МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Сумський державний університет
Факультет електроніки та інформаційних технологій
Кафедра електроніки і комп'ютерної техніки

«До захисту допущено»
Завідувач кафедри ЕКТ

_____ __Анатолій ОПАНАСЮК_
(підпис) (Ім’я та ПРІЗВИЩЕ)
____ _____ 2024 р.

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня «магістр»

зі спеціальності 171 «Електроніка»

освітньо-професійної програми «Електронні системи»

на тему:

КРИПТОГРАФІЧНА СИСТЕМА ЗАХИСТУ ІНФОРМАЦІЇ ІЗ ЗАСТОСУВАННЯМ ОДНОРАЗОВОГО БЛОКНОТУ

Здобувача групи ЕС.м-31 _____ Бирина Олександра Олександровича

Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

_____ __Олександра БИРИНА_
(підпис) (Ім’я та ПРІЗВИЩЕ)

Керівник, доцент, к.т.н., доцент Ольга БЕРЕЖНА _____
(підпис)

Консультант з техніко-економічної частини,
доцент, к.е.н., доцент Олександр МАЦЕНКО _____
(підпис)

Суми – 2024

СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ

Факультет електроніки та інформаційних технологій

Кафедра електроніки і комп'ютерної техніки

Напрям підготовки 171 «Електроніка»

Освітня програма Електронні системи

ЗАТВЕРДЖУЮ

Зав. кафедрою Опанасюк А. С.

«__»_____ 2024 р.

З А В Д А Н Н Я

на кваліфікаційну роботу магістра

Бирину Олександру Олександровичу

1. Тема роботи «Криптографічна система захисту інформації із застосуванням одноразового блокноту» .

затверджена наказом по університету «01» жовтня 2024 р. № 1003-VI.

2. Термін здачі студентом завершеної роботи 05.12.2024.

3. Вихідні дані до роботи Розробити криптографічну систему захисту інформації із застосуванням одноразового блокноту. Принципову схему реалізувати із застосуванням мікроконтролеру.

4. Зміст розрахунково-пояснювальної записки (перелік питань, що належить розробити) 1) Огляд літератури та поставлення задачі роботи. 2) Науково-дослідна частина. 3) Розробка електронної системи з використанням отриманих результатів дослідження. 4) Техніко-економічна частина.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) 1) Схема електрична структурна. 2) Схема алгоритму. 3) Схема електрична функціональна. 4) Схема електрична принципова.

6. Консультанти з кваліфікаційної роботи

Розділи	Консультанти	Завдання видав	Завдання прийняв
Техніко-економічна частина	Маценко О. М.		

7. Дата видачі завдання _____

8. Керівник роботи _____

9. Завдання прийняв до виконання_____

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів дипломного проекту	Термін виконання етапів роботи	Примітки
1	Огляд літератури та постановка завдання проектування	04.11.24 – 09.11.24	
2	Науково-дослідна частина	10.11.24 – 15.11.24	
3	Розробка алгоритму функціонування та структурної схеми електронної системи	16.11.24 – 20.11.24	
4	Розробка функціональної схеми електронної системи	21.11.24 – 24.12.24	
5	Розробка схеми електричної принципової електронної системи	25.12.24 – 02.12.24	
6	Техніко-економічна частина	03.12.24 – 05.12.24	
8	Оформлення пояснювальної записки	06.12.24 – 08.12.24	
9	Оформлення графічного матеріалу	09.12.24 – 13.12.24	
10	Представлення роботи керівнику і отримання відгуку	14.12.24	
11	Представлення роботи кафедрі для отримання рецензії	15.12.24	

Студент _____

Керівник роботи _____

«___» _____ 2024 р.

РЕФЕРАТ

Записка: 75 сторінок, 19 рисунків, 7 таблиць, 9 джерел.

Тема роботи: «Криптографічна система захисту інформації із застосуванням одноразового блокноту».

Об'єктом розробки є криптографічна система захисту інформації із застосуванням одноразового блокноту.

Мета роботи – розробка апаратної частини криптографічної системи захисту інформації.

Пояснювальна записка складається з п’яти розділів, вступу, висновків і додатку.

У першому розділі наданий огляд методів та засобів захисту даних.

У другому розділі проводиться аналіз сучасних методів захисту.

У третьому розділі проводиться вибір структурної схеми та обґрунтування алгоритму її функціонування.

У четвертому розділі проводиться розробка функціональної та принципової електричних схем криптографічної системи захисту інформації і розглянуті характеристики всіх її блоків.

У п’ятому розділі представлено схему розрахунку собівартості проектованої електронної системи при серійному її виробництві.

У висновках наводяться результати розробки електронної системи.

Ключові слова: засоби захисту даних, шифр гамування, поточне шифрування, мікропроцесор.

ЗМІСТ

С.

ВСТУП

У сучасному світі інформація стала одним із найцінніших ресурсів, який потребує надійного захисту. Швидкий розвиток цифрових технологій призвів до зростання обсягів переданої та збереженої інформації, а також до збільшення кількості кіберзагроз, спрямованих на її компрометацію. У зв'язку з цим криптографія відіграє ключову роль у забезпеченні конфіденційності, цілісності та доступності даних.

Одним із найефективніших і математично доведених методів криптографічного захисту є система одноразового блокноту (One-Time Pad). Ця технологія базується на використанні унікального ключа для кожного сеансу шифрування, що робить її теоретично невразливою до зламу за умови дотримання визначених правил.

Метою даного дипломного проєкту є розробка криптографічної системи захисту інформації із застосуванням одноразового блокноту, яка дозволить забезпечити найвищий рівень безпеки для критично важливих даних. У межах дослідження буде розглянуто теоретичні аспекти методу, проведено аналіз його переваг і обмежень, а також запропоновано практичну реалізацію із використанням сучасних технологій.

Захист інформації сьогодні є не лише технічним викликом, але й одним із важливих факторів забезпечення інформаційної безпеки суспільства, що підтверджує актуальність теми проєкту.

1 ОГЛЯД ЛІТЕРАТУРИ І ПОСТАНОВКА ЗАВДАННЯ ПРОЕКТУВАННЯ

1.1 Основні методи та засоби захисту даних

З огляду на стрімкий розвиток комп'ютерних технологій та інформаційного простору, все актуальнішими стають питання, пов'язані з інформаційною безпекою.

Під інформаційною безпекою розуміють дії, спрямовані на запобігання несанкціонованому та ненавмисному видаленню, спотворенню, пошкодженню, перегляду та редагуванню, використанню та будь-яким іншим можливим операціям, що виконуються людьми (співробітниками, компаніями, зловмисниками, вірусами тощо), які не мають на це прав.

Засоби захисту даних поділяються на кілька груп: фізичні, апаратні, програмні (криптографічні), організаційні, етичні та законодавчо-правові. У роботі розглядатимуться засоби криптографічного захисту інформації (СКЗІ).

Криптографія (з грецької перекладається як «таємно пишу») є наукою, що займається забезпеченням цілісності, автентифікації та конфіденційності даних.

Протекція конфіденційності та забезпечення цілісності даних взаємопов'язані, і часто методи та засоби вирішення одного завдання допомагають у вирішенні іншого.

Криптографія, окрім шифрування та дешифрування, також займається управлінням ключами, електронними цифровими підписами, захистом інформації на рівні квантової фізики, а також отриманням (прихованої) інформації.

До методів криптографічного захисту даних застосовуються різні системи класифікації. Зокрема, за формою впливу на початкові дані вони поділяються на 4 підгрупи

Стеганографія (приховане письмо) являє собою розміщення прихованої інформації всередині відкритої [2]. Метою такого методу є:

1. Підтвердження авторського права шляхом впровадження, наприклад, імені автора у вигляді стеганографічного водяного знаку;

2. Створення цифрових відбитків, наприклад, для захисту виключного права;

3. Захист автентичності документів і подальше виявлення неоригінальних і підроблених даних;

4. Безпосередня передача прихованих даних різними способами так, щоб зловмисник не здогадувався про сам факт передачі інформації.

Кодування — це процес заміни деяких речень, смислових конструкцій або слів кодами (буквено-цифровими позначеннями, словами тощо) [3].

Для розшифрування інформації застосовуються словники або таблиці, які є у обох сторін (що передає і що приймає). Цей процес застосовується в обмеженій тематиці тексту через необхідність використання словників.

До недоліків кодування можна віднести також необхідність періодичної зміни словників і їх подальшого розповсюдження, щоб уникнути розкриття коду.

Стиснення даних умовно відносять до методів криптографічного перетворення інформації та застосовують для зменшення обсягу різними способами, наприклад, використовуючи правила скорочення, заміну тощо [4]. Такий метод є марним без зворотного перетворення отриманих даних у початкові. До того ж стиснення не можна назвати безпечним і надійним методом через легкість розшифрування із застосуванням статистичних методів дослідження.

Методом шифрування називають певний набір дій і перетворень даних з подальшим отриманням закритої інформації та можливістю її зворотного дешифрування. Шифрування виникло понад три тисячі років тому, і деякі способи, придумані в ті часи, використовуються й донині.

З появою комп’ютерних систем у суспільстві почали створювати нові способи шифрування, дешифрування, а також атаки на захищену інформацію з урахуванням можливостей сучасних технологій.

Атакою називають процес розшифрування інформації особами, які не мають доступу до ключів або алгоритму шифрування.

Шифрування може бути симетричним і асиметричним, залежно від кількості ключів. Якщо ключ один — шифрування симетричне, якщо ключів два — шифрування асиметричне.

До методів шифрування висуваються певні вимоги (диктовані реальними умовами в сфері інформаційної безпеки):

- Висока стійкість шифру;
- Неможливість застосування аналітичного способу розшифрування;
- Висока складність підбору ключа;
- Захищена інформація не повинна значно перевищувати за обсягом вихідну;
- У процесі шифрування не повинно виникати втрати або спотворення інформації;
- Помилки, що виникли під час шифрування, не повинні впливати на цілісність інформації;
- Відносно невеликий час дешифрування;
- Прийнятна вартість процесу дешифрування і шифрування.

Узагальнена схема криптосистеми наведена на рисунку 1.1.

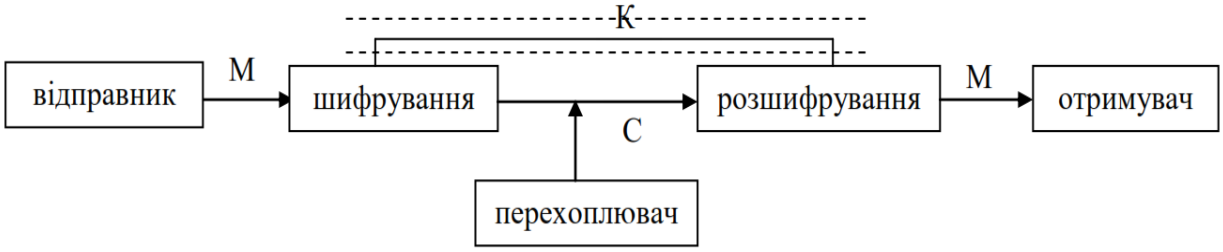

Рисунок 1.1 – Узагальнена схема криптосистеми

Симетричне шифрування реалізується наступним чином.

На схемі:

M – інформація, яку потрібно передати відправнику отримувачу.

Перехоплювач - особа, яка бажає отримати M.

Для передачі M через незахищений канал даних здійснюється шифрування M із використанням зворотного перетворення E_k, у результаті чого отримуються дані, захищені від прочитання C ($C=E_k(M)$).

Під час отримання зашифрованого тексту C отримувач здійснює його перетворення до початкового вигляду за допомогою дешифратора $D_k(C)$ і ключа K.

Криптографічна система має безліч різних варіантів реалізації: апаратне забезпечення, набір інструкцій, програмне забезпечення — усе це дозволяє зашифрувати інформацію і дешифрувати її різними способами.

1.1.1 Вимоги до криптосистем

Процес криптографічного захисту даних може здійснюватися як у програмному, так і в апаратному форматі. Апаратна реалізація є дорожчою, проте має переваги, зокрема високу продуктивність, простоту використання та надійний захист. Програмна реалізація більш універсальна і дозволяє застосовувати різні підходи до роботи з алгоритмами шифрування.

До сучасних криптографічних систем висуваються такі загальноприйняті вимоги:

- Доступ до зашифрованого повідомлення можливий тільки за наявності відповідного ключа.
- Кількість операцій для знаходження ключа на основі фрагмента шифрованого повідомлення та його відкритого тексту має дорівнювати кількості можливих ключів.
- Обчислювальна складність підбору ключа повинна мати нижню межу, що унеможливлює розшифрування навіть із використанням сучасних комп’ютерів та мережевих обчислень.
- Відомий алгоритм шифрування не має знижувати надійність захисту.
- Зміна ключа повинна призводити до суттєвої зміни вигляду зашифрованого повідомлення навіть за умови використання одного і того ж алгоритму.
- Елементи алгоритму шифрування повинні залишатися незмінними.
- Додаткова інформація, яка вводиться під час шифрування, має бути прихованою в зашифрованому тексті.

- Довжина зашифрованого тексту повинна відповідати довжині вихідного.
- Не повинно існувати простих залежностей між послідовними ключами.
- Кожен ключ із можливого набору повинен забезпечувати високий рівень захисту.
- Алгоритм має бути реалізованим як у програмному, так і в апаратному форматі, причому зміна довжини ключа не повинна впливати на якість шифрування.

1.1.2 Реалізація криптографічних методів

Реалізація криптографічних систем включає два ключових аспекти:

1. Розробка засобів, що втілюють алгоритми шифрування.
2. Розробка методик для ефективного використання цих засобів.

Методи криптографічного захисту можуть бути реалізовані програмно або апаратно.

Програмна реалізація можлива завдяки тому, що всі криптографічні методи є формальними та описуються у вигляді кінцевого алгоритму.

Апаратна реалізація передбачає використання спеціалізованих електронних пристроїв для виконання операцій шифрування та дешифрування.

Комбіновані методи, що поєднують програмний і апаратний підходи, стають дедалі популярнішими. Наприклад, багато сучасних пристроїв використовують криптографічні співпроцесори — спеціальні модулі, призначені для виконання криптографічних операцій (додавання за модулем, зсуви тощо). Використання такого підходу дозволяє змінювати метод шифрування за допомогою зміни програмного забезпечення, поєднуючи переваги обох підходів.

Американський стандарт DES залишається поширеним для реалізації засобів шифрування, тоді як у Росії популярні власні розробки, наприклад, пристрій «КРИПТОН».

Основна перевага програмних засобів — їхня гнучкість, яка дозволяє легко змінювати алгоритми.

Недоліком є нижча швидкодія у порівнянні з апаратними засобами, яка може бути в 10 разів меншою.

Комбіновані засоби поєднують продуктивність апаратного підходу з універсальністю програмного.

Вибір способу реалізації криптографічного захисту залежить від особливостей системи та базується на аналізі вимог до інформаційної безпеки [1].

1.2 Роль сучасних криптографічних алгоритмів

У епоху персональних комп'ютерів сучасні алгоритми, такі як симетричні блокові шифри та асиметричні алгоритми з відкритим ключем, замінили одноразові шифроблокноти через практичні міркування та вирішення проблем розподілу ключів. Сучасні криптографічні алгоритми забезпечують практичну (але не доведену) безпеку та конфіденційність, що є важливим для нашої економіки та повсякденного життя. Однак вищі командні структури збройних сил, а також деякі спеціальні військові та урядові установи потребують абсолютної безпеки та конфіденційності, яку може забезпечити лише одноразове шифрування.

Деякі експерти стверджують, що розподіл великої кількості одноразових шифроблокнотів або ключів є непрактичним. Це дійсно було обмеженням у добу паперових стрічок на котушках та паперових блокнотів. Проте сучасна електроніка, наприклад генератор SGCL-100M, що наведений на рисунку 1.2, може виступати як практично "нескінченне" джерело одноразових ключів. Здатність генерувати одноразові ключі більше не є обмеженням.

Сучасні реалізації машин для шифрування з використанням одноразових блокнотів з вбудованим програмним забезпеченням для поточного одноразового шифрування можуть обробляти великі обсяги даних на високій швидкості. Сучасні технології зберігання даних, такі як USB-накопичувачі, DVD-диски, зовнішні жорсткі диски, твердотілі накопичувачі або спеціалізовані модулі ОТК, дозволяють фізично транспортувати величезні обсяги дійсно випадкових ключів.

