

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
імені ІГОРЯ СІКОРСЬКОГО”**

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

**До захисту допущено:**

**В.о. завідувача кафедри**

Михайло НОВОТАРСЬКИЙ

(підпис)

“ ” \_\_\_\_\_ 2025 р.

**Дипломний проєкт**

**на здобуття ступеня бакалавра**

**за освітньо-професійною програмою “ Комп’ютерні системи та мережі ”**

**спеціальності 123 “ Комп’ютерна інженерія ”**

на тему: Метод та програмні засоби криптографічно-строгої ідентифікації на базі генераторів псевдовипадкових послідовностей

Виконала : студентка 4 курсу, групи ІО-11  
(шифр групи)

Чернишевич Мирослава Станіславівна

(прізвище, ім’я, по батькові)

(підпис)

Керівник доцент, к.т.н. Марковський О.П.

(посада, науковий ступінь, вчене звання, прізвище та ініціали)

(підпис)

Консультант (нормоконтроль) асистент Міщенко Л.Д.

(назва розділу)

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

(підпис)

Рецензент декан ФПМ, д.т.н., проф. Дичка І. А.

(посада, науковий ступінь, вчене звання, прізвище та ініціали)

(підпис)

Засвідчую, що у цьому дипломному  
проєкті немає запозичень з праць інших  
авторів без відповідних посилань.

Студент \_\_\_\_\_

(підпис)

Київ – 2025 р.

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
імені ІГОРЯ СІКОРСЬКОГО”

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

Рівень вищої освіти – перший (бакалавр)

Освітньо-професійна програма

“Комп’ютерні системи та мережі”

спеціальності 123 “Комп’ютерна інженерія”

**ЗАТВЕРДЖУЮ**

**В.о. завідувача кафедри**

\_\_\_\_\_ Михайло НОВОТАРСЬКИЙ

(підпис)

“\_\_\_” \_\_\_\_\_ 2025 р.

**ЗАВДАННЯ**

на бакалаврський дипломний проєкт студентки

Чернишевич Мирослави Станіславівни

1. Тема проєкту Метод та програмні засоби криптографічно-строкої ідентифікації на базі генераторів псевдовипадкових послідовностей

керівник проєкту: Марковський Олександр Петрович, к.т.н, доцент,

(прізвище, ім’я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від «23» травня 2025 року № 1705-с

2. Термін здачі студентом закінченого проєкту «02» червня 2025 р.

3. Вихідні дані до проєкту див. технічне завдання.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які розробляються)

Огляд існуючих засобів ідентифікації учасників віддаленої інформаційної взаємодії. Розробка методу криптографічно строгої

ідентифікації з використання генераторів ПВДП. Розробка програмних засобів моделювання роботи генератору ПВДП.

5. Перелік графічного матеріалу (з точним позначенням обов'язкових креслень) блок-схема алгоритму шифрування, структурна схема методу шифрування, структурна схема електрична функціональна.

6. Консультанта проєкту, з вказівкою розділів проєкту, які до них вносяться

| Розділ        | Консультант  | Підпис, дата   |                  |
|---------------|--------------|----------------|------------------|
|               |              | Завдання видав | Завдання прийняв |
| Нормоконтроль | Міщенко Л.Д. |                |                  |
|               |              |                |                  |

7. Дата видачі завдання «13» січня 2025 р.

#### Календарний план

| № п/п | Найменування етапів дипломного проєкту                     | Терміни виконання етапів проєкту | Примітки |
|-------|------------------------------------------------------------|----------------------------------|----------|
| 1.    | <i>Затвердження теми проєкту</i>                           | 15.04.2025-17.04.2025            |          |
| 2.    | <i>Вивчення та аналіз завдання</i>                         | 18.04.2025-24.04.2025            |          |
| 3.    | <i>Розробка архітектури та загальної структури системи</i> | 25.04.2025-02.05.2025            |          |
| 4.    | <i>Розробка структур окремих підсистем</i>                 | 05.05.2025-13.05.2025            |          |
| 5.    | <i>Програмна реалізація системи</i>                        | 14.05.2025-23.05.2025            |          |
| 6.    | <i>Оформлення пояснювальної записки</i>                    | 30.05.2025-04.06.2025            |          |
| 7.    | <i>Захист програмного продукту</i>                         | 06.06.2025                       |          |
| 8.    | <i>Передзахист</i>                                         | 09.06.2025                       |          |
| 9.    | <i>Захист</i>                                              | 16.06.2025                       |          |

Студентка-дипломник \_\_\_\_\_ Мирослава ЧЕРНИШЕВИЧ  
(підпис)

Керівник проєкту \_\_\_\_\_ Олександр МАРКОВСЬКИЙ  
(підпис)

## **АНОТАЦІЯ**

Досліджено та реалізовано систему віддаленого надання інформаційних, програмних та обчислювальних ресурсів віддаленим користувачам з розмежуванням та контролем прав доступу. Було підвищено ефективність системи ідентифікації віддалених користувачів за рахунок прискорення його обчислювальної реалізації при забезпечення високого рівня захисту прав доступу до ресурсів шляхом застосування криптографічно-строкої технології, основаної на концепції “нульових знань”. Розробка призначена для підвищення швидкості процесу ідентифікації віддаленого абонента.

## **ANNOTATION**

The system of remote provision of information, software and computing resources for remote users with differentiation and control of access rights. The efficiency of the system of identification of remote users was increased by accelerating its computational implementation while ensuring a high level of protection of access rights to resources through the use of cryptographically rigorous technology based on the concept of "zero knowledge". The development is designed to increase the speed of the process of identifying a remote subscriber.

