МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Міжнародний Гуманітарний університет

Кафедра комп'ютерної інженерії та інноваційних технологій

О.В. ОНАЦЬКИЙ, Л.Г. ЙОНА

КРИПТОГРАФІЧНІ СИСТЕМИ

Навчальний посібник
з дисциплін
«Криптографія та криптоаналіз»

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

Одеса
2023

Автори:

Онацький О. В. – кандидат технічних наук, доцент кафедри комп'ютерної інженерії та інноваційних технологій.

Йона Л. Г. – кандидат технічних наук, доцент кафедри комп'ютерної інженерії та інноваційних технологій.

е -Вид.2-ге, випр., доп. - Одеса:Міжнародний гуманітарний університет, 2023. 156 с.

Навчальний посібник призначено для студентів, котрі вивчають дисципліни «Криптографія та криптоаналіз». Містить систематичне викладення наукових основ від найпростіших прикладів та основних понять до сучасних криптографічних концепцій. Спрямовано на вивчення симетричних криптосистем та криптосистеми з відкритим ключем.

ЗМІСТ

ПЕРЕДМОВА

Дисципліни «Криптографія та криптоаналіз» впроваджені до учбового плану освітньо-професійної програми підготовки бакалаврів з напряму знань 12 «Інформаційні технології» за спеціальністю 125 «Кібербезпека»

Мета навчальної дисципліни «Криптографія та криптоаналіз» – формування у майбутніх фахівців умінь та компетенції в галузі теоретичної криптології: аналізувати побудови симетричних та асиметричних криптографічних систем, вибирати алгоритми шифрування для захисту інформаційних ресурсів у системах від порушення її конфіденційності, цілісності та доступності, вибирати режими роботи блочних алгоритмів шифрування для захисту інформації, виконувати розрахунки необхідних параметрів алгоритмів шифрування, виконувати розрахунки для схем електронного підпису (ЕП), перевірку ЕП та керування криптографічними ключами; спроможності досліджувати та аналізувати загрози безпеці криптографічних протоколів, алгоритмів шифрування; використовування формальних методів аналізу криптографічних протоколів; криптоаналіз несиметричних та симетричних криптосистем шифрування.

Дисципліна забезпечує студентів базовими знаннями з захисту інформаційних ресурсів у системах від порушення її конфіденційності, цілісності та доступності; принципів побудови симетричних та асиметричних криптографічних систем; режимів роботи блочних алгоритмів шифрування; концепції криптосистем з відкритим ключем; методів побудови схем електронних підписів; керування криптографічними ключами; здатності вирішувати завдання аналізу та розрахунків параметрів алгоритмів шифрування.

Теоретичний матеріал відповідає навчальній програмі дисципліни «Криптографія та криптоаналіз» і супроводжується достатньою кількістю наведених типових прикладів, контрольних запитань і задач та забезпечує можливість здійснити самоперевірку на якість засвоєння матеріалу.

ВСТУП

З виникненням писемності завдання забезпечення таємності й автентичності передаваних повідомлень стало надто актуальним. Насправді, повідомлення, передаване голосом чи жестами, є доступне для стороннього лише того короткого проміжку часу, допоки воно "в дорозі", а щодо його авторства й автентичності в одержувача жодних сумнівів виникати не могло, оскільки він бачив свого співрозмовника. Записане на папері повідомлення існує в матеріальному світі набагато довше, і в людей, котрі хочуть ознайомитися з його змістом всупереч волі відправляча й одержувача, виникає набагато більше шансів це зробити. Тому саме після виникнення писемності з'явилося мистецтво тайнопису, мистецтво "таємно писати" – набір методів, призначених для секретного передавання записаних повідомлень від однієї людини до іншої. Що жвавіше провадилося листування в суспільстві, тим більше відчувалася потреба в засобах його засекречування. Відповідно виникали все більш складні й хитромудрі шифри. Спочатку при зацікавлених особах з'явилися окремі шифрувальники, потім – групи з кількох шифрувальників, а потім – і цілі шифрувальні відділи. Коли обсяги підлягаючої утаємниченню інформації стали критичними, на допомогу людині було створено механічні пристрої для шифрування.

Класичні методи шифрування 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.

Сучасна криптографія містить чотири великих розділи:

1 Симетричні криптосистеми.

2 Криптосистеми з відкритим ключем.

3 Системи електронного підпису.

4 Керування ключами.

1 КЛАСИЧНІ МЕТОДИ ШИФРУВАННЯ

Як вважають вітчизняні й зарубіжні фахівці, 90 % інформації, яка не призначена для сторонніх очей та вух, таємно знімається з апаратів і мереж телефонного, телеграфного й комп'ютерного зв'язку. Чи великі при цьому є втрати? За даними зарубіжних джерел, втрата 20 % інформації, котра становить комерційну таємницю і стала „здобиччю" конкурентів, впродовж одного-півтора місяців у майже половині випадків призводить до розорення фірми.

Припустімо, що відправляч хоче надіслати повідомлення одержувачеві. Більш того, відправляч прагне засекретити це повідомлення, аби ніхто, окрім одержувача, не зміг його прочитати. Для цього він має захистити інформацію, яка міститься в повідомленні.

Завдання захисту інформації розв'язується у три основні способи.

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

Одразу ж прокоментуємо, 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.*

Одразу ж зауважимо, що розробляннням засобів та методів приховування факту передавання повідомлення займається ***стеганографія***.

Приміром, в часи давньогрецького історика Геродота (V ст. до н. е.) голову раба голили, на шкірі голови писали повідомлення й, коли волосся відростало, раба доправляли до адресата. Ще 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).

Добре відомі є випадки, коли секретні послання записувалися невидимим *симпатичним* чорнилом (у тому числі й на краватках, носовиках, нижній білизні тощо). Чорнило знову ставало видимим після обробляння спеціальним хімічним реактивом чи освітляння 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 називається ***зашифруванням***. Вибір конкретного типу перетворювання визначається ключем зашифровування. Інакше кажучи, зашифровування – це процес перетворювання відкритого повідомлення на закрите. Зашифроване повідомлення називається шифртекстом.

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

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.

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.

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.

Криптографи надають таке означення: ***відкрита інформація*** – це невтаємничена (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:

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

Перетворювання M_k визначається відповідним алгоритмом і значенням параметра K. 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 захищувано у різноманітні способи від всіляких погроз. Неправочинний користувач прагнутиме відвіднайти 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.

1.1 Шифри 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.

Шифр перестановки здійснює перетворювання переставляння літер у відкритому тексті. Типовим прикладом шифру перестановки є шифр Сцитала. Зазвичай відкритий текст розбивається на відрізки однакової довжини і кожен відрізок шифрується незалежно. Нехай, приміром, довжина відрізків дорівнює n та δ – взаємнооднозначне відбиття множини $\{1, 2, ..., n\}$ в собі. Тоді шифр перестановки впроваджується у такий спосіб: відрізок відкритого тексту $x_1...x_n$ перетворюється на відрізок шифрованого тексту $x_{\delta(1)}...x_{\delta(n)}$.

Оберемо ціле додатне число, скажімо, 5; розташуємо числа від 1 до 5 у дворядковий запис, в якому другий рядок – довільне переставляння чисел верхнього рядка:

1	2	3	4	5
3	2	5	1	4

Цю конструкцію називають перестановкою, а число 5 – степенем перестановки.

Зашифруємо фразу «ДО БУЛАВИ ТРЕБА ГОЛОВИ». У цій фразі 19 літер. Доповнимо її довільною літерою (наприклад Ь) до найближчого числа, кратного до 5, тобто 20. Випишемо цю доповнену фразу без пропусків, водночас розбивши її на п'ятизнакові групи:

ДОБУЛ АВИТР ЕБАГО ЛОВИЬ

Літери кожної групи переставимо відповідно до зазначеного дворядкового запису за таким правилом: перша літера ставиться на третє місце, друга – на друге, третя – на п'яте, четверта – на перше і п'ята – на четверте. Здобутий текст виписуємо без пропусків:

УОДЛБТВАРИГБЕОАИОЛЬВ

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.

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 зреалізовує один з варіантів сучасного так званого шифру маршрутної перестановки. 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. Він запропонував використовувати конусоподібний "спис", на який намотувався перехоплюваний ремінець; цей ремінець пересувався віссю доти, аж допоки не з'являвся осмислений текст.

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".

1.1.2 "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.

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 надавано певної "магічної" сили. Наведемо приклад: у квадрат розміром 4×4 вписуються цифри від 1 до 16.

Зашифровування за "магічним квадратом" здійснюється у такий спосіб.

Приміром, треба зашифрувати фразу: «ПРИЛІТАЮ СЬОГОДНІ». Літери цієї фрази вписуються послідовно до квадрата відповідно до записаних в них чисел, а в порожні клітинки (якщо такі є) проставляють довільні літери.

Після цього зашифрований текст записується вже в рядок:

ІИРОІЬОЮСТАГЛНДП

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".

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

16 **І**	3 **И**	2 **Р**	13 **О**
5 **І**	10 **Ь**	11 **О**	8 **Ю**
9 **С**	6 **Т**	7 **А**	12 **Г**
4 **Л**	15 **Н**	14 **Д**	1 **П**

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 схарактеризовуються частотним спектром. Їх можна розглядати в розгорненні в часі й за спектром.

Відомі є кілька типових перетворювань аналогового сигналу, котрі може бути легко зреалізовано 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. 1.1, 1.2).

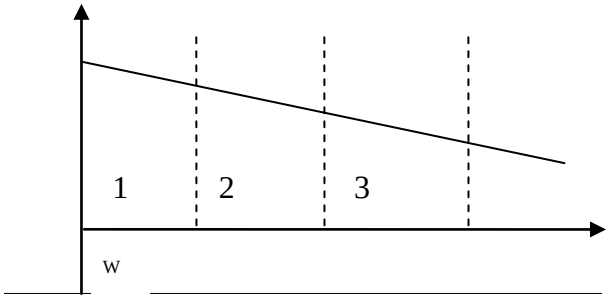


Рисунок 1.1 – Поділення ширини смуги

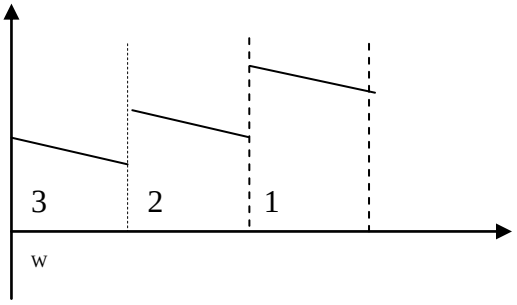


Рисунок 1.2 – Перестановка частотних смуг

Найпростіший скремблер обмежує захист уведенням подібних найпростіших частотних перетворювань. Серійний скремблер переставляє діапазони 250...675 Гц, 675...1100 Гц, 1100...1525 Гц та 1950...2375 Гц.

У даній схемі на вхід вузла накладання шифру подається одне й те саме керування, що воно зорганізовує переставляння.

Інвертування спектра. Більш складні скремблери додатково здійснюють інвертування спектра (рис. 1.3).

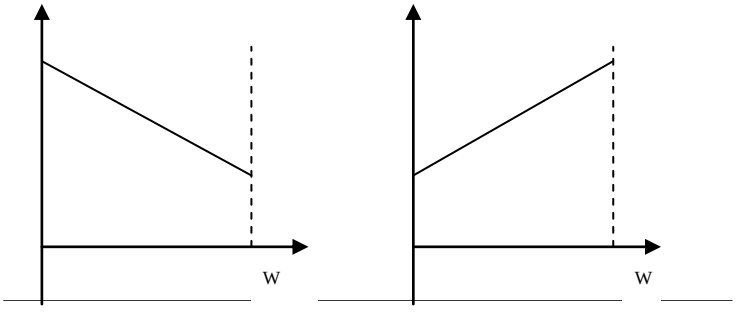


Рисунок 1.3 – Інвертування спектра

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. Про складність завдання зашифровування й розшифровування аналогових повідомлень писав Солженіцин у своєму «В круге первом»: «…Клиппирование, демпфирование, амплитудное сжатие, электронное дифференцирование и интегрирование привольной человеческой речи были таким же инженерным издевательством над ней, как если бы кто-нибудь взялся расчленить Новый Афон или Гурзуф на кубики вещества, втиснуть в миллиард спичечных коробков, перевезти самолетом в Нерчинск, на новом месте распутать, неотличимо собрать и воссоздать субтропики, шум прибоя, южный воздух и лунный свет. То же, в некоторых импульсах, надо было сделать и с речью, даже воссоздать ее так, чтобы не только было понятно, но Хозяин мог бы по голосу узнать, с кем говорит…». Так само барвисто й дещо зневажливо Солженіцин відгукувався про власне роботу з розробляння вітчизняних засобів захисту телефонної інформації. Однак тут він був неправий. На думку висококваліфікованих фахівців, того часу робота йшла вельми цілеспрямовано й апаратуру насправді випускали в призначений термін. Це – апаратура часової стійкості. За наявності спецтехніки типу видимого мовлення зміст перемов удається відновити. Якщо забути про інверсії, то ми маємо шифр перестановки.

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.

1.2 Шифри простої заміни

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

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

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

AA, AB, …, YZ, ZZ,

абетку з $26^3 = 17576$ триграм

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

$$\sum_{\text{укр}} = \{\text{АБВГГ'Д … ЮЯ}\}$$

та множиною цілих

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

поміж російською абеткою

$$\sum_{\text{рос}} = \{\text{АБВГДЕ … ЮЯ}\}$$

та множиною цілих

$$\overline{Z}_{32} = \{0, 1, 2, 3, \ldots, 31\};$$

поміж англійською абеткою

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

та множиною цілих

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

(табл. 1.1, 1.2 та 1.3).

Надалі буде зазвичай використовуватися абетка

$$\overline{Z}_{m} = \{0, 1, 2, 3, \ldots, m-1\},$$

яка містить m „літер” (у вигляді чисел).

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 $n = 1, 2, 3, \ldots$

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, …; потім під нею виписувалась та ж сама абетка, але з циклічним зсуненням на три літери ліворуч:

А Б В Г Ґ 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
Г Ґ 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

При зашифровуванні літера А замінювалася на літеру Г, В – замінювалася на Д, У – на Ц й т. д. Приміром, слово РИМ перетворювалося на УЙП. Одержувач повідомлення УЙП шукав ці літери в нижньому рядку і по літерах над ними відновлював вихідне слово РИМ. 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.

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 Винаходи Енея

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.

1.2.2.1 Диск Енея

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 перехоплянні 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.

1.2.2.2 Лінійка Енея

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 шифруванні 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.

1.2.2.3 Книжковий шифр

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.

1.2.3 Полібіанський квадрат

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 міститиме 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.

1.2.4 Спосіб шифрування Тритемія і його застосовування

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.

1.2.4.1 Шифр Белазо

Наступний крок у розвиненні запропонованого Тритемієм 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'ятовувані 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. Отже, якщо паролем є слово ROI, то при зашифровуванні слова FIGHT

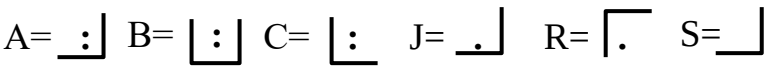
здобуваємо WWOYH. Аналогічні ідеї щодо зашифровування використовуються й сьогодні.

1.2.4.2 Шифр „братерства франкмасонів”

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:

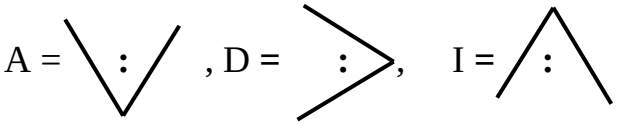
A:	**B:**	**C:**	**J.**	**K.**	**L.**	**S**	**T**	**U**
D:	**E:**	**F:**	**M.**	**N.**	**O.**	**V**	**W**	**X**
G:	**H:**	**I:**	**P.**	**Q.**	**R.**	**Y**	**Z**	

Відповідно до цих фігур літери набувають такого геометричного подання:

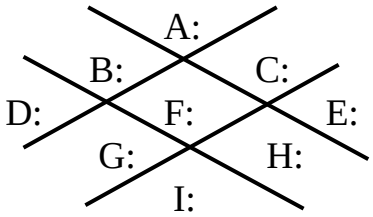


Фраза "We talk about" при зашифровуванні набуває вигляду

Геометричне подання може змінюватися, приміром, на



Тоді



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

1.2.5 Шифрувальні таблиці Трисемуса

1508 року 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 дістання 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.

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

1.2.6 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;

– 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.

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. 4.

Таблиця 1.4 – 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, зреалізовуваний 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
66	68	28	42	80	46	88

ET	CON	NON	CHE	ПУСТИШКИ
08	64	00	44	5, 7

Слово ARGENTI могло набути при зашифровуванні вигляду

5128066284580377 або 1772850682584780537.

Це було насправді серйозне ускладнення шифру заміни. Частісний 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. 1.4.

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).

Наведемо приклад:

Гасло: с к а р б н и ц я с к а р б н и ц я .....

Відкритий текст: п е р і о д и ч н и й ш и ф р .....

Шифртекст: з ш г і 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).

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

Рисунок 1.4 – Шифр Белазо для української мови

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 поверталися 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. Це є не що інше, як шифр перестановки.

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.

Запропонований Кардано шифр-ґрати покладено в основу славнозвісного шифру Рішельє, в якому шифртекст мав вигляд "безневинного" послання. З цупкого матеріалу вирізувався прямокутник розміром, приміром, 7×10 клітинок; у ньому робилися "вікна". Секретний текст вписувався в ці „вікна”, потім ґрати знімалися – і клітинки, що вони залишилися, заповнювалися у такий спосіб, аби вийшло "безневинне" повідомлення. Зрозуміло, використовування цього шифру спричинює утруднення й вимагає інтелекту певного рівня.

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

1585 року 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 зреалізовано 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.

При шифруванні вихідного повідомлення 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 зашифровуваною літерою, і рядка, визначуваного числовим значенням ключа.

Нехай ключова послідовність має довжину r, тоді ключем r-абеткового підставляння є r-рядок

$$\bar{\pi} = (\pi_0, \pi_1, ..., \pi_{n-1}).$$

Система шифрування Віженера перетворює відкритий текст $\bar{x} = (x_0, x_1, ..., x_{n-1})$ на шифртекст $\bar{y} = (y_0, y_1, ..., y_{n-1})$ за допомогою ключа $\bar{\pi} = (\pi_0, \pi_1, ..., \pi_{n-1})$ згідно з правилом

$$M_\pi : \bar{x} = (x_0, x_1, ..., x_{n-1}) \to \bar{y} = (y_0, y_1, ..., y_{n-1});$$

$$(y_0, y_1, ..., y_{n-1}) = (\pi_0(x_0), \pi_1(x_1), ..., \pi_{n-1}(x_{n-1})),$$

де $\pi_i = \pi_{(i \bmod r)}$.

Недолік цього шифру – його слабка стійкість: якщо використовувана таблиця Тритемія є відома, то для дешифрування досить опробувати першу (ключову) літеру – і шифр "розколюється".

Другий варіант використовування таблиці Тритемія, запропонований Віженером, полягає в застосовуванні гасла. Власне Віженер сполучив підходи Тритемія, Белазо, Порта до шифрування відкритих текстів, істотно не додавши до них нічого оригінального.

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 гаммування – одного з найпоширеніших шифрів у сучасній криптографії. 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 зашифровуванні 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 гаммування.

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 відрізнюється тим, що, якщо сума перевищує 26, від неї віднімається 26; обернена операція – віднімання – схарактеризовується 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.

Астрологічні захоплення Віженера навернули його до шифру, в якому шифрзнаками є положення небесних тіл у момент зашифровування. Він намагався перевести свої послання на "мову неба".

Історія іноді "забувається" на сторіччя. Нашого часу шифр Віженера, який полягає в періодичному продовженні ключового слова за таблицею Тритемія, витіснив імена попередників. При цьому цей шифр часто зпримітизовується до елементарності, завдаючи образи його авторові. На початку XX сторіччя один з популярних американських журналів подав вельми спрощену систему Віженера як шифр, який "неможливо розкрити!".

Шифри Віженера з коротким періодичним гаслом використовуються й у наші дні в системах шифрування, які не потребують високої криптографічної стійкості. Приміром, ці шифри використовуються в програмі-архіваторі "ARJ", у програмі "Word for Windows" тощо.

Шифр Віженера надалі модернізувався. Так, у XIX сторіччі англійський адмірал Бофор запропонував використовувати таблицю, подану на рис. 1.5. У такої таблиці є одна перевага: правила зашифровування й розшифровування за нею збігаються: літери вилучаються з верхнього ряду абетки. Це створює певні зручності при використовуванні шифрів: не треба запам'ятовувати два різні правила (зашифровування й розшифровування).

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 гаммування.

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

АБВГДЕЄ ЖЗИ І Ї Й К Л М Н О П РС Т У ФХЦЧШЩ Ь ЮЯ
1 3 5 6 7
2 4

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 такій 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. 1.6).

Л	В	Б	К	А	Ш	У	І	Ф	Д
Ш	И	Ф	Р	З	А	М	І	Н	И
…	…	…	…	…	…	…	…	…	…
А	К	О	Ф	Ц	Ь	Л	У	Е	Г
…	…	…	…	…	…	…	…	…	…
В	Н	Ф	Р	Л	Я	Н	Є	Ш	Х

Рисунок 1.6 – 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}.

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. 1.7).

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 долучуватися 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.

1.3.3 Одноразова система шифрування

Майже всі застосовувані на практиці шифри схарактеризовуються 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 шифртекст

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

за допомогою підстановки Цезаря

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

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

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

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 Москва–Київ. 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.

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 підгрунті 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 щонайменш 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 функціювання 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_j. 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 $K_j \in K$ певна ймовірність $p(K_j) \in P(K)$ і виконуються правила

$$\sum_{M_i \in M} p(M_i) = 1 \quad \text{та} \quad \sum_{K_i \in K} p(K_i) = 1.$$

Ключ, сформований з боку, звідки подається повідомлення з імовірністю $p(K_j)$, 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.