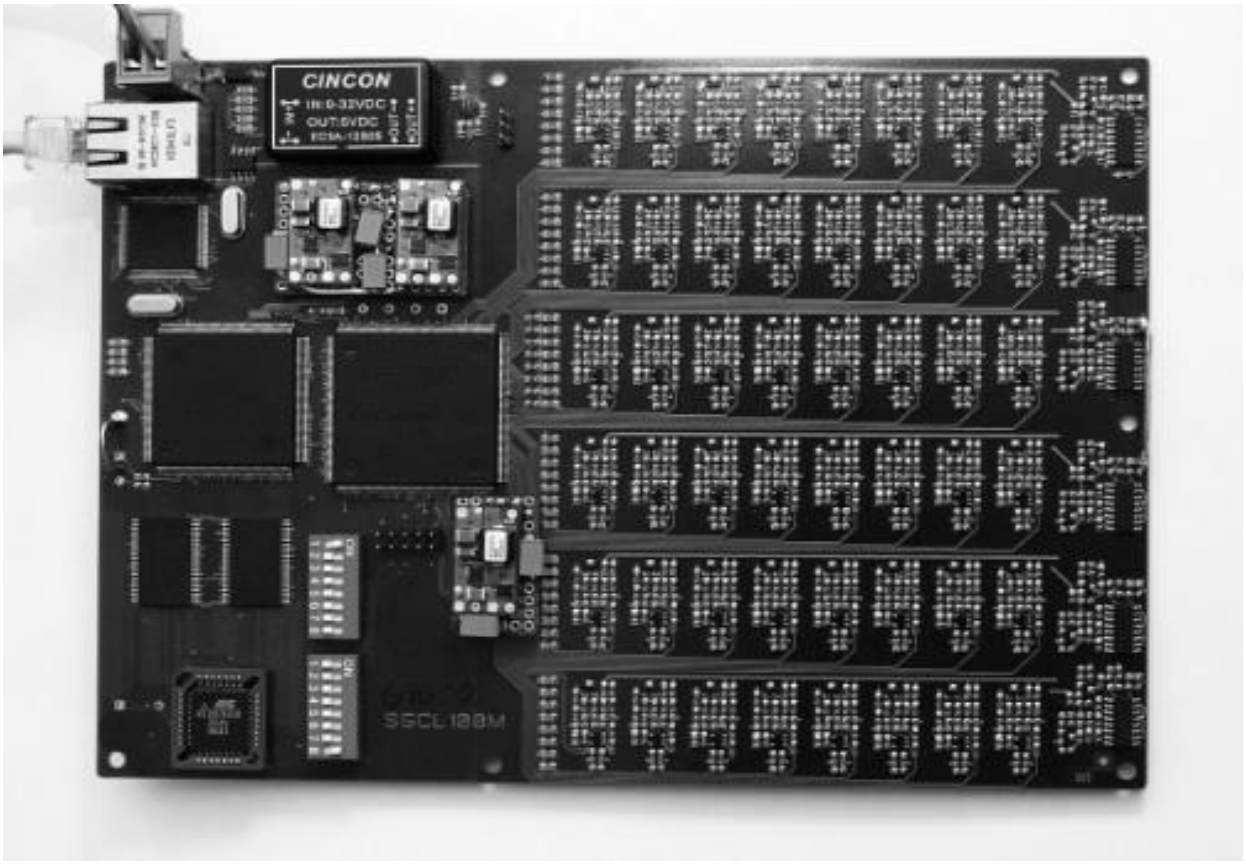

Рисунок 1.2 – Модель генератора SGCL-100M

Фактичні чутливі комунікації часто обмежуються невеликою кількістю важливих користувачів. У таких випадках однорангові комунікації з відповідним розподілом ключів, можливо, у конфігурації зі зіркоподібною топологією, вже не є практичною проблемою, особливо з огляду на переваги безпеки. Використовуючи так званий "sneakernet" (перенесення даних на знімних носіях за допомогою фізичної доставки), можна досягти пропускної здатності для одноразових ключів, що перевищує можливості мережі з обробки зашифрованих даних. Іншими словами, для транспортування терабайта ключового матеріалу, збереженого на зовнішньому диску, автомобілем може знадобитися кілька годин, тоді як споживання цієї кількості ключів у широкосмуговій мережі займе дні або навіть тижні. Терабайт ключів може легко забезпечити шифрування електронної пошти спеціальних (військових або дипломатичних) користувачів протягом року, включаючи вкладення [2].

1.3 Визначення одноразового блокноту

Одноразовий блокнот — це шифр Шеннона ε = (E,D), де ключі (k), повідомлення (m) і шифротексти (c) є бітовими рядками однакової довжини.

Іншими словами, одноразовий блокнот, що відповідає шифру Шеннона ε, визначається над множинами (K, M, C), де

$$\mathcal{K} := \mathcal{M} := \mathcal{C} := \{0,1\}^L$$

для деякого фіксованого параметра L.

Для ключа $k \in \{0,1\}^L$ і повідомлення $m \in \{0,1\}^L$ функція шифрування E(k,m) визначається як $k \oplus m$, де $\oplus$ позначає побітове додавання за модулем 2.

Одноразові блокноти працюють шляхом поєднання повідомлення mmm у відкритому тексті з випадковим секретним ключем (який називають одноразовим блокнотом). Потім кожен біт або символ повідомлення шифрується шляхом комбінування його з відповідним бітом або символом із блокнота, використовуючи модульну арифметику.

Якщо одноразовий блокнот відповідає наступним властивостям:

1. Він є дійсно випадковим.
2. Він має довжину, щонайменше рівну довжині відкритого тексту.
3. Він ніколи не використовується повторно повністю або частково.
4. Він зберігається в повній таємниці,

тоді можна довести, що шифротекст неможливо розшифрувати чи зламати, тобто він є "абсолютно безпечним".

1.4 Ідеальний захист

У криптографії "ідеальна безпека" є золотим стандартом безпеки та особливим випадком інформаційно-теоретичної безпеки. Вона полягає в тому, що для алгоритму шифрування, якщо існує шифротекст, створений за його допомогою, жодна інформація про повідомлення не надається без знання ключа.

1.4.1 Визначення ідеальної безпеки

Нехай ε = (E,D) — шифр Шеннона, визначений над множинами (K,M,C).

Розглянемо ймовірнісний експеримент, у якому випадкова змінна k рівномірно розподілена по K.

Якщо для всіх $m_0, m_1 \in M$ і всіх $c \in C$ виконується:

$$Pr[E(k,m_0)=c]=Pr[E(\mathbf{k},m_1)=c],$$

то ми кажемо, що ε є ідеально безпечним шифром Шеннона.

Іншими словами, якщо ймовірність того, що шифротекст c відповідає m0, дорівнює ймовірності того, що той самий шифротекст c відповідає m1, тоді шифр ε є ідеально безпечним.

Це означає, що ідеально безпечний шифр ε створює шифротекст, який має однакову ймовірність бути будь-яким повідомленням, тобто шифротекст c не дає жодної інформації про відкритий текст m.

Щоб довести це, спочатку надаємо кілька еквівалентностей.

Нехай ε=(E,D) — шифр Шеннона, визначений над (K,M,C).

Тоді наступні твердження еквівалентні:

– ε є ідеально безпечним;
– для кожного шифротексту c∈C існує Nc (який може залежати від c) так, що для всіх повідомлень $m \in M$ виконується:
–

$$|k \in K : E(k,m)=c| = N_c \text{iii}).$$

Якщо випадкова змінна k рівномірно розподілена по K, тоді кожна з випадкових змінних E(k,m) для m∈M має той самий розподіл.

Іншими словами, твердження, що ε є:

1. ідеально безпечним, еквівалентне твердженням, що:
2. для кожного шифротексту ccc існує певне число Nc , таке, що для всіх повідомлень mmm функція шифрування E може згенерувати ccc;

3. якщо випадкова змінна k рівномірно розподілена по K, то кожна змінна k, яка може бути використана для шифрування m, має той самий розподіл.

Використовуючи (2), наступне доводить, що одноразовий блокнот задовольняє вимогам ідеально безпечного шифру Шеннона:

Доведення, що одноразовий блокнот є ідеально безпечним шифром Шеннона.

Припустимо, що шифр Шеннона ε=(E,D) є одноразовим блокнотом і визначений над (K,M,C), де K:=M:=C:={0,1}^{L}. Для будь-якого фіксованого повідомлення m∈{0,1}^{L} і шифротексту c∈{0,1}^{L} існує унікальний ключ k∈{0,1}^{L}, який задовольняє рівняння k⊕m=c, а саме: k:=m⊕c. Отже, ε задовольняє умову ii) теореми (з Nc=1для кожного шифротексту c) [2].

1.5 Шифр гамування

Гамування — це метод шифрування, що полягає в поєднанні символів вихідного тексту з символами випадково згенерованої послідовності, яку називають гамою шифру. Надійність шифру безпосередньо залежить від довжини унікальної частини цієї гами. Сучасні обчислювальні системи дозволяють генерувати практично нескінченні послідовності, що робить метод гамування одним із ключових підходів до захисту інформації в інфокомунікаційних системах.

$$T_ш^{(i)} = Г_ш^{(i)} \oplus T_0^{(i)},\ i=1,..к$$ – процес шифрування

$$T_0^{(i)} = Г_ш^{(i)} \oplus T_ш^{(i)},\ i=1,..к$$ – процес дешифрування

Де $T_ш^{(i)}$- i-й блок шифру; $T_0^{(i)}$- i-й блок відритого тексту; $Г_ш^{(i)}$ - i-й блок гами; $к$- кількість блоків.

Перед процесом шифрування текст поділяється на блоки однакової довжини $T_0(i)$, як правило, по 64 біти. Гама шифру формується як послідовність блоків тієї ж довжини $Г_ш(i)$.

Гамування передбачає застосування гами шифру до відкритих даних за певним алгоритмом. Гама шифру є псевдовипадковою бінарною послідовністю, яка генерується для зашифрування даних та їх подальшого розшифрування.

Цей метод забезпечує високий рівень криптографічного захисту, адже ключова послідовність постійно змінюється. Гама для кожного нового тексту повинна бути унікальною. Якщо період гами перевищує довжину зашифрованого тексту, надійність шифру визначається довжиною ключа [3].

Вимоги до генератора гами шифру:

1. Довгий період: послідовність має бути достатньо довгою для шифрування повідомлень різної довжини.
2. Непередбачуваність: неможливість прогнозування наступного біта навіть за наявності попередніх даних і знання принципу роботи генератора.
3. Простота реалізації: створення генератора не повинно викликати значних технічних труднощів.

Довжина періоду гами є однією з найважливіших характеристик генератора псевдовипадкових чисел. Після завершення періоду числа в послідовності почнуть повторюватися, що робить їх вразливими до дешифрування. Величина періоду залежить від ступеня конфіденційності даних і обраного алгоритму генерації.

Для забезпечення непередбачуваності необхідно, щоб період був досить довгим, а всі можливі комбінації бітів певної довжини рівномірно розподілялися по всій послідовності.

Практична реалізація генератора може бути програмною або апаратною, але при цьому має забезпечувати необхідну швидкість роботи.

Історична довідка: Один із перших методів генерації псевдовипадкових чисел запропонував Джон фон Нейман у 1946 році. У цьому підході кожне нове число отримувалося шляхом зведення попереднього до квадрату з подальшим відкиданням крайніх цифр. Проте цей метод виявився ненадійним і з часом вийшов із використання.

1.6 Постановка завдання

Метою роботи є розробка криптографічної системи захисту інформації, яка повинна забезпечити необхідний рівень криптостійкості при шифруванні вхідних повідомлень за допомогою одноразового блокноту.

Для розроблення криптографічної системи захисту інформації із застосуванням одноразового блокноту необхідно виконати наступне:

1. Визначити завдання та основні функції, які буде виконувати система захисту інформації.

2. Виконати аналіз принципів поточного шифрування, шифрування методом гамування та методів генерації псевдовипадкових послідовностей. Спираючись на результати досліджень уточнити алгоритм роботи системи формування шифрів із застосуванням одноразових блокнотів.

3. Розробити схему алгоритму роботи та схему електричну структурну системи захисту інформації із застосуванням одноразового блокноту.

4. Розробити схему електричну функціональну та схему електричну принципову системи, що розроблюється.

5. Розрахувати собівартість та ціну системи.

2 НАУКОВО-ДОСЛІДНА ЧАСТИНА

2.1 Сучасний стан проблеми поточного шифрування

Для ефективного застосування поточних шифрів важливо детально оцінити їх властивості. Ефективність цих шифрів визначається співвідношенням параметрів стійкості, продуктивності та додаткових властивостей, які сприяють захисту інформації в інформаційно-телекомунікаційних системах (ІТКС). Під час оцінювання необхідно враховувати особливості ІТКС, такі як вимоги до продуктивності, сумісність із обладнанням та іншими системами. Аналіз властивостей алгоритмів і режимів шифрування забезпечує формалізований підхід до їх розробки.

Інтерес до поточних шифрів виник завдяки роботам Клода Шеннона, який досліджував одноразові гамми (шифр Вернама). Назва "одноразовий блокнот" стала популярною під час Другої світової війни, коли цей метод широко використовувався для шифрування повідомлень.

Одноразовий блокнот базується на використанні випадкової послідовності (гами), що накладається побітно на відкритий текст. Довжина цієї гами відповідає довжині повідомлення, а її використання можливе лише один раз. Такий підхід забезпечує абсолютну криптостійкість, однак вимагає значного обсягу унікальних гам для кожного повідомлення.

Поточні шифри: концепція та особливості

Поточні шифри намагаються відтворити концепцію одноразового блокнота, використовуючи короткий ключ для генерації псевдовипадкової шифруючої послідовності (гами). Ця послідовність повинна бути максимально схожою на випадкову. Генерація гами є центральним завданням у розробці поточних шифрів. Шифруюча послідовність створюється генератором гами, або генератором псевдовипадкової послідовності.

Вважається, що послідовність є випадковою, якщо її неможливо передбачити або описати простим чином. Якщо шифруюча гамма генерується ефективно, то, як правило, існує спосіб описати її закономірності. Для забезпечення високої криптостійкості генератор має формувати послідовність, яка не піддається аналізу і не дозволяє передбачити наступний елемент.

Рівень випадковості визначають за допомогою статистичних тестів, які перевіряють, чи відповідає послідовність необхідним вимогам. Криптографічно стійка псевдовипадкова послідовність повинна:

- бути непередбачуваною;
- складатися з елементів, що мають рівну ймовірність виникнення.

Поточні шифри мають низку переваг:

- Висока швидкість роботи, особливо у порівнянні з блоковими шифрами.
- Можливість генерувати шифруючу послідовність незалежно від тексту, що шифрується, чи самого шифротексту.
- Простота реалізації за рахунок швидких операцій гамування.

На початку розвитку криптографії популярними були генератори на основі регістрів зсуву з лінійним зворотним зв’язком і лінійні конгруентні генератори. Їх використовували через простоту апаратної та програмної реалізації. Проте, оскільки ці генератори базуються на лінійних алгоритмах, вони легко піддаються криптоаналізу. Сьогодні такі генератори не рекомендують використовувати самостійно, однак вони часто виступають компонентами складніших схем шифрування.

2.2 Загальні підходи до створення поточних шифрів

Серед сучасних підходів до створення поточних шифрів особливе місце займає концепція, розроблена швейцарським криптографом Райнером Рюппелем. У цій концепції виділено чотири основні підходи до побудови схем поточного шифрування. Вони базуються на різних припущеннях щодо можливостей і ресурсів криптоаналітика у процесі зламу шифру.

2.2.1 Підхід теорії інформації

Цей підхід ґрунтується на припущенні, що криптоаналітик має необмежені ресурси (час та обчислювальні можливості). Завдання криптоаналізу полягає у визначенні відкритого тексту або ключа, маючи лише шифротекст і попередні ймовірності для різних можливих ключів і текстів.

Шифрувальна система вважається абсолютно стійкою, якщо відкритий текст і зашифрований текст є статистично незалежними. Тобто криптоаналітик не отримує жодної переваги у зламі системи навіть після перегляду шифротексту. Таку систему називають ідеально стійкою, якщо незалежно від обсягу отриманих даних криптоаналітик не може визначити відкритий текст.

2.2.2 Підхід теорії систем

Цей підхід базується на створенні криптосистеми, яка становить складну та невідому проблему для криптоаналітика.

Розробник повинен переконатися, що жоден із базових принципів криптоаналізу не підходить для зламу системи. Для цього визначають набір конструктивних критеріїв для генераторів шифрувальної гами. До таких критеріїв належать:

- період гами;
- лінійна складність;
- статистичні властивості;
- розсіювання і складність.

Цей підхід є одним із найпоширеніших у практичній розробці криптосистем, оскільки дозволяє створювати шифри, які відповідають певним заданим параметрам.

2.2.3 Підхід теорії складності

Цей підхід ґрунтується на аналізі обчислювальної складності криптосистеми. Стійкість визначається довжиною ключа, а всі обчислення аналізуються з використанням асимптотичних методів.

Криптоаналіз у цьому випадку включає:

- передбачення окремих елементів шифрувальної гами;
- виявлення відмінностей між шифрувальною гамою і справді випадковою послідовністю.

Мета розробника полягає у створенні системи шифрування на основі проблеми, яка є обчислювально недосяжною. Ідеальний генератор гами повинен генерувати послідовності, які не піддаються аналізу та не виявляються жодними статистичними тестами за поліноміальний час. Проте такі генератори на сьогодні залишаються гіпотетичними.

2.2.4 Рандомізовані поточні шифри

Цей підхід передбачає використання аналітичного доказу, що криптоаналітична задача має необмежену складність.

У рамках цього підходу під час шифрування та розшифрування використовуються великі масиви випадкових даних. Секретний ключ вказує, які частини цих даних слід використовувати. Водночас опонент, не знаючи ключа, змушений обробляти весь масив, що робить криптоаналіз значно складнішим.

Стійкість рандомізованого поточного шифру оцінюється через середню кількість біт, які криптоаналітик повинен проаналізувати, щоб його шанси на злам перевищили випадкові здогадки. Очікувана кількість перевірок є нижньою межею для кількості кроків, необхідних для зламу системи.

Усі описані підходи спрямовані на створення генераторів псевдовипадкових послідовностей (ПВП), які унеможливлюють отримання достатньої кількості статистичних даних для криптоаналізу. Це досягається завдяки складним алгоритмам, які приховують секретні параметри або унеможливлюють ефективне відтворення гами.

2.3 Математичні принципи поточного шифрування

Шифрування в поточних шифрах здійснюється на основі складання ключової послідовності (гами) з відкритим текстом повідомлення. Складання здійснюється побітно за допомогою операції додавання за модулем два. Процес шифрування можна описати наступною формулою:

$$c_i = m_i \oplus g_i = \text{де } i = 1,2,3\ldots ,$$

де c_i — символ шифротексту;

m_i — символ відкритого тексту;

g_i— символ ключової послідовності.

Розшифровування виглядає таким чином:

$$m_i = c_i \oplus g_i = \text{де } i = 1,2,3\ldots$$

У якості символів можуть використовуватися як окремі біти, так й символи (байти). Таким чином, поточні шифри підходять для шифрування безперервних потоків даних.

У загальному вигляді схему шифру можна зобразити наступним чином (рис. 2.1).