| справки  | Формат | Значення           |      |      | Найменування                                                                                                                                   | Кіл.<br>листів                                    | №<br>екземпля | Додаток |
|----------|--------|--------------------|------|------|------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------|---------------|---------|
|          |        |                    |      |      | Документація загальна                                                                                                                          |                                                   |               |         |
|          |        |                    |      |      | Знову розроблена                                                                                                                               |                                                   |               |         |
|          |        |                    |      |      |                                                                                                                                                |                                                   |               |         |
|          | A4     | ІАЛЦ.467200.002 ТЗ |      |      | Метод та програмні засоби                                                                                                                      | 4                                                 |               |         |
|          |        |                    |      |      | криптографічно-строгої                                                                                                                         |                                                   |               |         |
|          |        |                    |      |      | ідентифікації на базі генераторів                                                                                                              |                                                   |               |         |
|          |        |                    |      |      | псевдовипадкових послідовностей                                                                                                                |                                                   |               |         |
|          |        |                    |      |      | Технічне завдання                                                                                                                              |                                                   |               |         |
|          | A4     | ІАЛЦ.467200.003 ПЗ |      |      | Метод та програмні засоби                                                                                                                      | 62                                                |               |         |
|          |        |                    |      |      | криптографічно-строгої                                                                                                                         |                                                   |               |         |
|          |        |                    |      |      | ідентифікації на базі генераторів                                                                                                              |                                                   |               |         |
|          |        |                    |      |      | псевдовипадкових послідовностей                                                                                                                |                                                   |               |         |
|          |        |                    |      |      | Пояснювальна записка                                                                                                                           |                                                   |               |         |
|          | A4     | ІАЛЦ.467200.004 Е1 |      |      | Метод та програмні засоби                                                                                                                      | 1                                                 |               |         |
|          |        |                    |      |      | криптографічно-строгої                                                                                                                         |                                                   |               |         |
|          |        |                    |      |      | ідентифікації на базі генераторів                                                                                                              |                                                   |               |         |
|          |        |                    |      |      | псевдовипадкових послідовностей                                                                                                                |                                                   |               |         |
|          |        |                    |      |      | Блок-схема алгоритму                                                                                                                           |                                                   |               |         |
|          |        |                    |      |      | шифрування                                                                                                                                     |                                                   |               |         |
|          | A4     | ІАЛЦ.467200.005 Д2 |      |      | Метод та програмні засоби                                                                                                                      | 1                                                 |               |         |
|          |        |                    |      |      | криптографічно-строгої                                                                                                                         |                                                   |               |         |
|          |        |                    |      |      | ідентифікації на базі генераторів                                                                                                              |                                                   |               |         |
|          |        |                    |      |      | псевдовипадкових послідовностей                                                                                                                |                                                   |               |         |
|          |        |                    |      |      | Схема електрична функціональна                                                                                                                 |                                                   |               |         |
|          |        |                    |      |      |                                                                                                                                                |                                                   |               |         |
|          | A4     | ІАЛЦ.467200.006 Д3 |      |      | Метод та програмні засоби                                                                                                                      | 1                                                 |               |         |
|          |        |                    |      |      | криптографічно-строгої                                                                                                                         |                                                   |               |         |
|          |        |                    |      |      | ідентифікації на базі генераторів                                                                                                              |                                                   |               |         |
|          |        |                    |      |      | псевдовипадкових послідовностей                                                                                                                |                                                   |               |         |
|          |        |                    |      |      | Структурна схема                                                                                                                               |                                                   |               |         |
|          |        |                    |      |      |                                                                                                                                                |                                                   |               |         |
|          | A4     | ІАЛЦ.467200.007 Д4 |      |      | Метод та програмні засоби                                                                                                                      | 10                                                |               |         |
|          |        |                    |      |      | криптографічно-строгої                                                                                                                         |                                                   |               |         |
|          |        |                    |      |      | ідентифікації на базі генераторів                                                                                                              |                                                   |               |         |
|          |        |                    |      |      | псевдовипадкових послідовностей                                                                                                                |                                                   |               |         |
|          |        |                    |      |      | Текст програмного коду                                                                                                                         |                                                   |               |         |
|          |        |                    |      |      |                                                                                                                                                |                                                   |               |         |
|          |        |                    |      |      | <b>ІАЛЦ.467200.001 ОА</b>                                                                                                                      |                                                   |               |         |
|          |        |                    |      |      |                                                                                                                                                |                                                   |               |         |
| Зм       | Лист   | № докум.           | Підп | Дата |                                                                                                                                                |                                                   |               |         |
|          |        |                    |      |      |                                                                                                                                                |                                                   |               |         |
| Розроб   |        | Чернишев М.С.      |      |      | Метод та програмні засоби<br>криптографічно-строгої<br>ідентифікації на базі<br>генераторів псевдовипадкових<br>послідовностей<br>Опис альбому | Літ.                                              | Арку<br>ш     | Аркушів |
| Перев    |        | Марковский О.П.    |      |      |                                                                                                                                                |                                                   | 1             | 1       |
| Н.контр. |        | Міщенко Л.Д.       |      |      |                                                                                                                                                | <b>КПІ ім. Ігоря<br/>Сікорського, ФІОТ, ІО-11</b> |               |         |

**ТЕХНІЧНЕ ЗАВДАННЯ**  
**ДО ДИПЛОМНОГО ПРОЄКТУ**

на тему: «Метод та програмні засоби криптографічно-строкої  
ідентифікації на базі генераторів псевдовипадкових послідовностей»

Київ – 2025

# ЗМІСТ

|                                            |   |
|--------------------------------------------|---|
| НАЙМЕНУВАННЯ ТА ОБЛАСТЬ ЗАСТОСУВАННЯ ..... | 2 |
| ПІДСТАВИ ДЛЯ РОЗРОБКИ.....                 | 2 |
| МЕТА ТА ПРИЗНАЧЕННЯ РОЗРОБКИ.....          | 2 |
| ДЖЕРЕЛА РОЗРОБКИ.....                      | 2 |
| ТЕХНІЧНІ ВИМОГИ.....                       | 3 |
| Вимоги до розробленого продукту.....       | 3 |
| Вимоги до програмного забезпечення .....   | 4 |
| Вимоги до апаратної частини .....          | 4 |
| ЕТАПИ РОЗРОБКИ .....                       | 4 |

|           |                  |          |        |      |                                                                                                                                                      |                                           |       |         |
|-----------|------------------|----------|--------|------|------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------|-------|---------|
|           |                  |          |        |      | ІАЛЦ.467200.002 ТЗ                                                                                                                                   |                                           |       |         |
|           |                  | № докум. | Підпис | Дата |                                                                                                                                                      |                                           |       |         |
| Розробив  | Чернишевич М.С.  |          |        |      | Метод та програмні засоби<br>криптографічно-строкої ідентифікації на<br>базі генераторів псевдовипадкових<br>послідовностей<br><br>Технічне завдання | Літ.                                      | Аркуш | Аркушів |
| Перевірив | Марковський О.П. |          |        |      |                                                                                                                                                      |                                           | 1     | 4       |
| Н. Контр. | Мищенко Л.Д.     |          |        |      |                                                                                                                                                      | КПІ ім. Ігоря<br>Сікорського, ФІОТ, ІО-11 |       |         |
| Затвердив |                  |          |        |      |                                                                                                                                                      |                                           |       |         |

# 1 НАЙМЕНУВАННЯ ТА ОБЛАСТЬ ЗАСТОСУВАННЯ

Технічне завдання стосується створення методу та програмних засобів криптографічно надійної ідентифікації, заснованої на використанні генераторів псевдовипадкових послідовностей.

Область впровадження — системи дистанційного надання інформаційних, програмних і обчислювальних ресурсів для віддалених користувачів із механізмами розмежування та контролю прав доступу.