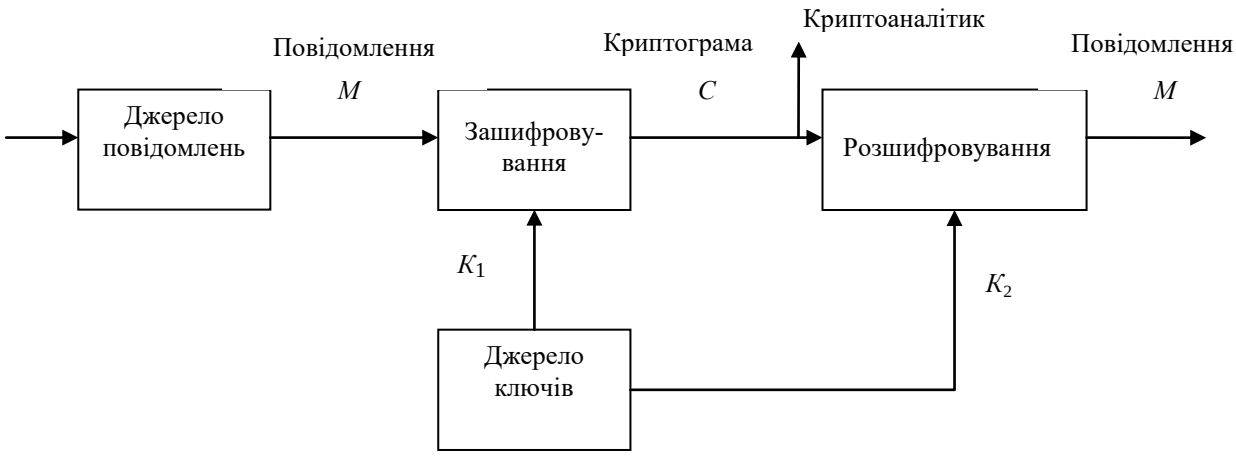


Рисунок 2.1 – Схема секретної системи зв'язку

2.3 Вимоги щодо реалізації сучасних криптоалгоритмів

Сучасні криптографічні системи може бути зреалізовувано 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 зреалізовувалися 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 зреалізовувані 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 і т.д.) 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, найчастіш 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 відрізнюється від нуля.

Дешифрування з імовірністю, яка відрізнюється 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 схарактеризовує 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 зреалізовує 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 схарактеризовує 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, опублікованій 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 зреалізовуватися програмними чи апаратними засобами.

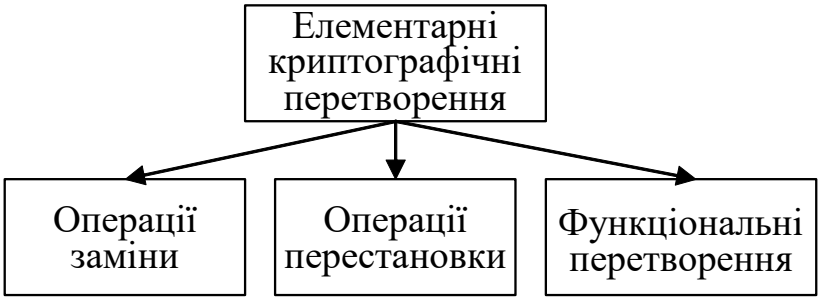


Рисунок 2.2 – Елементарні криптографічні перетворювання

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.

У загальному випадку шифр заміни можна описати алгебричною моделлю вигляду $\Sigma_K = (M, K, C, E, D)$.

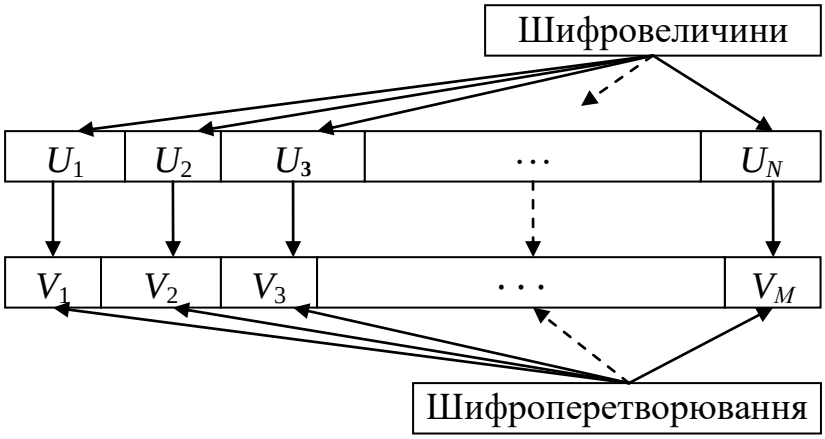


Рисунок 2.3 – Загальна засада заміни
(однозначна заміна)

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 шифровеличин та відповідних до них шифроперетворювань може не збігатися. Окрім того, кожній з шифровеличин може бути поставлено у відповідність понад одне шифроперетворювання. Тобто під час зашифровування повідомлення заміна може бути не безваріантною. Щодо розшифровування, то воно залишається однозначним, незалежно від способу зашифровування.

З урахуванням наведеного, множину шифровеличин U можна розглядати як множину слів U_i абетки A^*:

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

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

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

Тоді $M \subset A^*$, а $C \subset B^*$. Якщо кількість символів у абетках A та B відповідно дорівнює n та m, сумарна кількість шифровеличин U_i має становити $N > n$, а сумарна кількість шифроперетворювань V_i – $M > m$.

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

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

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

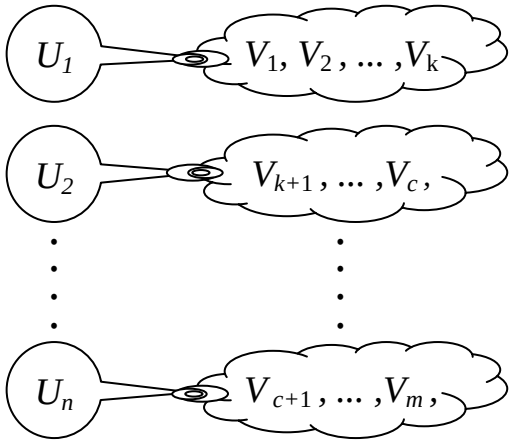


Рисунок 2.4 – Відповідність величин V_i до U_j багатозначна заміна)

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:

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

Отже, з урахуванням вищенаведеного, можна надати таке означення: *шифром заміни* є шифр, котрий передбачає заміну шифровеличин відкритого тексту на відповідні до них фіксовані шифроперетворювання або групи шифроперетворювань закритого тексту.

У найпростішому випадку шифровеличини збігаються із символами абетки A_m, за допомогою яких утворюються відкриті повідомлення, а шифроперетворювання – 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 шифровеличин та відповідних до них шифроперетворювань, 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

$$b_i^{'} = (a_i + k) \bmod m, \quad 1 \le k < m,$$

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 рисунком 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 змеханізовувати процедури шифрування великих обсягів повідомлень.

Шифрування відкритих текстів способом перестановки передбачає зміну позицій шифровеличин 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

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

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

$$\sigma : S \to S,$$

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

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

$$S = R_1 \cdot R_2$$

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

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

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

а система $R_2 - n$ ключів, які обираються відповідно з імовірностями

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

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

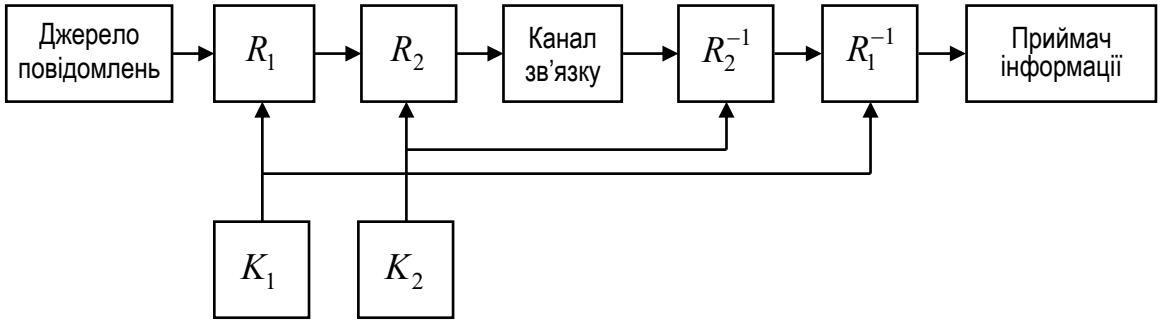


Рисунок 2.5 – Множення двох секретних систем $S = R_1 \cdot R_2$

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).

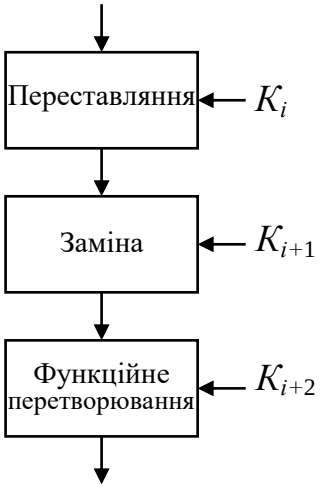


Рисунок 2.6 – Фрагмент складного шифру

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.

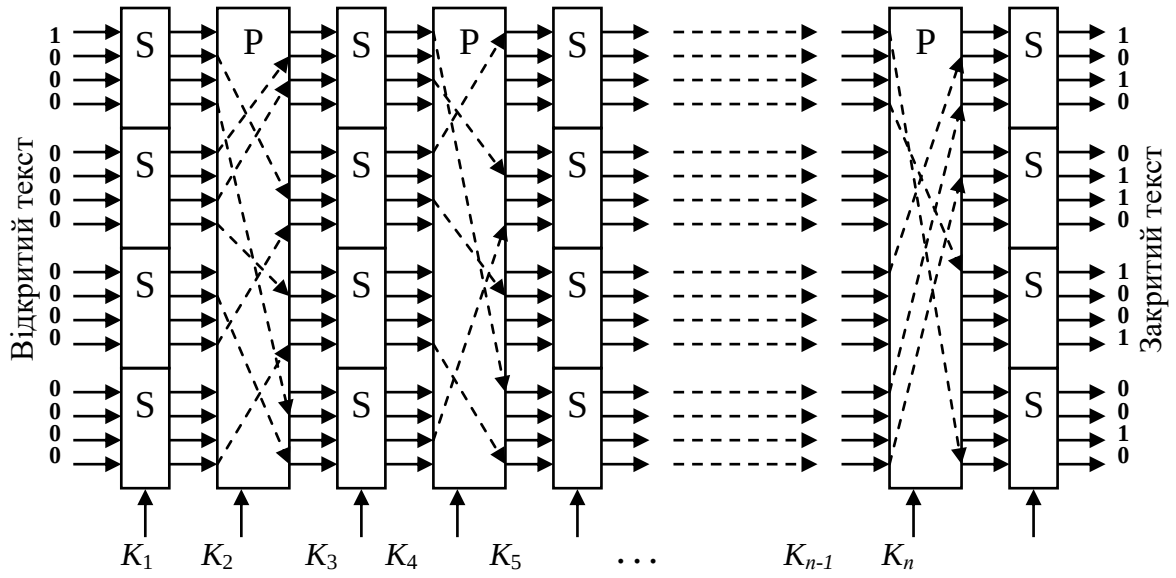


Рисунок 2.7 – Загальна засада побудови системи “Люцифер”

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 під’єднуються до загального блока P, який

забезпечує переставляння вхідних символів. Загальна пам'ять, якої потребує реалізація такого блока, має становити лише 32 біти.

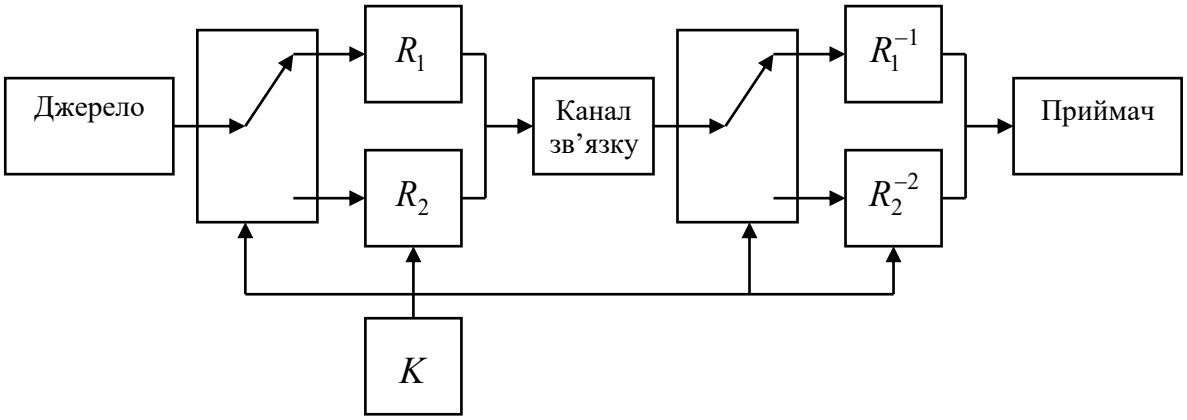


Рисунок 2.8 – "Виважена сума" секретних систем $S = p\,R_1 + q\,R_2$.

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 $S = p_1R_1 + p_2R_2$. 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 імовірностями

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

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 $M_1^1, M_2^1, ..., M_m^1$, з імовірностями $pp_1, pp_2, ..., pp_m$. Якщо ж, навпаки, буде обрано систему R_2, це повідомлення буде перетворене на одну з криптограм $M_1^2, M_2^2, ..., M_m^2$, з імовірностями $qp_1, qp_2, ..., qp_m$.

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

$$S = p_1R_1 + p_2R_2 + p_3R_3 + ... + p_nR_n, \quad \sum p_i = 1.$$

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.

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 $\pi \in \overline{SYM}\ (Z_{2,N})$, $\pi : x \to y = \pi\ (x)$, 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 $\pi \in \overline{SYM}\ (Z_{2,N})$, 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 $\overline{SYM}\ (Z_{2,N})$, тоді можна зробити певний висновок. Наприклад, коли

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

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 шифрованного 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 $\overline{SYM}\ (Z_{2,N})$

$$\Pi[K] = \{\pi\{K\} : K \in \overline{K}.\},$$

яка індексується за параметром $K \in \overline{K}$; де K – ключ, а $\overline{K}$. – простір ключей. 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 з К, 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 ключи 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 К;

– 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 $\overline{SYM}\ (Z_{2,N})$ може використовуватися як підстановка, так що $\overline{K} = \overline{SYM}\ (Z_{2,N})$.

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 певніх *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 $\overline{K} = \overline{SYM}\ (Z_{2,64})$ 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 $\pi\{K, x\}$: $x \rightarrow y = \pi(K, x)$ поміж вихідним повідомленням та шифртекстом, для того щоби аналітичні та/чи статистичні методи визначання вихідного тексту та/чи ключа на базі відповідності вихідного повідомлення та шифртексту не можна було зреалізовувати.

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.

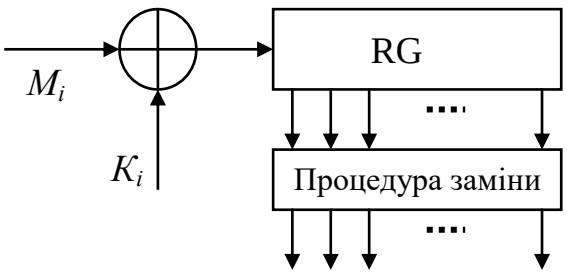


Рисунок 2.9 – Варіант попередньої комбінації даних і ключа

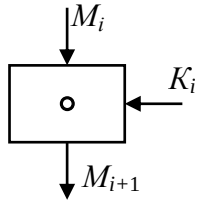


Рисунок 2.10 – Операція над шифрувальними даними на підставі ключа

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 маєбути придатна для реалізації умов побудови симетричної криптосистеми (рис. 2.10):

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

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

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

загальному вигляді цю функцію можна записати як

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

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.

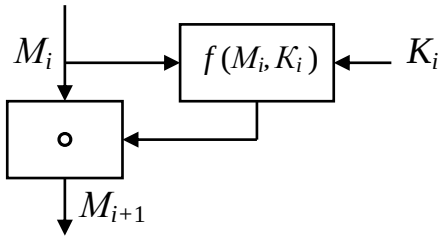


Рисунок 2.11– Операція над шифрувальними даними на підставі функційних перетворювань

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 Фейстеля для побудови блочних шифрів

Найпоширеним способом створювання блочних шифрів є використовування мереж Фейстеля (рис. 2.12).

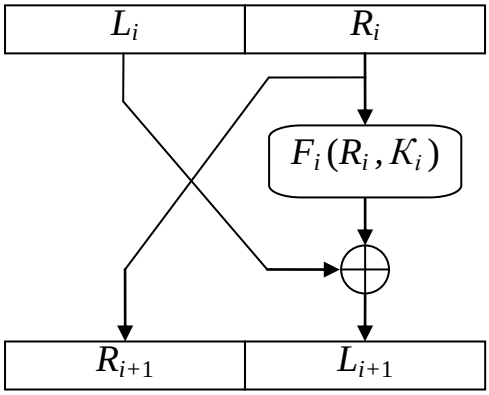


Рисунок 2.12 – Схема мережі Фейстеля

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 гамми. Спосіб її формування є найскладнішою з проблем сучасної криптографії.

Якщо гамма побудована за допомогою будь-якого, зреалізованого програмним методом генератора, вона неодмінно буде рекурентною, тобто її вміст матиме скінченну довжину й циклічно повторюватиметься через певну кількість кроків. Визначивши період повторювання такої гамми, 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 гамму за допомогою складних апаратних засобів, які використовують не рекурентні, а реальні випадкові процеси, процедура розподілу ключової інформації стає надто складною.

Хорст Фейстель підійшов до проблеми в такий спосіб. Він запропонував формувати шифрувальну гамму з самої інформації, яка підлягає шифруванню (див. 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 гамма, 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 гаммування лівої половини попереднього блока (складанням її "за модулем два" зі створеною гаммою) 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 Фейстеля; в іншому разі – це *розбалансована* схема Фейстеля. Це, своєю чергою, потребує формування різних за довжиною шифрувальних гамм та різних функцій для виконання відповідних нелінійних перетворювань, залежно від номера ітерації.

Зазвичай виникає запитання про те, чи буде перетворювання, зреалізоване за допомогою схеми Фейстеля, оберненим? Властивість схеми Фейстеля 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 Фейстеля 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 Фейстеля, цілком залежить від вигляду нелінійної функції гаммування $F_i(R_i, K_i)$, 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 Фейстеля, 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 Фейстеля 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 гаммою, 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 Фейстеля, 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 гаммування блока даних перед початком його шифрування. Гамма 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 містять чотири гілки (рис.2.13 та рис. 2.14).

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

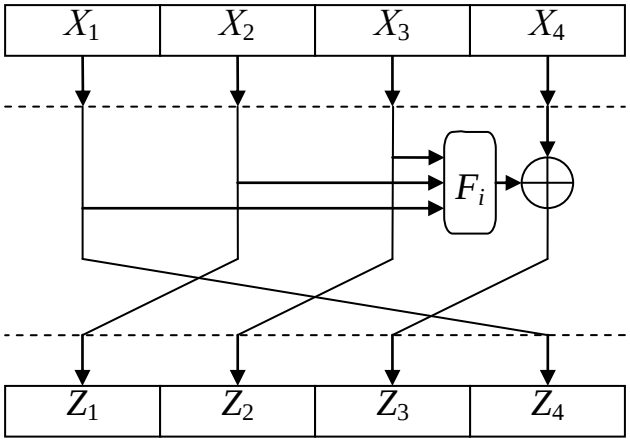


Рисунок 2.13 – Приклад схеми Фейстеля з чотирма гілками

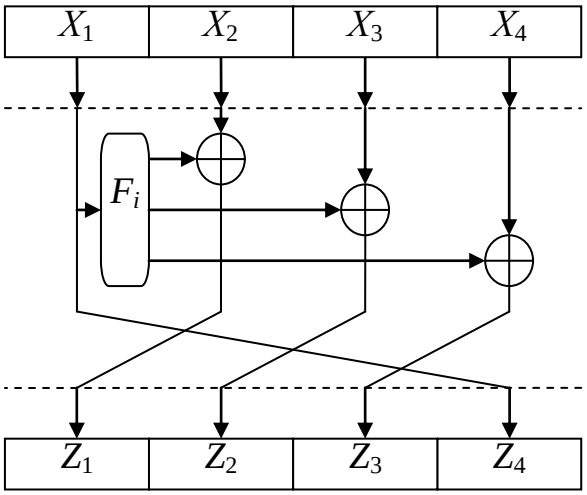


Рисунок 2.14 – Приклад схеми Фейстеля з чотирма гілками

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 гаммою. Від того, наскільки утворена гамма 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 змодифіковувати ключі, які шифрують повідомлення.

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

2.6 Шифри гаммування

2.6.1 Накладання гамми шифру на відкритий текст

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

Під гаммуванням розуміють процес накладання за певним законом гамми шифру на відкриті дані. *Гамма шифру* – це псевдовипадкова послідовність, створена за заданим алгоритмом для зашифровування відкритих даних і розшифровування зашифрованих даних.

Процес зашифровування полягає в генеруванні гамми шифру і накладанні здобутої гамми 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 $M_0^{(i)}$ однакової довжини, зазвичай по 64 біти. Гамма шифру створюється у вигляді послідовності блоків $\Gamma_{\text{ш}}^{(i)}$ аналогічної довжини.

Рівняння зашифровування можна записати у вигляді

$$M_{\text{ш}}^{(i)} = \Gamma_{\text{ш}}^{(i)} \oplus M_0^{(i)}, \quad i = 1, ..., N,$$

де $M_{\text{ш}}^{(i)}$ – i-тий блок шифртексту; $\Gamma_{\text{ш}}^{(i)}$ – i-тий блок гамми шифру; $M_0^{(i)}$ – i-тий блок відкритого тексту; N – кількість блоків відкритого тексту.

Процес розшифровування зводиться до повторного генерування гамми шифру і накладання цієї гамми на зашифровані дані. Рівняння розшифровування має вигляд

$$M_0^{(i)} = \Gamma_{\text{ш}}^{(i)} \oplus M_{\text{ш}}^{(i)}.$$

Здобуваний цим методом шифртекст є вельми складний для розкриття, оскільки тепер ключ є змінним. По суті, гамма шифру повинна змінюватися у випадковий спосіб для кожного зашифровуваного блока. Якщо період гамми 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 гаммування 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 гаммою шифру, чи просто гаммою (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 (гамми шифру) пред'являються три основних вимоги:

- період гамми повинен бути доволі великим для шифрування повідомлень різної довжини;
- гамма має бути практично непередбачуваною, що означає неможливість завбачати наступний біт гамми, навіть якщо відомі є тип генератора і попередній фрагмент гамми;
- генерування гамми не повинно спричинюватись до значних технічних складностей.