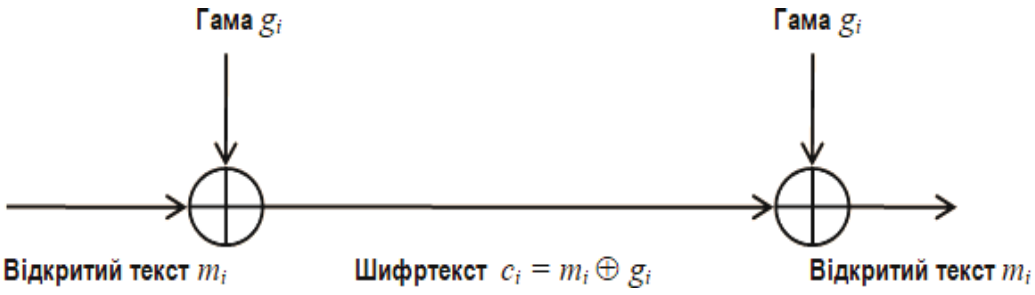

Рисунок 2.1 — Схема потокового шифру

Шифрування здійснюється за допомогою накладення шифруючої гами, й цей процес називається гамуванням. Гамма є ключем шифрування. Однак використання ключа, рівного за розміром зашифрованим даним, є проблематичним. Через це поточні шифри генерують вихідну гамму на основі іншого секретного ключа невеликого розміру.

Основна задача поточних шифрів полягає у створенні ключового потоку (вихідної гами) для шифрування. Поточні шифри поділяються на:

- синхронні;
- самосинхронізуючі (асинхронні).

Якщо ключовий потік (вихідна гамма) генерується незалежно від вихідного та зашифрованого тексту, то використовується синхронне поточне шифрування.

У цьому випадку схема виглядає таким чином (рис. 2.2).

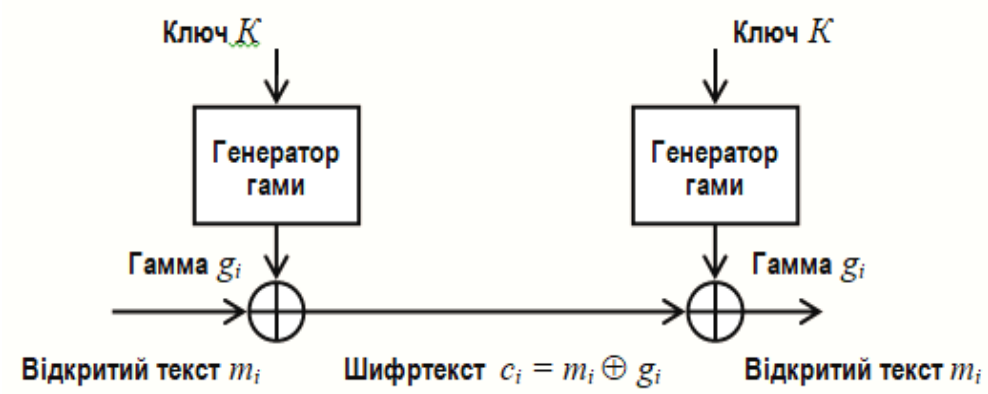

Рисунок 2.2 — Реалізація синхронного потокового шифру

Шифр генерує гамму на основі секретного ключа, яка накладається на відкритий текст, а результат передається іншому абоненту. Розшифровування виконується аналогічно.

Блок, який генерує шифруючу гамму, називається генератором гами або генератором псевдовипадкової гами (PRG, Pseudo Random Generator). Будь-який блоковий шифр у режимі OFB є синхронним поточним шифром.

До псевдовипадкової послідовності накладаються певні вимоги, які визначають її придатність для шифрування. Наприклад, у послідовності не повинно бути довгих серій нулів або одиниць, адже це може призвести до розкриття тексту.

Генератори гами створюють псевдовипадкові послідовності, які залежать від ключа шифрування, але за своїми характеристиками нагадують випадкові. Завдання полягає в тому, щоб за наявності частини послідовності було неможливо передбачити наступні біти. Крім того, “нулі” і “одиниці” на виході повинні бути рівноймовірними. Це перевіряється за допомогою статистичного аналізу.

Генерація псевдовипадкових послідовностей є однією з найважливіших задач криптографії. Сьогодні існує велика кількість генераторів таких послідовностей і статистичних тестів для оцінки їх якості.

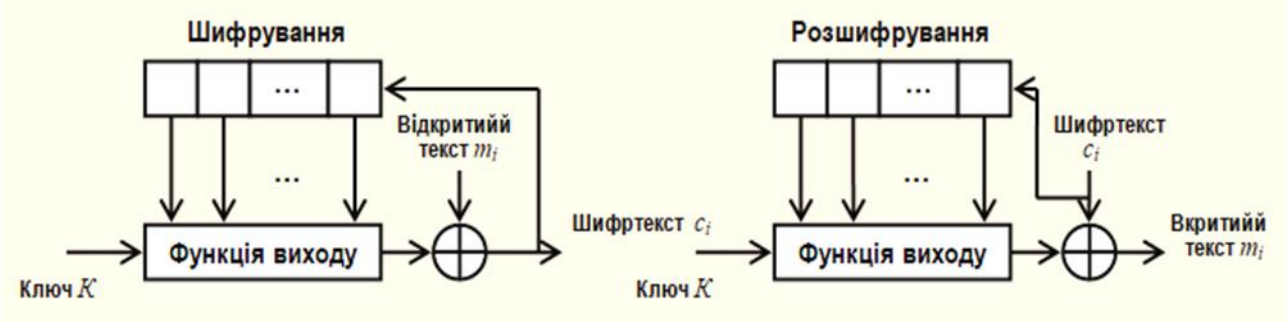

Рисунок 2.3 — Реалізація самосинхронізучогося потокового шифру

Вимоги до синхронних поточних шифрів:

1. Синхронізація. Відправник і отримувач повинні бути синхронізовані для генерування однакових ключових потоків. У разі порушення синхронізації процес розшифровування буде некоректним.

2. Відсутність розмноження помилок. Зміна одного символу шифротексту під час передачі не повинна впливати на інші символи.

3. Стійкість до активних атак. Вставка або видалення символів у шифротексті призводить до порушення синхронізації та виявляється отримувачем.

У самосинхронізуючихся шифрах кожен символ ключового потоку залежить від фіксованої кількості попередніх символів шифротексту. Їхня схема виглядає так:

До самосинхронізуючихся шифрів належать блочні шифри, які працюють у режимі CFB.

Властивості самосинхронізуючихся шифрів:

1. Самосинхронізація. У разі видалення або вставки символу процес розшифровування тимчасово порушується, але згодом синхронізація відновлюється.

2. Обмежене розмноження помилок. Помилка у шифротексті впливає лише на обмежену кількість символів, після чого розшифрування відновлюється.

3. Вразливість до активних атак. Будь-яка зміна символів може бути помічена, однак вставка або видалення символів складніше визначається.

4. Розсіювання статистики відкритого тексту. Кожен символ відкритого тексту впливає на весь наступний шифротекст, завдяки чому статистичні властивості відкритого тексту приховуються.

Таким чином, ключове завдання полягає у створенні ключової послідовності, яка забезпечує стійкість даних до всіх типів атак.

2.4 Шифрування методом гамування

Гамування — це процес накладання шифрувальної гами на відкриті дані за певним законом. Шифрувальна гама являє собою псевдовипадкову послідовність, що генерується за визначеним алгоритмом і використовується як для шифрування вихідних даних, так і для розшифрування зашифрованої інформації.

Оскільки псевдовипадкові послідовності формуються алгоритмічно, вони не є повністю випадковими. Проте зазвичай вважається, що вони мають властивості, характерні для випадкових послідовностей.

Процес шифрування полягає у створенні шифрувальної гами та накладанні її на відкритий текст з використанням оборотної операції, наприклад, додавання за модулем 2. Перед початком шифрування вихідний текст розбивається на блоки фіксованої довжини, зазвичай по 64 біти. Шифрувальна гама генерується у вигляді послідовності блоків такої ж довж. Рівняння для процесу шифрування можна подати у такій формі:

$$T_ш(i) = (Г_ш(i) + T_в(i)) mod 2,\ i=1 \dots M,$$

де $T_ш(i)$ - і-й блок шифртексту;

$Г_ш(i)$ - і-й блок гами шифру;

$T_в(i)$ - і-й блок відкритого тексту;

M - кількість блоків відкритого тексту.

Процес розшифрування зводиться до повторної генерації гами шифру і накладенню цієї гами на зашифровані дані. Рівняння розшифрування має наступний вигляд:

$$T_в(i) = (Г_ш(i) + T_ш(i))mod2,\ i=1 \ldots M.$$

Шифртекст, отриманий таким методом, є досить стійким до розкриття, оскільки ключ має властивість змінюватися. Шифрувальна гамма повинна випадково змінюватися для кожного блоку даних, що піддається шифруванню. Якщо період повторення гами перевищує довжину всього шифрованого тексту, а зловмиснику не відома жодна частина вихідного тексту, то розкрити шифр можна лише шляхом повного перебору всіх можливих значень ключа. У такому випадку криптографічна стійкість шифру визначається довжиною ключа (періодом, протягом якого гамма не повторюється).

Завдяки можливості генерації практично нескінченної шифрувальної гами з використанням комп'ютера, цей метод є одним із найпоширеніших для захисту інформації в автоматизованих системах.

Найпростішою та водночас найнадійнішою схемою шифрування є схема одноразового використання, яку часто пов'язують із винаходом Г. С. Вернама (див. рис. 2.4).

Формується t-розрядна випадкова двійкова послідовність – ключ шифру, відомий відправнику і одержувачу повідомлення.

Відправник виробляє побітове додавання за модулем 2 ключа та t-розрядної двійкової послідовності, що відповідає повідомленню, що пересилається:

$$c_i = m_i \oplus k_i,\ i = \overline{1,\ t},$$

де m_i, k_i, c_i – чергові i-ті біти відповідно до вихідного повідомлення, ключа та зашифрованого повідомлення;

t – кількість бітів відкритого тексту.

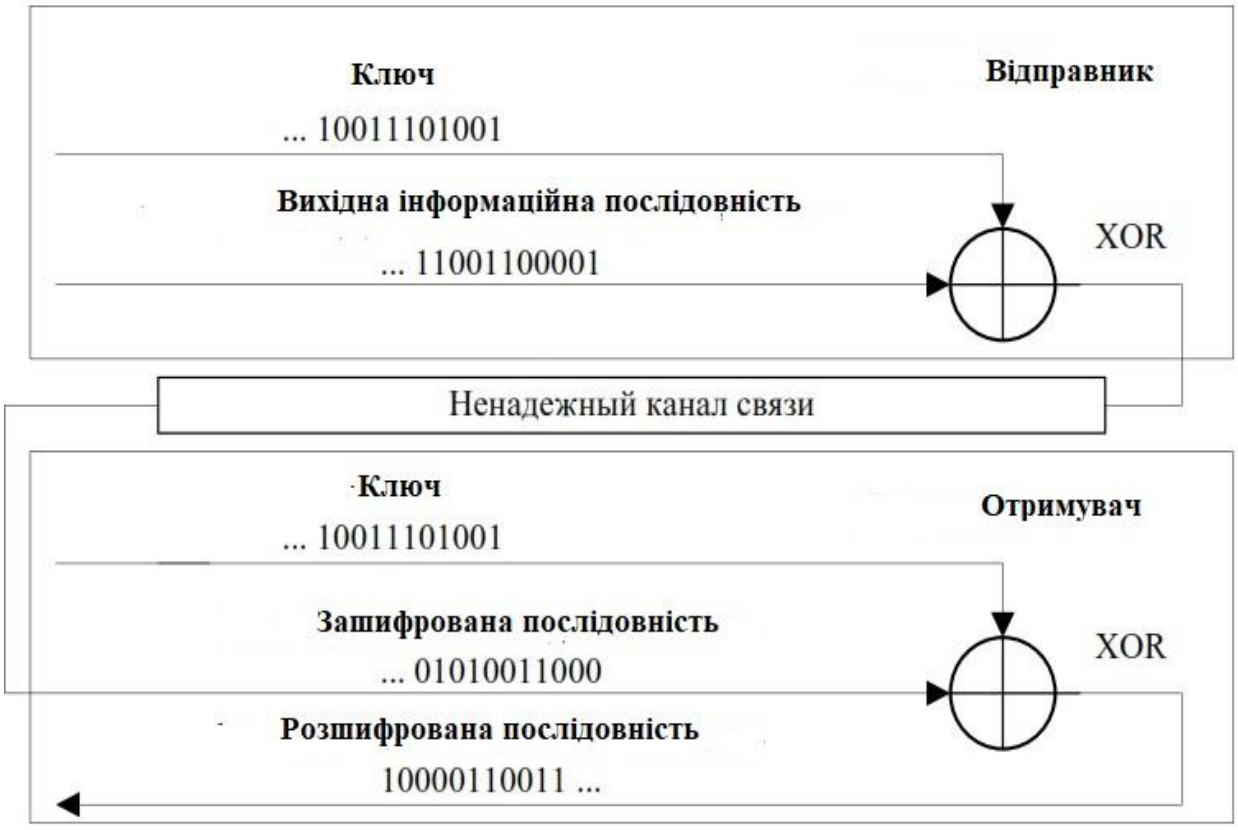

Рисунок 2.4 – Схема одноразового використання

Процес розшифрування зводиться до повторної генерації ключової послідовності та накладання її на зашифровані дані. Рівняння розшифрування має наступний вигляд:

$$m_i = c_i \oplus k_i,\ i = \overline{1,\ t}.$$

Клодом Шенноном доведено, що якщо ключ є фрагментом випадкової двійкової послідовності з рівномірним законом розподілу (причому його довжина дорівнює довжині вихідного повідомлення) і використовується цей ключ тільки один раз, після чого знищується, такий шифр є абсолютно стійким. Його неможливо розкрити, навіть якщо криптоаналітик має в своєму розпорядженні необмежений запас часу та необмежений набір обчислювальних ресурсів. Дійсно, противнику відомо лише зашифроване повідомлення c, при цьому всі різні ключові послідовності k є можливими та

рівноймовірними, отже, є можливими й будь-які повідомлення m, тобто криптоалгоритм не дає жодної інформації про відкритий текст.

Метою противника може бути розкриття криптосистеми, знаходження ключа, в крайньому випадку, дешифрування будь-якого закритого повідомлення. Однак його може задовільнити отримання навіть деякої ймовірної інформації про вихідний текст повідомлення.

Наприклад, відомий криптоаналітику факт написання тексту деякого повідомлення англійською мовою надає йому деяку апріорну інформацію про це повідомлення навіть до аналізу шифрування. У даному випадку він заздалегідь знає, що слово «HELLO» є більш ймовірним початком повідомлення, ніж набір літер «FGHKM». Тому однією із цілей криптоаналізу може стати збільшення інформації, що стосується кожного можливого повідомлення таким чином, щоб правильний текст був вірогіднішим.

Припустимо, противник перехопив шифр «ABCCD» і знає (або припускає), що використаний шифр – це шифр простої заміни. Аналіз шифрування дозволяє зробити висновок, що вихідне повідомлення складається з п’яти літер, причому на третьої та четвертої позиціях стоїть та сама літера, а інші літери відрізняються від неї і різні між собою. Противник не може вважати, що це повідомлення «HELLO», оскільки є й інші можливі повідомлення, наприклад «TEDDY». Проте апостеріорні ймовірності цих відкритих текстів зростають відносно їх апріорних ймовірностей. В цей час апостеріорна ймовірність таких відкритих текстів, як «PEACE» або «GATES», знижується до нуля незалежно від їх апріорної ймовірності. За Шенноном, в абсолютно секретних криптосистемах після аналізу закритих текстів апостеріорні ймовірності можливих відкритих текстів залишаються такими, якими були їхні апріорні ймовірності.

Необхідними та достатніми умовами абсолютної стійкості шифру є:

– повна випадковість ключа;
– рівність довжин ключа та відкритого тексту;
– одноразове використання ключа.

Абсолютна стійкість розглянутої схеми оплачується занадто великою ціною, вона є надзвичайно дорогою та непрактичною.

Основний її недолік – рівність обсягу ключової інформації та сумарного обсягу переданих повідомлень. Застосування схеми виправдане лише в каналах зв’язку, що рідко використовуються, для шифрування виключно важливих повідомлень.

Існує велика кількість модифікацій наведеної схеми, найбільш відомою з яких є схема, яка заснована на використанні одноразових шифрувальних блокнотів (рис. 2.5).

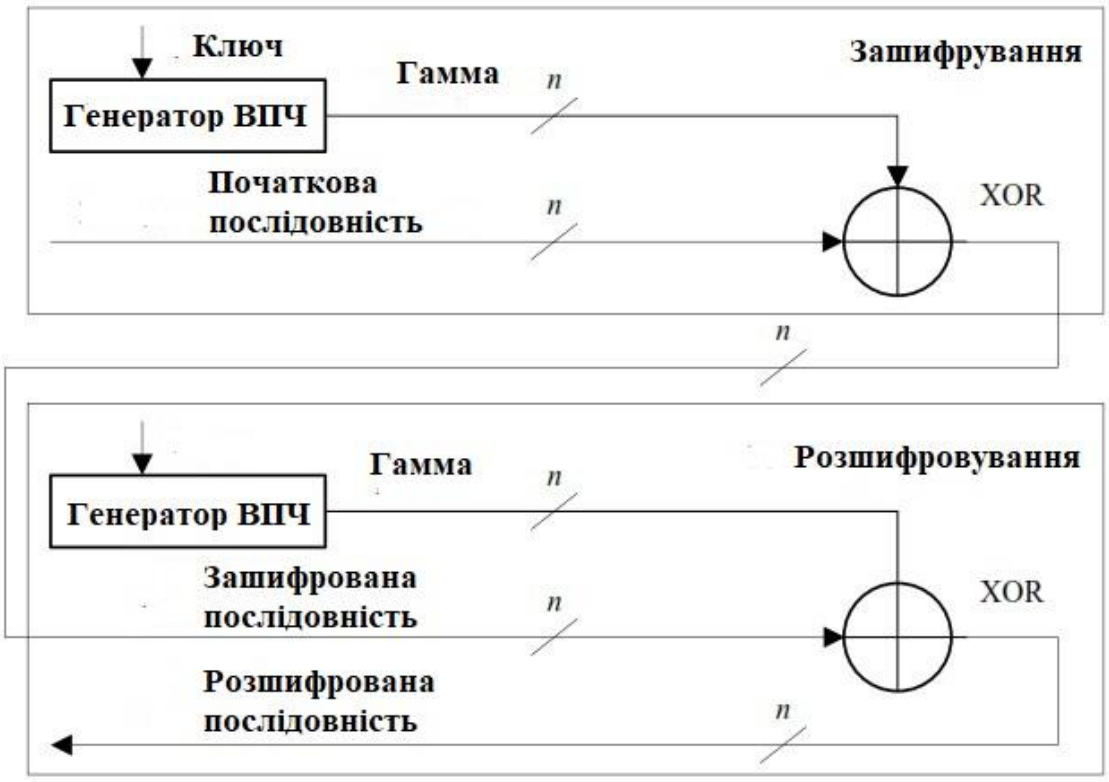

Рисунок 2.5 – Шифрування інформації методом гамування

Таким чином, побудувати ефективний криптоалгоритм можна лише відмовившись від абсолютної стійкості. Виникає завдання розроблення теоретично нестійкого шифру, для розкриття якого противнику потрібно було б виконати кількість операцій, які можуть бути виконані на сучасних і очікуваних у найближчій перспективи обчислювальних засобів за розумний час. Насамперед слід мати схему, яка використовує ключ невеликої розрядності, який надалі породжує значно більш довгу ключову послідовність.

