Міністерство освіти і науки України
Державний Університет інтелектуальних технологій і зв'язку

Кафедра кібербезпеки та технічного захисту інформації

Л. Г. Йона, О. В. Онацький, Ю. В. Бєлова

КРИПТОГРАФІЧНИЙ ЗАХИСТ ІНФОРМАЦІЇ

Навчальний посібник
з дисципліни
«Криптографічний захист інформації»

для освітньо-професійної підготовки бакалаврів
в галузі знань 12 «Інформаційні технології»
за спеціальністю 125 Кібербезпека

Одеса
2023

УДК 004.056.55:621.39 **План НМВ 2022 р.**

Р е ц е н з е н т
Захарченко М.В., д.т.н., професор кафедри кібербезпеки та технічного захисту інформації, державного університету інтелектуальних технологій і зв'язку

Л. Г. Йона, О. В. Онацький, Ю.В. Бєлова Криптографічний захист інформації: навч. посіб. Одеса, 2023. – 200 с.

Навчальний посібник призначено для студентів, які вивчають дисципліну "Криптографічний захист інформації". Містить систематичне викладення наукових основ від найпростіших прикладів та основних понять до сучасних криптографічних концепцій. Спрямовано на вивчення симетричних криптосистем та криптосистем з відкритим ключем. Розглянуті також питання автентифікації та убезпечення від несанкціонованого втручання в електронних системах.

ЗМІСТ

ВСТУП

При обміні електронними документами в інформаційно-комунікаційних системах істотно знижуються витрати на обробку і зберігання документів, прискорюється їх пошук. Проте зловмисники можуть завдати істотного збитку банківським і комерційним структурам, державним підприємствам і організаціям, а також приватним особам, які використовують електронний документообіг. Це ставить перед користувачем питання про захист інформації, що міститься в документі та встановленні достовірності автора і самого документу.

Захист інформації, яка міститься в повідомленнях, що передаються по комунікаційним каналам, можна забезпечити трьома способами:

– створенням надійного каналу зв'язку;

– криптографічними методами (шифруванням, автентифікацією та розподіленням ключів);

– стеганографічними методами (викриттям факту передачі інформації).

Очевидно, що саме криптографічні методи оптимальні для забезпечення захисту сучасного документообігу. Одним з найнадійніших засобів захисту інформації є використання криптографічних протоколів, за допомогою яких вирішуються різноманітні завдання, а саме: захист інформації шляхом шифрування; підтвердження дійсності користувача/документа за допомогою протоколів автентифікації; розподілення ключів.

Завданнями криптографічного захисту інформації є забезпечення:

– конфіденційності, тобто захисту від просочування інформації(вирішується шифруванням);

– доступності, тобто інформація має бути доступна тільки тому користувачеві, для якого вона призначена (вирішується шифруванням);

– цілісності, тобто інформація має бути захищена від несанкціонованої модифікації (вирішується електронним цифровим підписом);

– автентифікації, тобто підтвердженням достовірності (забезпечується електронним цифровим підписом і сертифікатом);

– неопровергаемости, тобто неможливості відмовитися від виконаної дії (забезпечується електронним цифровим підписом і сертифікатом).

1 КЛАСИЧНІ МЕТОДИ ШИФРУВАННЯ

Для того, щоби захистити інформацію, яка міститься в повідомленні, необхідно вирішити завдання, яке можна розв’язати у три основні способи.

1. *Створити абсолютно надійний (неприступний для інших) канал зв’язку поміж абонентами.*

Проте, слід зауважити, що за сучасного рівня розвинення науки й техніки створити такий канал зв'язку поміж віддаленими абонентами для неодноразового передавання великих обсягів інформації практично є нереально.

2. *Використовувати загальнодоступний канал зв’язку, але приховувати сам факт передавання інформації.*

Засоби та методи приховування факту передавання повідомлення вивчаються наукою ***стеганографією***.

У стеганографії є відомий такий спосіб утаємничування письмового повідомлення. Передаваний текст "розчиняється" у повідомленні більшого розміру із зовсім "стороннім" змістом, але якщо за певним правилом вилучити з цього тексту певні символи, то можна дістати таємне повідомлення (передаване повідомлення може бути приховане також у графічному файлі чи в іншому форматі).

Добре відомі є випадки, коли секретні послання записувалися невидимим *симпатичним* чорнилом (у тому числі й на краватках, манжетах, нижній білизні тощо). Чорнило знову ставало видимим після обробляння спеціальним хімічним реактивом чи променями, зазвичай ультрафіолетом, певної частини спектра.

3 *Використовувати загальнодоступний канал зв'язку, але передавати ним потрібну інформацію в перетвореному у такий спосіб вигляді, аби відновити її міг лише адресат.*

Процес перетворювання відкритого тексту з метою зробити незрозумілим його зміст для сторонніх осіб називається ***зашифруванням.*** Вибір конкретного типу перетворювання визначається ключем шифрування. Інакше кажучи, зашифрування – це процес перетворювання відкритого повідомлення на закрите за допомогою ключа. Зашифроване повідомлення називається шифртекстом чи криптограмою.

Процес оберненого перетворювання шифртексту на відкрите повідомлення на підставі ключа називається ***розшифруванням***.

Дешифрування (розкриття, зламування шифру) – процес здобування захищуваної інформації із зашифрованого повідомлення без знання застосованого ключа.

Ключ – це правило, згідно з яким здійснюються зашифровування та розшифровування текстів. Надійність алгоритму шифрування залежить від довжини (кількісті бітів) ключа. Множину можливих ключів називають *простором ключів*.

Зауважимо, що шифрування й кодування – це різні речі.

При шифруванні треба знати шифр (алгоритм) і секретний ключ (тип перетворювання).

При кодуванні ж немає нічого таємного, є лише певна заміна літер чи слів на заздалегідь окреслені символи. Методи кодування спрямовано НЕ НА ТЕ, аби приховати відкрите повідомлення, а НА ТЕ, аби подати його в більш зручному вигляді для передавання технічними засобами зв'язку, для зменшення довжини повідомлення, зменшення надлишковості, підвищення пропускної здатності каналу тощо. Шифрування з ключем має певні позитивні моменти.

1. Використовуючи ключ, можна застосовувати той самий алгоритм задля відправляння повідомлень різним людям. Головне – закріпити окремий ключ за кожним респондентом.

2. В разі „зламу" зашифрованого повідомлення достатньо лише змінити ключ, але переходити на новий алгоритм немає потреби.

Питаннями захисту інформації шляхом її перетворювання, яке виключає можливість прочитування інформації сторонньою особою, займається ***криптологія*** (kryptos – таємний, logos – наука). Криптологія поділяється на два напрями – *криптографію* й *криптоаналіз*. Цілі цих напрямів є прямо протилежні. Взаємини криптографії й криптоаналізу є очевидні: криптографія – захист, тобто розробляння шифрів, а криптоаналіз – напад, тобто атака на шифри. Однак ці дві дисципліни є щільно пов'язані, й не існує кваліфікованих криптографів, котрі не володіли б методами криптоаналізу.

Криптографія – одноліток історії людської мови. Більш того, спочатку писемність власне сама була криптографічною системою, тому що в стародавніх суспільствах нею володіли лише обрані.

З широким розповсюдженням писемності криптографія стала формуватися як самостійна наука. Дані про перші способи тайнопису є вельми уривчасті. Припускається, що криптографія була відома в давньому Єгипті й Вавілоні. До нашого часу дійшли свідчення про те, що мистецтво секретного письма використовувалося в давній Греції. Перші насправді вірогідні відомості з описанням методу шифрування належать до періоду зміни старої й нової ери.

Криптографія займається пошуком і дослідженням математичних методів перетворювання інформації.

Криптографія – це наука про способи перетворювання (шифрування) інформації з метою її захисту від неправочинних користувачів.

Криптографічна система – це система, реалізована програмно, апаратно або програмно-апаратно і здійснює криптографічне перетворення інформації з метою її захисту.

Сфера інтересів криптоаналізу – дослідження можливості дешифрування інформації без знання ключів.

Криптоаналіз – це наука про методи і способи розкривання шифрів.

Інший підхід захисту інформації від неправочинних користувачів – не приховувати власне факту передавання повідомлення, але зробити його неприступним для сторонніх. Для цього повідомлення має бути записане у такий спосіб, аби з його вмістом не міг ознайомитися ніхто, за винятком самих кореспондентів, – у цьому й полягає суть шифрування. І криптографія виникла власне як практична дисципліна, котра вивчає й розробляє способи шифрування

повідомлень.

Об’єктом криптографії є інформація (новини, вміст повідомлень) в перебігу передавання її каналами зв’язку.

Слід пам’ятати, що криптографія необхідна лише для інформації, котра потребує захисту. Зазвичай в таких випадках говорять, що інформація містить таємницю, чи є захищуваною, секретною, конфіденційною. Тому, як правило, вивчення криптографії як науки розпочинають з вивчення властивостей відкритої інформації.

Криптографії надають таке означення: ***відкрита інформація*** – це незашифрована інформація, призначена для передавання каналом зв’язку.

Відкрита інформація неодмінно має бути зафіксована на певному матеріальному носії (папері, фотоплівці, перфострічці тощо). У цьому разі її називають ***відкритим повідомленням***. Поняття відкритого повідомлення в криптографічній літературі розуміється подвійно: або це змістовний текст, котрий піддається смисловому читанню, або це текст, котрий підлягає зашифровуванню.

Вочевидь, що термін "криптографія" далеко відійшов від свого первинного значення – "тайнопис", "таємний лист". Сьогодні це наукова дисципліна, яка поєднує методи захисту інформаційних взаємовпливів різноманітного характеру, які спираються на перетворювання даних за секретними алгоритмами, включаючи алгоритми, котрі використовують секретні параметри. Термін "інформаційний взаємовплив", чи "процес інформаційного взаємовпливу" тут позначає такий процес взаємовпливу двох і більш суб’єктів, основним змістом якого є передавання та/чи опрацьовування інформації. Приміром, за криптографічну може вважатися будь-яка функція перетворювання даних, секретна сама собою чи залежна від секретного параметра K:

$$M' = f(M_k) \text{ або } M' = f(M, K).$$

Перетворювання M_k визначається відповідним алгоритмом і значенням параметра K – ключа. Ефективність зашифровування з метою захисту інформації залежить від зберігання таємниці ключа та криптостійкості шифру.

Криптостійкістю називається характеристика шифру, котра визначає його стійкість до дешифрування без знання ключа (тобто до криптоаналізу). Є кілька показників криптостійкості, з-посеред яких:

– кількість усіх можливих ключів;

– середній час, необхідний для криптоаналізу.

Однак, окрім перехоплення й розкриття шифру, неправочинний користувач може намагатися здобути захищувану інформацію багатьма іншими способами, коли криптографія просто неспроможна цю інформацію захистити (приміром, за агентурного способу, тобто за правочинного користувача, схиленого до співробітництва; або неправочинний користувач може намагатися не здобути, а знищити чи змодифікувати в перебігу передавання захищувану інформацію тощо).

Отже, на шляху від одного правочинного користувача до іншого інформацію має бути захищено у різноманітні способи від всіляких погроз. Неправочинний користувач прагнутиме віднайти найслабшу ланку в цьому ланцюзі, аби з найменшими витратами добутися до інформації. Тому правочинні користувачі мають враховувати "засаду рівнопотужності захисту", тобто всі ланки одного ланцюга має бути захищено однаково.

Не слід забувати ще про одне: про проблему співвідношення ціни інформації, витрат на її захист та витрат на її здобуття.

Перш ніж вдаватися до захисту інформації, варто задати два запитання:

1) чи є вона для неправочинного користувача більш вартісною, аніж вартість атаки?

2) чи є вона для правочинного користувача більш вартісною, аніж вартість захисту?

Саме зазначені міркування і є вирішальними при обиранні придатних засобів захисту: фізичних, стеганографічних, криптографічних тощо.

Криптографічні системи за типом алгоритму шифрування поділяються на групи: *симетричні* й *асиметричні*. Крім того, алгоритми поділяються за типом перетворення, за способом обробки інформації.

У *симетричних криптографічних системах* зашифрування й розшифрування проводяться за допомогою того самого ключа. І відповідно цей ключ необхідно зберігати в таємниці (звідси інша назва симетричних криптографічних систем – *криптографічні системи із секретним ключем*).

В *асиметричних криптографічних системах* існують два різні ключі: один використовується для зашифрування, який ще називають відкритим, інший – для розшифрування, який називають закритим, тому ці системи називаються асиметричними, або *криптографічні системи з відкритим ключем*.

Гібридними прийнято називати криптографічні системи, що поєднують обидва типи криптографічних систем, у них, як правило, текст повідомлення зашифровується з використанням симетричної криптографічної системи, а секретний ключ, використаної симетричною криптографічною системою, зашифровується з використанням асиметричної криптографічної системи.

За типом обробки вхідної інформаційної послідовності криптографічні системи поділяються на *потокові*, у яких перетворюються всі повідомлення одразу, і *блокові*, у яких повідомлення обробляються у вигляді блоків певної довжини.

На практиці часто використовуються алгоритми *перестановки*, *підстановки*, а також *комбіновані* методи.

У *методах перестановки* символи вхідного тексту міняються місцями один з одним за певним правилом (за ключем).

У *методах підстановки* (або заміни) символи відкритого тексту замінюються деякими еквівалентами зашифрованого тексту. Шифри підстановки можуть бути розбиті на дві категорії: *моноалфавітної* (*одноалфавітної*) й *багатоалфавітної* підстановки. У моноалфавітних шифрах підстановці літера (або символ) у початковому тексті завжди змінюється на ту саму літеру (або

символ) у зашифрованому тексті незалежно від її позиції в тексті. Багатоалфавітні шифри підстановки замінюють одні й ті самі літери (символи) відкритого тексту щоразу по-різному. Для шифрів підстановки довжина алфавіту відкритого тексту і довжина алфавіту зашифрованого тексту однакова.

З метою підвищення надійності зашифрування текст, зашифрований за допомогою одного методу, може бути ще раз зашифрований за допомогою іншого методу. У такому випадку виходить *комбінований* або *композиційний шифр*.

1.1 Шифри перестановки

Простий метод криптографічного перетворювання, який містить правило перестановки літер у відкритому тексті на підставі ключа шифрування [1]. Тобто, при шифруванні самі літери чи символи вихідного тексту не змінюються, а змінюється лише порядок розміщення символів. Проте, шифри перестановки мають невелику криптостійкість, тому їх не використовують без додаткових перетворювань.

Шифр перестановки здійснює перетворювання переставляння літер у відкритому тексті.

Перетворювання з шифром перестановки, призначене для зашифровування повідомлення довжиною n символів. Його можна подати за допомогою таблиці

$$\begin{matrix} 1 & 2 & \dots & n \\ i_1 & i_2 & \dots & i_n, \end{matrix}$$

де i_1 – номер місця шифртексту, на котре потрапляє перша літера вихідного повідомлення за обраного перетворювання; i_2 – номер місця для другої літери й так далі.

У верхньому рядку таблиці виписано одне за одним числа від 1 до n, а в нижньому – ті ж самі числа, але в довільному порядку (за ключем шифрування). Така таблиця називається перестановкою ступеня n.

Приклад 1.1. Зашифруємо слово СОНЦЕ класичним шифром перестановки.

Оберемо ціле додатне число 5 (за кількістю букв у слові); розташуємо числа від 1 до 5 у дворядковий запис, в якому другий рядок – довільне переставляння чисел верхнього рядка (за ключем):

Номер літери у тексті	1	2	3	4	5
Ключ	2	5	4	1	3

Цю конструкцію називають перестановкою, а число 5 – степенем перестановки. Відповідно до обраного правила шифрування, виходить зашифроване слово ЦСЕНО.

1.1.1 Прилад Сцитала

Одним з перших фізичних приладів, які зреалізовують шифр перестановки, є так званий прилад Сцитала. Його було винайдено у давній, "варварській", Спарті за часів Лікурга (V ст. до н. е.). Рим швидко скористався цим приладом. Для зашифровування тексту використовувався циліндр заздалегідь обумовленого діаметра. На циліндр намотувався тонкий ремінець з пергамента, і текст виписувався порядково вздовж осі циліндра. Потім ремінець змотувався й доправлявся одержувачеві повідомлення. Останній намотував його на циліндр того самого ж діаметра і зчитував текст по осі циліндра. У цьому прикладі ключем шифру є діаметр циліндра та його довжина, котрі, власне кажучи, породжують дворядковий запис, аналогічний до того, що його наведено вище.

Шифр Сцитала реалізує один з варіантів сучасного так званого шифру маршрутної перестановки [2]. Зміст цього шифру полягає в такому.

Відкритий текст виписується в прямокутну таблицю з n рядків та m стовпців. Припускається, що довжина тексту $t \le nm$ (у противному разі ділянка тексту, що залишилася, шифрується окремо за тим самим шифром). Якщо t є строго менше за nm, то порожні клітинки, що залишилися, заповнюються довільним набором літер абетки. Шифртекст виписується за цією таблицею заздалегідь обумовленим "маршрутом" – шляхом, що він проходить одноразово через усі клітинки таблиці. Ключем шифру є числа n та m і окреслений маршрут.

У такому трактуванні шифр Сцитала набуває описаного нижче вигляду. Нехай m – кількість обвитків ремінця на циліндрі; n – кількість літер, розташованих на одному обвитку. Тоді відкритий текст, виписаний порядково в зазначену таблицю, шифрується шляхом послідовного зчитування літер за стовпцями. Оскільки маршрут є відомий і незмінний, то ключем шифру є числа m та mn, зумовлені діаметром циліндра й довжиною ремінця. При перехопленні повідомлення (ремінця) єдиним секретним ключем є діаметр.

Винахід дешифрувального пристрою – Антисцитала – приписують великому Аристотелю [2]. Він запропонував використовувати конусоподібний "спис", на який намотувався перехоплюваний ремінець; цей ремінець пересувався віссю доти, аж допоки не з’являвся осмислений текст.

1.1.2 Шифрування за допомогою шифрів перестановки

Нагадаємо, що шифр, перетворювання з якого змінюють лише порядок слідування символів вихідного тексту, але не змінюють їх самих, називається шифром перестановки. Розглянемо різноманітні способи шифрування за допомогою шифрів перестановки.

У Прикладі 1 розглянуто класичний шифр перестановки за ключем. Якщо ускладнити завдання та взяти відкритий текст довшим, тоді процес шифрування буде таким, як показано в прикладі 1.2.

Приклад 1.2. Зашифруємо фразу ДО БУЛАВИ ТРЕБА ГОЛОВИ. Випишемо цю фразу без пропусків, водночас розбивши її (для зручності

шифрування) на п’ятизнакові групи. У цій фразі 19 літер. Доповнимо її довільною літерою (наприклад ф) до найближчого числа, кратного до 5, тобто 20. Таким чином, отримали вихідне повідомлення для зашифрування:

ДОБУЛ АВИТР ЕБАГО ЛОВИФ

Літери кожної групи переставимо відповідно до зазначеного ключем шифрування (32514) запису: перша літера ставиться на третє місце, друга – на друге (тобто залишається на своєму місці), третя – на п’яте, четверта – на перше і п’ята – на четверте. Здобутий текст криптограми виписуємо без пропусків:

УОДЛБТВАРИГБЕОАИОЛФВ

При розшифровуванні текст розбивається на групи по п’ять літер, які переставляються у зворотному порядку: перша – на четверте місце, друга – на друге, третя – на перше, четверта – на п’яте і п’ята – на третє. Ключем шифру є обране число 5 і порядок розташування 32514.

Одним з найпримітивніших табличних шифрів перестановки є проста перестановка, для якого ключем є розмір таблиці. Зашифрувати відкритий текст за допомогою табличного шифру перестановки показано у прикладі 1.3.

Приклад 1.3. Зашифруємо табличним шифром перестановки повідомлення

МАЄШ ГОЛОВУ МАЙ ЖЕ РОЗУМ

записується до таблиці за чергою по стовпцях. Розглянемо результат заповнення таблиці з п’яти рядків і чотирьох стовпців:

М	О	М	Р
А	Л	А	О
Е	О	Й	З
Ш	В	Ж	У
Г	У	Е	М

Після заповнення таблиці текстом повідомлення по стовпцях для формування шифртексту зчитують зміст таблиці по рядках. Якщо шифртекст записувати групами по п’ять літер, виходить таке шифроване повідомлення:

МОМРАЛАОЕОЙЗШВЖУГУЕМ

Природно, відправник та одержувач повідомлення повинні заздалегідь домовитися про спільний ключ у вигляді розміру таблиці. При розшифровуванні дії виконуються у зворотному порядку.

В часи середньовіччя європейська криптографія набула сумнівного

розголосу, який відлунює й дотепер. Криптографію стали ототожнювати з чорною магією, з певною формою окультизму, астрологією, алхімією, єврейською каббалою. До зашифровування інформації долучалися містичні сили. Приміром, рекомендувалося використовувати так звані "магічні квадрати".

"Магічними квадратами" називають квадратні таблиці з вписаними в їхні клітинки послідовними натуральними числами, розпочинаючи від 1, що вони дають у сумі по кожному стовпцю, кожному рядку і кожній діагоналі одне й те саме число [3].

Кількість "магічних квадратів" швидко зростає зі збільшенням розміру квадрата. Кількість "магічних квадратів" 4×4 становить 880, а кількість "магічних квадратів" 5×5 – близько 250 000.

Уперше ці квадрати виникли в Китаї, де їм було надано певної "магічної" сили. Наведемо приклад: у квадрат розміром 4×4 вписуються цифри від 1 до 16 таким чином, як показано нижче.

7	12	1	14
2	13	8	11
16	3	10	5
9	6	15	4

Зашифровування за допомогою "магічного квадрату" здійснюється у такий спосіб, що показано у прикладі 1.4.

Приклад 1.4. Зашифруємо за допомогою "магічного квадрату" фразу

СВЯТКОВИЙ НАСТРІЙ.

Спочатку потрібно пронумерувати літери відкритого тексту. Потім літери цієї фрази вписуються послідовно до квадрата відповідно до записаних в них чисел, а в порожні клітинки (якщо такі є) проставляють довільні літери (пустушки).

7**В**	12**С**	1**С**	14**Р**
2**В**	13**Т**	8**И**	11**А**
16**Й**	3**Я**	10**Н**	5**К**
9**Й**	6**О**	15**І**	4**Т**

Після цього зашифрований текст записується вже в рядок:

ВССРВТИАЙЯНКЙОІТ

При розшифровуванні текст вписується до квадрата – й відкритий текст читається в послідовності чисел "магічного квадрата".

Даний шифр – звичайний шифр перестановки, але вважалося, що особливої стійкості йому надає чаклунство "магічного квадрата".

Широкого розповсюдження набули шифри перестановки, котрі використовують певну геометричну фігуру.

Перетворювання з цього шифру полягають у тому, що до фігури вихідний текст вписується в перебігу одного маршруту, а потім – в перебігу іншого – виписується з неї. Такий шифр називають *маршрутною перестановкою*. Приміром, можна вписувати вихідне повідомлення до прямокутної таблиці, обравши такий маршрут: по горизонталі, розпочинаючи з лівого верхнього кута по черзі в напрямках ліворуч → праворуч та праворуч → ліворуч. В порожні клітинки проставляють довільні літери (пустушки). Виписуватимемо ж повідомлення за іншим маршрутом: по вертикалі, розпочинаючи з верхнього правого кута і рухаючись по черзі зверху донизу і знизу догори. Зазначеним способом відбувається зашифрування у прикладі 1.5.

Приклад 1.5. Зашифруємо за допомогою шифру маршрутної перестановки фразу:

ПРИКЛАД МАРШРУТНОГО ПЕРЕСТАВЛЯННЯ.

Використовуючи таблицю розміром 4×8, дістанемо:

П	Р	И	К	Л	А	Д	М
О	Н	Т	У	Р	Ш	Р	А
Г	О	П	Е	Р	Е	С	Т
Ь	Я	Н	Н	Я	Л	В	А

Зашифрована фраза має вигляд:

МАТАВСРДАШЕЛЯРРЛКУЕННПТИРНОЯЬГОП

Маршрути можуть бути значно витонченішими, однак заплутаність маршрутів ускладнює використовування таких шифрів.

Іноді запам’ятати маршрут чи ключову послідовність досить складно, тоді можна використовувати інший метод шифрування, де ключем є кодове слово.

Метод шифрування, називаний *одиночною перестановкою за ключем*, полягає в тім, що стовпці таблиці переставляються за ключовим словом чи відповідним набором чисел (згідно букв ключового слова в алфавітному порядку)

довжиною в рядок таблиці. Зашифрування одиночною перестановкою за ключем показано у прикладі 1.6.

Приклад 1.6. Зашифруємо методом одиночною перестановкою за ключем. Застосуємо за ключ, слово МАТРИЦЯ. Заповнимо ключовим словом та відповідною числовою послідовністю заголовок таблиці. Потім впишемо відкрите повідомлення по стовпцям у таблицю розміром 7×5,

М	А	Т	Р	И	Ц	Я
З	1	5	4	2	6	7
Л	А	Р	Щ	Е	З	Н
Ю	Б	У	О	В	К	Н
Д	Е	З	Д	О	О	Я
И	З	І	Е	Б	Р	Ь
Н	Д	В	Р	Е	І	Ф

після переставляння за ключем дістанемо.

А	И	М	Р	Т	Ц	Я
1	2	3	4	5	6	7
А	Е	Л	Щ	Р	З	Н
Б	В	Ю	О	У	К	Н
Е	О	Д	Д	З	О	Я
З	Б	И	Е	І	Р	Ь
Д	Е	Н	Р	В	І	Ф

У верхньому рядку даної таблиці записано ключ, а номери під літерами ключа визначено відповідно до природного порядку відповідних літер ключа в абетці. Якби в ключі зустрілися однакові літери, їх було б пронумеровано в порядку ліворуч → праворуч.

У здобутій таблиці стовпці переставлено відповідно до упорядкованих номерів літер ключа. При зчитуванні її вмісту по рядках і записуванні шифртексту без пробілів і розділових знаків, дістанемо шифроване повідомлення:

АЕЛЩРЗНБВЮОУКНЕОДДЗОЯЗБИЕІРЬДЕНРВІФ

Для забезпечення додаткового утаємничення можна вдруге зашифрувати повідомлення, яке вже пройшло шифрування.

Такий метод шифрування називається *подвійною перестановкою*. У разі подвійної перестановки стовпців і рядків таблиці переставляння визначаються

окремо для стовпців і окремо для рядків, при цьому використовуються два ключа шифрування, відповідно.

Спочатку в таблицю записується текст повідомлення, а потім по черзі за ключами переставляються стовпці, а за ними – рядки. При розшифровуванні порядок переставлянь має бути зворотним. Зашифрування одиночною перестановкою за ключем показано у прикладі 1.7.

Приклад 1.7. Зашифруємо методом подвійної перестановки фразу

ХЛІБ УСЬОМУ ГОЛОВА

Для зашифрування обираються ключі: по стовпцях – 4 1 3 2, по рядках – 3 1 4 2. Текст вписується у таблицю необхідного розміру, приміром, по рядках без пробілів і розділових знаків (у даному випадку, для шифрування тексту, що містить 16 символів, достатній розмір таблиці 4×4).

	4	**1**	**3**	**2**
3	Х	Л	І	Б
1	У	С	Ь	О
4	М	У	Г	О
2	Л	О	В	А

Перше перетворення перестановки здійснюється по стовпцях за ключем 4132. Результат перестановки стовпців показано у таблиці нижче.

	1	**2**	**3**	**4**
3	Л	Б	І	Х
1	С	О	Ь	У
4	У	О	Г	М
2	О	А	О	Л

Перетворення здійснюється тепер по рядках з ключем 3142.

	1	**2**	**3**	**4**
1	С	О	Ь	У
2	О	А	О	Л
3	Л	Б	І	Х
4	У	О	Г	М

Таким чином, при зчитуванні по рядках, отримуємо шифртекст

СОЬУОАОЛЛБІХУОГМ.

Кількість можливих варіантів перестановки швидко змінюється в залежності від розміру таблиці. Так, для таблиці, розміром 3×3 існує 3!×3! = 36 варіантів перестановки; для таблиці, розміром 4×4 існує 4!×4! = 576 варіантів перестановки; а таблиці, розміром 5×5 існує, відповідно вже 5!×5! = 144000 варіантів перестановки.

У середині XVI сторіччя в Італії з'являється книга математика, лікаря й філософа Дж. Кардано "Про тонкощі" з доповненням "Про різні речі", в якій є розділи, присвячені криптографії. У ній набули відбиття нові ідеї криптографії: використовування частини найчастіш передаваного відкритого тексту як ключа до шифру і новий спосіб шифрування, котрий увійшов до історії як ґрати Кардано. Для виготовлення ґрат брався лист із твердого матеріалу (картон, пергамент, метал), котрий являв собою квадрат, в якому вирізано "вікна". При шифруванні ґрати накладалися на аркуш паперу і літери відкритого тексту вписувалися у "вікна". По заповненні всіх "вікон" ґрати повертались на 90° – знову літери відкритого тексту вписувалися у "вікна" повернених ґрат. Потім знову здійснювалось повертання на 90° і т. д. За один "захід" ґрати працювали чотири рази. Якщо текст було зашифровано не цілковито, то ґрати ставилися у вихідне положення – і вся процедура повторювалася. Запропонований Кардано шифр-ґрати покладено в основу славнозвісного шифру Рішельє, в якому шифртекст мав вигляд "безневинного" послання. Зрозуміло, використовування цього шифру спричинює утруднення й вимагає інтелекту певного рівня.

Головна вимога до ґрат Кардано: за всіх повертань "вікна" не повинні потрапляти на одне й те саме місце в квадраті, в якому утворюється шифртекст.

Якщо в квадраті після зняття ґрат виникали порожні місця, то в них вписувалися довільні літери. Потім літери квадрата виписувалися порядково, що й становило шифртекст.

Шифр «Поворотні грати» є різновидом шифру маршрутної перестановки. Для використовування шифру, називаного поворотними ґратами, виготовляється трафарет з прямокутного аркуша паперу розміру $2m \times 2k$ клітинок. У трафареті вирізано mk клітинок у такий спосіб, що при накладанні його на чистий аркуш паперу того самого розміру у чотири можливих способи його вирізи цілковито покривають усю площу аркуша.

Літери повідомлення послідовно вписуються у вирізи трафарету (по рядках, у кожнім рядку в напрямку ліворуч→праворуч) при кожнім з чотирьох його можливих положень у заздалегідь установленому порядку.

Пояснімо процес зашифрування на прикладі 1.8.

Приклад 1.8 Зашифруємо методом поворотних грат текст повідомлення

СКАЖИ МЕНІ ХТО ТВІЙ ДРУГ І Я СКАЖУ ТОБІ ХТО ТИ

Нехай за ключ використовуються ґрати 6×6, які наведено на рис. 1.1. Зашифруємо за їхньою допомогою текст.

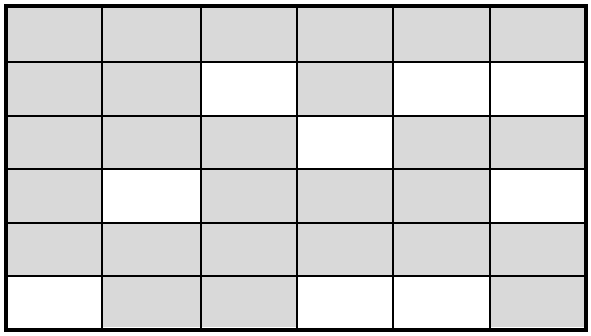


Рисунок 1.1 – Зразок грат

Наклавши ґрати на аркуш паперу, вписуємо перші дев’ять (за кількістю вирізів) літер повідомлення: СКАЖИ МЕНІ …

Знявши ґрати, ми побачимо текст, поданий на рис. 1.2.

Повернемо ґрати на 180°. У „віконечках” з’являться нові, ще не заповнені клітинки. Вписуємо в них наступні дев’ять літер.

Вийде запис, наведений на рис. 1.3. Потім повертаємо ґрати на інший бік і зашифровуємо залишок тексту в аналогічний спосіб (рис. 1.4 та 1.5).

		С		К	А
			Ж		
	И				М
Е			Н	І	

Рисунок 1.2 – Вигляд таблиці після першого кроку зашифровування

Х	Т	С	О	К	А
		Т	Ж		
В	И			І	М
Е	Й	Д	Н	І	Р

Рисунок 1.3 – Вигляд таблиці після другого кроку зашифровування

	У	Г			І
Х	Т	С	О	К	А
Я		Т	Ж	С	
В	И	К		І	М
А	Ж		У		
Е	Й	Д	Н	І	Р

Рисунок 1.4 – Вигляд таблиці після третього кроку зашифровування

Т	У	Г	О	Б	І
Х	Т	С	О	К	А
Я	І	Т	Ж	С	Х
В	И	К	Т	І	М
А	Ж	О	У	Т	И
Е	Й	Д	Н	І	Р

Рисунок 1.5 – Остаточний вигляд таблиці

Одержувач повідомлення, котрий має точно такі самі ґрати, без утруднень прочитає вихідний текст, накладаючи трафарет на шифртекст по порядку у чотири способи.

Можна довести, що кількість можливих трафаретів, тобто кількість ключів шифру "ґрати", становить $\overline{K} = 4^{mk}$. Цей шифр призначено для повідомлень довжини $n = 4mk$. Кількість всіх переставлянь у тексті такої довжини становитиме $(4mk)!$, що в багато разів більше за кількість $\overline{K}$. Однак вже за розміру трафарету 8×8 кількість можливих ґрат перевершує чотири мільярди.

1.1.3 Практичне застосовування шифрів перестановки в системах зв'язку

Практичну реалізацію шифру перестановки в системах зв'язку може бути подано на такому прикладі.

Особливістю телефонного зв'язку є те, що акустичний сигнал у телефонному терміналі перетворюється на електричний і потім, після опрацьовування й посилення, передається лініями зв'язку. На приймальному кінці електричний сигнал знову перетворюється на акустичний; при цьому вихідна форма сигналу більш-менш зберігається. Акустичний і, відповідно, електричний сигнали характеризуються частотним спектром. Їх можна розглядати в розгорненні в часі й за спектром.

Відомі є кілька типових перетворювань аналогового сигналу, котрі може бути легко реалізовано інженерними методами. Зазначимо головні з них.

Перестановка частот. За допомогою системи фільтрів уся ширина смуги стандартного телефонного каналу може бути поділена на певну кількість частотних смуг, котрі потім може бути переставлено поміж собою (рис. 1.6, 1.7).

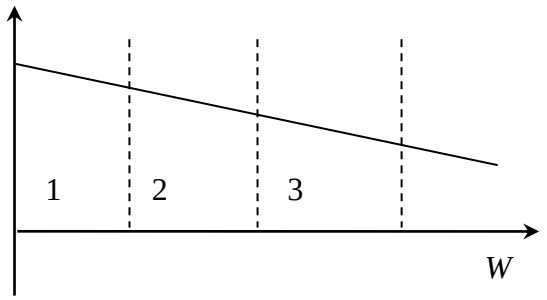


Рисунок 1.6 – Поділення ширини смуги

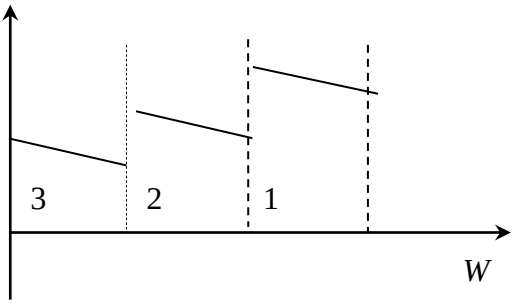


Рисунок 1.7 – Перестановка частотних смуг

Найпростіший скремблер обмежує захист уведенням подібних найпростіших частотних перетворювань.

Інвертування спектра. Більш складні скремблери додатково здійснюють інвертування спектра (рис. 1.8).

Частотно-часові перестановки. Ще більш складні системи розбивають сигнал на часовому інтервалі 60...500 мс і на кожнім інтервалі використовують у комбінації власні аналогові перетворювання. Змінюванням перетворювань у різні часові інтервали керує послідовність, яка надходить на вузол накладання шифру з блока ускладнювання. Той, хто просто послухає дешифроване аналоговим

сигналом мовлення, почує якийсь булькіт, шум. Якщо забути про інверсії, то ми маємо шифр перестановки.

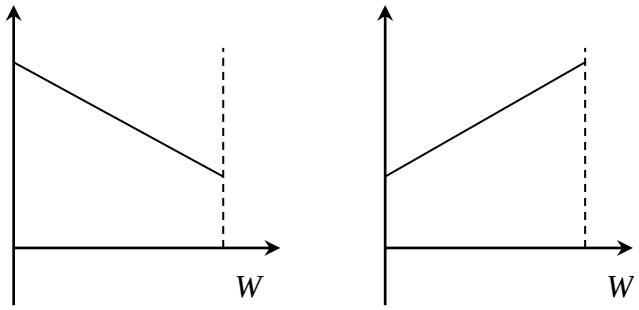


Рисунок 1.8 – Інвертування спектра

Для гарантованого засекречування телефонного мовлення його спочатку оцифровують – переводять у двійкову послідовність, а потім опрацьовують так само, як і будь-яке текстове повідомлення.

На практиці використовується апаратура як аналогового, так і цифрового засекречування. Цифрова апаратура забезпечує гарантовану надійність захисту, але вона є більш вимоглива до каналів зв'язку. Аналогова апаратура є менш стійка, але більш дешева, більш портативна, менш вимоглива до каналів зв'язку.

1.1.4 Завдання для самоперевірки з теми

Завдання 1.1.1. Розшифруйте повідомлення Я'ВОЗКЗ, здобуте внаслідок перетворювання за допомогою підставляння

1	2	3	4	5	6	7
4	3	2	6	1	7	5

Завдання 1.1.2. Зашифруйте за допомогою „магічного квадрата" 4×4

16	3	2	13
5	10	11	8
9	6	7	12
4	15	14	1

фразу: БЕРЕЖІТЬ ДРУЗІВ.

Завдання 1.1.3. Розшифруйте повідомлення

ОИРМ ЕОСЮ ВТАЬ ЛГОП

здобуте за допомогою „магічного квадрата" 4×4 (див. завд. 1.1.2).

Завдання 1.1.4. Зашифруйте методом подвійної перестановки повідомлення

ПОЧИНАТИ ПЕРЕДАЧУ

Ключем до шифру слугує послідовність номерів стовпців 4132 й номерів рядків 3142 вихідної таблиці. Відкритий текст вписувати та шифртекст зчитувати порядково блоками по чотири літери.

Завдання 1.1.5. Шифртекст

ЛЖОЕ НФЯН ЖРЕД ВЕЛИ

здобуто методом подвійної перестановки з ключами, наведеними у завданні 1.1.4. Віднайти вхідне повідомлення. Шифртекст вписувати порядково блоками по чотири літери.

Контрольні питання з теми

1. У чому полягає сутність шифрів перестановки?
2. Охарактеризуйте різновиди шифрів перестановки.
3. В чому полягає зміст шифру маршрутної перестановки?
4. Що є ключем шифру Сцитала? Кому приписують винахід дешифрувального пристрою цього шифру?
5. Як виконується процедура зашифровування за допомогою „магічного квадрата”?
6. Який метод шифрування називається подвійною перестановкою?
7. Що в шифрувальних таблицях використовується в якості ключа?

1.2 Шифри простої заміни

Шифри простої заміни (одноабеткова підстановка) – це простий метод перетворювань, який містить правило заміни символів вихідного тексту на інші символи тої самої абетки за ключем.

Шифр заміни є найпростішим та найбільш популярним шифром. Як випливає з самої назви, шифр заміни здійснює перетворювання (заміну) літер чи інших "частин" відкритого тексту на аналогічні "частини" шифрованого тексту. Легко навести математичний опис шифру заміни. Нехай X и Y – дві абетки (відкритого й зашифрованого тексту відповідно), котрі складаються з однакової кількості символів. Нехай також $g : X \to Y$ – взаємно однозначне відбиття X в Y. Тоді шифр заміни впливає у такій спосіб: відкритий текст $x_1x_2\ldots x_n$ перетворюється на зашифрований текст $g(x_1)\, g(x_2)\ldots g(x_n)$.

Як відкритий текст, так і шифртекст утворюються з літер, котрі входять до скінченної множини символів, називаних *абеткою*. Прикладами абеток є скінченна множина усіх великих літер, скінченна множина усіх великих і малих літер та цифр тощо.

У загальному вигляді певну абетку $\sum$ можна подати в такий спосіб:

$$\sum = \{a_0 + a_1 + a_2 + \ldots + a_{m-1}\}.$$

Поєднуючи за певним правилом літери з абетки $\sum$, можна створити нові абетки:

– абетку $\sum^2$, яка має m^2 біграм a_0a_0, a_0a_1, …, $a_{m-1}a_m$;

– абетку $\sum^3$, яка має m^3 триграм $a_0a_0a_0$, $a_0a_0a_1$, …, $a_{m-1}a_{m-1}a_m$.

Тоді, по'єднуючи по n літер, дістаємо абетку $\sum^n$, яка має m^n n-грам.

Приміром, англійська абетка

$$\sum = \{\text{ABCDEFGH … WXYZ}\},$$

яка містить $m = 26$ літер, дозволяє згенерувати за допомогою операції конкатенації абетку з $26^2 = 676$ біграм

$$\text{AA, AB, …, YZ, ZZ,}$$

абетку з $26^3 = 17576$ триграм

$$\text{AAA, AAB, …, ZZY, ZZZ}$$

тощо.

При виконанні криптографічних перетворювань корисно замінювати літери абетки на цілі числа – 0, 1, 2, 3, ... Це дозволяє спростити виконання необхідних алгебричних маніпуляцій.

Приміром, можна встановити взаємно однозначну відповідність:

– поміж українською абеткою

$$\sum_{\text{укр}} = \{\text{АБВГҐД … ЮЯ}\}$$

та множиною цілих

$$\overline{Z}_{33} = \{0, 1, 2, 3, \ldots, 32\};$$

– поміж англійською абеткою

$$\sum_{\text{англ}} = \{\text{ABCDEF … YZ}\}$$

та множиною цілих

$$\overline{Z}_{26} = \{0, 1, 2, 3, \ldots, 25\}.$$

Надалі буде зазвичай використовуватися абетка $\overline{Z}_m = \{0, 1, 2, 3, \ldots, m-1\}$, яка містить m „літер” (у вигляді чисел, табл. 1.1, 1.2 та 1.3).

Заміна літер традиційної абетки на числа дозволяє більш чітко сформулювати основні концепції та прийоми криптографічних перетворювань. Водночас у більшості ілюстрацій використовуватиметься абетка природної мови.

Таблиця 1.1 – Відповідність поміж українською абеткою та множиною цілих $\overline{Z}_{33} = \{0, 1, 2, 3, \ldots, 32\}$

Літера	Число	Літера	Число	Літера	Число	Літера	Число
А	0	З	9	О	18	Ч	27
Б	1	И	10	П	19	Ш	28
В	2	І	11	Р	20	Щ	29
Г	3	Ї	12	С	21	Ь	30
Ґ	4	Й	13	Т	22	Ю	31
Д	5	К	14	У	23	Я	32
Е	6	Л	15	Ф	24		
Є	7	М	16	Х	25		
Ж	8	Н	17	Ц	26		

Таблиця 1.2 – Відповідність поміж російською абеткою та множиною цілих $\overline{Z}_{32} = \{0, 1, 2, 3, \ldots, 31\}$

Літера	Число	Літера	Число	Літера	Число	Літера	Число
А	0	И	8	Р	16	Ш	24
Б	1	Й	9	С	17	Щ	25
В	2	К	10	Т	18	Ь	26
Г	3	Л	11	У	19	Ы	27
Д	4	М	12	Ф	20	Ъ	28
Е	5	Н	13	Х	21	Э	29
Ж	6	О	14	Ц	22	Ю	30
З	7	П	15	Ч	23	Я	31

Таблиця 1.3 – Відповідність поміж англійською абеткою та множиною цілих $\overline{Z}_{26} = \{0, 1, 2, 3, \ldots, 25\}$

Літера	Число	Літера	Число	Літера	Число
A	0	J	9	S	18
B	1	K	10	T	19
C	2	L	11	U	20
D	3	M	12	V	21
E	4	N	13	W	22
F	5	O	14	X	23
G	6	P	15	Y	24

H	7	Q	16	Z	25
I	8	R	17		

Текст з n літерами абетки $\overline{Z}_m$ можна розглядати як n-граму

$$\overline{x} = (x_0, x_1, x_2, \ldots, x_{n-1}),$$

де $x_i \in \overline{Z}_m$, $0 \le i < n$, для певного цілого n.

Через $\overline{Z}_{m,n}$ позначатимемо множину n-грам, утворених з літер множини $\overline{Z}_m$.

Криптографічне перетворювання E являє собою сукупність перетворювань

$$E = \{E^{(n)} : 1 \le n < \infty\};$$
$$E^{(n)}: \overline{Z}_{m,n} \to \overline{Z}_{m,n}.$$

Перетворювання $E^{(n)}$ визначає, як кожна n-грама відкритого тексту $\overline{x} \in \overline{Z}_{m,n}$ замінюється на n-граму шифртексту $\overline{y}$, тобто

$$\overline{y} = E^{(n)}(\overline{x}), \overline{x}, \overline{y} \in \overline{Z}_{m,n};$$

при цьому неодмінною є вимога взаємної безваріантності перетворювання $E^{(n)}$ на множину $\overline{Z}_{m,n}$.

Криптографічна система може трактуватися як сімейство криптографічних перетворювань

$$\overline{E} = \{E_K: K \in \overline{K}\},$$

позначених параметром K, називаним *ключем*.

Множина значень ключа утворює ключовий простір $\overline{K}$.

1.2.1 Шифр Цезаря

Історичним прикладом шифру заміни є шифр Цезаря (I ст. до н. е.), описаний істориком давнього Риму Светонієм. Гай Юлій Цезар використовував у своєму листуванні шифр власного винайдення. Стосовно сучасної української мови він полягав у такому. Виписувалась абетка: А, Б, В, Г, …, Я; потім під нею виписувалась та ж сама абетка, але з циклічним зсуненням на три літери ліворуч, що показано у табл. 1.4.

Таблиця 1.4 – Зашифрування за допомогою шифру Цезаря

Порядковий номер букви в алфавіті	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Літера вхідного тексту	**А**	**Б**	**В**	**Г**	**Ґ**	**Д**	**Е**	**Є**	**Ж**	**З**	**И**	**І**	**Ї**	**Й**	**К**	**Л**	**М**
Літера шифртексту	***Г***	***Ґ***	***Д***	***Е***	***Є***	***Ж***	***З***	***И***	***І***	***Ї***	***Й***	***К***	***Л***	***М***	***Н***	***О***	***П***

Порядковий номер букви в алфавіті	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
Літера вхідного тексту	**Н**	**О**	**П**	**Р**	**С**	**Т**	**У**	**Ф**	**Х**	**Ц**	**Ч**	**Ш**	**Щ**	**Ь**	**Ю**	**Я**
Літера шифртексту	***Р***	***С***	***Т***	***У***	***Ф***	***Х***	***Ц***	***Ч***	***Ш***	***Щ***	***Ь***	***Ю***	***Я***	***А***	***Б***	***В***

При зашифровуванні літера **А** замінювалася на літеру **Г**, **В** – замінювалася на **Д**, **У** – на **Ц** й т. д.

Приклад 1.9. Зашифруємо за допомогою шифру Цезаря українською абеткою слово **СІЛЬ**.

Згідно табл. 1.4 заміни Цезаря літера **С** замінювалася на літеру **Ф**, літера **І** замінювалась на літеру **К**, літера **Л** – на **О**, літера **Ь** – на **А**.

Тобто слово **СІЛЬ** перетворювалося на шифртекст **ФКОА**. Одержувач повідомлення **ФКОА** шукав ці літери в нижньому рядку і по літерах над ними відновлював вихідне слово **СІЛЬ**.

Ключем у шифрі Цезаря є величина зсунення нижнього рядка абетки.

Природне розвинення шифру Цезаря є очевидне: нижній рядок дворядкового запису літер абетки може бути з довільним розташуванням цих літер. Якщо в абетковому розташуванні літер у нижньому рядку існує всього 33 варіанти ключів (кількість літер в українській абетці), то за їхнього довільного розташування кількість ключів стає величезною. Вона становить 33! (33 факторіали), тобто приблизно 10^{35}. Цей момент є надто важливий. Якщо неправочинний користувач здогадався чи одержав відомості про використовуваний шифр (а шифри використовуються тривалого часу), то він може спробувати перебрати усі варіанти можливих секретних ключів при дешифруванні перехопленої криптограми. Навряд чи віднайдеться дешифрувальник, котрий навіть сьогодні обрав би цей шлях дешифрування. Однак у часи Цезаря, коли панувала суцільна неграмотність населення, сама можливість побачити осмислене повідомлення за "абракадаброю", навіть складеною зі знайомих літер, здавалася нездійсненною. Принаймні давньоримський історик Светоній не наводить випадків дешифрування переписки Цезаря. Нагадаємо, що сам Цезар усе життя використовував один і той самий ключ (зсунення на три літери). Цим шифром він користувався, зокрема, для обміну посланнями з Ціцероном. Проте, низька криптостійкість цього шифру сприяла його модифікації та подальшому розвитку.

У художній літературі класичним прикладом шифру заміни є відомий шифр "Танцюючі чоловічки" (К. Дойла). У ньому літери тексту замінювалися на

символічні фігурки людей. Ключем такого шифру були постави чоловічків, котрі замінювали літери.

Існували й інші способи захисту інформації, розроблені в античні часи.

1.2.2 Шифрування за допомогою шифрів простої заміни

Одне з перших історичних імен, котре згадується у зв'язку з криптографією, це ім'я Енея – легендарного полководця, захисника Трої. В царині тайнопису Енеєві належать два винаходи.

Перший з винаходів – так званий "диск Енея". Засади його побудови й дії є вельми прості. На диску діаметром 10...15 см і товщиною 1...2 см висвердлювалися отвори за кількістю літер абетки. В центрі диска містилася "котушка" з намотаною на ній ниткою потрібної довжини. При зашифровуванні нитка "витягалася" з котушки і послідовно протягалася через отвори відповідно до літер зашифровуваного тексту. Диск був посланням. Одержувач послання послідовно витягав нитку з отворів, що дозволяло йому зчитувати передаване повідомлення, але у зворотному порядку слідування літер. При перехопленні диска неправочинний користувач мав можливість прочитати повідомлення у той самий спосіб, що й одержувач. Але Еней передбачав можливість легкого знищення передаваного повідомлення в разі загрози захоплення диска. Для запобігання цьому досить було висмикнути "котушку" із закріпленим на ній кінцем нитки до повного виходу всієї нитки з отворів диска.