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

Друга вимога позв'язана з такою проблемою: в який спосіб можна вірогідно переконатися, що псевдовипадкова гамма конкретного генератора є насправді незавбачуваною? На сьогодні не існує таких універсальних і практично перевірних критеріїв та методик. Щоби гамма вважалася за незавбачувану, 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

$$Y_i = (a \cdot Y_{i-1} + b) \bmod m,$$

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:

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

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 зреалізовуються на комп'ютері; при цьому реалізація виходить надтопростою, 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.

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

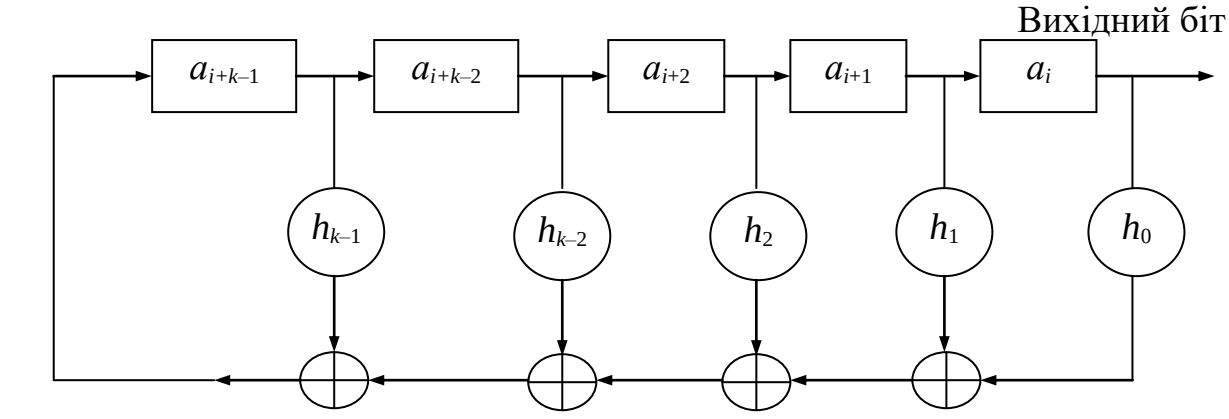
$$h(X) = \sum_{j=0}^{k} h_j X^n,$$

де X – формальна змінна; h_j – коефіцієнт при X^n, 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

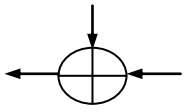
$$g(X) = (X^n - 1)/h(X),$$

тоді розв’язки рекурентних співвідношень

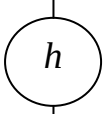
$$\sum_{j=0}^{k} h_j a_{i+j} = 0$$



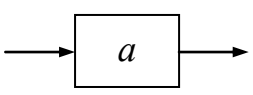
Позначення:



– суматор за модулем 2



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



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

Рисунок 2.15 – Генератор з регістром зсування

Тоді розв’язки рекурентних співвідношень

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

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

$$a(X) = a_0 X^{n-1} + a_1 X^{n-2} + \ldots + a_{n-2} X + a_{n-1},$$

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 $a_0, a_1, a_2, \ldots$ обчислюються в порядку зростання номерів, то коефіцієнти багаточлена $a(X)$ обчислюються, розпочинаючи з коефіцієнтів зі степенями вищих порядків. Слід також зазначити, що вигляд багаточлена

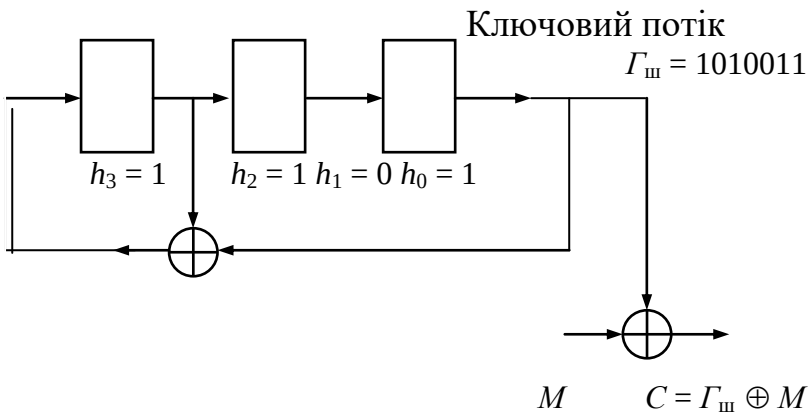
$$h(X) = \sum_{j=0}^{k} h_j X^n$$

визначає конфігурацію зворотних зв'язків (відводів) h_i в генераторі з регістром зсування. Інакше кажучи, якщо в багаточлена $h(X)$ коефіцієнт $h_i = 1$, це означає, що відвід h_i у схемі генератора є наявний, якщо ж у багаточлена $h(X)$ коефіцієнт $h_i = 0$, то відвід h_i 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 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 (гаммою 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 відзнайти 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

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

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

$$X(2) = A \cdot X(1) \bmod 2.$$

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.

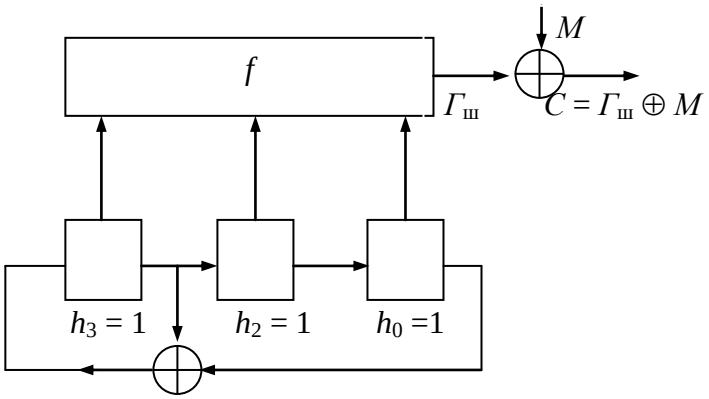


Рисунок 2.17 – Лінійний регістр зсування
з нелінійними логічними ланцюгами на виході

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 придатен і більш короткий регістр.

3 СИМЕТРИЧНІ КРИПТОГРАФІЧНІ СИСТЕМИ

3.1 Класичні симетричні криптосистеми

Класична криптографія, зокрема теорія зв'язку в секретних системах, заснована К. Шенноном, 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 гаммою. Стійкість поточних алгоритмів шифрування залежить від того, наскільки утворена гамма 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.

Гаммування – це перетворювання вихідного тексту, за якого символи відкритого тексту складаються з символами псевдовипадкової послідовності (гаммою). Гаммування 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 гаммування, 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 (найбільший спільний дільник) $(a, m) = 1$ 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

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

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 HIAT.

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.

3.3 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 відновляння 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 зреалізовувано 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 підставлянь, 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 зреалізовувано програмним, програмно-апаратним чи апаратним шляхом.

3.4 Стандарт шифрування DES

Стандарт шифрування даних DES (Data Encryption Standard) було опубліковано 1977 року Національним бюро стандартів США. Типовий блочний шифр, стандарт DES 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 найчастіш застосовується в системах захисту комерційної інформації. На сьогодні розроблено програмне забезпечення та спеціалізовані

ЕОМ, які зреалізовують шифрування інформації в мережах передавання даних.

Суть алгоритму зводиться до такого.

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.

Рисунок 3.1– Узагальнена схема шифрування в алгоритмі DES

З файлу вихідного тексту зчитується черговий 64-бітовий блок *M*. Процес його зашифровування полягає в початковій перестановці, шістнадцятьох циклах шифрування і, врешті, у завершувальній перестановці бітів.

Вхідна послідовність $M = m_1, m_2, \ldots, m_{64}$ перетворюється блоком початкових перестановок на послідовність IP $(M) = m_{58}, m_{50}, \ldots, m_7$, де m_i може набирати значень 0 чи 1. Перетворювання виконуються за фіксованою табл. 3.2.

Таблиця 3.2 – Початкова перестановка

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 – Завершальна перестановка

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 Фейстеля (див. 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:

$$L_i = R_{i-1}, \ i = 1, 2, \ldots, 15; \ L_{16} = L_{15} \oplus f(R_{15}, K_{16});$$
$$R_i = L_{i-1} \oplus f(R_{i-1}, K_i), \ i = 1, 2, \ldots, 15; \ R_{16} = R_{15}.$$

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 $L_1 = R_0$. 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.

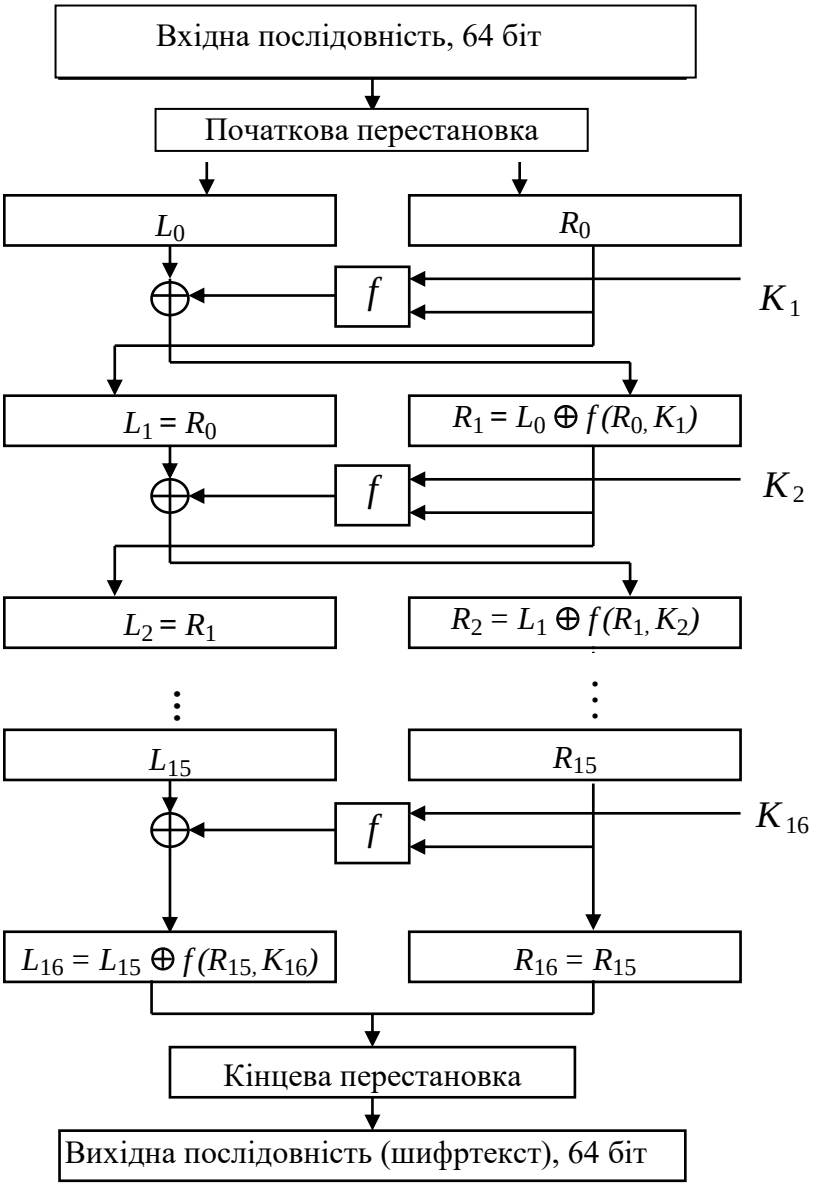


Рисунок 3.2 – Схема алгоритму DES

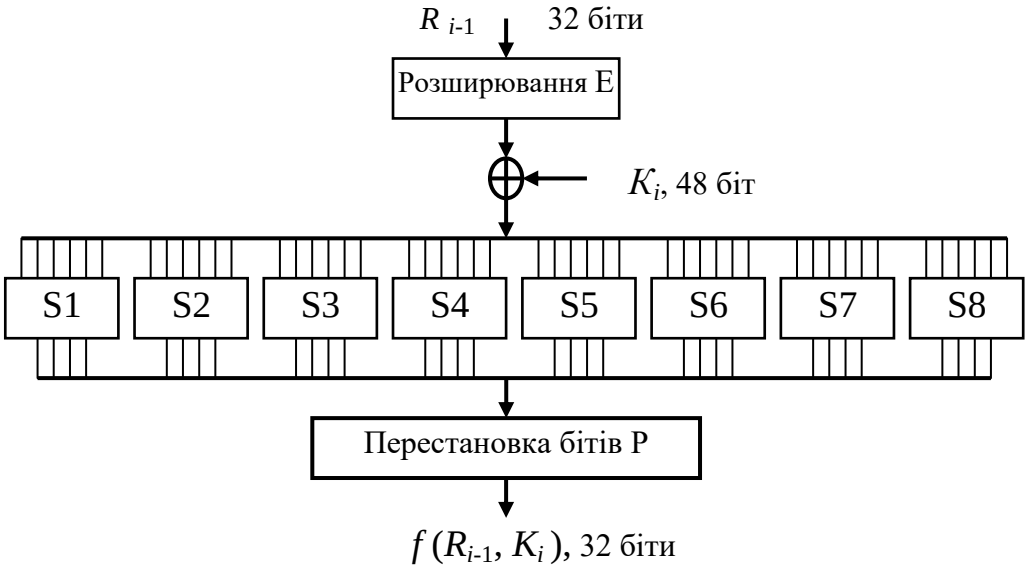


Рисунок 3.3 – Схема обчислювання функції шифрування $f(R_{i-1}, K_i)$

На початку 32-бітовий блок розширюється до 48-ми символів. 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

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_1b_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. Нехай $B = 011011_2$, тоді $S_1(011011) = 5_{10} = 0101_2$. Дійсно,

$$b_1b_6 = 01_2 = 1_{10},\ k = 1;$$
$$b_2b_3b_4b_5 = 1101_2 = 13_{10},\ l = 13.$$

Таким чином, при перетинанні стовпця 13 та рядка 1 для S_1 вузла заміни (табл. 3.5) задано число $5_{10} = 0101_2$. Отже, послідовність 011011 перетворена на послідовність 0101.

S-блоки (вузли заміни) забезпечують нелінійність і стійкість перетворення. На наступному етапі вісім тетрад (4-бітових векторів), що надходять з виходів S-блоків, утворюють 32-бітовий блок, який піддають черговій табличній перестановці P (табл. 3.8 або табл. 3.9, де $P(i)$ є номер позиції, в яку потрапляє i-й біт вхідного 32-бітового блока).

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

Функція перестановки бітів *P* задається табл.. 3.6.

Таблиця 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. 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 табл. 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

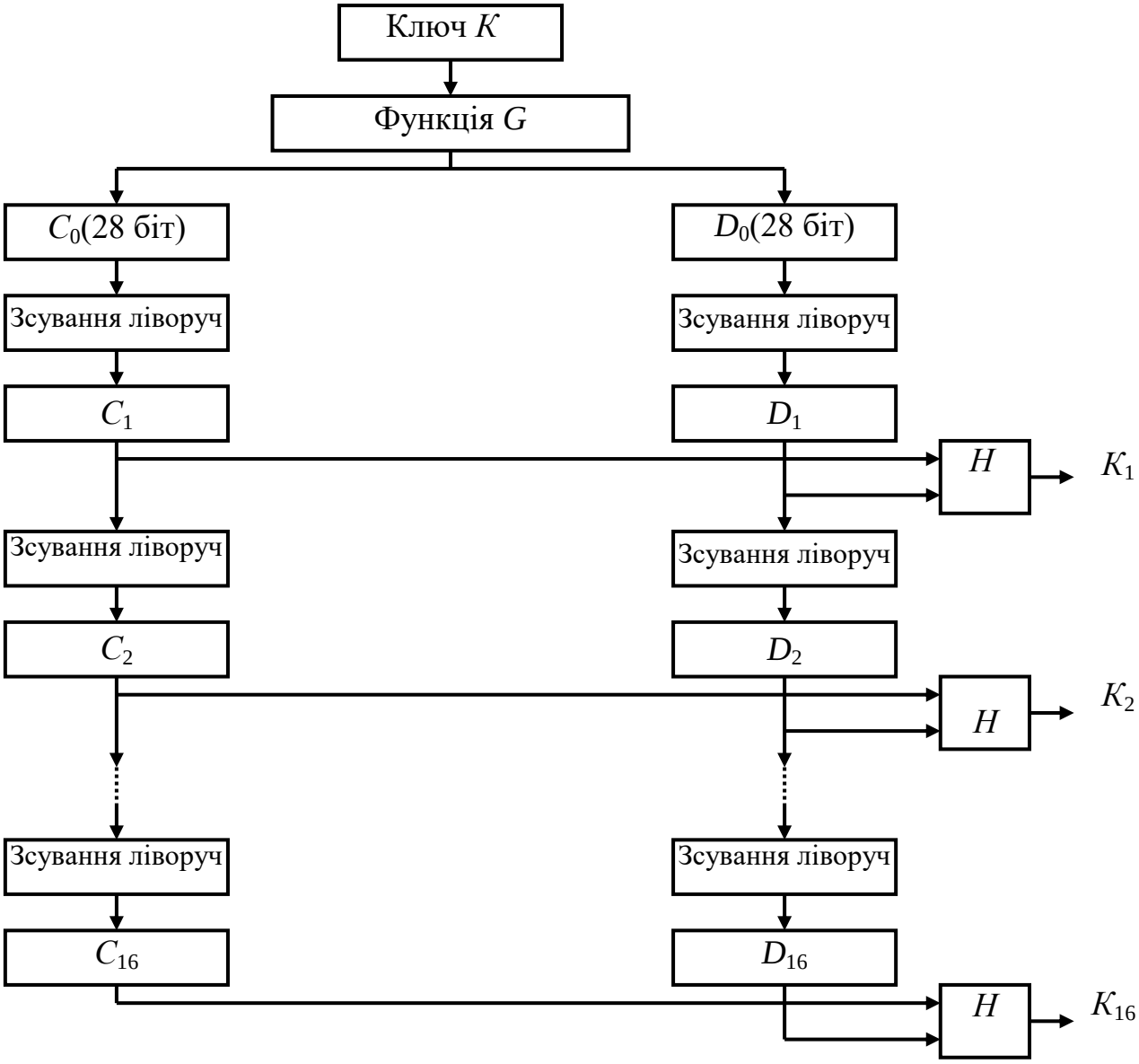


Рисунок 3.4 – Схема алгоритму обчислювання ключів K_i

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. 3.4).

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 добрано 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 відрізнюється лише черговістю формування ключів K_i. Вони повинні подаватися у зворотному порядку – K_{16} ... K_1.

Початково стандарт DES використовувався лише для шифрування та розшифровування даних ЕОМ. Потім він став використовуватися для інших завдань, приміром задля автентифікації чи захисту повідомлень електронної системи платежів при операціях поміж клієнтами та банками. 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. 3.5).

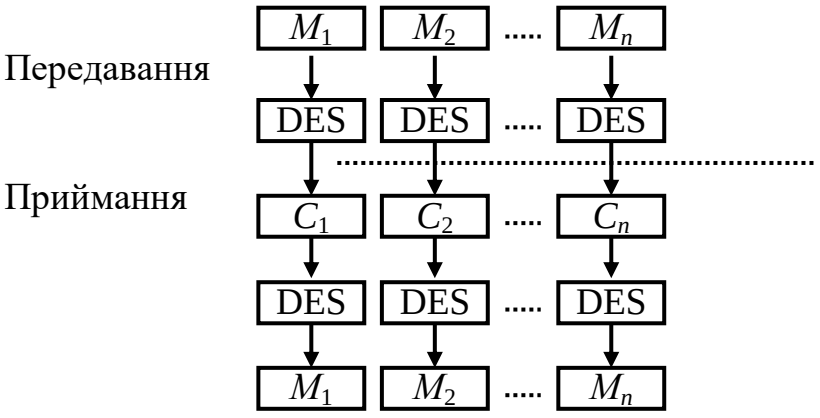


Рисунок 3.5 – Режим електронної кодової книги

Головною перевагою такого алгоритму є простота зреалізовування. Однак він є найменш тривалий щодо спроб розшифровування. Це пояснюється тим,

що за відносно невеликої довжини блока та великої довжини передаваного тексту 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

$$M = M_1 M_2 M_3 \ldots$$

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. 3.6).

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.

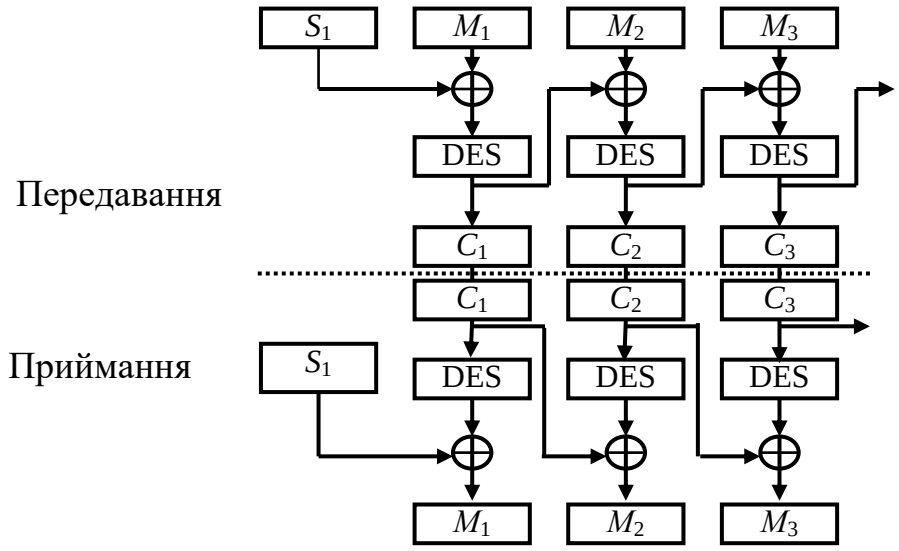


Рисунок. 3.6 – Режим "Зчеплювання блоків шифру"

3.4.3 Режим „Зворотний зв’язок за шифром”

Режим „Зворотний зв’язок за шифром” CFB (Cipher Feedback)припускає, щоби довжина блока відрізнювалась від 64 біт. Припустімо, що довжина блоків, на які розбивається зашифровувана послідовність, становить менше за 64 біти (рис. 3.7).