Цей результат може бути досягнутий при використанні гамування, схема якого наведена на рисунку 2.5.

Надійність шифрування методом гамування визначається якістю генератора гами.

Розрізняють гамування з кінцевою та нескінченною гамами. У першому випадку джерелом гами є апаратний або програмний генератор псевдовипадкової послідовності (ПВП). Прикладом нескінченної гами може бути послідовність цифр у десятковому записі числа π = 3,1415926….

Якщо множиною використовуваних для шифрування знаків є алфавіт, відмінний від бінарного ($Z_2 = \{0;1\}$), наприклад, алфавіт Z_{33} – українські (або російські) літери і пробіл, то його символи і символи гами замінюються цифровими еквівалентами, які потім підсумовуються за модулем N:

$$c_i = (m_i + \gamma_i) \bmod N,\ i = \overline{1,\ t},$$

де m_i, γ_i, c_i – черговий i-й знак вихідного повідомлення, гами та шифртексту відповідно;

t – кількість знаків відкритого тексту;

N – кількість символів в алфавіті.

2.5 Методи генерації псевдовипадкових послідовностей чисел

2.5.1 Формування псевдовипадкових послідовностей

Зазвичай, враховуючи, що обчислювальні пристрої самі по собі є детермінованими, отримати істинно випадкову послідовність за їх допомогою неможливо. Тим більше складно сформувати послідовність, яка складалася б з рівномірно розподілених комбінацій бінарних символів різної конфігурації.

Вимога рівноймовірності сформованих символів ключової гамми є обов’язковою. Незважаючи на те, що сформована шифрувальним пристроєм ключова послідовність детермінована, її можна вважати достатньо

випадковою для криптоаналітика супротивника, якщо він не здатний передбачити кожен наступний символ на основі аналізу попередньої послідовності з ймовірністю, відмінною від 0,5.

Ще однією важливою вимогою до сформованої гамми є наявність максимально великої довжини її періоду ТТТ. Як зазначено у [3], при невеликому ТТТ криптоаналітику супротивника відносно легко визначити метод побудови генератора гамми.

На сьогодні існує велика кількість простих генераторів, які розглядаються як елементарні й досить добре вивчені. Їх параметри, що забезпечують максимальну довжину періоду, зведені в таблиці й можуть використовуватися без проведення додаткових досліджень. До таких генераторів належать:

- лінійні конгруентні генератори (ЛКГ);
- генератори чисел Фібоначчі;
- генератори Хаффмана, побудовані на основі регістрів зі зворотним зв’язком (РЗЗ).

У [2] зазначено, що майже всі згадані генератори не можуть бути використані для формування гамми безпосередньо. Це пояснюється тим, що не існує загальної теорії їх побудови для криптографічних цілей. Зазвичай творці криптосистем визначають параметри генераторів методом проб і помилок. У літературі можна знайти конкретні реалізації, параметри яких подані у вигляді таблиць. Наприклад, у [4] містяться параметри ЛКГ, що забезпечують максимально можливу довжину періоду послідовності без циклів.

З огляду на це, параметри "хороших" генераторів є загальновідомими, і їх кількість відносно невелика. Відтак завдання криптоаналітика, яке полягає у переборі можливих варіантів застосовуваного генератора, значно спрощується. Звідси випливає висновок, що формувач шифрувальної гамми повинен бути складним. Тобто, він має являти собою комбінацію елементарних генераторів, здатних забезпечити максимально можливу довжину періоду послідовності рівномірно розподілених символів.

Побудова потокового шифру передбачає, насамперед, створення якісного генератора псевдовипадкових послідовностей (ПВП), який формує

гаму для шифрування. Період генератора має бути завідомо більшим за розмір будь-якого файлу, що шифрується у системі.

Генератор гами повинен бути складним, а його елементи мають включати, зокрема, лінійні конгруентні генератори, які вирізняються простотою програмної реалізації та економним використанням обчислювальних ресурсів.

Основною вимогою до генератора ПВП є рівномірність розподілу бінарних символів у вихідній послідовності.

При шифруванні методом гамування в якості ключа використовується випадковий рядок бітів, який об'єднується з відкритим текстом, також представленим в двійковому вигляді (наприклад A = 00000, B = 00001, C = 00010 й т.д.), за допомогою побітного додавання по модулю 2, й в результаті отримується шифрований текст.

Приклад:

Нехай
Г = 10110100 – гама шифру
А = 10101010 – відкритий текст

Пряме перетворення:
⊕10101010
 10110100
 А' = 00011110 – шифртекст

Зворотне перетворення:
⊕00011110
 10110100
 А = 10101010 – вихідний (відкритий) текст

Генерація непередбачуваних двійкових послідовностей великої довжини є однією з ключових задач класичної криптографії. Для цього часто застосовуються генератори псевдовипадкових двійкових послідовностей.

Псевдовипадкові числові послідовності, що генеруються такими пристроями чи програмами, зазвичай називаються гамою шифру або просто гамою (від літери грецького алфавіту, що часто позначає випадкові величини у математичних формулах).

Хоча програми, які використовуються для створення псевдовипадкових чисел, називають генераторами випадкових чисел, насправді вони формують детерміновані послідовності, які мають властивості, подібні до справжньої випадковості.

До криптографічно стійких генераторів псевдовипадкових чисел (гами шифру) висуваються наступні основні вимоги:

1. **Великий період послідовності**. Гама повинна мати достатньо довгий період, щоб забезпечити шифрування даних будь-якої довжини. Якщо період є коротким, послідовність почне повторюватися, що дозволить її передбачити.

2. **Непередбачуваність**. Навіть якщо відомий тип генератора або частина вже згенерованої послідовності, неможливо повинно бути вгадати наступний біт. Це є ключовим критерієм криптографічної стійкості.

3. **Технічна реалізація**. Генератор повинен бути простим у реалізації як на апаратному, так і на програмному рівнях, забезпечуючи при цьому необхідну швидкодію.

Період гами є одним із найважливіших параметрів генератора. Коли він вичерпується, числа починають повторюватися, що робить послідовність вразливою для криптоаналізу. Тривалість періоду залежить від конкретного алгоритму генерації та довжини ключа: чим довшим є ключ, тим складніше його зламати.

Непередбачуваність гами – інша важлива вимога, що пов’язана зі складністю визначення критеріїв випадковості. На сьогодні не існує універсальних методів або критеріїв, які б однозначно підтверджували, що певна псевдовипадкова послідовність є непередбачуваною. Для цього необхідно, щоб період послідовності був дуже великим, а різні комбінації бітів однакової довжини зустрічалися рівномірно.

Практична реалізація генератора також повинна забезпечувати достатню продуктивність, що дозволить використовувати його для шифрування великих обсягів даних у реальному часі.

Одним із перших методів генерації псевдовипадкових чисел на комп'ютерах був метод, запропонований Джоном фон Нейманом у 1946 році. Суть полягала у піднесенні до квадрата попереднього числа та відкиданні крайніх цифр. Проте цей метод виявився ненадійним і був швидко відкинутий.

Із відомих процедур генерації послідовності псевдовипадкових цілих чисел найбільш часто застосовується так званий **лінійний конгруентний генератор**.

Цей генератор виробляє послідовність псевдовипадкових чисел $Y_1, Y_2, \ldots, Y_{i-1}, Y_i, \ldots$, використовуючи співвідношення:

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

де Y_i - і-те (поточне) число послідовності;

Y_{i-1} - попереднє число послідовності;

a, b, m – константи;

m – модуль;

a – множник (коефіцієнт);

b – приріст;

Y_0 – число, що породжує (початкове значення).

Поточне псевдовипадкове число Y_i отримують із попереднього числа Y_{i-1} множенням його на коефіцієнт a, додаванням з приростом b і обчисленням залишку від ділення на модуль m. Дане рівняння генерує псевдовипадкові числа з періодом повторення, який залежить від обраних значень параметрів a, b і m й може досягати значення m. Значення модуля m обирається таким, щоб воно дорівнювало 2^n або простому числу, наприклад $m = 2^{31} - 1$. Приріст b повинен бути взаємно простим із m, коефіцієнт a повинен 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 мають хороші статистичні властивості. Однак така довжина періоду є замалою для криптографічного застосування. Крім того, доведено, що послідовності, що генеруються конгруентними генераторами, не є криптографічно стійкими.

Існує спосіб генерації послідовностей псевдовипадкових чисел на основі **лінійних рекурентних співвідношень**.

Розглянемо рекурентні співвідношення та їхні різницеві рівняння.

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

$$a_{i+k} = -\sum_{j=0}^{k-1} h_j a_{i+j}.$$

де $h_0 = 0$, $h_k = 1$ та кожне h_i належить полю GF(q).

Вирішенням цих рівнянь являється послідовність елементів $a_0, a_1, a_2, \ldots$ поля GF(q). Співвідношення визначає правило обчислення a_k за відомим значенням величин $a_0, a_1, a_2, \ldots, a_{k-1}$. Потім за відомим значенням $a_0, a_1, a_2, \ldots, a_k$ знаходять a_{k-1} й т.д. В результаті за початковими значеннями $a_0, a_1, a_2, \ldots, a_{k-1}$ можна побудувати нескінченну послідовність, при чому кожний її наступний член визначається із k попередніх. Послідовності такого виду легко реалізуються на комп'ютері, при цьому реалізація виходить особливо простою, якщо всі h_i та a_i набувають значень 0 та 1 із поля GF(2).

На рисунку 2.6 наведена лінійна послідовна перемикальна схема, яка може бути використана для обчислення розглянутих рекурентних співвідношень й для обчислення значення a_k за значеннями k попередніх членів послідовності. Початкові величини $a_0, a_1, a_2, \ldots, a_{k-1}$ розміщують в

розряди регістру зсуву, послідовні зсуви вмісту якого відповідають обчисленню послідовних символів, при цьому вихід після і-го зсуву дорівнює a_i. Даний пристрій називають генератором послідовних чисел, побудованим на базі регістру зсуву з лінійним зворотним зв'язком.

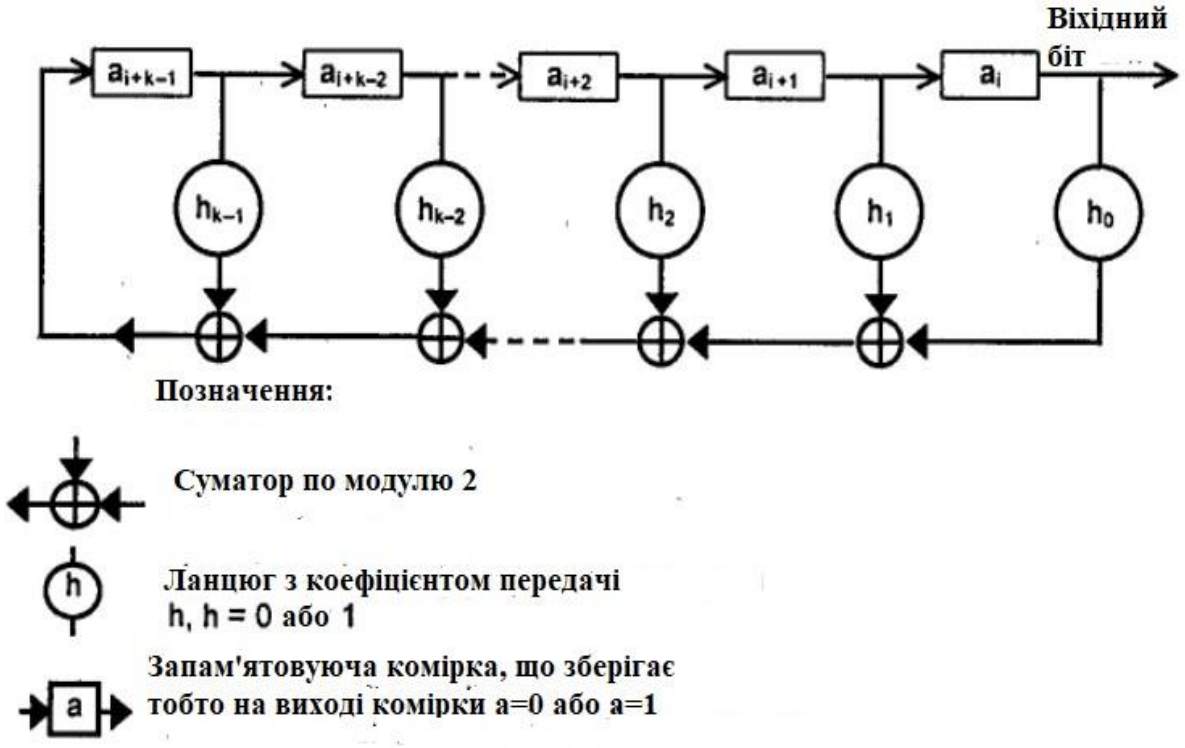

Рисунок 2.6 – Генератор з регістром зсуву

Розглянемо в якості прикладу трирозрядний регістр зсуву з лінійним зворотним зв'язком (рис. 2.6), що побудований відповідно до примітивного многочлена що не приводиться

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

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

Нехай ключем є 101. Регістр починає працювати з цього стану. Послідовність станів регістра наведена на рисунку 2.7.

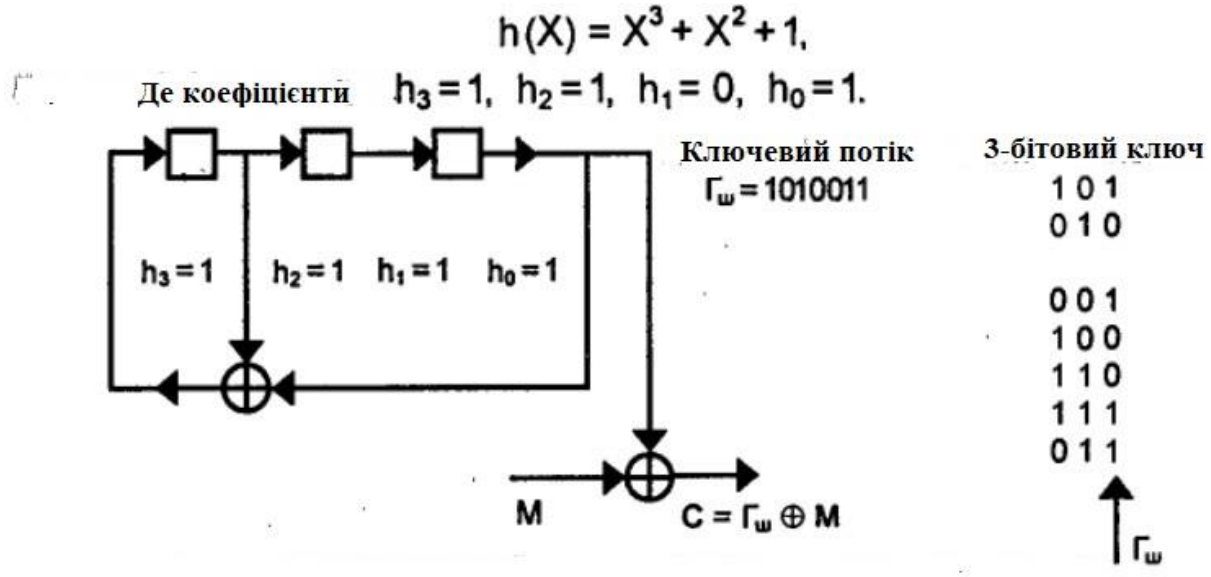

Рисунок 2.7 – Трирозрядний регістр зсуву зі зворотними зв’язками

Регістр проходить через усі сім ненульових станів і знову повертається у свій початковий стан 101. Це – найбільш довгий період даного регістру з лінійним зворотним зв’язком. Така послідовність називається **послідовністю максимальної довжини** для регістру зсуву (Maximal Length Shift Register Sequence – MLSRS).

Пітерсон і Уелдон довели, що для будь-якого цілого числа *m* існує m-бітна послідовність максимальної довжини (MLSRS) з періодом 2m-1. Наприклад, якщо *m = 100*, то така послідовність матиме період $2^{100} – 1$ і при передачі зі швидкістю 1 Мбіт/с не повторюватиметься протягом 10^{16} років.

У нашому прикладі вихідна послідовність (гама шифру) *Гш*, що генерується регістром зсуву зі зворотним зв'язком, має вигляд 1010011, яка циклічно повторюється. Ця послідовність містить чотири одиниці та три нулі, а їх розташування є максимально близьким до рівномірного розподілу для послідовності довжиною 7. Якщо аналізувати пари послідовних бітів, то пари 10 і 01 зустрічаються по два рази, а пари 00 і 11 — по одному разу. Це також вказує на близькість до рівномірного розподілу. У випадку послідовностей максимальної довжини для *m*-розрядного регістра зсуву ця властивість рівномірного розподілу зберігається для трійок, четвірок і до *m*-бітових груп. Завдяки такій властивості послідовності максимальної довжини широко застосовуються в криптографічних системах як псевдовипадкові

послідовності, що імітують роботу криптостійкої системи одноразового шифрування.