Ідею Енея було використано при створюванні й інших оригінальних шифрів заміни. Приміром, в одному з варіантів замість диска використовувалася лінійка з кількістю отворів, дорівнюваних кількості літер абетки. Кожен отвір позначався власною літерою; літери по отворах розташовувалися в довільному порядку. До лінійки було прикріплено котушку з намотаною на неї ниткою. Поруч з котушкою був проріз. При зашифруванні нитка протягалася через проріз, а потім через отвір, котрий відповідав першій літері зашифровуваного тексту, при цьому на нитці зав'язувався вузлик у місці проходження її через отвір; потім нитка поверталася до прорізу – й аналогічно зашифровувалася друга літера тексту й т. д.

Після завершення зашифровування нитка витягалася й передавалася одержувачеві повідомлення. Той, маючи ідентичну лінійку, протягав нитку через прорізи отворів, зумовлених вузлами, і відновлював вихідний текст за літерами отворів. Цей пристрій дістав назву ***лінійки Енея***. Шифр, зреалізовуваний лінійкою Енея, є одним з прикладів шифру заміни: у ньому літери замінюються на певній відстані поміж вузликами на нитці. Ключем шифру був порядок розташування літер по отворах у лінійці. Неправочинний користувач, котрий здобув нитку (навіть маючи лінійку, але без нанесених літер), не міг прочитати передаване повідомлення.

Аналогічне до лінійки Енея так зване "вузелкове письмо" ("кіпу") набуло поширення в індіанців Центральної Америки. Свої повідомлення вони також передавали у вигляді нитки, на якій зав'язувалися різнобарвні вузлики, котрі визначали зміст повідомлення.

Помітним внеском Енея до криптографії є запропонований ним так званий ***книжковий шифр***, описаний у творі "Про оборону укріплених місць". Еней запропонував проколювати малопомітні дірки в книзі чи в іншому документі над літерами таємного повідомлення. Варто відзначити, що в першій світовій війні минулого сторіччя німецькі шпигуни використовували аналогічний шифр, замінивши дірки на крапки, які наносилися спеціальними чорнилами на літери газетного тексту.

Винайдення друкарства Йоганном Гуттенбергом (1440, Німеччина, м. Майнц) помітно позначилось на підвищенні грамотності населення. Пожвавішало листування, став зростати обсяг обміну секретною інформацією. З іншого боку, доступні для всіх книги самі собою спричинилися до застосовування книжкових шифрів, використовуваних і до сьогодні.

Суть книжкового шифру полягає в заміні літер на номер рядка і номер цієї літери в рядку в заздалегідь обумовленій сторінці певної книги. Ключем такого шифру є сама книга й використовувана сторінка в ній. Існує чимало способів використовування книги для таємного обміну повідомленнями. Приміром, якщо адресати заздалегідь домовилися поміж собою про використовування дублікатів однієї й тієї самої книги як ключа шифру, то їхні таємні послання могли б складатися з таких елементарних одиниць: $n|m|t$, де n – номер сторінки книги, m – номер рядка, t – номер літери в рядку; по цих літерах і читається таємне послання. Поряд з нумерацією літер можуть використовуватися позначення слів і навіть цілих фраз.

Книжковий шифр став "довгожителем" і застосовувався навіть у часи другої світової війни минулого сторіччя.

Розглянемо метод шифрування за рахунок перетворювання числового повідомлення на літерне

Скористаємося 30-літерною абеткою української мови

АБВГДЕЄЖЗИІКЛМНОПРСТУФХЦЧШЩЬЮЯ.

У поданій тут абетці відсутні літери Ґ, Ї, Й, що практично не обмежує можливостей за складання відкритих повідомлень українською мовою.

В абетці будь-якої природної мови літери слідують одна за одною у певному порядку. Це надає можливість надати кожній літері абетки її природний порядковий номер.

Занумеруємо літери української абетки відповідно до табл. 1.5.

Таблиця 1.5 – Відповідність поміж українською абеткою та множиною цілих

А	Б	В	Г	Д	Е	Є	Ж	З	И	І	К	Л	М	Н
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ь	Ю	Я
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30

Якщо у відкритому повідомленні кожну літеру замінити на її природний порядковий номер у розглядуваній абетці, то перетворювання числового повідомлення на літерне дозволяє безваріантно відновити вихідне відкрите повідомлення. Приміром, числове повідомлення 1 13 22 1 3 11 20 перетвориться на літерне повідомлення: АЛФАВІТ.

Розвинення методів шифрування дає безліч варіантів методів *одноабеткової підстановки.* Нехай повідомлення шифрується українською мовою й при цьому заміні підлягає кожна літера відкритого тексту. Формально в цьому разі шифр заміни можна описати в такий спосіб. Для кожної літери *а* вихідної абетки будується певна множина символів M_a, аби множини M_a та $M_в$ попарно не перетинались за $а \neq в$, тобто будь-які дві різні множини не містили однакових елементів. Множина M_a називається множиною шифропозначань для літери *а.*

Табл. 1.6 є ключем шифру заміни. Знаючи її, можна здійснювати як зашифровування, так і розшифровування.

Таблиця 1.6 – Зашифровування літер відкритого повідомлення за допомогою будь-якого символу з множини M_j

а	*б*	*в*	...	*я*
$M_а$	$M_б$	$M_в$	…	$M_я$

При зашифровуванні кожна літера *а* відкритого повідомлення, розпочинаючи з першої, замінюється на будь-який символ з множини M_a. Якщо в повідомленні міститься кілька літер *а*, то кожна з них замінюється на будь-який символ з M_a. За рахунок цього за допомогою одного ключа (табл. 1.7) можна дістати різноманітні варіанти зашифрованого повідомлення для одного й того самого відкритого повідомлення. Наприклад, якщо ключем є табл. 1.7, то повідомлення «мені відомі шифри заміни» може бути зашифровано кожним з трьох способів, поданих в цій таблиці.

Оскільки множини M_a, $M_б$, $M_в$, …, $M_я$ попарно не перетинаються, то за кожним символом шифрованого повідомлення можна безваріантно визначити, якій множині він належить й, отже, яку саме літеру відкритого повідомлення він замінює. Тому розшифровування є можливе й відкрите повідомлення визначається у єдиний спосіб, це показано у прикладі 1.10

Приклад 1.10. Зашифруємо шифром простої заміни повідомлення

МЕНІ ВІДОМІ ШИФРИ ЗАМІНИ

ключем до шифру служить табл. 1.7.

Таблиця 1.7 – Зашифровування літер відкритого повідомлення за допомогою одного з трьох числових символів множини M_j

а	*б*	*в*	*г*	*д*	*е*	*є*	*ж*	*з*	*и*	*і*	*к*	*л*	*м*	*н*	*о*	*п*
21	37	14	22	01	24	74	62	73	46	65	23	12	08	27	53	35
40	26	63	47	31	83	17	88	30	02	34	91	72	32	77	68	60

10	03	71	82	15	70	42	11	55	90	92	69	38	61	54	09	84

р	*с*	*т*	*у*	*ф*	*х*	*ц*	*ч*	*ш*	*щ*	*ь*	*і*	*ю*	*я*
04	20	13	59	25	75	43	19	29	06	48	36	28	16
44	52	39	07	49	33	85	58	80	50	56	78	64	41
45	89	67	93	76	18	51	87	66	81	79	86	05	57

Варіант зашифрування відкритого повідомлення зведено у табл. 1.8

Таблиця 1.8 – Результат зашифрування

Відкритий текст	М	Е	Н	І	В	І	Д	О	М	І	Ш	И	Ф	Р	И	З	А	М	І	Н	И
Шифртекст	8	24	27	65	14	34	1	53	32	92	29	46	25	4	2	73	21	61	34	77	90

При зашифруванні однакові літери відкритого тексту маскуються різними числами, якими пропонується зашифровувати відповідні літери у табл. 1.5. Цей метод шифрування зменшує частость появи кожної літери в тексті та ускладнює криптоаналіз шифртексту.

Ще один винахід древніх греків – так званий квадрат Полібія (Полібій – грецький державний діяч, полководець, історик, III сторіччя до н. е.). Стосовно сучасної латинської абетки з 26 літер шифрування за цим квадратом здійснюється у такий спосіб. До квадрата розміром 5×5 клітинок виписуються всі літери абетки, при цьому літери I та J не розрізнюються (J ототожнюється з літерою I):

	A	B	C	D	E
A	A	B	C	D	E
B	F	G	H	I	K
C	L	M	N	O	P
D	Q	R	S	T	U
E	V	W	X	Y	Z

Зашифровувана літера замінюється на координати квадрата, в якому її записано. Приміром, B замінюється на AB, F – на BA, R – на DB і т. д. При розшифровуванні кожна така пара визначає відповідну літеру повідомлення. Зауважимо, що секретом у даному разі є сам спосіб замінювання літер. Ключ у цій системі є відсутній, оскільки використовується фіксований порядок слідування літер.

Існував і інший варіант шифрування за допомогою квадрата Полібія. При зашифровуванні в цьому квадраті відшукували чергову літеру відкритого тексту й записували до шифртексту літеру, розташовану нижче за неї в тім самім стовпці. Якщо літера тексту містилася в нижньому рядку таблиці, то для шифртексту брали найверхню літеру з того самого стовпця.

Ускладнений варіант шифру Полібія полягає в записуванні літер до квадрата у довільному (неабетковому) порядку. Цей довільний порядок і є ключем. Тут, однак, виникла й певна незручність. Довільний порядок літер важко запам'ятати, тому користувачеві шифру було необхідно завжди мати при собі ключ – квадрат. Виникла небезпека щодо таємного ознайомлення з цим ключем сторонніх осіб. Як компромісний розв'язок в якості ключа було запропоновано пароль. Легко запам'ятовуваний пароль вписувався без повторювання літер до квадрата; у клітинки, котрі залишилися порожніми, за абеткою вписувалися літери абетки, відсутні в паролі. Приміром, нехай паролем є слово THE TABLE. Тоді квадрат матиме вигляд

T	H	E	A	B
L	C	D	F	G
I	K	M	N	O
P	Q	R	S	U
V	W	X	Y	Z

Такий квадрат уже не треба мати при собі. Досить запам'ятати ключ-пароль. Зауважимо, до речі, що в такий самий спосіб можна запам'ятовувати порядок розташовування літер при використовуванні лінійки Енея, а також шифру заміни Цезаря (за довільного розташовування літер у нижньому рядку). Цікаве употужнення шифру Полібія було запропоновано одним криптографом-аматором вже XIX сторіччя. Зміст цього ускладнення з'ясуємо на прикладі.

Нехай маємо такий квадрат Полібія

	1	2	3	4	5
1	E	K	T	L	B
2	H	I,J	A	D	U
3	M	S	G	C	V
4	F	P	Q	R	W
5	O	Y	X	Z	N

Зашифруємо за ним слово THE APPLE. Дістанемо шифртекст:

13.21.11.23.42.42.14.11. (1.1)

На цьому зашифровування за Полібієм завершено. Це був шифр простої заміни типу шифру Цезаря, в якому кожна літера відкритого тексту замінювалася на певне двознакове десяткове число, і ця заміна не змінювалася по всьому тексту. Кількість ключів цього шифру дорівнює 25!.

Ускладнений варіант полягає в такому. Здобутий первинний шифртекст зашифровується вдруге. При цьому він виписується без розбивання на пари:

1321112342421411 (1.2)

Здобута послідовність цифр зсувається циклічно ліворуч на один крок:

3211123424214111

Ця послідовність знову розбивається на біграми:

32.11.12.34.24.21.41.11

й за таблицею замінюється на остаточний шифртекст:

SEKCDHFE (1.3)

Кількість ключів у цьому шифрі залишається тією самою (25!), але він є вже значно стійкіший. Зауважимо, що цей шифр вже не є шифром простої заміни (літера Е відкритого тексту переходить у різні літери: К, Е; літера Р – у літери D, Н). Було виявлено й негативний момент. Якщо в шифрі простої заміни шифртекст буде написано з однією помилкою (наприклад у тексті (1.1) замість четвертої літери 23 буде написано 32), то розшифрований текст містиме лише одну помилку: THE SPPLE, що вона легко виправляється одержувачем повідомлення. Якщо ж у тексті (1.3) буде спотворено четверту літеру (літеру С замінено, приміром, на К), то в розшифрованому тексті буде вже два спотворення: THE HIPLE, що вже утруднює відновлення вихідного повідомлення.

Аналогічно обстоїть справа з помилками вигляду "пропускання літер". Нехай у тексті (1.3) пропущено літеру С. Шифртекст набере вигляду SEKDHFE, чи

32.11.12.24.21.41.11.

Після розшифровування здобудемо THE IPLE, тобто поряд з пропущенням літери в розширеному тексті наявне й спотворення іншої літери. За пропущення в (1.3) першої літери при розшифровуванні здобудемо ЕЕ APPLE.

Слід зауважити, що в дещо зміненому вигляді шифр Полібія добувся до наших днів і дістав своєрідної назви "тюремний шифр". Для його використовування потрібно знати лише природний порядок розташування літер абетки (як у зазначеному вище прикладі квадрата Полібія для англійської мови). Сторони квадрата позначаються не літерами (ABCDE), а цифрами (12345). Цифра 3, наприклад, передається шляхом потрійного стукоту. При передаванні літери спочатку "відстукується" цифра, котра відповідає рядкові, в якому міститься літера, а потім – номер відповідного стовпця. Приміром, літера F передається подвійним стукотом (другий рядок) і потім – одноразовим (перший стовпець).

Із застосовуванням цього шифру пов’язано певні історичні казуси. Приміром, декабристи, впроваджені до в’язниці після невдалого повстання, не спромоглися встановити зв’язок з князем Одоєвським, який перебував у „одиночці”. Виявилося, що князь (добре освічена для свого часу людина) не пам’ятав природного порядку розташування літер у російській та французькій абетках (іншими мовами він не володів). Декабристи для російської абетки використовували прямокутник розміром 5×6 (5 рядків і 6 стовпців) і скорочену до 30 літер абетку.

„Тюремний шифр”, строго кажучи, – не шифр, а спосіб перекодовування повідомлення з метою його приведення до вигляду, зручного для передавання „каналом зв’язку” (через стінку). Річ у тім, що в таблиці використовувався природний порядок розташування літер абетки. Отже, секретом є сам шифр (а не ключ), як у Полібія.

Як вже зазначалося вище, за один з перших шифрів простої заміни вважається так званий *полібіанський квадрат*. Існує кілька варіантів шифру Полібія. Проте, один з найвідоміших методів шифрування розглянемо зараз.

За два сторіччя до нашої ери Полібій винайшов для цілей зашифровування квадратну таблицю розміром 5×5, заповнену літерами грецької абетки у довільному порядку (рис. 1.9).

λ	ε	υ	ω	γ
ρ	ζ	δ	σ	ο
μ	η	β	ξ	τ
ψ	π	θ	α	ς
χ	ν	-	φ	ί

Рисунок 1.9 – Полібіанський квадрат, заповнений у довільний спосіб 24-ма літерами грецької абетки й прогалиною

При зашифровуванні в цьому квадраті відшукували чергову літеру відкритого тексту й записували до шифртексту літеру, розташовану нижче за неї в тім самім стовпці. Якщо літера тексту містилася в нижньому рядку таблиці, то для шифртексту брали найверхню літеру з того самого стовпця. Приміром, для слова

τ α υ ρ ο σ χ виходить шифртекст ς φ δ μ τ ξ λ.

Концепція полібіанського квадрата виявилася плідною й набула застосовування в криптосистемах подальших часів.

Розглянемо у прикладі 1.11 зашифрування за допомогою квадрата Полібія українською мовою.

Приклад 1.11. Зашифруємо повідомлення СОНЯЧНЕ СЯЙВО за допомогою полібіанського квадрата, який показано на рис.1.10

Е	І	М	З	К	Ь
Н	Ч	Є	Ц	У	Ґ
Ж	И	Ш	Ї	Я	Ю
Р	П	В	Л	Ф	*
О	Б	Щ	Г	Х	+
А	Т	Й	С	Д	-

Рисунок 1.10 – Полібіанський квадрат для української абетки

Результат зашифрування відкритого повідомлення зведено у табл. 1.9

Таблиця 1.9 – Результат зашифрування

Відкритий текст	С	О	Н	Я	Ч	Н	Е	С	Я	Й	В	О
Шифртекст	З	А	Ж	Ф	И	Ж	Н	З	Ф	М	Щ	А

Ключем даного шифру є таблиця, що заповнена у довільний спосіб ЗЗ-ма літерами української абетки й пустушками.

У XV сторіччі абат Тритемій (Німеччина) зробив дві новаторські пропозиції в царині криптографії: він запропонував шифр "Аве Марія" й шифр, на підставі ключа, котрий періодично зсувається.

Шифр "Аве Марія" ґрунтовано на засаді заміни літер зашифровуваного тексту на заздалегідь обумовлені слова. З цих слів складалося зовнішньо "безневинне" повідомлення. Наведемо приклад.

Замінимо літери Н, І на такі слова:

Н = *ЗЕЛЕНИЙ, МІЙ, БІЛЯ*;
І = *КЛЮЧ, МОРЕ, ГАЙ,*

тоді негативна секретна відповідь НІ на задане запитання може мати кілька "безневинних" варіантів: *Біля моря, Мій ключ, Зелений гай.*

Найбільш вагома пропозиція Тритемія щодо захисту інформації, котра дійшла до наших днів, полягає у створеній ним таблиці – таблиці Тритемія. Еквівалент її для англійської абетки наведено в додатку А. Перший рядок цієї таблиці є водночас і рядком літер відкритого тексту. Перша літера тексту зашифровується за першим рядком, друга літера – за другим рядком і т. д.; після використовування останнього рядка – знову повертаються до першого рядка. Приміром, слово "fight" (боротьба) набуває вигляду "fjikx".

Зреалізовування таблиці Тритемія не потребувало використовування якихось механічних пристосувань; шифрабетка з кожним кроком зашифровування зсувається на одиницю ліворуч. Однак у первинному варіанті в шифрі Тритемія був відсутній ключ. Секретом був сам спосіб шифрування. Надалі ускладнювання шифру пішло двома шляхами:

– упровадження довільного порядку розташовування літер вихідної абетки зашифровуваного тексту замість лексикографічно упорядкованої абетки;

– застосовування ускладнюваного порядку вибору рядків таблиці при зашифровуванні.

Ці ускладнення дозволили застосовувати ключові множини значного обсягу.

Відзначимо, що шифр простої заміни є варіантом шифру Тритемія: у ньому всі літери зашифровуються за одним і тим самим рядком таблиці.

Наступний крок у розвиненні запропонованого Тритеміем способу шифрування було зроблено італійцем Джованні Белазо. 1553 року виходить друком його брошура "Шифр сеньйора Белазо". У цьому шифрі ключем був так званий пароль – легко запам’ятовуванні фраза чи слово. Пароль записувався періодично над літерами відкритого тексту. Літера пароля, розміщена над відповідною літерою відкритого тексту, зазначала номер рядка в таблиці Тритемія, за якою треба було проводити заміну (зашифровування) цієї літери. Аналогічні ідеї щодо зашифровування використовуються й сьогодні.

Ще одною модифікацією шифру є система шифрування Цезаря з ключовим словом є одноабетковою системою підставляння. Особливістю цієї системи є використовування ключового слова для зсунення та змінювання порядку символів в абетці підставляння.

Оберемо певне число k, $0 \le k < 25$, і слово чи коротку фразу за ключове слово. Бажано, щоби усі літери ключового слова були різні. Оберемо слово DIPLOMAT за ключове й число $k = 5$.

Ключове слово записується під літерами абетки, розпочинаючи з літери, числовий код якої збігається з обраним числом k:

```
0  1  2  3  4  5              10              15              20              25
A  B  C  D  E  F  G  H  I  J  K  L  M  N  O  P  Q  R  S  T  U  V  W  X  Y  Z
          D  I  P  L  O  M  A  T
```

Решта літер абетки підставляння записуються після ключового слова за абеткою:

```
                5                 12
A  B  C  D  E  F  G  H  I  J  K  L  M  N  O  P  Q  R  S  T  U  V  W  X  Y  Z
V  W  X  Y  Z  D  I  P  L  O  M  A  T  B  C  E  F  G  H  J  K  N  Q  R  S  U
```

Тепер маємо підстановку для кожної літери довільного повідомлення.

Вихідне повідомлення

SEND MORE MONEY

перетворюється на шифртекст

HZBY TCGZ TCBZS

Слід зауважити, що вимога щодо розходження всіх літер ключового слова не є обов'язковою. Можна просто записати ключове слово (чи фразу) без повторення однакових літер, що продемонструємо у прикладі 1.12.

Безперечним достоїнством системи Цезаря з ключовим словом є те, що кількість можливих ключових слів є практично невичерпна.

Недоліком цієї системи є можливість зламування шифртексту на підставі аналізу частості з'являння літер.

Приклад 1.12 Зашифруємо слово ЩАСТЯ за допомогою шифру Цезаря з ключовою фразою ДРУЖБА ТА БРАТСТВО НАЙБІЛЬШЕ БАГАТСТВО і число $k = 3$ породжують таблицю підставляння, подану табл. 1.10.

Таблиця 1.10 – Створення таблиці заміни літер за допомогою ключової фрази

Порядковий номер літери в алфавіті	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Літера вхідного тексту	А	Б	В	Г	Ґ	Д	Е	Є	Ж	З	И	І	Ї	Й	К	Л	М
Літера шифртексту	Щ	Ю	Я	**Д**	**Р**	**У**	**Ж**	**Б**	**А**	**Т**	**С**	**В**	**О**	**Н**	**Й**	**І**	**Л**

Порядковий номер літери в алфавіті	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
Літера вхідного тексту	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ь	Ю	Я
Літера шифртексту	**Ь**	**Ш**	**Е**	**Г**	Ґ	Є	З	И	Ї	К	М	П	Ф	Х	Ц	Ч

Ще одним різновидом шифру Цезаря є шифрування за допомогою афінної системи. У системі шифрування Цезаря використовувалися лише адитивні властивості цілих $\overline{Z}_m$. Однак символи множини $\overline{Z}_m$ можна також перемножувати за модулем m. Застосовуючи водночас операції додавання та множення за модулем m над елементами множини $\overline{Z}_m$, можна здобути систему підстановки, що її називають афінною системою підстановки Цезаря.

Визначимо перетворювання в такій системі:

$$E_{a,b}: \overline{Z}_m \to \overline{Z}_m;\ E_{a,b}: t \to E_{a,b}(t);\ E_{a,b}(t) \equiv at + b \pmod m,$$

де a, b – цілі числа, $0 \le a, b < m$, НСД $(a, m) = 1$ (НСД – найбільший спільний дільник).

У даному перетворюванні літера, котра відповідає числу t, замінюється на літеру, котра відповідає числовому значенню ($at + b$) за модулем m.

Слід зауважити, що перетворювання $E_{a,b}(t)$ є взаємно однозначним відбиттям на множину $\overline{Z}_m$ лише в тому разі, якщо найбільший спільний дільник чисел a та m дорівнює одиниці, тобто a та m повинні бути взаємно простими числами. Приміром, нехай $m = 26$, $a = 3$, $b = 5$, НСД (3,26) = 1 і можна здобути відповідність поміж числовими кодами літер, подану в табл. 1.11.

Якщо перетворити числа на літери англійської мови, дістанемо відповідність для літер відкритого тексту та шифртексту, подану в табл. 1.12.

Таблиця 1.11 – Відповідність поміж числовими кодами літер

t	0	1	2	3	4	5	6	7	8	9	10	11	12
$3t + 5$	5	8	11	14	17	20	23	0	3	6	9	12	15

t	13	14	15	16	17	18	19	20	21	22	23	24	25
$3t + 5$	18	21	24	1	4	7	10	13	16	19	22	25	2

Таблиця 1.12 – Відповідність для літер англійської мови відкритого тексту та шифртексту

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
F	I	L	O	R	U	X	A	D	G	J	M	P	S	V

P	Q	R	S	T	U	V	W	X	Y	Z
Y	B	E	H	K	N	Q	T	W	Z	C

Вхідне повідомлення НОРЕ перетворюється на шифртекст AVYR.

Достоїнством афінної системи є зручне керування ключами: ключі зашифровування й розшифровування подаються в у вигляді пари чисел (a, b).

Недоліки афінної системи є аналогічні до недоліків системи шифрування Цезаря.

Приклад 1.13. Зашифруємо слово КРИПТОГРАФІЯ за допомогою афінної системи підставляння Цезаря для української абетки за умовами:

$$E_{a,b}(t) \equiv at + b \pmod{m}, \text{ де } m = 31,\ a = 3,\ b = 5.$$

Зведемо у табл. 1.13 розраховану відповідність номеру літери відкритого тексту t до номеру літери у шифртексті за формулою $(3t + 5) \bmod 31$

Таблиця 1.13 – відповідність поміж числовими кодами літер відкритого та зашифрованого текстів

t	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$3t + 5$	5	8	11	14	17	20	23	26	29	1	4	7	10	13	16	19
t	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	
$3t + 5$	22	25	28	0	3	6	9	12	15	18	21	24	27	30	2	

Якщо перетворити числа на літери української мови, дістанемо відповідність для літер відкритого тексту та шифртексту, подану в табл. 1.14.

Таблиця 1.14 – Відповідність для літер української мови відкритого тексту та шифртексту

Порядковий номер літери в алфавіті	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Літера	А	Б	В	Г	Д	Е	Є	Ж	З	И	І	Й	К	Л	М	Н
Порядковий номер літери в алфавіті	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	
Літера	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ь	Ю	Я	

У поданій тут абетці відсутні літери Ґ, Ї, що практично не обмежує можливостей за складання відкритих повідомлень українською мовою. Згідно табл..1.13 та 1.14 зашифруємо вихідне слово та отримаємо шифртекст:

Вхідний текст	К	Р	И	П	Т	О	Г	Р	А	Ф	І	Я
Шифртекст	І	Ь	Б	Ч	Г	Ф	М	Ь	Е	И	Д	В

Для більш надійного шифрування можна здійснювати подвійне шифрування. Перетворювання шифру простої заміни в абетці $A = \{a_1, a_2, \ldots, a_n\}$, яка складається з n різних літер, полягає в заміні кожної літери тексту, який треба зашифрувати, на літеру тієї самої абетки, причому різні літери замінюються на різні. Ключем до шифру простої заміни є табл. 1.15, в якій зазначено, на яку саме літеру треба замінити кожну літеру української абетки A.

Таблиця 1.15 – Зашифровування літер за допомогою шифру простої заміни

Літера вхідного тексту	А	Б	В	Г	Ґ	Д	Е	Є	Ж	З	И	І	Ї	Й	К	Л	М
Літера шифртексту	Ч	Я	Ю	Е	Є	Ь	Щ	Ш	Ц	Х	Ф	У	Б	Д	Т	З	В

Літера вхідного тексту	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ь	Ю	Я
Літера шифртексту	Р	П	Л	А	Ґ	О	Ж	И	Ї	К	М	С	Г	Н	І	Й

Багаторазове зашифрування за допомогою шифрувальної таблиці показано у прикладі 1.14

Приклад 1.14. Зашифруємо слово ТЕРМІНОВО простою заміною за допомогою табл. 1.15, що є ключем шифрування та розрахуємо в якому циклі вперше з'явиться вихідне слово, якщо процес зашифрування продовжувати необмежено.

Після першого циклу зашифрування вийде слово ОЩАВУРПЮП.

Зашифрувавши здобуте слово за допомогою того самого ключа вдруге, здобудемо слово ПГЧЮЖАЛІЛ.

Розрахуємо скільки всього всіляких слів можна здобути, якщо зазначений процес зашифровування продовжувати необмежено. Оскільки розглядуваний шифр має таку властивість, що різні літери замінюються на різні, то при зашифровуванні різних слів виходять різні слова. З іншого боку, однакові літери замінюються на однакові незалежно від циклу зашифровування, тому що використовується один і той самий ключ. Отже, при зашифровуванні однакових слів виходять однакові слова, тобто кількість різних слів, які можна здобути в зазначеному процесі зашифровування з початковим словом ТЕРМІНОВО, збігається з найменшим номером циклу зашифровування, котрий дає це початкове слово.

Оскільки літера Т повторюється в кожнім циклі зашифровування, номер якого дорівнює 24 (Т-О-П-Л-З-Х-Ї-Б-Я-Й-Д-Ь-Н-Р-А-Ч-М-В-Ю-І-У-Ж-Ц-К-Т), літера Е змінюється в кожнім циклі з номером 3 (Е-Щ-Г-Е), а літери Р, М, І, Н, О, В – у циклах за номером 24, то слово ТЕРМІНОВО з'явиться вперше в циклі з номером, порівнюваним

$$НСК(3, 24) = 24$$

де НСК – найменше спільне кратне.

Цікавим є шифр „братерства франкмасонів", чи „вільних каменярів", що його вони використовували для спілкування поміж собою. За сучасними поняттями і всупереч поширеній думці, зовсім не є стійкий, але становить певний інтерес. Наведемо невеликий приклад (стосовно англійської мови). Нарисуємо три фігури такого вигляду:

A:	**B:**	**C:**
D:	**E:**	**F:**
G:	**H:**	**I:**

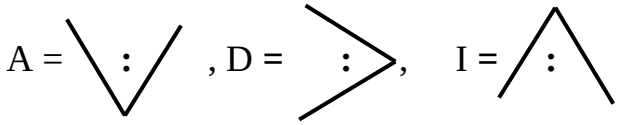
J.	**K.**	**L.**
M.	**N.**	**O.**
P.	**Q.**	**R.**

S	**T**	**U**
V	**W**	**X**
Y	**Z**	

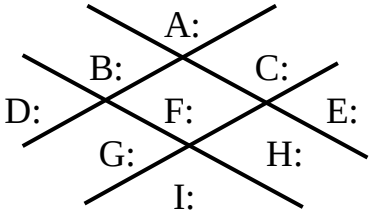
Відповідно до цих ключових фігур літери набувають такого геометричного подання:

Фраза "We talk about" при зашифровуванні набуває вигляду

Геометричне подання може змінюватися, приміром, таким чином



тоді



Варто зауважити, що при поході на Росію Наполеон використовував у нижчих ланках свого зв’язку подібні шифри. Їх було розкрито, що вельми вплинуло на перебіг бойових дій.

Дуже зручним методом шифрування є шифрувальні таблиці Трисемуса. В 1508 році абат з Німеччини Йоганн Трисемус надрукував працю з криптології за назвою "Поліграфія". У цій книзі він уперше систематично описав застосовування шифрувальних таблиць, заповнюваних абеткою у довільному порядку. Для здобуття такого шифру заміни зазвичай використовувались таблиці для записування літер абетки й ключове слово (чи фраза). У таблицю спочатку вписувалося по рядках ключове слово, причому повторювані літери відкидалися. Потім ця таблиця доповнювалася літерами абетки, які не ввійшли до неї, одна за одною. Оскільки ключове слово чи фразу легко зберігати в пам’яті, то такий підхід спрощував процеси зашифровування чи розшифровування.

При шифруванні відшуковують у цій таблиці чергову літеру відкритого тексту й записують до шифртексту літеру, розташовану нижче за неї в тім самім стовпці. Якщо літера тексту міститься в нижньому рядку таблиці, тоді для шифртексту беруть першу верхню літеру з того ж самого стовпця.

Зашифрування за допомогою таблиці заміни Трисемуса можна показати на прикладі 1.15.

Приклад 1.15. Зашифруємо за допомогою таблиці Трисемуса повідомлення

ДІЛОМАЙСТРА ВЕЛИЧАЄ

Оберемо за ключ слово БАНДЕРОЛЬ.

Для української абетки таблиця Трисемуса може мати розмір 4×8. Розглянемо шифрувальну таблицю з ключем, поданим у табл. 1.16.

Таблиця 1.16 – Шифрувальна таблиця Трисемуса з ключовим словом

Б	А	Н	Д	Е	Р	О	Л
Ь	В	Г	Є	Ж	З	И	І
Ї	Й	К	М	П	С	Т	У
Ф	Х	Ц	Ч	Ш	Щ	Ю	Я

Як і в разі полібіанського квадрата, при зашифровуванні відшукують у цій таблиці чергову літеру відкритого тексту й записують до шифртексту літеру, розташовану нижче за неї в тім самім стовпці. Якщо літера тексту міститься в нижньому рядку таблиці, тоді для шифртексту беруть першу верхню літеру з того ж самого стовпця.

При зашифровуванні дістаємо шифртекст

ЄУІИЧВХЩЮЗВЙЖІТДВМ.

Такі табличні шифри називаються *монограмними*, тому що шифрування виконується за однією літерою.

Крім монограмних шифрів є шифри, що потребують перед шифруванням вихідного тексту розбивання повідомлення на біграми (по дві літери). Одним з таких шифрів є біграмний шифр Плейфейра.

Основою шифру Плейфейра є шифрувальна таблиця з випадково розташованими літерами абетки вихідних повідомлень.

У цілому структура таблиці системи шифрування Плейфейра є майже аналогічна до структури таблиці Трисемуса.

Процедура зашифровування містить такі вимоги:

Відкритий текст вихідного повідомлення розбивається на пари літер (біграми). Текст повинен мати парну кількість літер, і в ньому не повинно бути біграм, котрі містили б дві однакові літери. Якщо цих вимог не додержано, то текст змодифіковується навіть через незначні орфографічні помилки.

Послідовність біграм відкритого тексту перетворюється за допомогою шифрувальної таблиці на послідовність біграм шифртексту за такими правилами:

– якщо дві літери відкритого тексту не потрапляють до одного рядка чи стовпця шифрувальної таблиці, тоді відшукують літери в кутах прямокутника, визначуваного даною парою літер. Послідовність літер у біграмі шифртексту має бути дзеркально розташована стосовно послідовності літер у біграмі відкритого тексту (як саме це буде відбуватися, заздалегіть домовляються відправляч та одержувач повідомлення). В нашому прикладі заміна букв біграми буде здійснюватись по рядку першої букви біграми вихідного тексту, тобто перша буква біграми шифртексту має бути в тому ж рядку, що і буква біграми вихідного тексту;

– якщо обидві літери біграми відкритого тексту належать до одного стовпця таблиці, то за літери шифртексту вважаються літери, котрі розміщено під ними.

Коли при цьому літера відкритого тексту перебуває в нижньому рядку, то для шифртексту береться відповідна літера з верхнього рядка того самого стовпця;

– якщо обидві літери біграми відкритого тексту належать до одного рядка таблиці, то за літери шифртексту вважаються літери, котрі розміщено праворуч від них. Коли при цьому літера відкритого тексту перебуває в крайньому правому стовпці, то для шифру беруть відповідну літеру з лівого стовпця в тому самому рядку.

При розшифровуванні застосовується зворотний порядок дій.

Процедуру зашифрування за допомогою біграмного шифру Плейфейра показано у прикладі 1.16.

Приклад 1.16 Зашифруємо за допомогою шифру Плейфейра текст:

ВСЯКИЙ СВОГО ЩАСТЯ КОВАЛЬ

В якості ключа візьмемо табл. 1.16. Розбиття цього тексту на біграми дає

ВС ЯК ИЙ СВ ОГ ОЩ АС ТЯ КО ВА Л Ь

Результат зашифрування зведемо у табл. 1.17

Таблиця 1.17 – Зашифровування біграм за допомогою шифру Плейфейра

Біграма вхідного тексту	ВС	ЯК	ИЙ	СВ	ОГ	ОЩ	АС	ТЯ	КО	ВА	ЛЬ
Шифртекст	ЗЙ	ЦУ	ВТ	ЙЗ	НИ	РЮ	РЙ	УЮ	ТН	ЙВ	БІ

При розшифровуванні застосовується зворотний порядок дій.

1.2.3 Завдання для самоперевірки з теми

Завдання 1.2.1 Використовуючи шифр Цезаря, зашифрувати власні іменні дані (прізвище, ім’я, по батькові).

Завдання 1.2.2 Зашифрувати власні іменні дані, якщо ключем є таблиця 1.7.

Завдання 1.2.3 Зашифрувати слово МРІЯ простою заміною за допомогою ключа, наведеного у табл. 1.15. Розрахувати в якому циклі вперше з’явиться вихідне слово, якщо процес зашифрування продовжувати необмежено.

Завдання 1.2.4 За допомогою афінної системи підставляння Цезаря для української абетки перетворити відкрите повідомлення КЛЮЧ на шифртекст.

$$E_{a,b}(t) \equiv at + b \pmod{m}, \text{ де } m = 31,\ a = 5,\ b = 7.$$

Завдання 1.2.5 Зашифрувати за допомогою таблиці Трисемуса з ключовим словом КРИПТОАНАЛІЗ власні іменні дані (прізвище та ім’я).

Завдання 1.2.6 Зашифрувати за допомогою шифру Плейфейра власні іменні дані. За ключ взяти слово КРИПТОАНАЛІЗ.

Контрольні питання з теми.

1. У чому полягає сутність шифрів простої заміни?
2. Схарактеризуйте різновиди шифрів простої заміни.
3. Що є ключем до шифру Цезаря?
4. Чим відрізняються монограмні шифри від біграмних?
5. У чому полягає сутність подвійного шифрування?
6. Який головний недолік шифрування методом простої заміни?

1.3 Шифри складної заміни

При шифруванні за допомогою шифрів складної заміни закон перетворювання змінюється від символу до символу. Шифри складної заміни називають багатоабетковими шифрами, тому що для шифрування кожного символу вихідного повідомлення застосовують власний шифр простої заміни. Багатоабеткове підставляння послідовно й циклічно змінює використовувані абетки.

При r-абетковому підставлянні символ x_0 вихідного повідомлення замінюється на символ y_0 з абетки B_0, символ x_1 – на символ y_1 з абетки B_1 і т. д.; символ x_{r-1} замінюється на символ y_{r-1} з абетки B_{r-1}, символ x_r замінюється на символ y_r знову з абетки B_0 і т. д.

Загальна схема багатоабеткового підставляння для випадку $r = 4$ наведена у табл. 1.18.

Таблиця 1.18 – Схема r-абеткового підставляння для випадку $r = 4$

Вхідний символ	x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9
Абетка підставляння	B_0	B_1	B_2	B_3	B_0	B_1	B_2	B_3	B_0	B_1

Ефект використовування багатоабеткового підставляння полягає в тім, що забезпечується маскування природної статистики вихідної мови, тому що конкретний символ з вихідної абетки A може бути перетворено на кілька різних символів шифрувальних абеток B_j.

1.3.1 Розвинення шифрів складної заміни

Багатоабеткові шифри заміни запропонував і запровадив у практику криптології Леон Батист Альберті, котрий був також відомим архітектором і теоретиком мистецтва. Його книга "Трактат про шифр", написана 1466 року, являла собою першу в Європі наукову працю з криптології. Криптологи усього світу вважають Л. Альберті за засновника криптології.

Леон Альберті уперше запропонував ідею подвійного шифрування: текст, здобутий внаслідок першого зашифровування, піддавався повторному зашифровуванню. У трактаті Альберті було наведено і його власний шифр, який він назвав "шифром, вартим королів". Він стверджував, що цей шифр є недешифровний. Реалізація шифру здійснювалася за допомогою шифрувального диска, який поклав початок цілої серії багатоабеткових шифрів. Пристрій являв собою пару дисків: зовнішній, нерухомий (на ньому було нанесено літери в природному порядку й цифри від 1 до 4) і внутрішній – рухомий (на ньому літери було переставлено). Процес зашифрування полягав у перебуванні літери відкритого тексту на зовнішньому диску й заміні її на відповідну (яка містилася під нею) літеру шифртексту. Після зашифровування кількох слів внутрішній диск зсувався на один крок. Ключем даного шифру є порядок розташування літер на внутрішньому диску і його початкове положення щодо зовнішнього диска.

Шифр, реалізовуваний диском Альберті, нашого часу дістав назви багатоабеткового. Зміст цієї назви полягає в такому.

Повернімося до дворядкового запису шифру заміни Ю. Цезаря. Назвемо верхній рядок абеткою відкритого тексту, а нижній – шифрабеткою. Якщо в перебігу шифрування шифрабетка не змінюється, то шифр є одноабетковим (чи шифром простої заміни); якщо ж ця абетка змінюється, то шифр є багатоабетковим. Отже, шифр Цезаря – це шифр простої заміни, а в багатоабетковому шифрі Альберті кількість абеток дорівнює кількості літер в абетках відкритого тексту плюс чотири. Альберті – винахідник багатоабеткових шифрів, які, переважно, використовуються й у наші дні. Однак засіб продукування послідовності абеток шифртексту та їхній вибір є надто ускладнений; в Альберті він визначався циклічним зсуненням на одиницю через заздалегідь обумовлену кількість літер, які треба зашифрувати, тобто процес шифрування став "динамічним". Другий винахід Альберті – літерно-цифровий код (щоправда, малого обсягу). Цифри на диску Альберті (1, 2, 3, 4) шифруються так само, як і літери. Альберті запропонував використовувати упорядковані дво-, три- й чотирицифрові комбінації в якості кодопозначань для літер, слів і цілих фраз (кількість таких комбінацій дорівнює 336). Не виключено, що такі коди використовувалися й раніше, але в історичних документах їх позв’язують з ім’ям Альберті. Особливо відзначимо, що кодовані повідомлення потім повторно шифрувались, тобто використовувався код з перешифровуванням. Ця ідея використовується і в сучасному шифруванні.

У XVI сторіччі вагомий внесок до розвинення криптографії вклав криптограф папи римського Маттео Ардженті, котрий успадкував мистецтво тайнопису від свого дядька. Саме Ардженті запропонував використовувати слово-пароль для надання абетці легко запам’ятовуваного змішаного вигляду. Про це вже йшлося при розгляданні ускладненого шифру Полібія.

Ардженті рекомендував не відокремлювати слова, застосовувати омофонні заміни, вставляти в шифртекст велику кількість "пустишок", усувати пунктуацію, не вставляти в шифртекст відкриті слова ("клер") і т. д. Для утруднення дешифрування шифрів заміни він запропонував таке: замінювати літери чи на цифри (від 0 до 9), чи на числа (від 00 до 99), причому, аби уникнути плутанини при розшифровуванні, цифри, використовувані як самостійні шифропозначення, не повинні входити до двознакових позначань. Оскільки однознакових позначань виявляється порівняно небагато, то, аби не впадала в око їхня мала частість з’явлення в шифртексті, Ардженті рекомендував додавати однознакові позначання літер, які найчастіше зустрічаються у відкритому тексті.

Наведемо приклад шифру заміни Ардженті для італійської мови. У цій заміні поряд із шифруванням використовується шифрування-кодування певних дво- й трилітерних часто вживаних сполучень.

A	B	C	D	E	F	G	H	I	L	M	N	O
1	86	02	20	62, 82	22	06	60	3	24	26	84	9

P	Q	R	S	T	U	Z	ET	CON	NON	CHE	ПУСТИШКИ
66	68	28	42	80	46	88	08	64	00	44	5, 7

Слово ARGENTI могло набути при зашифровуванні вигляду

5128066284580377 або 17728506825847805З7.

Це було насправді серйозне ускладнення шифру заміни. Частотний аналіз дешифрувальника істотно ускладнювався.

Арджентi займався також ускладненням кодів (номенклаторів). Зокрема, він уперше розробив літерний код, у якому 1200 літер, складів, слів і цілих фраз замінювалися на групу з літер.

Порта видозмінив шифрувальний диск Альберті, перетворивши абетку шифртексту на улюблені ним символіко-геометричні фігури. Зрозуміло, жодного ускладнення при цьому шифр Альберті не набув, додалося лише екзотики у шифртексті.

Основна книга Порта про тайнопис – це книга "Про таємну переписку". У ній Порта висвітлив слабкості широко розповсюджених того часу шифрів, у тому числі й шифрів масонів, котрі він іронічно назвав шифрами "сільських жителів, жінок та дітей", і запропонував так званий біграмний шифр.

Цей шифр є шифр біграмної (дволітерної) заміни, в якому кожному дволітерному сполученню відкритого тексту в шифртексті відповідав спеціально вигаданий знак. Знаки шифртексту мали форму символіко-геометричних фігур. Власне це був той самий шифр простої заміни, але на рівні дволітерних сполучень. Криптографічна стійкість за такої заміни порівняно з політерним шифруванням помітно ставала потужнішою.

Порта також запропонував змеханізовувати процес шифрування за його таблицею. Він навів опис механічного дискового пристрою, який зреалізовує біграмну заміну. Порта рекомендував не використовувати в переписуванні стандартних слів та виразів; більш того, він пропонував записувати відкритий текст з помилками, аби утруднити роботу дешифрувальника. Він писав: "... коли тема переписування є відома, аналітик може робити проникливі припущення щодо слів ...", що може істотно полегшити роботу дешифрувальника.

Порта запропонував певну модифікацію шифру Белазо. У застосуванні до української мови він являє собою прямокутну таблицю з літер абетки в порядку, наведеному на рис. 1.11.

Шифрування здійснюється за допомогою секретного гасла. Це гасло періодично виписується над відкритим текстом, за першою літерою цього гасла відшуковується абетка (великі літери на початку рядків), у верхній чи нижній напівабетці відшукується перша літера відкритого тексту і замінюється на відповідну їй літеру з верхнього чи нижнього рядка. Аналогічно шифруються й інші літери (інтервали поміж словами не враховуються).

Приклад 1.17. Зашифруємо відкритий текст ПЕРІОДИЧНИЙ ШИФР гаслом є слово СКАРБНИЦЯ. Ключем шифрування є табл. 1.19.

Таблиця 1.19 – Шифр Порта для української мови

1	А Б	а н	б о	в п	г р	д с	е т	є у	ж ф	з х	и ц	і ч	ї ш	й щ	к ь	л ю	м я
2	В Г	а о	б п	в р	г с	д т	е у	є ф	ж х	з ц	и ч	і ш	ї щ	й ь	к ю	л я	м н
3	Д Е	а п	б р	в с	г т	д у	е ф	є х	ж ц	з ч	и ш	і щ	ї ь	й ю	к я	л н	м о
4	Є Ж	а р	б с	в т	г у	д ф	е х	є ц	ж ч	з ш	и щ	і ь	ї ю	й я	к н	л о	м п
5	З И	а с	б т	в у	г ф	д х	е ц	є ч	ж ш	з щ	и ь	і ю	ї я	й н	к о	л п	м р
6	І Ї	а т	б у	в ф	г х	д ц	е ч	є ш	ж щ	з ь	и ю	і я	ї н	й о	к п	л р	м с
7	Й К	а у	б ф	в х	г ц	д ч	е ш	є щ	ж ь	з ю	и я	і н	ї о	й п	к р	л с	м т
8	Л М	а ф	б х	в ц	г ч	д ш	е щ	є ь	ж ю	з я	и н	і о	ї п	й р	к с	л т	м у
9	Н О	а х	б ц	в ч	г ш	д щ	е ь	є ю	ж я	з н	и о	і п	ї р	й с	к т	л у	м ф
10	П Р	а ц	б ч	в ш	г щ	д ь	е ю	є я	ж н	з о	и п	і р	ї с	й т	к у	л ф	м х
11	С Т	а ч	б ш	в щ	г ь	д ю	е я	є н	ж о	з п	и р	і с	ї т	й у	к ф	л х	м ц
12	У Ф	а ш	б щ	в ь	г ю	д я	е н	є о	ж п	з р	и с	і т	ї у	й ф	к х	л ц	м ч
13	Х Ц	а щ	б ь	в ю	г я	д н	е о	є п	ж р	з с	и т	і у	ї ф	й х	к ц	л ч	м ш
14	Ч Ш	а ь	б ю	в я	г н	д о	е п	є р	ж с	з т	и у	і ф	ї х	й ц	к ч	л ш	м щ
15	Щ Ь	а ю	б я	в н	г о	д п	е р	є с	ж т	з у	и ф	і х	ї ц	й ч	к ш	л щ	м ь
16	Ю Я	а я	б н	в о	г п	д р	е с	є т	ж у	з ф	и х	і ц	ї ч	й ш	к щ	л ь	м ю

Результат зашифрування зведемо у табл. 1.20.

Таблиця 1.20 – Зашифрування за таблицею Порта

Гасло	С	К	А	Р	Б	Н	И	Ц	Я	С	К	А	Р	Б	Н
Вхідний текст	П	Е	Р	І	О	Д	И	Ч	Н	И	Й	Ш	И	Ф	Р
Шифртекст	З	Ш	Г	Р	Б	Щ	Ь	Л	Б	Р	П	Ї	П	Ж	Ї

За цей шифр Порта пізніше назвали батьком сучасної криптографії, але свого часу цей шифр не набув широкого застосовування. Причина цього – необхідність завжди мати при собі зазначену таблицю і складність процесу

шифрування. Однак було надано імпульсу для з’явлення інших систем шифрування (наприклад шифру Віженера).

Блез де Віженер (XVI ст.), посол Франції в Римі, ознайомившись з працями Тритемія, Белазо, Кардано, Порта, Альберті, також захопився криптографією.

У 1585 році Блез де Віженер написав "Трактат про шифри", в якому викладено основи криптографії. У цій праці він зауважує: "Усі речі в світі являють собою шифр. Уся природа є просто шифром і секретним посланням". Цю думку було пізніше повторено Блезом Паскалем – одним із засновників теорії ймовірностей, а потім і Норбертом Вінером – "батьком" кібернетики. У цьому трактаті знову „взято на озброєння” ідею використовування найбільш відкритого тексту як ключа. Заздалегідь обумовлюється одна ключова літера абетки, й перша літера повідомлення шифрується таблицею Тритемія за рядком, що він відповідає першій літері шифрованого повідомлення, і т. д. Отже, було реалізовано ідею, раніше запропоновану Кардано.

Система Віженера є подібна до такої системи шифрування Цезаря, в якій ключ підставляння змінюється від літери до літери. Цей шифр багатоабеткової заміни можна описати таблицею шифрування, яку називають таблицею (квадратом) Віженера (Додаток Б).

Таблиця Віженера використовується для зашифровування й розшифровування.

Таблиця має два входи:

– верхній рядок підкреслених символів, який використовується для зчитування чергової літери вихідного відкритого тексту;

– крайній лівий стовпець ключа.

Послідовність ключів зазвичай здобувають з числових значень літер ключового слова.

Віженер сполучив підходи Тритемія, Белазо, Порта до шифрування відкритих текстів, істотно не внісши до них нічого оригінального.