## 2 ПІДСТАВИ ДЛЯ РОЗРОБКИ

Підставою для розробки є завдання на виконання дипломного проєкту кваліфікаційно-освітнього рівня “бакалавр програмної інженерії”, затверджене кафедрою обчислювальної техніки Національного технічного університету України “Київський політехнічний інститут імені Ігоря Сікорського”.

## 3 МЕТА ТА ПРИЗНАЧЕННЯ РОЗРОБКИ

Мета проєкту полягає у підвищенні ефективності процесу ідентифікації віддалених абонентів шляхом прискорення обчислювальних процедур при збереженні високого рівня захисту прав доступу до ресурсів, досягається за допомогою криптографічно-строкої технології, заснованої на концепції нульових знань.

Розробка спрямована на скорочення часу ідентифікації за рахунок використання швидкодіючих криптографічних генераторів псевдовипадкових послідовностей.

## 4 ДЖЕРЕЛА РОЗРОБКИ

4.1. Chernyshevych M. Zero-knowledge identification of remote users by utilization of pseudorandom sequences / M. Chernyshevych, I. Daiko, V. Selivanov, O. Markovskiy // Information, Computing and Intelligent systems. – 2022. – № 3.- P.42-48. DOI: 10.20535/2708-4930.3.2022

|     |      |          |        |      |                    |      |
|-----|------|----------|--------|------|--------------------|------|
|     |      |          |        |      | ІАЛЦ.467200.002 ТЗ | Арк. |
|     |      |          |        |      |                    | 2    |
| Зм. | Арк. | № докум. | Підпис | Дата |                    |      |

4.2. Васильєва М.Д.. Метод швидкої ідентифікації віддалених абонентів на основі концепції нульових знань / Васильєва М.Д., Дайко І.В., Саницький А.П. // Наука і техніка сьогодні. 2024.-№ 3(31).-С.791-803. [https://DOI.org/10.52058/2786-6025-2024-3\(31\)-791-804](https://DOI.org/10.52058/2786-6025-2024-3(31)-791-804)

4.3. Марковський О.П. Метод криптографічно строгої ідентифікації на основі незворотних перетворень на основі табличних перетворень на полях Галуа / О.П. Марковський, І.В. Дайко // Адаптивні системи автоматичного управління.- 2024.- Том.1 - № 44.- С.127-141. DOI: 10.20535/1560-8956.44.2024.302429

## 5 ТЕХНІЧНІ ВИМОГИ

### 5.1. Вимоги до розробленого продукту

5.1.1 Для забезпечення високого рівня захисту прав доступу до інформаційних ресурсів як від зовнішніх атак так і від атак з боку системи метод ідентифікації має базуватися на криптографічно-строгій концепції “нульових знань”.

5.1.2 Для забезпечення високої швидкості ідентифікації метод має базуватися на застосуванні швидкодіючих криптографічних генераторів псевдовипадкових послідовностей, що використовуються в діючих стандартах потокового шифрування даних.

5.1.3 Розрядність сеансових паролів, які мають змінюватися в кожному циклі ідентифікації віддаленого користувача має бути 256 біт.

5.1.4 В якості генераторів псевдовипадкових бітових послідовностей мають використовуватися табличні генератори на лінійних зсувних регістрах лінійною функцією зворотного зв'язку, які забезпечують період повторення послідовності не менше ніж  $2^{1200}$  і можливість встановлення поточного стану генератора. Для нелінійних перетворень мають використовуватися таблиці булевих функцій, від 12-ти змінних, що мають високу нелінійність та

|     |      |          |        |      |                    |      |
|-----|------|----------|--------|------|--------------------|------|
|     |      |          |        |      | ІАЛЦ.467200.002 ТЗ | Арк. |
|     |      |          |        |      |                    | 3    |
| Зм. | Арк. | № докум. | Підпис | Дата |                    |      |

задовольняють критерію лавинного ефекту задля стійкості до лінійного та диференційного криптоаналізу

## 5.2. Вимоги до програмного забезпечення

- Операційна система MS Windows 10
- Visual Studio 2017
- Juwa

## 5.3. Вимоги до апаратної частини

- Процесор рівня Intel i5 і вище.
- Оперативна пам'ять не менше 500 МБ.

Вільне місце на жорсткому диску не менше 100 МБ.

# 6 ЕТАПИ РОЗРОБКИ

| Назва етапів виконання                              | Термін виконання      |
|-----------------------------------------------------|-----------------------|
| Затвердження теми роботи                            | 15.04.2025-17.04.2025 |
| Вивчення та аналіз завдання                         | 18.04.2025-24.04.2025 |
| Розробка архітектури та загальної структури системи | 25.04.2025-02.05.2025 |
| Розробка структур окремих частин системи            | 05.05.2025-13.05.2025 |
| Програмна реалізація системи                        | 14.05.2025-23.05.2025 |
| Виправлення помилок                                 | 24.05.2025-27.05.2025 |
| Оформлення пояснювальної записки                    | 30.05.2025-04.06.2025 |

**ПОЯСНЮВАЛЬНА ЗАПИСКА  
ДО ДИПЛОМНОГО ПРОЄКТУ**

на тему: «Метод та програмні засоби криптографічно-строкої ідентифікації на  
базі генераторів псевдовипадкових послідовностей»

Київ – 2025

## ЗМІСТ

|                                                                                                                           |    |
|---------------------------------------------------------------------------------------------------------------------------|----|
| ПЕРЕЛІК СКОРОЧЕНЬ.....                                                                                                    | 3  |
| ВСТУП.....                                                                                                                | 4  |
| РОЗДІЛ 1.....                                                                                                             | 6  |
| 1.1 Аналіз вимог до програмних засобів ідентифікації учасників .....                                                      | 6  |
| 1.2 Огляд програмних засобів ідентифікації віддалених користувачів....                                                    | 12 |
| 1.3 Аналітичний огляд наявних схем криптографічно-строгої ідентифікації.....                                              | 15 |
| ВИСНОВОК ДО РОЗДІЛУ 1.....                                                                                                | 21 |
| РОЗДІЛ 2 .....                                                                                                            | 23 |
| 2.1 Обґрунтування вибору методу незворотного перетворення.....                                                            | 24 |
| 2.2 Розробка методу криптографічно-строгої ідентифікації за використання генераторів псевдовипадкових послідовностей..... | 31 |
| 2.3 Оцінка ефективності.....                                                                                              | 35 |
| ВИСНОВОК ДО РОЗДІЛУ 2.....                                                                                                | 41 |
| РОЗДІЛ 3.....                                                                                                             | 43 |
| 3.1 Розробка компонентів програми.....                                                                                    | 44 |
| 3.1.1 Клас InitialStateConfig .....                                                                                       | 45 |
| 3.1.2 Клас LFSRUnit .....                                                                                                 | 47 |
| 3.1.3 Клас BitStreamGenerator .....                                                                                       | 48 |
| 3.1.4 Виконавчий метод main() .....                                                                                       | 50 |