Однак, попри імітацію криптостійкої системи одноразового шифрування, сама система на основі MLSRS не є стійкою. За наявності відомого відкритого тексту її можна розкрити за кілька секунд за допомогою комп'ютера. Якщо відводи регістра зворотного зв'язку зафіксовані, то для визначення початкового стану регістра достатньо знати *m* бітів відкритого тексту. Щоб отримати *m* бітів ключового потоку, *m* бітів відомого відкритого тексту складаються за модулем 2 з відповідними *m* бітами шифртексту. У результаті виходить стан регістра зсуву на певний момент часу у зворотному напрямку. Далі, моделюючи роботу регістра у зворотному порядку, можна обчислити його початковий стан.

Для підвищення криптографічної стійкості послідовностей максимальної довжини (MLSRS) можна використовувати нелінійну логіку. Один із варіантів передбачає застосування нелінійно «фільтрованого» вмісту регістра зсуву як ключового потоку, тоді як для отримання послідовності максимальної довжини використовується лінійний зворотний зв'язок. Такий підхід ілюструється на рисунку 2.8

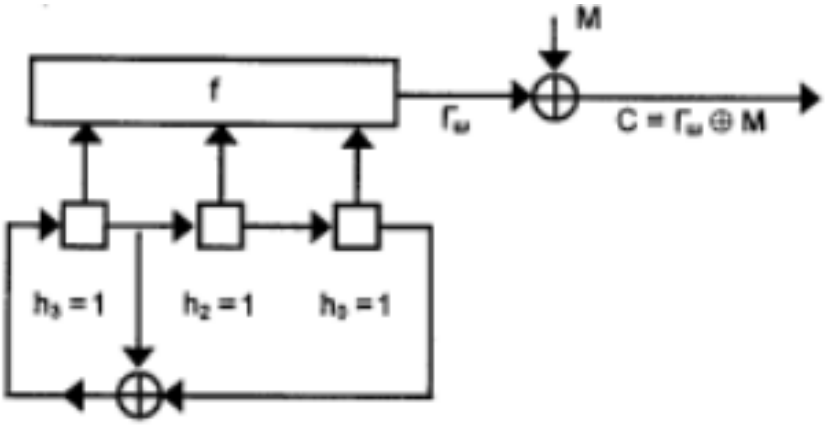

Рисунок 2.8 – Лінійний регістр зсуву з нелінійними логічними ланцюгами на виході

Функція *f* повинна вибиратись таким чином, щоб забезпечити хороший баланс між нулями і одиницями, й щоб фільтрована послідовність мала розподіл, близький до рівномірного. Необхідно також, щоб фільтрована послідовність мала великий період. Якщо $(2^m - 1)$ є простим числом (як в

прикладі - при m = 3 маємо $2^3 - 1 = 7$), то фільтрована послідовність може мати період $(2^m - 1)$ (при виборі структури регістра зсуву відповідно до примітивного багаточлена *h(X)* ступеня *m*, що не приводиться).

До таких значень *m* відносяться наступні: 2, 3, 5, 7, 13, 17, 19, 31, 61, 89, 107, 127, 521, 607, 1279, 2203, 2281, а отримані таким чином прості числа називаються простими числами Мерсена.

Не дивлячись на те, що фільтровану вихідну послідовність зазвичай неможна отримати за допомогою *m*-розрядного регістра здвигу з лінійним зворотним зв'язком, її завжди можна отримати за допомогою регістра зсуву більшої довжини з лінійним зворотним зв'язком. Регістр довжиною $(2^m - 1)$ завжди дозволить це зробити, а іноді придатний і більш короткий регістр.

Ще більш привабливим є використання в ланцюгу зворотного зв'язку нелінійної логіки, однак теорія таких схем недостатньо добре розглянута (у відкритій літературі).

Впровадження сучасних інфокомунікаційних систем вимагає посилення вимог до безпеки інформації, що надає особливої актуальності пошуку високопродуктивних алгоритмів захисту інформації, що передається, з необхідною криптографічною стійкістю.

Аналіз методів захисту інформації показав, що при використанні асиметричних шифрів відсутня необхідність пересилання секретних ключів, але реалізація таких алгоритмів потребує виконання складних обчислень і, відповідно, вимагає більше часу для шифрування в порівнянні з симетричними шифрами. Тому доцільно розглянути можливість використання алгоритмів симетричного шифрування, які характеризуються швидким шифруванням з високою криптостійкістю.

За результатами дослідження пропонується використання методу гамування вхідних повідомлень, який забезпечує найбільшу криптостійкість за умови використання гами довжиною не менше ніж довжина вхідного повідомлення. Різновидом такого методу шифрування є метод книжкового гамування, який дозволяє використовувати в якості гами сторінки шифрувального блокноту. Принцип шифрування полягає у заміні символів вхідного повідомлення і символів гами цифровими еквівалентами, які потім підсумовуються за модулем N, де N – кількість символів у алфавіті, що застосовується. Неможливість проведення частотного аналізу зашифрованого

таким методом повідомлення значно підвищує стійкість даного шифру до несанкціонованого розшифрування. Складність передачі гами шифру отримувачу зашифрованих повідомлень пропонується подолати шляхом формування множини шифрувальних блокнотів (можливо із застосуванням відкритих джерел) і алгоритму вибору сторінок блокноту для здійснення операцій шифрування/розшифрування.

Запропонований метод книжкового гамування є більш ефективним при апаратній реалізації, що дозволить в інфокомунікаційних системах забезпечити швидке шифрування з високим рівнем криптостійкості [9].

3 РОЗРОБЛЕННЯ, ОБҐРУНТУВАННЯ АЛГОРИТМУ ФУНКЦІОНУВАННЯ ТА СТРУКТУРНОЇ СХЕМИ ПРИСТРОЮ, ЩО ПРОЕКТУЄТЬСЯ

3.1 Розроблення алгоритму роботи пристрою захисту інформації

У цій роботі розглянуто ідеальний шифр, відомий як шифр з використанням одноразового блокнота. Суть методу полягає в тому, що відправник використовує кожен символ із блокнота для шифрування лише одного символу відкритого тексту повідомлення, причому кожен блокнот застосовується тільки один раз.

Основною перевагою такого шифру є його абсолютна стійкість до частотного аналізу, що значно ускладнює несанкціоноване розшифрування. Однак головним недоліком цього методу є труднощі з передачею одноразового блокнота.

Для вирішення цієї проблеми було запропоновано використання заздалегідь визначеної, статичної та загальнодоступної бібліотеки літературних текстів, яка виступає як набір різноманітних комбінацій одноразових блокнотів.

Однозначність вибору блокнота між відправником і одержувачем забезпечується ключем, який не передається між сторонами, а визначається за наперед заданим алгоритмом із відкритого джерела. Це дозволяє кожному учаснику обміну самостійно генерувати необхідний блокнот, усуваючи потребу в його передачі.

Шифрування здійснюється шляхом посимвольної заміни символів вихідного повідомлення на порядкові номери відповідних символів у одноразовому блокноті.

Алгоритм роботи пристрою наведений на рисунку 3.1.

Алгоритм функціонування пристрою захисту інформації на базі методу книжкового гамування полягає у наступному.

Крок 1.Введення відкритого тексту

Крок 2. Перевірка на здійснення введення тексту

Крок 3. Визначення алфавіту

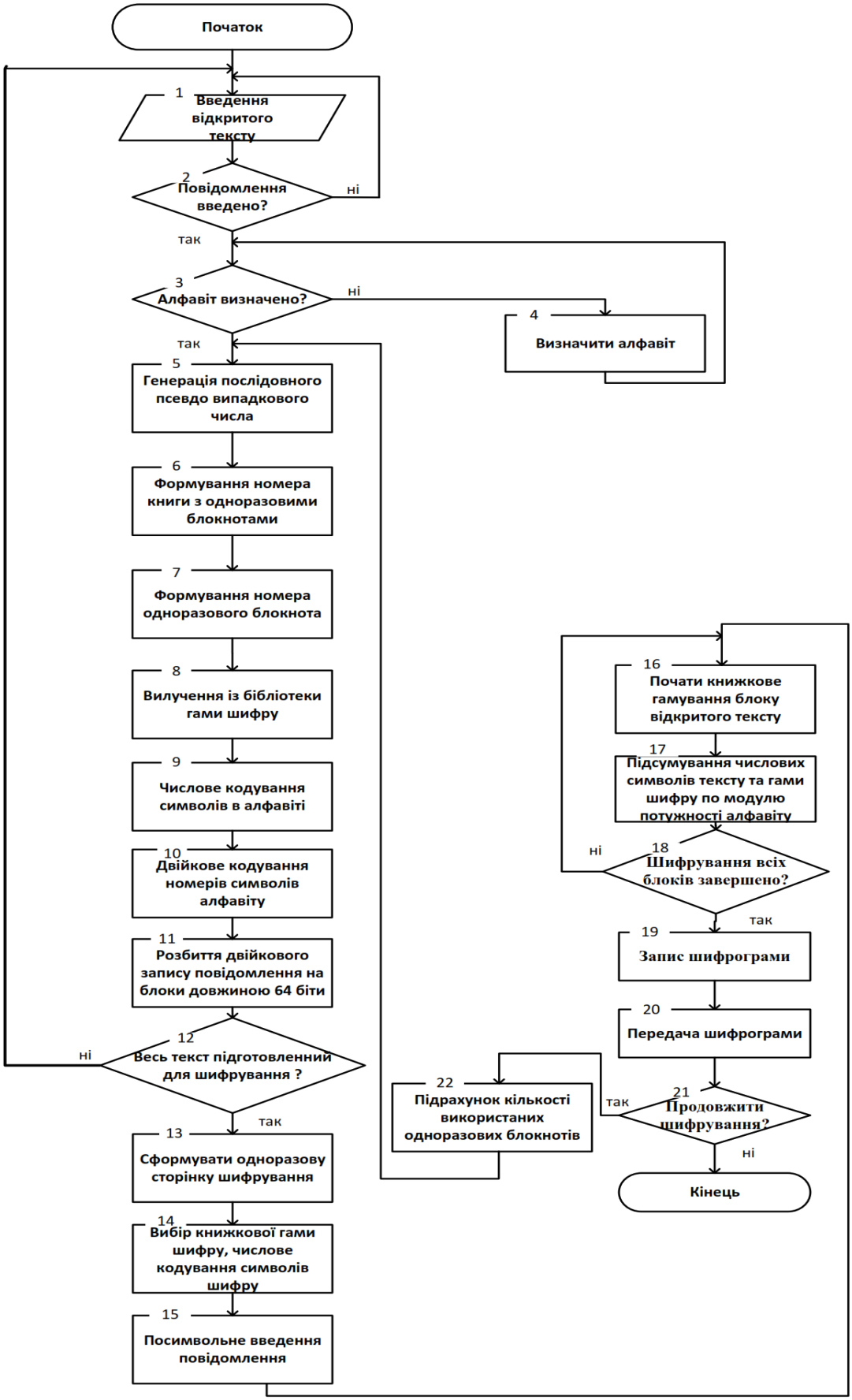

Рисунок 3.1 - Схема алгоритму роботи пристрою

Крок 4. Виконується числове кодування символів в алфавіті.

Крок 5. Генерація послідовного псевдо випадкового числа.

Крок 6. Формування номера книги з одноразовими блокнотами.

Крок 7. Формування номера одноразового блокнота.

Крок 8. Вилучення із бібліотеки гами шифру.

Крок 9. Числове кодування символів в алфавіті.

Крок 10. Двійкове кодування номерів символів алфавіту

Крок 11. Виконується розбиття двійкового запису повідомлення на блоки довжиною 64 біти.

Крок 12. Перевірка готовності тексту до шифрування.

Крок 13. Формування одноразової сторінки шифрування.

Крок 14. Вибір книжкової гами шифру, числове кодування символів шифру.

Крок 15. По символьне введення повідомлення.

Крок 16. Початок книжкового гамування відкритого тексту.

Крок 17. Підсумування числових символів тексту та гами шифру по модулю потужності алфавіту.

Крок 18. Перевірка чи всі блоки зашифровано.

Крок 19. Запис шифрограми.

Крок 20. Передача шифрограми.

Крок 21. Запит за продовження роботи.

Крок 22. Підрахунок кількості використаних одноразових блокнотів.

3.2 Розробка схеми електричної структурної пристрою

На основі алгоритму функціонування розробляється структурна схема пристрою. Вона являє собою сукупність блоків з відображенням відповідних зв’язків між ними.

На рисунку 3.2 представлена структурна схема пристрою.

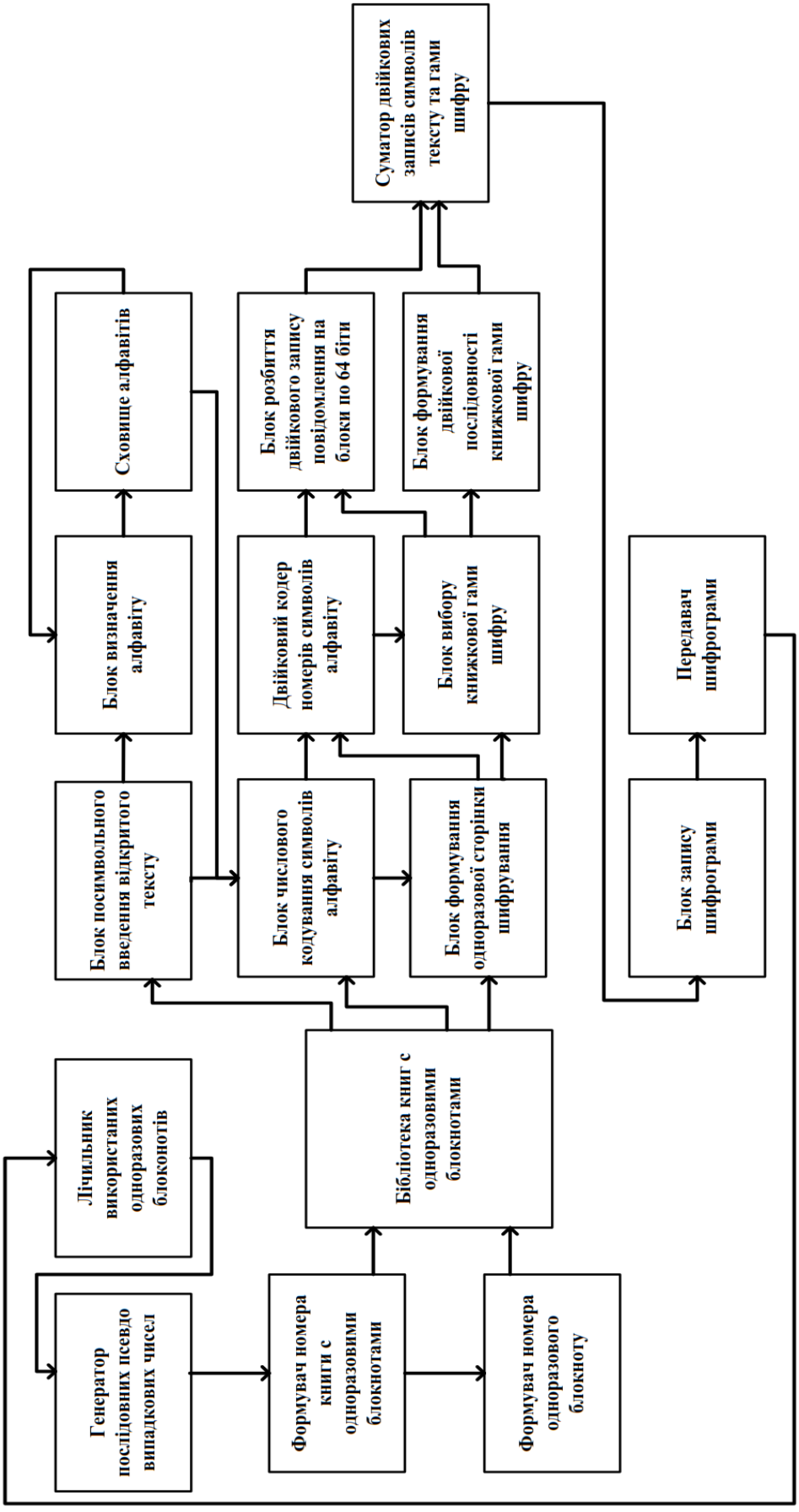

Рисунок 3.2 – Структурна схема пристрою

Усі блоки, що входять до складу структурної схеми, виконують певні функції:

- **Формування ключа**: забезпечує введення ключа для шифрування заданого текстового рядка.
- **Введення команд**: служить для введення команд управління.
- **Вибір рядка**: дозволяє обрати рядок, який містить текст для шифрування.
- **Перевірка символів у рядку**: здійснює виявлення помилок у текстовому рядку.
- **Аналіз**: визначає мову тексту у вибраному рядку.
- **Заміна символів**: коригує помилки, знайдені в рядку.
- **Керування**: генерує сигнали, необхідні для коректного функціонування всіх блоків.
- **Гамування**: відповідає за формування гамми.
- **Пам’ять**: використовується для зберігання інформації.
- **Шифрування**: виконує шифрування вказаного тексту.
- **Шифрограма**: результат у вигляді зашифрованого тексту.

Принцип роботи пристрою для захисту конфіденційної інформації полягає в наступному:

Керування пристроєм здійснюється за допомогою пульта управління, який формує необхідні команди, шифрує їх методом книжкового гамування і передає одержувачу через канал зв’язку. Детальний опис процесу шифрування наведено в розділі, присвяченому алгоритму роботи пристрою.

4 РОЗРОБЛЕННЯ ФУНКЦІОНАЛЬНОЇ ТА ПРИНЦИПОВОЇ ЕЛЕКТРИЧНОЇ СХЕМИ ПРИСТРОЮ

4.1 Вибір елементної бази

Для побудови пристрою захисту інформації необхідно вибрати серію мікросхем, на яких будуть розроблені схеми пристрою.

Аналізуючи умови та функціонал приладу підібрані такі мікросхеми:

- мікроконтролер – КР1816ВЕ51;
- буферний регістр– KR580ИР82;
- логічного елемента 2І-НЕ – 564ЛА10;
- пам’ять постійного зберігання – КР573РФ2.
- пам’ять з довільним зберіганням – КР537РУ10;
- програмований контролер паралельного вводу-виводу–КР580ВВ55;
- перетворювач сигналу – MAX232.