При шифруванні вихідного повідомлення шифром Віженера його виписують у рядок, а під ним записують ключове слово (чи фразу). Якщо ключ виявився коротше за повідомлення, то його циклічно повторюють. У перебігу шифрування відшуковують у верхньому рядку таблиці чергову літеру вихідного тексту й у лівому стовпці – чергове значення ключа. Чергова літера шифртексту перебуває на перетинанні стовпця, визначуваного літерою відкритого тексту, і рядка, визначуваного числовим значенням ключа.

Розшифровування відбувається в спосіб відшукування букви шифртексту в рядку, де розміщується літера ключа. Літера відкритого тексту буде у верхньому рядку таблиці.

Приклад зашифрування за допомогою таблиці Віженера наведено у прикладі 1.18.

Приклад 1.18. Зашифруємо за допомогою таблиці Віженера повідомлення ПРИЛІТАЮ СЬОМОГО. Нехай за ключове слово обране АМБРОЗІЯ.

Випишемо відкрите повідомлення в рядок і запишемо під ним ключове слово з повторенням. В третій рядок будемо виписувати літери шифртексту,

визначувані з таблиці Віженера (Додаток А).

Результат зашифрування зведемо у табл. 1.21.

Таблиця 1.21 – Зашифрування за таблицею Віженера

Повідомлення	П	Р	И	Л	І	Т	А	Ю	С	Ь	О	М	О	Г	О
Ключ	А	М	Б	Р	О	З	І	Я	А	М	Б	Р	О	З	І
Шифртекст	П	В	І	Б	Ш	Ь	І	Ь	С	Й	П	В	В	Ї	Ш

Пізніше шифр Віженера значно спростив для його практичного використовування граф Гронсфельд – керівник першого в Німеччині державного дешифрувального органа ("криптографічної лабораторії"). Його пропозиція призвела до з’яви так званого шифру гамування – одного з найпоширеніших шифрів у сучасній криптографії.

Окремим випадком системи шифрування Віженера є система шифрування Вернама. Конкретна версія цього шифру, запропонована 1926 року Гілбертом Вернамом, співробітником фірми AT&T США, використовує двійкове подавання символів вихідного тексту (модуль $m = 2$).

Кожен символ відкритого тексту з англійської абетки {A, B, C, D, ..., Z}, розширеного шістьма допоміжними символами (прогалина, повертання каретки й т. п.), передусім кодувався в п’ятибітовий блок (b_0, b_1, …, b_4) телеграфного коду Бодо. Випадкова послідовність двійкових ключів k_0, k_1, k_2, ... заздалегідь записувалася на паперовій стрічці. Зашифровування вихідного тексту, попередньо перетвореного на послідовність двійкових символів x, здійснювалося шляхом додавання за модулем 2 символів x з послідовністю двійкових ключів k. Символи шифр тексту

$$y = x \oplus k.$$

Розшифровування полягає в додаванні за модулем 2 символів y шифртексту з тією самою послідовністю ключів k:

$$x = y \oplus k.$$

При цьому послідовності ключів, використовувані при зашифровуванні й розшифровуванні, компенсують одна одну (при додаванні за модулем 2) – і як наслідок відновлюються символи x вхідного тексту.

При розроблянні своєї системи Вернам перевіряв її за допомогою закільцьованих стрічок, установлених на передавачі й приймачі для того, аби використовувалась одна й та сама послідовність ключів.

Слід зазначити, що метод Вернама не залежить від довжини послідовності ключів і, окрім того дозволяє використовувати випадкову послідовність ключів. Однак при реалізації методу Вернама виникають серйозні проблеми, пов’язані з необхідністю доставляння одержувачеві такої самої послідовності ключів, як і у

відправляча, або з необхідністю безпечного зберігання ідентичних послідовностей ключів у відправляча й одержувача. Ці недоліки системи шифрування Вернама подолано при шифруванні методом гамування.

Суть цієї пропозиції полягає в такому.

Випишемо латинську абетку:

ABCDEFGHIJKLMNOPQRSTUVWXYZ.

За ключ-гасло обирається число, котре легко запам'ятовується, наприклад 13579. Це гасло періодично виписується над літерами відкритого тексту (одна цифра над літерою). При зашифруванні літера відкритого тексту замінюється на літеру, котра відстоїть від неї праворуч (циклічно) в абетці на кількість літер, зумовлюваних відповідною цифрою гасла. Так, приміром, за зазначеного гасла слово THE TABLE перетворюється на послідовність UKJAJCOJ.

Подальша модернізація призвела нашого часу до шифру модульного гамування.

Пронумеруємо літери абетки: A = 01, B = 02, C = 03, …, Z = 26.

Слово THE TABLE набуває вигляду

20.08.05.20.01.02.12.05.

Застосуймо операцію циклічного (модульного) додавання. Від звичайного додавання ця операція відрізняється тим, що, якщо сума перевищує 26, від неї віднімається 26; обернена операція – віднімання – характеризується тим, що, якщо в результаті виходить від'ємне число, до нього додається 26.

Зашифровування провадиться за операцією модульного додавання. Випишемо гасло 13579 періодично під відкритим текстом і складемо відповідні числа. Здобудемо шифртекст: 21.11.10.01.10.13.15.10., що відповідає сполученню літер UKJAJCOJ. При розшифровуванні гасло віднімається з літер шифртексту.

Шифр складної заміни, називаний *шифром Гронсфельда*, являє собою модифікування шифру Цезаря з числовим ключем. Для цього під літерами відкритого повідомлення записують цифри числового ключа. Якщо ключ є коротший за повідомлення, то його запис циклічно повторюють. Шифртекст здобувають приблизно так само, як в шифрі Цезаря, але відлічують за абеткою не третю літеру (як це чинять в шифрі Цезаря), а обирають ту літеру, котру зміщено за абеткою на відповідну цифру ключа. Шифр Гронсфельда являє собою, власне кажучи, окремий випадок системи шифрування Віженера (ключ подається в числовому вигляді), тому для шифрування відкритого повідомлення користуються таблицею Віженера (Додаток А).

Приклад зашифрування методом Гронсфельда за допомогою таблиці Віженера для української мови наведено у прикладі 1.19.

Приклад 1.19. Зашифруємо шифром Гронсфельда відкрите повідомлення

БЕЗ ОХОТИ НЕМА РОБОТИ,

де як ключ оберемо послідовність 2718.

Результат зашифрування зведемо у табл. 1.22.

Таблиця 1.22 – Зашифрування шифром Гронсфельда за таблицею Віженера

Повідомлення	Б	Е	З	О	Х	О	Т	И	Н	Е	М	А	Р	О	Б	О	Т	И
Ключ	2	7	1	8	2	7	1	8	2	7	1	8	2	7	1	8	2	7
Шифртекст	Г	Й	И	Ц	Ч	Х	У	О	П	Й	Н	З	Т	Х	В	Ц	Ф	Н

Слід зазначити, що шифр Гронсфельда розкривається відносно легко, якщо врахувати, що в числовому ключі кожна цифра має лише десять значень, а отже, є лише десять варіантів прочитування кожної літери шифртексту. З іншого боку, шифр Гронсфельда припускає подальші модифікування, що вони поліпшують його стійкість, зокрема подвійне шифрування різними числовими ключами.

Зауважимо, що з розвиненням математики зникла потреба у таблицях Віженера й Бофора при зашифровуванні й розшифровуванні.

У XVII сторіччі Дж. Фальконер (Англія) видав книгу "Розкриття секретних повідомлень". У ній викладено певні розроблені ним методи дешифрування. Зокрема він запропонував використовувати перебирання можливих відкритих слів за їхньою довжиною, якщо в шифртексті залишалося розбивання на слова.

Запропонована Фальконером система шифрування поєднує два шифри: вертикальної перестановки й гамування.

Наведемо приклад. Оберемо гасло ЛІЛІПУТ. За цим гаслом будується так званий номерний ряд за таким правилом: літери гасла нумеруються за абеткою; при цьому однакові літери набувають послідовних номерів. Зазначеному гаслові відповідає такий номерний ряд: 3 1 4 2 5 7 6

А	Б	В	Г	Д	Е	Є	Ж	З	И	І	Ї	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ь	Ю	Я
										1				3				5			6	7									
										2				4																	

У 1854 році англієць Чарльз Уїтстон розробив новий метод шифрування біграмами, котрий називають *подвійним квадратом*. Своєї назви цей шифр дістав за аналогією з полібіанським квадратом. Шифр Уїтстона відкрив новий етап в історії розвинення криптографії. На відміну від полібіанського, шифр "подвійний квадрат" використовує одразу дві таблиці (рис. 1.14), розміщувані на одній горизонталі, а шифрування здійснюється біграмами, як у шифрі Плейфейра. Ці не надто складні модифікації призвели до з'явлення на світ якісно нової криптографічної системи ручного шифрування. Шифр "подвійний квадрат" виявився вельми надійним та зручним і застосовувався в Німеччині навіть у другій світовій війні.

З'ясуємо процедуру шифрування цим шифром на прикладі. Нехай є дві таблиці з довільно розташованими в них українськими абетками. Перед

зашифровуванням повідомлення розбивають на біграми. Кожна біграма шифрується окремо. Першу літеру біграми відшукують у лівій таблиці, а другу літеру – у правій таблиці. Потім подумки будують прямокутник у такий спосіб, аби літери біграми містилися в його протилежних вершинах. Інші дві вершини цього прямокутника надають літери біграми шифртексту.

Ж	Щ	Н	Ю	Р
И	Т	Ь	Ц	Б
Я	М	Е	Л	С
В	І	П	Ч	А
Х	Д	У	О	К
З	Є	Ф	Г	Ш

И	Ч	Г	Я	Т
Ф	Ж	Ь	М	О
З	Ю	Р	В	Щ
Ц	У	П	Е	Л
Є	А	Н	Б	Х
І	К	С	Ш	Д

Рисунок 1.14 – Дві таблиці з довільно розташованими символами української абетки для шифру "подвійний квадрат"

Припустімо, що шифрується біграма вихідного тексту ИЛ. Літера И міститься в стовпці 1 і рядку 2 лівої таблиці. Літера Л міститься в стовпці 5 і рядку 4 правої таблиці. Це означає, що прямокутник утворено рядками 2 й 4, а також стовпцями 1 лівої таблиці й 5 правої таблиці. Отже, до біграми шифртексту входять літера О, розміщена в стовпці 5 і рядку 2 правої таблиці, і літера В, розташована в стовпці 1 і рядку 4 лівої таблиці, тобто здобуваємо біграму шифртексту ОВ.

Якщо обидві літери біграми повідомлення містяться в одному рядку, то й літери шифртексту беруть з того самого рядка. Першу літеру біграми шифртексту беруть з лівої таблиці в стовпці, який відповідає другій літері біграми повідомлення. Друга ж літера біграми шифртексту береться з правої таблиці в стовпці, який відповідає першій літері біграми повідомлення. Тому біграма повідомлення ТО перетворюється на біграму шифртексту БЖ. В аналогічний спосіб шифруються всі біграми повідомлення. Слід зауважити, що відправляч і одержувач повідомлення можуть змінювати умови шифрування. Варіант зашифрування шифру "подвійний квадрат" показано у прикладі 1.20.

Приклад 1.20. Зашифруємо біграмним шифром "подвійний квадрат" відкрите повідомлення НАДІЙНЕ ШИФРУВАННЯ. Ключем шифрування є таблиця (рис. 1.14).

Перед шифруванням відкритий текст розбивається на біграми без пробілів. Результати зашифрування за правилами, що наведено вище, зведено у табл. 1.23.

Таблиця 1.23 – Зашифрування шифром "подвійний квадрат"

Повідомлення	НА	ДІ	ЙН	ЕШ	ИФ	РУ	ВА	НН	ЯЬ
Шифртекст	ЧУ	ЄЄ	ЬХ	ВФ	ИФ	ЧА	УХ	ГУ	РИ

Шифрування методом подвійного квадрата надає вельми стійкий до розкриття і простий у застосовуванні шифр. Зламування шифртексту "подвійний квадрат" потребує потужних зусиль, при цьому довжина повідомлення має бути не менш ніж тридцять рядків.

1.3.2 Роторні машини

Першою спробою побудувати роторну машину була так звана машина Джефферсона, створена наприкінці XVIII сторіччя першим державним секретарем CIF Томасом Джефферсоном. Вона являла собою певну кількість дисків, які надівались на вісь, утворюючи в такий спосіб циліндр. На ободі кожного колеса рівномірно й випадково було нанесено символи абетки, з яких формувався текст криптограм. Розподілювання символів було випадковим, і на кожному колесі воно відрізнювалось від розподілення на інших колесах. Колеса можна було знімати та змінювати за місцями. Уздовж циліндра могли пересуватись та фіксуватись дві паралельні планки. Прокручуючи кожне колесо, блок відкритого тексту набирали уздовж першої планки. Текст, що складався в такий спосіб уздовж другої планки і не мав жодного смислу, являв собою відповідний блок зашифрованого тексту (криптограми). Ключем у даному разі були розподілення літер на колесах, послідовність цих коліс та відстань поміж планками (рис. 1.12).

Рисунок 1.12 – Схема Коліс Джефферсона

Всього машина Джефферсона мала 36 дисків, на кожному з яких було нанесено по 26 літер латинської абетки. Це означає, що сумарна кількість варіантів, якими могло бути зашифроване повідомлення, дорівнювала $26! \cdot 36!$ і мала порядок 10^{60}.

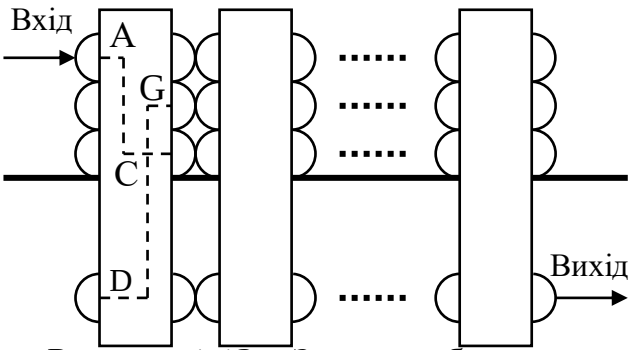


Рисунок 1.13 – Засада побудови роторної машини

На жаль, винахід Джефферсона було забуто на багато років, через те що рівень розвинення математики на той час не дозволив правильно оцінити стійкість такого способу шифрування. Сам Джефферсон та його адміністрація продовжували користуватись іншими, значно гіршими шифрами.

20-ми роками XX сторіччя було винайдено електромеханічні пристрої шифрування, котрі автоматизують процеси зашифровування й розшифровування. Робота таких машин базується на засаді багатоабеткової заміни символів вихідного тексту за довгим ключем відповідно до версії шифру Віженера. Більшість з них – американська машина SIGABA (M-134), англійська TYPEX, німецька ENIGMA, японська PURPLE – були роторними машинами (рис. 1.13).

Головною деталлю роторної машини є ротор (чи колесо) із дротовими перемичками всередині. Ротор має форму диска (розміром з хокейну шайбу). На кожному боці диска розташовано рівномірно за колом m-електричні контакти, де m – кількість знаків абетки (в разі латинської абетки $m = 26$, української – 33). Кожен контакт на передньому боці диска сполучено із одним з контактів на задньому боці. Як наслідок електричний сигнал, котрий являє собою знак, буде переставлено відповідно до того, як він проходить через ротор від переднього боку до заднього.

Приміром, ротор можна закомутувати дротовими перемичками для підставляння А замість Ф, Б – замість У, С – замість Л і т. п. При цьому вихідні контакти одного ротора мають долучатися до вхідних контактів ротора, який слідує за ним. Тоді, приміром, якщо на клавіатурі чотириторної машини натискалася клавіша А, то перший ротор міг перетворити її на літеру Ф, яка, пройшовши через другий ротор, могла стати літерою Т, яку третій ротор міг замінити на літеру К, котра могла бути перетворена четвертим ротором на літеру Е шифртексту. Після цього ротори поверталися – і наступного разу заміна була іншою. Аби спантеличити криптоаналітиків, ротори оберталися з різною швидкістю.

Щоби роторна машина була оптимальною, мають виконуватись такі умови:

– період повторювання має бути великим;

– після зашифровування кожного знаку якомога більша частина роторів повинна змінювати своє положення.

Щодо машини ENIGMA, то другому пунктові вимог вона не відповідає, але цим забезпечується простота її технічної реалізації. Ускладнення способу обертання дисків має виконуватись в такий спосіб, щоби зберігалась однозначність шифрування, а це є надто складна річ.

Роторна машина може бути налаштована за ключем зміною будь-яких її змінних:

– роторів;

– порядку розташування роторів;

– кількості місць зупинки на колесо;

– характеру руху тощо.

Оскільки перекомутовувати ротори складно, то зазвичай на практиці машини забезпечували комплектом роторів, у якому перебувало більше роторів, аніж можна водночас розмістити в машині. Первинне налаштовування за ключем здійснювалося вибором роторів, які складають комплект. Вторинне налаштовування за ключем здійснювалося вибором порядку розташування роторів у машині й установленням параметрів, керуючих рухом машини. З метою утруднення розшифровування шифртекстів неправочинним користувачем ротори щодня переставляли місцями чи замінювали. Більша частина ключа визначала початкові положення роторів і конкретні переставляння на комутаційній дошці, за допомогою якої здійснювалося початкове переставляння вихідного тексту до його зашифровування.

Роторні машини були найважливішими криптографічними пристроями під час другої світової війни й домінували, принаймні, до кінця 50–х років минулого сторіччя.

Майже всі застосовувані на практиці шифри характеризуються як умовно надійні, оскільки вони можуть бути переважно розкриті за наявності необмежених обчислювальних можливостей. Абсолютно надійні шифри не можна зруйнувати навіть за використання необмежених обчислювальних можливостей. Існує єдиний такий шифр, застосовуваний на практиці, – одноразова система шифрування. Характерною рисою одноразової системи шифрування є одноразове використовування ключової послідовності.

Одноразова система шифрує вихідний відкритий текст

$$\overline{X} = (X_0, X_1, ..., X_{n-1})$$

на шифр текст

$$\overline{Y} = (Y_0, Y_1, ..., Y_{n-1})$$

за допомогою підстановки Цезаря

$$Y_i \equiv (X_i + K_i) \bmod m,\ 0 < i < n,$$

де K_i – i-й елемент випадкової ключової послідовності.

Ключовий простір $\overline{K}$ одноразової системи являє собою набір дискретних випадкових величин з $\overline{Z}_m$ і містить m^n значень.

Процедура розшифровування описується співвідношенням

$$X_i \equiv (Y_i - K_i) \bmod m,$$

де K_i – i-й елемент тієї ж самої випадкової ключової послідовності.

Одноразову систему винайдено 1917 року американцями Дж. Моборном та Г. Вернамом. Для реалізації цієї системи підставляння іноді використовують одноразовий нотатник. Цей нотатник складено з відривних листків, на кожному з яких надруковано таблицю з випадковими числами (ключами) K_i. Нотатник виконується у двох екземплярах: один використовується відправлячем, а другий – одержувачем. Для кожного символу X_i повідомлення використовується власний ключ K_i з таблиці лише одноразово. Після того як таблицю використано, її має бути вилучено з нотатника і знищено. Шифрування нового повідомлення розпочинається з нового листка.

Цей шифр буде абсолютно надійний, якщо набір ключів K_i буде насправді випадковий і непередбачуваний. Якщо криптоаналітик спробує використовувати для заданого шифртексту всі можливі набори ключів і відновити всі можливі варіанти вихідного тексту, то вони усі виявляться рівноймовірними. Не існує жодного способу обрати вихідний текст, що його було насправді надіслано. Теоретично доведено, що одноразові системи є нерозкривними системами, оскільки їхній шифртекст не містить достатньої інформації для відновлення відкритого тексту.

Здавалося б, що завдяки даному достоїнству одноразові системи варто застосовувати завжди, коли є вкрай потрібна абсолютна інформаційна безпека. Однак можливості застосовування одноразової системи є обмежені чисто практичними аспектами. Істотним моментом є вимога одноразового використовування випадкової ключової послідовності. Ключова послідовність з довжиною, не меншою за довжину повідомлення, повинна передаватися одержувачеві повідомлення заздалегідь чи окремо певним секретним каналом. Ця вимога не буде надто обтяжливою для передавання насправді важливих одноразових повідомлень, приміром гарячою лінією. Однак така вимога практично є нездійсненна для сучасних систем опрацьовування інформації, де потрібно зашифровувати безліч мільйонів символів.

У певних варіантах одноразового нотатника вдаються до більш простого керування ключовою послідовністю, але це призводить до певного зниження надійності шифру. Наприклад, ключ зумовлюється зазначенням місця в книзі, відомій і відправлячеві й одержувачеві повідомлення. Ключова послідовність розпочинається з певного місця цієї книги й використовується в такий самий спосіб, як і в системі Віженера. Іноді такий шифр називають шифром з рухомим ключем.

1.3.3 Завдання для самоперевірки з теми

Завдання 1.3.1 Зашифрувати методом "подвійного квадрата" свої іменні дані.

Ж	**Щ**	**Н**	**Ю**	**Р**
И	**Т**	**Ь**	**Ц**	**Б**
Я	**М**	**Е**	**Л**	**С**
В	**І**	**П**	**Ч**	**А**
Х	**Д**	**У**	**О**	**К**
З	**Є**	**Ф**	**Г**	**Ш**

И	**Ч**	**Г**	**Я**	**Т**
Ф	**Ж**	**Ь**	**М**	**О**
З	**Ю**	**Р**	**В**	**Щ**
Ц	**У**	**П**	**Е**	**Л**
Є	**А**	**Н**	**Б**	**Х**
І	**К**	**С**	**Ш**	**Д**

Завдання 1.3.2 Зашифрувати за допомогою таблиці Віженера власні іменні дані (П.І.Б.). Ключове слово – ДОКУМЕНТАЦІЯ.

Завдання 1.3.3 Зашифрувати шифром Гронсфельда власні іменні дані. Як ключ застосувати дату свого народження (приміром, 25 березня 1987 року буде подано такою послідовністю чисел: 25.03.19.8.7).

Контрольні питання з теми.

1. Чим відрізняються шифри складної заміни від шифрів простої заміни?
2. У чому полягає відмінність поміж шифром Гронсфельда й шифром Віженера?
3. Чому шифри складної заміни мають більшу крипостійкість, аніж шифри простої заміни?
4. Що є ключем до шифру Уїтстона "подвійний квадрат"?
5. Які змінні ключа можуть використовуватись у роторних машинах?

2 ЗАСАДИ КРИПТОГРАФІЧНОГО ЗАХИСТУ ІНФОРМАЦІЇ

У першому розділі розглянуто класичні методи шифрування, які лежать у підґрунті побудови будь-якої криптографічної системи.

Перші криптосистеми виникають ще до початку нової ери. Приміром, Юлій Цезар у своєму листуванні використовував уже більш-менш систематичний шифр, котрий дістав його ім'я.

Бурхливого розвинення криптографічні системи набули роками першої й другої світових війн минулого сторіччя. Розпочинаючи з післявоєнного часу й по сьогодні розвинення обчислювальних засобів прискорює розробляння й удосконалювання криптографічних методів.

Чому проблема використовування криптографічних методів в інформаційних системах (ІС) стала сьогодні надто актуальна?

З одного боку, розширилося використовування комп'ютерних мереж, зокрема глобальної мережі Інтернет, якими передаються потужні обсяги інформації державного, військового, комерційного та приватного характеру, що вона не припускає можливості доступу до неї сторонніх осіб.

З іншого боку, поява нових потужних комп'ютерів, технологій мережних та нейронних обчислень уможливила дискредитування криптографічних систем, котрі ще донедавна вважалися за стійкі.

2.1 Завдання, розв'язувані криптографічними методами

Перш ніж розпочати вивчення основних криптографічних систем та методів, що лежать у підґрунті їхньої побудови, треба визначитися із основними поняттями у сфері криптографічного захисту інформації. Тобто треба надати відповідь на запитання про те, що саме має захищатися в телекомунікаційних системах, а також від чого та від кого воно має захищатись і в який спосіб.

Відповідь на перше запитання вимагає побудови моделі інформаційного процесу. Для цього треба визначитися з тим, хто є учасниками інформаційного процесу, які завдання перед ними стоять і як вони збираються їх розв'язувати.

Відповіді на друге запитання повинні надавати критерії нормального перебігу інформаційних процесів у системі й визначати потенційно можливі обставини, які призводитимуть до відхилення їхнього перебігу від нормального. Такі обставини називають загрозами, а осіб, котрі зумисне чи то незумисне можуть їх утворювати, називають потенційними порушниками.

Розгорнута відповідь на друге запитання є моделлю порушника. Порушник – це окрема особа, сума цілей і можливостей, яка відповідає засаді: два суб'єкти, котрі мають однакові цілі й можливості, – це один суб'єкт.

Існує доволі велика численна кількість методів захисту інформації. Щодо криптографічних, то до їхнього складу відносять методи, у підґрунті яких лежать секретні алгоритми. Термін секретний алгоритм має широке тлумачення, але насамперед, мається на увазі алгоритм, цілковито або якийсь параметр чи частина його параметрів є відомі лише учасникам інформаційного процесу.

Надалі під *інформаційним процесом* матимемо на увазі процес взаємодії кількох суб'єктів, що передбачає обмін інформацією поміж ними. Таких суб'єктів може бути щонайменше два. Що стосується відхилення у перебігу інформаційного процесу від норми, то жодного об'єктивного критерію такої норми не існує. Тому загрозами ми й будемо називати будь-які можливі обставини, які призводять до такого відхилення. Порушники, котрі здійснюють загрози, можуть бути або законними учасниками інформаційного процесу, або особами, котрі просто мають доступ до інформації, яка опрацьовується чи передається в перебігу інформаційного процесу й, отже, можуть впливати на його перебіг.

Якщо учасники інформаційного процесу не мають змоги впливати на його перебіг, то такий процес називають *взаємодією в умовах довіри один до одного*. У противному разі процес називають *взаємодією недовіри один до одного*. Процеси останнього типу в сучасних системах є найбільш розповсюдженими. Слід зауважити, що тут йдеться не про особисту недовіру учасників процесу один до одного, а про дещо більше. Мають враховуватись всі чинники, як у внутрішньому, так і у зовнішньому середовищі функціонування системи, які можуть впливати на перебіг інформаційних взаємин.

Отже, завдання, розв'язувані криптографічними методами, можна поділити за такими критеріями:

– згідно з характером взаємин, котрі мають захищатись;

– згідно з цілями, які ставить перед собою злочинник;

– згідно з можливостями злочинників.

Найпростішим випадком інформаційних взаємодій є передавання даних від одного суб'єкта до другого. Відповідно, найрозповсюдженим завданням, розв'язуваним криптографічними методами, є захист конфіденційності інформації, котра зберігається й передається каналами зв'язку. Водночас це завдання є й найважливішим, незважаючи на те, що втілення інформаційних технологій в різноманітні сфери людської діяльності значно розширило їхнє коло.

2.2 Секретна система зв'язку

Завдання, яке полягає у захисті інформації під час передавання каналами зв'язку, вперше було сформульовано К. Шенноном у вересні 1945 року, а опубліковано у 1949 року в технічному журналі Bell System Technical Journal. Він запропонував секретну систему зв'язку, наведену на рис. 2.1.

Згідно із запропонованою системою, на боці, де передається повідомлення M_i, розміщено джерело повідомлень і джерело ключів K_i. При цьому на обох множинах M і K задано розподілення $P(M)$ та $P(K)$. Це означає, що для будь-якого $M_i \in M$ певна ймовірність $p(M_i) \in P(M)$, а для будь-якого $K_i \in K$ певна ймовірність $p(K_i) \in P(K)$ і виконується правило

$$\sum_{K_i \in K} p(K_i) = 1.$$

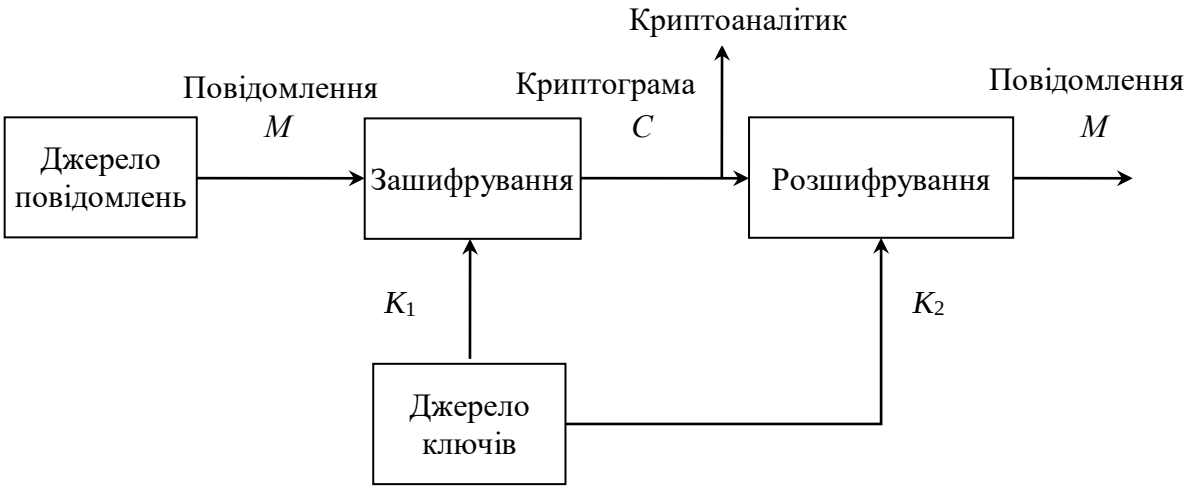


Рисунок 2.1 – Схема секретної системи зв'язку

Ключ, сформований з боку, звідки подається повідомлення з імовірністю $p(K_i)$, передається на протилежний бік через окремий закритий канал зв'язку, до якого злочинник не повинен мати доступу. Необхідність такого каналу є вельми серйозним недоліком секретних систем, оскільки в мережах з великою кількістю користувачів їхня реалізація вимагає надто потужних ресурсів.

Зашифровування відкритого тексту M за допомогою ключа K_1 здійснюється за допомогою функції зашифрування E та описується рівнянням:

$$C = E_{K_1}(M).$$

Розшифровування шифртексту C за допомогою ключа K_2 здійснюється за допомогою функції розшифрування D та описується рівнянням:

$$M = D_{K_2}(C).$$

При цьому, якщо $K_1 = K_2$ – це симетрична криптосистема з секретним ключем; якщо $K_1 \neq K_2$ – це асиметрична криптосистема, де K_1 – відкритий ключ; K_2 – відповідний секретний ключ.

2.3 Вимоги щодо реалізації сучасних криптоалгоритмів

Сучасні криптографічні системи може бути реалізовувано програмним, програмно-апаратним чи апаратним шляхом.

Існують такі загальні вимоги щодо реалізації криптографічних алгоритмів:

– криптографічні алгоритми має бути адаптовано до новітньої програмно-апаратної бази (приміром, алгоритми блочного шифрування в програмній реалізації має бути адаптовано до операцій з 64-розрядними числами);

– обсяг ключа має відповідати сучасним методам та засобам розшифровування зашифрованих повідомлень;

– операції шифрування й розшифровування мають за можливістю бути простими, щоби задовольняти сучасним вимогам швидкісних характеристик;

– обсяг повідомлення в перебігу виконання операцій шифрування має зводитися до мінімуму;

– має бути виключено явище розмножування помилок.

До недавнього часу алгоритми шифрування реалізовувалися у вигляді окремих пристроїв, що зумовлювалося використовуванням криптографії задля засекречування різних методів передавання інформації (телеграф, телефон, радіозв'язок). Сьогодні апаратна реалізація також набула широкого застосовування. Це зумовлено такими причинами.

По-перше, апаратна реалізація має кращі швидкісні характеристики, аніж програмно реалізовувані алгоритми шифрування. Використовування спеціальних чипів, адаптованих до реалізації процедур зашифровування й розшифровування, призводить до того, що, на відміну від процесорів загального призначення, вони дозволяють оптимізовувати багато математичних операцій, застосовуваних алгоритмами шифрування.

По-друге, апаратні засоби захисту інформації мають незрівнянно більшу захищеність як від побічних електромагнітних випромінювань, що виникають у перебігу роботи апаратури, так і від безпосереднього фізичного впливу на пристрої, де здійснюються операції шифрування і зберігання ключової інформації. Ці пристрої виконують в такий спосіб, що, в разі виявлення несанкціонованого доступу до них, вони саморуйнуються.

По-третє, апаратні засоби є більш зручні в експлуатації, тому що дозволяють здійснювати операції зашифровування й розшифровування для користувача в прозорому режимі; окрім того, їх легко інсталювати.

По-четверте, з огляду на різноманіття варіантів застосовування засобів криптографічного захисту інформації, апаратні засоби повсюдно використовуються для захисту телефонних перемовин, відправляння факсимільних повідомлень та інших видів передавання інформації, де неможливо використовувати програмні засоби.

До переваг програмної реалізації можна віднести те, що програма, написана під одну операційну систему, може бути змодифікована під будь-який тип ОС.

Окрім того, поновити програмне забезпечення можна з меншими часовими й фінансовими витратами. До того ж багато сучасних досягнень в області криптографічних протоколів є недоступні для реалізації у вигляді апаратних засобів.

До недоліків програмних засобів криптографічного захисту слід віднести можливість небажаного втручання в дію алгоритмів шифрування й одержання доступу до ключової інформації, що зберігається в загальнодоступній пам'яті.

Отже, слабка фізична захищеність програмних засобів є одним з головних недоліків подібних методів реалізації алгоритмів шифрування. Окрім того, програмна реалізація засобів криптографічного захисту не в змозі забезпечити виконання певних характеристик, необхідних для надійного використовування алгоритмів шифрування. Приміром, генерування ключової інформації не повинно

здійснюватися програмними давачами випадкових чисел; для цього треба використовувати спеціальні апаратні пристрої.

Програмно-апаратна реалізація дозволяє користувачам усувати певні недоліки програмних засобів захисту інформації і при цьому зберігати їхні переваги (за винятком цінового критерію).

Головними функціями, покладеними на апаратну частину програмно-апаратного комплексу криптографічного захисту інформації, зазвичай є генерування ключової інформації і зберігання ключової інформації в пристроях, захищених від несанкціонованого доступу з боку злочинника. Окрім того, за допомогою методик такого типу можна здійснювати автентифікацію користувачів за допомогою паролів (статичних чи динамічно змінюваних, котрі можуть зберігатися на різних носіях ключової інформації – смарт-карти, touch-memory тощо) або на підставі унікальних для кожного користувача біометричних характеристик.

2.3.1 Параметри сучасних шифрів

В реальних умовах за побудови криптографічних систем використовуються шифри, які не є досконалими. Це відбувається через те, що надто важко зреалізувати систему, яка формувала б велику кількість випадкових ключів, котрі відповідали б умовам Шеннона, та забезпечувала їхнє вчасне секретне розподілювання у мережах зв'язку з великою кількістю користувачів. Окрім того, недосконаліть шифру зовсім не означає, що він обов'язково буде зламаний. Насправді все залежить від інтелектуальних та обчислювальних ресурсів криптоаналітика. Якщо криптографічне перетворювання передбачає виконання досить великої кількості складних операцій, у потенційного злочинника не вистачить можливостей для того, щоби дешифрувати перехоплене повідомлення за розумний термін.

Таким, чином сучасні криптографічні системи мають будуватись з урахуванням можливостей імовірного злочинника, від якого захищається інформація, і, в такому разі, зрозуміло, виникає запитання про те, в який спосіб ці можливості мають враховуватись.

В якості критерію, за яким оцінюється складність обчислювального процесу за дешифрування повідомлень, найчастіше використовують кількість елементарних операцій w певного типу, які виконуються впродовж часу дешифрування одного повідомлення або визначання одного ключа. В кожному окремому випадку обговорюється, що саме містить термін *елементарна операція*. Це може бути чи сукупність операцій, виконуваних на конкретній апаратурі за один етап шифрування, чи то сукупність операцій, виконуваних упродовж часу однієї спроби підбирання ключа, чи щось інше. Головне – щоби вибір „одиниці" цього виміру було обґрунтовано.

Складність процедур дешифрування залежить від кількості апріорної інформації, яка є в розпорядженні у криптоаналітика. За найскладнішу вважається ситуація, коли, окрім перехопленої криптограми довжиною в N символів, у нього

нічого немає. Такий спосіб атаки на шифр називають *аналізом на підставі перехоплених криптограм*.

В цій ситуації єдиним способом дешифрування криптограми є перевірка усіх можливих значень ключа. Якщо його довжина дорівнює довжині повідомлення, сумарна кількість ключів, які треба буде перебрати, становитиме величину 2^N. Цю величину називають *потужністю ключового простору.* Підвищення потужності ключового простору, зрозуміло, збільшує криптографічну стійкість шифру, але не завжди й не неодмінно. Приміром, для шифру простої заміни потужність ключового простору дорівнює величині $m!$, де m – кількість символів у абетці. Хоча за доволі великої довжини тексту потужність ключового простору буде надто великою, а, як відомо, шифри простої заміни ламаються статистичними методами досить легко.

На жаль, не існує універсальних методів, що їх можна було б застосовувати задля криптографічного аналізу будь-яких шифрів. Кожен з відомих шифрів потребує розробляння індивідуального способу аналізу й дешифрування. Надалі під методом аналізу шифру розумітимемо сукупність правил та дій, сформульованих на підставі дослідження конкретного шифру й використання яких дає змогу виконувати дешифрування криптограми з імовірністю, котра відрізняється від нуля.

Дешифрування з імовірністю, яка відрізняється від нуля, означає, що, в разі, якщо ми обиратимемо ключ не з усієї з множини K, а лише з певної її частини $K_l \subseteq K$, яка сформована випадковим вибором з K, ймовірність дешифрування збігається з імовірністю потрапляння ключа, за допомогою якого зашифровано повідомлення, до множини K_l.

Для подальшого вивчання засад побудови шифрів, які відповідають заданим умовам щодо їхньої криптографічної стійкості, впровадимо необхідні означення.

Мінімальна ймовірність практичного дешифрування – обґрунтована числова величина, що характеризує практичну значимість алгоритму дешифрування. Ця величина залежить від багатьох параметрів і, першою чергою, від вимог щодо безпеки. Один і той самий шифр за різних умов матиме різні величини цього параметра.

Алгоритм дешифрування – обчислювальний алгоритм, побудований на підставі конкретного методу аналізу шифру, який надає змогу виконувати дешифрування.

Обчислювач – пристрій, який реалізує алгоритм дешифрування і має фіксовану кількість команд, кожна з яких називається *елементарною операцією* й виконується за один такт роботи.

Потужність обчислювача – кількість тактів, виконуваних за одну секунду.

Пам'ять обчислювача – доступна пам'ять, використовувана в перебігу обчислювання та зберігання даних.

Надійність алгоритму дешифрування – ймовірність дешифрування, визначувана обраним методом дешифрування та алгоритмом його реалізації.

Складність алгоритму дешифрування – кількість операцій, необхідних задля реалізації алгоритму дешифрування на даному обчислювачі, поділена на

надійність даного алгоритму (якщо обчислювач припускає його реалізацію).

Максимально припустима пам'ять – числова величина, яка характеризує практичну значимість алгоритму дешифрування.

Нехай M – множина всіх можливих методів аналізу шифру F з імовірністю дешифрування більше чи дорівнюваною ρ (мінімальна ймовірність практичного дешифрування). Через S визначимо множину всіх алгоритмів дешифрування, котрі потребують пам'яті не більш за величину RAM і відповідають множині M. Окрім того, нехай C – множина всіх обчислювачів. Тоді визначимо через $Q(c, s)$ складність дешифрування алгоритму S по відношенню до обчислювача Q для всіх $s \in S$ та $c \in C$.

З урахуванням розглянутого, можна зробити такий висновок. Стійкістю шифру F, за заданої мінімальної надійності π та максимального обсягу пам'яті RAM, називається величина

$$Q(F) = \min_{c \in C, s \in S} Q(c, s).$$

Інакше кажучи, *стійкість шифру* – це кількість операцій, які має бути виконано за умови використовування найбільш ефективного алгоритму та найбільш потужного обчислювача.

Окрім терміну *стійкість*, часто використовують термін *практична стійкість*, який означає складність реалізації найшвидшого з відомих алгоритмів, що відповідає найбільш ефективному з відомих методів на найшвидшому з доступних обчислювачів.

Отже, при створюванні того чи іншого шифру треба враховувати можливості, які має криптоаналітик злочинника. Побудова шифрів, стійкість яких значно перевершує необхідну, не є виправдана, оскільки вона досягається збільшенням часу, витрачуваного на виконання процедур зашифровування та розшифровування, а також використовуванням надто великих обчислювальних ресурсів.

2.3.2 Елементарні криптографічні перетворювання

У статті [22], опублікованій Клодом Шенноном 1949 року, було запропоновано будувати шифри, які задовольняють певним умовам щодо їхньої стійкості, шляхом використовування послідовності операцій заміни, перестановок та функційних перетворювань. Такі операції називають елементарними криптографічними перетворюваннями (рис. 2.2). Щоби перетворювання можна було називати елементарним, воно повинно досить легко реалізовуватися програмними чи апаратними засобами.

Історично так склалося, що на початку розвинення криптографії утворювались та використовувались шифри, які містили одне просте перетворення. Сьогодні такі перетворювання входять до складу сучасних шифрів у якості елементарних. Здебільшого це були заміни одних символів на інші або

їхні переставляння місцями в межах повідомлення, котре мало бути зашифрованим. Причому найбільшу й найрізноманітнішу групу становлять саме *шифри заміни*, тому розглядання елементарних криптографічних перетворювань розпочнемо саме з них.

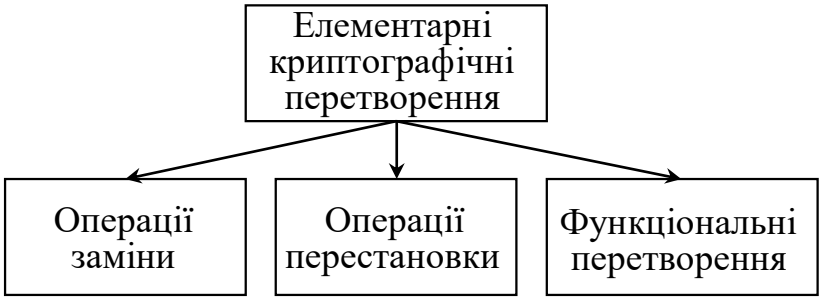


Рисунок 2.2 – Елементарні криптографічні перетворювання

У загальному випадку шифр заміни можна описати алгебричною моделлю вигляду

$$\Sigma_K = (T, K, T', E, D).$$

Вважатимемо також, що слова відкритого тексту складаються з символів абетки A_n, а слова криптограм – з символів абетки B_m. У загальному випадку у відкритому тексті групи символів, які складаються з абетки A_n, замінюються на групи символів, що складаються з абетки B_m. При цьому довжина шифрвеличин та відповідних до них шифрперетворювань може не збігатися. Окрім того, кожній з шифрвеличин може бути поставлено у відповідність понад одне шифрперетворювання. Тобто під час зашифровування повідомлення заміна може бути не безваріантною. Щодо розшифровування, то воно залишається однозначним, незалежно від способу зашифровування.

З урахуванням наведеного, множину шифрвеличин U можна розглядати як множину слів U_i абетки A^* (рис. 2.3, 2.4):

$$U = \{U_1, U_2, \ldots, U_n\},$$

а множину шифрперетворювань V – як множину слів V_i абетки B^*:

$$V = \{V_1, V_2, \ldots, V_n\}.$$

Тоді $T \subset A^*$, а $T' \subset B^*$. Якщо кількість символів у абетках A та B відповідно дорівнює n та m, сумарна кількість шифрвеличин U_i має становити $N > n$, а сумарна кількість шифрперетворювань V_i – $M > m$.

За такого способу задавання шифру заміни потужність ключового простору r, яка визначає стійкість шифру, буде надто великою.

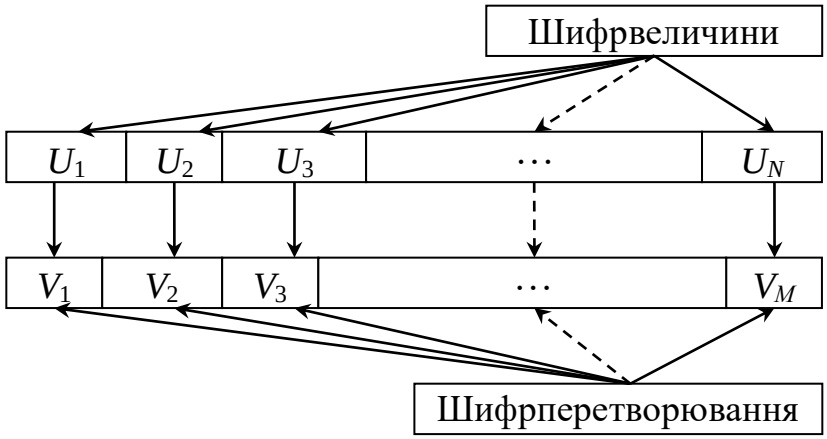


Рисунок 2.3 – Загальна засада заміни (однозначна заміна)

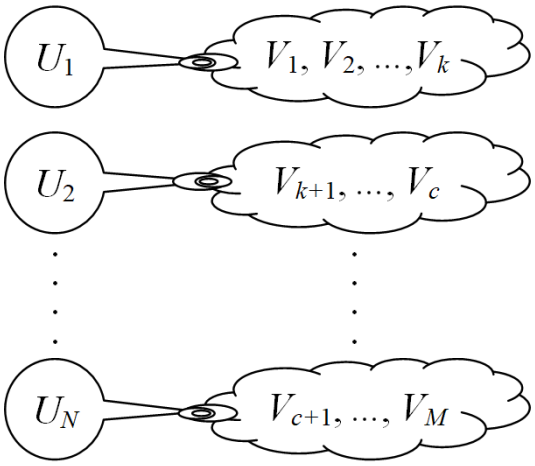


Рисунок 2.4 – Відповідність величин V_i до U_j.(багатозначна заміна)

Множину V можна подати як сімейство з r варіантів такого її розподілення:

$$V = \bigcup_{i=1}^{N} V_{\alpha}^{(i)}, \quad \alpha = 1,\ldots,r.$$

Для того щоби шифрування було однозначним, всі підмножини $V_{\alpha}^{(i)}$ не повинні перетинатись, тобто

$$V_{\alpha}^{i} \bigcap V_{\alpha}^{(j)} = \varnothing,$$

і кожна з підмножин $V_{\alpha}^{(i)}$ не повинна бути порожньою:

$$V_{\alpha}^{i} \neq \varnothing, \quad i = 1,\ldots,N, \quad \alpha = 1,\ldots,r.$$

Отже, з урахуванням вищенаведеного, можна надати таке означення: *шифром заміни* є шифр, котрий передбачає заміну шифрвеличин відкритого

тексту на відповідні до них фіксовані шифрперетворювання або групи шифрперетворювань закритого тексту.

У найпростішому випадку шифрвеличини збігаються із символами абетки A_m, за допомогою яких утворюються відкриті повідомлення, а шифрперетворювання – із символами абетки B_n, з якої утворюються криптограми, причому B_n найчастіш є підстановкою на A_m, що утворюється за допомогою відображення $A_m \to A_m$. Тоді $n = m$ і $N = M$. Шифри, які передбачають збіжність довжини шифрвеличин та відповідних до них шифрперетворювань, називають *шифрами рівнозначної заміни*. У противному разі мають місце *шифри різнозначної заміни*.

Прикладом такого шифру є шифр Цезаря, котрий вважається за один з перших історично відомих шифрів. Згідно із засадою шифрування, яку він передбачає, кожен символ a_i закритого повідомлення T_i перетворюється на відповідний до нього символ b_i закритого повідомлення T_i' за правилом

$$b_i' \equiv (a_i + k) \bmod m, \quad 1 \le k < m,$$

де k – є ключ до шифру.

У цьому шифрі використовуються лише адитивні операції над елементами множини A_m. Потужність його ключового простору обмежується довжиною абетки, тобто величиною m. Через це його стійкість є надто низька, але, зважаючи на те, що він використовувався в ті часи, коли переважна більшість людей взагалі читати не вміла, вона була достатньою.

Інший варіант цього шифру, котрий з'явився значно пізніше і відомий під назвою афінної системи підстановок Цезаря, окрім адитивних, використовує мультиплікативні операції, передбачає шифрування за правилом

$$b_i' = (ca_i + d) \bmod m, \quad 1 \le c, d < m, \quad (c, m) = 1,$$

де c, d – є цілими числами, причому числа c та m мають бути взаємнопростими для того, щоби забезпечувалась однозначність шифрування.

Потужність ключового простору такого шифру є значно більша по відношенню до попередньої його реалізації й дорівнює $m!$, але він так само має дуже низьку криптографічну стійкість. Статистичні характеристики символів відкритого тексту за таких способів шифрування цілковито переходять на символи у відповідній до нього криптограмі. Це відбувається через те, що обидва наведені приклади відповідають шифрам безваріантної заміни, коли кожен символ відкритого тексту під час його зашифровування завжди перетворюється на один і той самий символ криптограми. Існує багато інших прикладів таких шифрів. До їхнього складу можна віднести шифр Полібія, шифрування за допомогою магічних квадратів, шифрування за допомогою таблиць Трисемуса тощо.

Щоби запобігти повному перенесенню статистичних характеристик відкритих повідомлень на відповідні до них криптограми, минулими часами до складу абетки B_n, з якої утворюються закриті криптограми, додавали додаткові символи в такий спосіб, щоби n було менше за m. Під час зашифровування повідомлень у такий спосіб кожен символ відкритого тексту можна було замінити на один з кількох різних символів заздалегідь окресленої підмножини абетки B_n. Ця засада добре ілюструється рис. 2.4. Шифри такого типу називають *шифрами багатозначної заміни,* або *шифрами пропорційної заміни*.

Ідея пропорційної заміни полягає в тому, що символи відкритого тексту, які зустрічаються доволі рідко порівняно з іншими, повинні передбачати більшу кількість варіантів заміни. Така процедура є вельми ефективним способом підвищення стійкості шифру стосовно способів його статистичного криптоаналізу. На відміну від шифрів багатозначної заміни, шифри, котрі передбачають лише один варіант заміни, називають *шифрами безваріантної заміни*.