|           |                  |          |        |      |                                                                                                                                                |                                        |       |         |
|-----------|------------------|----------|--------|------|------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------|-------|---------|
|           |                  |          |        |      | ІАЛЦ.467200.003 ПЗ                                                                                                                             |                                        |       |         |
| Зм.       | Арк.             | № докум. | Підпис | Дата | Метод та програмні засоби криптографічно-строгої ідентифікації на базі генераторів псевдовипадкових послідовностей<br><br>Пояснювальна записка | Літ.                                   | Аркуш | Аркушів |
| Розробив  | Чернишев М.С.    |          |        |      |                                                                                                                                                |                                        | 1     | 62      |
| Перевірив | Марковський О.П. |          |        |      |                                                                                                                                                |                                        |       |         |
| Реценз.   | Дичка І.А.       |          |        |      |                                                                                                                                                |                                        |       |         |
| Н. Контр. | Міщенко Л.Д.     |          |        |      |                                                                                                                                                |                                        |       |         |
| Затвердив |                  |          |        |      |                                                                                                                                                | КПІ ім. Ігоря Сікорського, ФІОТ, ІО-11 |       |         |

|                                    |    |
|------------------------------------|----|
| 3.2 Розробка тестового модулю..... | 51 |
| ВИСНОВОК ДО РОЗДІЛУ 3.....         | 55 |
| ВИСНОВКИ .....                     | 57 |
| СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....   | 59 |
| ДОДАТОК 1                          |    |
| ДОДАТОК 2                          |    |
| ДОДАТОК 3                          |    |
| ДОДАТОК 4                          |    |

|     |      |          |        |      |                    |      |
|-----|------|----------|--------|------|--------------------|------|
|     |      |          |        |      | ІАЛЦ.467200.002 ТЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата |                    | 2    |

## ПЕРЕЛІК СКОРОЧЕНЬ

|      |                                                      |
|------|------------------------------------------------------|
| GDPR | General Data Protection Regulation                   |
| ZKP  | Zero-knowledge proof                                 |
| IoT  | Internet of Things                                   |
| ГПВП | Генератор псевдовипадкових послідовностей            |
| AES  | Alternative Encryption Standard                      |
| АНФ  | Алгебраїчна нормальная форма                         |
| DES  | Data Encryption Standard                             |
| GF   | Galoise Fields                                       |
| ЗЗ   | Зворотне заміщення байту мультиплікативною інверсією |
| ЗСР  | Зворотний зсув рядків матриці стану                  |
| НВІС | Надвелика інтегральна схема                          |
| ПЗР  | Прямий зсув рядків матриці стану                     |
| ПЛІС | Програмована логічна інтегральна схема               |
| ПЗ   | Пряме заміщення байту мультиплікативною інверсією    |
| ШБ   | Шифроблок                                            |
| ХП   | Хеш-перетворювач                                     |

## ВСТУП

Стрімкий розвиток інтернет-технологій зумовив широке впровадження систем віддаленого доступу до обчислювальних, інформаційних та програмних ресурсів. Такі системи забезпечують ефективну взаємодію користувачів із сервісами незалежно від їхнього фізичного місцезнаходження, що значно підвищує продуктивність і зручність виконання прикладних завдань у різних сферах діяльності.

Проте, зі зростанням масштабів використання таких технологій, різко зростає і значущість питань безпеки, зокрема - контрольованості доступів до різних віддалених ресурсів. Надійний розподіл прав доступу є критично важливим не лише для запобігання несанкціонованим діям, а й для збереження прав інтелектуальної власності, комерційної безпеки та відповідності нормативним вимогам. Центральним елементом цих процесів є ідентифікація користувача - процедура визначення особи, яка запитує доступ до ресурсу, та перевірки її повноважень [1].

Актуальність удосконалення механізмів ідентифікації пояснюється рядом факторів, що є визначальними у сучасному цифровому середовищі:

Зростанням комерційної цінності цифрових сервісів [2]. Багато ресурсів надаються за платною моделлю або вимагають ліцензійного доступу. Це створює передумови для спроб шахрайства або імітації користувачів, а отже - вимагає підвищеного рівня достовірності ідентифікації.

Поширенням віддаленого доступу в чутливих і критичних системах, таких як державні інформаційні ресурси, банківські сервіси, системи моніторингу та автоматизованого керування. У таких випадках надійність ідентифікації безпосередньо впливає на загальну інформаційну безпеку системи.

Збільшенням інтенсивності та складності кіберзагроз, що спрямовані на обхід і компрометацію механізмів доступу. Атаки типу "підміна особи",

|     |      |          |        |      |                    |      |
|-----|------|----------|--------|------|--------------------|------|
|     |      |          |        |      | ІАЛЦ.467200.003 ПЗ | Арк. |
|     |      |          |        |      |                    | 4    |
| Зм. | Арк. | № докум. | Підпис | Дата |                    |      |

використання вкрадених облікових даних або фальсифікованих токенів вимагають від систем високої стійкості до зловмисних впливів саме на етапі ідентифікації.

Значним приростом кількості користувачів, що призводить до зростання навантаження на системи автентифікації та ідентифікації. Це вимагає пошуку рішень, здатних забезпечити високу швидкодію, масштабованість та ефективність без втрати рівня безпеки. До таких рішень належать, зокрема, алгоритмічна оптимізація, використання методів розпаралелювання та впровадження сучасних криптографічних підходів.

У зв'язку з вищевикладеним, підвищення ефективності наявних механізмів ідентифікації у всіх системах віддаленого доступу набуває особливої актуальності. Розв'язання цієї задачі сприятиме зміцненню захищеності цифрових сервісів, підвищенню довіри користувачів та створенню умов для подальшого розвитку віддалених технологій у різних галузях.

|     |      |          |        |      |                    |      |
|-----|------|----------|--------|------|--------------------|------|
|     |      |          |        |      | ІАЛЦ.467200.003 ПЗ | Арк. |
|     |      |          |        |      |                    | 5    |
| Зм. | Арк. | № докум. | Підпис | Дата |                    |      |

# РОЗДІЛ 1

## ОГЛЯД НАЯВНИХ ЗАСОБІВ ІДЕНТИФІКАЦІЇ УЧАСНИКІВ ВІДДАЛЕНОЇ ІНФОРМАЦІЙНОЇ ВЗАЄМОДІЇ, ВИЗНАЧЕННЯ НАПРЯМКІВ ПІДВИЩЕННЯ ЕФЕКТИВНОСТІ

Для розробки ефективних програмних засобів ідентифікації варто здійснити комплексний аналіз поточного стану технологій та методів віддаленої взаємодії, а також ідентифікувати пов'язані з ними загрози для інформаційної безпеки. Важливим завданням є дослідити цілі, завдання та стратегії поведінки всіх учасників даного процесу - зокрема системи, користувача та потенційного злоумисника.