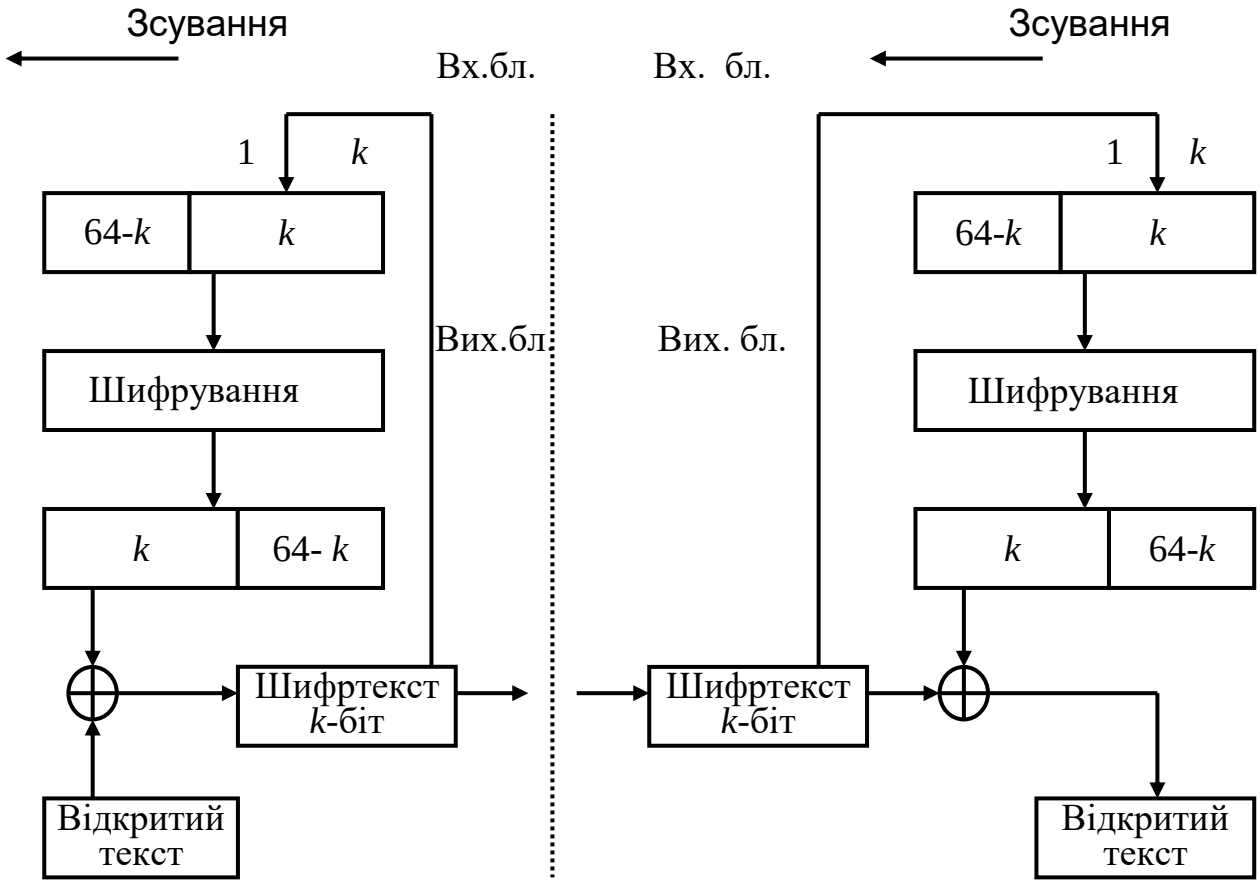


Рисунок 3.7 – Схема алгоритму в режимі зворотного зв’язку за шифром

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. 3.8.

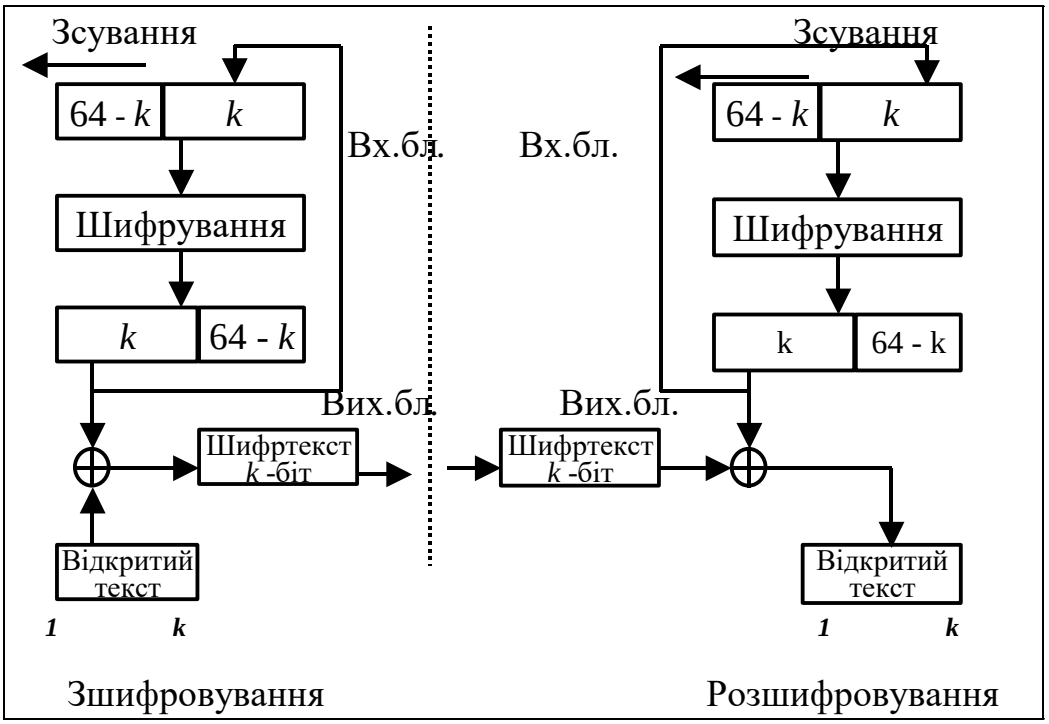


Рисунок 3.8 – Схема алгоритму у режимі зворотного зв’язку за виходом

3.5 Алгоритм шифрування IDEA

Перший варіант шифру було опубліковано 1990 року. Його автори Ксуеджа Лай та Джеймс Мессі. Остаточна назва розшифровується як *International Data Encryption Algorithm* (*IDEA*). Існує думка, що *IDEA* – це один з найбезпечніших блочних алгоритмів, опублікованих на сьогодні. Цей алгоритм передбачає попередній поділ вихідної послідовності на 64-розрядні

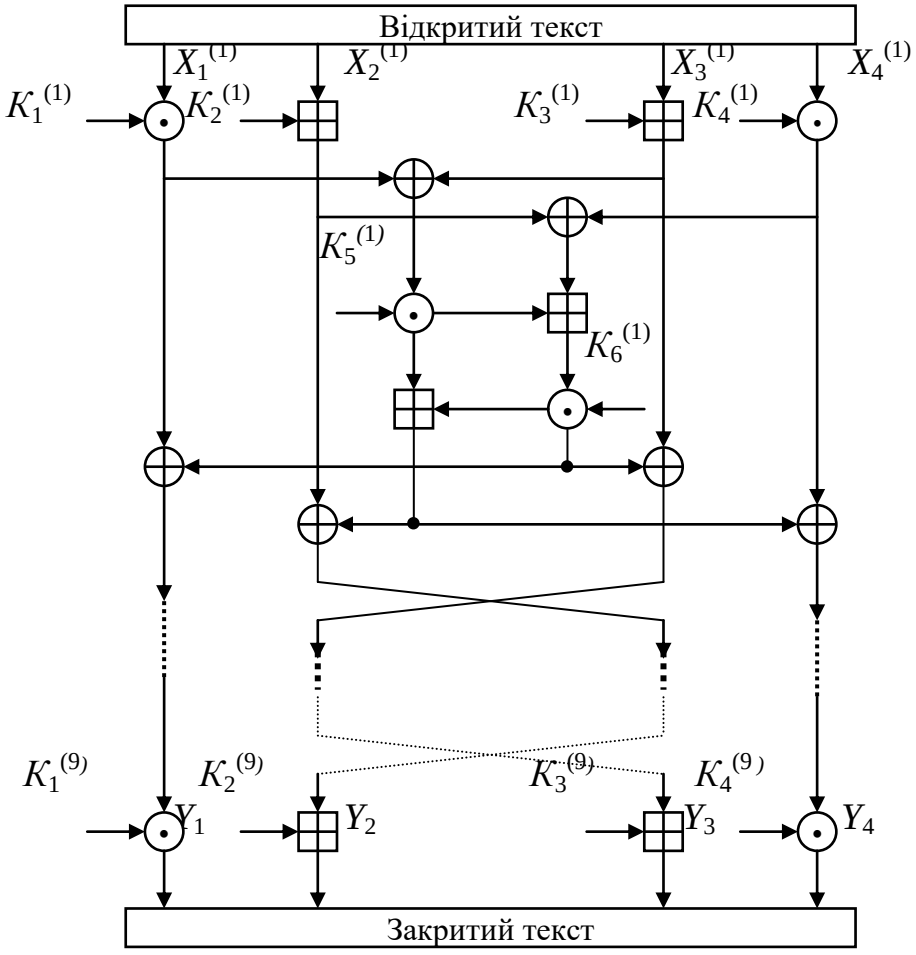
блоки, шифровані за допомогою 128-бітового ключа. Алгоритм побудовано на базі мережі Фейстеля (див. рис. 2.12). Алгоритм шифрування є обернений.

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 3.9 – Один етап шифрування IDEA

Кожний етап передбачає виконання таких операцій:

1 помножуються X_1 та перший підключ K_1;
2 виконується додавання за модулем 2^{16} X_2 та другого підключа K_2;
3 виконується додавання за модулем 2^{16} X_3 та третього підключа K_3;
4 помножуються X_4 та четвертий підключ K_4;
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 K_5;
8 виконується додавання за модулем 2^{16} результатів 6 та 7 кроків;
9 помножуються результати кроку 8 та шостий підключ K_6;
10 виконується додавання за модулем 2^{16} 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.

Алгоритм шифрування IDEA передбачає використовування 52-х підключів (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. Потім 128-бітовий ключ зсовують на 25 біт ліворуч і знову поділяють на вісім ключів. Перші чотири підключа використовують у другому циклі, останні чотири – у третьому циклі. Потім формування наступних ключів виконується аналогічно до завершення алгоритму.

Процес розшифровування супроводжується генерацією ключів у зворотному порядку, причому певні з них змінюються оберненими величинами.

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.

Цей алгоритм запатентовано у країнах Європи та США.

3.6 Стандарт шифрування ГОСТ 28147–89

Алгоритм криптографічного перетворювання даних за ГОСТ 28147–89 призначено для апаратної й програмної реалізації. Цей алгоритм задовольняє потребам, які пред′являються до сучасних криптосистем, не накладає обмежень на секретність передаваної інформації.

Алгоритм побудовано на підгрунті мережі Фейстеля (рис. 3.10). На кожному етапі блок вихідного повідомлення розбивається на ліву L_0 і праву R_0 частини, які шифрують за правилом

$$L_i = R_{i-1};$$
$$R_i = L_{i-1} \oplus f(R_{i-1}, K_i);$$

при цьому використовують проміжне значення ключа K_i. Відмінність від алгоритму DES полягає у тому, що замість 56-розрядного ключа використовують 256-розрядний ключ. В алгоритмі ГОСТу 28147–89 використовують нелінійну функцію $f_{\text{ГОСТ}}$, яка суттєво відрізняється від функції f_{DES}, що використовується в алгоритмі DES. Кількість етапів шифрування є вдвічі більше порівняно з DES, тобто дорівнює 32.

Окрім цього, операція заміни, виконувана *S*-блоками, не є постійна, а може змінюватися за необхідності й, до того ж, її треба тримати в секреті, що є еквівалентне до збільшування ключа практично до 610 біт.

Алгоритм ГОСТу 28147–89 припускає кілька режимів роботи, до яких належать режим простої заміни, режим гаммування, режим гаммування зі зворотним зв′язком та режим формування імітовставки.

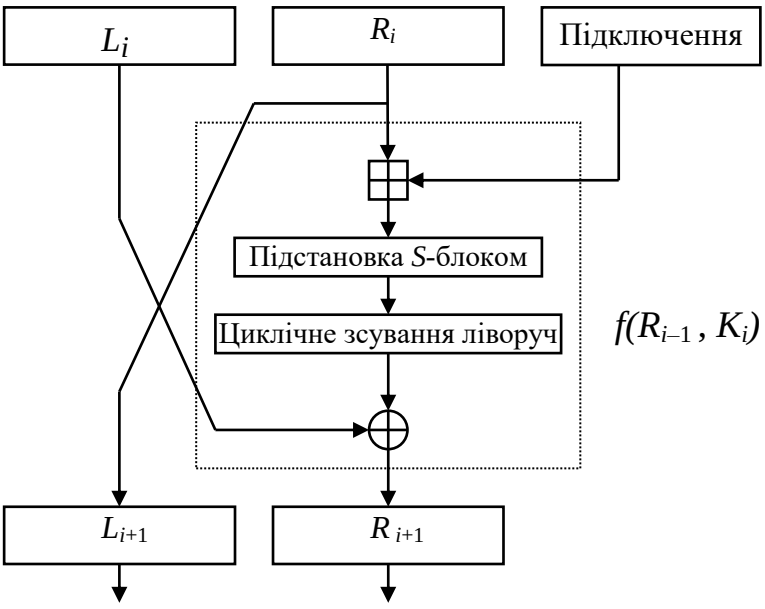
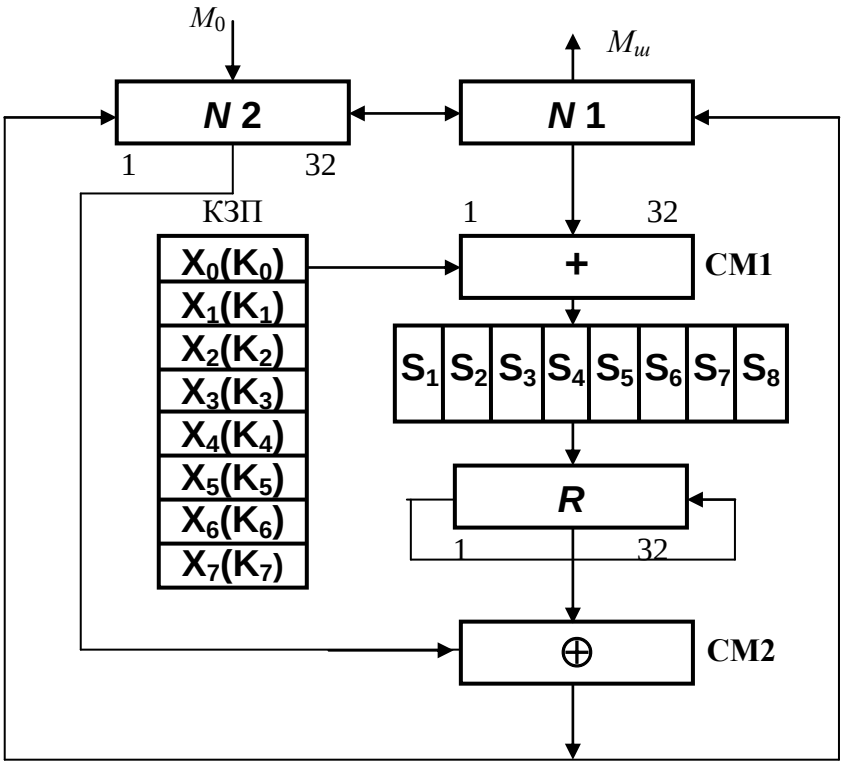


Рисунок 3.10 – Реалізація мережі Фейстеля в алгоритмі ГОСТ 28147–89

3.6.1 Режим простої заміни

Режим простої заміни передбачає розділення шифрувальної послідовності на блоки довжиною 64 розряди і є складовою частиною інших режимів шифрування. Цей режим можна використовувати лише для шифрування блоків, довжина яких є кратна 64-м бітам. У зв′язку з цим його використовують лише при передаванні наступного ключа. Процедура зашифровування в цьому режимі містить 32 цикли. До ключового запам'ятовувального пристрою вводять 256 біт ключа *K* у вигляді 8-ми 32-розрядних підключів. Зашифровування даних здійснюється за допомогою блока підстановки *S*, який складається з 8-ми вузлів заміни та регістру зсування, який виконує циклічне зсування ліворуч (на 11 розрядів) 32-розрядного вектора, здобутого з виходу блока *S*.

Схему алгоритму шифрування в режимі простої заміни зображено на рис. 3.11.



КЗП – ключовий запам'ятовувальний пристрій на 256 біт; СМ1 – 32-розрядний суматор за модулем 2^{32}; СМ2 – 32-розрядний за модулем 2; *N*1, *N*2 – 32-розрядні накопичувачі; *R* – 32-розрядний регістр циклічного зсування; *S* – 32-розрядний блок підстановки, який складається з восьми вузлів заміни S_1 ... S_8 з пам'яттю 64 біт кожен

Рисунок 3.11 – Шифрування в режимі простої заміни

Відкрите повідомлення розбивається на 64-розрядні блоки M_0.

Процедура шифрування містить 32 цикли ($j = 1, \ldots, 32$).

Перед початком шифрування до ключового запам'ятовувального пристрою вводять 256-розрядний ключ, який поділяється на блоки завдовжки в

32 розряди кожен.

$$K = K_7\, K_6\, K_5\, K_4\, K_3\, K_2\, K_1\, K_0.$$

При цьому вхідна послідовність розбивається навпіл:

$$M_0 = a_1(0),\, a_2(0),\, \ldots,\, a_{32}(0),\, b_1(0),\, b_2(0),\, \ldots,\, b_{32}(0).$$

Біти $b(0)$ – такі, що стоять ліворуч, а $a(0)$ – біти, що стоять праворуч.

Послідовність

$$a(0) = a_{32}(0),\, a_{31}(0),\, ...,\, a_0(0)$$

вводять до накопичувача $N1$, послідовність

$$b(0) = b_{32}(0),\, b_{31}(0),\, \ldots,\, b_0(0)$$

– до накопичувача $N2$.

Перший цикл процедури шифрування можна описати в такий спосіб:

$$\begin{cases} a(1) = f[a(0) \boxplus K_0] \oplus b(0); \\ b(1) = a(0). \end{cases}$$

У даному виразі $a(1)$ – заповнення $N1$ після першого циклу зашифровування; $b(1)$ – заповнення $N2$ після першого циклу зашифровування; f – функція шифрування.

Аргументом функції f є сума за модулем 2^{32} числа $a(0)$ та числа K_0. Ця функція включає дві операції над здобутою 32-розрядною сумою $(a(0) + K_0)$.

Перша операція є заміною і виконується блоком підстановки S.

Блок складається з 8-ми вузлів заміни S_1... S_8 з пам'яттю 64 біт кожен. 32-розрядний блок, який надходить від суматора, розбивають на вісім 4- розрядних груп, кожна з яких замінюється на іншу 4-розрядну групу. Кожний вузол заміни можна подати у вигляді таблиці-перестановки 16-ти 4-розрядних двійкових чисел в межах 0000…1111. Вхідний вектор – це адреса рядка в таблиці, а число в цьому рядку є вектором на виході. Порядок заміни, здійснюваної блоком S, тримається у секреті й відносно рідко змінюється.

Друга операція є циклічним зсуванням 32-розрядного вектора ліворуч на 11 розрядів. Ця операція виконується за допомогою регістру зсування R.

На наступному етапі здійснюється складання за модулем 2 результатів обчислювання функції f та вмісту накопичувача $N2$. Здобутий результат записують до накопичувача $N1$, а до накопичувача $N2$ переписують попереднє значення вектора, який містився в $N1$. На цьому завершується перший цикл зашифровування.

Решта циклів виконуються аналогічно. Відмінність полягає лише в тому, що, починаючи з 25-го циклу, ключі подаються з блока КЗП у зворотному порядку – K_7...K_0.

Отже, упродовж всіх 32-х циклів подавання ключів здійснюється в такому порядку:

$$K_0, K_1, K_2, K_3, K_4, K_5, K_6, K_7, K_0, K_1, K_2, K_3, K_4, K_5, K_6, K_7,$$
$$K_0, K_1, K_2, K_3, K_4, K_5, K_6, K_7, K_7, K_6, K_5, K_4, K_3, K_2, K_1, K_0.$$

У 32-му циклі результат з суматора СМ2 вводиться до накопичувача $N2$, а в накопичувачі $N1$ зберігаються його попереднє значення. При цьому вміст обох накопичувачів – $N1$ і $N2$ – є блоком зашифрованих даних $M_{ш}$, відповідних до блока відкритих даних M_0.

У загальному вигляді рівняння, котрі описують процес зашифровування, може бути подано у вигляді

$$\begin{cases} a(j) = f[a(j-1) \boxplus K_{j-1 \,(\mathrm{mod}\, 8)}] \oplus b(j-1) \\ b(j) = a(j-1) \end{cases} \quad \text{за } j = 1, ..., 24;$$

$$\begin{cases} a(j) = f[a(j-1) \boxplus K_{32-j}] \oplus b(j-1) \\ b(j) = a(j-1) \end{cases} \quad \text{за } j = 25, ..., 31;$$

$$\begin{cases} a(32) = a(32) \\ b(j) = f[f(31) \boxplus K_0] \oplus b(31) \end{cases} \quad \text{за } j = 32,$$

де $a(j) = a_{32}(j), a_{31}(j), ..., a_1(j)$ – заповнення $N1$ після j-го циклу зашифровування, а $b(j) = b_{32}(j), b_{31}(j), ..., b_{1(j)}$ – заповнення $N2$ після j-го циклу зашифровування;

$j = 1, 2, ..., 31.$

Блок зашифрованих даних виводиться з обох накопичувачів в такому порядку:

$$M_{ш} = [a_1(32), a_2(32), \ldots, a_{32}(32), b_1(32), b_2(32), ..., b_{32}(32)].$$

Решта блоків відкритих даних зашифровуються в режимі простої заміни аналогічно.