Наведені вище приклади шифрів заміни є одноабетковими шифрами. Намагання подальшого підвищення їхньої стійкості стосовно різноманітних способів статистичного криптоаналізу призвело до побудови так званих багатоабеткових шифрів. Їхнє формальне подавання наведено вище.

Багатоабетковим шифром заміни є шифр, у якому залежність поміж шифровеличинами у відкритому тексті та відповідними до них шифроперетворюваннями у закритому тексті зберігається лише для окремих частин повідомлення, котре підлягає шифруванню.

Щоби зреалізувати багатоабеткову заміну, повідомлення, котре підлягає зашифровуванню, має бути поділене на блоки однакової довжини (останній блок може мати меншу довжину). Надалі кожен символ чергового блока, котрий має бути зашифрованим, перетворюється на символ криптограми шляхом заміни його на відповідний символ окремої абетки. Сумарна кількість таких абеток дорівнює довжині блока.

Зрозуміло, що шифри багатоабеткової заміни мають вищу стійкість до різноманітних способів статистичного криптоаналізу порівняно з одноабетковими шифрами. Розроблено багато варіантів таких шифрів, котрі забезпечують достатньо високу криптографічну стійкість навіть з огляду вимог нинішнього часу. З-посеред них найбільш цікавими були роторні машини, які дозволяли механізувати процедури шифрування великих обсягів повідомлень.

Шифрування відкритих текстів способом перестановки передбачає зміну позицій шифрвеличин у відкритому повідомленні. У сучасних криптосистемах такі шифри використовуються зазвичай у комбінації з іншими типами криптографічних перетворювань.

Для подальшої формалізації шифрів перестановки надамо означення перестановки.

Перестановкою σ набору цілих чисел від 0 до $N-1$ називатимемо змінювання порядку їхнього запису. Для того, щоби зазначити, що символ a_i переставлено з i-тої позиції на позицію $\sigma(i)$, де $0 \le (i) < n$, скористаємось таким

записом:

$$\sigma = (\sigma(0), \sigma(1),..., \sigma(N-1)).$$

Сумарна кількість переставлянь на множині (0, 1, ..., $N-1$) дорівнює

$$n! = (1 \cdot 2 \cdot 3 \cdot ... \cdot (N-1)N).$$

Далі перестановку набору $S = \{S_0, S_1, ..., S_{N-1}\}$ символів розглядатимемо як взаємно-однозначне відображення на себе:

$$\sigma : S \to S,$$
$$\sigma : s_i \to s_{\sigma(i)}, \ 0 \le i < N-1.$$

Якщо повідомлення, котре підлягає зашифровуванню, сформоване символами абетки A, сумарна кількість яких дорівнює m, сумарна кількість варіантів, які дають змогу утворити перестановку у групі з n символів, дорівнює $m^n!$.

Реалізація шифрів перестановки засадничо вимагає попереднього поділу відкритого тексту на блоки однакової довжини в тих ситуаціях, коли сумарна довжина тексту перевищує певну величину, яка визначається типом шифру. При такому поділі кожен блок зашифровується незалежно від решти блоків.

Довжина блока визначає потужність ключового простору. Якщо вона складається з бінарних символів і дорівнює п’яти, існує 120 способів її переставлянь. Це означає, що для реалізації такого шифру потрібен пристрій, пам’ять якого має вміщувати всі 120 таблиць відповідних переставлянь, тобто 2880 біт. У разі ж, коли довжина блока становить 128 символів, операція переставляння стає технічно неможливою, хоча потужність ключового простору, який вона може забезпечити, дорівнюватиме 2^{128}, або 10^{38}. В цьому полягає фундаментальна дилема криптографії: ми знаємо, що є ідеал, але для нас він є практично недосяжний.

Щодо згаданої ситуації, то слід пам’ятати, що очевидною слабкою стороною шифрів перестановки є повне перенесення статистичних характеристик символів відкритих текстів на символи криптограм. Однак ця властивість не є їхньою невід’ємною негативною властивістю – вона зумовлена обраною довжиною блока.

За приклади шифрів перестановки може слугувати велика кількість табличних шифрів, використовуваних у “докомп’ютерний” період криптографії. У сучасних криптографічних системах операції переставляння символів входять як обов’язкова складова частина загального алгоритму.

2.3.3 Побудова композиційних шифрів

Будь-яка секретна система являє собою відображення множини повідомлень M, які мають передаватись через незахищений канал, у множину криптограм M'.

Раніше було показано, що реалізація таких відображень за допомогою елементарних перетворювань супроводжується перенесенням статистичних характеристик відкритих текстів на відповідні до них криптограми. Намагання скасувати цей недолік певного часу породило ідею побудови композиційних шифрів, суть якої полягає в тому, що комбінація двох чи більшої кількості секретних систем має більшу криптографічну стійкість, аніж кожна зі складових систем окремо. Дотепер існує два основних способи перетворювання двох чи більшої кількості секретних систем на певну спільну систему.

Першим таким способом є використовування операції *множення* кількох секретних систем. Приміром, якщо їх дві – R_1 і R_2 (як це подано на рис.2.5), то спочатку повідомлення зашифровується за допомогою системи R_1, а потім, утворена в такий спосіб криптограма, знову підлягає зашифровуванню вже за допомогою секретної системи R_2. Ключ спільної системи S, який являє собою добуток складових систем і визначається за правилом

$$S = R_1 R_2$$

містить у собі обидва відповідні складові ключі, що обираються незалежно один від одного, кожен з власною ймовірністю.

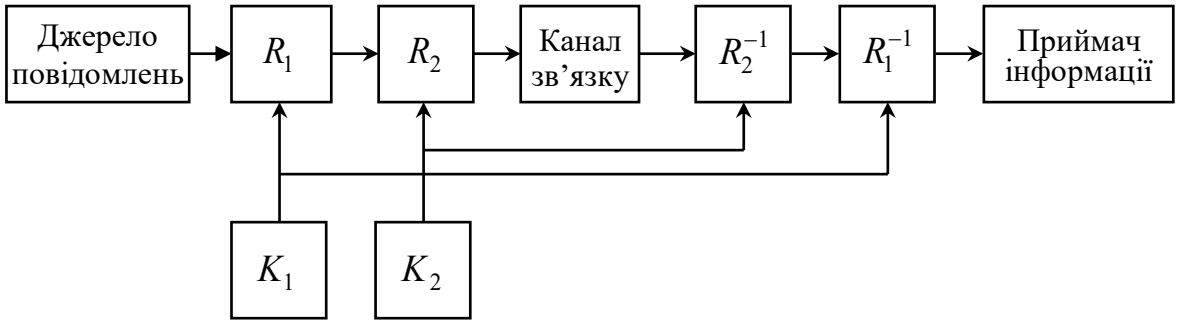


Рисунок 2.5 – Множення двох секретних систем $S = R_1 R_2$

Отже, якщо система R_1 передбачає наявність m ключів, які обираються відповідно з ймовірностями

$$p_1, p_2, \ldots, p_m,$$

а система R_2 – n ключів, які обираються відповідно з ймовірностями

$$p_1', p_2', \ldots, p_n',$$

спільна секретна система матиме mn ключів, які обираються з ймовірностями $p_i p_j'$ відповідно.

Слід пам'ятати, що множення секретних систем у загальному випадку не є комутативним (тобто $R_1 R_2 \neq R_2 R_1$), хоча можуть існувати й виняткові ситуації.

Принцип реалізації складної секретної системи зв’язку з використовуванням множення простих систем подано на рис. 2.5. З цього рисунка видно, що на приймальному боці процедури розшифровування мають виконуватись у зворотному порядку у відповідності до процедур зашифровування на боці джерела повідомлень.

Множення секретних систем у сучасних криптографічних системах використовується досить часто. Наприклад, спочатку використовується шифрування способом підстановки (заміни), а потім — шифрування способом перестановки, або ще якось інакше. Саме такий *лінійний спосіб* побудови складних секретних систем свого часу було рекомендовано К. Шенноном. Він стверджував, що досягнення необхідної стійкості має відбуватись за рахунок чергування достатньої кількості різних елементарних операцій криптографічного перетворювання. При цьому слід зауважити, що набір функційних перетворювань, використовуваних у якості елементарних, обмежується набором команд процесора обчислювального пристрою. Найоптимальніше, якщо такі елементарні перетворювання будуть параметричними, тобто такими, котрі залежать кожне від власного ключа (рис. 2.6).

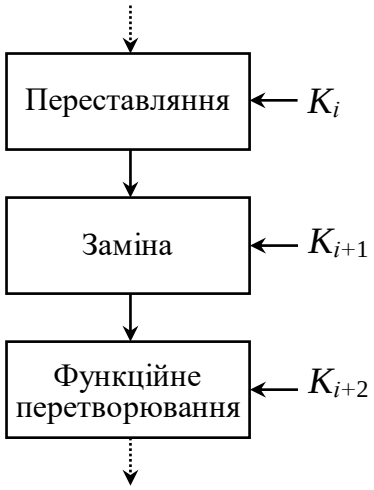


Рисунок 2.6 – Фрагмент складного шифру

Дотримання рекомендацій, запропонованих К. Шенноном, забезпечує необхідну стійкість за рахунок *перемішування* та *розсіювання* символів відкритого тексту.

Розсіювання – це властивість шифру, яка вказує на те, в який спосіб кожен символ у відкритому тексті впливає на шифрування символів у закритому тексті. У оптимальному разі кожен із символів відкритого тексту має впливати на всі символи у закритому тексті. Це означає, що коли у відкритому тексті (блоці) змінити хоча б один символ, то у відповідному до нього закритому тексті (блоці) кожен символ змінить власне значення на протилежне з імовірністю, дорівнюваною 0,5.

Перемішування – це здатність шифру приховувати статистичну залежність поміж символами відкритого тексту та символами у відповідній до нього криптограмі. Якщо алгоритм зашифровування забезпечує доволі якісне

перемішування символів відкритого тексту, то поміж символами відкритого та закритого текстів не існуватиме жодних статистичних та функційних залежностей.

Шифр, який достатньою мірою відповідає цим умовам, може бути розкрито лише повним перебиранням ключів по всій їхній множині. Першою серйозною спробою побудувати такий шифр була система “Люцифер”. Вона передбачає поступове чергування замін та переставлянь символів відкритого тексту.

Спрощену систему “Люцифер” наведено на рис. 2.7.

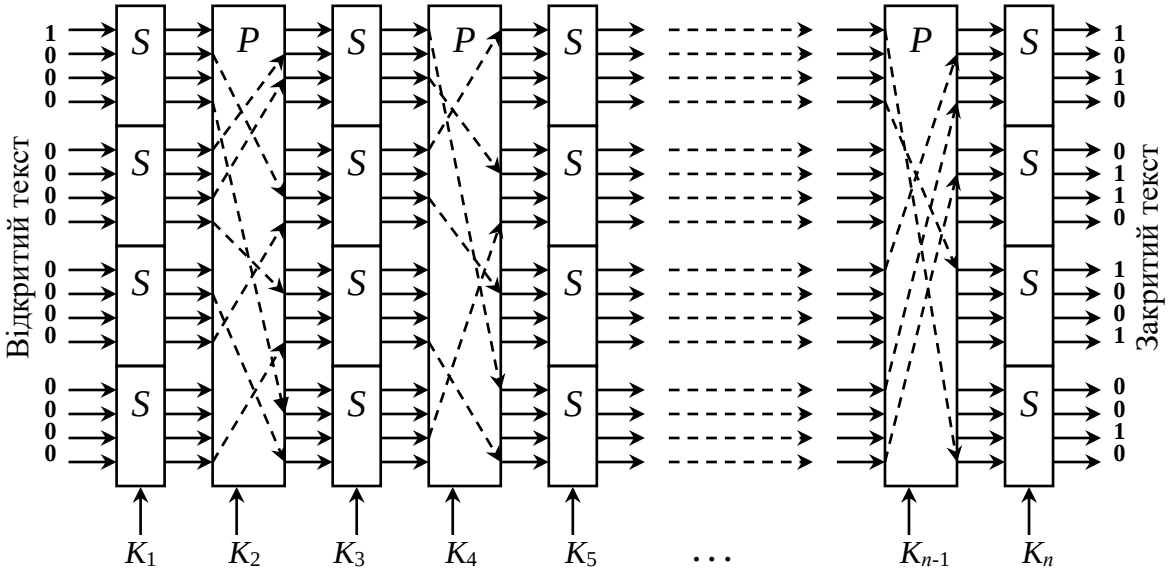


Рисунок 2.7 – Загальна засада побудови системи “Люцифер”

Перед тим як виконувати зашифровування, повідомлення поділяють на блоки фіксованої довжини.

У зв’язку з тим, що реалізація замін та перестановок символів потребує доволі великої пам’яті, черговий блок інформації, який підлягає зашифровуванню, поділяється на блоки меншої довжини. У наведеному на рисунку прикладі 16-розрядний блок поділяється на чотири 4-розрядні блоки, для кожного з яких передбачається наявність 32-бітового блока пам’яті *S*, що забезпечують заміну вхідних символів.

Виходи всіх блоків *S* приєднуються до загального блока *P*, який забезпечує переставляння вхідних символів. Загальна пам’ять, якої потребує реалізація такого блока, має становити лише 32 біти.

Саме виконання рівномірного переставляння у межах всього 16-розрядного блока надає ефективності розсіюванню символів блока, який підлягає зашифровуванню.

Для того щоби наведена система забезпечувала високу криптографічну стійкість, кількість етапів чергування замін та переставлянь повинна бути великою, а кожен символ на її виході має бути складною функцією від символів на всіх її входах. Що стосується секретності цієї системи, то для її забезпечення необхідно, щоби заміни та переставляння, які виконуються блоками *S* та *P*

відповідно, були параметричними. Це означає, що для кожного етапу криптографічного перетворювання має існувати власний окремий і незалежний від інших ключ K_i.

Система "Люцифер" була майже єдиною спробою зреалізувати ефективну лінійну систему криптографічного перетворювання, котра виконувалась корпорацією IBM. Як стало зрозуміло пізніше, забезпечення необхідної криптографічної стійкості у такій системі потребувало великої кількості етапів шифрування і, як наслідок, великих обчислювальних ресурсів.

Другий спосіб формування складних криптографічних систем називається *виваженою сумою* і передбачає наявність кількох простих систем (двох і більш, наприклад R_1 та R_2). Кожного разу перед зашифровуванням чергового повідомлення, випадково, з імовірністю p, обирається система шифрування R_1 або система шифрування R_2, з імовірністю $q = 1 - p$. Отже, в загальному випадку, у криптографічному перетворюванні повідомлення братиме участь одна з двох окремих секретних систем із спільною областю визначання за правилом

$$S = p_1 R_1 + p_2 R_2 .$$

Щоби таку систему можна було практично зреалізувати, необхідно, аби вибір секретної системи повністю зумовлювався складом ключа. В цьому разі буде забезпечено збіжність процедур зашифровування та розшифровування як на боці джерела повідомлень, так і на приймальному боці.

Наведена система передбачає наявність m ключів, які обираються відповідно з ймовірностями

$$p_1,\ p_2,...,\ p_m,$$

кожен з яких перетворює повідомлення M_i на одну з m можливих криптограм. Якщо на підставі обраного ключа буде обрано систему R_1, то повідомлення M_i буде перетворене на одну з криптограм $T_1^1, T_2^1,..., T_m^1$, з ймовірностями $pp_1,\ pp_2,...,\ pp_m$. Якщо ж, навпаки, буде обрано систему R_2, це повідомлення буде перетворене на одну з криптограм $T_1^2, T_2^2,..., T_m^2$, з ймовірностями $qp_1, qp_2,..., qp_m$.

Наведений спосіб формування композиційного шифру дає змогу робити вибір з будь-якої кількості секретних систем. У загальному випадку її можна подати як

$$S = p_1 R_1 + p_2 R_2 + p_3 R_3 + ... + p_n R_n, \quad \sum p_i = 1.$$

Це означає, що криптоаналітик злочинника, окрім визначення ключа, має робити вибір щодо секретної системи, що значно ускладнює дешифрування. Щоби це було насправді так, алгоритм вибору системи повинен також триматись у секреті.

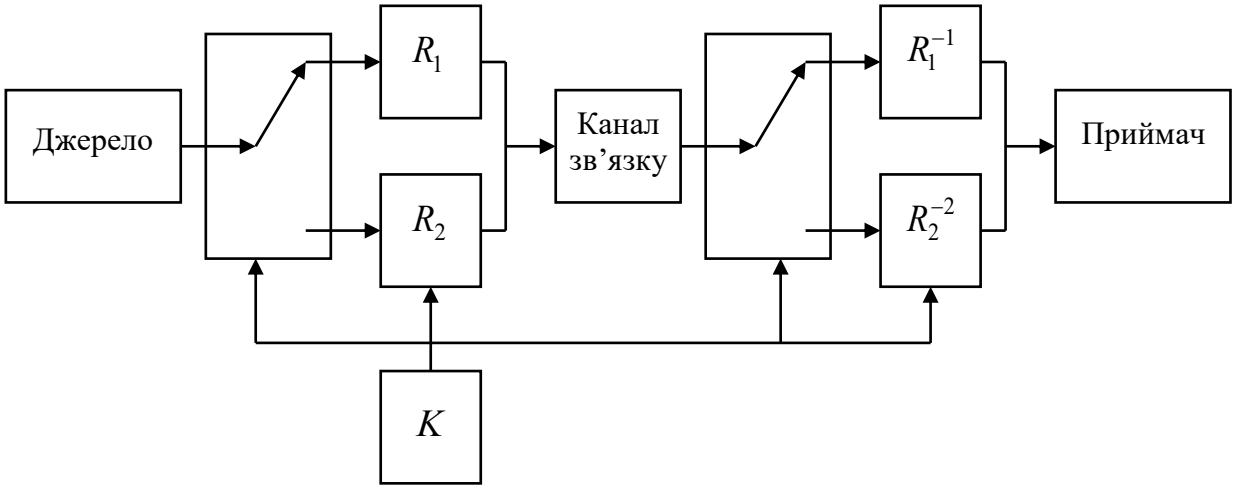


Рисунок 2.8 – Спосіб формування криптосистеми «Виважена сума»

2.4 Блочні шифри

2.4.1 Загальні відомості про блочні шифри

При блочному шифруванні вихідний текст спочатку розбивається на однакові за довжиною блоки, а потім для перетворювання блока відкритого тексту на блок шифртексту застосовується функція шифрування, яка залежить від ключа.

Поділ повідомлення на блоки є пов'язаний з технічними та алгоритмічними особливостями реалізації криптографічних систем.

Вибір довжини блока визначає потужність вхідної абетки. Збільшення такої потужності зумовлює необхідність попереднього досліджування обраних способів криптографічних перетворювань, які обираються до складу алгоритму шифрування. При цьому за критерій якості способу поділу на блоки та архітектури створеного алгоритму обираються експлуатаційні характеристики криптографічної системи. Власне кажучи, потужність вхідної абетки першою чергою залежить від довжини слова мікропроцесора, який використовується обчислювальною системою, котра є у розпорядженні власників інформаційних ресурсів та потенційних порушників. Блочні шифри, на відміну від поточних, мають додаткові можливості підвищування їхньої криптографічної стійкості.

Під час шифрування текстів з малою ентропією існує можливість побудови для них статистичних способів дешифрування, подібних до тих, котрі використовуються при аналізуванні шифрів простої заміни. Підвищування довжини вхідного слова зумовлює зростання середньої ентропії на знак, а це, своєю чергою, призводить до нелінійного зростання складності статистичного криптоаналізу зашифрованого тексту.

Зворотний бік зростання складності криптоаналізу блочних шифрів –

складність доказового обґрунтовування їхньої криптографічної стійкості.

Щоби позбутись цієї проблеми, довжина блока обирається не надто великою і операції шифрування, котрі входять до складу алгоритму шифрування, повторюються кілька разів. У переважної більшості криптографічних систем, які було запропоновано розробниками у якості державних стандартів упродовж 70–80 років двадцятого сторіччя, довжина блока дорівнює 64 бітам. Якщо зменшити довжину блока, дешифрування криптограм легко виконується із застосовуванням незначних обчислювальних ресурсів. Збільшення цієї довжини, навпаки, призводить до зростання терміну шифрування та кількості необхідних обчислювальних ресурсів.

Усі блочні шифри мають ще два суттєвих недоліки, для усунення яких треба вживати спеціальних заходів.

Перший полягає у тому, що довжина вхідного слова, передбаченого алгоритмом шифрування, є фіксованою, а повідомлення, які підлягають шифруванню, в загальному випадку мають довжину, що не є кратною довжині вхідного слова. Це означає, що останній блок, як правило, буде мати меншу довжину. Якщо доповнити його будь-якими символами, наприклад нулями, то, у середньому, половина останнього блока буде відома криптоаналітикові злочинника. А це, своєю чергою, значно зменшить криптографічну стійкість алгоритму шифрування. З метою усунення такого недоліку розроблено як загальні, прийнятні для усіх блочних алгоритмів способи, так й індивідуальні, прийнятні для конкретних криптографічних систем.

Друга проблема полягає у тому, що однакові блоки будуть зашифровані однаково, а це дозволить злочинникові отримувати допоміжну інформацію, що значно підвищить його можливості у виконанні статистичного криптоаналізу. Для усунення цього недоліку також свого часу було запропоновано відповідні заходи щодо захисту, опис яких буде наведено далі.

У сучасних блочних шифрах блоки відкритого тексту та шифртексту являють собою двійкові послідовності довжиною зазвичай 64 біти, тобто кожний блок може набувати 2^{64} значень. Тому підстановки виконуються в абетках надто великого обсягу ($2^{64} \approx 10^{19}$ символів).

Під *N-розрядним блоком* розумітимемо послідовність з одиниць та нулів довжини N:

$$x = (x_0, x_1, \ldots, x_{N-1}) \in Z_{2,N},$$

де x в $Z_{2,N}$ можна тлумачити як вектор та як двійкове подавання цілого числа

$$\|x\| = \sum_{i=0}^{N-1} x_i 2^{N-i-1}.$$

Приміром, якщо $N = 4$, то

(0,0,0,0) → 0	(0,1,0,0) → 4	(1,0,0,0) → 8	(1,1,0,0) → 12
(0,0,0,1) → 1	(0,1,0,1) → 5	(1,0,0,1) → 9	(1,1,0,1) → 13
(0,0,1,0) → 2	(0,1,1,0) → 6	(1,0,1,0) → 10	(1,1,1,0) → 14
(0,0,1,1) → 3	(0,1,1,1) → 7	(1,0,1,1) → 11	(1,1,1,1) → 15

Незважаючи на те, що блочні шифри є випадковою подією підстановок (лише на абетках надто великого розміру), їх треба розглядати окремо.

По-перше, більшість симетричних шифрів, використовуваних в системах передавання повідомлень, є блочними шифрами.

По-друге, блочні шифри зручніше описувати в алгоритмічному вигляді, а не як звичайні підстановки.

Блочним шифром називатимемо елемент $\pi \in \overline{SYM}\left(Z_{2,N}\right)$, π: $x \to y = \pi(x)$, де $x = (x_0, x_1, \ldots, x_{N-1})$, $y = (y_0, y_1, \ldots, y_{N-1})$.

Припустімо, що $\pi(x_i) = y_i$, $0 \le i < m$, для певного $\pi \in \overline{SYM}\left(Z_{2,N}\right)$, вихідного тексту $X = \{x_i: x_i \in Z_{2,N}\}$ та шифртексту $Y = \{y_i\}$.

Якщо $x \notin \{x_i\}$, то $\pi(x_i) \notin \{y_i\}$, оскільки π є перестановкою на $Z_{2,N}$.

Коли відомо, що π належить до невеликої підмножині П з $\overline{SYM}\left(Z_{2,N}\right)$, тоді можна зробити певний висновок. Наприклад, коли

$$\Pi = \{\pi_j: 0 \le j < 2^N\},\ \pi_j(i) \equiv (i + j) \pmod{2^N},\ 0 \le i < 2,$$

то значення $\pi(x)$ за заданого значення x безваріантно визначає π. В цьому разі X є підмножиною підстановки Цезаря на $Z_{2,N}$.

Криптографічне значення цієї властивості має бути явним: якщо вихідний текст шифрується підстановкою π, яку обрано з повної симетричної групи, тоді злочинник, котрий шукатиме відповідність вихідного та шифрованого текстів $x_i \leftrightarrow y_i$, $0 \le i < m$, не зможе визначити вихідний текст, який відповідав би $y \notin \{y_i\}$.

Якщо для шифрування вихідного тексту використовується підсистема π з $\Pi \in \overline{SYM}\,(Z_{2,N})$, тоді систему, яку буде здобуто після підстановок П, називатимемо системою *блочних шифрів*, чи *системою блочних підстановок*.

Ключовою системою блочних шифрів є підмножина П[K] симетричної групи $\overline{SYM}\,(Z_{2,N})$ П[K] = $\{\pi\{K\}: K \in \overline{K}\}$, яка індексується за параметром $K \in \overline{K}$; де K – ключ, а $\overline{K}$ – простір ключів. При цьому треба, щоби різні ключі відповідали різним підстановкам $Z_{2,N}$.

Ключова система блочних шифрів П[K] використовується в такий спосіб. Користувач i та користувач j певним чином укладають угоду відносно ключа K, тобто обирають елемент з П[K] й передають текст, зашифровуваний з використанням обраної підстановки. Для того щоби позначити N-розрядний блок шифрованого тексту, який відповідає N-розрядному блокові вихідного тексту з

використанням підстановки $\pi\{K\}$ за допомогою ключа K, наведемо запис $y = \pi\{K, x\}$.

Припустімо, що злочинникові

– відомий простір ключів;

– відомий алгоритм визначання підстановки $\pi\{K\}$ за значенням ключа K;

– невідомо, який саме ключ обрано користувачем.

Тоді злочинник може:

– здобути ключа внаслідок недбалості користувача i та користувача j;

– перехопити (шляхом перехоплення телефонних та комп'ютерних повідомлень) шифрований текст y, який передається від користувача i до користувача j, а потім „перебирати" усі ключі з ключового простору $\overline{K}$, доти, аж допоки не буде одержано повідомлення вихідного тексту, яке можна було б розуміти;

– здобути відповідні вихідний та шифрований тексти ($x \leftrightarrow y$) та скористатися методом „перебирання" усіх ключів з ключового простору K;

– зробити каталог N-розрядних блоків, де записувалась частість їхньої появи у вихідному та шифрованому текстах.

Каталог надає змогу вишуковувати найвірогідніші слова, користуючись при цьому такою інформацією:

– лістинг мовою асемблера схарактеризовується вельми виявленим структурованим форматом;

– цифрове подавання графічної та звукової інформації має невеликий набір знаків.

Припустімо, що $N = 64$ та кожний елемент $\overline{SYM}\left(Z_{2,N}\right)$ може використовуватися як підстановка, так що $\overline{K} = \overline{SYM}\left(Z_{2,N}\right)$, тоді:

– існує 2^{64} 64-розрядних блоків; злочинник не може використовувати каталог з $2^{64} \approx 1.8 \cdot 10^{19}$ рядками;

– випробування на ключ за кількості ключів, дорівнюваній $(2^{64})!$, практично є неможливими; відповідність вихідного та зашифровуваного текстів для певних N-розрядних блоків $\pi\{K, x_i\} = y_i$, $0 \le i < m$, не надає злочинникові жодної інформації стосовно значення $\pi\{K, x\}$ для $x \notin \{x_i\}$.

Системи шифрування з блочними шифрами, абеткою $Z_{2,64}$ та простором ключів $\overline{K} = \overline{SYM}\left(Z_{2,64}\right)$ є неподільними (тому що виходять за межі можливостей як злочинника, так і того користувача, котрий створив вихідний текст).

Отже, вимоги щодо блочного шифру:

– доволі велике N (64 чи більше), для того щоби ускладнити користування каталогом;

– доволі великий простір ключів, для того щоби виключити можливість підбирання ключів;

– складні співвідношення $\pi\{K, x\}$: $x \to y = \pi(K, x)$ поміж вихідним повідомленням та шифртекстом, для того щоб аналітичні та/чи статистичні

методи визначання вихідного тексту та/чи ключа на базі відповідності вихідного повідомлення та шифртексту не можна було реалізовувати.

2.4.2 Генерування блочних шифрів

2.4.2.1 Використовування нелінійних структур задля побудови блочних шифрів.

Комбінація простих шифрувальних перетворювань у лінійний ланцюжок дозволяє створити цілковито надійний складний шифр. Проблема полягає в тому, що для розв'язування такого завдання потрібна надто велика кількість перетворювань. З цієї причини в сучасних криптосистемах набула поширювання інша схема, відмінність якої полягає у використовуванні нелінійних структур та операцій над шифрувальними блоками даних. Нелінійність структури шифрувального перетворювання набагато ускладнює процедури криптоаналізу. Застосовування, на додаток до цього, в алгоритмі шифрування операцій функційного перетворювання також істотно ускладнює процедуру перетворювання шифрувального блока (кількості можливих варіантів).

Встановлено, що більш оптимально є не перетворювати переданий блок відповідно до застосовуваного ключа, а спочатку скомбінувати цей блок зі значенням ключа, а вже потім піддавати його певному перетворюванню.

У сучасних криптографічних системах, застосовують зазвичай адитивні та, значно рідше, мультиплікативні операції над шифрувальними даними.

Слід мати на увазі, що, застосовуючи певну операцію «$\circ$», треба подбати, щоби для неї існувала обернена до неї операція «$\bullet$», тобто вона має бути придатна для реалізації умов побудови симетричної криптосистеми (рис. 2.10):

$$(M \circ K) \bullet K = M.$$

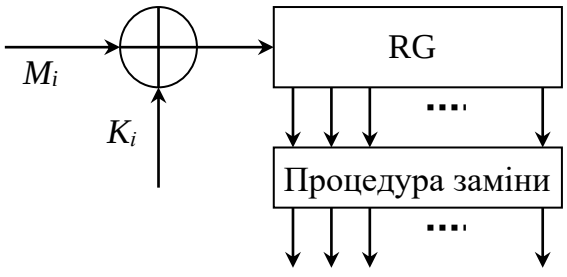


Рисунок 2.9 – Варіант попередньої комбінації даних і ключа

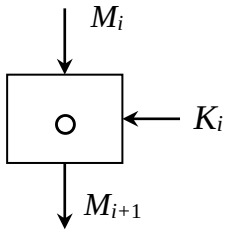


Рисунок 2.10 – Операція над шифрувальними даними на підставі ключа

Складність операції також може бути підвищено, якщо комбінуватиме дані не сам ключ, а певний скомбінований з нього код, котрий залежить від даних:

$$M_{i+1} = M_i \circ f(M_i, K_i).$$

У загальному вигляді цю функцію можна записати як $M_{i+1} = F(M_i, K_i)$.

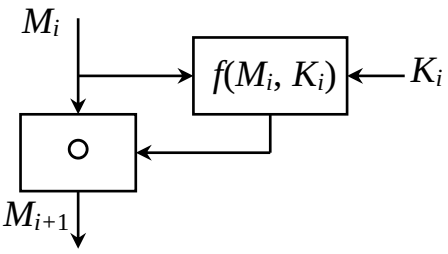


Рисунок 2.11– Операція над шифрувальними даними на підставі функційних перетворювань

Відмінність цих виразів полягає в тому, що функція F повинна мати обернену операцію, а функція f – необов'язково.

Від функції f потрібні максимально можлива складність (кількість можливих варіантів перетворювання) та нелінійність перетворювань. Іншою вимогою до функції є виконання умови оберненості перетворювань. Складність проблеми полягає в тім, що оберненість перетворювань є особливістю лінійних систем. Розв'язок нелінійних рівнянь щодо одного з аргументів у загальному вигляді є відсутній.

Оберненість виразу означає, що має існувати ефективна обчислювальна функція $G(M_{i+1}, K_i)$, яка задовольняла б умові

$$f(M_i, K_i) = G(M_{i+1}, K_i).$$

З розглянутого випливає, що задля побудови ефективного шифру слід визначати пари функцій, які були б нелінійні й, водночас, обернені одна до одної.

За приклад пари операцій такого типу, які широко застосовуються в сучасних симетричних криптосистемах, слугують подані нижче перетворювання.

2.4.2.2 Використовування мереж Файстеля для побудови блочних шифрів

Найпоширенішим способом створювання блочних шифрів є використовування мереж Файстеля (рис. 2.12).

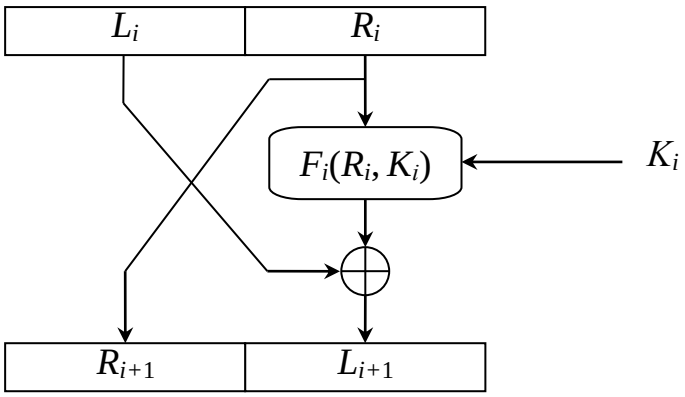


Рисунок 2.12 – Схема мережі Файстеля

Підвищення ефективності шифрування було можливим лише у разі відмови від лінійної структури алгоритму. Але в цьому разі надто важко забезпечити обернений характер криптографічних перетворювань. Проблема полягала у тім, щоби побудувати алгоритм, який не містив би їх взагалі. І це завдання було розв'язано співробітниками лабораторії фірми IBM *Watson Research Lab* нової оригінальної архітектури симетричного шифру на базі необернених перетворювань.

Як відомо, ідея використовування оберненої операції додавання за модулем два, виникла доволі давно. Її запропонував Вернам ще 1925 року для побудови шифру, котрий, як стало зрозуміло значно пізніше, був досконалим. Стійкість такого способу визначається властивостями шифрувальної гами. Спосіб її формування є найскладнішою з проблем сучасної криптографії.

Якщо гама побудована за допомогою будь-якого, реалізованого програмним методом генератора, вона неодмінно буде рекурентною, тобто її вміст матиме скінченну довжину й циклічно повторюватиметься через певну кількість кроків. Визначивши період повторювання такої гами, віднайти її вміст не так вже й складно. В усякому разі, це завдання не є складним для криптоаналітика, котрий має доволі великі обчислювальні ресурси. Якщо ж формувати гаму за допомогою складних апаратних засобів, які використовують не рекурентні, а реальні випадкові процеси, процедура розподілу ключової інформації стає надто складною.

Хорст Файстель підійшов до проблеми в такий спосіб. Він запропонував формувати шифрувальну гаму з самої інформації, яка підлягає шифруванню (рис.2.12).

Спочатку масив інформації, який має бути зашифрованим, поділяється на блоки фіксованої довжини. Кожен з таких блоків під час кожної ітерації шифрування поділяється на дві однакові частини: ліву L_i й праву R_i. Потім з правої частини формується шифрувальна гама, довжина якої дорівнює половині блока. Принцип її формування визначається за допомогою складного нелінійного функційного перетворювання $F_i(R_i, K_i)$, яке залежить від ключа. Перша (ліва) половина нового (утвореного впродовж однієї ітерації) блока L_{i+1} утворюється з правої половини попереднього блока R_i, тобто

$$L_{i+1} = R_i.$$

Щодо другої (правої) половини нового блока, то вона є результатом гамування лівої половини попереднього блока (складанням її "за модулем два" зі створеною гамою) за правилом

$$R_{i+1} = L_i \oplus F_i(R_i, K_i).$$

Як видно з цього виразу, перетворювання правої частини блока залежить від вмісту ключа K_i. Це означає, що для кожної ітерації шифрування потрібен окремий ключ. Отже, як і у схемі, яку запропонував К. Шеннон, кожне

перетворювання блока, який підлягає шифруванню, має бути параметричним (тобто таким, що залежить від певного параметра). Окрім того, зважаючи на те, що впродовж однієї ітерації буде зашифровано лише половину блока, сумарна їхня кількість має бути парною. І самі ключі K_i і функції $F_i(R_i, K_i)$ можуть мати власний вигляд для кожного етапу шифрування. Що стосується проміжних ключів, то вони можуть бути сформовані з головного ключа за допомогою певного алгоритму чи окремо, в такий спосіб, щоби вони були незалежні один від одного.

У разі якщо довжина лівої частини блока дорівнюватиме довжині його правої частини, матиме місце так звана *збалансована* схема Файстеля; в іншому разі – це *розбалансована* схема Файстеля. Це, своєю чергою, потребує формування різних за довжиною шифрувальних гам та різних функцій для виконання відповідних нелінійних перетворювань, залежно від номера ітерації.

Зазвичай виникає запитання про те, чи буде перетворювання, зреалізоване за допомогою схеми Файстеля, оберненим? Властивість схеми Файстеля полягає в тому, що навіть коли в якості нелінійного перетворювання використовується функція, яка сама не є оберненою, або до якої не існує оберненої функції, перетворення залишається оберненим. Річ тут у тім, що для виконання оберненого перетворювання не треба обчислювати функцію $F_i^{-1}(R_i, K_i)$. Єдине, про що слід пам’ятати: під час розшифровування даних проміжні ключі мають використовуватися у зворотному порядку. Більш того, схема Файстеля є симетричною.

Наявність в алгоритмі операції додавання за модулем два дозволяє виконувати операції розшифровування даних за допомогою того самого алгоритму, яким їх було зашифровано.

Стійкість криптосистеми, побудованої на базі схеми Файстеля, цілком залежить від вигляду нелінійної функції гамування $F_i(R_i, K_i)$, яке виконується впродовж кількох ітерацій. Через це, задля забезпечення надійного зашифровування загальна кількість етапів має бути доволі великою. Чим більше їхня кількість, тим оптимальніше виконується розсіювання та перемішування символів відкритого тексту й тим вище буде стійкість шифру до диференційного та лінійного криптоаналізу.

Практично всі відомі шифри побудовано на базі збалансованої схеми Файстеля, а нелінійне функційне перетворювання, використовуване на кожному етапі шифрування, є одне й те саме. Такі схеми називають *гомогенними*, у противному разі – *гетерогенними*. Використовування гетерогенної схеми Файстеля забезпечує потужнішу стійкість шифрів, але їхня побудова являє собою надто складне завдання через необхідність забезпечення оберненості алгоритму шифрування. Популярність гомогенної збалансованої схеми визначається тим, що їхня реалізація є значно дешевше, швидкість шифрування вище, а потрібна стійкість легко досягається ускладненням нелінійної функції шифрування та незначним збільшенням кількості етапів перетворювання відкритого тексту.

Для того щоби ще більш употужнити стійкість гомогенних алгоритмів, ключ, який подається на вхід нелінійної функції, складається за будь-яким

модулем із гамою, утвореною на попередньому етапі. Така операція поширює вплив результатів шифрування кожної ітерації на шифрування на наступних етапах.

Майже всі шифри, котрі стали державними стандартами шифрування у різних країнах світу, використовують схему Файстеля, яка вельми добре пройшла випробування часом. У таких шифрах незначні модифікації, як правило, стосуються додаткових початкових та завершальних операцій над блоком даних, який підлягає шифруванню, і виконують його рандомізацію.

Щодо рандомізації, то ця операція є не що інше як гамування блока даних перед початком його шифрування. Гама повинна мати такі статистичні дані, щоби символи у здобутому після цього блоці були рівномірно розподілені по його довжині. Останніми роками зустрічаються модифікації, які передбачають збільшення кількості гілок у запропонованій схемі. Це пояснюється тим, що збільшення довжини блока понад 128 символів призводить до ускладнення виконання математичних операцій за модулем 64 й вище. Найчастіше зустрічаються схеми, котрі мітять чотири гілки (рис. 2.13, 2.14).

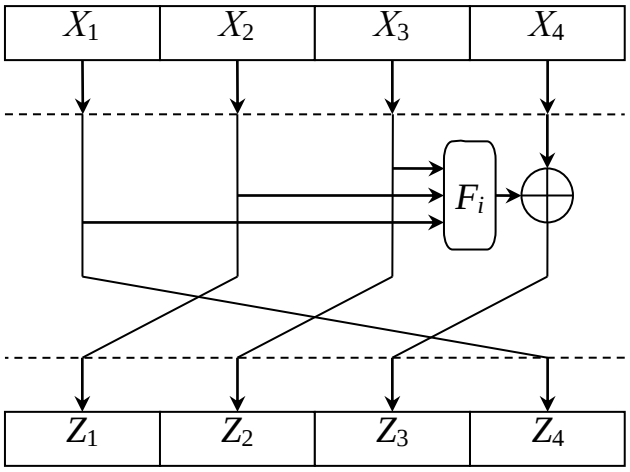


Рисунок 2.13 – Приклад схеми Файстеля з чотирма гілками

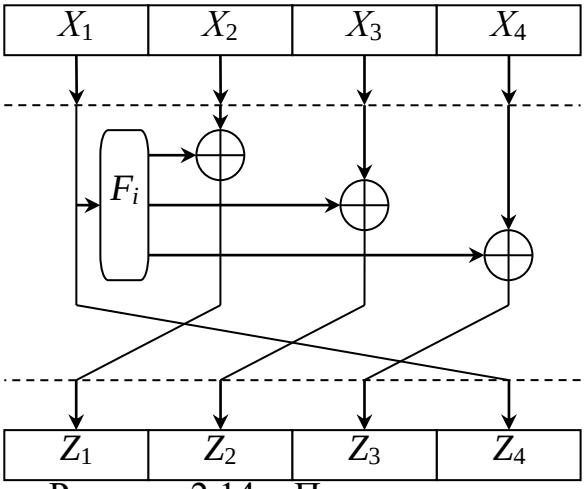


Рисунок 2.14 – Приклад схеми Файстеля з чотирма гілками

Насамкінець слід зауважити, що на понад 95 відсотків стійкість шифру, побудованого на базі схеми Файстеля, визначається складністю функції нелінійного перетворювання $F_i(R_i, K_i)$ і правилом обчислювання ключа K_i.

2.4.2.3 Вимоги щодо побудови блочних криптосистем

1. Знання алгоритму шифрування не повинно знижувати криптостійкість шифру.

2. Прочитання зашифрованого повідомлення повинно бути можливим тільки за допомогою ключа шифрування.

3. Шифр має залишатися стійким навіть тоді, коли зловмисникові відомо досить велика кількість вихідних даних і відповідних їм зашифрованих даних.

4. Кількість операцій, необхідних для дешифрування повідомлення шляхом перебору всіх можливих ключів, повинно мати сувору нижню оцінку та повинно чи виходити за межі можливостей сучасних комп'ютерів, чи вимагати використання дуже коштовних обчислювальних систем.

5. Незначна зміна ключа або вихідного тексту повинна приводити до істотної зміни шифртекста.

6. Структурні елементи алгоритму шифрування повинні бути незмінними.

7. Довжина шифртекста повинна дорівнювати довжині вхідного тексту.

8. Додаткові біти, що вводяться в повідомлення в процесі шифрування, мають бути повністю і надійно приховані в шифртексті.

9. Не повинно бути простих і легко встановлюваних залежностей між ключами, які послідовно використовуються в процесі шифрування.

10. Будь-який ключ з безлічі можливих повинен забезпечувати надійний захист інформації.

2.5 Поточні шифри

Зазвичай, шифри, котрі потребують попереднього поділяння відкритого тексту на блоки однакової довжини й подальшого зашифровування кожного з них окремо, називають *блочними шифрами*. У противному разі, мають місце *поточні шифри*. Інакше кажучи, в поточних алгоритмах кожен символ відкритого тексту шифрується незалежно від інших і розшифровується в такий самий спосіб.

Поточне шифрування полягає в тому, що біти вихідного тексту додаються за модулем два до псевдовипадкової послідовності, яку іноді називають гамою. Від того, наскільки утворена гама матиме властивість рівноймовірності появи її символів, залежить стійкість поточних алгоритмів шифрування.

До переваг поточних шифрів слід віднести те, що вони мають високу швидкість шифрування, відносно просто зреалізовуються та при цьому методі шифрування відсутнє розповсюдження помилок. Недоліком їхнім є необхідність передавання синхронізувальної інформації раніш за повідомлення. Це може становити загрозу криптостійкості системи. Тому використовують додатковий, випадковий ключ повідомлення, який має модифіковувати ключі, які шифрують повідомлення.

На практиці цей метод шифрування використовують тоді, коли треба зашифрувати повідомлення потужної довжини.

2.6 Шифри гамування

2.6.1 Накладання гами шифру на відкритий текст

Гамування не можна цілковито виокремити в окремий клас криптографічних перетворювань, позаяк ця псевдовипадкова послідовність може створюватися, наприклад, за допомогою блочного шифру.

Під гамуванням розуміють процес накладання за певним законом гами шифру на відкриті дані.

Гама шифру – це псевдовипадкова послідовність, створена за заданим алгоритмом для зашифровування відкритих даних і розшифровування зашифрованих даних.

Процес зашифровування полягає в генеруванні гами шифру і накладанні здобутої гами на вихідний відкритий текст у зворотний спосіб, приміром з використовуванням операції додавання за модулем два.

Слід зазначити, що перед зашифровуванням відкриті дані розбивають на блоки $M^{(i)}$ однакової довжини, зазвичай по 64 біти. Гама шифру створюється у вигляді послідовності блоків $Г_{ш}^{(i)}$ аналогічної довжини.

Рівняння зашифровування можна записати у вигляді

$$C^{(i)} = M^{(i)} \oplus Г_{ш}^{(i)},$$

де $C^{(i)}$ – i-тий блок шифртексту; $Г_{ш}^{(i)}$ – i-тий блок гами шифру; $M^{(i)}$ – i-тий блок відкритого тексту.

Процес розшифровування зводиться до накладання тієї самої гами шифру на зашифровані дані. Рівняння розшифровування має вигляд

$$M^{(i)} = C^{(i)} \oplus Г_{ш}^{(i)}.$$

Здобуваний цим методом шифртекст є вельми складний для розкриття, оскільки тепер ключ є змінним. По суті, гама шифру повинна змінюватися у випадковий спосіб для кожного зашифровуваного блока. Якщо період гами перевищує довжину всього зашифровуваного тексту і злочинникові невідома жодна з частин відкритого тексту, то такий шифр можна розкрити лише безпосереднім перебиранням усіх варіантів ключа. У цьому разі криптостійкість шифру визначається довжиною ключа.

2.6.2 Методи генерування псевдовипадкових послідовностей чисел

При шифруванні методом гамування як ключ використовується випадковий рядок бітів, котрий поєднується з відкритим текстом, також поданим у двійковому вигляді (наприклад A = 00000, B = 00001, C = 00010 і т. д.), за допомогою побітового додавання за модулем два – і в результаті виходить шифрований текст.

Генерування непередбачуваних двійкових послідовностей великої довжини є однією з важливих проблем класичної криптографії. Для розв’язування цієї проблеми широко використовуються генератори двійкових псевдовипадкових послідовностей.

Генеровані псевдовипадкові ряди чисел часто називають гамою шифру, чи просто гамою (за назвою літери γ грецької абетки, часто використовуваної в математичних формулах для позначання випадкових величин).

Зазвичай для генерування послідовності псевдовипадкових чисел застосовують комп'ютерні програми, котрі, хоча й називаються генераторами випадкових чисел, насправді видають детерміновані числові послідовності, котрі за своїми властивостями є надто схожі на випадкові.

До криптографічно стійкого генератора псевдовипадкової послідовності чисел (гами шифру) пред'являються три основних вимоги:

– період гами повинен бути доволі великим для шифрування повідомлень різної довжини;

– гама має бути практично непередбачуваною, що означає неможливість завбачати наступний біт гами, навіть якщо відомі є тип генератора і попередній фрагмент гами;

– генерування гами не повинно спричинюватись до значних технічних складностей.

Довжина періоду гами є найважливішою характеристикою генератора псевдовипадкових чисел. По завершенні періоду числа розпочнуть повторюватися – і їх можна буде завбачати. Необхідна довжина періоду гами визначається мірою закритості даних. Чим довше є ключ, тим складніше його добрати. Довжина періоду гами залежить від обраного алгоритму здобування псевдовипадкових чисел.

Друга вимога пов'язана з такою проблемою: в який спосіб можна вірогідно переконатися, що псевдовипадкова гама конкретного генератора є насправді непередбачуваною? На сьогодні не існує таких універсальних і практично перевірних критеріїв та методик. Щоби гама вважалася непередбачуваною, тобто істинно випадкову, необхідно, аби її період був доволі великим, а різноманітні комбінації бітів певної довжини було рівномірно розподілено по всій її довжині.

Третя вимога зумовлює можливість практичної реалізації генератора програмним чи апаратним шляхом із забезпеченням потрібної швидкодії.

Один з перших способів генерування псевдовипадкових чисел на ЕОМ запропонував 1946 року Джон фон Нейман. Суть цього способу полягає в тому, що кожне наступне випадкове число утворюється піднесенням до квадрата попереднього числа з відкиданням цифр молодших і старших розрядів. Однак цей спосіб виявився ненадійним – і від нього невдовзі відмовились.

З відомих процедур генерування послідовності псевдовипадкових цілих чисел найчастіше застосовується так званий лінійний конгруентний генератор.

Цей генератор продукує послідовність псевдовипадкових чисел Y_1, Y_2, ..., Y_{i-1}, Y_i, використовуючи співвідношення

$$Y_i \equiv (aY_{i-1} + b) \bmod m,$$

де Y_i – i-те (поточне) число послідовності; Y_{i-1} – попереднє число послідовності; a, b та m – константи; m – модуль; a – множник (коефіцієнт); b – приріст. Поточне псевдовипадкове число Y_i дістають з попереднього числа Y_{i-1} множенням його на коефіцієнт a, додаванням з приростом b та обчислюванням остачі від ділення на модуль m. Дане рівняння генерує псевдовипадкові числа з періодом

повторювання, котрий залежить від обираних значень параметрів a, b та m і може сягати значення m. Значення модуля m обирається дорівнюваним 2^n або простому числу, наприклад $m = 2^{31} - 1$. Приріст b має бути взаємно простим з m, коефіцієнт a має бути непарним числом.

Конгруентні генератори, які працюють за алгоритмом, запропонованим Національним бюро стандартів США, використовуються, зокрема, в системах програмування. Ці генератори мають довжину періоду 2^{24} і добрі статистичні властивості. Однак така довжина періоду є замала для криптографічних застосовувань. Окрім того, доведено, що послідовності, генеровані конгруентними генераторами, не є криптографічно стійкі.

Існує спосіб генерування послідовностей псевдовипадкових чисел на підставі лінійних рекурентних співвідношень.