4.2 Мікроконтролер – КР1816ВЕ51

Однокристальний мікроконтролер КР1816ВЕ51 (МК51) виготовлений у корпусі ВІС із використанням високорівневої n-МОП технології, яка забезпечує вищий ступінь інтеграції порівняно з біполярними структурами. Корпус ВІС містить 40 зовнішніх виводів, схема цокольовки та назви виводів наведені на рисунку 4.1. Для роботи мікроконтролера необхідне одне джерело живлення з напругою +5 В.

МК51 взаємодіє із зовнішнім середовищем через чотири програмовані порти вводу-виводу, які відповідають стандарту ТТЛ-схем із трьома станами входу. У корпусі мікроконтролера передбачено два виводи для підключення кварцового резонатора, чотири виводи для сигналів, що керують режимами роботи МК, а також 8 ліній порту 3. Лінії порту 3 можуть бути запрограмовані користувачем для виконання спеціалізованих (альтернативних) функцій обміну даними із зовнішнім середовищем.

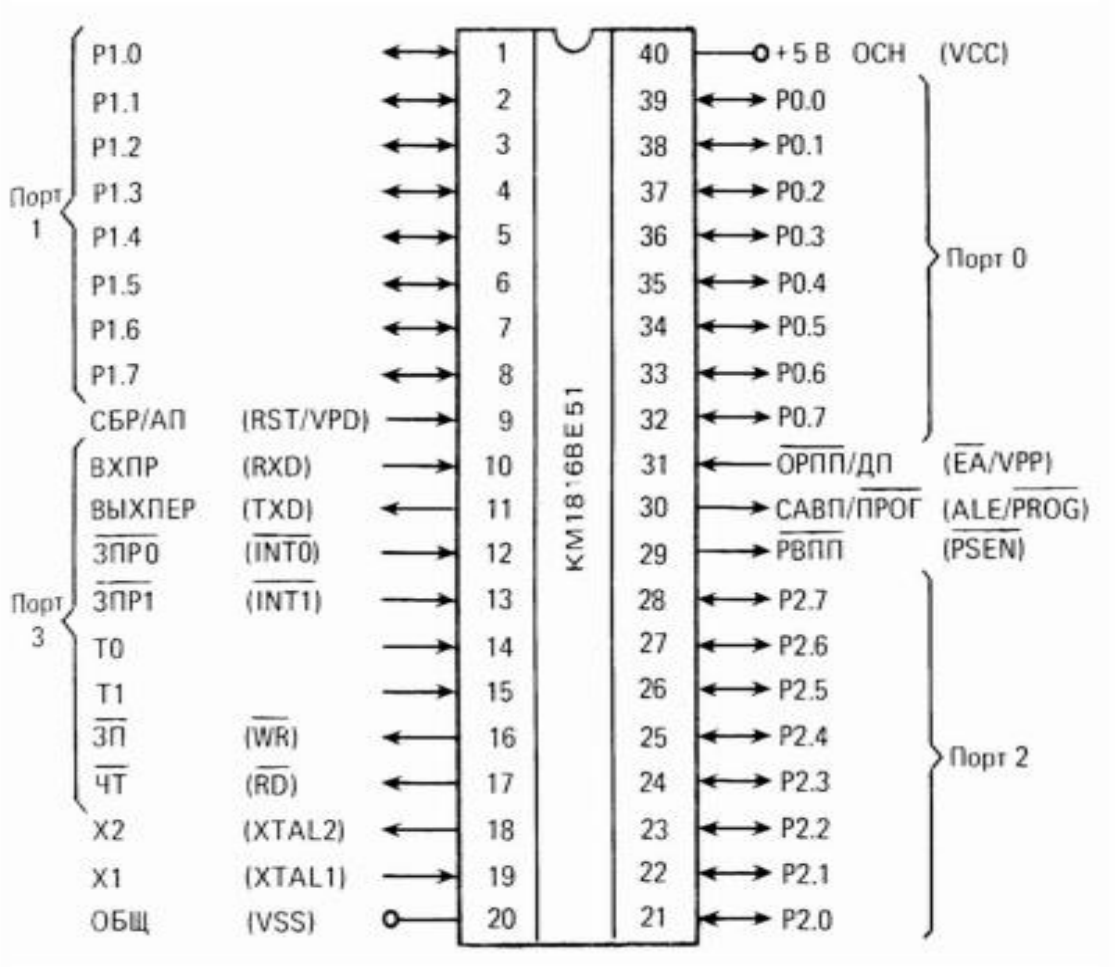

Рисунок 4.1 – Корпус мікросхеми МК51 і найменування виводів

Назва виводів і значення вхідних і вихідних рівнів приводяться нижче:

ТТЛ: Uin ≥ 2,0В Uil ≤ 0,8 В – вхідні рівні;

Uoh ≥ 2,4В Uol ≤ 0,4В – вихідні рівні;

Опис керуючих сигналів і ліній:

- СБР (RST): сигнал скидання;
- ВхПр (RxD): вхід приймача універсального асинхронного приймача-передавача (УАПП);
- ВихПер (TxD): вихід передавача УАПП;
- ЗПр (INT): запит переривання;
- Т0, Т1: таймер/лічильник подій;
- Х1, Х2: входи для кварцового резонатора;
- ЗП (WR): сигнал запису;
- ЧТ (RD): сигнал читання;

- ОРПП (EA): сигнал відключення вбудованої пам’яті програм;
- САВП (ALE): сигнал формування адреси зовнішньої пам’яті;
- ПРОГ (PROG): сигнал програмування резидентної пам’яті програм;
- ДЗПП (PSEN): сигнал дозволу доступу до зовнішньої пам’яті програм;
- ОБЩ (VSS): загальний потенціал (заземлення);
- +5В (VCC): живлення +5 В;
- X1 (XTAL1): вхід для підключення кварцового резонатора або сигналу зовнішньої синхронізації;
- X2 (XTAL2): вхід для підключення другого виводу кварцового резонатора.

Основу мікроконтролера становить внутрішня двонаправлена 8-бітна шина, що зв’язує основні вузли та пристрої. До них належать:

- 8-розрядний центральний процесор (ЦП): містить АЛУ, акумулятор з розширювачем, регістр слова стану, блок спеціальних регістрів функцій, пристрій управління та синхронізації, розміщений на кристалі;
- Пам’ять програм (ПЗП): обсяг 4 Кбайт;
- Пам’ять даних (ОЗП): обсяг 128 байт;
- 32 лінії вводу-виводу: чотири паралельних порти (P0, P1, P2, P3);
- Послідовний порт вводу-виводу;
- Два 16-розрядні таймери/лічильники;
- Дворівнева система переривань із п’ятьма або шістьма джерелами запитів.

Всі ці компоненти формують резидентну частину мікроконтролера, розміщену на кристалі. Додатково можна підключити зовнішню пам’ять програм і даних обсягом до 64 Кбайт кожна.

Особливості інтерфейсу: Для зменшення ширини фізичного інтерфейсу більшість логічних ліній поєднуються. Наприклад, порт P0 виконує роль комбінованої шини адреси та даних під час роботи із зовнішньою пам’яттю, а порт P2 служить для передачі старших адресних бітів. Лінії порту P3 можуть виконувати альтернативні функції керування【4】.

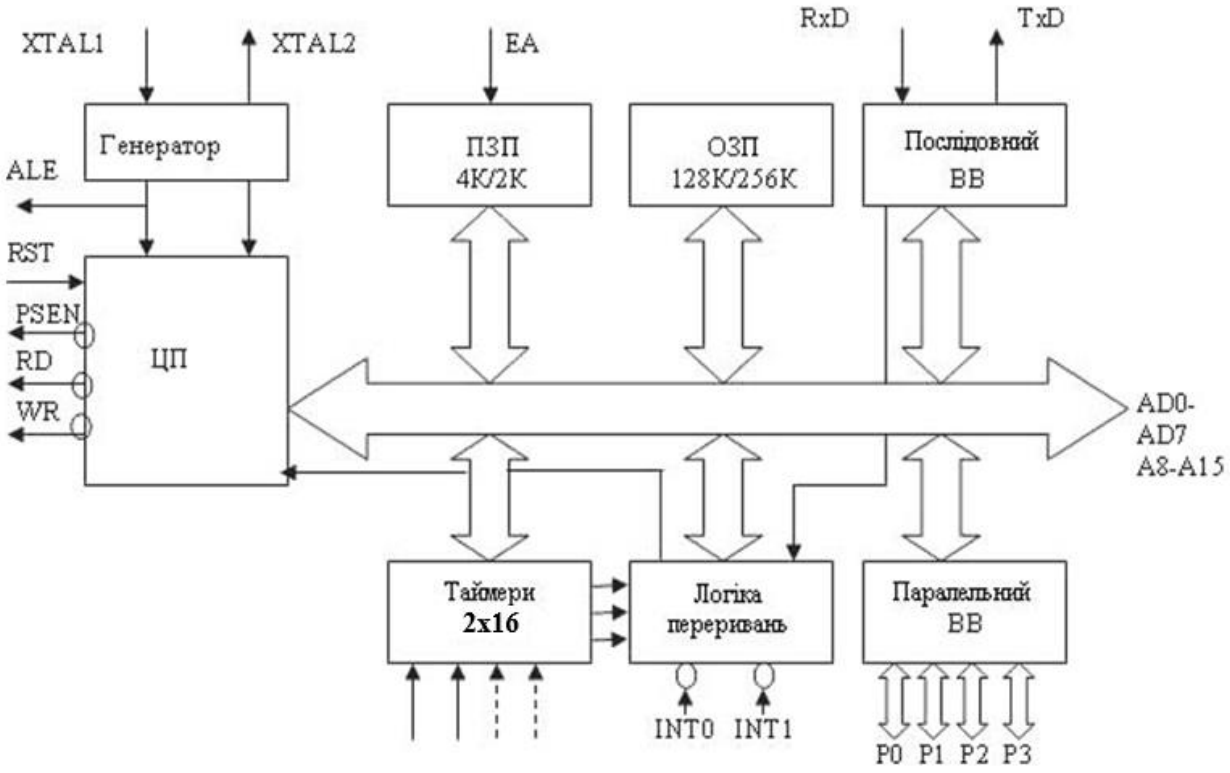

Рисунок 4.2 – Узагальнена структурна схема МК51

В архітектурі МК51 реалізовано принцип незалежності середовищ для зберігання програм і даних, що відповідає концепції Гарвардської архітектури.

Резидентна пам'ять

Пам'ять програм і даних, розташовані на кристалі МК51, фізично та логічно ізольовані одна від одної. Вони використовують різні методи адресації, керуються різними сигналами та виконують окремі функції.

Пам'ять програм

Програма зберігається у постійній пам’яті (ПЗП або СППЗУ), обсяг якої становить 4 Кбайти. Вона призначена для зберігання команд, констант, початкових значень, таблиць перекодування змінних тощо. Доступ до пам’яті програм забезпечується через 16-бітну шину адреси, яка підключена до лічильника команд або до регістра покажчика даних (РПД).

Пам'ять даних

Дані зберігаються у оперативній пам’яті (ОЗП) обсягом 128 байт. Ця пам’ять використовується для зберігання змінних під час виконання програми. Вона адресована 8-бітним адресним простором. У пам’яті даних

розміщені також регістри спеціальних функцій (РСФ), регістр покажчика даних і регістр програмної пам’яті, організація доступу до яких зображена на рис. 5.3. Програмно доступний набір регістрів наведено на рис. 5.4.

Оскільки структура МК51 є акумуляторною з банками перемикання робочих регістрів, основним регістром виступає акумулятор.

Основні регістри

- **Акумулятор (А):** 8-розрядний регістр, який виконує функції основного арифметичного блоку. Він є джерелом операндів і приймачем результатів арифметичних, логічних та деяких інших операцій. За допомогою акумулятора виконуються операції зсуву, перевірки на нуль, роботи з прапором паритету тощо.
- **Регістр В:** Також 8-розрядний, служить розширенням акумулятора А. Використовується для операцій множення та ділення, виступаючи як джерело або приймач операндів. В інших випадках регістр В може виконувати користувацькі функції. При скиданні регістри А і В автоматично встановлюються у нуль.

Контролер МК51 здатний виконувати численні команди без прямого використання акумулятора. Дані можуть переміщуватись між будь-якою коміркою пам’яті даних і регістром, або регістр може бути безпосередньо завантажений операндом.

4.3 Буферний регістр– KR580ИР82

Введення та виведення інформації здійснюється через порти, які представлені у вигляді 8- або 16-розрядних регістрів, оснащених схемами вибору та управління операціями читання і запису. Використання регістра КР580ИР82 для підключення до пристроїв введення та виведення ілюструється на рисунку 4.3.

У разі використання регістра як порту введення, дані від пристрою введення передаються до регістра через лінії DI7-DI0 та записуються під впливом стробувального сигналу STB.

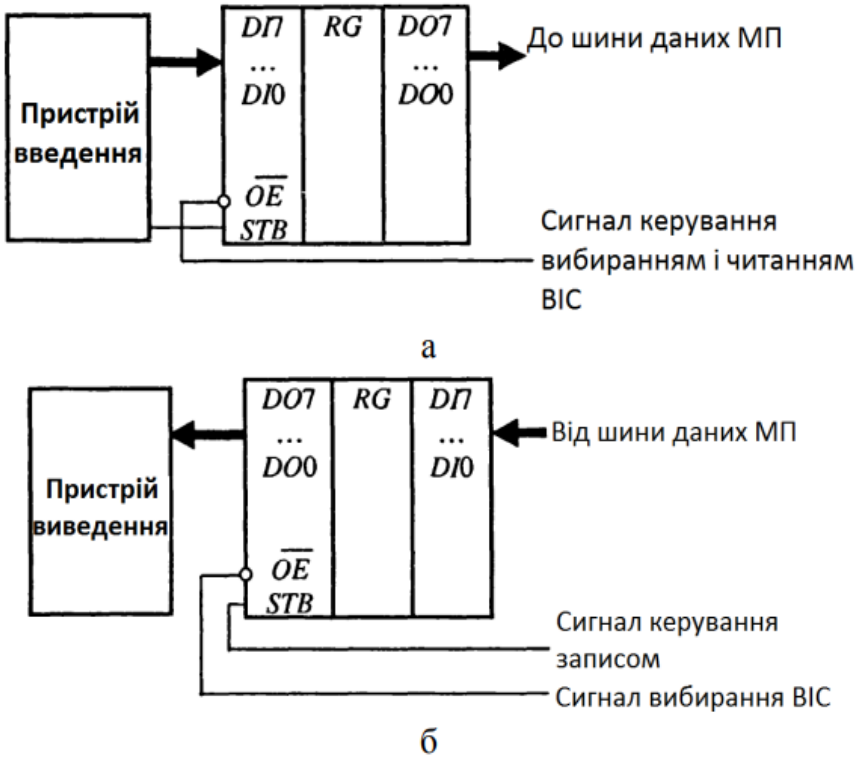

Рисунок 4.3 – Використання регістра КР580ИР82 для з'єднання:
а – з пристроєм введення;
б – з пристроєм виведення

Вихідна інформація порту, представлена на лініях DO7-DO0, передається до мікропроцесорної системи (МПС) по шині даних. Мікропроцесор також формує керуючі сигнали для читання та вибору порту, які подаються на вхід OE.

Якщо регістр використовується як порт виведення, дані від мікропроцесора передаються через шину даних на входи DI7-DI0 порту і супроводжуються керуючими сигналами для запису та вибору ВІС. Вихідна інформація порту, представлена на лініях DO7-DO0, спрямовується до пристрою виведення.

Процес введення або виведення даних може бути реалізований двома способами:

– за допомогою окремого адресного простору для пристроїв введення/виведення (ПВВ);
– шляхом використання спільного адресного простору з пам'яттю,

тобто з відображенням пристроїв введення/виведення в адресний простір пам’яті [6].

Таблиця 4.1 – Призначення виводів KR580ИР82

№ виводу	Позначення	Тип	Призначення
1–8	D0–D7	Вхід	Вхідна шина даних
9	OE	Вхід	Дозвіл виходу
10	Gnd	—	Загальний
11	Stb	Вхід	Строб
12–19	Q7–Q0	Вихід	Вихідна шина даних
20	Vcc	—	Живлення +5 В

Таблиця 4.2 – Основні параметри KR580ИР82

Параметр	Значення
Напруга живлення	+5 В ±10%
Споживаний струм	<160 мА
Вихідна напруга низького рівня	<0,45 В
Вихідна напруга високого рівня	>2,4 В
Максимальний вихідний струм низького рівня (лог. 0)	32 мА
Максимальний вихідний струм високого рівня (лог. 1)	5 мА
Вхідний струм високого рівня	<50 мкА
Вхідний струм низького рівня	<200 мкА
Час затримки розповсюдження	10–30 нс
Вхідний рівень "0"	<0,8 В
Вхідний рівень "1"	>2,0 В
Вхідна ємність (при f = 10 МГц)	<12 пФ
Робочий діапазон температур	-10°C...+70°C
Корпус	2140.20-1 (DIP-20)

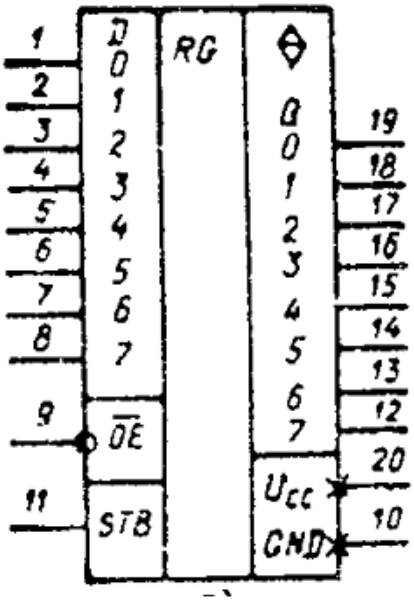

Рисунок 4.4 – Корпус мікросхеми KR580ИР82 і найменування виводів

4.4 Пам’ять постійного зберігання – КР573РФ2

Мікросхема КР573РФ2 є перепрограмовуваним постійним запам’ятовуючим пристроєм (ПЗП), що підтримує стирання даних за допомогою ультрафіолетового випромінювання та має ємність 2048 байт. Умовне графічне зображення цієї мікросхеми подано на рисунку 4.5.