Процедура розшифрування має той самий вигляд, що й процедура зашифровування. Прийнятий з каналу зв'язку блок $M_{ш}$ записують до накопичувачів $N1$ та $N2$ в такий спосіб, щоби початковий вміст накопичувача $N1$ мав вигляд

$$[a_{32}(32), a_{31}(32), \ldots, a_2(32), a_1(32)],$$

а вміст накопичувача $N2$

$$[b_{32}(32), b_{31}(32), \ldots, b_2(32), b_1(32)] .$$

Розшифровування здійснюється в такий самий спосіб, що й зашифровування, проте порядок подавання 32-х розрядних ключів здійснюється в такому порядку:

$$K_0, K_1, K_2, K_3, K_4, K_5, K_6, K_7, K_7, K_6, K_5, K_4, K_3, K_2, K_1, K_0,$$
$$K_7, K_6, K_5, K_4, K_3, K_2, K_1, K_0, K_7, K_6, K_5, K_4, K_3, K_2, K_1, K_0 .$$

Рівняння розшифрування при цьому мають вигляд

$$\begin{cases} a(32-j) = f\,[a(32-j+1) \boxed{+} K_{j-1}] \oplus b(32-j+1) \\ b(32-j) = a(32-j+1) \end{cases} \quad \text{за } j = 1, \ldots, 8;$$

$$\begin{cases} a(32-j) = f\,[a(32-j+1) \boxed{+} K_{32-j(\mathrm{mod}\,8)}] \oplus b(32-j+1) \\ b(32-j) = a(32-j+1) \end{cases} \quad \text{за } j = 9, \ldots, 31;$$

$$\begin{cases} a(0) = a(1) \\ b(0) = f\,[a(1) \boxed{+} K_0] \oplus b(1) \end{cases} \quad \text{за } j = 32.$$

Здобуті після 32-х циклів роботи заповнювання накопичувачів $N1$ та $N2$ утворюють блок відкритих даних

$$M_0 = [a_1(0), a_2(0), \ldots, a_{32}(0), b_1(0), b_2(0) \ldots, b_{32}(0)].$$

Аналогічно розшифровується й решта блоків зашифрованих даних.

Введемо функцію алгоритму шифрування методом простої заміни $A(M_0) = M_{\text{ш}}$.

Режим простої заміни є сенс застосовувати для шифрування даних лише в обмежених випадках – при утворюванні ключа й зашифровуванні його із забезпеченням імітозахисту задля передавання каналами зв′язку чи задля зберігання в пам'яті комп'ютера.

3.6.2 Режим гаммування

При використовуванні шифрування в режимі гаммування вихідна послідовність, як і у попередньому випадку, розбивається на блоки завдовжки 64 біти:

$$M_0^{(1)}, M_0^{(2)}, M_0^{(3)}, \ldots, M_0^{(i)}, \ldots, M_0^{(t)},$$

де $M_0^{(i)}$ – i-тий 64-розрядний блок відкритих даних; $i = 1, \ldots, t$.

Зашифровування цих блоків здійснюється в режимі гаммування шляхом накладання на них шифрувальної гамми:

$$Г_{ш} = (Г_{ш}^{(1)}, Г_{ш}^{(2)}, \dots, Г_{ш}^{(i)}, \dots, Г_{ш}^{(t)}),$$

де $Г_{ш}^{(i)}$ – i-тий 64-розрядний блок відкритих даних; $i = 1, \dots, t$.

Кількість двійкових розрядів у блоці може бути менше за 64. В такому разі зайва частина гамми відкидається.

Рівняння шифрування має вигляд

$$M_{ш}^{(i)} = M_0^{(i)} \oplus Г_{ш}^{(i)},$$

де $Г_{ш}^{(i)} = A(Y_{i-1} + C_2, Z_{i-1} + C_1)$, $i = 1, \dots, t$; $M_{ш}^{(i)}$ – i-тий блок 64-розрядного блока зашифрованого тексту; $A(\cdot)$ – зашифровування в режимі простої заміни; C_1, C_2 – 32-розрядні двійкові константи; Y_i, Z_i – 32-розрядні двійкові послідовності.

Величини Y_i , Z_i визначаються ітераційно в міру формування гамми шифру:

$$(Y_i, Z_i) = A(S),$$

де S – синхронадсилання (64-розрядна двійкова послідовність).

Процедура гаммування здійснюється в такий спосіб (рис. 3.12).

До накопичувачів $N6$ та $N5$ наперед записуються константи, котрі мають такі значення в шістнадцятковій формі:

$$C1 = 01010104_{н} \qquad C2 = 01010101_{н}$$

До ключового запам’ятовувального пристрою КЗП вводиться 256 біт ключа, до накопичувачів $N1$ та $N2$ – 64-розрядна двійкова послідовність (синхронадсилання)

$$S = (S_1, S_2, ..., S_i, \dots, S_{64}).$$

Це синхронадсилання є початковим заповненням накопичувачів $N1$ та $N2$ для послідовного формування m блоків гамми шифру:

$$N1 \rightarrow (S_{32}, S_{31} ..., S_2, S_1) \quad N2 \rightarrow (S_{64}, S_{63}, ... , S_{32}, S_{31}).$$

$$(Y_i , Z_i) = A(Y_{i-1} + C_2, Z_{i-1} + C_1), i = 1, \dots, t.$$

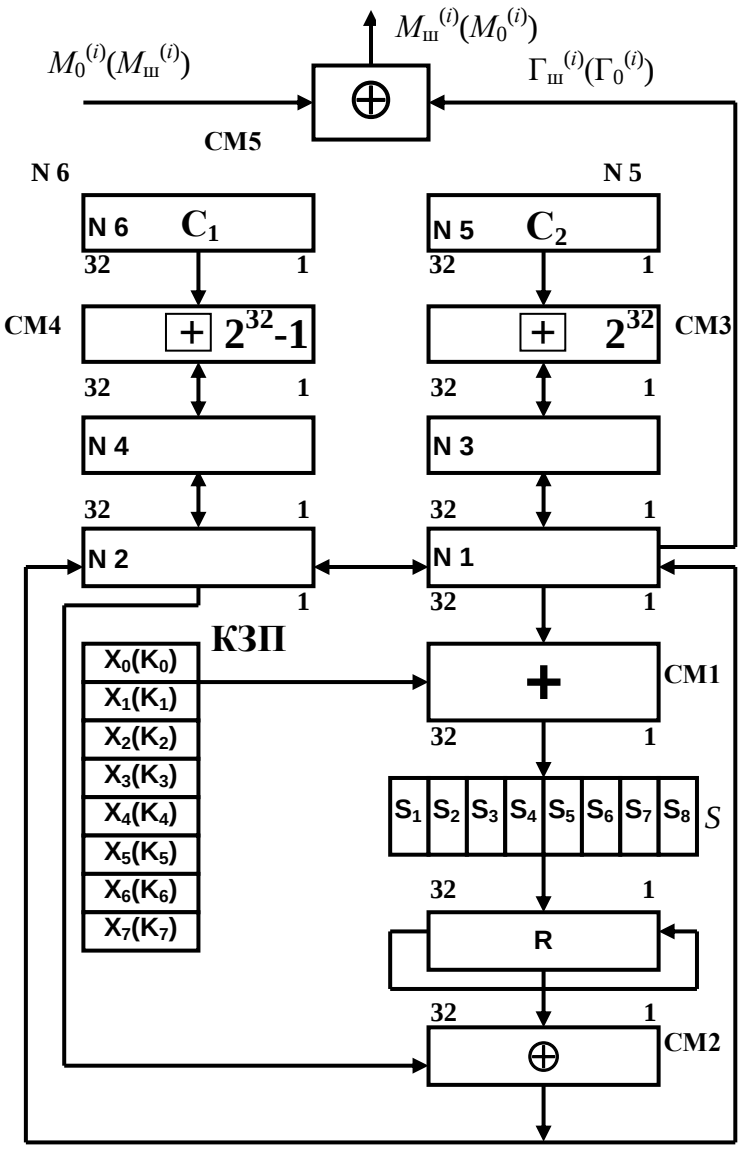
Синхронадсилання S зашифровується в режимі простої заміни, а результат її шифрування

$$A(S) = (Y_0, Z_0)$$

переписується до 32-розрядних накопичувачів $N3$ та $N4$ в такий спосіб:

$$N1 \to N3 \qquad N2 \to N4.$$

Після цього вміст накопичувачів $N6$ та $N4$ підсумовується за модулем $2^{32} - 1$ в СМ4, а вміст накопичувачів $N5$ та $N3$ підсумовується за модулем 2^{32} в СМ3. При цьому в першому випадку результат підсумовування записується до накопичувача $N4$, а в другому – до накопичувача $N3$.



КЗП – ключовий запам'ятовувальний пристрій на 256 біт;
СМ1 –32-розрядний суматор за модулем 2^{32};
СМ2 – 32-розрядний суматор за модулем 2;
СМ3 –32-розрядний суматор за модулем 2^{32};
СМ4 –32-розрядний суматор за модулем $2^{32} - 1$;
$N1$, $N2$ – 32-розрядні накопичувачі;
R – 32-розрядний регістр циклічного зсування;
S – 32-розрядний блок підстановки, який складається з 8-ми вузлів заміни $S_1 ... S_8$ з пам'яттю 64 біт кожен

Рисунок 3.12 – Схема реалізації режиму гаммування

На наступному кроці вміст накопичувача $N4$ переписується до накопичувача $N2$, а вміст накопичувача $N3$ переписується до $N1$. При цьому вміст накопичувачів $N4$ та $N3$ зберігається, а вміст накопичувачів $N1$ та $N2$ використовується як шифрувальна гамма

$$\Gamma_{ш}^{(1)} = (\gamma_1^{(1)}, \gamma_2^{(1)}, \ldots, \gamma_{63}^{(1)}, \gamma_{64}^{(1)}) ,$$

яка порозрядно підсумовується за модулем 2 в суматорі СМ5 з першим блоком послідовності

$$M_0^{(1)} = (m_1^{(1)}, m_2^{(1)}, ..., m_{63}^{(1)}, m_{64}^{(1)}) .$$

Як наслідок на виході суматора СМ5 дістають перший зашифрований блок

$$M_{ш}^{(1)} = M_0^{(1)} \oplus \Gamma_{ш}^{(1)} = (\tau_1^{(1)}, \tau_2^{(1)}, \ldots , \tau_{63}^{(1)}, \tau_{64}^{(1)}).$$

де $\tau_i^{(1)} = m_i^{(1)} \oplus \gamma_i^{(1)}$; $i = 1, \ldots , 64$.

На наступному етапі шифрування, коли криптографічному перетворюванню піддається блок $M_0^{(2)}$, зміст накопичувачів ще раз підсумовується з константами С1 і С2, які містяться в накопичувачах $N5$ та $N6$. Коли результати підсумовування знов опиняться в накопичувачах $N1$ та $N2$, вони наново шифруються в режимі простої заміни й, отже, формується нова гамма $\Gamma_{ш}^{(2)}$.

Формування подальших блоків гамми $\Gamma_{ш}^{(i)}$ здійснюється аналогічно.

Розшифровування в режимі гаммування здійснюється так само, як і зашифровування. Рівняння розшифровування має вигляд

$$M_0^{(i)} = M_{ш}^{(i)} \oplus \Gamma_{ш}^{(i)} = M_{ш}^{(i)} \oplus A(Y_{i-1} + C_2 , Z_{i-1} + C_1), \qquad i = 1, \ldots, t.$$

Процес розшифровування є можливий лише за наявності синхронадсилання, яке не є секретним числом і передається разом з повідомленням у відкритому вигляді.

3.6.3 Режим гаммування зі зворотним зв′язком

Використовування даного алгоритму, як і в попередніх випадках, припускає розділення передаваної послідовності на блоки, кратні 64-м символам:

$$M_0^{(1)}, M_0^{(2)}, M_0^{(3)}, \ldots , M_0^{(i)}, \ldots , M_0^{(t)}.$$

Зашифровування, так само як і у попередньому випадку, здійснюється шляхом підсумовування шифрованого блока з гаммою шифру

$$\Gamma_{ш} = \Gamma_{ш}^{(1)}, \Gamma_{ш}^{(2)}, \ldots , \Gamma_{ш}^{(i)}, \ldots , \Gamma_{ш}^{(t)}.$$

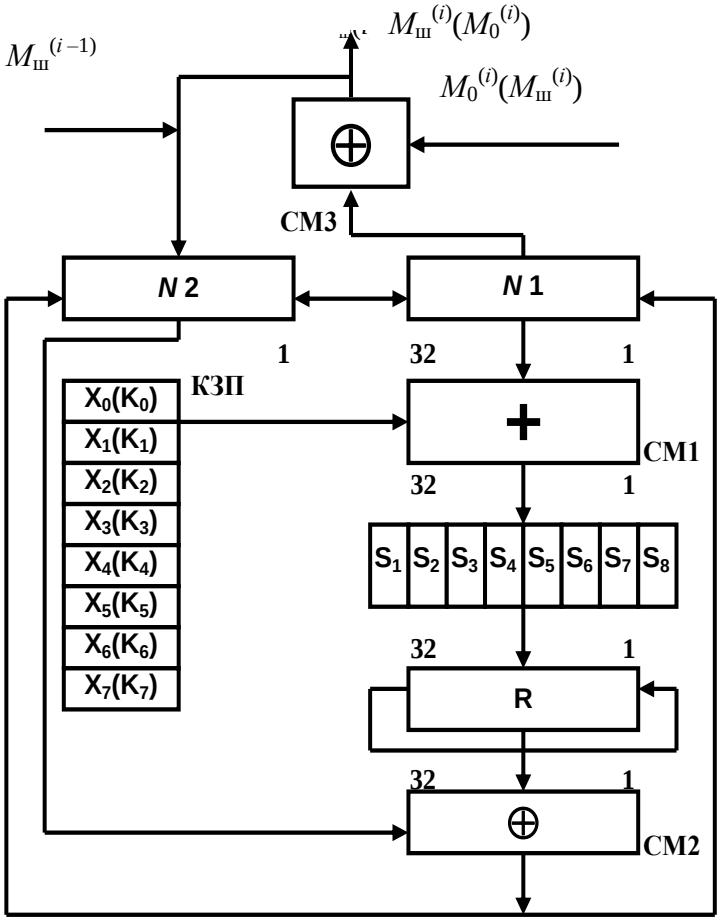
При цьому рівняння зашифровування має вигляд

$$M_{ш}^{(1)} = A(S) \oplus M_0^{(1)} = \Gamma_{ш}^{(1)} \oplus M_0^{(1)}$$

$$M_{ш}^{(i)} = A(M_{ш}^{(i-1)}) \oplus M_0^{(i)} = \Gamma_{ш}^{(i)} \oplus M_0^{(i)},\ i = 2, \ldots, t,$$

де $M_{ш}^{(i)}$ – i-тий блок, 64-розрядний блок зашифрованого тексту; $A(S)$ – зашифровування синхронадсилання в режимі простої заміни; t – визначає обсяг відкритих даних.

Аргументом $A(S)$ на першому кроці ітеративного алгоритму є синхронадсилання S, а на решті всіх етапів – попередній блок зашифрованих даних $M_{ш}^{(i-1)}$.



КЗП – ключовий запам'ятовувальний пристрій на 256 біт;
СМ1 –32-розрядний суматор за модулем 2^{32};
СМ2 – 32-розрядний суматор за модулем 2;
СМ3 – 64-розрядний суматор за модулем 2;
$N1$, $N2$ – 32-розрядні накопичувачі;
R – 32-розрядний регістр циклічного зсування;
S – 32-розрядний блок підстановки, який складається з 8-ми вузлів заміни

Рисунок 3.13 – Шифрування методом гаммування зі зворотним зв'язком

Процедура зашифровування виконується в такий спосіб. До КЗП вводиться 256 біт ключа; до накопичувачів $N1$ та $N2$ – 64-розрядна двійкова послідовність (синхронадсилання)

$$S = (S_1 , S_2 , \dots , S_i , \dots, S_{64}).$$

Це синхронадсилання зашифровується у звичайний спосіб в режимі простої заміни. Отже, після 32-х циклів опрацьовування в накопичувачах $N1$ та $N2$ буде сформовано перший 64-бітовий блок шифрувальної гамми $\Gamma_{ш}^{(1)} = A(S)$. Цей блок підсумовується за модулем 2 в суматорі СМ3 з блоком відкритих даних:

$$M_0^{(1)} = (m_1^{(1)}, m_2^{(1)}, \dots, m_{63}^{(1)}, m_{64}^{(1)}) .$$

В результаті здобувають перший блок зашифрованих даних

$$M_{ш}^{(1)} = M_0^{(1)} \oplus \Gamma_{ш}^{(1)} = (\tau_1^{(1)}, \tau_2^{(1)}, \dots , \tau_{63}^{(1)}, \tau_{64}^{(1)}) .$$

Водночас із гаммуванням по колу зворотного зв'язку блок записується до накопичувачів $N1$ та $N2$; в подальшому з нього формують чергову 64-розрядну гамму $\Gamma_{ш}^{(2)}$.

Всі подальші операції із зашифровування відкритих блоків повідомлення виконуються аналогічно. До каналу зв'язку, тобто, передаються синхронадсилання і блоки зашифрованих даних

$$M_{ш} = M_{ш}^{(1)}, M_{ш}^{(2)}, \dots , M_{ш}^{(i)}, \dots , M_{ш}^{(t)}, i = 1, \dots , t.$$

Розшифровування прийнятих з каналу зв'язку даних здійснюється у зворотному порядку. Рівняння розшифровування має вигляд

$$M_0^{(1)} = A(S) \oplus M_{ш}^{(1)} = \Gamma_{ш}^{(1)} \oplus M_{ш}^{(1)};$$
$$M_0^{(i)} = \Gamma_{ш}^{(i)} \oplus M_0^{(i)} = A(M_{ш}^{(i-1)}) \oplus M_{ш}^{(i)}, i = 2, \dots , t.$$

Процедура розшифровування здійснюється аналогічно до зашифровування. Вона розпочинається з формування першої гамми $\Gamma_{ш}^{(1)}$ з синхронадсилання S. "Дзеркальність" процедур зашифровування й розшифровування пояснюється особливістю виконання операції підсумовування за модулем 2.

3.6.4 Режим формування імітовставки

Імітовставка – це блок з P бітів, який утворюють за певним правилом з відкритих даних з використовуванням ключа, а потім додають до зашифрованих даних для забезпечення імітозахисту.

Імітозахист – це захист системи шифрування від нав'язування помилкових даних.

Імітовставка I_p утворюється з блоків відкритих даних або перед передаванням повідомлення, або водночас з його зашифровуванням.

Значення параметра P (*кількість розрядів в імітовставці*) обирають з

урахуванням того, що ймовірність нав'язування помилкових завад дорівнює $1/(2^P)$.

Перший блок відкритих даних $M_0^{(1)}$ зашифровують в режимі простої заміни $A'(\cdot)$ упродовж перших 16-ти його циклів.

Здобуте після 16-ти циклів 64-розрядне число $A'(M_0^{(1)})$ підсумовують за модулем 2 з другим блоком даних $M_0^{(2)}$. Результат підсумовування

$$(A'(M_0^{(1)}) \oplus M_0^{(2)})$$

знову піддають перетворюванню $A'(\cdot)$. Потім процедура повторюється знов. Здобуте 64-розрядне число $(A'(M_0^{(1)}) \oplus M_0^{(2)})$ підсумовують за модулем 2 з третім блоком $M_0^{(3)}$ і знову піддають перетворюванню $A'(\cdot)$ й т. д.

Останній блок $M_0^{(t)}$ (за необхідності доповнений нулями до повного 64-розрядного блока) підсумовують за модулем 2 з результатом підсумовування на кроці $(t - 1)$, після чого зашифровують в режимі простої заміни за допомогою перетворювання $A'(\cdot)$.

Зі здобутого 64-розрядного числа обирають відрізок I_p (імітовставку), яка має довжину P бітів:

$$I_p = [a^{(m)}_{(32-p+1)}(16), a^{(m)}_{(32-p+2)}(16), ..., a^{(m)}_{32}(16)],$$

де $a_i^{(m)}$ – і-тий біт 64-розрядного числа, котре здобули після 16-го циклу останнього перетворювання $A'(\cdot)$, $32 - p + 1 \le i \le 32$.

Імітовставка I_p передається каналом зв'язку наприкінці зашифрованих даних:

$$M_{\text{ш}}^{(1)}, M_{\text{ш}}^{(2)}, ..., M_{\text{ш}}^{(m)}, I_{p.}$$

Зашифровані дані, котрі було здобуто одержувачем, розшифровуються і з блоків розшифровуваного тексту утворюють імітовставку I'_p. Ця імітовставка I'_p дорівнюється до імітовставки I_p, здобутої водночас з шифртекстом. Якщо $I'_p \neq I_p$, дані, здобуті при розшифровуванні, визнаються за спотворені.

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.

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.