Розглянемо рекурентні співвідношення та їхні різницеві рівняння:

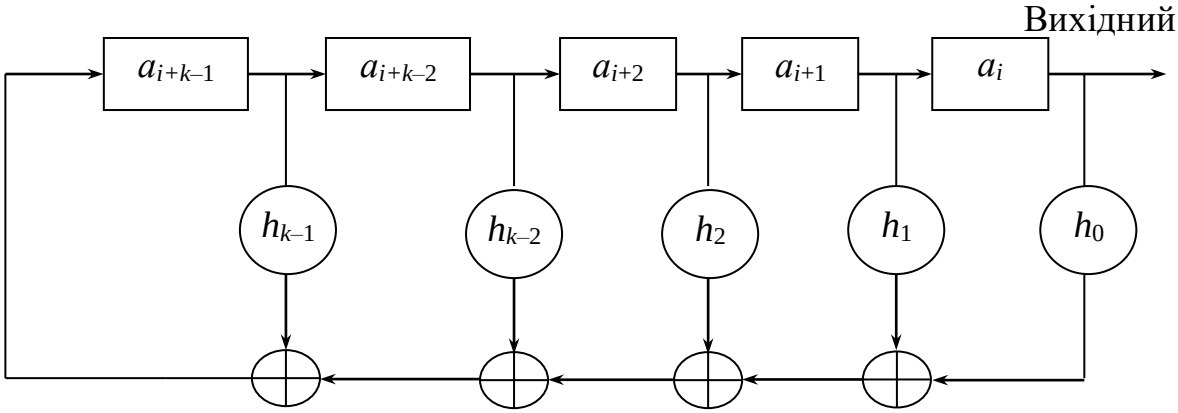
$$\sum_{j=0}^{k} h_j a_{i+j} = 0; \quad a_{i+k} = -\sum_{j=0}^{k-1} h_j a_{i+j}, \qquad (2.1)$$

де $h_0 \neq 0$, $h_k = 1$ і кожне h_i належить до поля $GF(q)$.

Розв'язком цих рівнянь є послідовність елементів a_0, a_1, a_2, ... поля $GF(q)$. Співвідношення (2.1) визначає правило обчислювання a_k за відомими значеннями величин $a_0, a_1, a_2, ..., a_{k-1}$. Потім за відомими значеннями $a_0, a_1, a_2, ..., a_k$ визначають a_{k+1} і т. д. Як наслідок за початковими значеннями $a_0, a_1, a_2, ..., a_{k-1}$ можна побудувати нескінченну послідовність, причому кожен її наступний член визначається з k попередніх. Послідовності такого вигляду легко реалізовуються на комп'ютері; при цьому реалізація виходить надто простою, якщо всі h_i та a_i набувають значень 0 та 1 з поля $GF(2)$.

На рис. 2.15 подано лінійну послідовну перемикальну схему, яку може бути використано для обчислювання суми (2.1) і, отже, для обчислювання значення a_k за значеннями k попередніх членів послідовності.

Вихідні величини $a_0, a_1, a_2, ..., a_{k-1}$ вміщують в розряди регістру зсування, послідовні зсунення вмісту якого відповідають обчислюванню послідовних символів; при цьому вихід i-го зсунення дорівнює a_i. Даний пристрій називають генератором послідовності чисел, побудованим на базі регістру зсування з лінійним зворотним зв'язком



Позначення:

– суматор за модулем 2;

h – ланцюг (відвід) з коефіцієнтом передавання h, $h = 0$ або 1;

a – запам'ятовувальна комірка, котра зберігає a, тобто на виході комірки $a = 0$ або $a = 1$.

Рисунок 2.15 – Генератор з регістром зсування

.

Розв'язки лінійних рекурентних співвідношень, реалізовувані генератором з регістром зсування, описуються такою теоремою. Нехай багаточлен

$$h(X) = \sum_{j=0}^{k} h_j X_j,$$

де X – формальна змінна; h_j – коефіцієнт при X_j, який набуває значення 0 чи 1; $h_0 \neq 0$, $h_k = 1$, і нехай n – найменше ціле додатне число, для якого багаточлен $X^n - 1$ ділиться на $h(X)$. Окрім того, багаточлен

$$g(X) = (X^n - 1)/h(X),$$

тоді розв'язки рекурентних співвідношень

$$\sum_{j=0}^{k} h_j a_{i+j} = 0$$

у вигляді послідовності елементів a_0, a_1, a_i, …, a_{n-1} є періодичні з періодом n і сукупність, складена з перших періодів усіх можливих розв'язків, розглядуваних як багаточлени за модулем $(X^n - 1)$, тобто

$$a(X) = a_0X^{n-1} + a_1X^{n-2} + \ldots + a_{n-2}X + a_{n-1},$$

збігається з ідеалом, породженим багаточленом $g(X)$ в алгебрі багаточленів за модулем $(X^n - 1)$.

Зауважимо, що якщо за такого визначення багаточлена $a(X)$ елементи a_0, a_1, a_2, … обчислюються в порядку зростання номерів, то коефіцієнти багаточлена $a(X)$ обчислюються, розпочинаючи з коефіцієнтів зі степенями вищих порядків. Слід також зазначити, що вигляд багаточлена

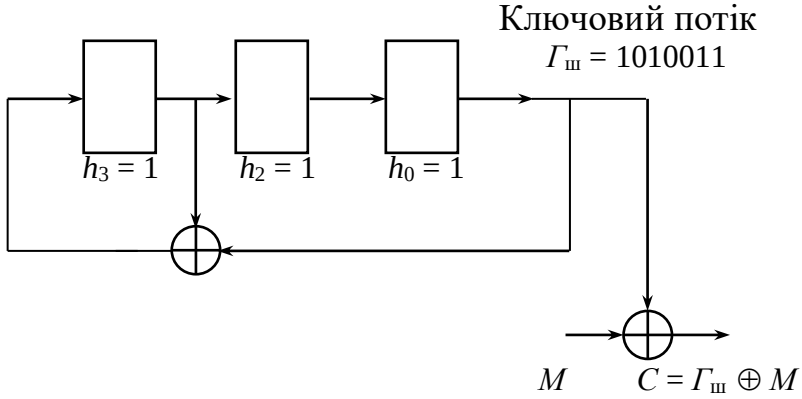
$$h(X) = \sum_{j=0}^{k} h_j X_j,$$

визначає конфігурацію зворотних зв'язків (відводів) h_j в генераторі з регістром зсування. Інакше кажучи, якщо в багаточлена $h(X)$ коефіцієнт $h_j = 1$, це означає, що відвід h_j у схемі генератора є наявний, якщо ж у багаточлена $h(X)$ коефіцієнт $h_j = 0$, то відвід h_j в схемі генератора є відсутній. У якості $h(X)$ слід обирати незвідний примітивний багаточлен. За такого обрання багаточлена $h(X)$ зі старшим степенем m генератор забезпечує видавання псевдовипадкової послідовності двійкових чисел з максимально можливим періодом $2^m - 1$.

Розглянемо за приклад трирозрядний регістр зсування з лінійним зворотним зв'язком (рис. 2.16), побудований відповідно до незвідного примітивного багаточлена

$$h(X) = X^3 + X^2 + 1,$$

де коефіцієнти $h_3 = 1$, $h_2 = 1$, $h_1 = 0$, $h_0 = 1$.



Трибітовий ключ

1	0	1
0	1	0
0	0	1
1	0	0
1	1	0
1	1	1
0	1	1

Рисунок 2.16 – Трирозрядний регістр зсування зі зворотними зв'язками

Нехай ключем є послідовність 101. Регістр розпочинає працювати з цього стану; послідовність станів регістру наведено на рис. 2.16. Регістр проходить через усі сім ненульових станів – і знову повертається до свого вихідного стану 101. Це є найдовший період даного регістру з лінійним зворотним зв'язком. Така послідовність називається *послідовністю максимальної довжини* для регістру зсування (Maximal Length Shift Register Sequence – MLSRS). За будь-якого цілого m існує m-бітова послідовність MLSRS з періодом $2^m - 1$. Зокрема за $m = 100$ послідовність матиме період $2^{100} - 1$ і не повторюватиметься 10^{16} років при передаванні лініями зв'язку зі швидкістю 1 Мбіт/с.

У нашому прикладі вихідною послідовністю (гамою шифру) $Г_ш$ регістру зсування зі зворотним зв'язком є послідовність 1010011, котра циклічно повторюється. У цій послідовності є чотири одиниці й три нулі та їхній розподіл є настільки близький до рівномірного, наскільки це є можливе в послідовності, котра має довжину 7. Якщо розглянути пари послідовних бітів, то пари 10 та 01 з'являються двічі, а пари 00 та 11 – одноразово, що знову стає настільки близьким до рівномірного розподілу, наскільки це є можливо. У разі послідовності максимальної довжини для m-розрядного регістру ця властивість рівнорозподілюваності поширюється на трійки, четвірки й т. д. бітів, аж до m-бітових груп. Внаслідок такої близькості до рівномірного розподілу послідовності максимальної довжини часто використовуються в якості псевдовипадкових послідовностей у криптографічних системах, котрі імітують роботу криптостійкої системи одноразового шифрування.

Хоча така криптографічна система здійснює імітацію завбачувано криптостійкої системи одноразового шифрування, сама вона не відрізнюється стійкістю і може бути розкрита за кілька секунд роботи комп'ютера за умови наявності відомого відкритого тексту.

Якщо відводи регістру зі зворотним зв'язком зафіксовано, то для віднайдення початкового стану регістру доволі знати m бітів відкритого тексту. Щоби знайти m бітів ключового потоку, m бітів відомого відкритого тексту складають за модулем два з відповідними m бітами шифртексту. Здобуті m бітів надають стан регістру зсування зі зворотним зв'язком у зворотному напрямку на певний момент часу. Потім, моделюючи роботу регістру у зворотному напрямку, можна визначити його вихідний стан.

Якщо відводи регістру зі зворотним зв'язком не є фіксовані, а є частиною ключа, то досить $2m$ бітів відомого відкритого тексту, аби порівняно швидко визначити розташування відводів регістру та його початковий стан.

Нехай $S(i)$ – вектор-стовпець, який складається з m символів 0 та 1 й визначає стан регістру в i-тий момент часу. Тоді

$$S(i + 1) = AS(i) \bmod 2,$$

де A – матриця розміром $m \times m$, котра визначає положення відводів регістру зі зворотним зв'язком.

Для трирозрядного регістру (див. рис. 2.16)

$$A = \begin{vmatrix} 1 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{vmatrix}.$$

Матриця A завжди має таку структуру: у її першому рядку відбито послідовність відводів у регістрі, безпосередньо під головною діагоналлю розташовуються одиниці, а в решті позицій – нулі.

$2m$ бітів відомого відкритого тексту дозволяють обчислити $2m$ послідовних бітів ключового потоку. Для спрощення позначень припустімо, що це – перші $2m$ бітів ключового потоку. Отже,

$S(1)$ – перша група m відомих бітів ключового потоку;

$S(2)$ – наступна група (розпочинаючи з номера 2) з m відомих бітів ключового потоку;

$S(m + 1)$ – остання група з m відомих бітів ключового потоку.

Далі можна утворити дві матриці розміром $m \times m$:

$$X(1) = [S(1), S(2), \ldots, S(m)];$$

$$X(2) = [S(2), S(3), \ldots, S(m + 1)],$$

пов’язані співвідношенням

$$X(2) = AX(1) \bmod 2.$$

Можна довести, що для будь-якої послідовності максимальної довжини матриця $X(1)$ завжди є несингулярна, тому матрицю A можна обчислити як

$$A = X(2)\,[X(1)]^{-1} \bmod 2.$$

Обертання матриці $X(1)$ потребує (щонайбільш) біля m^3 операцій, тому легко виконується за будь-якого розумного значення m.

Для криптографії послідовності максимальної довжини MLSRS можна зробити більш криптостійкими, використовуючи нелінійну логіку. Зокрема, як ключовий потік використовується нелінійно "фільтрований" вміст регістру зсування, а для здобуття послідовності максимальної довжини – лінійний зворотний зв’язок, як це подано на рис. 2.17.

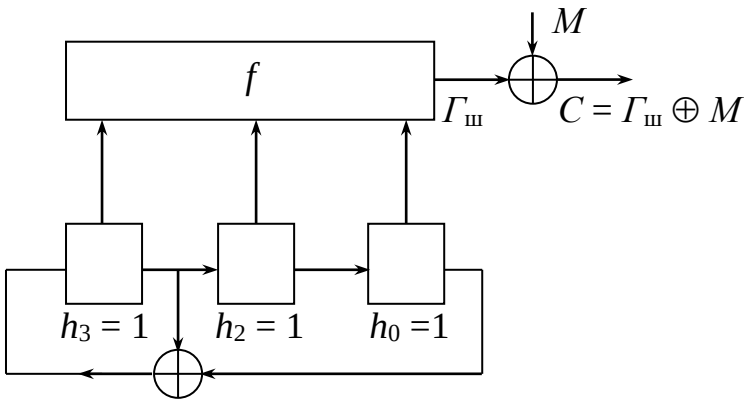


Рисунок 2.17 – Лінійний регістр зсування з нелінійними логічними ланцюгами на виході

Функція f має обиратися у такий спосіб, аби забезпечити оптимальний баланс поміж нулями й одиницями, а „фільтрована" послідовність має розподіл, близький до рівномірного. Необхідно також, аби „фільтрована" послідовність мала великий період. Якщо $(2^m - 1)$ є простим числом (як у прикладі: за $m = 3$ маємо $2^3 - 1 = 7$), то „фільтрована" послідовність може мати період $(2^m - 1)$ (при обранні структури регістру зсування відповідно до незвідного примітивного багаточлена $h(X)$ степеня m). До згаданих значень m належать, зокрема, такі: 2, 3, 5, 7, 13, 17, 19, 31, 61, 89, 107, 127, 521, 607, 1279, 2203, 2281, а здобуті в такий спосіб прості числа називаються *простими числами Мерсена*.

Незважаючи на те, що „фільтровану" вихідну послідовність зазвичай не можна здобути за допомогою m-розрядного регістру зсування з лінійним зворотним зв'язком, її завжди можна здобути за допомогою регістру зсування більшої довжини з лінійним зворотним зв'язком. Регістр довжиною $(2^m - 1)$ завжди дозволить це зробити, а іноді для цього придатний і більш короткий регістр.

2.6.3 Завдання для самоперевірки з теми

Завдання 2.6.1 Закодувати свої іменні дані рівномірним двійковим кодом.

Завдання 2.6.2 Зашифрувати текст методом гамування на підставі даних, які були отримані у завданні 2.6.1, де: вихідне повідомлення – прізвище та ім'я студента, гама шифру – по-батькові студента.

Завдання 2.6.3 Розшифрувати криптограму, яка була отримана в завданні 2.6.2

Контрольні питання з теми

1. У чому полягає сутність процесу гамування?
2. Що називають гамою шифру?
3. Чому має дорівнювати гама шифру?
4. Як здійснюється процес зашифрування методом гамування?
5. Як здійснюється процес розшифрування методом гамування?

3 СИМЕТРИЧНІ КРИПТОГРАФІЧНІ СИСТЕМИ

3.1 Класичні симетричні криптосистеми

Класична криптографія, зокрема теорія зв'язку в секретних системах, заснована К. Шенноном [22], виходила з того, що для шифрування й розшифровування інформації використовується один ключ і його передавання має здійснюватися надійним каналом обміну ключовою інформацією. Подібні алгоритми названо симетричними.

Класичні методи зашифровування відрізняються симетричною функцією зашифровування. До них відносять шифри перестановки, шифри простої й складної заміни, а також певні їхні модифікації й комбінації. Слід зазначити, що комбінації шифрів перестановки й шифрів заміни утворюють усе різноманіття застосовуваних на практиці симетричних шифрів.

Різноманітні симетричні криптосистеми базуються на поданих нижче основних класах перетворювань.

Перестановки. Простий метод криптографічного перетворювання, який містить правило переставляння літер у відкритому тексті. Шифри перестановки мають невелику криптостійкість, тому їх не використовують без додаткових перетворювань.

Одно- та багатоабеткові підстановки. Одноабеткові підстановки – це простий метод перетворювань, який містить правило заміни символів вихідного тексту на інші символи тої самої абетки. В разі однооабеткової підстановки (шифри простої заміни) кожний символ вихідного тексту замінюється на символ шифрованого тексту за одним законом перетворювання.

При шифруванні за допомогою щифрів багатоабеткової підстановки (шифрів складної заміни) закон перетворювання змінюється від символу до символу. Іноді один і той самий шифр може розглядатися і як одно- , і як багатоабетковий залежно від визначуваної абетки. Наприклад, шифр Плейфейра (підстановка біграм) з огляду на звичайну абетку є одноабетковим, а з огляду на абетку біграм – багатоабетковим.

Через певний час симетричні алгоритми було поділено на два великих класи – блочні й поточні.

Блочні шифри. Фактично блочний шифр – це система підстановки в абетці блоків (вона може бути як одно-, так і як багатоабетковою, залежно від режиму блочного шифру). Застосовування блочних алгоритмів криптозахисту припускає поділ переданої до каналу зв'язку послідовності символів відкритого тексту на блоки фіксованої довжини з подальшим шифруванням кожного блока окремо. При цьому однаковим шифрувальним блокам відповідатимуть однакові

шифртексти. На сьогодні блочні шифри є найбільш поширені. Приміром, вітчизняний та американський стандарти шифрування належать до блочних шифрів.

Поточні шифри. У поточних алгоритмах кожен символ відкритого тексту шифрується незалежно від інших і розшифровується в такий же спосіб. Інакше кажучи, перетворювання кожного символу відкритого тексту змінюється від одного символу до іншого.

При поточному шифруванні довжина ключової послідовності дорівнює довжині вихідного повідомлення. Її іноді називають гамою. Стійкість поточних алгоритмів шифрування залежить від того, наскільки утворена гама матиме властивість рівноймовірності появи її символів.

Поточні алгоритми мають високу швидкість шифрування, а їхня структура дозволяє здійснювати ефективну апаратну реалізацію, однак при програмному використовуванні виникають певні труднощі, які звужують область практичного застосовування таких алгоритмів.

Гамування – це перетворювання вихідного тексту, за якого символи відкритого тексту складаються з символами псевдовипадкової послідовності (гамою). Гамування не можна цілковито виділити в окремий клас криптографічних перетворювань, позаяк ця псевдовипадкова послідовність може утворюватися, приміром, за допомогою блочного шифру. В разі, коли послідовність є насправді випадковою і її кожний фрагмент використовується лише одиноразово, то маємо криптосистему з одноразовим ключем.

Ще однією проблемою використовування алгоритмів, котрі припускають гамування, є вимога щодо синхронності виконування операцій шифраторами на приймальному й передавальному боці.

3.2 Криптосистема Хілла

Лестером Хіллом було сформульовано алгебричний метод, котрий узагальнює афінну підстановку Цезаря

$$E_{a,b}: \rightarrow \overline{Z}_m \rightarrow \overline{Z}_m;$$

$$E_{a,b}: t \rightarrow E_{a,b}(t);$$

$$E_{a,b}(t) = (at + b) \bmod m,$$

де a, b – цілі числа, $0 \le a, b < m$; НСД(a, m) = 1 для визначання n-грам.

Множина цілих $\overline{Z}_m$, для якої визначено операції додавання, віднімання та множення за модулем m, являє приклад кільця. Кільце являє собою алгебричну систему, в якій визначено операції додавання, віднімання та множення пар елементів.

Нехай $\overline{A}$ є лінійним перетворюванням, описуваним квадратною матрицею, причому

$$\overline{A}: \overline{Z}_m \to \overline{Z}_m.$$

Криптосистема, розроблена Хіллом, базується на лінійній алгебрі.

Нехай простори вихідних повідомлень та криптотекстів збігаються і дорівнюють $\sum$, де $\sum$ – англійська абетка. Надамо літерам номери відповідно до порядку їхнього слідування в абетці (табл. 3.1).

Таблиця 3.1 – Відповідність поміж англійською абеткою та множиною цілих

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z

Всі арифметичні операції виконуються за модулем 26 (кількість літер в абетці). Це означає, що 26 ототожнюється з 0; 27 – з 1; 28 – з 2 і т. д.

Оберемо ціле число $d \geq 2$. Воно зазначає розмірність використовуваних матриць. У процедурі зашифровування набори з d літер вихідного повідомлення зашифровуються разом. Візьмемо $d = 2$.

Нехай тепер A – квадратна $d \times d$ матриця. Елементами A є цілі числа від 0 до 25. Вимагатимемо далі, аби матриця A була невиродженою, тобто існувала обернена матриця A^{-1}. Приміром,

$$A = \begin{pmatrix} 3 & 3 \\ 2 & 5 \end{pmatrix} \quad \text{та} \quad A^{-1} = \begin{pmatrix} 15 & 17 \\ 20 & 9 \end{pmatrix}.$$

Нагадаємо, що арифметичні операції провадяться за модулем 26. Це дає, приміром,

$$2 \cdot 17 + 5 \cdot 9 = 79 = 1 + 3 \cdot 26 = 1,$$

тобто, як і мало бути, одиницю на головній діагоналі одиничної матриці.

Зашифровування здійснюється за допомогою рівняння

$$AP = C,$$

де P та C – d-розмірні вектори-стовпці. Більш докладно: кожен набір з d літер вихідного повідомлення визначає вектор P, компонентами якого є номери літер. Врешті, C знову інтерпретується як набір d літер криптотексту.

Приміром, HELP визначає два вектори:

$$P_1 = \begin{pmatrix} H \\ E \end{pmatrix} = \begin{pmatrix} 7 \\ 4 \end{pmatrix} \quad \text{та} \quad P_2 = \begin{pmatrix} L \\ P \end{pmatrix} = \begin{pmatrix} 11 \\ 15 \end{pmatrix}.$$

З рівнянь

$$AP_1 = \begin{pmatrix} 7 \\ 8 \end{pmatrix} = C_1 \quad \text{та} \quad AP_2 = \begin{pmatrix} 0 \\ 19 \end{pmatrix} = C_2$$

дістаємо криптотекст HITE.

Розглянемо тепер сферу діяльності криптоаналітика. Припустімо, що аналітик здогадався, що $d = 2$. Йому потрібно віднайти матрицю A чи, ще краще, обернену матрицю A^{-1}. З цією метою він обирає вихідне повідомлення HELP і довідується, що відповідний криптотекст є HITE. Криптоаналітикові відомо, що

$$A \begin{pmatrix} 7 \\ 4 \end{pmatrix} = \begin{pmatrix} 7 \\ 8 \end{pmatrix} \quad \text{та} \quad A \begin{pmatrix} 11 \\ 15 \end{pmatrix} = \begin{pmatrix} 0 \\ 19 \end{pmatrix}.$$

Це може бути записано у вигляді

$$A = \begin{pmatrix} 7 & 0 \\ 0 & 19 \end{pmatrix} \begin{pmatrix} 7 & 11 \\ 4 & 15 \end{pmatrix}^{-1} = \begin{pmatrix} 7 & 0 \\ 8 & 19 \end{pmatrix} \begin{pmatrix} 19 & 19 \\ 14 & 21 \end{pmatrix} = \begin{pmatrix} 3 & 3 \\ 2 & 5 \end{pmatrix}.$$

Обернена матриця A^{-1} одразу ж обчислюється з матриці A. Після цього який завгодно криптотекст може бути дешифровано за допомогою M^{-1}.

Важливим моментом у цих обчислюваннях є існування оберненої матриці до $\begin{pmatrix} 7 & 11 \\ 4 & 15 \end{pmatrix}$. З іншого боку, наш криптоаналітик обрав вихідне повідомлення HELP, котре породжує матрицю $\begin{pmatrix} 7 & 11 \\ 4 & 15 \end{pmatrix}$, тобто він здійснює вибір у такий спосіб, аби результуюча матриця мала обернену.

Припустімо тепер, що криптоаналітик працює з іншою початковою постановою – "відоме є певне вихідне повідомлення". Більш докладно: нехай криптоаналітикові відомо, що CKVOZI – криптотекст, який відповідає вихідному повідомленню SAHARA. Хоча ми маємо тут приклад довшого повідомлення, аніж раніш, однак добуваної з нього інформації є набагато менше.

Насправді, тепер рівняння для повідомлення криптотексту мають вигляд

$$A \begin{pmatrix} 18 \\ 0 \end{pmatrix} = \begin{pmatrix} 2 \\ 10 \end{pmatrix}, \quad A \begin{pmatrix} 7 \\ 0 \end{pmatrix} = \begin{pmatrix} 21 \\ 14 \end{pmatrix} \quad \text{та} \quad A \begin{pmatrix} 17 \\ 0 \end{pmatrix} = \begin{pmatrix} 25 \\ 8 \end{pmatrix}.$$

Не існує оберненої квадратної матриці, котру може бути утворено з трьох векторів-стовпців, які з’являються як коефіцієнти M.

Криптоаналітик виявляє, що кожна обернена квадратна матриця

$$A' = \begin{pmatrix} 3 & x \\ 2 & y \end{pmatrix}$$

може бути базисом криптосистеми, оскільки вона шифрує SAHARA як CKVOZI. Отже, криптоаналітик може зупинитися на матриці

$$A' = \begin{pmatrix} 3 & 1 \\ 2 & 1 \end{pmatrix},$$

для якої оберненою є матриця

$$(A')^{-1} = \begin{pmatrix} 1 & 25 \\ 24 & 3 \end{pmatrix}.$$

Криптоаналітик є готовий до перехоплення криптотексту. Він дістає текст NAFG й одразу обчислює

$$\begin{pmatrix} 1 & 25 \\ 24 & 3 \end{pmatrix} \begin{pmatrix} 13 \\ 0 \end{pmatrix} = \begin{pmatrix} 13 \\ 0 \end{pmatrix} \quad \text{та} \quad \begin{pmatrix} 1 & 25 \\ 24 & 3 \end{pmatrix} \begin{pmatrix} 5 \\ 6 \end{pmatrix} = \begin{pmatrix} 25 \\ 8 \end{pmatrix}.$$

Два вектори-стовпці породжують вихідне повідомлення NAZI. Однак легальний користувач знає оригінальну матрицю A та її обернену матрицю й обчислює

$$\begin{pmatrix} 15 & 17 \\ 20 & 9 \end{pmatrix} \begin{pmatrix} 13 \\ 0 \end{pmatrix} = \begin{pmatrix} 13 \\ 0 \end{pmatrix} \quad \text{та} \quad \begin{pmatrix} 15 & 17 \\ 20 & 9 \end{pmatrix} \begin{pmatrix} 5 \\ 6 \end{pmatrix} = \begin{pmatrix} 21 \\ 24 \end{pmatrix},$$

що дає вихідне повідомлення NAVY.

Криптоаналітик припустився прикрої помилки, котра могла спричинитися до хибних кроків.

Алгоритм шифрування в криптосистемі Хілла

Дано квадратну матрицю вигляду $\begin{pmatrix} a_{11} & a_{21} \\ a_{12} & a_{22} \end{pmatrix}$

$$A = \begin{pmatrix} 3 & 3 \\ 2 & 5 \end{pmatrix}.$$

1. Обчислюється визначник D матриці A:

$$D = a_{11}a_{22} - a_{12}a_{21} = 3 \cdot 5 - 2 \cdot 3 = 15 - 6 = 9.$$

2. Визначається долучена матриця алгебричних доповнень A^*, складена з алгебричних доповнень до елементів матриці A, причому алгебричне доповнення до елемента a_{ij} перебуває на перетинанні j-того рядка й i-того стовпця:

$$A^* = \begin{pmatrix} A_{11} & A_{21} \\ A_{12} & A_{22} \end{pmatrix}.$$

Під алгебричним доповненням A_{ij} елемента a_{ij} розуміють мінор M_{ij}, помножений на $(-1)^{i+j}$:

$$A_{ij} = (-1)^{i+j} M_{ij},$$

Тобто

$$A_{11} = (-1)^{1+1} \cdot M_{11} = (-1)^2 \cdot a_{22} = 1 \cdot 5 = 5;$$
$$A_{12} = (-1)^{2+1} \cdot M_{12} = (-1)^3 \cdot a_{12} = -1 \cdot 2 = -2;$$
$$A_{21} = (-1)^{1+2} \cdot M_{21} = (-1)^3 \cdot a_{21} = -1 \cdot 3 = -3;$$
$$A_{22} = (-1)^{2+2} \cdot M_{22} = (-1)^4 \cdot a_{11} = 1 \cdot 3 = 3.$$

Отже, можна записати здобуту матрицю алгебричних доповнень:

$$A^* = \begin{pmatrix} 5 & -3 \\ -2 & 3 \end{pmatrix}.$$

3. Визначається обернена матриця A^{-1}.

За обернену матрицю для A слугуватиме матриця, котра виходить із долученої матриці A^* діленням усіх її елементів на D:

$$A^{-1} = \begin{pmatrix} \frac{A_{11}}{D} & \frac{A_{21}}{D} \\ \frac{A_{12}}{D} & \frac{A_{22}}{D} \end{pmatrix} = \begin{pmatrix} \frac{5}{9} & \frac{-3}{9} \\ \frac{-2}{9} & \frac{3}{9} \end{pmatrix}.$$

Перевіримо, чи виконується умова $A \cdot A^{-1} = E$, де E – квадратна одинична матриця.

За правилом перемножування матриць, елемент, який розміщено в i-тому рядку й j-тому стовпці матриці добутку, дорівнює сумі добутків відповідних елементів i-того рядка матриці A та j-того стовпця матриці A^{-1}.

$$A \cdot A^{-1} = \begin{pmatrix} 3 & 3 \\ 2 & 5 \end{pmatrix} \cdot \begin{pmatrix} \frac{5}{9} & \frac{-3}{9} \\ \frac{-2}{9} & \frac{3}{9} \end{pmatrix} = \begin{pmatrix} 3\frac{5}{9} - 3\frac{2}{9} & 3\left(\frac{-3}{9}\right) + 3\frac{3}{9} \\ 2\frac{5}{9} - 5\frac{2}{9} & 2\left(\frac{-3}{9}\right) + 5\frac{3}{9} \end{pmatrix} = \begin{pmatrix} \frac{15-6}{9} & \frac{9-9}{9} \\ \frac{10-10}{9} & \frac{15-6}{9} \end{pmatrix} =$$

$$= \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = E.$$

Отже, можна дійти висновку, що обернену матрицю віднайдено правильно.

4. Зведення оберненої матриці за модулем 26:

$$A^{-1} \bmod 26 = \begin{pmatrix} \frac{5}{9} & \frac{-3}{9} \\ \frac{-2}{9} & \frac{3}{9} \end{pmatrix} \bmod 26 = \begin{pmatrix} \frac{135}{9} & \frac{153}{9} \\ \frac{180}{9} & \frac{81}{9} \end{pmatrix} \bmod 26 = \begin{pmatrix} 15 & 17 \\ 20 & 9 \end{pmatrix}.$$

Нотатка. Число 135 – це результат пошуку числа, яке дає залишок 5 за модулем 26. При цьому виконується додаткова умова – віднайдене число повинно ділитися на 9 без залишку. Аналогічно здобуваються числа 153; 180; 81.

5. Зашифровування відкритого повідомлення.

Зашифровування здійснюється за допомогою рівняння $A \cdot P = C$, де A – матриця перетворення; P – вектор, компонентами якого є номери літер відкритого повідомлення відповідно до табл. 3.1; C – вектор, компонентами якого є номери літер закритого повідомлення відповідно до табл. 3.1.

Зашифруємо слово HELP.

Спочатку розіб'ємо n-граму відкритого тексту на біграми. Потім у кожній біграмі відкритого тексту замінимо кожну літеру на її числовий еквівалент відповідно до табл. 3.1.

Вихідне повідомлення HELP визначає два вектори:

$$P_1 = \begin{pmatrix} H \\ E \end{pmatrix} = \begin{pmatrix} 7 \\ 4 \end{pmatrix} \quad \text{та} \quad P_2 = \begin{pmatrix} L \\ P \end{pmatrix} = \begin{pmatrix} 11 \\ 15 \end{pmatrix};$$

Зашифровування має вигляд:

$$C_1 = A \cdot P_1 = \begin{pmatrix} 3 & 3 \\ 2 & 5 \end{pmatrix} \cdot \begin{pmatrix} 7 \\ 4 \end{pmatrix} = \begin{pmatrix} 3 \cdot 7 + 3 \cdot 4 \\ 2 \cdot 7 + 5 \cdot 4 \end{pmatrix} \bmod 26 = \begin{pmatrix} 33 \\ 34 \end{pmatrix} \bmod 26 = \begin{pmatrix} 7 \\ 8 \end{pmatrix};$$

$$C_2 = A \cdot P_2 = \begin{pmatrix} 3 & 3 \\ 2 & 5 \end{pmatrix} \cdot \begin{pmatrix} 11 \\ 15 \end{pmatrix} = \begin{pmatrix} 3 \cdot 11 + 3 \cdot 15 \\ 2 \cdot 11 + 5 \cdot 15 \end{pmatrix} \bmod 26 = \begin{pmatrix} 78 \\ 97 \end{pmatrix} \bmod 26 = \begin{pmatrix} 0 \\ 19 \end{pmatrix}.$$

Отже, здобули послідовність чисел 7; 8; 0; 19.

З таблиці 3.1 дістаємо криптотекст HIAT.

Процес розшифровування йде за оберненим алгоритмом:

$$P_1 = A^{-1} \cdot C_1 = \begin{pmatrix} 15 & 17 \\ 20 & 9 \end{pmatrix} \cdot \begin{pmatrix} 7 \\ 8 \end{pmatrix} = \begin{pmatrix} 15 \cdot 7 + 17 \cdot 8 \\ 20 \cdot 7 + 9 \cdot 8 \end{pmatrix} \bmod 26 = \begin{pmatrix} 241 \\ 212 \end{pmatrix} \bmod 26 = \begin{pmatrix} 7 \\ 4 \end{pmatrix};$$

$$P_2 = A^{-1} \cdot C_2 = \begin{pmatrix} 15 & 17 \\ 20 & 9 \end{pmatrix} \cdot \begin{pmatrix} 0 \\ 19 \end{pmatrix} = \begin{pmatrix} 15 \cdot 0 + 17 \cdot 19 \\ 20 \cdot 0 + 9 \cdot 19 \end{pmatrix} \bmod 26 = \begin{pmatrix} 323 \\ 171 \end{pmatrix} \bmod 26 = \begin{pmatrix} 11 \\ 15 \end{pmatrix}.$$

Здобута послідовність 7; 4; 11; 15 відповідно до таблиці 3.1 надає можливість відновити вихідний текст: HELP.

3.2.1 Завдання для самоперевірки з теми

Завдання 3.2.1. Зашифрувати 4 літери свого Прізвища, використовуючи криптосистему Хілла, якщо $A = \begin{pmatrix} 3 & 3 \\ 2 & 5 \end{pmatrix}$.

Завдання 3.2.2 Розшифрувати криптограму, яка була отримана в завданні 3.2.1

Контрольні питання з теми

1 Від чого залежить модуль, за яким виконуються обчислення у криптосистемі Хілла?

2 З якою метою перевіряють виконання умови одиничної квадратної матриці у криптосистемі Хілла?

3.3 Сучасні симетричні криптосистеми

К. Шеннон в своїх роботах з теорії зв′язку дійшов висновків, що в практичних шифрах слід використовувати дві засади: розсіювання та перемішування.

Розсіювання являє собою розповсюдження впливу одного знаку відкритого тексту на безліч знаків шифртексту, що дозволяє приховувати статистичні властивості вихідного тексту.

Перемішування припускає використовування таких перетворювань, які ускладнюють відновлення взаємозв′язку статистичних властивостей вихідного та шифрованого текстів.

Поширеним способом досягання ефектів розсіювання та перемішування є використовування складного шифру, тобто такого шифру, який може бути реалізовано у вигляді певної послідовності простих шифрів, причому кожний з них має вкласти власний внесок в сумарне розсіювання та перемішування.

У складних шифрах, які будуються на підставі простих шифрів, в основі криптографічних алгоритмів лежать математичні перетворювання, котрі дозволяють вимагати високої практичної стійкості. В криптографії існують лише два основних типи криптографічних перетворювань – заміна й перестановка символів. Всі інші алгоритми є лише комбінацією цих двох типів.

За багаторазового переміжненю простих перестановок та підстановок, за допомогою доволі довгого секретного ключа, можна здійснити стійке шифрування з добрим розсіюванням та перемішуванням. Сучасні криптографічні системи, які наведено нижче, побудовано згідно із зазначеною методологією.

Сучасні криптографічні системи може бути реалізовано програмним, програмно-апаратним чи апаратним шляхом.

Сьогодні в основі всіх технологій із захисту даних від несанкціонованого доступу або електронної комерції лежить наука криптографія, яка є частиною науки криптології. У свою чергу, ця наука використовує різні методи і алгоритми криптографічних перетворювань.

Для того, щоб інформація, пройшовши шифрування, перетворилася в безглуздий набір символів, використовуються спеціально розроблені алгоритми шифрування, які розробляються вченими математиками.

Алгоритми шифрування діляться на два великі класи: симетричні та асиметричні. Симетричні алгоритми шифрування використовують один і той же секретний ключ для зашифровування інформації і для її розшифровування, проте асиметричні алгоритми використовують два ключа - один для зашифровування (відкритий ключ), інший для розшифрування (секретний ключ).

3.4 Алгоритм шифрування DES

Алгоритм шифрування даних DES (Data Encryption Standard) було опубліковано 1977 року Національним бюро стандартів США. Типовий блочний шифр, тодішній стандарт DES було призначено для захисту від несанкціонованого доступу до важливої, але несекретної інформації в державних та комерційних організаціях США. Алгоритм, покладений у підґрунтя стандарту, 1980 року було схвалено Національним інститутом стандартів та технологій США (НІСТ). Алгоритм DES найчастіше застосовується в системах захисту комерційної інформації. На сьогодні розроблено програмне забезпечення та спеціалізовані ЕОМ, які реалізують шифрування інформації в мережах передавання даних.

Суть алгоритму зводиться до наступного. Передана до каналу зв'язку послідовність розбивається на блоки довжиною в 64 біти. Кожний з цих блоків шифрується незалежно від інших за допомогою 64-бітового ключа, в якому значущими є лише 56 біт (інші 8 біт – перевірні біти для контролю на парність).

Система DES використовує операції заміни перестановок символів та додавання за модулем 2.

Зручністю застосовуваного алгоритму є те, що операції зашифровування й розшифровування в DES є оберненими. Узагальнена схема процесу зашифровування в алгоритмі DES має вигляд рис.3.1.

З файлу вихідного тексту зчитується черговий 64-бітовий блок Т. Процес його зашифровування полягає в початковій перестановці, шістнадцятьох циклах шифрування і, врешті, у завершальній перестановці бітів.

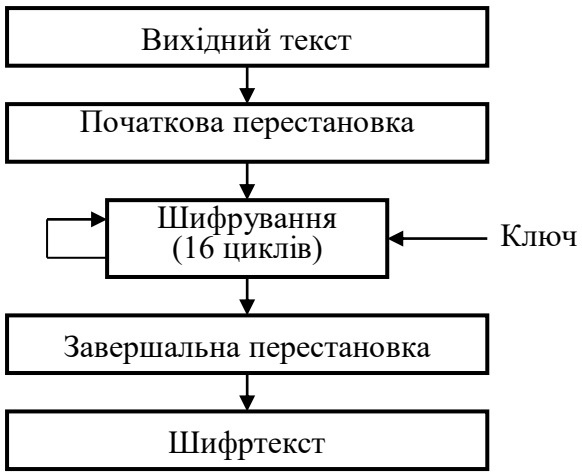


Рисунок 3.1 – Узагальнена схема шифрування в алгоритмі DES

Вхідна послідовність $M = m_1, m_2, ..., m_{64}$ перетворюється блоком початкових перестановок на послідовність

$$IP(M) = m_{58}, m_{50}, ..., m_7,$$

де m_i може набирати значень 0 чи 1.

Перетворювання виконуються за фіксованою таблицею 3.2.

Таблиця 3.2 – Початкові перестановки

58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

За допомогою матриці початкової перестановки IP, яка містить 8 стовпців і 8 рядків, здійснюється перестановка символів вхідного блока.

Символи блока вписують до таблиці по стовпцях, розпочинаючи з лівої нижньої клітинки. Потім рядки змінюють місцями в порядку 2, 4, 6, 8, 1, 3, 5, 7. Далі символи, розташовані у клітинках таблиці, зчитуються по рядках.

Завершальне перетворювання алгоритму – обернена перестановка – здійснюється за допомогою матриці оберненої перестановки IP^{-1}.

Символи блока вписують до таблиці по стовпцях, розпочинаючи з лівої нижньої клітинки. Потім стовпці змінюють місцями в порядку 5, 1, 6, 2, 7, 3, 8, 4. Далі символи, розташовані в клітинках таблиці 3.3, зчитуються по рядках.

Таблиця 3.3 – Завершальні перестановки

40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

Алгоритм побудовано на базі мережі Файстеля (рис. 2.12).

Процес шифрування здійснюється в такий спосіб. Здобута після першої перестановки 64-бітова послідовність M розбивається навпіл на два 32-розрядних блоки L_0 та R_0 . Потім виконується ітеративний процес шифрування, котрий складається з 16-ти кроків.

Нехай M_i — результат i -тої ітерації:

$$M_i = L_i R_i,$$

де $L_i = m_1, m_2, \ldots, m_{32}$; $R_i = m_{33}, m_{34}, \ldots, m_{64}$... У цьому разі результат ітерації описується формулами:

$$L_i = R_{i-1}, \; i = 1, 2, \ldots, 16;$$

$$R_i = L_{i-1} \oplus f(R_{i-1}, K_i), \; i = 1, 2, \ldots, 16.$$

Функція f називається функцією шифрування. Її аргументами є послідовність R_{i-1}, здобута на попередньому кроці шифрування, та 48-бітовий ключ шифрування K, що формується за певним правилом з 64-бітового ключа шифрування K.

Усі перетворювання, виконувані в межах алгоритму DES, ілюструє схема подана на рис. 3.2.

Значення L_0 та R_0 здобувають звичайною розбивкою блока M, який було піддано початковій перестановці, навпіл. Далі блок L_1 визначають, за правилом $L_1 = R_0$.

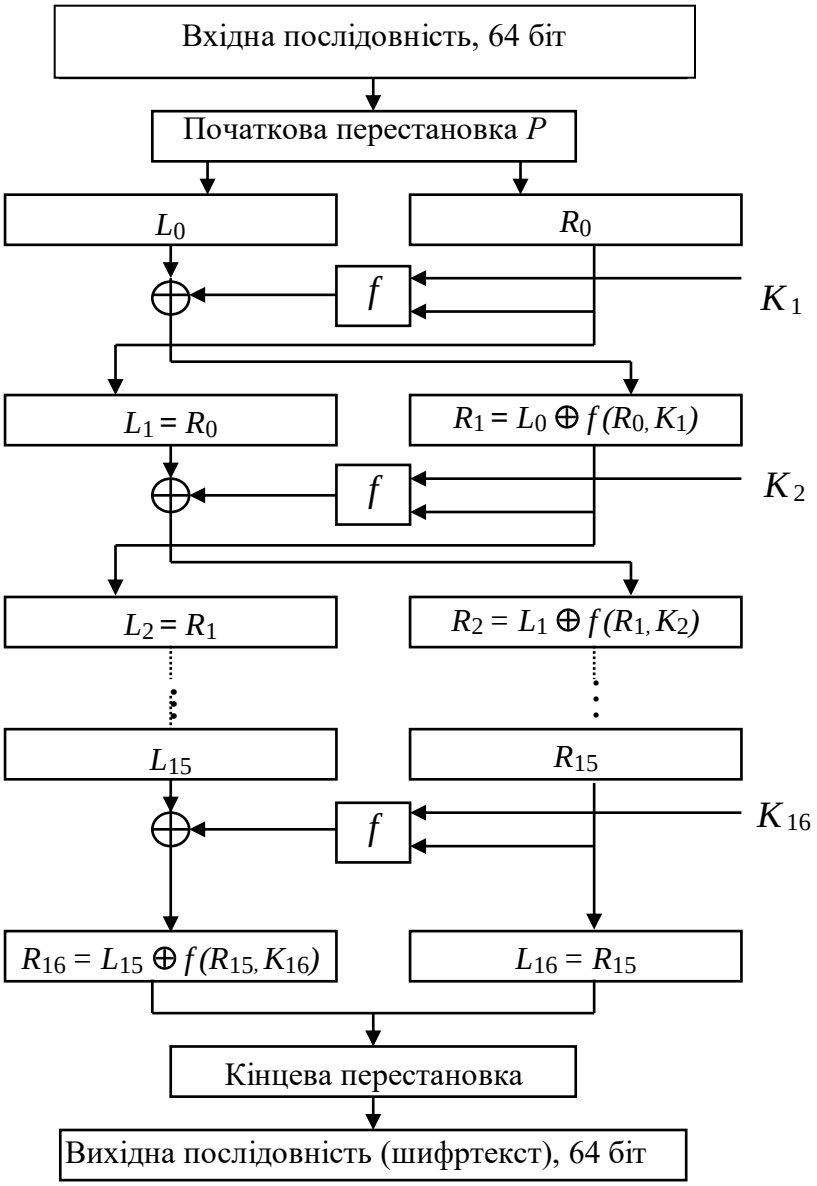


Рисунок 3.2 – Схема алгоритму DES

Задля одержання значення R_1, треба обчислити значення функції $f(R_0, K_1)$. Ця операція здійснюється у відповідності з алгоритмом, поданим на рис. 3.3. Для можливості здійснення операції додавання за модулем 2, 32-розрядний підблок R_i розширюється до 48 бітів. Далі результат додавання розподіляється на вісім блоків заміни S.

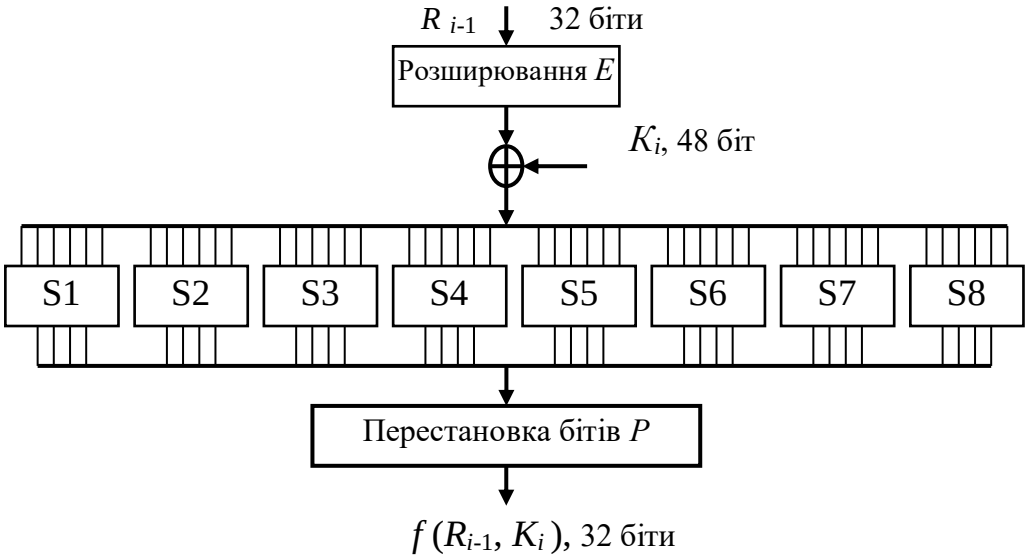


Рисунок 3.3 – Схема обчислювання функції шифрування $f(R_{i-1}, K_i)$.

На початку 32-бітовий блок розширюється до 48-ми символів (рис.3.4).

При цьому додаткових 16 символів беруться, за певним правилом, з розширюваного блока згідно з табл. 3.4.

Таблиця 3.4 – Функція розширювання E

32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

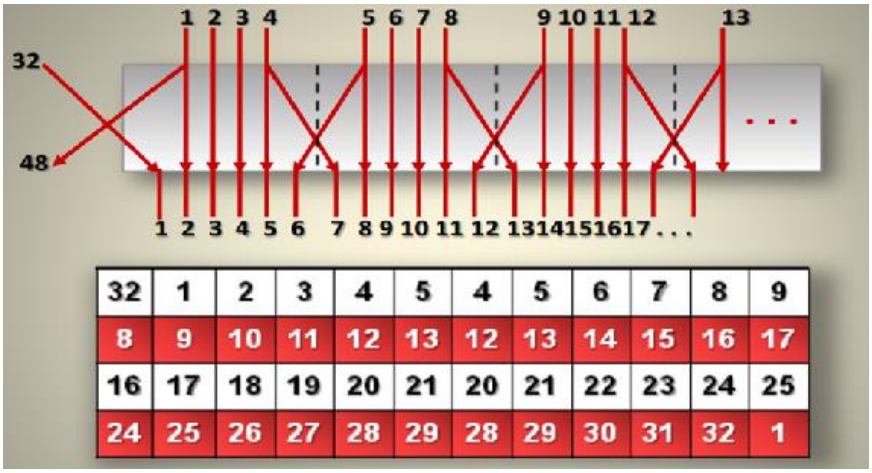


Рисунок 3.4 – Реалізація функції розширювання E

Потім до здобутого 48-розрядного блока додають за модулем два символи першого 48-розрядного ключа K_1. На наступному етапі здобута 48-розрядна послідовність розбивається на 8 груп по 6 символів. Кожна 6-символьна група вихідної послідовності B_j перетвориться на 4-символьну групу відповідно до шифрувальної таблиці 3.5. Після цього здобуті в такий спосіб вісім 4-символьних груп складатимуть 32-бітовий блок, який піддають черговій табличній перестановці (табл. 3.6).

Кожна з функцій S_j (B_j) перетворює 6-бітову послідовність на 4-бітову за таким алгоритмом:

– перший та останній біти вихідної послідовності B_j ($B_j = b_1\, b_2\, b_3\, b_4\, b_5\, b_6$, де $b_j = 0$ чи 1) визначають номер рядка k. Тобто число $b_1 b_6$, яке переведено з двійкової системи обчислювання до десяткової, – це номер рядка в таблиці 3.5;

– другий, третій, четвертий та п’ятий біти послідовності B_j визначають номер стовпчика l. Тобто число $b_2\, b_3\, b_4\, b_5$, яке переведено з двійкової системи обчислювання до десяткової, – це номер стовпця в таблиці 3.5;

– результат перетворювання обирається на перетинанні рядка k та стовпця l.

