керування. Легко і зручно за допомогою пульта керувати системами своїм будинком, ставити чи знімати з охороні різні.

Пульти таких систем працюють в середовищі повному різноманітних завад. Такий же пульт в сусідній квартирі буде заважати роботі. Крім того сучасний світ заповнений різними пристроями, що створюють завади. Наприклад мобільні телефони та мікрохвильові печі. Отже завадостійке кодування вкрай важливе для застосування в таких пультах. І з часом ця важливість буде тільки зростати.

Інший фактор – це зловмисники. Зловмисники в разі використання нешифрованого сигналу можуть прочитати чи замінити інформацію, ща надходить з пульта і отримати доступ до систем будинку. Це не допустимо.

Отже сучасні пульти таких систем мають використовувати поєднаний алгоритм кодування та шифрування інформації, що передається.

4.5 Розробка стартапу систем дистанційного керування з подвійним захистом

Мета послуги: Розробити системи дистанційного керування з поєднанням процедур кодування та шифрування інформації під час її передачі.

- Захист інформації та систем керування від різноманітних завад.
- Захист інформації та систем від зловмисників.

Сфера застосування:

- Системи інтелектуального дому.
- Система управління відео наглядом.
- Системи охоронної сигналізації.

Аналіз ринку систем:

1. Поступове збільшення кількості систем відео нагляду, систем охоронної сигналізації, інтелектуальних будинків в багато заселених будинках.
2. Збільшення рухомих та нерухомих об’єктів що потребують захисту

Переваги в порівнянні з існуючими

- Запропоновані системи мають захист одночасно і від впливу різноманітних завад, наприклад сусідніх системи, і від дій зловмисників, що намагаються нашкодити системі чи зупинити її дію.
- Здешевлення елементів системи із за використання об’єднаного алгоритму захисту інформації.

Канали збуду:

- Реклама через за допомогою власного сайту.
- Дірект маркетинг з охоронними компаніями та компаніями по обслуговуванню інтелектуальних будинків.

Яку проблематику ринку вирішує продукт:

- Підвищується надійність роботи систем відео нагляду, охоронних систем та інтелектуальних будинків.
- Зручність в роботі таких систем.

Конкуренти:

- Наявний захист інформації лише від втручання зловмисників.
- Використання двох незалежних систем захисту.

Конкурентні переваги:

- Наявність двох різних рівнів захисту.
- Поєднання процедур захисту в одному алгоритмі.

Споживачі продукції і цінність пропозиції:

- Власники автомобілів, котеджів та новобудов.
- Фірми по обслуговуванню охоронних та інших систем.

Обґрунтування бізнес моделі:

- Пропозиція є цінною для споживачів.
- Широкий сегмент споживачів.
- Низька цінова пропозиція.
- Можливість взаємодії з партнерами.

Точка беззбитковості:

- Ціна налаштування одиниці системи дистанційного керування складає 200 грн. Налаштовано 100 одиниць.
- Постійні витрати (оренда приміщення, обладнання) на написання програми за новим алгоритмом становлять 30000 грн.
- Змінні витрати (зарплата, відрядження) становлять 10000 грн.
- Середні змінні витрати = 10000/100 = 100 грн
- $\text{ТБ} = \frac{30000}{200-100} = 300$ одиниць . Тобто точка беззбитковості – налаштування за новим алгоритмом 300 одиниць систем дистанційного керування. Валовий доход складе $300 \times 200 = 60000$ грн

Висновки проекту:

- За результатами розгляду стартапу можна відмітити перспективність впровадження систем дистанційного керування з поєднаними процедурами кодування та шифрування інформації в охоронних системах, системах відео нагляду та інтелектуальних домах.
- Необхідно також відмітити можливість модернізації уже встановленого подібного обладнання.

ОСНОВНІ РЕЗУЛЬТАТИ РОБОТИ ТА ВИСНОВКИ

Проведено дослідження поєднання різноманітних процедур, що використовуються для захисту та збереження інформації в різноманітних інформаційних системах. Вивчено загрози, які виникають під час обробки та збереження інформації, проведено класифікацію цих загроз та проаналізовано ступінь їх важливості. Проаналізовано засоби, що необхідні для підвищення ефективності різноманітних процедур захисту та підвищення надійності. Важливими засобами захисту інформації є програмні засоби. Саме вони виконують основну роботу по захисту, тому важливим є вивчення критеріїв надійності роботи програмного забезпечення при налаштуванні процедур захисту інформації.