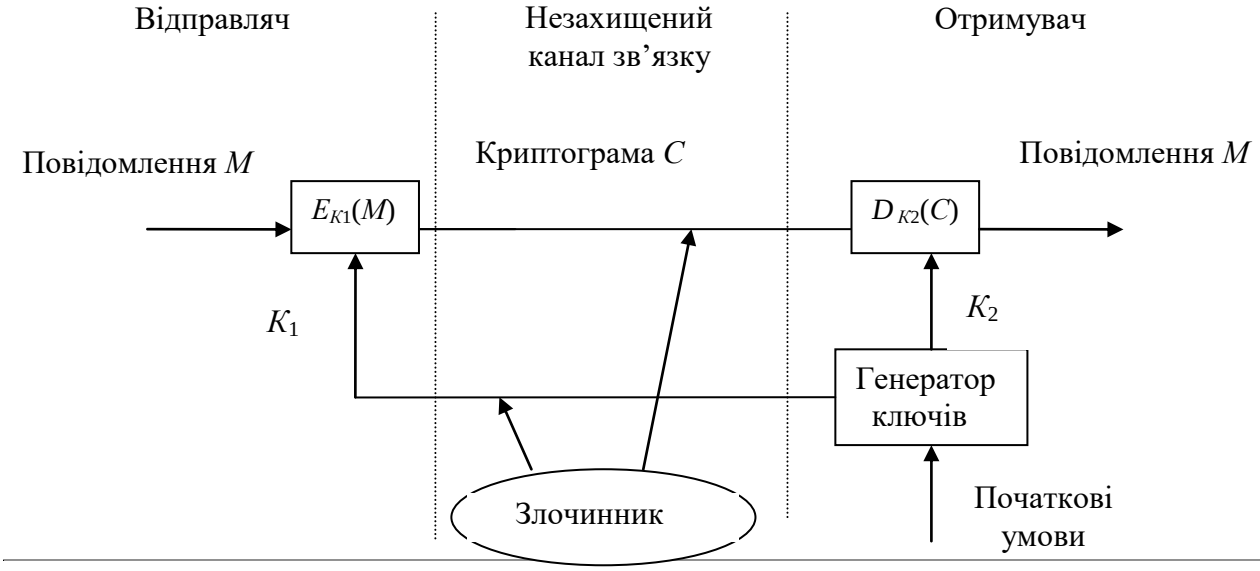
2 Алгоритми зашифровування

$$E_{K1} : M \rightarrow C$$

та розшифровування

$$D_{K2} : C \rightarrow M$$

є відкритими.



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

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 відправляч, 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 K_1 та криптограма C, за спроби обчислити вихідне повідомлення M наштовхується на задачу, яку не можна розв'язати.

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

$$f_{A,N} : Z_N \rightarrow Z_N ;$$
$$f_{A,N}(x) \equiv A^x \pmod{N},$$

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 степеня, яку використовують в криптосистемі шифрування даних RSA.

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:

$$1 < K_1 \le \varphi(N); \quad \text{НОД}\,(K_1, \varphi(N)) = 1; \quad \varphi(N) = (P-1)(Q-1),$$

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 відправлячем, а користувач B – отримувачем повідомлення. Криптосистему утворює отримувач повідомлення, тобто користувач B. Розглянемо послідовність дій користувачів A та B на прикладі.

П р и к л а д. 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

$$1 < K_1 \leq 20; \quad \text{НОД}(K_1, 20) = 1.$$

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 $K_2 = 3$:

$$M_i = C_i^{K2} (\text{mod } N) = C_i^{3} (\text{mod } 33)$$

$M_1 \equiv (9^3) (\text{mod } 33) \equiv 729 (\text{mod } 33) \equiv 3$;
$M_2 \equiv (1^3) (\text{mod } 33) \equiv 1 (\text{mod } 33) \equiv 1$;
$M_3 \equiv (29^3) (\text{mod } 33) \equiv 24389 (\text{mod } 33) \equiv 2$.

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 зреалізовуються 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.

4.4 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. Довжина шифртексту є удвічі більша за довжину вихідного повідомлення M.

Для того щоби розшифрувати шифртекст (a, b), обчислюють

$$M \equiv b/a^X \pmod P.$$

Позаяк

$$a^X \equiv G^{KX} \bmod P,$$

можна записати

$$b/a^X \pmod P \equiv Y^K M/a^X \pmod P \equiv G^{KX} M/G^{KX} \pmod P \equiv M \pmod P.$$

П р и к л а д. 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: $M = 5$.

4.5 Комбінований метод шифрування

Переваги асиметричних криптосистем полягають в тому, що вони є потенційно небезпечні – 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 захищенність інформації, передаваної 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 $K_в$ 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 табл. 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.

4.6 Проблема ідентифікації та автентифікації даних

З кожним об’єктом комп’ютерної системи (КС) пов’язана певна інформація, яка безваріантно 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 повноваг).

Згадані три операції ініціалізації є процедурами захисту й належать до одного об’єкта КС.

За обміну електронними документами мережою 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.

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. Засадничо новим розв’язком цієї проблеми є електронний цифровий підпис (ЕЦП).

Електронний цифровий підпис використовується для автентифікації текстів, передаваних телекомунікаційними каналами. Функційно він є аналогічний 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, передаваної разом з підписуваним текстом.

Система ЕЦП включає дві процедури:

1 процедуру формування підпису;

2 процедуру перевірки підпису.

У процедурі постановки підпису використовується секретний ключ відправляча повідомлення, у процедурі перевірки підпису – відкритий ключ відправляча.

При формуванні ЕЦП відправляч найперш обчислює геш-функцію $h(M)$ тексту M, який він підписує. Обчислене значення геш-функції $h(M)$ являє собою один короткий блок інформації m, котра схарактеризовує весь текст M у цілому. Потім число m зашифровується секретним ключем відправляча. 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 відправляча 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 відправляч (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 відправляч 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.

Припустімо, що відправляч 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.

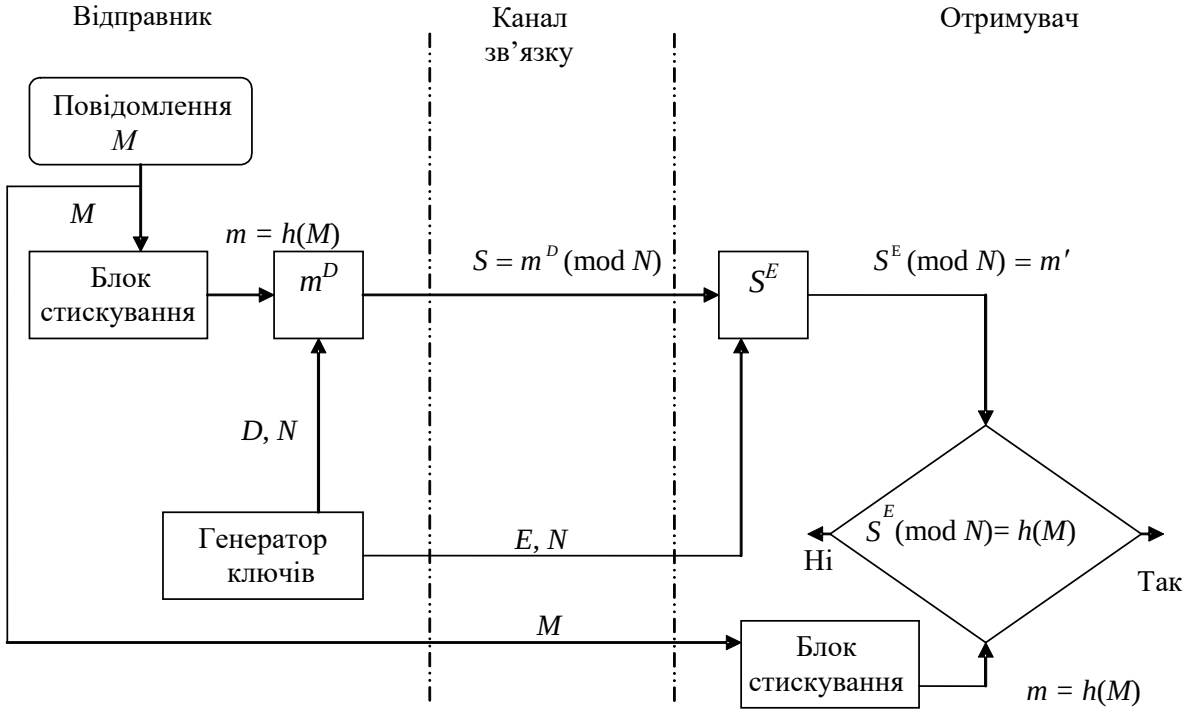


Рисунок 4.2 – Узагальнена схема цифрового підпису RSA

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, алгоритма 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 сформовувати підписи під тими документами, в яких результат гешування можна обчислити як добуток результатів гешування, вже підписаних документів.

П р и к л а д. 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,

$$S_1 \cdot S_2 (\text{mod } N) = m_1^D \cdot m_2^D (\text{mod } N) = (m_1 m_2)^D (\text{mod } N) = m_3^D (\text{mod } N) = S_3.$$

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 $G < P$. Відправляч 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.

Відправляч обирає випадкове ціле число X, $1 < X \leq (P - 1)$, і обчислює

$$Y \equiv G^X \text{ mod } P.$$

Число Y є відкритим ключем, використовуваним для перевірки підпису відправляча. Число Y відкрито передається всім потенційним одержувачам документів.

Число X є секретним ключем відправляча для підписування документів і має зберігатися в секреті.

Для того щоби підписати повідомлення M, спочатку відправляч 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 відправляч 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

$$m \equiv X \cdot a + K \cdot b(\bmod(P-1)).$$

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

$$A \equiv Y^a a^b (\bmod P)$$

і визнає повідомлення M чинним, якщо, й лише якщо

$$A \equiv G^m (\bmod P).$$

Інакше кажучи, одержувач перевіряє слушність співвідношення

$$Y^a a^b (\bmod P) = G^m (\bmod P).$$

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 відправлячем повідомлення M був власник саме даного секретного ключа X, не розкриваючи при цьому сам ключ, і що відправляч 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 відправлячем, то він зможе розкрити секретний ключ X відправляча.

П р и к л а д. Оберемо числа $P = 11$, $G = 2$ та секретний ключ $X = 8$. Обчислимо значення відкритого ключа:

$$Y \equiv G^x \bmod P \equiv 2^8 \bmod 11 \equiv 3.$$

Припустімо, що вихідне повідомлення M схарактеризовується 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:

$$a \equiv G^K \bmod P \equiv 2^9 \bmod 11 \equiv 6,$$

елемент b визначаємо, використовуючи розширений алгоритм Евкліда:

$$m \equiv X \cdot a + K \cdot b(\bmod(P-1)).$$

За $m = 5$, $a = 6$, $X = 8$, $K = 9$, $P = 11$ дістаємо

$$5 \equiv (6 \cdot 8 + 9 \cdot b)(\bmod 10)$$

або

$$9 \cdot b \equiv -43(\bmod 10).$$

Розв’язок: $b = 3$. Цифровий підпис являє собою пару: $a = 6$, $b = 3$. Далі відправляч 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) $Y^a \cdot a^b(\bmod) \equiv 3^6 \cdot 6^3(\bmod 11) \equiv 10\ (\bmod 11)$;

2) $G^m\ (\bmod P) \equiv 2^5\ (\bmod 11) \equiv 10\ (\bmod 11)$.

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.

Відправляч 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.

Відправляч вибирає випадкове ціле число X, $1 < X < q$. Число X є секретним ключем відправляча для формування електронного цифрового підпису.

Потім відправляч обчислює значення

$$Y \equiv G^x \bmod P .$$

Число Y є відкритим ключем для перевірки підпису відправляча. Число Y передається всім одержувачам документів.

Цей алгоритм також передбачає використовування однонапрямленої функції гешування $h(\cdot)$. 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

$$r \equiv (G^K \bmod P) \bmod q.$$

Потім відправляч обчислює за допомогою секретного ключа X ціле число

$$s \equiv \frac{m + r \cdot X}{K} \bmod q.$$

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

$$w \equiv \frac{1}{s} \bmod q,$$

геш-значення

$$m = h(M)$$

і числа

$$u_1 \equiv (m \cdot w) \bmod q,$$

$$u_2 \equiv (r \cdot w) \bmod q\,.$$

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 відправляч повідомлення володіє саме даним секретним ключем X (не розкриваючи при цьому значення ключа X) і що відправляч 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:

$$s \equiv \frac{m + rX}{K} (\bmod q),\ \ w \equiv \frac{1}{s} (\bmod q),$$

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.

4.8 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;
- розподілювання ключів.

Розглянемо, як вони мають зреалізовуватися для того, щоби забезпечити безпеку ключової інформації в ІС.

4.8.1 Генерування ключів

Найперш 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 Розподілювання ключів

Розподілювання ключів – найвідповідальніший процес керування ключами. До нього пред'являють такі вимоги:

1 оперативність та точність розподілювання;

2 таємність розподілювання ключів.

Розподілювання ключів поміж користувачами зреалізовуються двома методами:

1 *шляхом побудови одного чи кількох центрів розподілювання 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 перевіріти 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 розподілювання ключів, який забезпечив би:

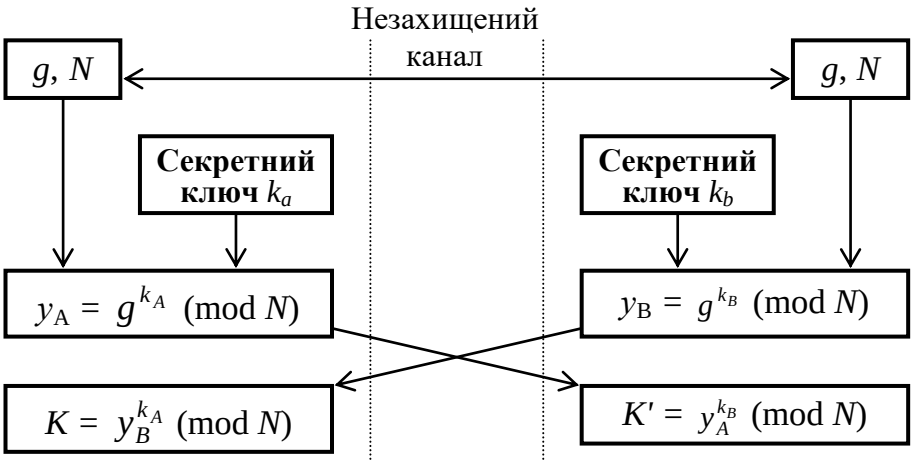
- змогу відмовитися від центра розподілювання ключів;
- взаємне потвердження правочинності користувачів;
- потвердження правочинності сеансу за допомогою механізму позначання часу механізмом, використовування для цього програмних чи апаратних засобів;

- 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.



$$K = K' \equiv g^{k_A k_B} \pmod N$$

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

$$y_A \equiv g^{k_A} \pmod N,$$

а користувач B – відкритий ключ

$$y_B \equiv g^{k_B} \pmod{N}.$$

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 $\quad K = (y_B)^{k_A} \equiv (g^{k_B})^{k_A} \pmod{N}$,

Сторона B $\quad K' = (y_A)^{k_B} \equiv (g^{k_A})^{k_B} \pmod{N}$.

При цьому $K = K'$, тому що

$$(g^{k_B}) = (g^{k_A})^{k_B}.$$

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

$$M = D_K(C) = C^{K\text{-}1} \pmod{N}.$$

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.

П р и к л а д. 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.

5 ПРИКЛАДИ РОЗВ'ЯЗУВАННЯ ЗАВДАНЬ

5.1 Розв'язування криптографічних завдань 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.

5.1.1 Шифрування за допомогою класичних шифрів перестановки

Розглянемо 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 т. д. У верхньому рядку таблиці виписано одне за одним числа від 1 до n, а в нижньому – ті ж самі числа, але в довільному порядку. Така таблиця називається підставлянням ступеня n.

Знаючи підставляння, котре задає перетворювання, можна здійснювати як зашифровування, так і розшифровування тексту.

Наприклад, якщо для перетворювання використовується підставляння

$$\begin{matrix} 1 & 2 & 3 & 4 & 5 \\ 5 & 2 & 3 & 1 & 4 \end{matrix}$$

і відповідно до нього зашифровується слово УЧЕНЬ, то виходить НЧЕЬУ.

5.1.2 Шифрування за допомогою табличних шифрів перестановки

Одним з найпримітивніших табличних шифрів перестановки є просте переставляння, для якого за ключ слугує розмір таблиці. Цей метод шифрування є подібний до шифру Сцитала.

Наприклад, 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**
Є	**О**	**Й**	**З**
Ш	**В**	**Ж**	**У**
Г	**У**	**Е**	**М**

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

МОМРА ЛАОЄО ЙЗШВЖ УГУЕМ

Природно, відправляч та одержувач повідомлення повинні заздалегідь домовитися про спільний ключ у вигляді розміру таблиці. Слід зауважити, що сполучення літер шифртексту в п’ятилітерні групи не входить до ключа шифру і здійснюється для зручності запису шифртексту. При розшифровуванні дії виконуються у зворотному порядку.

5.1.3 Шифрування за допомогою шифрів маршрутної перестановки

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. Виписуватимемо ж повідомлення за іншим маршрутом: по вертикалі, розпочинаючи з верхнього правого кута і рухаючись по черзі зверху донизу і знизу догори.

Зашифруємо, приміром, зазначеним способом фразу:

ПРИКЛАД МАРШРУТНОГО ПЕРЕСТАВЛЯННЯ

Використовуючи прямокутник розміру 4×8, дістанемо таблицю:

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

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

5.1.4 Шифрування за допомогою шифрів одиночної перестановки за ключем

Метод шифрування, називаний *одиночною перестановкою за ключем*, полягає в тім, що стовпці таблиці переставляються за ключовим словом, фразою чи набором чисел довжиною в рядок таблиці.

Застосуємо за ключ, приміром, слово ПЕЛИКАН. Заповнимо текстом повідомлення і ключовим словом таблицю:

П	Е	Л	И	К	А	Н
7	2	5	3	4	1	6
Т	Н	П	В	Е	Г	Л
Е	А	Р	А	Д	О	Н
Р	Т	И	Е	Ь	В	О
М	О	Б	Т	М	П	Ч
И	Р	Ы	С	О	О	Ь

Після переставляння дістанемо:

А	Е	И	К	Л	Н	П
1	2	3	4	5	6	7
Г	Н	В	Е	П	Л	Т
О	А	А	Д	Р	Н	Е
В	Т	Е	Ь	И	О	Р
П	О	Т	М	Б	Ч	М
О	Р	С	О	Ы	Ь	И

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 $\rightarrow$ праворуч.

У здобутій таблиці стовпці переставлено відповідно до упорядкованих номерів літер ключа. При зчитуванні її вмісту по рядках і записуванні шифртексту групами по п'ять літер дістанемо шифроване повідомлення:

ГНВЕП ЛТОАА ДРНЕВ ТЕЬИО РПОТМ БЧМОР СОЫЬИ

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. Спочатку в таблицю записується текст повідомлення, а потім по черзі переставляються стовпці, а за ними – рядки. При розшифровуванні порядок переставлянь має бути зворотним.

5.1.5. Шифрування за допомогою поворотних грат

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 шифрування на прикладі. Нехай за ключ використовуються ґрати 6×10, які наведено на рис. 5.1. Зашифруємо за їхньою допомогою текст

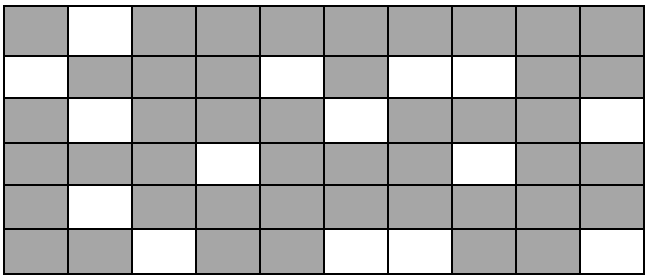


Рисунок 5.1 – Зразок ґрат

ШИФРРЕШЕТКАЯВЛЯЕТСЯЧАСТНЫМСЛУЧАЕМШИФРАМАРШРУТНОЙПЕРЕСТАНОВКИ

Наклавши ґрати на аркуш паперу, вписуємо перші п’ятнадцать (за кількістю вирізів) літер повідомлення:

ШИФРРЕШЕТКАЯВЛЯ …

Знявши ґрати, ми побачимо текст, поданий на рис. 5.2. Повернемо ґрати на 180°. У „віконечках” з'являться нові, ще не заповнені клітинки. Вписуємо в них наступні п'ятнадцять літер. Вийде запис, наведений на рис. 5.3. Потім повертаємо ґрати на інший бік і зашифровуємо залишок тексту в аналогічний спосіб (рис. 5.4 та 5.5).

	Ш								
И				Ф		Р	Р		
	Е				Ш				Е
			Т				К		
	А								
		Я			В	Л			Я

Рисунок 5.2 – Вигляд таблиці після першого кроку зашифровування

Е	Ш		Т	С			Я		
И				Ф		Р	Р	Ч	
	Е	А			Ш	С			Е
Т			Т	Н			К	Ы	
	А	М	С		Л				У
		Я			В	Л		Ч	Я

Рисунок 5.3 – Вигляд таблиці після другого кроку зашифровування

Е	Ш	А	Т	С	Е	М	Я		Ш
И	И			Ф		Р	Р	Ч	
	Е	А	Ф		Ш	С	Р		Е
Т	А		Т	Н	М		К	Ы	А
Р	А	М	С	Ш	Л	Р	У		У
	Т	Я			В	Л		Ч	Я

Рисунок 5.4 – Вигляд таблиці після третього кроку зашифровування

Е	Ш	А	Т	С	Е	М	Я	Н	Ш
И	И	О	Й	Ф	П	Р	Р	Ч	Е
Р	Е	А	Ф	Е	Ш	С	Р	С	Е
Т	А	Т	Т	Н	М	А	К	Ы	А
Р	А	М	С	Ш	Л	Р	У	Н	У
О	Т	Я	В	К	В	Л	И	Ч	Я

Рисунок 5.5 – Остаточний вигляд таблиці

Одержувач повідомлення, котрий має точно такі самі ґрати, без утруднень прочитає вихідний текст, накладаючи ґрати 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.

5.2 Розв'язування криптографічних завдань за допомогою шифрів простої заміни

Шифри заміни схарактеризовуються тим, що окремі частини повідомлення (літери, слова тощо) замінюються на певні інші літери, числа, символи і т. д. При цьому заміна здійснюється у такий спосіб, аби потім за шифрованим повідомленням можна було безваріантно відновити передаване повідомлення.

5.2.1 Шифрування одноабетковою підстановкою

Нехай зашифровується повідомлення українською мовою й при цьому заміні підлягає кожна літера повідомлення. 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. 5.1 є ключем шифру заміни. Знаючи її, можна здійснювати як зашифровування, так і розшифровування.

Таблиця 5.1 – 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. 5.2) можна дістати різноманітні варіанти зашифрованого повідомлення для одного й того самого відкритого повідомлення. Наприклад, якщо ключем є табл. 5.2, то повідомлення «мені відомі шифри заміни» може бути зашифровано кожним з трьох способів, поданих в цій таблиці.

Оскільки множини $M_а$, $M_б$, $M_в$, …, $M_я$ 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.