Приклад 3.1. Нехай $B = 011011$. Тоді $S(1)(B) = 0101$.

Дійсно,

$$k = 1\ (b_1\, b_6 = 01_B = 1_D);$$

$$l = 13\ (b_2\, b_3\, b_4\, b_5 = 1101_B = 13_D).$$

На перетинанні стовпця 13 та рядка 1 задано число $5_D = 0101_B$. Отже, послідовність 011011 перетворена на послідовність 0101.

Функція перестановки бітів P задається таблицею 3.6.

Таблиця 3.5 – Функційні перетворювання S_j

Рядки	Стовпці															
	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
								S_1								
0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
2	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13
								S_2								
0	15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
1	3	13	4	7	15	2	8	14	12	0	1	10	6	9	4	5
2	0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
3	13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9
								S_3								
0	10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
1	13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
2	13	6	4	9	8	15	3	0	11	1	2	12	15	10	14	7
3	1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12
								S_4								
0	7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
1	13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
2	10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
3	3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14
								S_5								
0	2	12	14	1	7	10	11	6	8	5	3	15	13	0	14	9
1	14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6
2	4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
3	11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3
								S_6								
0	12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
1	10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
2	9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6
3	4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13
								S_7								
0	4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
1	13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
2	1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
3	6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12
								S_8								
0	13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
1	1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2
2	7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
3	2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11

Таблица 3.6 – Функція перестановки P

16	7	20	21
29	12	28	17
1	15	23	26
5	18	31	10
2	8	24	14
32	27	3	9
19	13	30	6
22	11	4	25

На кожному кроці ітерації для функційного перетворювання використовується нове значення ключа K_i.

Нове значення ключа K_i обчислюється з його початкового значення. Ключ являє собою 64-бітовий блок з вісьмома бітами контролю за парністю, розташованих у позиціях 8, 16, 24, 32, 40, 48, 56, 64.

Задля вилучання контрольних бітів та підготовки ключа до роботи використовується функція первинної підготовки ключа G (табл. 3.7).

Таблиця 3.7 – Функція первинного встановлення ключа

57	49	41	33	25	17	9
1	58	50	42	34	26	18
10	2	59	51	43	35	27
19	11	3	60	52	44	36
63	55	47	39	31	23	15
7	62	54	46	38	30	22
14	6	61	53	45	37	29
21	13	5	28	20	12	4

Результат перетворювання $G(K)$ розбивається на дві частини – C_0 та D_0 по 28

біт кожна. Перші чотири рядки матриці визначають біти послідовності C_0 (символи ключа зчитуються по рядках). Наступні чотири рядки матриці визначають біти послідовності D_0. Після визначення C_0 та D_0 рекурсивно визначаються значення C_i та D_i. З цією метою виконуються операції циклічного зсування ліворуч на кількість кроків, обумовлені таблицею 3.8

Таблиця 3.8 – Циклічні зсування

Номер ітерації	Кількість зрушень ліворуч	Номер ітерації	Кількість зрушень ліворуч
1	1	9	1
2	1	10	2
3	2	11	2
4	2	12	2
5	2	13	2
6	2	14	2
7	2	15	2
8	2	16	1

Операції зсування виконуються незалежно для послідовностей C_i та D_i. Ключ K_i, обумовлений на кожнім кроці ітерації, є результат вибору конкретних 48-ми бітів з 56-бітової послідовності та їхньої перестановки (рис. 3.5).

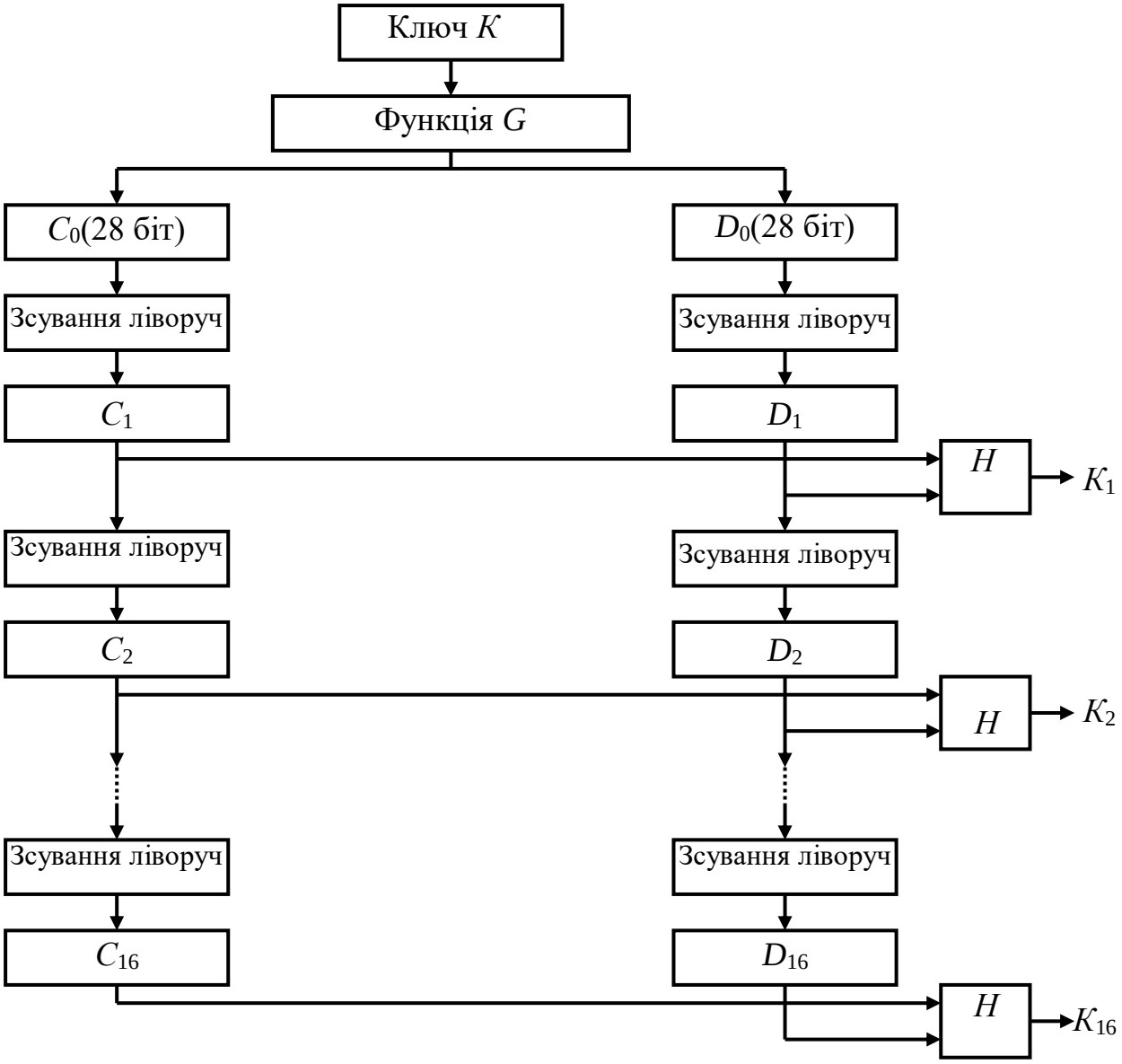


Рисунок 3.5 – Схема алгоритму обчислювання ключів K_i

Порядок вибору визначається функцією H:

$$K_i = H\,(C_i D_i).$$

Функція H визначається матрицею, яка завершує формування ключа за таблицею 3.9.

Символи послідовностей C_i та D_i. вписуються до матриці H відповідно до номерів її клітинок і потім зчитуються по рядках. Отже, виконується перестановка символів.

Усі перестановки й коди в таблицях обрано розроблювачами в такий спосіб, щоби максимально утруднити процес розшифровування шляхом підбирання ключа.

Таблиця 3.9 – Функція *H*

14	17	11	24	1	5
3	28	15	6	21	10
23	19	12	4	26	8
16	7	27	20	13	2
41	52	31	37	47	55
30	40	51	45	33	48
44	49	39	56	34	53
46	42	50	36	29	32

Операції, виконувані в перебігу шифрування з використовуванням алгоритму DES, є обернені. Процедура розшифровування на боці приймача відрізняється лише черговістю формування ключів K_i. Вони повинні подаватися у зворотному порядку – K_{16} ... K_1.

Щоби можна було використовувати алгоритм DES для розв'язування різноманітних криптографічних завдань, розроблено кілька режимів його використовування.

3.4.1 Режим „Електронна кодова книга"

Режим „Електронна кодова книга" ECB (Electronic Code Book) передбачає реалізацію криптографічних перетворювань у відповідності з головним алгоритмом DES. Кожний 64-розрядний блок перетворюють з використовуванням одного й того самого ключа (рис. 3.6).

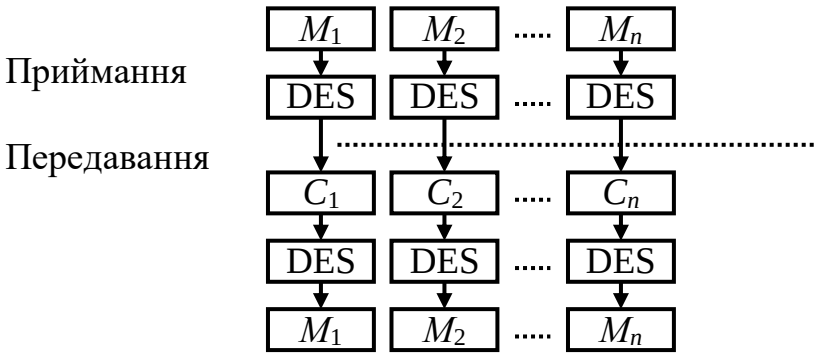


Рисунок 3.6 – Режим електронної кодової книги

Головною перевагою такого алгоритму є простота реалізації. Однак він є найменш тривалий щодо спроб розшифровування. Це пояснюється тим, що за відносно невеликої довжини блока та великої довжини тексту, що передається певні блоки можуть повторюватися. Це надає криптоаналітикові певну інформацію щодо змісту повідомлення.

3.4.2 Режим „Зчеплювання блоків шифру”

Режим „Зчеплювання блоків шифру” CBC (Cipherblock Chaining)передбачає розбиття повідомлення, що передається, на блоки

$$M = M_1 M_2 M_3$$

До першого блока додається за модулем два первинний вектор S, який регулярно змінюється на приймальному й передавальному боці. Після цього здобута сума шифрується з використовуванням алгоритму DES. Отже, в канал передається 64-бітове повідомлення C_1.

Здобута на боці приймача сума C_1 розшифровується, потім від неї віднімається первинний вектор S.

На наступному етапі шифрування, при передаванні наступного блока, замість первинного вектора S до зашифрованого повідомлення додається передаване повідомлення C_1. Ця процедура повторюється за передавання кожного наступного 64-розрядного блока (рис. 3.7).

Отже, останній блок шифртексту буде функцією секретного ключа, первинного вектора і кожного біта відкритого тексту, незалежно від його довжини. Цей блок називають кодом автентифікації повідомлення (КАП).

КАП можна легко здобути на боці приймача шляхом повторювання процедур, які виконуються на боці передавача.

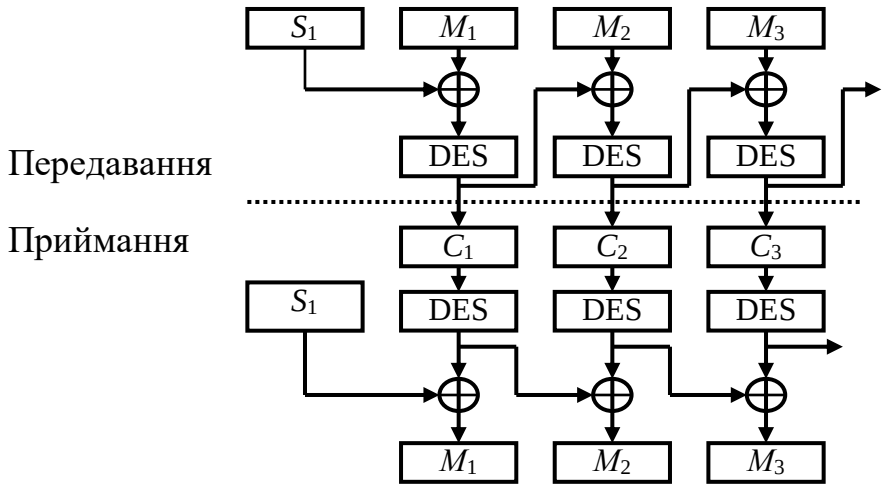
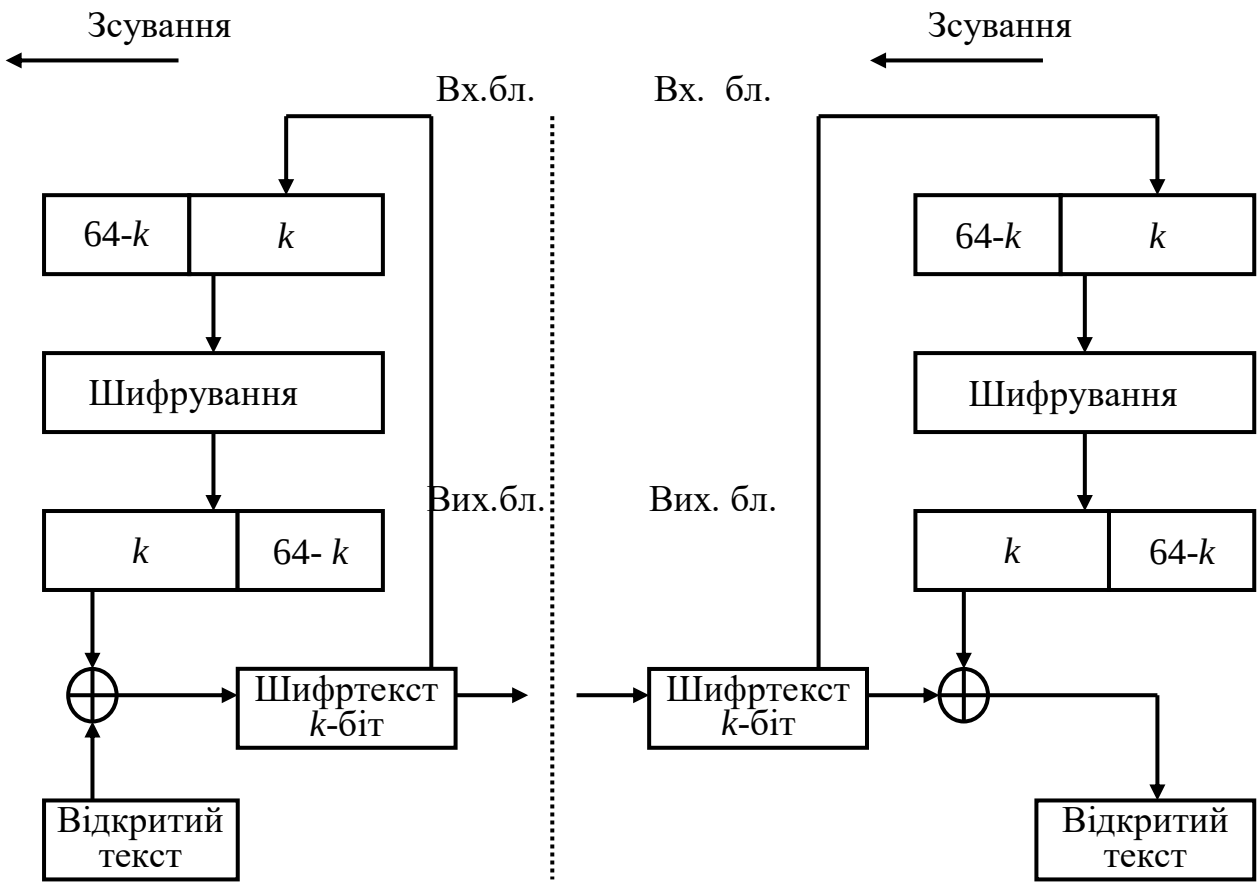


Рисунок. 3.7 – Режим "Зчеплювання блоків шифру"

3.4.3 Режим „Зворотний зв'язок за шифром"

Режим „Зворотний зв'язок за шифром" CFB (Cipher Feedback) припускає, щоби довжина блока відрізнялась від 64 біт. Припустімо, що довжина блоків, на які розбивається зашифровувана послідовність, становить менше за 64 біти (рис. 3.8).



Зашифровування Розшифровування

Рисунок 3.8 – Схема алгоритму в режимі зворотного зв'язку за шифром

До вхідного блока вписується вектор ініціалізації. Потім його вміст шифрується за допомогою алгоритму DES.

Припустімо, що внаслідок розбивання на блоки здобуто n блоків довжиною k біт кожний (залишок дописується нулями).Тоді для кожного $i = 1\ldots n$ блок шифр тексту є

$$C_i = M_i \oplus P_{i-1},$$

де P_{i-1} – k старших бітів попереднього зашифровуваного блока.

Оновлювання регістру здійснюється на ланцюзі зворотного зв'язку шляхом вилучання його k старших бітів та запису C_i до регістру.

Відновлювання прийнятого повідомлення здійснюється виконанням перелічених операцій у зворотному порядку:

$$M_i = C_i \oplus P_{i-1}.$$

3.4.4 Режим „Зворотний зв'язок за виходом"

Режим „Зворотний зв'язок за виходом" OFB (Output Feedback), аналогічно до розглянутого CFB, припускає, щоби довжина блока відрізнювалась від 64 біт. Різниця даного методу полягає в організації зворотного зв'язку. Оновлювання вмісту вхідного блока здійснюється не змістом шифртексту, передаваного до каналу зв'язку, а змістом зашифрованого вектора ініціалізіції на першому і наступних кроках шифрування блоків.

Нехай

$$M = M_1 M_2, \ldots, M_n.$$

Для усіх $i = 1, \ldots, n$

$$C_i = M_i \oplus P_i,$$

де P_i – старші k бітів операції DES C_{i-1}.

Алгоритм шифрування ілюструє схема на рис. 3.9.

Слабкість алгоритму DES пов'язується з невеликою довжиною ключа. У 1993 року Вінер продемонстрував, що за 1 000 000 $ можна створити машину для пошуку ключів алгоритму DES. За відомих вихідного та зашифровуваного текстів ця машина зможе віднайти ключ за 3,5 години. Окрім того, 15 липня 1998 року коаліція компаній Cryptography Research, Advanced Wireless Technology та Electronic Frontier Foundation оголосила про атаку, яку успішно було здійснено на алгоритм DES. Ці компанії створили машину, котра здійснювала пошук ключів, під назвою DES Cracker (відомою також під назвою Deep Cracker), яка коштувала 250 000 $. За допомогою DES Cracker віднайшли ключ алгоритму DES Challenge за 56 годин. Це було доказом того, що з урахуванням рівня обчислювальних технологій кінця 1990–х років, ключ, який має довжину 56 біт, є надто малим задля того, щоби забезпечувати стійкість шифру з секретним ключем.

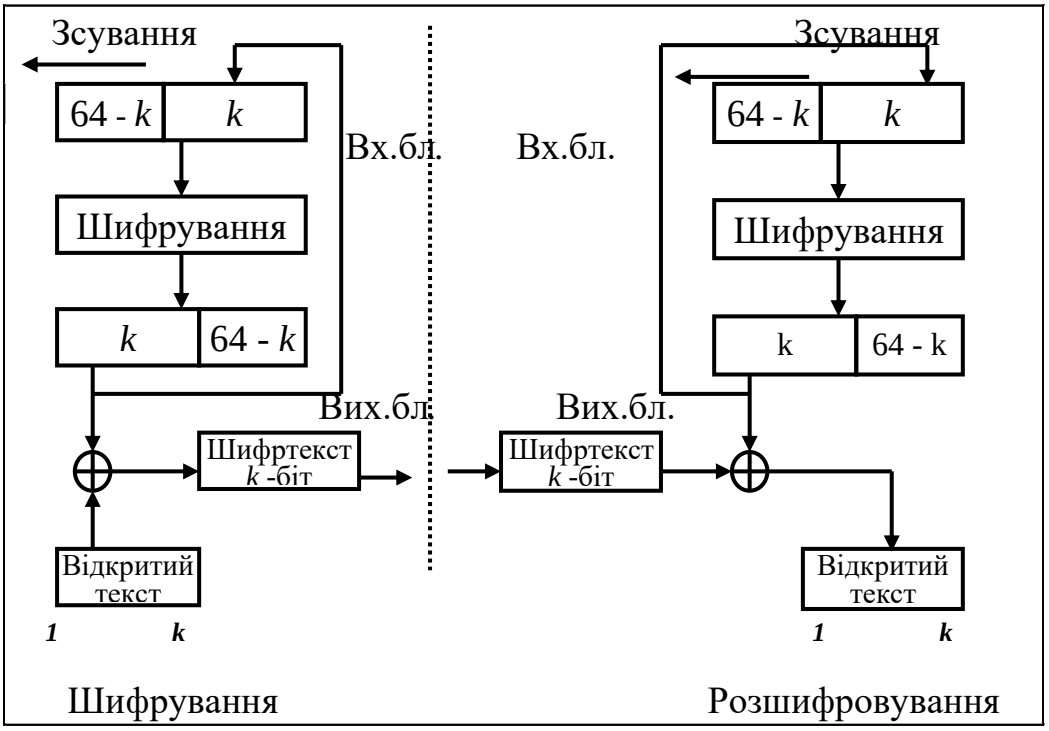


Рисунок 3.9 – Схема алгоритму у режимі зворотного зв’язку за виходом

3.4.5 Завдання для самоперевірки з теми

Завдання 3.4.1. Вхідна послідовність *S* блоку B = 100111. Використовуючи рис.3.3 та табл. 3.5 визначити, яка послідовність буде на виході блоку S_3.

Завдання 3.4.2 Вхідна послідовність S_8 блоку B = 000000. Використовуючи рис.3.3 та табл. 3.5 визначити, яка послідовність буде на виході блоку S_8.

Контрольні питання з теми

1. Де застосовується алгоритм DES?
2. Який розмір блоку даних, загальний розмір ключа та розмір ключів раунда в алгоритмі DES?
3. Яка кількість раундів в DES?
4. Які математичні операції використовуються в алгоритмі DES?
5. Чи є операції зашифрування та розшифрування оберненими в алгоритмі DES?
6. Як здійснюється процес шифрування в алгоритмі DES?
7. Опишіть схему обчислення функції шифрування f в алгоритмі DES?
8. Чим відрізняється процедура розшифрування на боці приймача від шифрування в алгоритмі DES?
9. Які існують режими роботи алгоритму DES?
10. Які слабкі місця алгоритму DES?

3.5 Алгоритм шифрування TDEA (3-DES) .

Ще донедавна криптосистему DES використовували для шифрування телефонного зв'язку та при супутникових комунікаціях, в електронних банківських системах та в паролях комп'ютерних систем. Проте довжина ключа в 64(56) біт незабаром стала очевидно замалою і для збільшення стійкості фахівці рекомендували, виконувати шифрування два або три рази з різними ключами.

На сьогодні алгоритм DES має недоліки та вважається нестійким, оскільки:

- по-перше, розмір ключа в 64 (56) бітів — замалий, тому існує реальна загроза дешифрування методом пошуку ключа послідовним перебором (лобовою атакою);
- по-друге, алгоритм DES нестійкий до лінійної атаки, що дозволяє зловмиснику знайти ключ DES швидше, ніж методом послідовного перебору (лінійний криптоаналіз).

В той же час, алгоритм DES має переваги, а саме, при шифруванні 64-розрядного блоку вихідного тексту впродовж 16 раундів, він і досі стійкий до статистичної атаки (диференціального криптоаналізу).

Через високу розповсюдженість використання алгоритму DES було запропоновано багато ідей щодо підвищення його безпеки. Одним із рішень було використання особливого режиму шифрування звичайного алгоритму DES.

TDEA (Triple DES чи 3-DES) – це блочний симетричний шифр, який тричі застосовує алгоритм шифрування DES до кожного блоку даних, що надає більш убезпечене шифрування.

На рисунку 3.10 показано схему алгоритму шифрування 3-DES.

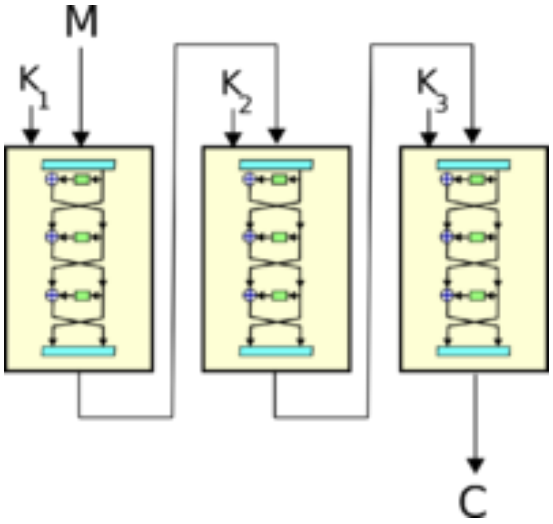
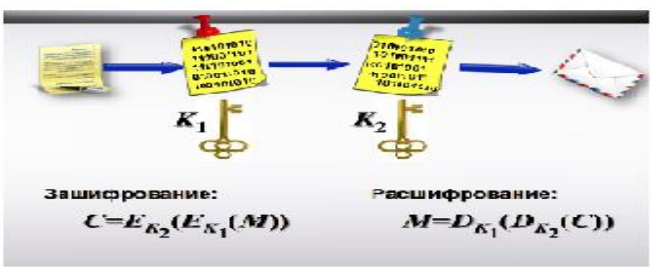
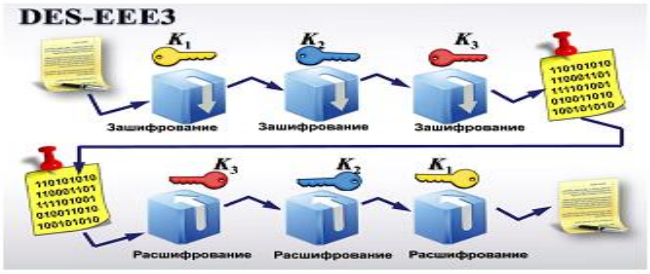


Рисунок 3.10 – Схема потрійного шифрування блоку вихідного тесту 3-DES.





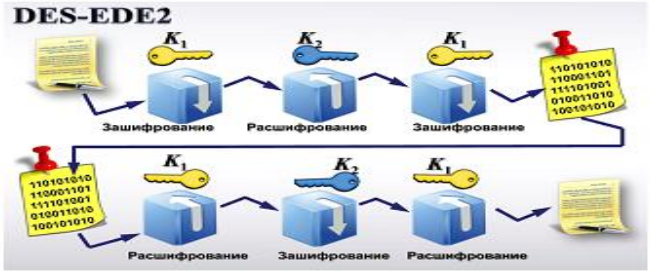


Рисунок 3.11 – Варіанти модифікації алгоритму шифрування 3-DES.

Взагалі алгоритм має 3 модифікації:

1. DES - EEE3, тобто шифрування здійснюється 3 рази з різними ключами;

Зашифровування виконується у такий спосіб:

$$C= E_{k3}(E_{k2}(E_{k1}(M))), \qquad (3.1)$$

де C – шифртекст, E – операція зашифровування DES
$k1$, $k2$, $k3$ – 56-бітові ключі DES,
отже довжина спільного секретного ключа дорівнює 168 біт.

Розшифрування полягає у виконанні зворотних дій:

$$M= D_{k1}(D_{k2}(D_{k3}(C))), \qquad (3.2)$$

2. DES - EDE3, тобто зашифровування виконується в такий спосіб :

$$C= E_{k3}(D_{k2}(E_{k1}(M))), \qquad (3.3)$$

де C – шифртекст, E – операція зашифровування, D – операція розшифрування.
$k1$, $k2$, $k3$ – 56-бітові ключі DES, тобто довжина спільного секретного ключа дорівнює 168 біт.

Розшифрування полягає у виконанні зворотних дій:

$$M= D_{k1}(E_{k2}(D_{k3}(C))), \qquad (3.4)$$

3. DES - EEE3 та DES – EDE2, тобто шифрування виконується аналогічно 1 та 2 способу з використанням однакового ключа на першому та третьому кроках, отже довжина спільного секретного ключа дорівнює 112 біт.

Найчастіше використовувана реалізація 3-DES – це DES - EDE3.

Якщо $k_1=k_2=k_3$, тоді потрійний DES сумісний зі звичайним алгоритмом DES.

Шифрування більш ніж одного блока відбувається за допомогою різноманітних режимів, які можна означити незалежно від конкретного блочного шифру.

Хоча довжина ключа потрійного DES втричі більша за звичайний DES, існує теоретична атака на схему потрійного шифрування, що вимагає лише 2^{2n} операцій (де n – довжина ключа в бітах, для DES $n = 56$), тобто стійкість потрійного DES вдвічі більша за стійкість звичайного. Однак, така атака додатково вимагає 2^n блоків пам’яті, що робить її надзвичайно важкою в практичній реалізації.

Стійкість 2^{112} біт робить потрійний DES досить надійним. Однак, низька швидкість та наслідування усіх інших недоліків (наприклад, незручності для програмної реалізації) зумовлюють неконкурентоспроможність потрійного DES порівняно з алгоритмом AES.

3.5.1 Завдання для самоперевірки з теми

Контрольні питання з теми

1. Який головний недолік криптосистеми DES?
2. Як вирішується проблема підвищення криптостійкості DES в алгоритмі шифрування TDEA?
3. Що таке TDEA?
4. Які модифікації має алгоритм TDEA**?**
5. Яка стійкість потрійного DES?
6. Недоліки використання потрійного DES?

3.6 Стандарт шифрування AES.

У 1998 р., Національний інститут стандартів і технологій (NIST) опублікував запит на заміну стандарту шифрування, в якому описувався передбачуваний “Вдосконалений стандарт шифрування” (Advanced Encryption Standard – AES), що повинен прийти на зміну алгоритму DES. Захист інформації, який повинен забезпечити стандарт був найвищим пріоритетом для створення алгоритму AES.

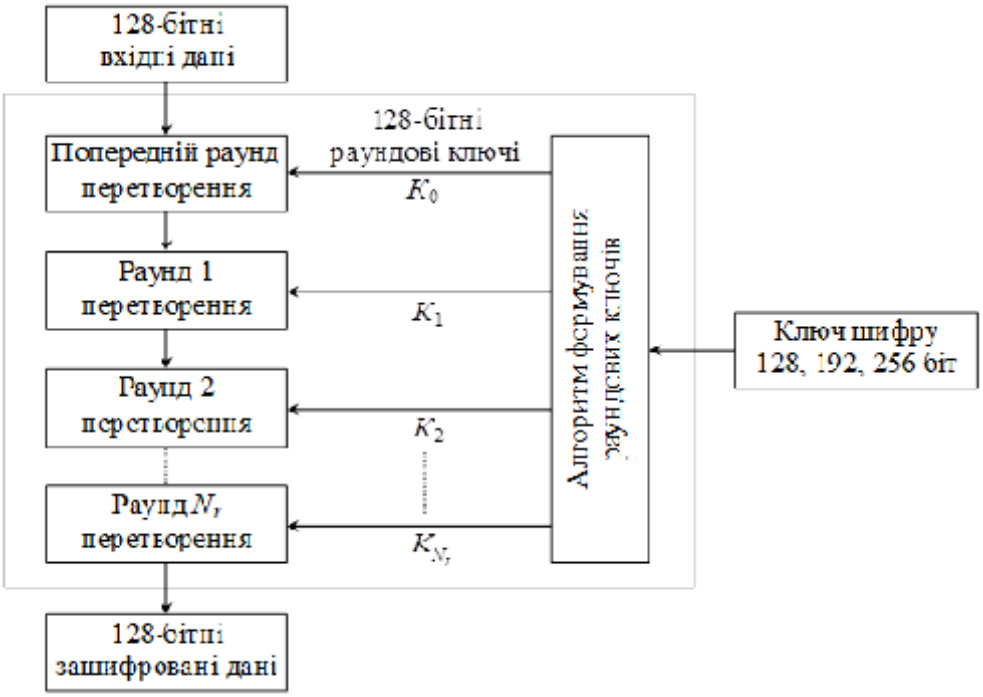


Рисунок 3.12 – Функційна схема стандарту шифрування AES.

Таблиця 3.10 - Відповідність між довжиною ключа, розміром блока даних і кількістю раундів у стандарті AES

Стандарт	Довжина ключа біт	Розмір блока даних біт	Кількість раундів
AES-128	128	128	10
AES-192	192	128	12
AES-256	256	128	14

Для того, щоб алгоритм AES став гідною заміною DES, було сформульовано вимоги до його архітектури [11–14], тобто, щоби він міг:

- надавати високий ступінь захисту;
- мати просту структуру та високу продуктивність;

- мати надійність, достатню для того, щоб протистояти майбутнім спробам його злому;
- бути настільки простим, щоб гарантувати ефективну процедуру шифрування;
- являти собою блочний шифр та працювати з 128-розрядними блоками даних;
- реалізувати шифрування приватним ключем, який може змінювати довжину (128, 192 і 256 розрядів).;
- використовувати операції, які легко реалізуються як апаратно, так і програмно;
- орієнтуватися на 32-розрядні процесори;
- не ускладнювати без необхідності структуру шифру .

Advanced Encryption Standard (AES), прийнятий як стандарт шифрування урядом США за результатами конкурсу. Алгоритм також відомий як Rijndael – це симетричний алгоритм блокового шифрування (розмір блоку дорівнює 128 біт, проте ключ може приймати різні значення, а саме: 128, 192 чи 256 біт).

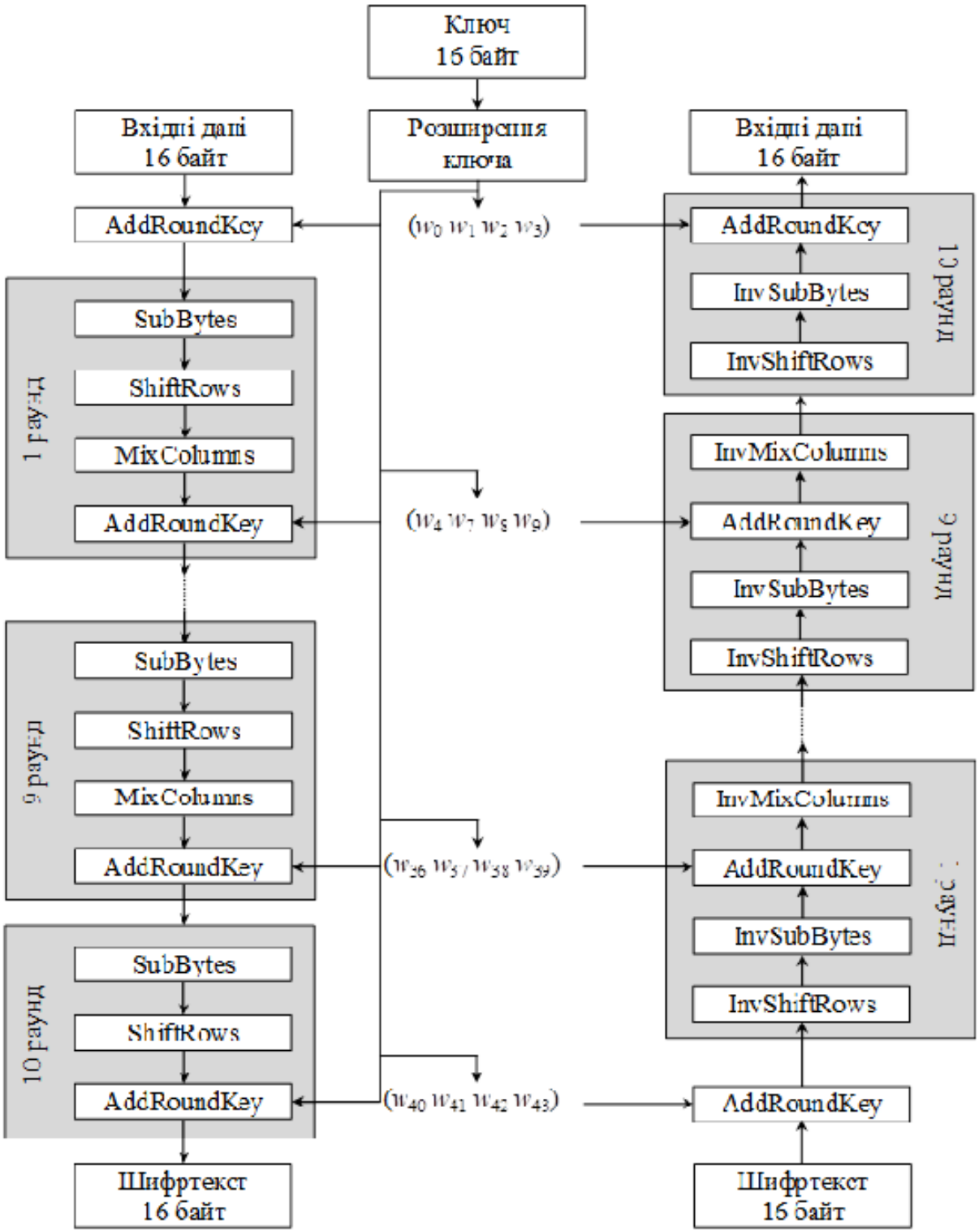


Рисунок 3.13 – Схема реалізації алгоритму AES

При цьому, початковий алгоритм Rijndael допускає довжину ключа і розмір блоку від 128 до 256 біт з кроком в 32 біта. Для позначення вибраних довжин input (введення), State (стан) і Cipher Key (ключ шифру) в 32-бітових словах використовується нотація N_b = 4 для input і State, N_k = 4, 6, 8 для Cipher Key відповідно для різних довжин ключів.

На початку шифрування input копіюється в масив State за правилом:

$$\text{State [r, c]} = \text{Input [r + 4c]}, \qquad (3.5)$$

для $0 \le r < 4$ та $0 \le c < N_b$.

Після цього до State застосовується процедура Add Round Key (додати циклічний ключ) і потім State проходить процедуру трансформації (раунд) 10, 12, або 14 разів (залежно від довжини ключа), при цьому потрібно врахувати, що останній раунд дещо відрізняється від попередніх. У результаті, після завершення останнього раунду трансформації, State копіюється в Output (вихід) за правилом:

$$\text{Output [}r + 4c\text{]} = \text{State [}r, c\text{]}, \qquad (3.6)$$

для $0 \le r < 4$ та $0 \le c < N_b$.

Таким чином, в алгоритмі AES використовуються 4 функціональних перетворення:

- Sub Bytes/State (Стан);
- Shift Rows (Зміщення);
- Mix Columns (Перемішування стовпців);
- Add Round Key (Додати циклічний ключ).

Кожна функція призначена для шифрування та є оберненою, тобто розшифрування передбачає застосування обернених функцій в зворотному напрямку. Внутрішні функції визначені в кінцевому полі. Це поле складається з усіх поліномів за модулем $m = f(x)$ незвідного полінома $f(x) = x^8+x^4+x^3+x+1$ над полем F_2.

1. Функція **Sub Bytes/State** виконує нелінійну підстановку кожного байту (числа x), тобто функція призначена для реалізації нелінійного шифру заміни. Треба пам’ятати, що саме нелінійність – це властивість блокового шифру, яка захищає його від криптоаналізу. Схему процедури Стану показано на рис.3.14.

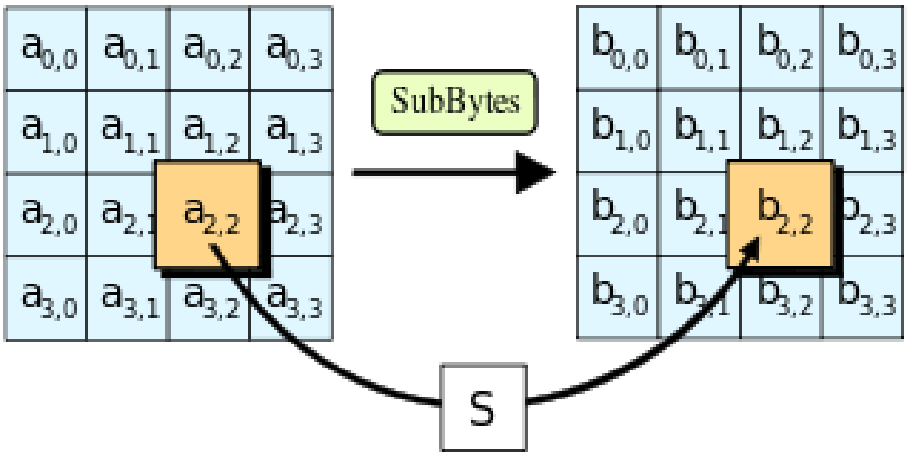


Рисунок 3.14 – Схема процедури SubBytes/Стан.

У процедурі SubBytes, кожен байт в State замінюється відповідним елементом у фіксованій 8-бітовій таблиці пошуку, $S; b_{ij} = S(a_{ij})$.

Процедура SubBytes обробляє кожен байт стану, незалежно здійснює нелінійну заміну байтів, при цьому використовує таблицю замін(S - box). Така операція забезпечує нелінійність алгоритму шифрування. Побудова S - box складається з двох кроків. По-перше, робиться узяття зворотного числа в полі Галуа $GF(2^8)$. По-друге, до кожного байта b з яких складається S - box застосовується наступна операція:

$$b'_i = b_i \oplus b_{(i+4)\bmod 8} \oplus b_{(i+5)\bmod 8} \oplus b_{(i+6)\bmod 8} \oplus b_{(i+7)\bmod 8} \oplus c_i , \quad (3.7)$$

де $0 \le i < 8$, і b_i – i-ий біт b,
а c_i – i-ий біт константи $c = 63_{16} = 99_{10} = 01100011_2$.

Таким чином забезпечується захист від атак, грунтованих на простих властивостях алгебри.

2. Функція **Shift Rows** – це шифр перестановки по рядках, є оберненим перетворенням. Схему процедури Зміщення показано на рис.3.15.

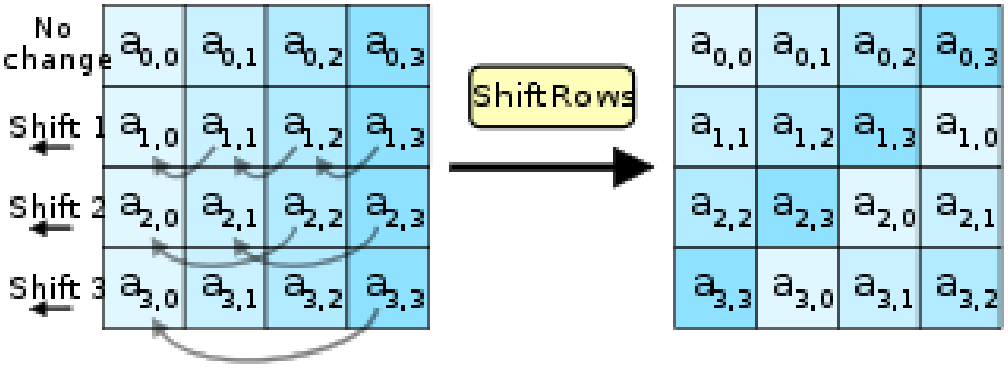


Рисунок 3.15 – Схема процедури Shift Rows / Зміщення.

У процедурі ShiftRows, байти в кожному рядку State циклічно зсуваються ліворуч. Розмір зсуву байтів кожного рядка залежить від її номеру. При цій трансформації рядка стани циклічно зсуваються на r байт по горизонталі, залежно від номера рядка. Для нульового рядка $r = 0$, для першого рядка r = 1 Би і т. д. Таким чином, кожна колонка вихідного стану після застосування процедури ShiftRows складається з байтів з кожної колонки початкового стану. Для алгоритму Rijndael патерн зміщення рядків для 128 - і 192-бітових рядків однаковий. Проте для блоку розміром 256 біт відрізняється від попередніх тим, що 2-і, 3-и і 4-і рядки зміщуються на 1, 3 і 4 байти, відповідно.

3. Функція **Mix Columns** застосовується до кожної колонки матриці. У процедурі MixColumns, кожна колонка стану перемножується з фіксованим багаточленом c(x). Схему процедури Перемішування стовпців показано на рис.3.16.

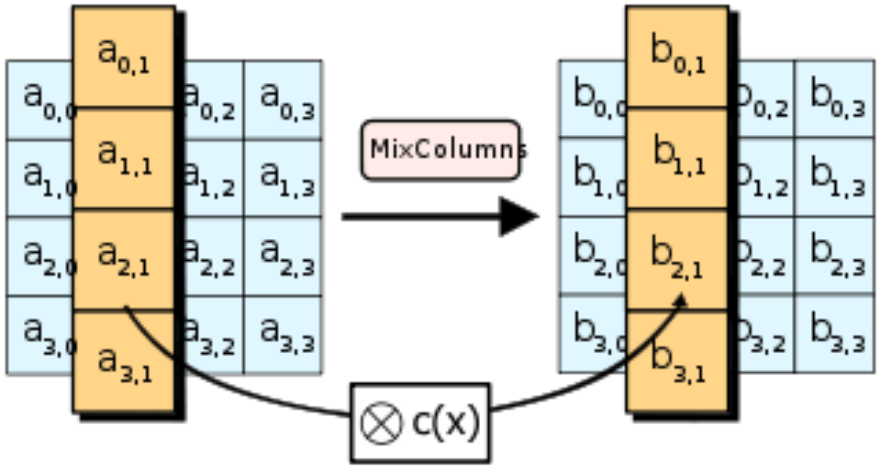


Рисунок 3.16 – Схема процедури MixColumns / Перемішування стовпців.

У процедурі MixColumns чотири байти кожної колонки State змішуються, використовуючи для цього оборотну лінійну трансформацію. MixColumns обробляє стани по колонках, трактуючи кожну з них як поліном четвертої міри. Над цими поліномами робиться множення в $GF(2^8)$ за модулем: $x^8+x^4+x^3+x+1$ на фіксований багаточлен $C(x) = 3x^3+x^2+x+2$.

Разом з ShiftRows, MixColumns вносить дифузію в шифр, тобто вони призначені для змішування байтів, розміщених в різних місцях блоку вихідного повідомлення. Суміш байтів, що стоять в різних позиціях блоку повідомлення, розширює розподіл повідомлень.

4. Функція **Add Round Key** забезпечує необхідну секретну випадковість розподілу повідомлень. Схему процедури додавання циклічного ключа показано на рис.3.17.

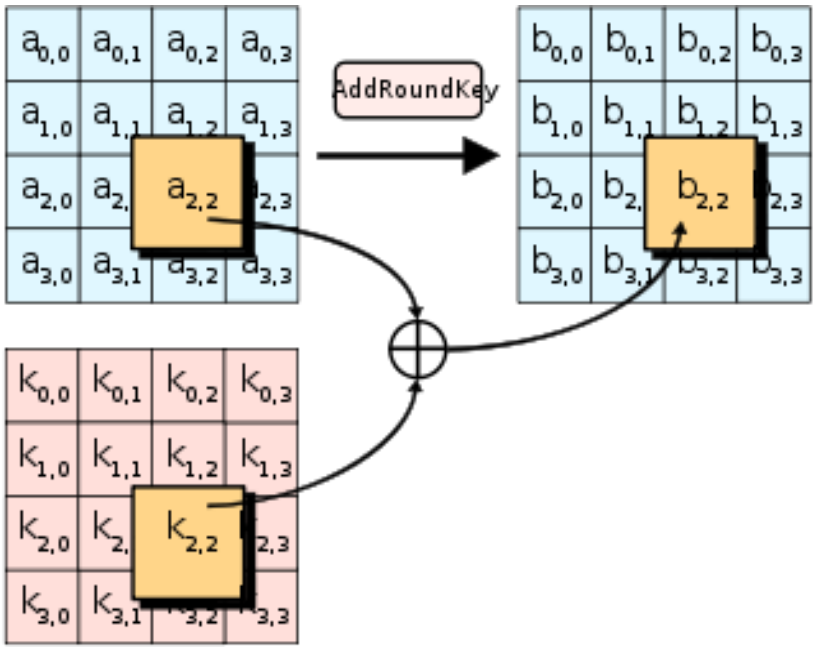


Рисунок 3.17 – Схема процедури Add Round Key / Додавання циклічного ключа.

У процедурі AddRoundKey кожен байт стану об’єднується з RoundKey, використовуючи операцію складання за модулем XOR($\oplus$).

У процедурі AddRoundKey, RoundKey кожного раунду об’єднується з State. Для кожного раунда Roundkey виходить з CipherKey використовуючи процедуру розширення ключа KeyExpansion; кожен RoundKey такого ж розміру, що і State. Процедура робить побітове складання XOR кожного байта State з кожним байтом RoundKey.

Функції застосовуються багаторазово, щонайменше 10 разів при довжині ключа 128 біт. Тобто від довжини ключа залежить кількість етапів шифрування. Так, при довжині ключа 128 біт, 64-розрядний блок відкритого тексту шифрується протягом 10 етапів шифрування. Якщо використовується довжина ключа 192 біта, то кількість етапів шифрування збільшується до 12 раундів. При довжині ключа в 256 біт, шифрування блоку відкритого тексту здійснюється впродовж 14 етапів шифрування.

Для алгоритмів блочного шифрування розроблені різноманітні режими, які забезпечують необхідні властивості блочних шифртекстів. Приміром, випадковість, можливість вирівнювання вихідних повідомлень до будь-якого розміру (щоби довжина шифртексту не була пов’язана з довжиною вихідного тексту), контроль за поширенням помилок, генерація ключа і так далі.