З огляду на отримані результати слід сформулювати чіткі вимоги до засобів ідентифікації учасників обміну у сучасному інформаційному середовищі. Виходячи з цих вимог, доцільним є проведення критичного аналізу існуючих методів ідентифікації і автентифікації, в тому числі їх відповідність актуальним викликам та здатність протистояти новітнім загрозам. Особливу увагу варто приділити виявленню вразливостей, недоліків і обмежень, притаманних поточним рішенням.

Підсумком такого дослідження має стати визначення найбільш перспективних напрямів розвитку, що дозволять створити нові підходи до ідентифікації користувачів, здатні краще відповідати сучасним вимогам безпеки у сфері віддаленої взаємодії в рамках обміну інформацією.

### 1.1 Аналіз вимог до програмних засобів ідентифікації учасників

Ідентифікація користувача є фундаментальним компонентом сучасної системи інформаційної безпеки, що відіграє критично важливу роль у

|     |      |          |        |      |                    |      |
|-----|------|----------|--------|------|--------------------|------|
|     |      |          |        |      | ІАЛЦ.467200.003 ПЗ | Арк. |
|     |      |          |        |      |                    | 6    |
| Зм. | Арк. | № докум. | Підпис | Дата |                    |      |

забезпеченні захищеного доступу до інформаційних ресурсів, особливо в умовах поширеного використання віддалених технологій [3]. У сучасному світі цифрова трансформація поширюється на всі сфери діяльності - від адміністративного управління до індустріального виробництва - надійність механізмів ідентифікації набуває стратегічного значення. Ідентифікація дозволяє системі однозначно визначити, хто саме намагається отримати доступ до її функцій, і на основі цього приймати рішення про рівень доступу до тих чи інших даних.

В умовах стрімкого зростання кількості користувачів, які працюють з інформаційними системами в режимі онлайн, зокрема через Інтернет або мобільні платформи, збільшується потреба у вдосконалених механізмах ідентифікації. Така ідентифікація повинна не лише ефективно захищати систему від несанкціонованого доступу, але й бути зручною для користувачів, забезпечуючи простоту у використанні, мінімальне навантаження на інфраструктуру та відповідність вимогам продуктивності.

Типова модель віддаленого доступу передбачає використання публічних каналів зв'язку, що значно підвищує ризики, пов'язані з перехопленням інформації, атаками типу "людина посередині" (Man-in-the-Middle), фішинговими атаками, підміною даних та іншими шкідливими діями. Крім того, не менш загрозливою є можливість внутрішніх атак, коли компрометований пристрій або обліковий запис відкриває зловмиснику доступ до системи під виглядом авторизованого користувача.

Існує безліч технічних сценаріїв, за яких здійснюється ідентифікація: від простої перевірки логіна і пароля до реалізації багаторівневих криптографічних протоколів, що використовують часові мітки, одноразові паролі, сертифікати та біометричні параметри. У більш складних системах застосовуються протоколи із динамічною генерацією сеансових ключів, багатофакторна автентифікація (2FA), а також методи на основі поведінкової біометрії - все це створює додатковий рівень безпеки та ускладнює роботу потенційному зловмиснику.

|     |      |          |        |      |                    |      |
|-----|------|----------|--------|------|--------------------|------|
|     |      |          |        |      | ІАЛЦ.467200.003 ПЗ | Арк. |
|     |      |          |        |      |                    | 7    |
| Зм. | Арк. | № докум. | Підпис | Дата |                    |      |

У сучасних умовах зростає кількість вимог до ідентифікаційних систем. По-перше, це вимога до надійності, яка передбачає стійкість до всіх відомих типів атак, зокрема атак перебору, атак повторного використання сесії, атак на інтерфейс користувача, атак на криптографічні функції. По-друге, важливою є вимога до продуктивності: системи повинні витримувати велике навантаження, швидко реагувати на запити і не створювати затримок у роботі сервісів. По-третє, особливої актуальності набуває вимога до масштабованості: ідентифікаційні механізми мають однаково ефективно працювати як у локальних мережах, так і в глобальних хмарних інфраструктурах з мільйонами користувачів.

У контексті технічної реалізації існують декілька ключових підходів до організації ідентифікації. Перший - це використання криптографічних методів, зокрема асиметричного шифрування, цифрових підписів, хешування та протоколів нульового знання (zero-knowledge proof) [4][5]. Другий - впровадження апаратних засобів ідентифікації, наприклад, токенів, смарт-карт або Trusted Platform Module. Третій - це біометричні системи, які аналізують фізіологічні або поведінкові параметри користувача. Кожен з цих підходів має свої переваги і недоліки, але у сукупності вони можуть забезпечити високий рівень захисту.

Сучасна тенденція розвитку систем автентифікації свідчить про перехід від паролів до систем безпарольної ідентифікації, яка базується на унікальності пристрою користувача або його біометричних характеристиках. Такі системи реалізуються на базі стандартів FIDO2/WebAuthn, де підтвердження особи здійснюється за допомогою криптографічного обміну з ключами, що не залишають пристрій користувача, тим самим мінімізуючи ризик їхнього викрадення.

Таким чином, ідентифікація користувача в умовах віддаленого доступу є надзвичайно важливим і складним процесом, що потребує комплексного підходу до побудови системи захисту. Вона повинна включати не лише

|     |      |          |        |      |                    |      |
|-----|------|----------|--------|------|--------------------|------|
|     |      |          |        |      | ІАЛЦ.467200.003 ПЗ | Арк. |
|     |      |          |        |      |                    | 8    |
| Зм. | Арк. | № докум. | Підпис | Дата |                    |      |

класичні методи перевірки облікових даних, але й сучасні технології криптографії, машинного навчання, аналізу поведінкових моделей та інтеграції із зовнішніми сервісами контролю безпеки.

Ефективна система ідентифікації має бути адаптивною до передових загроз, гнучкою, простою у використанні, сумісною з різноманітними платформами, а також здатною до масштабування без втрати швидкодії або рівня безпеки. Її впровадження - не лише технічне завдання, але й стратегічне рішення, котре впливатиме на довіру від користувачів, конкурентоспроможність продукту та загальну стійкість організації до кіберзагроз.