Проведено можливість поєднання процедур шифрування та кодування. Алгоритми шифрування AES та DES та кодування по циклічному коду і коду Хеммінга. Таке поєднання процедур можливо виконати в одному алгоритмі, що дає за певних умов скорочення арифметичних операцій під час проходження інформації. Це може бути суттєвим чинником за наявністю обмеженого за обчислювальною потужністю процесора та іншими характеристиками пристроїв передавання інформації. Одночасне використання процедур кодування та шифрування крім того приведе до збільшення рівня захищеності інформації під час її передавання по каналам зв’язку. Так як лише кодована інформація може бути прочитана зловмисниками, а лише шифрована може бути пошкоджена в результаті дії різноманітних завад на каналах зв’язку. Слід враховувати також те, що в шифрованій інформації на відміну від простого нешифрованого тексту важко помітити помилку, що виникла в результаті дії завад. Одночасне кодування та шифрування суттєво знижує ймовірність такої події.

На основі проведеної роботи запропоновано стартап впровадження пультів дистанційного керування відео та охоронних систем, інтелектуальних домів з

використанням поєднаного алгоритму кодування та шифрування. Передача даних такими пультами буде більш захищені від дій різноманітних завад та зловмисників.

ПЕРЕЛІК ЛІТЕРАТУРНИХ ДЖЕРЕЛ

1. Характеристика джерел загроз безпеці персональних даних в ІСПДн. - um.co.ua - учбові матеріали та реферати - http://um.co.ua/6/6-9/6-9756.html.

2. Агурьянов И. - Классификация методов и средств защиты информации - https://www.securitylab.ru/blog/personal/aguryanov/30011.php.

3. Шишов О.В. - Информационная безопасность информация как объект защиты. Необходимость и направления защиты - https://studopedia.org/13-97373.html.

4. Программные средства защиты информации Материал из Национальной библиотеки им. Н. Э. Баумана -22 марта 2015. - https://ru.bmstu.wiki/Программные_средства_защиты_информации.

5. Носачёв С.В. - Архиваторы и архивация данных. Методические указания к выполнению лабораторной работы по дисциплине «Вычислительные машины, системы и сети» Ростов-на-Дону 2010.

6. Тарнавський Ю.А, Технології Захисту Інформації. - Київ КПІ ім. Ігоря Сікорського 2018. - 161 с.

7. Набатов. К.А., Громов Ю.Ю., Иванова О.Г., Мосягина Н.Г. – Надёжность информационных систем : учебное пособие – Тамбов : Изд-во ГОУ ВПО ТГТУ, 2010. – 160 с. – 100 экз

8. Песков С.Н. директор МВКПК, к.т.н.; А.Е. Ищенко директор ООО «ТехноСат» – Расчет вероятности ошибки в цифровых каналах связи. «Теле Спутник» | ноябрь | 2010

9. Сорока Н. И., Кривинченко Г.А. – Телемеханика. Модуляция и кодирование информации. – Минск БГУИР 2020 – 196 с.

10. Соловьева Ф.И. Введение в теорию кодирования Учебное пособие Редактор С. Д. Андреева Подписано в печать 14.06.2006 г. Формат 84×120 / 8. Офсетная печать. Уч.-изд. л. 17,7. Усл. печ. л. 14,4. Тираж 200 экз/