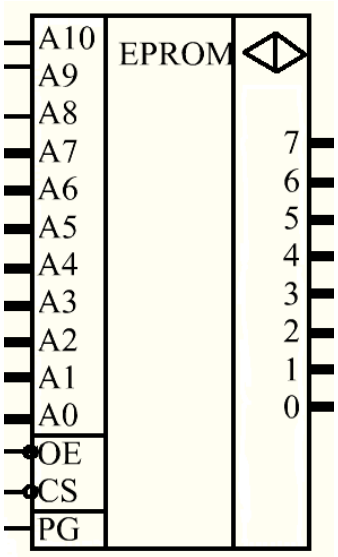

Рисунок 4.5 - КР573РФ2

Дані з пам'яті комірки, адреса якої присутня на адресних входах протягом всього циклу, зчитуються при подачі сигналів нульового рівня на вхід вибору кристала CS і вхід дозволу виходу CEO.

Основні характеристики КР573РФ2:

- Ємність пам'яті: 16 Кбіт (2 Кб × 8).
- Напруга живлення: +5 В ±10%.
- Максимальний споживаний струм: до 160 мА.
- Рівні вихідної напруги:
  - Логічний 0: не більше 0,45 В.
  - Логічний 1: не менше 2,4 В.
- Максимальний вихідний струм:
  - Логічний 0: до 32 мА.
  - Логічний 1: до 5 мА.
- Вхідні рівні напруги:
  - Логічний 0: не більше 0,8 В.
  - Логічний 1: не менше 2,0 В.
- Вхідна ємність (при f=10 МГц): не більше 12 пФ.
- Час затримки розповсюдження сигналу: 10–30 нс.
- Діапазон робочих температур: від -10°С до +70°С.
- Корпус: DIP-24 (тип 210Б.24-5), маса не більше 5 г.

4.5 Пам’ять з довільним зберіганням – КР537РУ10

Мікросхема КР537РУ10 є оперативним запам'ятовуючим пристроєм статичного типу, виготовленим за КМОП технологією.

Умовне графічне зображення мікросхеми наведено на рисунку 4.6.

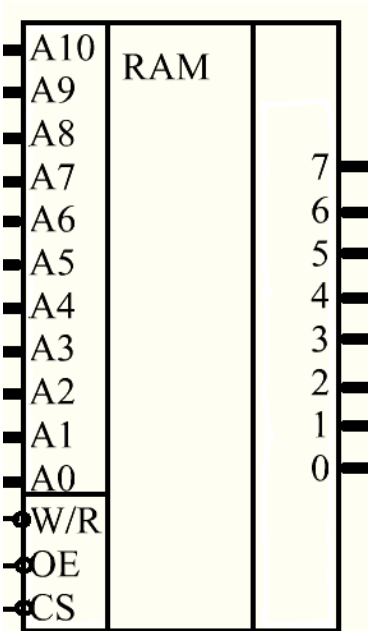

Рисунок 4.6 - КР537РУ10

Таблиця 4.3 – Призначення виводів KR580ИР82

Номер виводу	Позначення	I/O	Призначення
1-8, 19, 22, 23	A0-A10	Вхід	Адресні входи
9-11, 13-17	I/O1-I/O8	Вхід/Вихід	Лінії даних (вхід/вихід)
12	GND	-	Загальний (земля)
18	CS#	Вхід	Вибір кристала
20	OE#	Вхід	Дозвіл виходу
21	WE#	Вхід	Дозвіл запису
24	Vcc	-	Живлення

Інформаційна ємність мікросхеми становить 2048×8 біт. На адресні входи подається повний код адреси комірки пам'яті (на відміну від попереднього випадку, де використовувались окремі адреси рядка і стовпця), який утримується протягом усього циклу.

Вхід **CS** призначений для вибору кристала. Цикл читання або запису в мікросхемі виконується тільки тоді, коли на цьому вході присутній сигнал низького рівня.

Таблиця 4.4 – Основні параметри KR580ИР82

Параметр	Значення
Напруга живлення	5В ±5%
Напруга живлення в режимі зберігання	+2..5В
Струм споживання	<60мА
Струм споживання в режимі зберігання	<400мкА
Вихідна напруга лог.0	<0,4В
Вихідна напруга лог.1	>2,4В
Час вибірки (CS, OE)	<180нс
Час циклу запису	<180нс

Вхід OE використовується для дозволу виходу. При подачі сигналу низького рівня на цей вхід виходи даних мікросхеми переходять із високоімпедансного стану в активний режим, і на них з'являються дані з комірки пам'яті, адреса якої в цей момент подана на адресні входи. Запис даних у мікросхему відбувається при позитивному перепаді сигналу на вході WR/RD за умови, що на вході OE знаходиться сигнал високого рівня, а на вході CS – сигнал низького рівня [7].

4.6 Програмований контролер паралельного вводу-виводу – КР580ВВ55

Програмований контролер паралельного вводу-виводу КР580ВВ55 призначений для введення-виведення паралельної інформації у 8-байтовому форматі, що дозволяє реалізувати більшість відомих протоколів обміну по паралельних каналах. Може використовуватись для з’єднання МП зі стандартним периферійним устаткуванням (дисплеєм, телетайпом, накопичувачем, тощо).

Структурну схему ВІС КР580ВВ55 показано на рисунку 4.7.

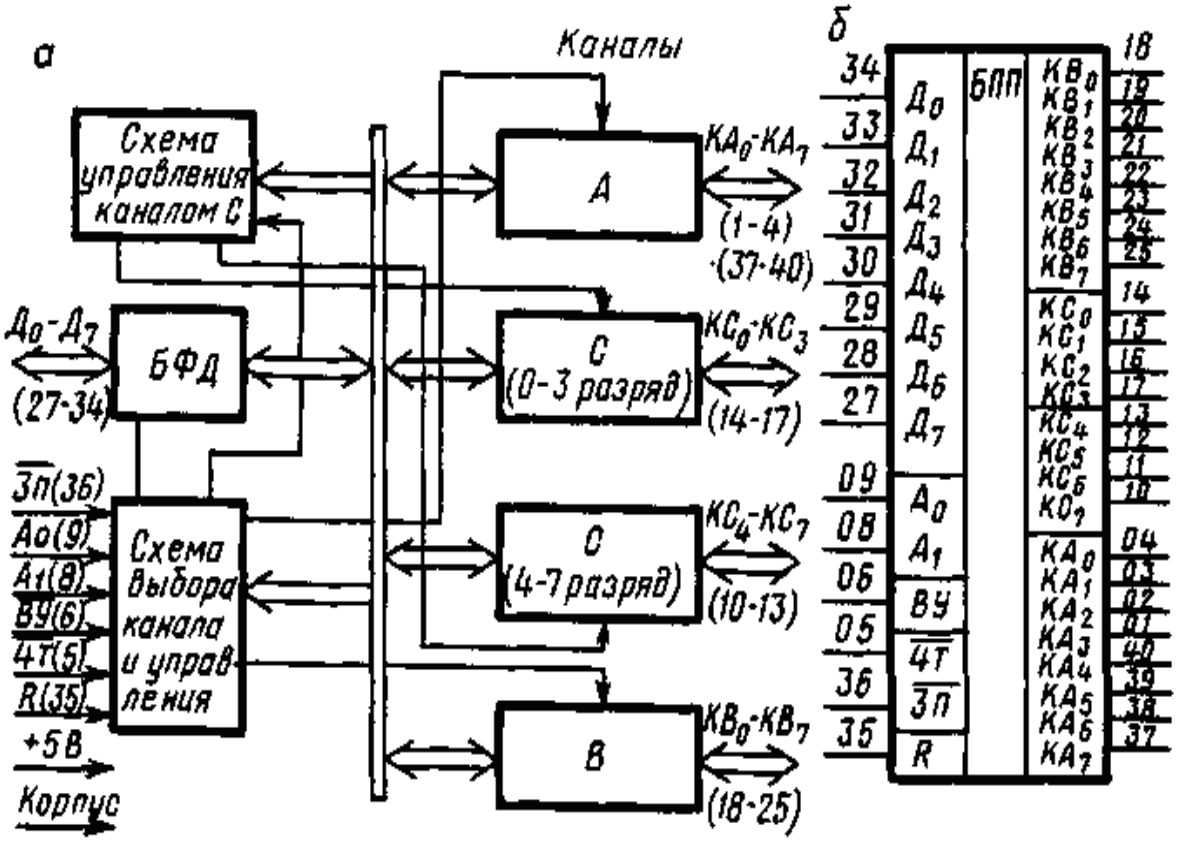

Рисунок 4.7 - Спрощена схема програмованого паралельного інтерфейсу КР580ВВ55 (а) та його умовне позначення (б)

До складу контролера входять такі компоненти:

- **Блок керування читанням/записом** (*Read/Write Control Unit*, **RWCU**), який відповідає за керування внутрішнім та зовнішнім обміном даними та керувальними словами.
- **Двонапрямлений 8-розрядний буфер даних** (*Buffer of Data*, **BD**), що забезпечує з’єднання ліній даних мікросхеми із системною шиною даних.
- **Три 8-розрядні порти введення/виведення** (*Port A, Port B, Port C*) для передачі інформації. Порт **С** поділено на два 4-розрядні сегменти: **С’** (РС7–РС4) та **С”** (РС3–РС0). Порти **А** та **С’** утворюють **групу А**, а порти **В** та **С”** – **групу В**.

Мікросхема також містить блоки керування **групою А** (*Control Unit A*, **CUA**) та **групою В** (*Control Unit B*, **CUB**), які формують керувальні сигнали для відповідних груп.

Блок RWCU містить **регістр керувального слова**, що зберігає керувальні слова, які надходять від мікропроцесора.

Призначення виводів мікросхеми КР580ВВ55:

- **D7-D0** – лінії для передачі даних (вхід/вихід);
- **RD** – сигнал читання: рівень **L** дозволяє зчитування даних із регістра, що адресовано за допомогою ліній **A0, A1**, на шину **D7-D0**;
- **WR** – сигнал запису: рівень **L** дозволяє запис даних із шини **D7-D0** у порт, що адресовано лініями **A0, A1**;
- **A0, A1** – адресні входи для вибору внутрішніх регістрів ППІ;
- **RESET** – сигнал скидання: рівень **H** скидає регістр керувального слова і переводить усі порти в режим введення;
- **CS** – сигнал вибору мікросхеми: рівень **L** з'єднує шину даних **D7-D0** мікросхеми із системною шиною;
- **RA7-RA0** – лінії вводу/виводу **порту А**;
- **RB7-RB0** – лінії вводу/виводу **порту В**;
- **RC7-RC0** – лінії вводу/виводу **порту С**;
- **Vcc** – вивід для підключення напруги живлення **+5 В**;
- **GND** – загальний вивід (0 В).

4.7 Розробка схеми електричної функціональної

На основі структурної схеми розпочинається важливий етап – побудова функціональної схеми. Цей процес визначає взаємодію всіх компонентів та визначає порядок функціонування системи.

Інтегральною мікросхемою, що містить основні функціональні блоки, необхідні для керування пристроями, обробки даних і виконання програм є мікроконтролер.

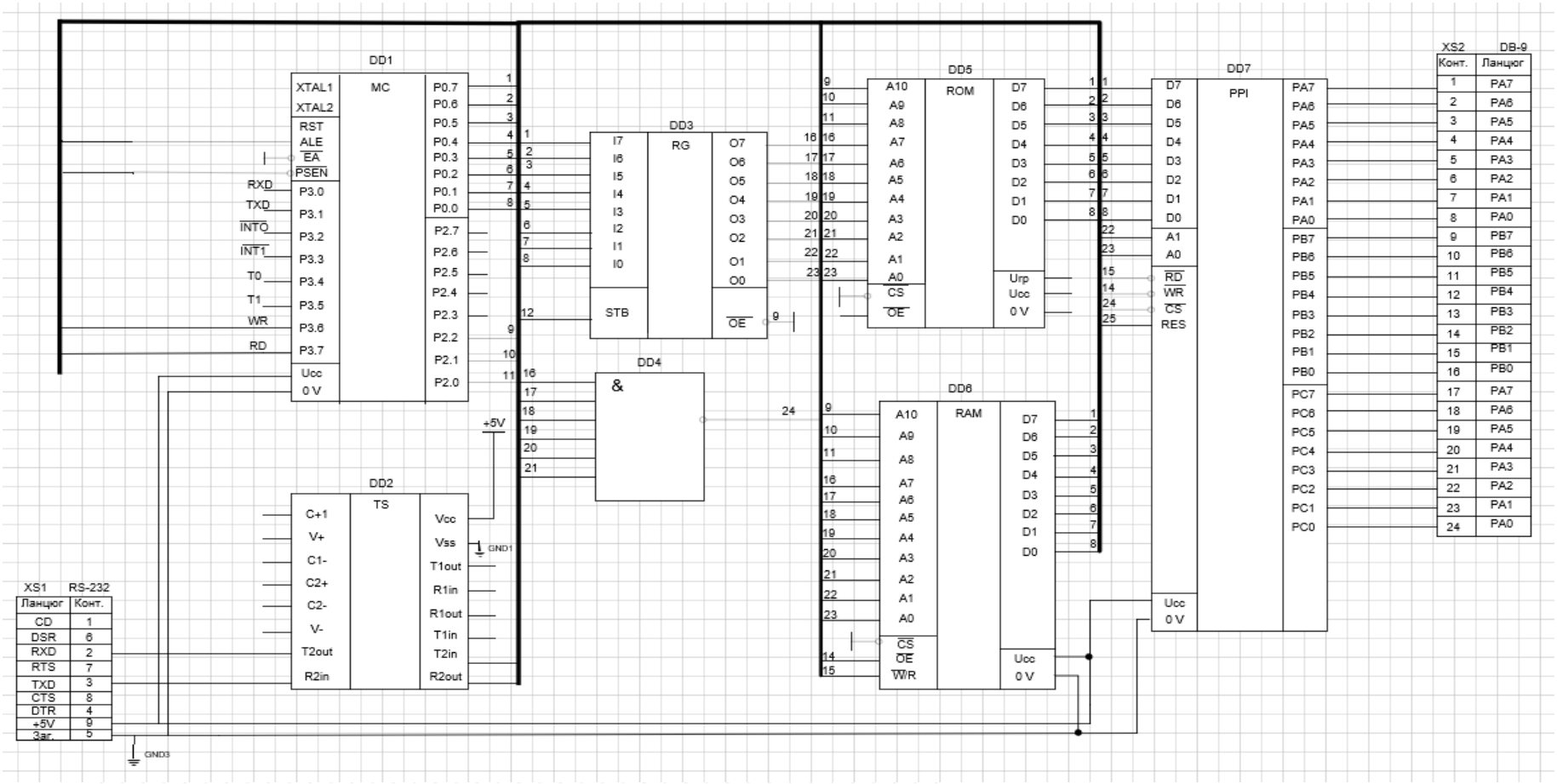

Рисунок 4.7 – Функціональна схема криптографічної системи захисту інформації із застосуванням одноразового блокноту

Буферний регістр у мікросхемах виконує важливі функції, пов'язані з тимчасовим зберіганням даних та забезпеченням передачі інформації між різними блоками системи. Основні функції буферного регістра включають: тимчасове зберігання даних, згладжування швидкості передачі, буферизація вводу/виводу.

Для реалізації логічної операції, побудови логічних блоків, узгодження сигналів и контролю логічного стану використовується логічний елемент 2І-НЕ.

Пам'ять у мікросхемі поділяється на постійну пам’ять (ROM – Read Only Memory) та довільного доступу (RAM – Random Access Memory). Обидва типи пам’яті виконують важливі функції у цифрових пристроях, зокрема в мікропроцесорах, мікроконтролерах та інших мікросхемах.

Для забезпечення обміну даними між мікропроцесором і зовнішніми пристроями через паралельні порти використовується програмований контролер паралельного вводу-виводу. Основна особливість таких контролерів – можливість програмно керувати режимами роботи портів.

Для здійснення перетворення сигналів із аналого-цифрове або цифрового-аналогового перетворення використовується перетворювач сигналу. Такі пристрої відіграють ключову роль у системах управління, зв'язку, вимірювальних приладах та цифрових системах.

4.8 Розробка програмного забезпечення для контролера КР580ВВ55

Контролер КР580ВВ55 є програмованим пристроєм паралельного введення/виведення (ППІ), який дозволяє організовувати обмін даними між мікропроцесором і зовнішніми пристроями. При розробці програмного забезпечення (ПЗ) для КР580ВВ55 необхідно враховувати його архітектуру та функціональні режими роботи.

```
; Ініціалізація КР580ВВ55
; Визначаємо порти A, B та C як вихідні у режимі 0

ORG 0000H          ; Початкова адреса програми
```

```
START:
    MVI A, 80H     ; Завантажити керуюче слово: порти A, B, C як вихідні, режим 0
    OUT 02H        ; Запис керуючого слова у регістр керування (адреса 02H)

; Запис даних у порти
    MVI A, 55H     ; Завантажити дані (наприклад, 55H)
    OUT 00H        ; Запис даних у порт A (адреса порту A - 00H)

    MVI A, AAH     ; Завантажити інші дані (наприклад, AAH)
    OUT 01H        ; Запис даних у порт B (адреса порту B - 01H)

; Зчитування даних із порту
    IN 00H         ; Зчитати дані з порту A (адреса 00H)
    MOV B, A       ; Скопіювати дані в регістр B

; Завершення програми
    HLT            ; Зупинка програми
```

Пояснення програми:

1. Ініціалізація портів:

У регістр керування на адресу 02H записується керуюче слово 80H, що переводить порти A, B, C у вихідний режим (режим 0).

2. Запис даних у порти:

Команда OUT використовується для запису даних у порти A та B.

Порт A має адресу 00H, порт B – 01H.

3. Читання даних із порту:

Команда IN зчитує дані з порту A.

Зчитані дані можна зберегти у регістрі (у прикладі – у регістрі B).

4. Зупинка програми:

Команда HLT завершує виконання програми.