Сучасні тенденції розвитку інформаційної безпеки вимагають переосмислення підходів до ідентифікації, особливо у зв'язку з впровадженням децентралізованих платформ, блокчейн-технологій, а також посиленням законодавчих вимог щодо захисту персональних даних (такий як, GDPR у ЄС). Це все формує нові рамки, в межах яких повинні функціонувати засоби ідентифікації - вони повинні бути одночасно гнучкими, прозорими та технічно захищеними.

Важливим є також врахування соціотехнічного аспекту: не всі користувачі однаково сприймають складні механізми автентифікації, особливо коли йдеться про літніх людей або осіб із обмеженими можливостями. Тому ефективна ідентифікація повинна бути адаптивною не лише з технічної, але й з гуманітарної точки зору, тобто орієнтованою на зручність та зрозумілість у використанні. Це зумовлює актуальність досліджень у сфері UX-дизайну безпечних інтерфейсів автентифікації, що стає новим трендом в інформаційній безпеці.

Поряд із цим, актуальним є і питання захисту ідентифікаційної інформації у разі фізичного доступу до пристроїв, які використовуються для автентифікації. Наприклад, втрата смартфона з активним токеном доступу або USB-ключа може створити серйозні ризики. Тому необхідним є впровадження

|     |      |          |        |      |                    |      |
|-----|------|----------|--------|------|--------------------|------|
|     |      |          |        |      | ІАЛЦ.467200.003 ПЗ | Арк. |
|     |      |          |        |      |                    | 9    |
| Зм. | Арк. | № докум. | Підпис | Дата |                    |      |

додаткових захисних механізмів - таких як прив'язка токена до геолокації або контроль часу доби, коли зазвичай здійснюється доступ користувача.

З урахуванням вищенаведеного можна зробити висновок, що сучасна система ідентифікації повинна розглядатися як багатофакторна динамічна структура, яка постійно адаптується до змін у технічному середовищі, поведінці користувачів і нормативно-правовій базі. Вона повинна інтегрувати у собі не лише криптографічні алгоритми, але й аналіз поведінки, аналітику ризиків та контекстну обробку даних у режимі реального часу.

Крім базових принципів захисту, доцільним є впровадження механізмів динамічної адаптації ідентифікаційних процедур до контексту користувацької активності. Зокрема, йдеться про системи, здатні змінювати параметри автентифікації залежно від часу доби, місця входу або попередньої історії дій користувача. Такі системи дозволяють своєчасно реагувати на потенційні відхилення та активізувати додаткові рівні перевірки, коли поведінка користувача відрізняється від звичних шаблонів.

Розширена аналітика в межах ідентифікаційних систем також стає невід'ємною частиною безпекової інфраструктури. Завдяки сучасним засобам моніторингу можливо не лише здійснювати оцінку ефективності конкретних методів автентифікації, але й прогнозувати ймовірність інцидентів, ґрунтуючись на статистичних закономірностях. Такі методики реалізуються з використанням алгоритмів машинного навчання, що безперервно вдосконалюють свої оцінки на основі накопиченого масиву даних.

Сучасні міжнародні стандарти у сфері інформаційної безпеки, зокрема ISO/IEC 27001 та NIST SP 800-63, підкреслюють необхідність побудови систем ідентифікації з урахуванням принципів багаторівневого захисту, розподілу довіри та мінімізації обсягу оброблюваних даних. Це означає, що навіть у випадку часткового компрометування системи порушник не повинен мати змоги отримати критичну інформацію без проходження усіх рівнів автентифікації.

|     |      |          |        |      |                    |      |
|-----|------|----------|--------|------|--------------------|------|
|     |      |          |        |      | ІАЛЦ.467200.003 ПЗ | Арк. |
|     |      |          |        |      |                    | 10   |
| Зм. | Арк. | № докум. | Підпис | Дата |                    |      |

Особливу увагу при формуванні вимог до ідентифікаційних засобів варто приділяти відповідності регіональному законодавству, зокрема положенням GDPR для країн ЄС або аналогічним національним нормативам. Це забезпечує не лише технічну, а й правову захищеність процесів, пов'язаних з обробкою персональних даних користувачів, і є критично важливим при інтеграції з державними або комерційними платформами.

Інноваційним напрямом досліджень у цій сфері виступає розробка систем децентралізованої ідентифікації (Decentralized Identifiers, DID), що дозволяють користувачеві самостійно зберігати та контролювати свої цифрові ідентифікатори без необхідності централізованої бази даних. Такі підходи мають потенціал змінити парадигму автентифікації в розподілених системах, таких як блокчейн-екосистеми, і зменшити вразливість до масових витоків даних.

Варте уваги і те, що питання ідентифікації користувачів не обмежується лише технічними аспектами реалізації алгоритмів автентифікації. Воно також включає правові, етичні та організаційні чинники, що має значний вплив на формування стратегії захисту інформаційних систем. Наприклад, законодавство з питань захисту персональних даних, таке як Загальний регламент захисту даних (GDPR) в ЄС, запроваджує обов'язкові вимоги до обробки, зберігання та захисту ідентифікаційної інформації користувачів.

Одним із важливих напрямів розвитку є інтеграція механізмів ідентифікації з системами штучного інтелекту (ШІ), які здатні проводити аналіз поведінкових патернів користувача в реальному часі. Це дозволяє виявляти потенційно підозрілу активність ще до моменту завершення сеансу ідентифікації. Такі системи вже знаходять застосування в банківському секторі та у сфері електронної комерції.

Окрему категорію становлять ризики, пов'язані з людським фактором. Часто саме людська недбалість або необережність є причиною компрометації облікових записів. Тому важливо проводити регулярні тренінги з підвищення

|     |      |          |        |      |                    |      |
|-----|------|----------|--------|------|--------------------|------|
|     |      |          |        |      | ІАЛЦ.467200.003 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата |                    | 11   |

обізнаності персоналу щодо основ безпечного поводження з ідентифікаційною інформацією. Адміністративні заходи безпеки мають іти пліч-о-пліч з технічними засобами захисту.

У контексті мобільних пристроїв особливої актуальності набувають методи, що поєднують біометричну автентифікацію (наприклад, розпізнавання обличчя або відбитка пальця) з криптографічними механізмами захисту. Такі методи не лише забезпечують високий рівень захисту, але й суттєво підвищують зручність користування системою.

Ще одним аспектом, що потребує уваги, є питання зберігання облікових даних у хмарному середовищі. Забезпечення конфіденційності та цілісності таких даних у розподіленій інфраструктурі потребує використання спеціалізованих протоколів шифрування та розмежування доступу. Зокрема, поширення отримали моделі Zero Trust Architecture, які не довіряють жодному компоненту інфраструктури за замовчуванням.