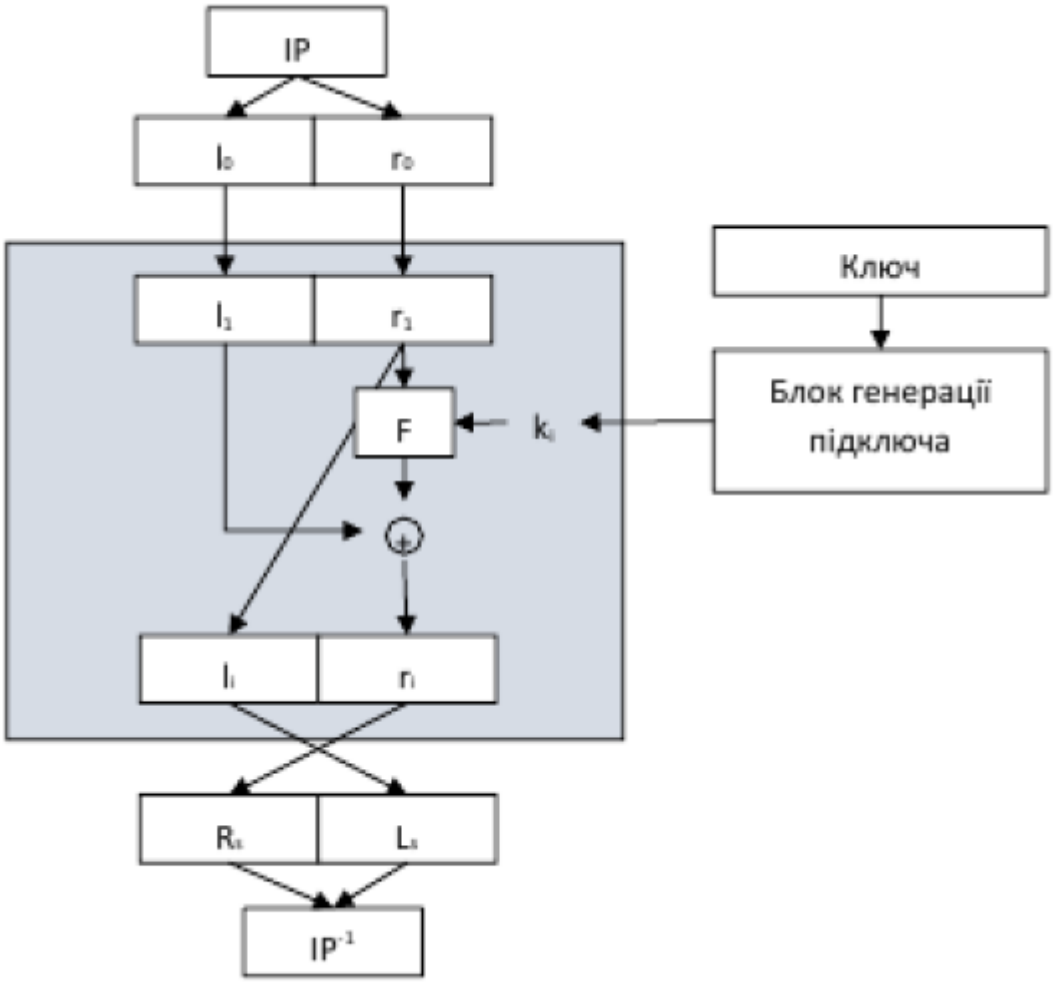
11. Банкет В. Л. Помехоустойчивое кодирование в телекоммуникационных системах: учеб. пособ. по изучению модуля 4 дисциплины ТЭС / В.Л. Банкет, П.В. Иващенко, Н.А. Ищенко. – Одесса: ОНАС им. А. С. Попова, 2011. – 104 с.

12. Сушко С.А. – Практическая криптология лекция 6 Специальность: 6.170101 – БСИТ