Таблиця 5.2 – 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*	*ї*	*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

м е н і в і д о м і ш и ф р и з а м і н и

08	24	54	65	14	34	31	53	32	92	09	61	89	29	90	30	40	08	65	27	46

32	83	77	34	71	65	01	68	61	92	68	08	20	66	90	73	40	61	34	77	02

61	70	27	34	63	92	15	09	08	65	68	32	20	80	02	55	10	32	92	54	90

5.2.2 Шифрування за допомогою квадрата Полібія

Як вже зазначалося вище, за один з перших шифрів простої заміни вважається так званий *полібіанський квадрат*. За два сторіччя до нашої ери Полібій винайшов для цілей зашифровування квадратну таблицю розміром 5×5, заповнену літерами грецької абетки у довільному порядку (рис. 5.6).

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 5.6 – 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.

5.2.3 Зашифровування за рахунок перетворювання числового повідомлення на літерне

Розглянемо 30-літерну абетку української мови:

АБВГДЕЄЖЗИІКЛМНОПРСТУФХЦЧШЩЬЮЯ

У поданій тут абетці відсутні літери 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. 5.3.

Таблиця 5.3 – 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.

5.2.4 Розшифровування повідомлення за відомою умовою шифрування

Для зашифрованого повідомлення з n літер обирається ключ K – певна послідовність з n літер абетки, наведеної в табл. 5.3. Зашифровування кожної літери повідомлення полягає в додаванні її номера в таблиці з номером відповідної літери ключової послідовності й заміні дістаної суми на літеру абетки, номер якої має ту саму остачу від ділення на 30, що й ця сума.

Необхідно прочитати шифроване повідомлення: РБЬНПТИТСРРЕЗОХ, якщо відомо, що шифрувальна послідовність не містила жодних літер, окрім А, Б та В.

Кожну літеру шифрованого повідомлення розшифруємо у трьох варіантах, припускаючи послідовно, що відповідна літера шифрувальної послідовності є літера А, Б чи В (табл.5.4)

Таблиця 5.4 – Три варіанти розшифровування шифрованого повідомлення

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

Обираючи з кожного стовпчика таблиці 5.4 лише по одній літері, віднайдемо осмислене повідомлення НАШКОРЕСПОНДЕНТ, яке й є шуканим.

Зауваження. З дістаної таблиці можна було б віднайти не одне осмислене повідомлення. Кількість решти різноманітних варіантів вихідних повідомлень без обмежень на осмисленість дорівнює 3^{15}, або 14348907, тобто понад 14 мільйонів!

5.2.5 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. 5.5, в якій зазначено, на яку саме літеру треба замінити кожну літеру російської абетки A.

Таблиця 5.5 – Зашифровування літер за допомогою шифру простої заміни

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

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

Якщо слово СРОЧНО зашифрувати простою заміною за допомогою ключа таблиці 5.5, то вийде слово ВЗДАБД. Зашифрувавши здобуте слово за допомогою того самого ключа вдруге, здобудемо слово ЮШЫЧЯЫ.

Слід визначити, 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 СРОЧНО, збігається з найменшим номером циклу зашифровування, котрий дає це початкове слово.

Оскільки літера С повторюється в кожнім циклі зашифровування, номер якого дорівнює 5, а літери Р, О, Ч, Н – у кожнім циклі, номери яких є кратні до 13, 7, 2 та 3 відповідно, то слово СРОЧНО з'явиться вперше в циклі з номером, дорівнюваним

$$\text{НСК}\,(2, 3, 5, 7, 13) = 2 \cdot 3 \cdot 5 \cdot 7 \cdot 13 = 2730,$$

де НСК – найменше спільне кратне.

5.2.6 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:

$$E_{a,b} : \overline{Z}_m \rightarrow \overline{Z}_m,$$
$$E_{a,b} : t \rightarrow E_{a,b}(t),$$
$$E_{a,b}(t) = at + b \pmod m,$$

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

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. 5.6.

Таблиця 5.6 – 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

Якщо перетворити числа на літери англійської мови, дістанемо відповідність для літер відкритого тексту та шифртексту, подану в табл. 5.7.

Таблиця 5.7 – 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

Вихідне повідомлення HOPE перетворюється на шифртекст AVYR.

Достоїнством афінної системи є зручне керування ключами: ключі зашифровування й розшифровування подаються в компактній формі у вигляді

пари чисел (*a*, *b*). Недоліки афінної системи є аналогічні до недоліків системи шифрування Цезаря.

5.2.7 Шифрування за допомогою системи Цезаря з ключовим словом

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
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 ми маємо підставляння для кожної літери довільного повідомлення.

Вихідне повідомлення

SEND MORE MONEY

шифрується як

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. Приміром, ключова фраза

КАК ДЫМ ОТЕЧЕСТВА НАМ СЛАДОК И ПРИЯТЕН

і число $k = 3$ породжують таблицю підставляння, подану табл. 5.8.

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.

Таблиця 5.8 – Зашифровування літер за допомогою ключової фрази у шифрі простої заміни

0			З												
А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
Ъ	Э	Ю	К	А	Д	Ы	М	О	Т	Е	Ч	С	В	Н	Л

Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ь	Ы	Ъ	Э	Ю	Я
И	П	Р	Я	Б	Г	Ж	З	Й	У	Ф	Х	Ц	Ш	Щ	Ь

5.2.8 Шифрування за допомогою таблиці Трисемуса

Для здобуття шифру заміни Трисемуса зазвичай використовувалися таблиці для записування літер абетки й ключове слово. У таблицю спочатку вписувалося по рядках ключове слово, причому повторювані літери відкидалися. Потім ця таблиця доповнювалася літерами абетки, які не увійшли до неї одна за одною.

Для української абетки таблиця Трисемуса може мати розмір 4×8. Оберемо за ключ слово БАНДЕРОЛЬ. Розглянемо шифрувальну таблицю з ключем, поданим у табл. 5.9.

Таблиця 5.9 – Шифрувальна таблиця 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.

Наприклад, при зашифровуванні за допомогою цієї таблиці повідомлення

ДІЛОМАЙСТРАВЕЛИЧАЄ

дістаємо шифртекст

ЄУІИЧВХЩЮЗВЙЖІТДВМ

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

5.2.9 Шифрування за допомогою біграмного шифру Плейфейра

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

У цілому структура таблиці системи шифрування Плейфейра є цілком аналогічна до структури таблиці Трисемуса. Тому для пояснення процедур

зашифровування й розшифровування в системі Плейфейра скористаємося табл. 5.9.

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 (як, наприклад, літери А та И у табл. 5.9), тоді відшукують літери в кутах прямокутника, визначуваного даною парою літер. (У нашому прикладі це літери АИОВ. Пара літер АИ відбивається в пару ОВ.) Послідовність літер у біграмі шифртексту має бути дзеркально розташована стосовно послідовності літер у біграмі відкритого тексту;

– якщо обидві літери біграми відкритого тексту належать до одного стовпця таблиці, то за літери шифртексту вважаються літери, котрі розміщено під ними (наприклад біграма НК дає біграму шифртексту ГЦ). Якщо при цьому літера відкритого тексту перебуває в нижньому рядку, то для шифртексту береться відповідна літера з верхнього рядка того самого стовпця (наприклад біграма ВХ дає біграму шифртексту ЙА);

– якщо обидві літери біграми відкритого тексту належать до одного рядка таблиці, то за літери шифртексту вважаються літери, котрі розміщено праворуч від них (наприклад біграма НО дає біграму шифртексту ДЛ). Якщо при цьому літера відкритого тексту перебуває в крайньому правому стовпці, то для шифру беруть відповідну літеру з лівого стовпця в тому самому рядку (наприклад біграма ХЯ дає біграму шифртексту ЦФ).

Зашифруємо текст:

ВСЯКИЙ СВОГО ЩАСТЯ КОВАЛЬ

Розбиття цього тексту на біграми дає

ВС ЯК ИЙ СВ ОГ ОЩ АС ТЯ КО ВА Л Ь

Дана послідовність біграм відкритого тексту перетворюється за допомогою шифрувальної таблиці (див. табл. 5.9) на таку послідовність біграм шифртексту:

ЗЙ ЦУ ВТ ЙЗ НИ РЮ РЙ УЮ ТН ЙВ БІ

При розшифровуванні застосовується зворотний порядок дій.

5.3 Розв’язування криптографічних завдань за допомогою шифрів складної заміни

5.3.1 Шифрування за допомогою таблиці Віженера

Нехай за ключове обране слово *АМБРОЗІЯ*. Необхідно зашифрувати повідомлення *ПРИЛІТАЮ СЬОМОГО*.

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 Б).

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

5.3.2 Шифрування за допомогою шифру Гронсфельда

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 Б).

Приміром, застосовуючи як ключ групу з чотирьох початкових цифр числа *e* (основу натурального логарифму), а саме 2718, здобуваємо для відкритого повідомлення *БЕЗ ОХОТИ НЕМА РОБОТИ* такий шифртекст :

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.

5.3.3 Шифрування за допомогою подвійного квадрата

1854 року 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. 5.7), 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.

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

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.

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:

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.

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 вихідного 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 шифр-тексту з тією самою послідовністю ключів k:

$$x = y \oplus k = x \oplus k \oplus k.$$

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 гаммування.

6 ДЕШИФРУВАННЯ КЛАСИЧНИХ ШИФРІВ

6.1 Дешифрування шифру простої заміни

Уведемо математичні моделі відкритих текстів, використовувані при криптографічному аналізуванні шифрів, а також окреслимо певні частісні властивості відкритих текстів. Далі використаємо деякі з цих властивостей і акцентуємо увагу на ймовірнісно-статистичних закономірностях, які мають місце у відкритому тексті й використовуються при дешифруванні шифрів простої заміни й переставляння.

Існують стійкі закономірності відкритого тексту, які слід враховувати при дешифруванні шифрів простої заміни й переставляння. Можливість дешифрування певного шифру значною мірою залежить від того, наскільки криптографічні перетворення руйнують ймовірнісно-статистичні закономірності, наявні у відкритому тексті. До найбільш стійких закономірностей відкритого повідомлення належать такі:

1) в осмислених текстах кожної природної мови різні літери зустрічаються з різною частістю, при цьому відносні частості вживання літер у різних текстах однієї мови є близькі поміж собою. Те ж саме можна стверджувати й про частості вживаності пар, трійок літер відкритого тексту;

2) будь-яка природна мова має так звану надлишковість, що дозволяє з великою ймовірністю "вгадувати" зміст повідомлення, навіть якщо частина літер у повідомленні є невідома.

У табл. 6.1 зазначено відносну частість вживаності літер абетки російської мови.

Таблиця 6.1 – Частість вживаності літер абетки російської мови

1	а - 0,062	11	к - 0,028	21	ф - 0,002
2	б - 0,014	12	л - 0,035	22	х - 0,009
3	в - 0,038	13	м - 0,026	23	ц - 0,004
4	г - 0,013	14	н - 0,053	24	ч - 0,012
5	д - 0,025	15	о - 0,090	25	ш - 0,006
6	е - 0,072	16	п - 0,023	26	щ - 0,003
7	ж - 0,007	17	р - 0,040	27	ы - 0,016
8	з - 0,016	18	с - 0,045	28	ъ,ь - 0,014
9	и - 0,062	19	т - 0,053	29	э - 0,003
10	й - 0,010	20	у - 0,021	30	ю - 0,006
				31	я - 0,018

Подібні таблиці наводяться в багатьох джерелах. Їх здобуто на підставі підрахунків частостей вживаності певних літер на великих обсягах відкритого

тексту. З огляду на те, що для експериментів залучається різний вихідний матеріал, значення ймовірностей дещо відрізнюються поміж собою.

Якщо упорядкувати літери за спаданням імовірностей, то дістанемо варіаційний ряд

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

Як запам'ятати перші 10 найчастіш вживаних літер російської абетки? Пригадуєте, як запам'ятовують основні кольори у фізиці? Треба запам'ятати фразу: *Каждый охотник желает знать, где сидит фазан.* Перші літери слів фрази вказують на основні кольори. У криптографії треба запам'ятати слово СЕНОВАЛИТР — у ньому наявні всі 10 найбільш вживаних літер.

Частісна діаграма вживаності літер, зрозуміло, залежить від мови. У табл. 6.2 подано у відсотках відносні частості найбільш уживаних літер деяких мов.

Таблиця 6.2 – Відносні частості найбільш уживаних літер деяких мов

Мова	Частість вживаності літер, %					
Англійська	e – 12,75	t – 9,25	r – 8,50	i – 7,75	h – 7,75	o – 7,50
Французька	e – 17,75	a – 8,25	s – 8,25	i – 7,25	n – 7,25	r – 7,25
Німецька	e – 18,50	n –11,50	I – 8,00	r – 7,50	s – 7,00	a – 5,00
Арабська	a – 17,75					
Грецька	a – 14,25					
Японська	p – 15,75					
Латина	a – 11,00					
Малайська	a – 20,25					
Санскрит	a – 31,25					

Частості вживаності знаків абетки залежать не лише від мови, але й від характеру тексту. Приміром, у тексті з криптографії буде підвищено ймовірність використання літер Ф, Ш (через часто вживані слова "шифр", "криптографія"). У математичному тексті ймовірніше буде завищено частість використання літери Ф (через слова "функція", "функціонал" і т. п.).

У стандартних текстових файлах найчастіше вживано символ "прогалина"; в *exe*-файлах найчастіше вживано символ 0; в текстах, написаних у текстовому процесорі ChiWriter, зручному для оформлення математичних текстів, перше місце посів символ "\" – backslash.

Частісна діаграма є стійкою характеристикою тексту. З теорії ймовірностей випливає, що за вельми слабких обмежень на ймовірнісні властивості випадкового процесу є справедливий закон великих чисел, тобто відносні частості V_k знаків збігаються за ймовірністю зі значеннями їхніх імовірностей P_k:

$$P\left\{\left|\frac{\vartheta_k}{N} - p_k\right| > \varepsilon\right\} \xrightarrow[N\to\infty]{} 0.$$

Це є слушно для послідовності незалежних випробувань, для кінцевого регулярного однорідного ланцюга Маркова. Експерименти засвідчують, що це є слушно й для відкритих текстів.

З позицій сучасної криптографії, шифри переставляння і простої заміни мають істотний недолік: вони не цілковито руйнують ймовірнісно-статистичні властивості, наявні у відкритому повідомленні.

При дешифруванні тексту, зашифрованого шифром простої заміни, використовують частісні характеристики відкритого тексту. Власне, якщо підрахувати частості вживаності знаків у шифрованому тексті, упорядкувати їх за спаданням та порівняти з варіаційним рядом імовірностей відкритого тексту, то ці дві послідовності будуть близькі. Найімовірніше перше місце посяде прогалина, далі слідуватимуть літери О, Е, А, И.

Зрозуміло, якщо текст не є надто довгий, то не є неодмінним повний збіг. Може опинитися на другому місці О, а на третьому – Е, але в кожному разі в перших і других рядах однакові літери розташовуватимуться неподалік одна від одної, й щоближче до початку (щобільше ймовірність вживання знаків), тим менше буде відстань поміж знаками.

Аналогічна картина спостерігається і для пар сусідніх літер (біграм) відкритого тексту (найчастіше зустрічається біграма російського відкритого тексту – СТ). Однак для здобуття стійкої картини довжина послідовності має бути істотно більше. На порівняно невеликих відрізках відкритого тексту ця картина є трохи розмита. Більш стійкою характеристикою біграм є відсутність в осмисленому тексті, як говорять, певних біграм; наявність заборонних біграм, котрі мають імовірність, дорівнювану практично нулю.

Чи бачили ви коли-небудь у відкритому тексті біграми ЪЬ, "голосна" Ь, "прогалина" Ь? Знання й використовування зазначених особливостей відкритого тексту значно полегшує дешифрування шифрів переставляння й заміни.

Розглянемо приклад дешифрування шифру простої заміни.

Нехай маємо такий шифртекст:

ДОЧАЛЬ ИЬЦИО ЛИОЙО ВНЫИЮШ ХЕМВЛНХЕИ ДОСОЛЬ ЧСО ИА ТЬЖАТСР ЬАС АКЕИОЙО ДОКЩОКЗЖАЙО КПЗ РТАЩ ТПЬЧНАР ТДО-ТОУН ХЕМВОРНИЕЗ ЕИМОВЛНЯЕЕ РЮУОВ БВЕД СОЙВНМЕЧАТБОГ ' ТЕТСАЛЮ ЫНРЕТЕС ОС ОТОУ АИИОТСАГ ЕИМОВЛНЯЕЕ АА ЯАИИ-ОТСЕ Е РОЫЛОЦИОТСАГ РПНКАПШЯАР ДО ЫНЖУСА ТРОАГ ЕИ-МОВЛНЯЕЕ ДВАЦКА РТАЙО ДОКЧАВБИАЛУОПШХОА ВНЫИООУ ВНЫЕА РЕКОР ЫНЖЕЖНАЛОГ ЕИМОВЛНЯАА КОБЬЛАИСНПШИНЗ САПАМОИИНЗ САПАРЕЫЕОИИНЗ БОЛДШЭСАВИНЗ БНЦКЮГ РЕК ЕИМОВЛНЯЕЕ УЛААС ТРОЕ ТДАЯЕМЕЧАТБЕА ОТОУАИИОТСЕ Е ФСК ОТОЧАИИОТСЕ ТЕПШИО РПЕЗЭС ИН РЮУОВ ЛАСОКОР ХЕМВОРЙЙВЗ ЕИМОВЛНЯЕЕ УОПШХОА ЫИНЧАИЕА ЕЛАЭС ОУЪАЛЮ Е СВАУЬАЛНЗ ТБОВОТСШ ДАВАКНЧЕ ХЕМВОРНИИОГ

Роботу слід розпочати з підрахунку частості вживаності символів у

шифрованому тексті. Після того як здійснено підрахунок, упорядкуємо символи за спаданням частостей:

$$b_1\ b_2\ b_3\ b_4\ b_5\ \ldots$$

Під цим рядом слід підписати варіаційний ряд імовірностей знаків у відкритому тексті:

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

За вельми великої довжини шифртексту, для того аби з шифрованого здобути відкритого тексту, стане замінити b_1 на 0, b_2 – на Е, b_3 – на А й т. д. Принаймні саме така ситуація матиме місце для найбільш імовірних літер. У нас матеріалу недостатньо. Переглянувши шифртекст після такої заміни, бачимо, що він не читається, – отже, матеріалу насправді є замало.

Що нам залишається? Вгадувати заміну. При цьому маємо все ж враховувати статистичні особливості відкритого тексту. У шифртексті через прогалину найімовірніше позначено прогалину; через літеру О – О чи А; через Е – О чи Е, чи А; через А – Е чи А, чи И й т. д.

Можна рекомендувати виписати шифртекст, а під ним у стовпець – найбільш імовірні заміни для цих літер. У нашому прикладі заміну добрано у такий спосіб, що літеру абетки замінено саме на найбільш імовірне для неї позначення. Тому сам шифртекст у даному прикладі виписувати немає потреби. Він збігається із середнім рядком послідовності стовпців найбільш імовірних замін.

Що рідше зустрічається літера, тим більшої глибини треба брати стовпець, аби була впевненість, що в стовпці міститься знак відкритого тексту. У нашому випадку стовпці взято однаковою глибиною в п'ять символів, але виписано їх лише для найчастіше вживаних літер.

При дешифруванні без використання засобів автоматизації далі треба вгадувати заміну. Якщо пильно придивитися до тексту, то зробити це не є вельми складно. В тексті є слово АА з двох часто вживаних літер. Що це може бути за слово? У російській мові немає слів ОО, ИИ, НН тощо. Так, перебираючи можливі слова, ми віднайдемо одне слово ЕЕ і дійдемо висновку, що літеру Е було замінено на А. Надто часто в шифртексті слова закінчуються біграмами ЕЕ. В російській мові типовими закінченнями є сполучення ЕЕ, ИИ. З огляду на те, що заміну для літери Е ми вже вгадали, дійдемо висновку, що в шифртексті літери И замінено на Е. Тоді усюди в шифртексті можна провести зворотну заміну. Тепер ми вже можемо вгадувати окремі слова. У такий спосіб, добряче поламавши голову, ми, як у грі "Поле чудес", зрештою відновимо увесь текст.

У стовпцях найбільш імовірних замін літери, котрі відповідають слушним зворотним замінам, має бути позначено як великі. Прочитавши відкритий текст, ви переконаєтесь, що він являє собою одну з кількох пропозицій з книги С.А. Дориченка та В.В. Ященка «25 этюдов о шифрах».

Для дешифрування в автоматизованому режимі насамперед слід завести до пам'яті комп'ютера словник відповідної мови.

Програма дешифрування повинна, переглядаючи шифртекст, здійснювати спробні зворотні заміни, у припущенні, що на даному фіксованому місці у відкритому тексті перебувало перевірюване слово. Після кожної заміни програма частково відновлює ключове підставляння, частково розшифровує текст і відкидає варіант підставляння, якщо в певному місці відновленого тексту виникає літеросполучення, якого не може бути у відкритому тексті розглядуваної мови. Після відновлення певної кількості замін у тексті виникають ділянки, в яких віднайдено значну кількість літер. Решта літер добираються шляхом перебирання словника і підставляння до тексту слів, які не суперечать відновленим попередньо замінам.

Слід зазначити, що практичних навичок з дешифрування шифру простої заміни можна набути лише після проведення самостійних дослідів з дешифрування.

6.2 Дешифрування шифру переставляння

Для відновлення відкритого тексту шифру переставляння слід переставити стовпці у такий спосіб, аби в рядках з'явився осмислений текст.

Розглянемо приклад дешифрування шифру переставляння восьми стовпців.

Нехай шифртекст має вигляд, наведений в табл. 6.3.