## 1.2 Огляд програмних засобів ідентифікації віддалених користувачів

На поточному етапі динамічних змін та розвитку у інформаційних технологіях, таких як мережевих рішень та архітектур хмарних обчислень, дедалі більшого поширення набуває потреба в надійній ідентифікації користувачів, що працюють з інформаційними системами на відстані. Більшість таких систем, у тому числі й ті, що використовуються в комерційних, урядових, освітніх та медичних структурах, продовжує покладатися на класичний підхід до ідентифікації, що ґрунтується на комбінації імені користувача (логіну) та пароля. Цей метод залишається найбільш розповсюдженим завдяки своїй простоті реалізації та зручності для користувача, однак у сучасних умовах він має суттєві недоліки.

Сутність цього підходу полягає у порівнянні введених користувачем облікових даних зі збереженими у внутрішній базі даних системи. Якщо

|     |      |          |        |      |                    |      |
|-----|------|----------|--------|------|--------------------|------|
|     |      |          |        |      | ІАЛЦ.467200.003 ПЗ | Арк. |
|     |      |          |        |      |                    | 12   |
| Зм. | Арк. | № докум. | Підпис | Дата |                    |      |

відповідність підтверджується, користувач отримує доступ до ресурсу. Проте така схема має низку вразливостей, які критично впливають на загальний рівень безпеки інформаційної взаємодії. Найсерйознішою проблемою є низька надійність паролів, які створюють самі користувачі. Дослідження показують, що велика частина людей використовує короткі, легко передбачувані або навіть повторювані паролі, які можна зламати за допомогою словникових атак або brute-force методів упродовж кількох хвилин або годин, залежно від потужності обчислювальних засобів.

Крім того, зловмисники можуть застосовувати різноманітні техніки соціальної інженерії, спрямовані на психологічний вплив для одержання несанкціонованого доступу до користувацьких облікових записів. Часто користувачі самостійно, навіть несвідомо, розголошують свої паролі, реагуючи на фішингові листи або веб-інтерфейси, що імітують справжні сервіси. Також серйозною загрозою залишається використання шкідливого програмного забезпечення - кейлогерів, троянів, шпигунських модулів - які здатні перехоплювати паролі без відома користувача.

Ще однією критичною проблемою є передача облікових даних через незахищені або недостатньо зашифровані канали зв'язку. У разі відсутності надійного криптографічного захисту, зловмисники можуть перехопити ці дані, особливо у відкритих Wi-Fi мережах, які часто використовуються при віддаленому доступі до систем.

Для підвищення надійності класичних систем ідентифікації впроваджуються певні технічні заходи, зокрема ускладнення паролів шляхом використання різних комбінацій цифр, великих та малих літер, спеціальних символів, а також періодична зміна паролів. Проте ці заходи, хоч і підвищують криптостійкість пароля, водночас суттєво знижують зручність використання системи - користувачеві складно запам'ятати складні паролі, особливо коли їх потрібно змінювати часто або створювати окремо для кожного сервісу.

|     |      |          |        |      |                    |      |
|-----|------|----------|--------|------|--------------------|------|
|     |      |          |        |      | ІАЛЦ.467200.003 ПЗ | Арк. |
|     |      |          |        |      |                    | 13   |
| Зм. | Арк. | № докум. | Підпис | Дата |                    |      |

Як один з варіантів вирішення цієї проблеми активно впроваджуються менеджери паролів - спеціалізовані програми, які автоматично створюють, зберігають та підставляють у потрібний момент складні паролі, зашифровані всередині захищеного середовища. Такі програми зменшують імовірність людської помилки, а також знижують ризик компрометації паролів. Удосконалені менеджери дозволяють також створювати одноразові або тимчасові паролі, що суттєво підвищує рівень безпеки.

Паралельно з класичними підходами, у технічному середовищі активно розвиваються більш сучасні методи автентифікації, зокрема схеми, що реалізують концепцію нульових знань. Основна ідея таких схем полягає в тому, що користувач може довести свою особу, не розкриваючи при цьому жодної приватної інформації. Замість передавання самого пароля або його похідного значення, користувач створює криптографічно захищену відповідь на виклик системи. Це зменшує ймовірність того, що навіть у випадку перехоплення даних під час сесії автентифікації злоумисник зможе повторно їх використати.

Наприклад, в UNIX-системах широко застосовується багаторівнева схема перевірки користувача [6]. На першому етапі система перевіряє, чи існує у її базі ID, який вводить користувач. Якщо такий запис є, то надалі перевіряється відповідність автентифікаційної інформації, яка обробляється з використанням спеціальних хеш-функцій. Завдяки цьому в системі не зберігаються самі паролі, що додатково знижує ризик витоку даних.

Методи ідентифікації можна умовно поділити на дві великі категорії: слабкі та сильні. До слабких належать ті, що побудовані на основі знання пароля. До сильних - ті, що впроваджують криптографічні протоколи високого рівня складності, такі як zero-knowledge proof. Перевага останніх полягає у високій стійкості до атак типу «людина посередині», а також у відсутності необхідності передавати або зберігати паролі в явному вигляді.

Найбільш перспективними вважаються протоколи, що ґрунтуються на незворотних математичних перетвореннях, таких як дискретне логарифмування

|     |      |          |        |      |                    |      |
|-----|------|----------|--------|------|--------------------|------|
|     |      |          |        |      | ІАЛЦ.467200.003 ПЗ | Арк. |
|     |      |          |        |      |                    | 14   |
| Зм. | Арк. | № докум. | Підпис | Дата |                    |      |

або задача факторизації великих чисел [7]. Реалізація таких підходів забезпечує високий рівень захисту навіть у випадках використання відкритих каналів зв'язку.

Таким чином, можна зробити висновок, що у зв'язку з актуальними викликами цифрової безпеки традиційні схеми ідентифікації потребують модернізації або заміни на більш досконалі криптографічно-стійкі механізми. Отже, основною метою наступних досліджень має бути підвищення продуктивності криптографічних алгоритмів, зменшення затримок у процесі автентифікації та підвищення зручності користування при збереженні максимально можливого рівня захисту.

### **1.3 Аналітичний огляд наявних схем криптографічно-строкої ідентифікації**

У контексті зростаючих вимог до конфіденційності, цілісності та автентичності в цифрових комунікаційних системах, криптографічні протоколи з нульовим розголошенням (ZKP) набувають все більшого поширення як інструмент, що поєднує строгі математичні гарантії безпеки з практичною ефективністю реалізації [22][26][28]. Їх особливе значення підкреслюється в умовах, коли взаємодія між сторонами відбувається у відкритому або недовіреному середовищі, де будь-який витік або компрометація даних може мати критичні наслідки.