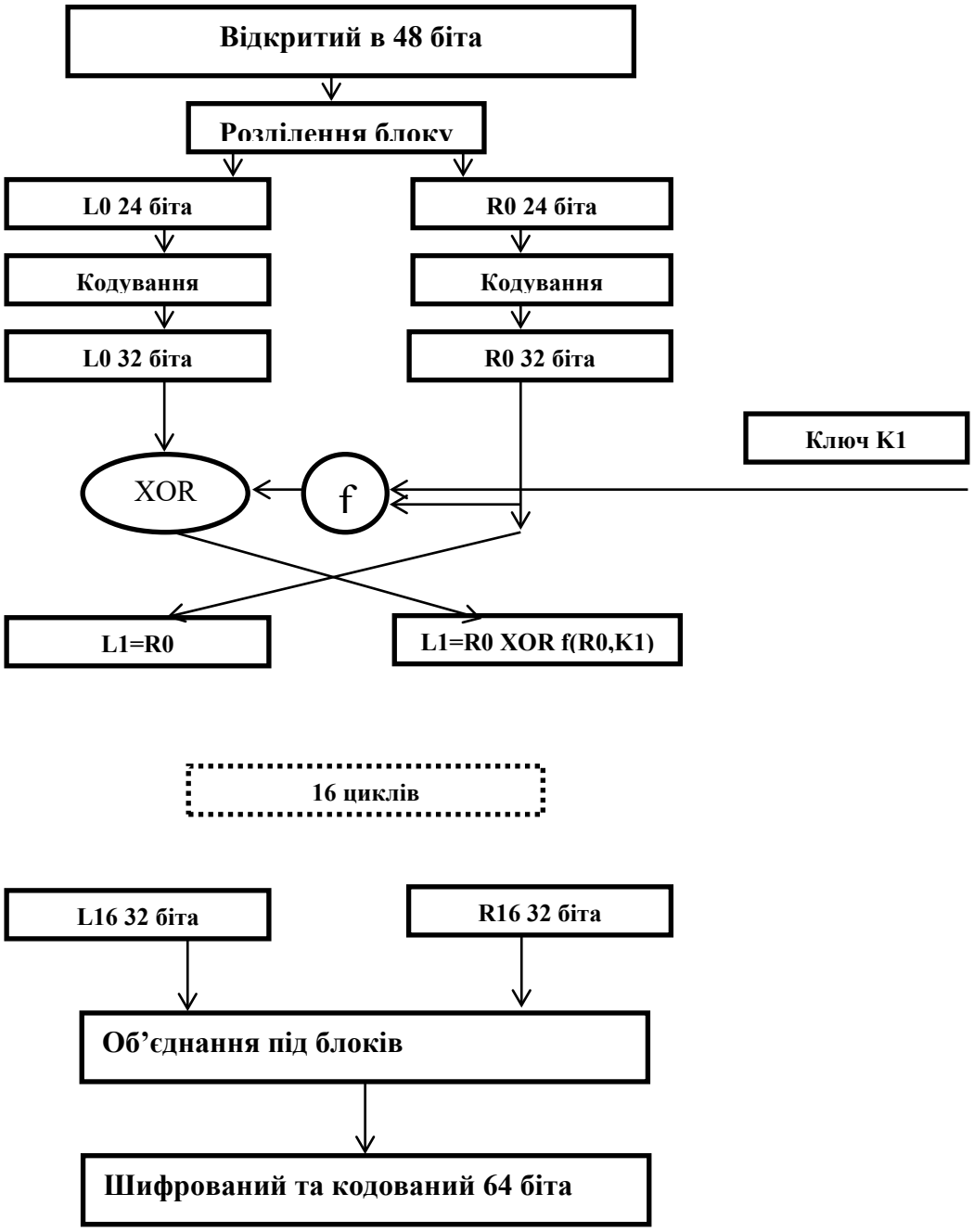
13. brainunit - OpenSSL: простое шифрование с открытым ключом- 9 июня 2009 - https://habr.com/ru/post/61670/

14. Что такое SSL? Предназначение. Разница между SSL и TLS. - http://specialcom.net/it-tehnologii/website-development/chto-takoe-ssl-prednaznachenie-raznica-mezhdu-ssl-i-tls/ - 2012 - 2020 © Specialcom.net

ДОДАТОК А



Структура алгоритму DES.



Алгоритм поєднання шифрування DES та кодування кодом Хеммінга.