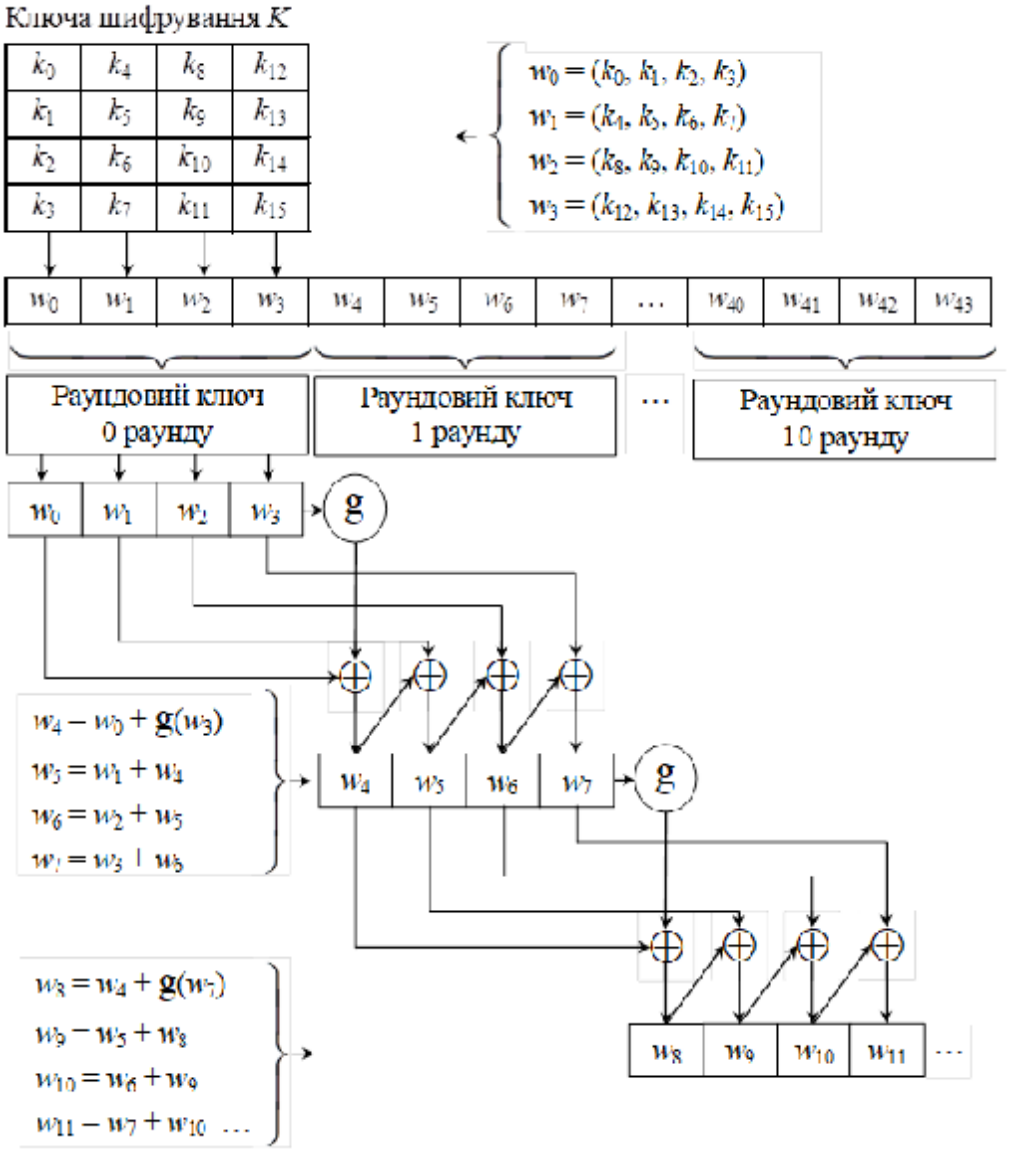


Рисунок 3.18 -

В криптосистемі AES використовується 6 режимів роботи.

1. Режим *Електронної книги кодів* (*electronic codebook, ECB*), він же режим простої заміни.

Повідомлення розбивається на блоки і кожен блок шифрується окремо, здійснюється присвоювання кодових слів, що взяті з шифрувальної книги.

Недоліком цього методу є те, що однакові блоки відкритого тексту шифруються в однакові блоки шифртексту.

Схема

Рисунок 3.19 -

2 Режим *Зчеплювання блоків шифру* (*cipher-block chaining, CBC*).

Усі блоки шифрованого тексту залежать не тільки від відповідного блоку вихідного тексту, а ще й від усіх блоків, які були оброблені до нього, а також від синхропосилання.

схема

Рисунок 3.20 –

3 Режим *Зворотного зв'язку по шифртексту* (*cipher feedback, CFB*) передбачає повернення шифр тексту на вхід алгоритму. Інакше алгоритм цього режиму називають гамуванням зі зворотним зв'язком. При цьому, алгоритм блочного шифрування можна замінити на відповідну односпрямовану геш-функцію.

Так само, як і в попередньому режимі, усі блоки шифрованого тексту залежать не тільки від відповідного блоку вихідного тексту, а ще й від усіх блоків, які були оброблені до нього, а також від синхропосилання.

схема

Рисунок 3.21 –

4 Режим *Зворотного зв'язку по виходу* (*output feedback, OFB*) можливо використовувати для шифрування повідомлень, для яких повторне передавання не є можливим. Інакше алгоритм цього режиму називають гамуванням. При цьому, алгоритм блочного шифрування можна замінити на відповідну односпрямовану геш-функцію.

схема

Рисунок 3.22 –

5 Режим *Лічильника* (*режим CTR)* передбачає повернення на вхід відповідного алгоритму значення лічильника, що є накопиченням з моменту старту. До поточного ключу і блокам вихідного повідомлення застосовується операція складання за модулем 2. Якщо зворотний звя'зок відсутній, алгоритм шифрування та розшифрування можуть виконуватися паралельно.

схема

Рисунок 3.23 –

Режим Лічильника перетворює блочний шифр в потоковий шифр. Він породжує наступний блок потоку ключа шифруванням послідовних значень лічильника. Лічильник може бути будь-якою функцією, що видає послідовність, яка гарантовано не повторюється впродовж тривалого часу. Найпростішими і

найпоширенішими є прості лічильники, що на кожному кроці збільшуються на одиницю.

6 AES-GSM схема

Рисунок 3.24 –

Щодо криптостійкості Advanced Encryption Standard, то вважається, що використовуваний в AES ключ в 128 біт - досить надійний захист проти лобової атаки, тобто з чисто математичної точки зору підібрати правильний пароль з усіх можливих - важкоздійсненне завдання. Незважаючи навіть на деякі недоліки AES, добути захищену за допомогою цього алгоритму інформацію практично не можливо.

Довжина ключа, що використовується при шифруванні визначає практичну доцільність виконання повного перебору, адже повідомлення, що зашифроване довшими ключами експоненціально складніше дешифрувати.

Обчислення кількості можливих варіантів ключа можна обчислити за формулою:

$$N = 2^n, \qquad (3.8)$$

де n – довжина ключової послідовності.

У Таблиці 3.11 надано можливе число комбінацій з урахуванням розміру ключа, згідно (3.8).

Таблиця 3.11 – Кількість можливих комбінацій з урахуванням розміру ключа

Розмір ключа, біт	Кількість можливих комбінацій
1	2
2	4
4	16
8	256
16	65536
32	4.2×10^9
56 (DES)	7.2×10^{16}
64	1.8×10^{19}

128 (AES, IDEA)	3.4×10^{38}
192 (AES)	6.2×10^{57}
256 (AES)	1.1×10^{77}

У міру збільшення розміру ключа кількість комбінацій зростає експоненціально. Математичні числення доводять, що розмір ключа в 128 біт надійнішим чином захищає від лобової атаки. У таблиці 3.12 показано час необхідний на злом ключа з урахуванням можливостей сучасних комп'ютерів.

Таблиця 3.12 – Час, необхідний на злом ключа

Розмір ключа, біт	Необхідний час для злому шифру
56	399 секунд
128	1.02×10^{18} років
192	1.872×10^{37} років
256	3.31×10^{56} років

Таким чином, навіть суперкомп'ютеру знадобилася б незліченно величезна кількість часу, щоб отримати доступ до інформації під захистом AES за допомогою лобової атаки. Для порівняння: вік Всесвіту - десь між 13 і 14 мільярдами років. Навіть якщо припустити, що деякий супер-суперкомп'ютер міг зламати алгоритм DES за одну секунду, то на злом AES у нього пішло б близько 149 трильйонів років. Тобто розміру ключа в 128 біт цілком достатньо для забезпечення криптостійкості сучасних алгоритмів шифрування, хоча цілком таємна інформація все одно шифрується з розміром ключа в 256 біт.

На сьогодні є багато областей, в яких AES зараз комерційно використовується. VPN програмне забезпечення найвищого класу містить реалізацію AES, зокрема пропозицій від Checkpoint, Cisco і Symantec. AES тепер часто застосовують у мережевих пристроях. Voice Over IP виробники використовують AES для безпеки телефону. Продавці тепер використовують AES для убезпечення управління технологічними процесами (SCADA) систем. AES навіть додано в загальний файл стиснення програм, таких як WinZip.

3.6.1 Завдання для самоперевірки з теми

Завдання 3.6.1. Нехай раундовий ключ для 8 раунду дорівнює

K_8 = (EA D2 73 21 B5 8D BA D2 31 2B F5 60 7F 8D 29 2F)$_{16}$.

Знайти перші 4 байти (32-розрядное слово) для раундового ключа 9 раунду (K_9).

Завдання 3.6.2. Нехай на вхід функції KeyExpansion(InputKey) алгоритму AES-128 надходить ключ шифрування *K*. Сформувати чотирибайтові слова розширеного ключа: w_4, w_5, …, w_{11} (знайти раундови ключі 1 та 2 раунду)

K = InputKey = (2B 7E 15 16 28 AE D2 A6 AB F7 15 88 09 CF 4F 3C)$_{16}$.

Завдання 3.6.3.Виконати перетворення AddRoundKey (State, RoundKey) для матриці стану State та раундовим ключем RoundKey

State = (38 56 0D D6 55 DD C5 3D 66 B1 BF 87 C5 97 D9 22)$_{16}$;

RoundKey = (5A BC 1E 4D F5 9D 85 9D 4C A3 50 50 87 7F D2 83)$_{16}$.

Завдання 3.6.4.Виконати перетворення InvMixColumns для матриці стану State.

State = (38 56 0D D6 55 DD C5 3D 66 B1 BF 87 C5 97 D9 22)$_{16}$

Завдання 3.6.5.Виконати перетворення MixColumns для матриці стану State.

State = (5A BC 1E 4D F5 9D 85 9D 4C A3 50 50 87 7F D2 83)$_{16}$

Завдання 3.6.6. Виконати перетворення InvShiftRows для матриці стану State.

State = (50 BC 00 3F 1E E4 CA AF CC BC AF F8 D0 D0 B8 BC)$_{16}$

Завдання 3.6.7. Виконати перетворення ShiftRows для матриці стану State.

State = (59 AB 76 63 6B 82 2B 59 47 72 01 63 6B 63 AF 2B)$_{16}$

Завдання 3.6.8. Виконати перетворення InvSubBytes для матриці стану State.

State = (53 65 63 75 72 69 74 79 4B 65 79 41 70 70 6C 65)$_{16}$

Завдання 3.6.9. Виконати перетворення InvSubBytes для байтів {36}$_{16}$, {DA}$_{16}$.

Завдання 3.6.10. Виконати перетворення SubBytes для матриці стану State.

State = (15 0E 0F 00 05 11 0B 15 16 1E 09 00 05 00 1B 0B)$_{16}$

Завдання 3.6.11. Представити блок даних у вигляді матриці стану AES-128 використовуючи таблицю ASCII кодів символів Windows. Текстовий блок – SecurityKeyApple.

Завдання 3.6.12. Представити блок даних у вигляді матриці стану AES-128. Текстовий блок – СКЛАДНІСТЬЗАДАЧІ.

Завдання 3.6.13. Для байтів $\{FD\}_{16}$, $\{94\}_{16}$, $\{E1\}_{16}$, $\{FF\}_{16}$ у полі $GF(2^8)$ AES-128 знайти мультиплікативно обернені елементи.

Завдання 3.6.14. Для байтів $\{F8\}_{16}$, $\{57\}_{16}$ у полі $GF(2^8)$ із незвідним багаточленом $p(x) = x^8 + x^4 + x^3 + x + 1$ знайти мультиплікативно обернений елемент та виконати перевірку ($p(x) = a(x)$ у алгоритмі Евкліда).

Завдання 3.6.15. Нехай задані многочлени

$$a(x) = \{03\}x^3 + \{01\}x^2 + \{01\}x + \{02\};$$
$$b(x) = \{30\}x^3 + \{5D\}x^2 + \{BF\}x + \{D4\}.$$

Знайти добуток $a(x)$ і $b(x)$ у полі AES, тобто многочлен $d(x)$

$$d(x) = a(x)\, b(x) = d_3x^3 + d_2x^2 + d_1x + d_0.$$

Контрольні питання з теми

1. Назвіть основні параметри (розмір блоку, ключа та кількість раундів) для 3-х версій AES ?
1. Які вимоги до архітектури AES ?
2. Які функціональні перетворення використовуються в алгоритмі AES?
3. Що таке функція Sub Bytes/State?
4. Що таке функція Shift Rows?
5. Що таке функція Mix Columns?
6. Що таке функція Add Round Key?
7. Опишіть раундові перетворення стандарту AES.
8. Яке розширене поле використається в стандарті AES?
9. Опишіть ітераційні перетворення в стандарті AES.
10. Які існують режими роботи AES?
11. Яка криптостійкість алгоритму AES?
12. Де може використовуватися алгоритм AES?

3.7 Алгоритм шифрування IDEA

Перший варіант шифру було опубліковано 1990 року. Його автори Ксуеджа Лай та Джеймс Мессі. Остаточна назва *IDEA* розшифровується як *International Data Encryption Algorithm* (Міжнародний Алгоритм Шифрування Даних). Існує

думка, що *IDEA* — це один з найбезпечніших блочних алгоритмів, опублікованих на сьогодні.

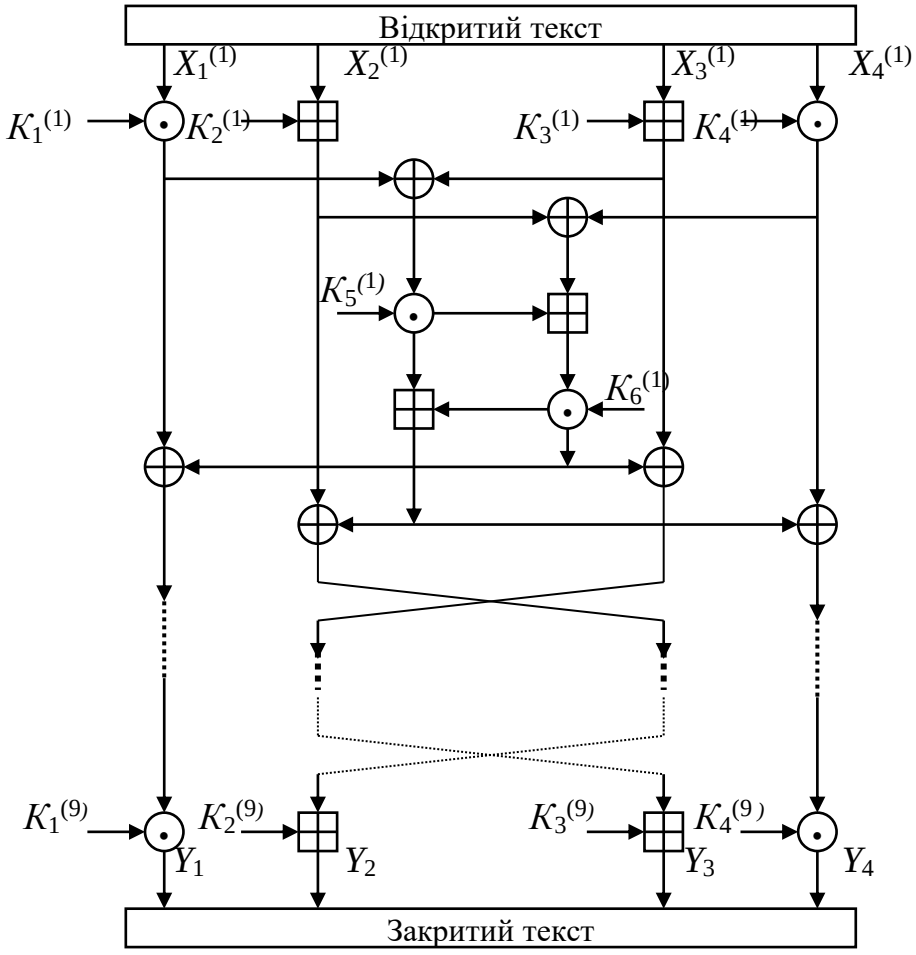
Цей алгоритм передбачає попередній поділ вихідної послідовності на 64-розрядні блоки, що шифруються за допомогою 128-бітового ключа. Алгоритм побудовано на базі модифікованої мережі Файстеля (рис. 3.25). Алгоритм шифрування є обернений.

Загальною засадою шифрування IDEA, як і у більшості блочних алгоритмів, є змішування та розсіювання. Алгоритм передбачає сполучування операцій основних алгебричних груп, до яких належать:

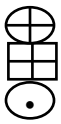
- XOR (додавання за модулем 2);
- додавання за модулем 2^{16};
- перемножування за модулем $2^{16} + 1$.

Всі перелічені операції виконуються з 16-бітовими підблоками.

Спочатку 64-розрядний блок розбивається на чотири 16-розрядних підблоки X_1, X_2, X_3 та X_4, які є вхідними даними для алгоритму. Усього алгоритм передбачає виконання восьми етапів. На кожному етапі чотири підблоки піддаються операціям додавання за модулем 2, додавання за модулем 2^{16} та множення за модулем $2^{16} + 1$ кожен з кожним та з шістьома 16-бітовими підключами.



X_i – 16-бітовий підблок відкритого тексту; Y_i – 16-бітовий підблок шифртексту;
$K_i^{(r)}$ – 16-бітовий підблок ключа;



– побітове додавання за модулем 2 16-бітових підблоків;
– додавання за модулем 2^{16} 16-бітових цілих чисел;
– перемножування за модулем $2^{16} + 1$ 16-бітових цілих чисел

Рисунок 3.25 – Схема шифрування одного блоку вихідного тексту IDEA

Кожний етап передбачає виконання таких операцій:

1 помножуються X_1 та перший підключ K_1;
2 виконується додавання за модулем 216 Х2 та другого підключа К2;
3 виконується додавання за модулем 216 Х3 та третього підключа К3;
4 помножуються Х4 та четвертий підключ К4;
5 виконується додавання за модулем 2 результатів 1 та 3 кроків;
6 виконується додавання за модулем 2 результатів 2 та 4кроків;
7 помножуються результати кроку 5 та п`ятий підключ К5;
8 виконується додавання за модулем 216 результатів 6 та 7 кроків;
9 помножуються результати кроку 8 та шостий підключ К6;
10 виконується додавання за модулем 216 результатів 7 та 9 кроків;
11 виконується додавання за модулем 2 результатів 1 та 9 кроків;
12 виконується додавання за модулем 2 результатів 3 та 9 кроків;
13 виконується додавання за модулем 2 результатів 2 та 10 кроків;
14 виконується додавання за модулем 2 результатів 4 та 10 кроків.

Виходом цикла є чотири підблоки, які здобуваються як результат виконання останніх чотирьох кроків алгоритму шифрування. Цикл шифрування завершується переставлянням двох внутрішніх блоків (за винятком останнього циклу). Отже, формуються входи для наступного циклу.

По завершенні останнього, восьмого циклу, виконується вихідне перетворювання:

1 помноження першого підблока X_1 та першого підключа K_1;
2 додавання за модулем 2^{16} другого підблока X_2 та другого підключа K_2;
3 додавання за модулем 2^{16} третього підблока X_3 та третього підключа K_3;
4 помноження четвертого підблока X_4 та першого підключа K_4.

Здобуті після виконання цих чотирьох операцій результати Y_1, Y_2, Y_3 та Y_4 знову сполучуються, утворюючи шифртекст. Аналогічно шифруються наступні 64-бітові блоки відкритого тексту.

Алгоритм IDEA для шифрування кожного 64-бітного блоку відкритого тексту передбачає використовування 52-х ключів (по шість для кожного з восьми

циклів та ще чотирьох для формування вихідного блока).

Спочатку 128-бітовий ключ розбивають на вісім 16-бітових підключів, шість із яких використовуються в першому циклі, а два залишаються для наступного циклу ($K_1^{(1)}$, $K_2^{(1)}$, $K_3^{(1)}$, $K_4^{(1)}$, $K_5^{(1)}$, $K_6^{(1)}$, $K_1^{(2)}$, $K_2^{(2)}$).

Потім 128-бітовий ключ зсовують на 25 біт ліворуч і знову поділяють на вісім ключів ($K_3^{(2)}$, $K_4^{(2)}$, $K_5^{(2)}$, $K_6^{(2)}$, $K_1^{(3)}$, $K_2^{(3)}$, $K_3^{(3)}$, $K_4^{(3)}$). . Перші чотири підключа використовують у другому циклі, останні чотири – у третьому циклі.

Формування наступних ключів виконується аналогічно до завершення алгоритму, тобто до тих пір, поки не будуть згенеровані 52 16-бітових підключа.

Результат формування підключів приведено у таблиці 3.13.

Процес розшифровування супроводжується генерацією ключів у зворотному порядку, причому певні з них змінюються оберненими величинами.

Таблиця 3.13 – Сформовані підключі для кожного етапу шифрування одного 64-бітного блоку даних

Номер етапу шифрування	Сформовані 16-бітові підключі	Лічільник підключів
1	($K_1^{(1)}$, $K_2^{(1)}$, $K_3^{(1)}$, $K_4^{(1)}$, $K_5^{(1)}$, $K_6^{(1)}$)	6
2	($K_1^{(2)}$, $K_2^{(2)}$, $K_3^{(2)}$, $K_4^{(2)}$, $K_5^{(2)}$, $K_6^{(2)}$)	12
3	($K_1^{(3)}$, $K_2^{(3)}$, $K_3^{(3)}$, $K_4^{(3)}$, $K_5^{(3)}$, $K_6^{(3)}$)	18
4	($K_1^{(4)}$, $K_2^{(4)}$, $K_3^{(4)}$, $K_4^{(4)}$, $K_5^{(4)}$, $K_6^{(4)}$)	24
5	($K_1^{(5)}$, $K_2^{(5)}$, $K_3^{(5)}$, $K_4^{(5)}$, $K_5^{(5)}$, $K_6^{(5)}$)	30
6	($K_1^{(6)}$, $K_2^{(6)}$, $K_3^{(6)}$, $K_4^{(6)}$, $K_5^{(6)}$, $K_6^{(6)}$)	36
7	($K_1^{(7)}$, $K_2^{(7)}$, $K_3^{(7)}$, $K_4^{(7)}$, $K_5^{(7)}$, $K_6^{(7)}$)	42
8	($K_1^{(8)}$, $K_2^{(8)}$, $K_3^{(8)}$, $K_4^{(8)}$, $K_5^{(8)}$, $K_6^{(8)}$)	48
Вихідне перетворення	($K_1^{(9)}$, $K_2^{(9)}$, $K_3^{(9)}$, $K_4^{(9)}$)	52

Алгоритм IDEA може працювати в будь-якому режимі, передбачуваному для блочного шифрування. Він має низку переваг відносно алгоритму DES. По-перше, він є безпечніше, оскільки його ключ є удвічі довше. По-друге, його внутрішня структура є більш складна й тому є більш стійка до криптоаналізу. По-третє, його програмна реалізація удвічі швидша, ніж DES. Завдяки перетворенням, що використовуються у алгоритмі, його можна реалізувати апаратно на підставі інтегральних схем, що забезпечує набагато більшу швидкодію, ніж при програмній реалізації. Саме цю особливість часто використовують при апаратному шифруванні потоку даних у мережах. Алгоритм IDEA застосувався в пакеті програм шифрування PGP. Цей алгоритм запатентовано у країнах Європи та США.

3.7.1 Завдання для самоперевірки з теми

Завдання 3.7.1. Привести схему реалізації мережі Фастеля в криптосистемі IDEA.

Контрольні питання з теми

1. Які основні параметри (розмір блоку, розмір ключа шифру, розмір ключа раунду та кількість раундів) алгоритму IDEA?
2. Які прості операції використовуються в раундах IDEA?
3. Які функціональні перетворення використовуються в алгоритмі IDEA?
4. Поясніть алгоритм формування раундових ключів при зашифруванні в IDEA?
5. Які переваги алгоритму IDEA над DES?
6. Які параметри алгоритму IDEA впливають на його криптостійкість?

3.8 Стандарт шифрування ДСТУ 7624:2014 «Калина»

Державний стандарт України ДСТУ 7624:2014 «Калина» [19] визначає сучасний алгоритм симетричного блокового шифрування для забезпечення конфіденційності й цілісності інформації при її обробці та встановлює режими його роботи.

Стандарт симетричного блокового шифрування «Калина» є результатом багаторічної плідної співпраці Державної служби спеціального зв'язку та захисту інформації України та провідних українських вчених. Даний алгоритм шифрування враховує досвід і результати проведення міжнародних та відкритих національних конкурсів криптографічних алгоритмів. Алгоритм ДСТУ 7624:2014 забезпечує досить високий рівень криптостікості порівняно з міжнародним стандартом AES (ISO/IEC 18033-3:2010), оскільки дає можливість застосовування блоку даних і ключа шифрування розміром до 512 біт. На сьогодні це єдиний у світі стандарт блокового шифрування, що підтримує 512-бітові симетричні ключі та дозволяє забезпечити надвисокий рівень захисту навіть секретної інформації, що може дати змогу стати йому основою для створення ефективних засобів криптографічного захисту майбутніх поколінь.

Національний стандарт підтримує змінні величини розміру блоку і довжини ключа шифрування, а саме 128, 256 і 512 біт в різних поєднаннях. Основні параметри шифру, такі як довжина ключа k і блоку даних l, кількість

раундів t та кількість стовпців матриці стану c пов'язані залежностями, які представлено у таблиці 3.14.

Таблиця 3.14 – Основні параметри шифру «Калина»

Довжина ключа k, біт	Довжина блоку відкритого тексту, l, біт	Кількість етапів шифрування одного блоку відкритого тексту t	Кількість стовпців матриці стану c
128, 256	128	10	2
256, 512	256	14	4
512	512	18	8

Стандарт ДСТУ 7624:2014 «Калина» забезпечує нормальний, високий та надвисокий рівень стійкості.

При розробці національного стандарту було прийнято рішення забезпечити прозорість проектування та використовувати консервативний підхід з застосуванням добре досліджених конструкцій, які мають забезпечувати запас міцності для можливості використання алгоритму в умовах криптоаналітичних технік, що швидко розвиваються. Даний стандарт розроблено з урахуванням існуючих та потенційних загроз, подальшого інтенсивного розвитку інформаційних технологій та необхідності активного використання протягом кількох наступних десятиліть.

В алгоритмі шифрування даних «Калина» використовуються криптографічні перетворення, які відповідають сучасним вимогам до рівня криптостійкості та швидкодії. Стандарт блокового симетричного шифрування ДСТУ 7624:2014 має десять різних режимів роботи, що широко поширені відповідно до міжнародних стандартів ISO/IEC 10116:2006. Ефективність реалізації систем, засобів та протоколів криптографічного захисту інформації в інформаційно-телекомунікаційних системах різного призначення може бути забезпечена саме наявністю такої кількості режимів роботи алгоритму.

Це спрямовано на забезпечення широкого застосування ДСТУ 7624:2014, у тому числі для захисту інформації, що передається комп'ютерними мережами, прозорого шифрування жорстких дисків і змінних носіїв, електронних документів, ключових даних.

Національний стандарт шифрування ДСТУ 7624:2014 «Калина» належить до SPN, байт-орієнтованих шифрів.

До блокового шифру «Калина» ставляться такі вимоги:

- високий рівень криптографічної стійкості з достатнім запасом у разі появи нових атак протягом тривалого часу;
- висока швидкодія програмної реалізації на сучасних та перспективних платформах;
- компактність програмної та програмно-апаратної реалізації; можливість ефективної інтеграції декількох алгоритмів в одному засобі криптографічного захисту;
- прозорість проектування, консервативний підхід до забезпечення стійкості;
- вища (або однакова) ефективність порівняно з найкращими світовими рішеннями.

Оскільки в алгоритмах симетричного блокового шифрування «Калина» та AES («Rijndael») використовуються аналогічні криптографічні перетворення, то можна порівняти їх за основними параметрами [20].

Основні відмінності стандартів «Калина» та AES :

- збільшена кількість етапів шифрування (до 18 при довжині ключа 512 біт);
- використання додавання за модулем 2^{64} і за модулем 2 для введення ключової інформації (захист від алгебричних атак, лінійного та диференціального криптоаналізів, інтерполяційної атаки тощо);
- використання чотирьох блоків нелінійного перетворення (S-блоків) замість одного (додатковий захист від алгебричних атак, поліпшення властивостей розсіювання алгоритму – покращені статистичні властивості, відповідно, більш високий рівень стійкості до диференціального та лінійного криптоаналізів тощо);
- використання випадково сформованих чотирьох блоків, відібраних критеріями стійкості до диференціального, лінійного криптоаналізів, ступені нелінійності булевих функцій (на відміну від S-блоку Rijndael/Camellia та інших шифрів, що використовують звернення в полі та, відповідно, квадратичні залежності між входом і виходом, – захист від алгебричних атак);
- принципово нова схема створення підключів (захист від усіх відомих атак на схеми створення підключів);
- досить висока продуктивність;
- можливість відновлення сеансового ключа за окремим підключем (додатковий захист від атак, що виконують відновлення підключів).

Крім того, стандарт «Калина» має аналогічну або навіть більш високу швидкодію на сучасних і перспективних програмних та програмно-апаратних платформах. Усі поліпшення алгоритму спрямовані на збільшення криптостійкості та запобігання потенційним вразливостям відносно алгоритму Rijndael. Основними перевагами шифру порівняно з іншими міжнародними аналогами є можливість застосовувати блок даних і ключ шифрування розміром до 512 біт, збільшена кількість циклів шифрування та принципово нова схема створення підключів, що забезпечують захист від усіх відомих атак на схеми їх створення. Калина має 10 режимов роботи.

Контрольні питання з теми

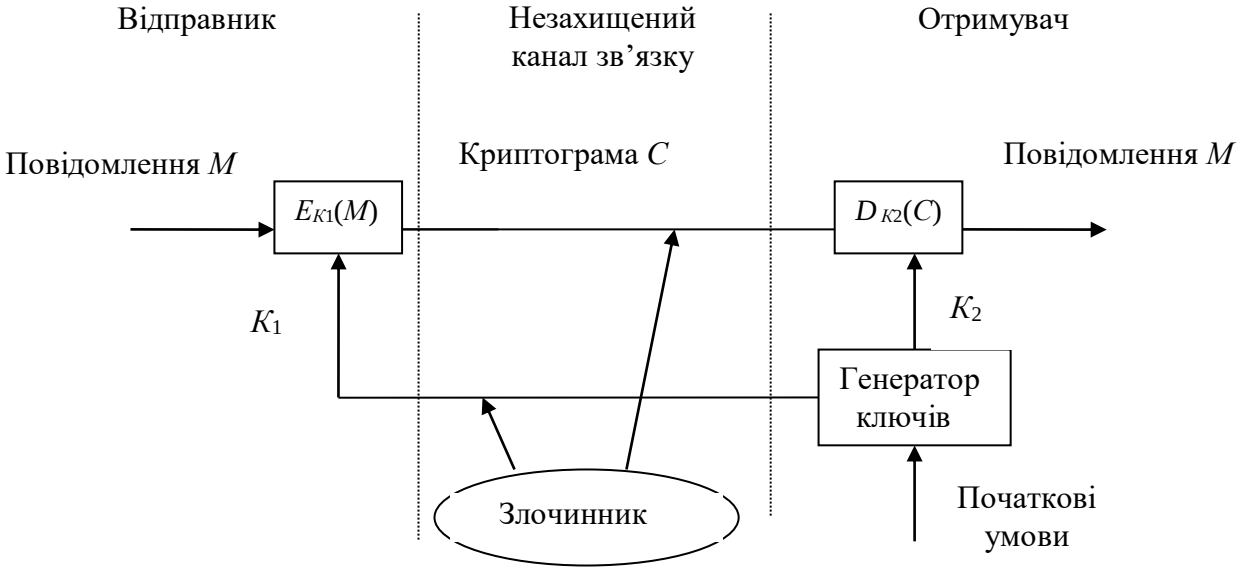
1. Назвіть основні параметри (розмір блоку, ключа та кількість раундів) для 3-х версій шифру «Калина»?
2. Які основні вимоги до блокового шифру «Калина»?
3. Які основні відмінності стандартів «Калина» та AES?
4. Назвіть режими роботи стандарту «Калина»?
5. Опишіть раундові перетворення стандарту ДСТУ 7624:2014.
6. Опишіть ітераційні перетворення в стандарті ДСТУ 7624:2014.
7. Яке розширене поле використовується в стандарті ДСТУ 7624:2014?
8. В якій послідовності виконуються перетворення в стандарті ДСТУ 7624:2014?
9. Які параметри стандарту ДСТУ 7624:2014 впливають на його криптостійкість?

4 АСИМЕТРИЧНІ КРИПТОГРАФІЧНІ СИСТЕМИ

4.1 Концепція криптосистеми з відкритим ключем

Ефективними системами криптографічного захисту даних є асиметричні криптосистеми. Такі системи також називають криптосистемами з відкритим ключем. Такої назви вони дістали внаслідок того, що для зашифровування даних використовується один ключ, K_1 (відкритий, несекретний), проте для розшифровування – другий ключ, K_2 (секретний). Розшифровування даних за допомогою відкритого ключа не є можливим. Ключ розшифровування не можна здобути з ключа зашифровування.

Узагальнену схему криптосистеми з відкритим ключем подано на рис. 4.1.



K_1 – відкритий ключ відправляча; K_2 – закритий ключ отримувача;
$E_{K1}(M)$ – зашифровування відкритого тексту M за допомогою ключа K_1;
$D_{K2}(C)$ - розшифровування шифртексту C за допомогою ключа K_2

Рисунок 4.1 – Узагальнена схема асиметричної криптосистеми

Генератор ключів розташовується на боці отримувача, щоби не пересилати секретний ключ K_2 незахищеним каналом. Значення ключів K_1та K_2 не залежать від початкового стану генератора ключів.

Характерні особливості асиметричних криптосистем:

1 Відкритий ключ K_1 та криптограма C можуть пересилатись незахищеним каналом зв'язку.

3 Алгоритми зашифрування

4

$$E_{K1} : M \rightarrow C$$

та розшифрування

$$D_{K2} : C \rightarrow M$$

є відкритими.

Захист інформації в асиметричній криптосистемі базується на секретності ключа K_2.

У. Диффі та М. Хеллман сформулювали вимоги, які необхідно виконувати для того, щоби шифрування за допомогою асиметричної криптосистеми було надійним:

1 обчислювання ключів K_1 та K_2 має бути легким, щоби отримувач міг згідно з початковими умовами швидко їх віднайти;

2 відправник, який знає відкритий ключ K_1 та повідомлення M, може легко обчислити криптограму

$$C = E_{K1}(M);$$

3 отримувач за допомого секретного ключа K_2 та криптограми C може легко відновити вихідне повідомлення:

$$M = D_{K2}(C) = D_{K2}(E_{K1}(M));$$

4 злочинник, якому відомий відкритий ключ K_1, за спроби обчислити секретний ключ K_2 наштовхується на задачу, яку не можна розв'язати;

5 злочинник, котрому є відомі відкритий ключ $K1$ та криптограма C, за спроби обчислити вихідне повідомлення M наштовхується на задачу, яку не можна розв'язати.

4.1.1 Завдання для самоперевірки з теми
4.1.2

Завдання 4.1.1 Привести схему секретної системи зв'язку для асиметричної криптосистеми.

Контрольні питання з теми

1. Що таке криптосистемами з відкритим ключем?
2. Які особливості асиметричних криптосистем?
3. Які вимоги пред'являються до асиметричних криптосистем?

4.2 Однонапрямлені функції

Концепція асиметричних криптосистем базується на використовуванні однонапрямлених функцій. Неформально однонапрямлену функцію можна визначити в такий спосіб.

Нехай X та Y – певні довільні множини. Функція

$$f : X \to Y$$

є однонапрямленою, коли для всіх $x \in X$ можна легко обчислити функцію

$$y = f(x), \text{ де } y \in Y.$$

Водночас для більшості $y \in Y$ є надто складно здобути значення $x \in X$ таке, щоби $f(x) = y$ (при цьому вважається, що існує щонайменш одне таке значення x).

Головним критерієм віднесення функції f до класу однонапрямлених функцій є відсутність ефективних алгоритмів обернених перетворювань $Y \to X$.

В якості прикладу однонапрямленої функції розглянемо цілочислове множення. Задача обчислювання добутку двох надто великих цілих чисел P та Q

$$N = P \cdot Q$$

є відносно легкою для ЕОМ. Навпаки, обернена задача – розкладання на складові великого цілого числа $N = P \cdot Q$ (обчислювання дільників P та Q) – є задачею, яку не можна розв'язати за великих значень N.

Іншим прикладом однонапрямленої функції є модульна експонента з фіксованою основою та модулем. Нехай A та N – цілі числа, такі що $1 \leq A < N$. Обчислимо множину

$$Z_N = \{0, 1, 2, \ldots, N - 1\}.$$

Тоді модульна експонента з основою A за модулем N являє собою функцію

$$f_{A,N} : Z_N \to Z_N;$$
$$f_{A,N}(x) \equiv A^x \pmod{N},$$

де X – ціле число, $1 \leq x \leq N - 1$.

Існують ефективні алгоритми, спроможні вельми швидко обчислити значення функції $f_{A,N}(x)$.

Якщо $y = A^x$, то можна записати $x = \log_A(y)$. Тому задачу обернення функції

$f_{A,\ N}(x)$ називають задачею віднаходження дискретного логарифму, яку можна сформулювати в такий спосіб.

Для відомих цілих A, N та y треба віднайти ціле число x, таке що

$$A^x \pmod N = y.$$

Це є задачею, яку не можна розв'язати за припустимий час. Тому модульна експонента вважається за однонапрямлену функцію.

Другим класом функцій, використовуваних при побудові криптосистем з відкритим ключем, є такі, котрі називаються однонапрямленими функціями з секретом. Функція називається однонапрямленою з секретом тоді, якщо вона є однонапрямленою й, окрім того, існує змога ефективного обчислювання оберненої функції, коли є відомий „таємний хід" (секретне число, рядок чи інша інформація, яка асоціюється з цією функцією). Саме такою функцією є модульна експонента з фіксованим модулем та показником степені, яку використовують в криптосистемі шифрування даних RSA.

Контрольні питання з теми

1. Що таке однонапрямлені функцій?
2. Які існують однонапрямлені функції?
3. Чому в асиметричних криптосистемах використовують однонапрямлені функції?
4. Що таке однонапрямлена функція з секретом?

4.3 Криптосистема шифрування даних RSA

Першою і найбільш відомою криптографічною системою з відкритим ключем є так звана система RSA, запропонована 1978 року. Її назва походить від перших літер прізвищ авторів Rivest, Shamir та Aldeman, які розробили її під час спільних досліджувань у Массачусетському технологічному інституті 1977 року. Алгоритм RSA став першим алгоритмом, здатним працювати як в режимі шифрування даних, так і в режимі електронного цифрового підпису.

Криптостійкість алгоритму грунтується на складності розкладання надто великих цілих чисел на прості співмножники та на складності обчислювання дискретних логарифмів.

У криптосистемі RSA відкритий ключ K_1, секретний ключ K_2, повідомлення M та криптограма C належать до множини цілих чисел

$$Z_N = \{0, 1, 2, \ldots, N-1\},$$

де N – модуль:

$$N = P \cdot Q.$$

Тут P та Q – випадкові великі прості числа. Для забезпечення максимальної безпеки обирають P та Q однакової довжини і зберігають це у таємниці.

Множина Z_N з операціями складання та множення за модулем N утворює арифметику за модулем N.

Відкритий ключ K_1 обирають випадково в такий спосіб, щоби виконувалися умови:

$$1 < K_1 \leq \varphi(N); \quad \text{НСД}\,(K_1, \varphi(N)\,) = 1; \quad \varphi(N) = (P-1)\,(Q-1),$$

де $\varphi(N)$ – функція Ейлера, котра зазначає кількість додатних цілих чисел в інтервалі від 1 до N, які є взаємно простими з N.

Друга умова вказує на те, що відкритий ключ та функція Ейлера мають бути взаємно простими.

Далі, використовуючи розширений алгоритм Евкліда, обчислюють секретний ключ K_2, такий що

$$K_1 \cdot K_2 \equiv 1 (\text{mod}\ \varphi(N))$$

чи

$$K_1 = K_2^{-1}\,[\text{mod}\ (P-1)\,(Q-1)].$$

Це можна обчислити, тому що отримувач знає числа P та Q й може легко віднайти $\varphi(N)$.

Відкритий ключ K_1 використовують для зашифровування, а секретний ключ K_2 – для розшифровування даних.

Перетворювання зашифровування визначає криптограму C згідно з формулою:

$$C = E_{K1}\,(M) \equiv M^{K1}\ (\text{mod}\ N).$$

Проте обернену задачу (розшифровування криптограми C) можна розв’язати згідно з формулою

$$M = D_{K2}\,(C) \equiv C^{\,K2}\ (\text{mod}\ N).$$

Припустімо, що користувач A хоче передати користувачеві B повідомлення,

зашифроване за допомогою криптосистеми RSA. Тоді користувач A є відправником, а користувач B – отримувачем повідомлення. Криптосистему утворює отримувач повідомлення, тобто користувач B. Розглянемо послідовність дій користувачів A та B на прикладі.

Приклад 4.1. Треба зашифрувати повідомлення “CAB” за допомогою криптосистеми RSA.

Задля простоти використовуватимемо маленькі числа (на практиці застосовуються набагато більші).

1 Користувач B обирає два випадкових простих числа : $P = 3$ та $Q = 11$.

2 Користувач B визначає модуль $N = 3 \cdot 11 = 33$.

3 Користувач B визначає функцію Ейлера

$$\varphi(N) = \varphi(33) = (P - 1)(Q - 1) = 2 \cdot 10 = 20.$$

Відкритий ключ K_1 обирають випадково, в такий спосіб, щоби виконувалися умови

$$1 < K_1 \leq 20; \quad \text{НСД}\,(K_1, 20) = 1.$$

Отже, нехай буде $K_1 = 7$.

4 Користувач B обчислює значення секретного ключа K_2 за допомогою розширеного алгоритму Евкліда, для якого задовольняється співвідношення

$$K_1 \cdot K_2 \equiv 1 (\text{mod}\ \varphi(N)),$$

тобто $K_2 \equiv 7^{-1} (\text{mod}\ 20) \equiv 3$.

5 Користувач B передає користувачеві A пару чисел ($N = 33$, $K_1 = 7$).

6 Користувач A подає шифроване повідомлення як послідовність цілих чисел за допомогою відображення: $A \rightarrow 1$, $B \rightarrow 2$, $C \rightarrow 3$. Тоді повідомлення набере вигляду 312 (у двійковій формі 011001010), тобто $M_1 = 3$, $M_2 = 1$, $M_3 = 2$.

7 Користувач A зашифровує за допомогою ключа $K_1 = 7$ та модуля $N = 33$ текст, розбитий на блоки в такий спосіб:

$$C_i = M_i^{K1} (\text{mod}\ N) = M_i^{7} (\text{mod}\ 33)$$

$C_1 \equiv (3^7) (\text{mod}\ 33) \equiv 2187 (\text{mod}\ 33) \equiv 9$;

$C_2 \equiv (1^7) (\text{mod}\ 33) \equiv 1 (\text{mod}\ 33) \equiv 1$;

$C_3 \equiv (2^7) (\text{mod}\ 33) \equiv 128 (\text{mod}\ 33) \equiv 29$.

Користувач A передає користувачеві B криптограму $C_1, C_2, C_3 = 9, 1, 29$.

8 Користувач B розшифровує отримане повідомлення (9, 1, 29) за допомогою секретного ключа $K_2 = 3$:

$$M_i = C_i^{K2} (\text{mod}\ N) = C_i^{3} (\text{mod}\ 33)$$

$M_1 \equiv (9^3) (\text{mod}\ 33) \equiv 729 (\text{mod}\ 33) \equiv 3$;

$M_2 \equiv (1^3) \pmod{33} \equiv 1 \pmod{33} \equiv 1;$

$M_3 \equiv (29^3) \pmod{33} \equiv 24389 \pmod{33} \equiv 2.$

Отже, відновлено вихідне повідомлення – CAB.

Недоліком системи RSA є менша, порівняно з симетричними криптосистемами, стійкість, унаслідок чого довжина ключів у таких криптосистем повинна бути вчетверо-вп'ятеро більше.

Криптосистеми RSA реалізовується як в апаратний, так і в програмний спосіб. Апаратна реалізація RSA майже в 1000 разів є повільніша за апаратну реалізацію симетричного криптоалгоритму DES. Програмна реалізація RSA майже в 100 разів повільніша за програмну реалізацію симетричного криптоалгоритму DES. З розвитком технологій ці характеристики можуть змінюватися, але швидкодія асиметричних систем ніколи не сягне швидкодії симетричних криптосистем.

4.3.1 Завдання для самоперевірки з теми

Завдання 4.3.1 Зашифрувати повідомлення "BAD" за допомогою криптосистеми RSA, якщо $P = 7$ та $Q = 11$.

Контрольні питання з теми

1. На чому ґрунтується криптостійкість алгоритму RSA?
2. Опишіть алгоритм зашифровування та розшифровування в RSA?
3. Яка швидкодія асиметричних криптосистем порівняно з симетричними?
4. Яке є призначення асиметричних криптосистем?
5. За якими принципами обирається пара ключів в асиметричній криптосистемі RSA?
6. Який ключ використовується для зашифровування відкритого повідомлення в криптосистемі RSA?
7. Який ключ використовується для розшифровування криптограми в криптосистемі RSA?

…..

Контрольні питання з теми.

Опишіть процедуру генерації ключів в RSA.

Побудуйте схему криптосистеми з відкритим ключем.

Від чого залежить криптостійкість алгоритму RSA?

Опишить атаку на алгоритм RSA.

4.4 Схема шифрування Ель Гамаля

Схема Ель Гамаля здатна працювати як в режимі шифрування даних, так і в режимі електронного цифрового підпису.

Криптостійкість алгоритму ґрунтується на складності обчислювання дискретних логарифмів в кінцевому полі.

Для того щоби генерувати пару ключів (відкритий ключ, секретний ключ), спочатку обирають певне велике просте ціле число P й велике ціле число G, причому $G < P$. Числа P та G не є секретними.

Потім обирають випадкове ціле число X, причому $X < P$. Число X є секретним ключем.

Далі обчислюють значення відкритого ключа Y

$$Y = G^X \bmod P.$$

Для того, щоби зашифрувати повідомлення M, обирають випадкове ціле число K, $1 < K < (P - 1)$, таке, що K та $(P - 1)$ є взаємно простими. Потім обчислюються цілі числа :

$$a \equiv G^K \bmod P;$$
$$b \equiv Y^K M \bmod P.$$

Пара чисел (a, b) є шифртекстом.

Для того щоб розшифрувати шифртекст (a, b), обчислюють

$$M \equiv b/a^X \pmod P.$$
$$a^X \equiv G^{KX} \bmod P,$$

можна записати

$$b/a^X \pmod P \equiv Y^K M/a^X \pmod P \equiv G^{KX} M/G^{KX} \pmod P \equiv M \pmod P.$$

Приклад 4.1. Треба зашифрувати повідомлення $M = 5$ за допомогою криптосистеми Ель Гамаля.

Оберемо $P = 11$, $G = 2$, секретний ключ $X = 8$.

Обчислимо значення відкритого ключа Y:

$$Y \equiv G^X \bmod P \equiv 2^8 \bmod 11 \equiv 256 \bmod 11 \equiv 3,$$

тобто $Y = 3$.

Оберемо певне випадкове число $K = 9$.

Переконаємося, що НСД $(K, P - 1) = 1$. Дійсно, НСД $(9, 10) = 1$.

Обчислимо пару чисел a та b:

$a \equiv G^K \bmod P \equiv 2^9 \bmod 11 \equiv 512 \bmod 11 \equiv 6;$

$b \equiv Y^K M \bmod P \equiv 3^9 \cdot 5 \bmod 11 \equiv 19683 \cdot 5 \bmod 11 \equiv 98415 \bmod 11 \equiv 9.$

Пара чисел (6, 9) є шифртекстом.

Розшифруємо цей шифртекст. Обчислимо повідомлення *M*, використовуючи секретний ключ $X = 8$:

$$M \equiv b/a^X \pmod{P} \equiv 9/6^8 \bmod 11 \equiv 9/1679616 \bmod 11 \equiv 5,$$

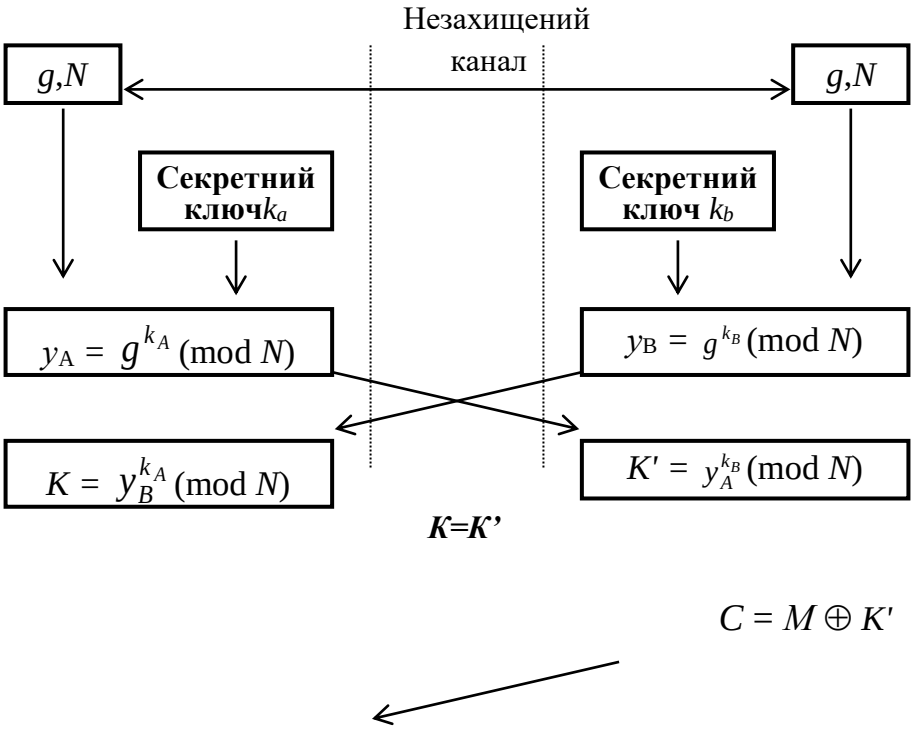
тобто $1679616 \cdot M \equiv 9 \bmod 11$.

Розв'язок цього порівняння: $M = 5$.