Таблиця 6.3 – Шифртекст російською мовою

1	2	3	4	5	6	7	8
п	а	я		в		и	м
0	ч	ш	г		У		е
е	б	ж	л		е		0
м		ч	–	0	т	0	я
	е	г	е		У	с	Щ
	а	к	ь	З	а	т	т
я	Р	е		е	п		ь
	ю	З	в	а	н	в	
0	и	а	в	е	ш	л	
	е	е	я	м		п	н
ь	Р	Р	н	З	е	е	е
З	а	м	а	н		а	к
ч	с	т	а		ь	а	н
0	я	л	м		а	л	
0	ь	ч	х	т	а	т	
в		е	0	а	я	е	я
0	е	Р	м	т	ь	е	
д	с	г	ы		0	а	т

е	б	в	н		ы		
	а	у	и	н	З	н	л
	г	и	а	0	к	к	д
	а	0	б	д	г	н	
	ж	а	У	е	д	я	д
х	л	и		е	м	0	а
к	Р	т	д		ь	0	е
	ь	х	в	т	0	н	
р	л	е		е	д	а	ю
р		З	е	в		е	д
ш		в	а	е	н	е	н
т	и	и	е	в			д
		в		с	д		

Зіставимо переставлянню стовпців таблицю 8×8; при цьому поставимо на перетинанні *i*-того рядка й *j*-того стовпця одиницю, якщо *j*-тий стовпець після зворотного переставляння має слідувати за *i*-тим. Наше завдання – відновити таблицю, відповідну слушному переставлянню стовпців.

Далі попарно прилаштовуватимемо один стовпець до одного. Якщо при цьому в певних рядках виникатимуть заборонні біграми, то стовпці не зможуть у відкритому тексті слідувати один за одним, – й відповідна клітинка закреслюється. В нашому прикладі шостий стовпець не може слідувати за четвертим, тому що інакше в тексті в першому рядку йтимуть дві прогалини поспіль. Переглянемо, наприклад, шостий рядок. Якби четвертий стовпець слідував за першим, то в тексті були б слова, які розпочинаються з Ь. Після переглядання всіх рядків ми здобудемо табл. 6.4.

Таблиця 6.4 – Дешифрування шифру переставляння

	1	2	3	4	5	6	7	8
1	X	X		X	X	X		X
2		X		X		X		
3			X					X
4	X	X		X		X	X	X
5	X				X	X	X	X
6	X	X		X		X	X	
7	X			X	X	X	X	X
8	X	X			X	X	X	X

Якби текст був довшим і мав більше рядків, то в кожнім рядку й у кожнім стовпці залишилося б лише по одній незакресленій клітинці – і переставляння було б відновлено. Стосовно нашої таблиці ми можемо лише стверджувати, що шостий стовпець слідує за третім (позначимо цю подію як 3 $\rightarrow$ 6), якщо шостий стовпець не є останнім. Для шостого стовпця можливі два

варіанти продовження:

$$\begin{array}{l} 8 \\ \downarrow \\ 3 \to 6 \to 5 \end{array}$$

Нам треба розглянути обидва варіанти й спробувати відкинути хибний. Якщо це не вдасться, то слід продовжувати обидва варіанти:

$$\begin{array}{ll} 8 \to 4 & 1 \\ \uparrow & \uparrow \\ 3 \to 6 \to 5 \to 2 & \to 7 \end{array}$$

У підсумку здобуваємо певне дерево можливого слідування стовпців у відкритому тексті:

$$\begin{array}{l} \qquad\qquad 7 \\ \qquad\qquad \downarrow \\ 3 \to 6 \to 5 \to 2 \to 8 \to 4 \to 1 \to 7 \\ \quad \downarrow \ \downarrow \ \downarrow \\ \quad 8 \ \ 4 \ \ 1 \to 7 \\ \quad \downarrow \ \downarrow \\ \quad 4 \ \ 1 \to 7 \\ \quad \downarrow \\ \quad 5 \\ \quad \downarrow \\ \quad 2 \to 7 \\ \quad \downarrow \\ \quad 1 \to 7 \end{array}$$

Кожній гілці дерева відповідає певне переставляння стовпців.

Далі перевіряємо кожен варіант на осмисленість і дістаємо слушний варіант:

$$3 \to 6 \to 5 \to 2 \to 8 \to 4 \to 1 \to 7$$

Зауважимо, що не обов'язково було будувати дерево до кінця. Наприклад, гілку $3 \to 6 \to 8 \to 4 \to 5$ можна було відкинути одразу. Хіба можна визнати за осмислений фрагмент тексту

З	6	8	4	5
я		м		в
ш	У	е	г	
ж	е	0	л	
ч	т	я		0
г	У	Щ	е	З
к	а	т	ь	е
е	а	т		а

Така процедура відкидання гілок була б необхідна, якби рядків було дещо менше й дерево було, відповідно, набагато більш розгалуженим. Запропоновану процедуру легко автоматизувати і зробити придатною для зреалізовування на ЕОМ.

Алгоритм дешифрування має складатися з таких етапів.

1 Попередня робота. Аналізуючи вельми потужний обсяг відкритих текстів, побудувати множину заборонних біграм.

2 Попередня робота. Скласти словник усіляких *v*-грам для $v = 2, 3, ..., d$, які можуть зустрітися у відкритому тексті. Число d обирається виходячи з можливостей обчислювальної техніки. Побудувати таблицю 8×8. При цьому перебираються послідовно всі помітні біграми і для кожної опробуваної біграми – послідовно всі рядки. Якщо хоча б в одному рядку перший символ біграми зустрічається в i-тому стовпці, а другий – у j-тому, то клітинка $i \times j$ таблиці закреслюється.

3 Обрати певний стовпець за вихідний.

4 Розпочати процедуру побудови дерева шляхом прилаштовування до вихідного стовпця усіх варіантів стовпців.

5 Для кожного здобутого варіанта додати ще один з решти стовпців. Якщо хоча б в одному з рядків таблиці зустрінеться триграма, відсутня у словнику розміщених триграм, – то варіант відкидається.

6 Для кожного з невідкинутих варіантів додаємо ще одного стовпця і провадимо відкидання хибних варіантів за словником дозволених чотириграм.

Якщо словник побудовано лише для $d \leq 3$, то відкидання провадиться шляхом перевірки на припустимість триграм, які зустрілися в останніх трьох стовпцях кожного рядка. Продовжуємо цей процес до здійснення цілковитого переставляння. Нижче наведено відновлений для нашого прикладу текст.

Таблиця 6.5 – Відновлений текст

	1	2	3	4	5	6	7	8
1	я		в	а	м		п	и
2	ш	у		ч	е	г	0	
3	ж	е		б	0	л	е	
4	ч	т	0	1	я		М	0
5	г	У		е	Щ	е		с
6	к	а	З	а	т	ь		т

7	е	п	е	Р	ь		я	
8	З	н	а	ю		в		в
9	а	ш	е	и		в	0	л
10	е		м	е	н	я		п
11	Р	е	З	Р	е	н	ь	е
12	м		н	а	к	а	З	а
13	т	ь		с	н	а	ч	а
14	л	а		я		м	0	л
15	ч	а	т	ь		х	0	т
16	е	л	а		п	0	в	е
17	Р	ь	т	е		м	0	е
18	г	0		с	т	ы	д	а
19	в	ы		б		н	е	
20	у	З	н	а	л	и		н
21	е	к	0	г	д	а		к
22	0	г	Д	а		б		н
23	а	д	е	ж	д	У		я
24	и	м	е	л	а		х	0
25	т	ь		Р	е	д	к	0
26	х	0	т	ь		в		н
27	е	д	е	л	ю		Р	а
28	З		в		д	е	Р	е
29	в	н	е		н	а	ш	е
30	и		в	и	д	е	т	ь
31	в	а	с					

Зауваження стосовно ступеня неоднозначності відновлення відкритого тексту. В обох прикладах нам вдалося, хоча й за певних зусиль, відновити відкриті тексти. Робота з дешифрування значно полегшується зі збільшенням довжини шифртексту (чи, як говорять, обсягу наявного в розпорядженні матеріалу перехоплення). Якби обсяг матеріалу було збільшено кількаразово, то в другому прикладі в таблиці 8×8 з великою ймовірністю було б закреслено всі клітинки, окрім відповідних справжньому порядку слідування один за одним стовпців у відкритому тексті. У першому прикладі варіаційний ряд частості вживаності літер відкритого тексту майже цілковито збігався б з варіаційним рядом імовірностей знаків відкритого тексту і нам залишалося підписати один варіаційний ряд під іншим і здійснити зворотну заміну. Окремі помилки легко усувалися б при першому ж прочитуванні дешифрованого варіанта тексту. З іншого боку, при зменшенні обсягу матеріалу завдання дешифрування істотно ускладнюється. Насправді, уявіть собі, що вам запропоновано дешифрувати зашифрований за допомогою простої заміни шифртекст, складений з одного першого слова нашого першого прикладу – ДОЧАЛЬ. Для кожного слова з шести різних літер віднайдеться проста заміна, застосовуючи яку ми здобудемо даний шифртекст. Проблема полягає не в тім,

аби віднайти ключове підставляння, а в тім, як обрати з численної кількості варіантів відновлених відкритих текстів саме той, котрий було зашифровано.

Отже, важливою характеристикою ефективності криптографічного захисту є ступінь неоднозначності відновлення відкритого тексту за шифрованим. Подамо якісне означення цього поняття. Під ступенем неоднозначності природно розуміти математичне сподівання кількості варіантів відкритих текстів, які може бути здобуто за застосовування алгоритму дешифрування. Якщо припустити, що за випадкового вибору хибного ключа (ключа, розбіжного зі справжнім ключем, із застосуванням якого було здобуто шифртекст, відновлений при розшифровуванні) і розшифровування на цьому випадковому хибному ключі розшифрований текст не відрізняється від випадкового тексту, то ймовірність того, що здобутий текст довжини N буде осмисленим (за класичним означенням імовірностей), дорівнює відношенню кількості сприятливих завершень – кількості $A(N)$ осмислених відкритих текстів довжини N до загальної кількості текстів довжини N, котра дорівнює, вочевидь, n^N, де n – кількість літер в абетці відкритих текстів. Для того аби дістати ступінь неоднозначності, слід помножити здобуте відношення на кількість K варіантів ключів, звідки дістаємо формулу

$$r = r(N) = KA(N)n^{-N} \qquad (6.1)$$

для обчислення ступеня неоднозначності L відновлення відкритого тексту. Отже, шифрування забезпечує надійний захист повідомлення, якщо неоднозначність відновлення відкритого тексту за шифрованим є надто велика.

Ступінь неоднозначності дозволяє класифікувати шифри за ступенем їхньої криптографічної стійкості. Якісні міркування тут є такі. Якщо неоднозначність дешифрування (можлива кількість здобуваних відкритих текстів) є $r(N)$, а мета дешифрування визначена здобуттям відкритого справжнього тексту, то ймовірність здобуття саме його оцінюється величиною $l/r(N)$. Шифри, для яких $r(N) = A(N)$, називаються ***абсолютно (теоретично) стійкими*** (для абсолютно стійкого шифру застосування будь-якого алгоритму дешифрування не надає жодної інформації про відкритий текст). До формули (5.1), її висновку та її трактувань слід ставитися надто обережно. Насправді для здобуття абсолютної стійкості шифру досить мати n^N ключів.

Шифри, що вони теоретично не є стійкими, для яких надійність криптографічного захисту забезпечується надто великою складністю зреалізовування відомих алгоритмів дешифрування, називають ***практично стійкими***.

Шифри, котрі не належать до класів теоретично стійких чи практично стійких шифрів, називають ***шифрами тимчасової стійкості***.

Кількість осмислених відкритих текстів довжини N для великих значень N зазвичай вважається за дорівнювану 2^{HN}, де H – ентропія (в бітах) на знак відкритого тексту. Для літературного тексту, як засвідчують експериментальні розрахунки, величина H є наближена до одиниці. Для більш формалізованих

текстів (організаційно-розпорядчих листів, документів тощо) величина H перебуває в межах від 0,5 до 0,8.

Повертаючи до аналізу шифру простої заміни, зауважимо, що ступінь неоднозначності відновлення відкритого тексту для цього шифру має вигляд

$$r(N) = \frac{n! 2^{HN}}{n^N},$$

де H – ентропія на літеру відкритого тексту.

Шифр простої заміни є теоретично стійким при шифруванні літературного тексту (n = 32) довжиною в одне-два слова і забезпечує лише тимчасову стійкість за великого обсягу матеріалу.

Експерименти засвідчують, що практичне дешифрування шифру простої заміни для літературного тексту є можливе в разі, якщо довжина шифртексту вдвічі-втричі перевищує розмір абетки відкритого тексту, тобто дорівнює 60...100 літерам.

СПИСОК РЕКОМЕНДОВАНОЇ ЛІТЕРАТУРИ

1. Горбенко І. Д., Горбенко Ю. І. Прикладна криптологія. Теорія. Практика: монографія. Харків: Видавництво «Форт», 2012. 880 с.

2. Корченко О. Г., Сіденко В. П., Дрейс Ю. О. Прикладна криптологія: системи шифрування: підручник. Київ: ДУТ, 2014. 448 с.

3. Schneier B. Applied Cryptography: Protocols, Algorithms and Source Code in C: 20th Anniversary Edition. Wiley, 2015. 784 p.

4. Stavroulakis P., Stamp M. Handbook of Information and Communication Security. Berlin: Springer-Verlag, 2010. 863 p.

5. Daemen J., Rijmen V. Design of Rijndael. AES – The Advanced Encryption Standard. Berlin: Springer-Verlag, 2002. 238 p.

6. Захарченко М. В., Онацький О. В., Йона Л. Г., Шинкарчук Т. М. Асиметричні методи шифрування в телекомунікаціях: навч. посіб. Одеса: ОНАЗ ім. Попова, 2011. 184 с.

7. Йона Л. Г., Онацький О. В., Швець О. В. Системи банківської безпеки: навч. посіб. Одеса: ДУІТЗ, 2022. 192 с.

8. Shannon Claude. Communication Theory of Secrecy Systems / Bell System Technical Journal, 1949. 59 p.

9. Бабенко Т. В., Гулак Г. М., Сушко С. О., Фомичова Л. Я. Криптологія у прикладах, тестах і задачах: навч. посіб. Дніпро: Національний гірничий університет, 2013. 318 с.

10. Остапо С. Е., Євсеєв С. П., Король О. Г. Технології захисту інформації: навч. посіб. Харків: ХНЕУ, 2013. 476 с.

11. Кузнецов Г. В., Фомічов В. В., Сушко С. О. Математичні основи криптографії. Дніпропетровськ: НГУ, 2004. 389 с.

12. Горбенко Ю. І., Горбенко І. Д. Інфраструктури відкритих ключів. Електронний цифровий підпис. Теорія та практика: монографія. Харків: Видавництво «Форт», 2010. 608 с.

13. Peter Stavroulakis, Mark Stamp. Handbook of Information and Communication Security. Berlin: Springer-Verlag, 2010. 863 p.

14. Menezes A., Oorschot P., Vanstone S. Handbook of Applied Cryptography. CRC Press, 1996. 816 p.

15. DES Animation. URI: https://www.youtube.com/watch?v=Vcld7CMAnNs

16. Фільштінський В. А., Бережний А. В. Математичні основи криптографії: конспект лекцій. Суми: Сумський державний університет, 2011. 138 с.

17. Козіна Г. Л. Криптографія від історії до сучасних стандартів: навч.посіб. Запоріжжя: НУ «Запорізька політехніка», 2020. 192 с.

ДОДАТОК А

Таблиця шифрування Тритемія

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

Таблиця шифрування Бофора для англійської абетки

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

ТЕСТИ ДЛЯ ПЕРЕВІРКИ ЗНАНЬ

1 З яких частин складається автоматизована система?

а) мережа зв’язку, апаратно-програмні засоби, персонал, інформаційні ресурси;

б) інформаційні технології, зовнішнє середовище, обчислювальна система, персонал та інформація;

в) інформаційні ресурси, персонал, операційна система, зовнішнє середовище, розподілена обчислювальна система.

2 Які властивості інформації підлягають захисту в автоматизованих системах та системах телекомунікації?

а) конфіденційність, цілісність, доступність, захищеність;

б) конфіденційність, цілісність, доступність;

в) цілісність, доступність, секретність.

3 Що означає поняття „документ” відповідно до нормативної бази України? Це:

а) інформація, зафіксована на будь-якому матеріальному носії;

б) інформація, зафіксована на будь-якому матеріальному носії у визначеному законом порядку;

в) інформація, зафіксована на будь-якому матеріальному носії та зареєстрована у державному органі.

4 У яких трьох аспектах має розвиватися проблема захисту інформації в телекомунікаційних системах?

а) вдосконалення відповідної нормативної бази, організаційних заходів та програмно-апаратних засобів;

б) вдосконалення відповідної нормативної бази та розподільних обчислювальних мереж, розробляння сучасних захищуваних інформаційних технологій.

5 Що називають загрозами інформаційних об’єктам?

а) потенційно можливі події, які призводять до порушень політики безпеки;

б) потенційно можливі події, які призводять до втрат конфіденційності, цілісності та доступність інформації;

в) потенційно можливі події, які призводять до втрат електронних документів.

6 Що називають атаками на інформаційні об’єкти?

а) спроба реалізації загрози;

б) підготовка, реалізація несанкціонованого доступу до захищуваного об’єкта та усунення слідів нападу;

в) факт несанкціонованого доступу до захищуваного об’єкта.

7 Атака – це спроба реалізації загрози?

а) так;

б) ні.

8 Загрози, які мають суб’єктивну природу, можуть бути лише випадковими?

а) так;

б) ні.

9 Конфіденційність, цілісність та доступність – властивості інформації?

а) так;

б) ні.

10 Цілісність – це відсутність зумисного спотворення інформації?

а) так;

б) ні.

11 Автоматизована система – це:

а) система, що здійснює автоматизоване опрацьовування даних, до складу якої входять технічні засоби їхнього опрацьовування (засоби обчислювальної техніки та зв'язку), а також методи і процедури, програмне забезпечення;

б) система, за допомогою якої забезпечується зв'язок поміж користувачами.

12 Які алгоритми називають симетричними?

а) блочні, потокові, з відкритим ключем;

б) алгоритми, в яких операції шифрування виконуються одним ключем.

13 Які криптографічні перетворювання використовуються для створювання симетричних криптографічних алгоритмів?

а) гамування, переставляння і заміни;

б) функційні перетворювання, переставляння та заміни;

в) параметричні перетворювання, переставляння та заміни, гамування.

14 Які криптографічні перетворювання використовуються у стандарті DES?

а) зворотні табличні переставляння, табличні заміни, переставляння з розширенням блоків, переставляння зі стисненням блоків, циклічні переставляння;

б) табличні переставляння, табличні заміни, функційні перетворювання.

15 Які криптографічні перетворювання використовуються у стандарті ГОСТ 28147–89?

а) зворотні табличні переставляння, табличні заміни, переставляння з розширенням блоків, переставляння зі стискуванням блоків, циклічні переставляння;

б) табличні переставляння, табличні заміни та функційні перетворювання.

16 До потокових алгоритмів шифрування належить:

а) алгоритм, який використовує різні ключі зашифровування і розшифровування;

б) алгоритм, в якому кожен символ відкритого тексту зашифровується незалежно від інших і розшифровується так само.

17 ГОСТ 28147–89 працює в робочих режимах:

а) гамування зі зворотним зв'язком по виходу;

б) зчеплення блоків шифру;

в) проста заміна.

18 Циклічний регістр зсуву в алгоритмі за ГОСТом 28147–89 виконується:

а) на 11 розрядів праворуч;

б) на 11 розрядів ліворуч;

в) на 11 розрядів праворуч, потім ліворуч, залежно від номера циклу.
19 Кількість циклів шифрування в алгоритмі DES:
а) 16;
б) 32;
в) 64;
г) 128;
д) 256.
20 Розмір блока даних в алгоритмі за ГОСТом 28147–89:
а) 16;
б) 32;
в) 64;
г) 128;
д) 256.
21 Розмір ключа в алгоритмі за ГОСТом 28147–89:
а) 16;
б) 32;
в) 64;
г) 128;
д) 256.
22 Розмір ключа в алгоритмі DES:
а) 16;
б) 32;
в) 48;
г) 59;
д) 64.
23 Кількість циклів шифрування в алгоритмі за ГОСТом 28147–89:
а) 16;
б) 32;
в) 64;
г) 128;
д) 256.
24 Які параметри обираються в алгоритмі RSA:
а) два великих простих числа, ключ зашифровування;
б) функція Ейлера, ключ розшифровування;
в) ключ зашифровування і розшифровування.
25 Для чого призначено алгоритм Діффі–Хеллмана:
а) для шифрування повідомлень;
б) для генерування секретного ключа.
26 Алгоритм Ель–Гамаля можна використовувати:
а) для шифрування повідомлень і генерування секретного ключа;
б) для генерування секретного ключа;
в) для шифрування повідомлень та створювання цифрового підпису.
27 Геш-функція застосовується у криптографії:
а) для шифрування повідомлень;
б) для генерування секретного ключа;

в) у протоколах автентифікації цифрового підпису.

28 Зашифрувати повідомлення M за допомогою алгоритму RSA, якщо відомо: $n = 15$, ключ зашифровування $e = 3$, $M = 3$.

а) 16;

б) 9;

в) 12;

г) 25.

29 Розшифрувати криптограму C за допомогою алгоритму RSA, якщо відомо: $n = 21$, ключ розшифровування $d = 3$, $C = 5$.

а) 20;

б) 17;

в) 15;

г) 7.

30 Знайти найбільший спільний дільник для чисел 85, 34:

а) 1;

б) 5;

в) 7;

г) 17.

31 Обчислити $4^8 \pmod{14}$.

а) 1;

б) 2;

в) 4^4;

г) 14.

32 Знайти x, якщо $3^x \equiv 5 \pmod{7}$.

а) 3;

б) 5;

в) 7;

г) 6.

33 Розв’язати порівняння $6x \equiv 2 \pmod{11}$.

а) 3;

б) 4;

в) 6;

г) 11.

34 Обчислити функцію Ейлера $\varphi(49)$.

а) 26;

б) 36;

в) 48;

г) 42.

35 Зашифрувати повідомлення M за допомогою алгоритму Рабіна, якщо відомо: $p = 3$, $q = 7$, $M = 9$.

а) 27;

б) 60;

в) 18;

г) 21.

36 Зашифрувати повідомлення M за допомогою алгоритму Ель–Гамаля, якщо відомо: $p = 5$, відкритий ключ сторони $A - Y_a = 3$, секретний ключ сторони $B - k_b = 2$, $M = 6$.

а) 1;

б) 2;

в) 3;

г) 4.

37 Визначити ключ методом Діффі–Хеллмана, якщо відомо: $p = 5$, відкритий ключ сторони $A - Y_a = 6$, секретний ключ сторони $B - k_b = 3$.

а) 1;

б) 2;

в) 3.