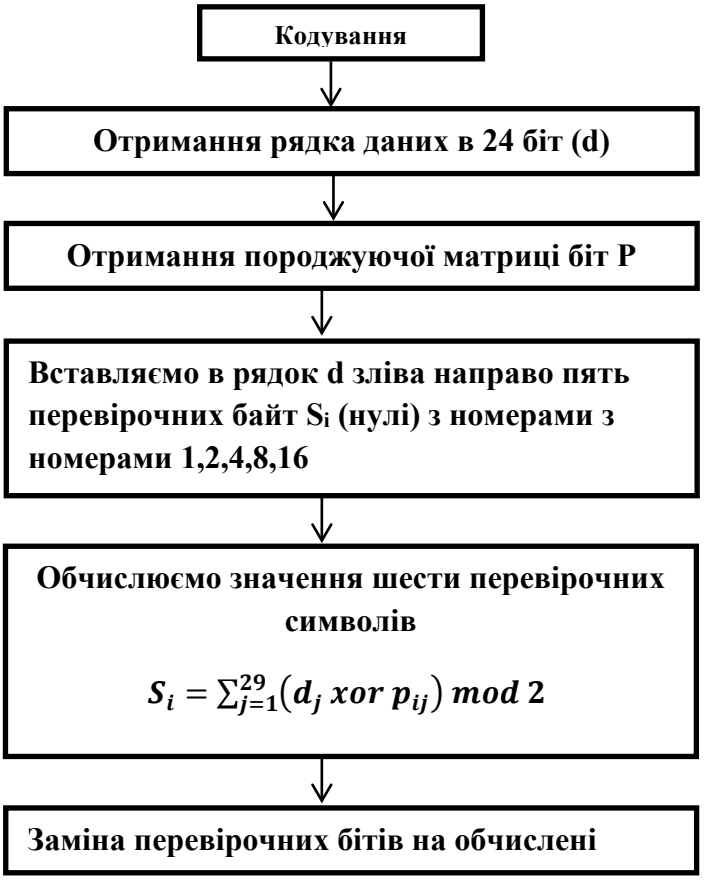
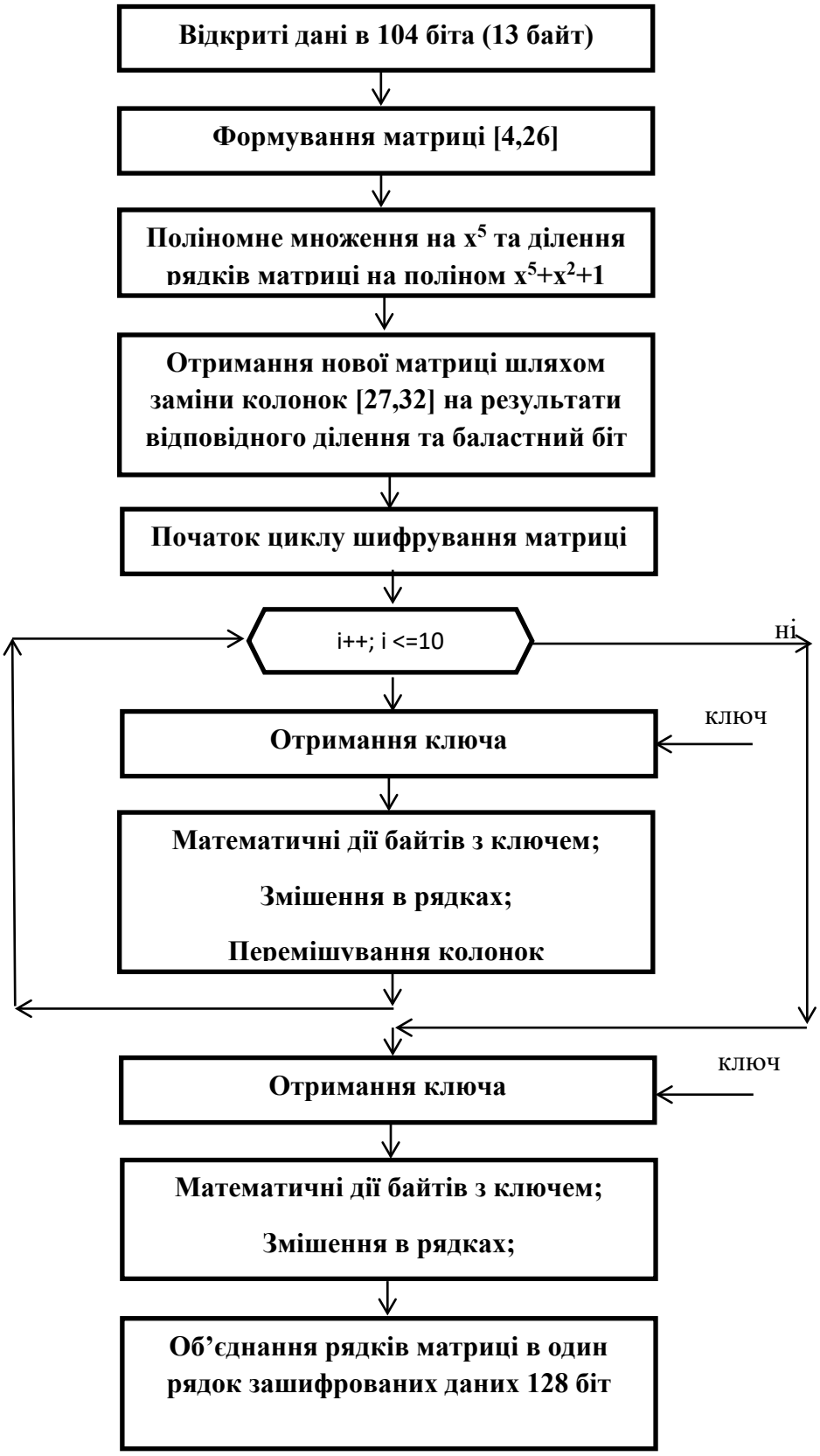


Рис 4.2 Процедура коду Хеммінга.

Твірна матриця

$$G\,[5,29] = \begin{vmatrix} 10101010101010101010101010101 \\ 01100110011001100110011001100 \\ 00011110000111100001111000011 \\ 00000001111111100000000111111 \\ 00000000000000011111111111111 \end{vmatrix}$$



Алгоритм поєднання шифрування AES та кодування циклічним кодом.