Алгоритм Ель Гамаля призначений для зашифрування та розшифрування повідомлення за допомогою узгодженого сеансового ключа на підставі протоколу Діффі-Хеллмана (рис.4.2). Алгоритм Діффі-Хеллмана призначений для обміну криптографічними ключами. За цим протоколом кожна із сторін узгоджує ключ шляхом передачі власних публічних ключів, при цьому сам сеансовий ключ не передається незахищеним каналом зв'язку.

При шифруванні методом гамування, текст та ключ мають бути представлені у двійковому вигляді.

Приклад 4.2. Зашифруємо повідомлення M=14 та розшифруємо криптограму C за алгоритмом Ель Гамаля, якщо N=43;g=11; $к_a$=14; $к_b$=21



де *K=K'– спільний сеансовий ключ*

M – відкритий текст; C – криптограма.

Рисунок 4.2 – Схема протоколу шифрування Ель Гамаля.

При шифруванні методом гамування, текст та ключ мають бути представлені у двійковому вигляді.

1. Розрахуємо публічний ключ y_A:

$$y_A = g^{k_A} = 11^{14}(mod\ 43) = 1$$

2. Розрахуємо публічний ключ y_B:

$$y_B = g^{k_B} = 11^{21}(mod\ 43) = 1$$

3. Розрахуємо сеансовий ключ *K(K')* :

$$K = y_B^{k_A} \text{ (mod } N) = 1^{14}(\text{mod } 43) = 1$$

$$K' = y_A^{k_B} \text{ (mod } N) = 1^{21}(\text{mod } 43) = 1$$

4. Зашифруємо повідомлення M:

$$C = M \oplus K' = 14 \oplus 1 = 15$$

$$1110 \oplus 0001 = 1111$$

5. Розшифруємо криптограму C:

$$M = C \oplus K = 15 \oplus 1 = 14$$

$$1111 \oplus 0001 = 1110$$

4.4.1 Завдання для самоперевірки з теми

Завдання 4.4.1 Зашифрувати повідомлення *M*=14 та розшифрувати криптограму *C* за алгоритмом Ель Гамаля, якщо варіантом завдання є номер студента у списку групи (за модулем 10)

:

1 варіант $N=23;\ g=11;\ к_a=14;\ к_b=21;$
2 варіант $N=29;\ g=11;\ к_a=14;\ к_b=21;$
3 варіант $N=31;\ g=11;\ к_a=14;\ к_b=21;$
4 варіант $N=37;\ g=11;\ к_a=14;\ к_b=21;$
5 варіант $N=43;\ g=11;\ к_a=14;\ к_b=21;$
6 варіант $N=23;\ g=12;\ к_a=15;\ к_b=19;$
7 варіант $N=29;\ g=12;\ к_a=15;\ к_b=19;$
8 варіант $N=31;\ g=12;\ к_a=15;\ к_b=19;$
9 варіант $N=37;\ g=12;\ к_a=15;\ к_b=19;$
10 варіант $N=43;\ g=12;\ к_a=15;\ к_b=19.$

Контрольні питання з теми

1. На чому ґрунтується криптостійкість алгоритму Ель Гамаля?
2. Опишіть алгоритм зашифровування та розшифровування в Ель Гамаля.
3. Для чого використовується протокол Діффі-Хеллмана?

4.5 Комбінований метод шифрування

Переваги асиметричних криптосистем полягають в тому, що нема потреби передавати будь-кому значення секретного ключа чи перевіряти його слушність.

У симетричних системах існує небезпека розкриття секретного ключа під час передавання повідомлення. Але алгоритми, які лежать в підгрунті асиметричних криптосистем, мають такі недоліки:

- генерування нових секретних та відкритих ключів базується на генеруванні нових великих простих чисел, а перевірка чисел на простоту потребує багато часу для обчислювання на ЕОМ;
- процедури зашифровування та розшифровування, пов’язані з піднесенням до степеня багатознакового числа, є надто громіздкі.

Тому швидкодія криптосистем з відкритим ключем є набагато менша за швидкодію криптосистем з секретним ключем.

Комбінований метод зашифровування надає змогу поєднувати переваги великої секретності, які надають асиметричні криптосистеми, з перевагами великої швидкості роботи симетричних криптосистем.

За такого методу криптосистема з відкритим ключем використовується для зашифровування, передавання та розшифровування лише секретного ключа симетричної криптосистеми, а симетрична криптосистема застосовується для зашифровування та передавання вихідного відкритого тексту. В такому разі криптосистема з відкритим ключем не замінює, а доповнює симетричну криптосистему, надає змогу збільшувати захищеність інформації, що передається каналами зв’язку.

Якщо користувач *A* хоче передати користувачеві *B* повідомлення, зашифроване комбінованим методом, тоді алгоритм буде таким:

1 створити (наприклад згенерувати у випадковий спосіб) симетричний сеансовий ключ K_S;

2 зашифрувати повідомлення *M* за допомогою сеансового ключа K_S;

3 зашифрувати сеансовий ключ K_S за допомогою відкритого ключа K_B користувача *B* та власного секретного ключа K_A;

4 передати незахищеним каналом зв’язку на адресу користувача *B* повідомлення *M* разом з сеансовим ключем K_S, які є зашифровувані.

Дії користувача *B* будуть зворотними:

5 розшифрувати сеансовий ключ K_S за допомогою власного секретного ключа $K_в$ та відкритого ключа K_A користувача *A;*

6 за допомогою здобутного сеансового ключа K_S розшифрувати та прочитати повідомлення *M.*

При використовуванні комбінованого методу шифрування можна бути впевненим, що лише користувач *B* зможе розшифрувати сеансовий ключ K_S та прочитати повідомлення *M.*

Приміром, за комбінованого методу шифрування використовуються криптографічні ключі симетричних та асиметричних криптосистем. Вибір довжини ключів для кожного типу криптосистеми треба здійснювати в такий спосіб, щоби злочинникові було однаково важко атакувати будь-який механізм захисту комбінованої криптосистеми.

У таблиці 4.1 наведено поширені комбіновані довжини ключів для симетричних та асиметричних криптосистем, які надають однакову криптостійкість.

Таблиця 4.1 – Довжини ключів для симетричних та асиметричних криптосистем, які надають однакову криптостійкість

Довжина ключа симетричної криптосистеми, біт	Довжина ключа асиметричної криптосистеми, біт
56	384
64	512
80	768
112	1792
128	2304

Комбінований метод шифрування поєднує потужну швидкодію симетричного шифрування та високу криптостійкість, яку гарантують системи з відкритим ключем. Цей метод надає змогу виконувати процедуру автентифікації (перевірки чинності) повідомлення.

Контрольні питання з теми

1. Які недоліки асиметричних криптосистем?
2. Що таке комбінований метод зашифровування?
3. Які переваги використання комбінованого методу зашифровуванн

4.6 Проблема ідентифікації та автентифікації даних

З кожним об'єктом комп'ютерної системи (КС) пов'язана певна інформація, яка однозначно його ідентифікує. Це може бути число, рядок символів чи алгоритм, який ототожнює даний об'єкт. Цю інформацію називають ідентифікатором об'єкта.

Ідентифікація (ототожнювання) об'єкта – це перша функція підсистеми захисту, процедура, виконувана першою чергою, коли об'єкт здійснює спробу увійти до мережі.

Автентифікація (перевірка чинності, розпізнавання) об'єкта. Ця процедура встановлює, чи є даний об'єкт таким, яким він себе оголошує.

Після того коли проведено ідентифікацію та автентифікацію об'єкта, можна встановити сферу його дій та доступні для нього ресурси КС. Таку процедуру називають *авторизацією* (надаванням повноважень).

Згадані три операції ініціалізації є процедурами захисту й належать до одного об'єкта КС.

За обміну електронними документами мережею зв'язку істотно знижуються витрати на опрацьовування й зберігання документів, стає пришвидшеним їхній пошук. Але при цьому виникає проблема автентифікації автора документа й самого документа, тобто встановлення правочинності автора й відсутності змін у здобутому документі. У звичайній (паперовій) інформатиці ці проблеми розв'язуються за рахунок того, що інформація в документі й рукописний підпис автора жорстко пов'язані з фізичним носієм (папером). В електронних документах на машинних носіях такого зв'язку немає.

Метою автентифікації електронних документів є їхній захист від можливих зловмисних дій, якими є:

- активне перехоплювання – порушник, який долучився до мережі, перехоплює документи (файли) і змінює їх;
- маскарад-абонент *C* надсилає документ абонентові *B* від імені абонента *A*;
- ренегатство – абонент *A* заявляє, що не надсилав повідомлення абонентові *B*, хоча насправді й надсилав;
- підміна – абонент *B* змінює чи формує новий документ і заявляє, що одержав його від абонента *A*;
- повторювання – абонент *B* повторює раніше переданий документ, що його абонент *A* надсилав абонентові *B*.

Ці види зловмисних дій можуть завдати істотної шкоди банківським та комерційним структурам, державним підприємствам та організаціям, приватним особам, які застосовують у своїй діяльності комп'ютерні інформаційні технології.

Контрольні питання з теми

1.Що таке ідентифікація?

2. Що таке автентифікація?

3. Що таке авторизація?

4. Які зловмисні дії можливі при виконанні процедури автентифікації?

4.7 Алгоритми електронного цифрового підпису

При опрацьовуванні документів в електронній формі цілковито непридатні традиційні способи встановлювання чинності за рукописним підписом та відбитком печатки на паперовому документі. Розв'язанням проблеми підтвердження справжності документу є використання електронного цифрового підпису (ЕЦП).

Електронний цифровий підпис використовується для автентифікації текстів, які передаються комунікаційними каналами. Функційно він є аналогом звичайного рукописного підпису й має основні його переваги:

- засвідчує, що підписаний текст виходить від імені користувача, котрий поставив підпис;
- не надає саме цьому користувачеві можливості відмовитися від зобов'язань, пов'язаних з підписаним текстом;
- гарантує цілісність підписаного тексту.

Цифровий підпис являє собою відносно невелику кількість додаткової цифрової інформації, яка передається разом з підписуваним текстом.

Система ЕЦП включає дві процедури:

1 процедуру формування підпису;

2 процедуру перевірки підпису.

У процедурі постановки підпису використовується секретний ключ відправника повідомлення, у процедурі перевірки підпису – відкритий ключ відправника.

При формуванні ЕЦП відправник по-перше обчислює геш-функцію $h(M)$ тексту M, який він підписує. Обчислене значення геш-функції $h(M)$ являє собою один короткий блок інформації m, котра схарактеризує весь текст M у цілому. Потім число m зашифровується секретним ключем відправника. Здобувана при цьому пара чисел являє собою ЕЦП для тексту M.

При перевірці ЕЦП одержувач повідомлення знову обчислює геш-функцію $m = h(M)$ прийнятого каналом тексту M, після чого за допомогою відкритого ключа відправника перевіряє, чи відповідає здобутий підпис обчисленому значенню m геш-функції.

Засадничим моментом у системі ЕЦП є неможливість підробки ЕЦП користувача без знання його секретного ключа для підписування. В якості підписуваного документа може бути використано будь-який файл. Підписаний файл утворюється з непідписаного файла, до якого додається електронний підпис.

Кожний підпис містить таку інформацію:

- дата підпису;
- термін завершення дії ключа даного підпису;
- інформація про особу, котра підписала файл (П.І.Б., посада, коротке найменування фірми);
- ідентифікатор особи, котра поставила підпис;
- власне цифровий підпис.

Технологія застосовування системи ЕЦП припускає наявність мережі абонентів, котрі надсилають один одному підписані електронні документи. Для кожного абонента генерується пара ключів: секретний і відкритий. Секретний ключ зберігається абонентом у таємниці й використовується ним задля формування ЕЦП. Відкритий ключ є відомий всім іншим користувачам і призначений для перевірки ЕЦП одержувачем підписаного електронного документа. Інакше кажучи, відкритий ключ є необхідним інструментом, який дозволяє перевірити чинність електронного документа та автора підпису. Відкритий ключ не дозволяє обчислити секретного ключа.

Для генерування пари ключів (секретного й відкритого) в алгоритмах ЕЦП, як і в асиметричних системах шифрування, використовуються різні математичні схеми, грунтовані на застосовуванні однонапрямлених функцій. Ці схеми поділяються на дві групи. В підгрунті такого поділу лежать відомі складні обчислювальні завдання:

- факторизація (розкладання на множники) великих цілих чисел;
- дискретне логарифмування.

4.7.1 Алгоритм цифрового підпису за системою RSA

Першою й найбільш відомою в усьому світі системою ЕЦП стала система RSA, математичну схему якої, як було вже зазначено вище, розроблено 1977 року у Массачусетському технологічному інституті США.

Спочатку треба обчислити пару ключів (секретний ключ і відкритий ключ). Для цього відправник (автор) електронних документів обчислює два великих простих числа P та Q, потім знаходить їхній добуток

$$N = P \cdot Q$$

і значення функції Ейлера

$$\varphi(N) = (P-1)(Q-1).$$

Далі відправник обчислює число E з умов

$$E \le \varphi(N), \quad \text{НОД}(E, \varphi(N)) = 1$$

й число D з умов

$$D < N, \quad E \cdot D \equiv 1(\text{mod}\varphi(N)).$$

Пара чисел (E, N) є відкритим ключем. Цю пару чисел автор передає партнерам по листуванню для перевірки його цифрових підписів. Число D зберігається автором як секретний ключ для підписування.

Узагальнену схему формування й перевірки цифрового підпису RSA подано на рис. 4.2.

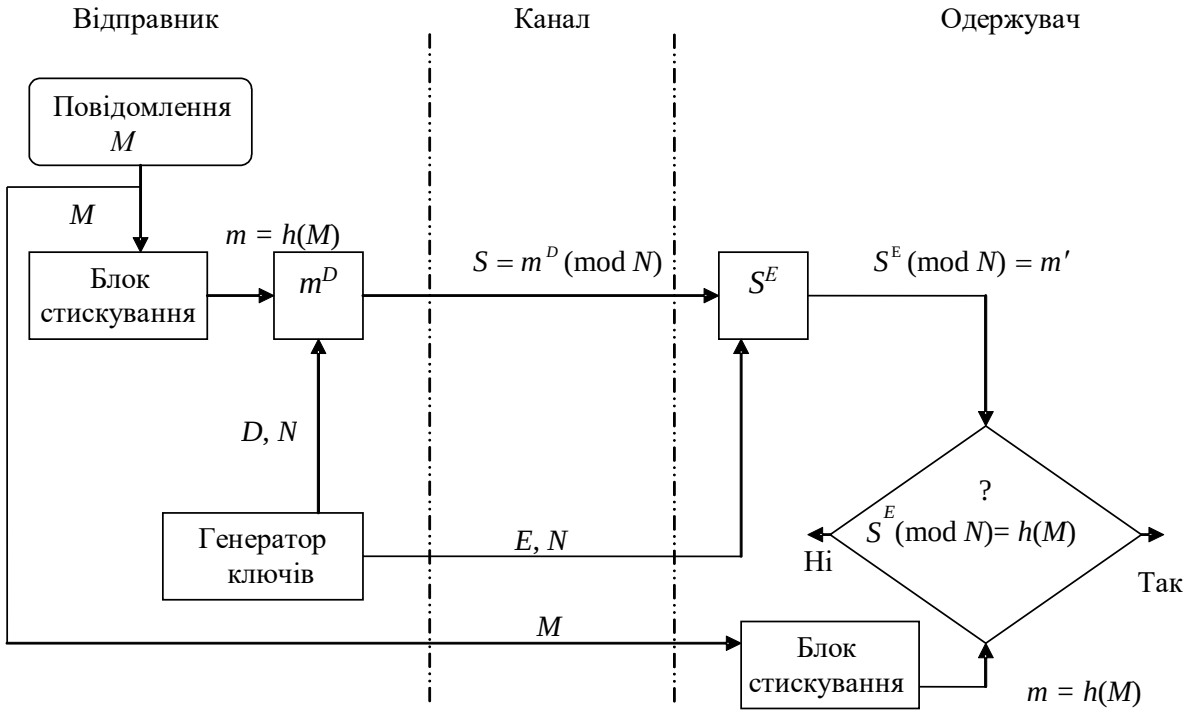


Рисунок 4.2 – Узагальнена схема цифрового підпису RSA

Припустімо, що відправник хоче підписати повідомлення M перед його відправленням. Спочатку повідомлення M (блок інформації, файл; таблиця) стискають за допомогою геш-функції $h(\cdot)$ в ціле число m:

$$m = h(M).$$

Потім обчислюють цифровий підпис S під електронним документом M, використовуючи геш-значення m та секретний ключ D:

$$S = m^D (\text{mod } N).$$

Пара (*M, S*) передається партнерові-одержувачеві як електронний документ *M*, підписаний цифровим підписом *S*, причому підпис *S* сформовано власником секретного ключа *D*.

Після прийняття пари (*M, S)* одержувач обчислює геш-значення повідомлення *M* двома різними способами. Насамперед він відновлює геш-значення *m*, застосовуючи криптографічне перетворювання підпису *S* з використовуванням відкритого ключа *E*:

$$m' = S^E (\text{mod} N).$$

Окрім того, він віднаходить результат прийнятого повідомлення *M* за допомогою такої самої геш-функції $h(\cdot)$:

$$m = h(M).$$

Якщо виконується рівність обчислених значень, тобто

$$S^E (\text{mod } N) = h(M),$$

то одержувач визнає пару (*M, S*) чинною. Доведено, що лише власник секретного ключа *D* може сформувати цифровий підпис *S* з документа *M*, а визначити секретне число *D* завдяки відкритому числу *E* є не легше, аніж розкласти модуль *N* на множники.

Окрім того, можна строго математично довести, що результат перевірки цифрового підпису *S* буде позитивним лише в тому разі, якщо при обчислюванні *S* було використано секретний ключ *D*, який відповідає відкритому ключу *E*. Тому відкритий ключ *E* іноді називають ідентифікатором особи, котра поставила підпис.

Недоліки алгоритму цифрового підпису за системою RSA.

1 При обчислюванні модуля *N*, ключів *E* та *D* для системи цифрового підпису RSA доводиться перевіряти велику кількість додаткових умов, що практично зробити важко. Невиконання кожної з цих умов уможливлює фальсифікацію цифрового підпису з боку того, хто виявить таке невиконання. Під час підписування важливих документів не можна припускати такої можливості навіть гіпотетично.

2 Задля забезпечення криптостійкості цифрового підпису RSA стосовно спроб фальсифікування на рівні, приміром, алгоритму DES, на шифрування інформації, тобто 10^{18}, необхідно використовувати при обчислюваннях *N*, *D* та *E* цілі числа не менш за 2^{512} (або близько до 10^{154}) кожне, що потребує великих обчислювальних витрат, які перевищують на 20...30% обчислювальні витрати

інших алгоритмів цифрового підпису при зберіганні того самого рівня криптостійкості.

3 Цифровий підпис RSA є вразливий щодо так званої мультиплікативної атаки. Інакше кажучи, алгоритм цифрового підпису RSA дозволяє злочинникові без знання секретного ключа D сформувати підписи під тими документами, в яких результат гешування можна обчислити як добуток результатів гешування, вже підписаних документів.

Приклад 4.3 Припустімо, що злочинник може сконструювати три повідомлення – M_1, M_2 та M_3,– в яких геш-значення є

$$m_1 = h(M_1),\ \ m_2 = h(M_2),\ \ m_3 = h(M_3),$$

причому $m_3 = m_1 \cdot m_2 (\text{mod } N)$.

Припустімо також, що для двох повідомлень – M_1 та M_2 – здобуто чинні підписи

$$S_1 = m_1^D (\text{mod } N) \quad \text{та} \quad S_2 = m_2^D (\text{mod } N).$$

Тоді злочинник може легко обчислити підпис S_3 для документа M_3, навіть не знаючи секретного ключа D:

$$S_3 = S_1 \cdot S_2 (\text{mod } N).$$

І справді,

$$S_1 \cdot S_2 (mod\ N) = m_1^D \cdot m_2^D (mod\ N) = (m_1 m_2)^D (mod\ N) = m_3^D (mod\ N) = S_3.$$

Більш надійний та зручний для реалізації на персональних комп'ютерах алгоритм цифрового підпису було розроблено 1984 року американцем арабського походження Тахером Ель Гамалем. 1991року НІСТ США обґрунтував перед комісією Конгресу США вибір алгоритму цифрового підпису Ель Гамаля як підгрунтя для національного стандарту.

4.7.2 Алгоритм цифрового підпису Ель Гамаля

Назва EGSA походить від слів El Gamal Signature Algorithm (алгоритм цифрового підпису Ель Гамаля). Ідею за системою EGSA базовано на тім, що для обґрунтування практичної неможливості фальсифікації цифрового підпису може бути використано більш складне обчислювальне завдання, чим розкладання на множники великого цілого числа, – завдання дискретного логарифмування. Окрім того, Ель Гамалю вдалося уникнути недоліку алгоритму цифрового підпису RSA, пов'язаного з можливістю підробки цифрового підпису під певними повідомленнями без знання секретного ключа.

Розглянемо докладніше алгоритм цифрового підпису Ель Гамаля. Для того щоби генерувати пару ключів (відкритий ключ – секретний ключ), спочатку

обирають певне велике просте ціле число P й велике ціле число G, причому $G < P$. Відправник і одержувач підписаного документа використовують при обчислюваннях однакові великі цілі числа P (~10^{308}, чи ~ 2^{1024}) та G (~10^{154}, чи ~2^{512}), які не є секретними.

Відправник обирає випадкове ціле число X, $1 < X \leq (P - 1)$, і обчислює

$$Y = G^x \bmod P.$$

Число Y є відкритим ключем, використовуваним для перевірки підпису відправника. Число Y відкрито передається всім потенційним одержувачам документів.

Число X є секретним ключем відправника для підписування документів і має зберігатися в секреті.

Для того щоб підписати повідомлення M, спочатку відправник гешує його за допомогою геш-функції $h(\cdot)$ в ціле число m:

$$m = h(M), \ 1 < m < (P - 1)$$

і генерує випадкове ціле число K, $1< K < (P - 1)$, таке що K і $(P - 1)$ є взаємно простими. Потім відправник обчислює ціле число a:

$$a = G^K \bmod P$$

і, застосовуючи розширений алгоритм Евкліда, обчислює за допомогою секретного ключа X ціле число b з рівняння

$$m = X \cdot a + K \cdot b(\bmod(P - 1)).$$

Пара чисел (a, b) є цифровий підпис

$$S = (a, b),$$

який ставиться під документом M.

Три числа (M, a, b) передаються одержувачеві, тоді як пара чисел (X, K) тримається в секреті.

Після прийняття підписаного повідомлення (M, a, b) одержувач повинен перевірити, чи відповідає підпис $S = (a, b)$ повідомленню M. Для цього одержувач спочатку обчислює за прийнятим повідомленням M число

$$m = h(M),$$

тобто гешує прийняте повідомлення M.

Потім одержувач обчислює значення

$$A = Y^a a^b (\bmod P)$$

і визнає повідомлення M чинним, лише якщо

$$A = G^m (\mathrm{mod}\ P).$$

Інакше кажучи, одержувач перевіряє слушність співвідношення

$$Y^a a^b (\mathrm{mod}\ P) = G^m (\mathrm{mod}\ P).$$

Можна строго математично довести, що остання рівність буде виконуватися тоді, й тільки тоді, коли підпис $S = (a, b)$ під документом M здобуто за допомогою саме того секретного ключа X, з якого було здобуто відкритий ключ Y. Отже, можна надійно впевнитися, що відправник повідомлення M був власник саме даного секретного ключа X, не розкриваючи при цьому сам ключ, і що відправник підписав саме цей конкретний документ M.

Слід зазначити, що обрання кожного підпису за допомогою методу Ель Гамаля потребує нового значення K, причому це значення має обиратися випадково. Якщо злочинник розкриє коли-небудь значення ключа K, вдруге використовуваного відправник, то він зможе розкрити секретний ключ X відправника.

Приклад 4.4 Оберемо числа $P = 11$, $G = 2$ та секретний ключ $X = 8$. Обчислимо значення відкритого ключа:

$$Y = G^x \bmod P = Y = 2^8 \bmod 11 = 3.$$

Припустімо, що вихідне повідомлення M характеризується геш-значенням $m = 5$.

Для того щоби обчислити цифровий підпис для повідомлення M, яке має геш-значення $m = 5$, спочатку оберемо випадкове ціле число $K = 9$. Переконаємося, що числа K та $(P – 1)$ є взаємно простими. І справді,

$$\text{НОД } (9, 10) = 1.$$

Далі обчислюємо елементи a й b підпису:

$$a = G^K \bmod P = 2^9 \bmod 11 = 6,$$

елемент b визначаємо, використовуючи розширений алгоритм Евкліда:

$$m = X \cdot a + K \cdot b (\mathrm{mod}(P - 1)).$$

За $m = 5$, $a = 6$, $X = 8$, $K = 9$, $P = 11$ дістаємо

$$5 = (6 \cdot 8 + 9 \cdot b)(\mathrm{mod}\ 10)$$

або

$$9 \cdot b \equiv -43 (\mathrm{mod}\ 10).$$

Розв’язання: b = 3. Цифровий підпис являє собою пару: a = 6, b = 3. Далі відправник передає підписане повідомлення. Прийнявши підписане повідомлення і відкритий ключ Y = 3, одержувач обчислює геш-значення для повідомлення M: m = 5, а потім обчислює два числа:

1) $Y(\text{mod }) = 3^6 \cdot 6^3 (\text{mod } 11) = 10 \text{ (mod } 11)$;
2) $G^m \text{ (mod P)} = 2^5 \text{ (mod } 11) = 10 \text{ (mod } 11)$.

Позаяк ці два цілих числа є рівні, прийняте одержувачем повідомлення визнається за чинне.

Схема цифрового підпису Ель Гамаля має низку переваг порівняно зі схемою цифрового підпису RSA:

1 за заданого рівня стійкості алгоритму цифрового підпису цілі числа, які беруть участь в обчислюваннях, мають запис на 25 % коротше, що зменшує складність обчислювань майже удвічі й дозволяє помітно скоротити обсяг використовуваної пам’яті;

2 за вибору модуля P достатньо перевірити, що це число є простим і що в числі (P – 1) є великий простий множник (тобто всього дві перевірюваних умови);

3 процедура формування підпису за схемою Ель Гамаля не дозволяє обчислювати цифрові підписи під новими повідомленнями без знання секретного ключа (як в RSA).

Однак алгоритм цифрового підпису Ель Гамаля має й певні недоліки порівняно зі схемою підпису RSA. Зокрема, довжина цифрового підпису виходить в 1,5 рази більше, що, своєю чергою, збільшує час її обчислювання.

4.7.3 Алгоритм цифрового підпису DSA

Алгоритм цифрового підпису DSA (Digital Signature Algorithm) запропоновано 1991 року у США для використовування в стандарті цифрового підпису DSS (Digital Signature Standard). Алгоритм DSA є розвиненням алгоритмів цифрового підпису Ель Гамаля й К. Шнорра.

Відправник та одержувач електронного документа використовують при обчислюванні великі цілі числа: G і P – прості числа, L бітів кожне ($512 \le L \le 1024$); q – просте число довжиною 160 біт (дільник числа (P – 1)). Числа G, P, q є відкритими й можуть бути спільними для всіх користувачів мережі.

Відправник вибирає випадкове ціле число X, $1 < X < q$. Число X є секретним ключем відправник для формування електронного цифрового підпису.

Потім відправник обчислює значення

$$Y = G^x \bmod P`.$$

Число Y є відкритим ключем для перевірки підпису відправника. Число Y передається всім одержувачам документів.

Цей алгоритм також передбачає використовування однонапрямленої функції гешування $h(M)$. У стандарті DSS визначено алгоритм безпечного гешування SHA (Secure Hash Algorithm).

Для того щоби підписати документ M, відправляч гешує його в ціле геш-значення m:

$$m = h(M),\ 1< m < q,$$

потім генерує випадкове ціле число K, $1< K< q$ і обчислює число

$$r = (G^K \bmod P) \bmod q.$$

Потім відправник обчислює за допомогою секретного ключа X ціле число

$$s = \frac{m + r \cdot X}{K} \bmod q.$$

Пара чисел r та s є цифровий підпис

$$S = (r, s)$$

під документом M.

Отже, підписане повідомлення являє собою три числа –(M, r, s).

Одержувач підписаного повідомлення (M, r, s) перевіряє виконання умов

$$0 < r < q, \qquad 0 < s < q$$

й відкидає підпис, якщо хоча б одну з цих умов не виконано.

Потім одержувач обчислює значення

$$W = \frac{1}{s} \bmod q,$$

геш-значення

$$m = h(M)$$

і числа

$$u_1 = (m \cdot w) \bmod q,$$

$$u_2 = (r \cdot w) \bmod q\,.$$

Далі одержувач за допомогою відкритого ключа Y обчислює значення

$$V = ((G^{u_1} \cdot Y^{u_2}) \bmod P) \bmod q$$

і перевіряє виконання умови

$$V = r.$$

Якщо умова $V = r$ виконується, тоді підпис $S = (r, s)$ під документом M визнається одержувачем за чинний.

Можна математично довести, що остання рівність виконуватиметься тоді, й лише тоді, коли підпис $S = (r, s)$ під документом M здобуто за допомогою саме того секретного ключа X, з якого було здобуто відкритий ключ Y. Отже, можна надійно впевнитися, що відправник повідомлення володіє саме даним секретним ключем X (не розкриваючи при цьому значення ключа X) і що відправник підписав саме документ M.

Порівняно з алгоритмом цифрового підпису Ель Гамаля алгоритм DSA має такі основні переваги:

1 за будь-якого припустимого рівня стійкості, тобто за будь-якої пари чисел G та P (від 512 до 1024 біт), числа q, X, r, s мають довжину по 160 біт, скорочуючи довжину підпису до 320 біт;

2 більшість операцій з числами K, r, s, X при обчислюванні підпису виконується за модулем числа q довжиною 160 біт, що скорочує час даної операції;

3 при перевірці підпису більшість операцій з числами u_1, u_2, V, W також виконується за модулем числа q довжиною 160 біт, що скорочує обсяг пам'яті й час обчислювання.

Недоліком алгоритму DSA є те, що при підписуванні й при перевірці підпису доводиться виконувати складні операції розподілу за модулем q:

$$s = \frac{m + rX}{K} \pmod q, \quad w = \frac{1}{s} \pmod q,$$

що не дозволяє набувати максимальної швидкодії.

Слід зазначити, що виконання алгоритму DSA може бути прискорене за допомогою виконання попередніх обчислювань. Значення r не залежить від повідомлення M і його геш-значення m. Можна заздалегідь утворити рядок випадкових значень K і потім для кожного з цих значень обчислити значення r. Можна також заздалегідь обчислити обернені значення K^{-1} для кожного значення K. Потім, по надходженні повідомлення M, можна обчислити значення s для даних значень r та K^{-1}. Ці попередні обчислювання значно прискорюють роботу алгоритму DSA.

4.7.4 Стандарт електронного підпису ДСТУ4145-2002

Стандарт ДСТУ 4145-2014 встановлює механізм електронного підписування, що базується на властивостях груп точок еліптичних кривих над

полями $GF(2^m)$, та правила застосування цього механізму до повідомлень, що пересилаються каналами зв'язку та/або обробляються у комп'ютеризованих системах загального призначення. Застосування цього стандарту гарантує цілісність підписаного повідомлення, автентичність його автора та неспростовність авторства.

Електронний підпис створюється за допомогою операцій над точками еліптичної кривої після хешування повідомлення. В якості геш-функції використовується стандарт України ДСТУ 7564:2014 або Міждержавний стандарт ГОСТ 34.311-95, геш-образ якого дорівнює 256 бітів.

Відкритими параметрами стандарту є еліптична крива, базова точка кривої P з відомим простим порядком n. Відкритим ключем підписанта є точка еліптичної кривої Q, $Q=-dP$, його секретним ключем є число d.

Електронний підпис формується абонентом A за допомогою його секретного ключа, перевірка підпису здійснюється абонентом B з використанням відкритого ключа абонента A.

Контрольні питання з теми

1. Що таке електронний цифровий підпис, які його переваги?
2. Які процедури включає ЕЦП?
3. Що таке геш-функція?
4. Яку інформацію містить ЕЦП?
5. Які математичні схеми використовуються для генерації пари ключів в алгоритмах ЕЦП?
6. Опишіть алгоритм ЕЦП на основі RSA?
7. Які недоліки ЕЦП за системою RSA?
8. Опишіть алгоритм ЕЦП на основі Ель Гамаля?
9. Які переваги ЕЦП Ель Гамаля порівняно з ЕЦП RSA?
10. Опишіть алгоритм ЕЦП DSA?
11. Опишіть алгоритм ЕЦП ДСТУ 4145-2002.
12. Дайте визначення поля.

4.8 Керування криптографічними ключами

Окрім вибору криптографічної системи, найважливішою є проблема керування ключами.

Ключова інформація – це сукупність усіх ключів, які задіяно в інформаційній системі (ІС). Якщо надійне керування ключовою інформацією не

забезпечено, то при її перехоплюванні злочинник матиме необмежений доступ до всієї інформації.

Керування ключами – це інформаційний процес, який містить в собі три елемента:

- генерування ключів;
- накопичування ключів;
- розподілення ключів.

Розглянемо, як вони мають реалізовуватися для того, щоби забезпечити безпеку ключової інформації в ІС.

4.8.1 Генерування ключів

Перш за все треба зазначити, що не слід використовувати випадкові ключі, щоби їх було легше запам'ятати.

В ІС використовують спеціальні апаратні та програмні засоби генерування випадкових ключів. Зазвичай застосовують давачі псевдовипадкових чисел (ПВЧ). Однак ступінь випадковості їхнього генерування має бути надто великим. Ідеальними генераторами є пристрої на базі „натуральних" випадкових процесів. В ІС із середніми вимогами щодо захищеності використовуються програмні генератори ключів, які обчислюють ПВЧ як функцію від часу, та (чи) числа, які вводяться користувачем.

4.8.2 Накопичування ключів

Накопичування ключів – це організація процесу зберігання, обліку та вилучання ключів.

Ключ – це найбажаніший об'єкт для злочинника, позаяк він надає змогу набути доступу до конфіденційної інформації. Тому питанням накопичування ключів слід приділяти особливу увагу.

Секретні ключі завжди треба зберігати у зашифрованому вигляді.

Ключі, які зашифровують ключову інформацію, називаються *майстер-ключами.* Треба, щоби майстер-ключі кожен користувач знав напам'ять і не зберігав їх на будь-яких носіях.

Надто важливо задля забезпечування безпеки інформації, щоби ключова інформація в ІС регулярно змінювалася. При цьому змінюватися мають як звичайні ключі, так і майстер-ключі.

4.8.3 Розподілення ключів

Розподілення ключів – найвідповідальніший процес керування ключами. До нього пред'являють такі вимоги:

1 оперативність та точність розподілення;

2 таємність розподілення ключів.

Розподілення ключів поміж користувачами реалізовується двома методами:

1 *шляхом побудови одного чи кількох центрів розподілення ключів* (*ЦРК*). Недоліком є те, що в ЦРК відомо, кому й які ключі призначено. Це надає змогу читати усі повідомлення, які є в ІС, що впливає на захист системи від злочинних дій;

2 *прямий обмін ключами* поміж користувачами ІС. Тут проблема полягає в тому, щоби ідентифікувати та перевірити правочинність користувачів.

В обох випадках треба, щоби було гарантовано правочинність сеансу зв'язку. Це можна забезпечити двома способами:

1 м*еханізм запиту–відповіді*. Якщо користувач *A* хоче бути впевненим, що повідомлення, які він одержує від користувача *B*, не є спотвореними, він додає до повідомлення певний елемент, який є випадковим (запит). При відповіді користувач *B* має виконати певну операцію з цим елементом (приміром, додати 1). Це не можна зробити раніше, тому що невідомо, яке саме випадкове число буде в запиті. Після того як відповідь, яка містить результати дій, одержано, користувач *A* може бути впевненим, що сеанс зв'язку є правочинним. Недоліком цього методу є змога встановляння закономірності поміж запитом та відповіддю;

2 *механізм позначання часу* (*"часовий штемпель"*). Суть полягає в тому, що фіксується час кожного повідомлення. В цьому разі кожний користувач ІС може дізнаватися, коли надійшло повідомлення.

В обох випадках слід використовувати шифрування, щоби бути впевненим, що відповідь надіслав не злочинник і штемпель позначання часу не змінено.

При використовуванні механізму позначання часу є проблема припустимого часу інтервалу затримки задля потвердження чинності сеансу. Повідомлення, котре має "часовий штемпель", не може бути одержано миттєво. Окрім того, годинники, які є у комп'ютерах передавача та отримувача, не можуть бути абсолютно синхронними. Тому в реальних ІС, приміром, в системах платежів кредитних карток, використовується саме механізм встановлення чинності та захисту від підробок. Інтервал позначання часу становить від однієї до кількох хвилин. Багато відомих способів крадіжок електронних грошей базується на "вклинюванні" в цей проміжок часу помилкових запитів при зніманні грошей.

Для обміну ключами можна використовувати криптосистеми з відкритим ключем за допомогою алгоритму RSA.

Проте надто ефективним став алгоритм Диффі–Хеллмана, який надає змогу двом користувачам, без посередників, обмінюватися ключем, який може потім використовуватися для симетричного шифрування.

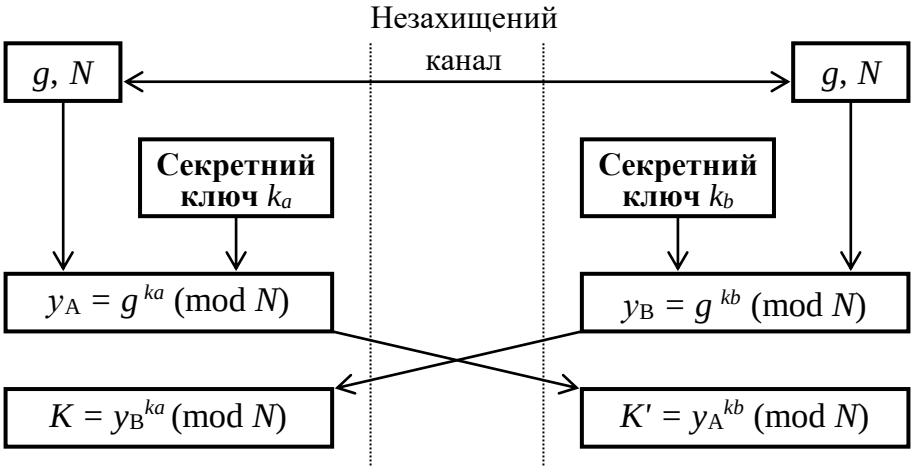
Отже, керування ключами є зведенням до пошуку такого протоколу розподілення ключів, який забезпечив би:

- змогу відмовитися від центра розподілення ключів;
- взаємне потвердження правочинності користувачів;
- потвердження правочинності сеансу за допомогою механізму позначання часу механізмом, використовування для цього програмних чи апаратних засобів;
- використовування при обміні ключами мінімальної кількості повідомлень.

4.8.3.1 Алгоритм розподілювання ключів Диффі–Хеллмана

Метод Диффі–Хеллмана, запропонований 1976 року, базується на використовуванні системи відкритого розподілювання ключів. Ця система надає змогу користувачам обмінюватися ключами на незахищених каналах зв'язку (алгоритм Диффі–Хеллмана є чинний на лініях зв'язку, які надійно захищаються від модифікування. У випадках, коли в каналі є ймовірність модифікування даних, виникає змога "вклинювання" до процесу генерування ключів "злочинника-посередника"). Вона базується на тих самих засадах, що й система шифрування з відкритим ключем. Безпека системи зумовлюється складністю обчислювання дискретних логарифмів.

Припустімо, що ЦРК є відсутній та учасники інформаційного обміну визначають значення ключів згідно з таким протоколом.



$$K = K' \equiv g^{ka\ kb} \pmod{N}$$

Рисунок 4.3 – Схема реалізації алгоритму Диффі–Хеллмана

Користувачі A та B обирають модуль N, який має бути простим числом та примітивним елементом $g \in Z_N$, $(1 \le g \le N - 1)$, котрий утворює усі ненульові елементи множини Z_N, чи

$$\{g, g^2, \ldots, g^{N-1} = 1\} = Z_N - \{0\}.$$

Цілі числа N та g держати в таємниці необов'язково. Зазвичай вони є спільними для усіх користувачів – учасників інформаційного обміну мережі.

Потім користувачі A та B, незалежно один від одного, обирають власні секретні ключі k_A та k_B (k_A та k_B – випадкові цілі великі числа, які зберігаються користувачами в таємниці).

Далі користувач A обчислює відкритий ключ

$$y_A \equiv g^{kA} \pmod{N},$$

а користувач B – відкритий ключ

$$y_B \equiv g^{kB} \pmod{N}.$$

Потім A та B обмінюються обчисленими значеннями ключів y_A та y_B по незахищеному каналу.

Далі користувачі обчислюють спільний секретний ключ за допомогою рівнянь:

Сторона A $\quad K = (y_B)^{kA} \equiv (g^{kB})^{kA} \pmod{N}$,

Сторона B $\quad K' = (y_A)^{kB} \equiv (g^{kA})^{kB} \pmod{N}$.

При цьому $K = K'$, тому що

$$(g^{kB})^{kA} = (g^{kA})^{kB}.$$

Ключ K може використовуватися в якості спільного секретного ключа в симетричній криптосистемі.

Окрім того, сторони можуть використовувати ключ K для несиметричного зашифровування згідно з правилом

$$C = E_K(M) = M^K \pmod{N}.$$

З цією метою слід віднайти ключ, призначений для розшифровування K^* із співвідношення

$$K \cdot K^{-1} \equiv 1 \pmod{N\text{-}1}.$$

Це дозволить відтворювати повідомлення за правилом

$$M = D_K(C) = C^{K-1} \pmod{N}.$$

Проте слід зауважити, що останній алгоритм матиме неоптимальні характеристики щодо криптостійкості, тому що ключ K не завжди задовольнятиме вимогам і тому процедуру частіш за все треба буде повторювати.

Істотний вплив на безпеку системи справляє вибір значень N та g. Модуль N має бути великим простим числом. Число $(N – 1)/2$ також має бути простим.

При використовуванні алгоритму Диффі–Хеллмана, слід пам'ятати про гарантування того, що користувач A одержав відкритий ключ саме від користувача B й навпаки, тобто необхідно, щоби під одержаними повідомленнями були цифрові підписи.

Метод Диффі–Хеллмана надає змогу забезпечувати кожний сеанс зв'язку новими ключами й не зберігати їх будь-де.

Окрім того, генерування ключів методом Диффі–Хеллмана здійснюється в сотні разів швидше порівняно з алгоритмом RSA.

Приклад 4.5 Нехай

$$p = 13,\ a = 7,\ m = 3,\ k = 4.$$

Відкритий ключ, передаваний стороною A, є

$$y_A \equiv 7^3 (\text{mod } 13) \equiv 343 (\text{mod } 13) \equiv 5.$$

Відкритий ключ, передаваний стороною B, є

$$y_B \equiv 7^4 (\text{mod } 13) \equiv 2401 (\text{mod } 13) \equiv 9.$$

Секретне число, обчислюване сторонами, є

$$K \equiv 5^4 (\text{mod } 13) \equiv 625 (\text{mod } 13) \equiv 1,$$

$$K' \equiv 9^3 (\text{mod } 13) \equiv 729 (\text{mod } 13) \equiv 1.$$

4.8.3.2 Переваги та недоліки алгоритму Диффі–Хеллмана

Диффі й Хеллман внесли пропозицію використовувати для створювання криптографічних систем з відкритим ключем *функцію дискретного піднесення до степеня*.

Необоротність перетворювання в цьому разі забезпечується тим, що легко обчислити показникову функцію в кінцевому полі Галуа, яке складається з p елементів (p – просте число чи просте в будь-якому степені).

Обчислювання логарифмів в таких полях – більш трудомістка операція. Приміром, якщо для прямого перетворювання 1000-бітових простих чисел треба

2000 операцій, то для оберненого перетворювання (обчислювання логарифму в полі Галуа) треба біля 10^{30} операцій.

Даний алгоритм відрізнюється від алгоритму RSA тим, що не дає змоги шифрувати саме інформацію. Іншим недоліком алгоритма Диффі–Хеллмана є відсутність гарантованої нижньої оцінки трудомісткості зламування ключа. Окрім того, залишається проблема автентифікації, тому що є небезпека імітування користувача.

4.8.1 Завдання для самоперевірки з теми

Завдання 4.8.1 Привести схему генерування та розподілення ключа методом Диффі-Хеллмана. Згенерувати спільний секретний ключ методом Диффі-Хеллмана, якщо варіантом є номер студента в журналі групи.

1 варіант N=43; g=12; $к_a$=15; $к_b$=19;
2 варіант N=37; g=12; $к_a$=15; $к_b$=19;
3 варіант N=23; g=11; $к_a$=14; $к_b$=21;
4 варіант N=29; g=11; $к_a$=14; $к_b$=21;
5 варіант N=31; g=11; $к_a$=14; $к_b$=21;
6 варіант N=37; g=11; $к_a$=14; $к_b$=21;
7 варіант N=43; g=11; $к_a$=14; $к_b$=21;
8 варіант N=23; g=12; $к_a$=15; $к_b$=19;
9 варіант N=29; g=12; $к_a$=15; $к_b$=19;
10 варіант N=31; g=12; $к_a$=15; $к_b$=19.

Контрольні питання з теми

1. Що таке ключова інформація?
2. Що таке процес керування ключами?
3. Опишіть процес генерування ключів.
4. Опишіть процес накопичування ключів.
5. Що таке майстер-ключ?
6. Які вимоги пред’являються до процесу розподілення ключів?
7. Назвіть методи розподілення ключів.
8. Якими засобами гарантується правочинність сеансу зв’язку?
9. Що повинен забезпечувати протокол розподілення ключів?
10. Назвіть методи розв’язання задачі дискретного логарифмування.
11. Опишіть схему Диффі-Хеллмана генерування спільного секретного ключа.
12. На чому заснована криптографічна стійкість схеми Диффі-Хеллмана?

13. Як впливає розмір простого числа N на криптостійкість схеми Диффі-Хеллмана?

14. Як визначити бітовий розмір спільного ключа по вхідних відкритих параметрах?

15. Які переваги та недоліки алгоритму Диффі–Хеллмана ?

ДОДАТОК А

Таблиця Віженера для української абетки

Ключ	**а**	**б**	**в**	**г**	**д**	**е**	**є**	**ж**	**з**	**и**	**і**	**ї**	**й**	**к**	**л**	**м**	**н**	**о**	**п**	**р**	**с**	**т**	**у**	**ф**	**х**	**ц**	**ч**	**ш**	**щ**	**ь**	**ю**	**я**
0	а	б	в	г	д	е	є	ж	з	и	і	ї	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ю	я
1	б	в	г	д	е	є	ж	з	и	і	ї	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ю	я	а
2	в	г	д	е	є	ж	з	и	і	ї	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ю	я	а	б
3	г	д	е	є	ж	з	и	і	ї	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ю	я	а	б	в
4	д	е	є	ж	з	и	і	ї	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ю	я	а	б	в	г
5	е	є	ж	з	и	і	ї	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ю	я	а	б	в	г	д
6	є	ж	з	и	і	ї	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ю	я	а	б	в	г	д	е
7	ж	з	и	і	ї	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ю	я	а	б	в	г	д	е	є
8	з	и	і	ї	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ю	я	а	б	в	г	д	е	є	ж
9	и	і	ї	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ю	я	а	б	в	г	д	е	є	ж	з
10	і	ї	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ю	я	а	б	в	г	д	е	є	ж	з	и
11	ї	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ю	я	а	б	в	г	д	е	є	ж	з	и	і
12	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ю	я	а	б	в	г	д	е	є	ж	з	и	і	ї
13	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ю	я	а	б	в	г	д	е	є	ж	з	и	і	ї	й
14	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ю	я	а	б	в	г	д	е	є	ж	з	и	і	ї	й	к
15	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ю	я	а	б	в	г	д	е	є	ж	з	и	і	ї	й	к	л
16	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ю	я	а	б	в	г	д	е	є	ж	з	и	і	ї	й	к	л	м
17	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ю	я	а	б	в	г	д	е	є	ж	з	и	і	ї	й	к	л	м	н
18	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ю	я	а	б	в	г	д	е	є	ж	з	и	і	ї	й	к	л	м	н	о
19	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ю	я	а	б	в	г	д	е	є	ж	з	и	і	ї	й	к	л	м	н	о	п
20	с	т	у	ф	х	ц	ч	ш	щ	ь	ю	я	а	б	в	г	д	е	є	ж	з	и	і	ї	й	к	л	м	н	о	п	р
21	т	у	ф	х	ц	ч	ш	щ	ь	ю	я	а	б	в	г	д	е	є	ж	з	и	і	ї	й	к	л	м	н	о	п	р	с
22	у	ф	х	ц	ч	ш	щ	ь	ю	я	а	б	в	г	д	е	є	ж	з	и	і	ї	й	к	л	м	н	о	п	р	с	т
23	ф	х	ц	ч	ш	щ	ь	ю	я	а	б	в	г	д	е	є	ж	з	и	і	ї	й	к	л	м	н	о	п	р	с	т	у
24	х	ц	ч	ш	щ	ь	ю	я	а	б	в	г	д	е	є	ж	з	и	і	ї	й	к	л	м	н	о	п	р	с	т	у	ф
25	ц	ч	ш	щ	ь	ю	я	а	б	в	г	д	е	є	ж	з	и	і	ї	й	к	л	м	н	о	п	р	с	т	у	ф	х
26	ч	ш	щ	ь	ю	я	а	б	в	г	д	е	є	ж	з	и	і	ї	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц
27	ш	щ	ь	ю	я	а	б	в	г	д	е	є	ж	з	и	і	ї	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч
28	щ	ь	ю	я	а	б	в	г	д	е	є	ж	з	и	і	ї	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш
29	ь	ю	я	а	б	в	г	д	е	є	ж	з	и	і	ї	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ
30	ю	я	а	б	в	г	д	е	є	ж	з	и	і	ї	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь
31	я	а	б	в	г	д	е	є	ж	з	и	і	ї	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ю

Навчальне видання

Лариса Григорівна ЙОНА
Олексій Віталійович ОНАЦЬКИЙ
Юлія Володимирівна БЄЛОВА

КРИПТОГРАФІЧНИЙ ЗАХИСТ ІНФОРМАЦІЇ

Навчальний посібник

Редактор