5 ТЕХНІКО - ЕКОНОМІЧНА ЧАСТИНА

5.1 Розрахунок повної собівартості проектованого пристрою

Собівартість продукції, що виготовляється підприємством, являє собою загальну суму витрат на її виробництво та реалізацію, виражену в грошовій формі. Витрати, які безпосередньо пов'язані з виробничим процесом, формують виробничу собівартість. Якщо до них додати витрати, пов'язані зі збутом, утворюється повна собівартість продукції. Розрахунок собівартості конкретного виробу за окремими статтями витрат називається калькуляцією. Такий розрахунок виконується відповідно до «Типового положення щодо планування, обліку та калькулювання собівартості продукції (робіт, послуг) у промисловості».

У процесі виготовлення будь-якої продукції використовуються різноманітні матеріали, комплектуючі, обладнання та інструменти, а також виконується значна кількість технологічних операцій . Для обліку фактичних витрат на виробництво і точного визначення собівартості велике значення має класифікація цих витрат. Розрахунок собівартості конкретного виду продукції базується на поділі витрат за калькуляційними статтями.

У плануванні та обліку собівартості продукції застосовується типове групування за статтями калькуляції :

- основна заробітна плата;
- додаткова заробітна плата;
- відрахування від заробітної плати;
- матеріали та комплектуючі;
- витрати на утримання та експлуатацію обладнання;
- виробничі витрати;
- адміністративні витрати;
- позавиробничі витрати (комерційні витрати).

Систематизація витрат за калькуляційними статтями витрат визначає рівень собівартості виробу та, відповідно, його ринкову ціну. Цей підхід дозволяє визначити, де саме виникають витрати та яке призначення вони мають.

Вихідними даними для складання калькуляції собівартості на проектований пристрій є статті калькуляції, що пов’язані із закупівлею комплектуючих виробів. Також необхідно враховувати вартість напівфабрикатів, які використовуються при виготовленні друкованої плати.

5.2.1 Матеріали та комплектуючі

Матеріали та комплектуючі розглядаються виходячи із відомостей (каталогів, прайс-листів, web-сайтів виробників та постачальників тощо) на матеріали, сировину, комплектуючі, операції з розрахунку на одну одиницю випуску.

Дані за цією статтею витрат наведені у таблиці 5.1.

Таблиця 5.1 – Розрахунок витрат на комплектуючі

№ п/п	Найменуванння	Кількість, од.	Ціна за одиницю, грн.	Сума, грн.
1	2	3	4	5
МІКРОСХЕМИ				
1	*КР1816ВЕ51*	1	150,00	150,00
2	*MAX232*	1	25,00	25,00
3	*КР580ИР82*	1	8,00	8,00
4	*564ЛА10*	1	300,00	300,00
5	*КС573РФ2*	1	40,00	40,00
6	КР537РУ10	1	30,00	30,00
7	*КР580ВВ55*	1	40,00	40,00
КОНДЕНСАТОРИ				
8	*SMD0805Y5V 1 мкФ 5В-10%*	7	90,00	630,00
9	СВВ60 10 мкФ	6	60,00	360,00
10	X7R 0.1 мкФ	7	30,00	210,20

Продовження таблиці 5.1

1	2	3	4	5
РЕЗИСТОРИ				
11	*С2-29В-0.125-220 ОМ+10%*	1	2,00	2,00
Вилки				
12	RS-232	1	200,00	200,00
13	DB-9	1	6,00	6,00
Кварцовий резонатор				
14	21М15В	1	180,00	180,00
Разом, К				2181,00

Загальна вартість усіх комплектуючих складає 2181,00 грн.

Розрахунок витрат за матеріали наведений у таблиці 5.2.

Таблиця 5.2 – Розрахунок витрат за матеріали

Матеріал	Одиниця вимірювання	Норма витрат	Ціна за од, грн.	Ціна, грн.
1	2	3	4	5
Провід монтажний	м	0,4	3.00	1,2
Стеклотекстолит	$м^2$	0,3	60	18
Каніфоль	кг	0,2	1008	201,6
Флюс	кг	0,05	800	40
Припой	кг	0,5	300	150
Лак	кг	0,1	150	15
Речовина для корпусу	кг	0,5	200	100
Разом, М				525,8

З урахуванням транспортно-заготівельних витрат (кт-з = 5 ÷ 15%) вартість комплектуючих та матеріалів становитиме:

$$КМ = (К + М) \cdot (100 + k_{т-з}) / 100. \quad (5.1)$$

КМ = (2181,00+ 525,8) · (100 + 10) / 100 = 2977.48 грн.

5.1.2 Витрати на основну заробітну плату:

$$Зо = \sum_{i=1}^{n} Тг_i \cdot Нч_i \ . \qquad (5.2)$$

де $Тг_i$ – годинна тарифна ставка окремого спеціаліста (інженера-електронника, лаборанта тощо), який задіяний у виробництві пристрою (установки), грн/год;

$Нч_i$ – витрачений час робітником на виробництво та налагодження пристрою (установки), год, $Нч_i$ =4 год.;

n – кількість працівників, задіяних у виробництві пристрою (установки), n = 4.

Годинна тарифна ставка розраховується, виходячи із величини місячного окладу спеціаліста:

$$Тг_i = \frac{Тм_i}{Вф_i \cdot 8} \ . \qquad (5.3)$$

де $Тм_i$ – місячний оклад (ставка) спеціаліста, грн;

$Вф_i$ – фактично відпрацьований час за розрахунковий період (місяць), днів (змін);

8 – кількість відпрацьованих годин за зміну.

$$Тг_i = \frac{Тм_i}{Вф_i \cdot 8} = \frac{14000}{22 \cdot 8} = 79{,}54 \text{ грн.}$$

$$З_о = \sum_{i=1}^{4} 79{,}54 \cdot 4 = 4 \cdot 79{,}54 \cdot 4 = 1272.72 \text{ грн}$$

5.1.3 Витрати на додаткову заробітну плату

Додаткова заробітна плата (10 – 30% від $З_о$):

$$Зд = Зо \cdot \frac{Кд}{100}. \quad (5.4)$$

де $К_д$ – відсоток додаткової заробітної плати, $К_д$ = 10%.

$$З_д = 1272.72 * (10\% / 100\%) = 127.27 \text{ (грн.)}.$$

5.1.4 Відрахування на соціальні виплати

Відрахування на соціальні виплати містять відрахування від сукупної основної та додаткової заробітної плати відповідно до встановлених тарифів. Ці утримання включають:

- обов’язкові внески до державної пенсійної системи;
- страхові внески у разі нещасних випадків;
- обов’язкові внески до державного соціального страхування від безробіття;
- витрати, що пов’язані з тимчасовою втратою працездатності;
- витрати, що пов’язані з народженням дитини та похованням.

Нарахування на заробітну плату – єдиний соціальний внесок у розмірі 22%.

$$В_{св} = (З_о + З_д) * 22/100. \quad (5.5)$$

$$В_{св} = (1272.72 + 127.27) * 22/100 = 308 \text{ (грн.)}.$$

5.1.5 Видатки на утримання та експлуатацію встаткування

Витрати на утримання та експлуатацію встаткування (ВУЕ) перебувають на балансі підприємства міста і розраховуються за такою формулою:

ВУЕ = основна заробітна плата * відсоток ВУЕ (приймають відсоток ВУЕ рівним 120 ÷ 150%).

$$ВУЕ = 1272.72 * 1,2 = 1526.4 \text{ грн.}$$

5.1.6 Загальновиробничі витрати

Загальновиробничі витрати містять у собі різноманітні витрати, пов'язані з керуванням підрозділом. Це охоплює витрати на управлінські заходи в межах підрозділу, витрати на відрядження співробітників, а також амортизаційні відрахування вид вартості основних фондів загальноцехового призначення та інші подібні витрати.

Загальновиробничі витрати ($В_{зв}$) визначаються у розмірі 130-250% від основної заробітної плати.

$$В_{зв} = З_{о} * \% В_{зв} = 1272.72 * 1,5 = 1909.08 \text{ (грн.).} \quad (5.6)$$

5.1.7 Виробнича собівартість

Виробнича собівартість розраховується за формулою:

$$В_{с} = З_{о} + З_{д} + В_{св} + КМ + ВУЕ + В_{зв} =$$
$$= 1272.72 + 127.27 + 308 + 2977.48 + 1526.4 + 1909.08 = 8120.95 \text{ (грн.).} \quad (5.7)$$

5.1.8 Адміністративні витрати

Адміністративні витрати можуть включати різноманітні компоненти, зокрема:

- витрати, що пов’язані з ефективним управлінням діяльністю підприємства, такі як витрати на управління плануванням, координацією та контролем за процесами;
- витрати на організацію службових відряджень для адміністративного персоналу підприємства;
- витрати, що пов’язані з утриманням пожежної та сторожової охорони на об’єктах підприємства;
- витрати на організацію навчання та перепідготовки кадрів для забезпечення їхньої кваліфікації;
- витрати на забезпечення транспортування працівників до місця роботи та назад;
- витрати на оплату відсотків за фінансові та комерційні кредити;
- витрати, які пов’язані з користуванням матеріальними цінностями, що знаходяться в оренді або у лізингу;
- витрати на оплату послуг комерційних банків та інших кредитно-фінансових установ;
- податки та інші відрахування.

Адміністративні витрати ($В_а$) визначаються у розмірі 140 - 200% від основної заробітної плати.

$$В_а = З_о * \% \; В_а = 1272.72 \; * 1,6 = 2036.352 \text{ (грн.).} \quad (5.8)$$

5.1.9 Витрати на збут

Витрати на збут ($В_з$) охоплюють різні складові, включаючи витрати на рекламу та передреалізаційну підготовку пристрою. Загалом, орієнтовно ці витрати визначаються у розмірі 5-10% від виробничої собівартості. Це важливий аспект ефективного підтримання ринкової активності та успішного впровадження пристрою на ринок.

Витрати на рекламу включають ресурси, витрачені на рекламні кампанії, створення промо-матеріалів, організацію рекламних заходи та інші заходи, спрямовані на просування продукту на ринку.

Ці витрати важливі для створення підтримки на ринку.

$$B_з = B_с * (5 - 10)\% = 8120.95 * 0,05 = 406 \text{ (грн.)}. \quad (5.9)$$

5.1.10 Повна собівартість пристрою

Повна собівартість пристрою (С) розраховується за формулою:

$$C = B_с + B_а + B_з. \quad (5.10)$$

$$C = 8120.95 + 2036.352 + 406 = 10563.3 \text{ (грн.)}.$$

Калькуляцію собівартості виробу зведено в таблицю 7.3.

Таблиця 5.3 – Калькуляція собівартості пристрою

№	Найменування статей калькуляції	Значення, грн.
1.	Основна заробітна плата	1526.4
2.	Додаткова заробітна плата	127.27
3.	Відрахування на соціальні виплати	308
4.	Видатки на утримання та експлуатацію встаткування	1526.4
5.	Загальновиробничі витрати	1909.08
6.	Матеріали та комплектуючі	2977.48
Виробнича собівартість		8120.95
7.	Адміністративні витрати	2036.352
8.	Витрати на збут	406
Повна собівартість пристрою		10563.3

5.2 Розрахунок ціни пристрою

5.2.1 Розрахунок оптової ціни пристрою

В ринковій економіці застосовують різні методи ціноутворення: собівартість плюс прибуток, забезпечення фіксованого обсягу прибутку, залежно від рівня попиту.

Розрахунок оптової ціни пристрою проведемо за схемою «собівартість плюс прибуток»:

$$Ц_{опт} = С + П, \quad (5.11)$$

де С – собівартість пристрою;

П – величина прибутку.

Прибуток визначається виходячи з нормативу рентабельності виробництва продукції:

$$R = (П / С) * 100\%, \quad (5.12)$$

де R - рентабельність продукції (продукту), що приймається у розмірі до 35%. R = 10%.

Тоді оптова ціна:

$$Ц_{опт} = С + (R * С / 100) = 10563.3 + 0,1 * 10563.3 = 11619.63 \text{ (грн.)}. \quad (5.13)$$

7.2.2 Розрахунок роздрібної ціни пристрою

Визначимо роздрібну ціну розробленого пристрою:

$$Ц_{розн} = Ц_{опт} * 1,2 = 11619.63 * 1,2 = 13943.55 \text{ (грн.)}, \quad (5.14)$$

де 20% ПДВ.

Методика визначення ціни, описана вище, має як переваги, так і недоліки. До позитивних аспектів можна віднести її простоту та прозорість, особливо в частині розрахунків, які передбачають покриття витрат на виробництво і забезпечення прибутковості створення та реалізації продукту.

Втім, ця методика має певні обмеження, зокрема недостатнє врахування ринкових факторів, таких як попит. Це може призводити до невідповідності встановленої ціни реальним умовам ринку.

Слабким місцем підходу є також ігнорування таких важливих чинників, як конкуренція, рентабельність продукції з точки зору державного регулювання та інших ринкових впливів. Таким чином, застосування цього методу доцільне лише в певних умовах, наприклад, за монопольної ситуації, обмеженої рентабельності, виконання одноразових замовлень чи виготовлення унікальних виробів.

Для встановлення більш обґрунтованої ціни, яка відповідала б сучасним ринковим умовам, необхідно провести додаткові маркетингові дослідження. Вони мають враховувати вплив конкуренції, рівень попиту та інші значущі фактори.

ВИСНОВКИ

Метою кваліфікаційної роботи магістра було створення пристрою для захисту інформації, що є особливо актуальним у сучасному світі через зростання загроз витоку конфіденційних даних.

Причиною для розробки стало забезпечення надійного захисту інформації, оскільки існуючі методи не завжди відповідають сучасним вимогам безпеки. Для досягнення цієї мети було проведено аналітичне дослідження, яке дозволило оцінити переваги та недоліки різних методів шифрування інформації. На основі цього аналізу було обрано метод книжкового гамування, який забезпечує високу криптографічну стійкість та надійність захисту даних.

В рамках проєкту було розроблено:

1. Алгоритм функціонування пристрою – що визначає принципи роботи системи захисту на основі обраного методу.
2. Структурну схему пристрою – яка дозволяє чітко розмежувати функціональні блоки системи та їх взаємодію.

Для побудови електричної принципової схеми оптимальним рішенням став вибір мікроконтролера КР1816ВЕ51.

Таким чином, розроблений пристрій на основі методу книжкового гамування та мікроконтролера КР1816ВЕ51 забезпечує надійний захист інформації. Проєкт поєднує у собі високу ефективність, доступність та економічну доцільність, що робить його перспективним для подальшого впровадження у сфері інформаційної безпеки.

СПИСОК ЛІТЕРАЛУРИ

1. Маліновська, О. О. Вимоги до криптографічної системи захисту інформації / О. О. Маліновська, О. І. Зінченко ; наук. кер. Я. Ю. Усов // Новітні технології у науковій діяльності і навчальному процесі : матеріали тез доп. Всеукр. наук.- практ. конф. студентів, аспірантів та молодих учених (м. Чернігів, 10 -11 квітня 2019 р.). - Чернігів : ЧНТУ, 2019. – С. 113-116.
2. Modern usage of “old” one-time pad/Mariusz Borowski and Marek Lesniewicz
3. Jorgen Veisdal, Shannon Ciphers and Perfect Security/Cantor’s Paradise/2020
4. Касянчук М. М., Якименко І. З., Свистун М. Ю. Фінанси : навч. посіб. Тернопіль – 2020 – 7с.
5. Грищук Ю.С. Мікропроцесорні пристрої: Навчальний посібник. – Харків: НТУ “ХПІ”, 2007.– 280с.
6. Огородник К. В., Книш Б. П.. Мікропроцесорна техніка : навчальний посібник – Вінниця : ВНТУ, 2018. – 106 с.
7. Принципова схема. URL:https://studfile.net/preview/10041830/page:2/ (дата звернення: 24.11.2019).
8. Вибір додаткових елементів схеми URL:https://studfile.net/preview/7052480/page:5/ дата звернення: 31.10.2018).
9. Бирин О.О. Захист інформації на базі методу книжкового гамування в інфокомунікаційних системах / Бережна О.В., Борисенко О.А., Горішняк А.О., Савченко Д.С.,// Фізика, електроніка, електротехніка (ФЕЕ-2022). Матеріали та програма науково-технічної конференції. – Суми: СумДу, 2023. – С.73.

Додаток А

Захист інформації на базі методу книжкового гамування в інфокомунікаційних системах

Борисенко О.А., *професор*; Бережна О.В., *доцент*;
Горішняк А.О., *аспірант*; Бирин О.О., *студент гр. ТК-91*;
Савченко Д.С., *студент гр. ТК-91*
Сумський державний університет, м. Суми, Україна

Впровадження сучасних інфокомунікаційних систем вимагає посилення вимог до безпеки інформації, що надає особливої актуальності пошуку високопродуктивних алгоритмів захисту інформації, що передається, з необхідною криптографічною стійкістю.

Аналіз методів захисту інформації показав, що при використанні асиметричних шифрів відсутня необхідність пересилання секретних ключів, але реалізація таких алгоритмів потребує виконання складних обчислень і, відповідно, вимагає більше часу для шифрування в порівнянні з симетричними шифрами. Тому доцільно розглянути можливість використання алгоритмів симетричного шифрування, які характеризуються швидким шифруванням з високою криптостійкістю.

За результатами дослідження пропонується використання методу гамування вхідних повідомлень, який забезпечує найбільшу криптостійкість за умови використання гами довжиною не менше ніж довжина вхідного повідомлення. Різновидом такого методу шифрування є метод книжкового гамування, який дозволяє використовувати в якості гами сторінки шифрувального блокноту. Принцип шифрування полягає у заміні символів вхідного повідомлення і символів гами цифровими еквівалентами, які потім підсумовуються за модулем N, де N – кількість символів у алфавіті, що застосовується. Неможливість проведення частотного аналізу зашифрованого таким методом повідомлення значно підвищує стійкість даного шифру до несанкціонованого розшифрування. Складність передачі гами шифру отримувачу зашифрованих повідомлень пропонується подолати шляхом формування множини шифрувальних блокнотів (можливо із застосуванням відкритих джерел) і алгоритму вибору сторінок блокноту для здійснення операцій шифрування/розшифрування.

Запропонований метод книжкового гамування є більш ефективним при апаратній реалізації, що дозволить в інфокомунікаційних системах забезпечити швидке шифрування з високим рівнем криптостійкості.