Загальновідомо, що стандартні методи автентифікації, такі як збереження паролів у хешованому вигляді, або використання пари логін–пароль, не гарантують достатнього рівня безпеки. Вони вразливі до атак типу «грубої сили», фішингу, перехоплення та повторного відтворення повідомлень. Протоколи на основі нульових знань, всупереч вказаним підходам, повністю виключають необхідність передавання секретної інформації, оскільки перевірка автентичності відбувається без її розкриття.

|     |      |          |        |      |                    |      |
|-----|------|----------|--------|------|--------------------|------|
|     |      |          |        |      | ІАЛЦ.467200.003 ПЗ | Арк. |
|     |      |          |        |      |                    | 15   |
| Зм. | Арк. | № докум. | Підпис | Дата |                    |      |

Ключова ідея полягає у тому, що доказувач переконує верифікатора у володінні деяким знанням або ключем, не відкриваючи сам ключ. У цьому полягає фундаментальна перевага ZKP - забезпечення інформаційної асиметрії, де одна сторона впевнена у достовірності другої без необхідності володіти змістом її секрету.

Прикладом є протокол Guillou–Quisquater [8][10]. У ньому використовуються два простих числа  $p$  та  $q$ , і модуль обчислюється як :

$$m = p * q$$

Користувач передає відкриті значення  $v$  та  $J$ , де  $J * B^v \bmod m = 1$ .

У процесі автентифікації обчислюється:  $P_1 = r^v \bmod m$ ,  $P_2 = r * B^d \bmod m$ , і перевірка:  $T = P_2^v * J^d \bmod m$ .

Іншим прикладом є Fiat–Shamir [9][12]. Там використовується:  $m = p * q$ ,  
 $v = s^{-2} \bmod m$ .

У раундах автентифікації:  $x = r^2 \bmod m$ , якщо  $b = 0$ , відправляється  $r$ , якщо  $b = 1$  -  $y = r * s \bmod m$ .

Перевірка:  $x = y^2 * v \bmod m$ .

Сучасні розробки також включають zk-SNARK, zk-STARK та post-quantum схеми на базі ґраток [10][11]. Вони демонструють перспективність у захисті високонебезпечних систем, включаючи блокчейн, електронне голосування, децентралізовані ідентичності (DID) тощо.

ZKP-схеми гарантують:

1. непідробність - тільки власник знання може пройти перевірку,
2. повну конфіденційність - секрет не розголошується,
3. стійкість до атак повторного використання.

З технічного погляду, формальною вимогою до ZKP є забезпечення повноти, коректності та нульового розголошення [12].

У реальному житті ефективність реалізації залежить від оптимізації модульних обчислень, підтримки апаратного прискорення (наприклад, через

TRM або криптографічні співпроцесори), а також використання ефективних хеш-функцій (SHA-256, Кессак тощо).

Таким чином, концепція нульових знань не лише забезпечує фундамент для побудови криптографічно стійких протоколів, але й відкриває нові горизонти для проєктування безпечних цифрових середовищ. Вона відповідає сучасним вимогам до анонімності, надійності та масштабованості, роблячи її незамінною у цифровому світі, що постійно змінюється.

Окремим напрямом у розвитку протоколів з нульовим розголошенням є їх застосування у сфері інтернету речей (IoT), де верифікація пристроїв часто відбувається без постійного підключення до інтернету або при обмежених обчислювальних можливостях. Тут особливо актуальною є потреба у легковагових протоколах, що забезпечують автентифікацію без значного навантаження на процесор чи енергоспоживання. Саме тому деякі версії ZKP, зокрема схемі Schnorr або її модифікації, почали застосовуватись в сенсорних мережах та системах Smart Home [11][14].

Ще однією перспективною галуззю для імплементації ZKP є біометричні системи, де існує постійна потреба у захисті шаблонів біометричних даних. Наприклад, ZKP дозволяє перевірити, чи відповідає відсканований відбиток пальця запису у базі, не розкриваючи сам шаблон. Це усуває ризики, пов'язані із втратою або викраденням біометричних даних, які, у протилежність пароллю, неможливо змінити.

У деяких криптографічних платформах реалізовано комбінацію між ідентифікацією з нульовим розголошенням та протоколами нульових знань без взаємодії (non-interactive ZKP або NIZKP). Такі протоколи дають змогу створити цифровий доказ, який можна перевірити будь-якою третьою стороною без участі доказувача. Це критично важливо, наприклад, для цифрового нотаріату або верифікації підписів у документах, що перевіряються після факту підписання.

|     |      |          |        |      |                    |      |
|-----|------|----------|--------|------|--------------------|------|
|     |      |          |        |      | ІАЛЦ.467200.003 ПЗ | Арк. |
|     |      |          |        |      |                    | 17   |
| Зм. | Арк. | № докум. | Підпис | Дата |                    |      |

Окремої уваги заслуговують дослідження в галузі постквантової криптографії, де класичні алгоритми втрачають стійкість у зв'язку з появою квантових обчислювальних потужностей. ZKP на основі ґраткових задач, таких як Learning With Errors (LWE) або Short Integer Solution (SIS), демонструють високий потенціал до стійкості в умовах квантової загрози. Однак їх реалізація пов'язана зі збільшенням обсягів передаваних даних та необхідністю оптимізації алгоритмів кодування.

Також варто відзначити роль криптографічних протоколів нульових знань у реалізації електронних паспортів та посвідчень, де забезпечується можливість доведення правомірності особи без надання надлишкової персональної інформації. Це дозволяє, наприклад, довести, що користувач досяг певного віку, не повідомляючи конкретну дату народження.

З огляду на зростаючі виклики у сфері цифрової безпеки та приватності, стає очевидним, що протоколи з нульовим розголошенням у найближчому майбутньому перейдуть із категорії академічних розробок до широкомасштабного промислового впровадження. Вони вже формують основу сучасних систем анонімної автентифікації, конфіденційних транзакцій, цифрових ідентичностей та безпечного зберігання даних.

Таким чином, дослідження та вдосконалення методів автентифікації на основі ZKP має стратегічне значення для побудови довірених цифрових платформ. Ці технології не лише забезпечують відповідність вимогам сучасного законодавства щодо захисту персональних даних, таких як GDPR або eIDAS, але й створюють інноваційне середовище, де безпека, приватність та зручність користування не суперечать одне одному, а гармонійно співіснують.

Значне поширення протоколів з нульовим розголошенням спостерігається також у банківському секторі, де вони використовуються для автентифікації клієнтів у системах дистанційного банкінгу. В наслідок забезпечення надійного рівня захисту без передачі конфіденційної інформації, ZKP стали ефективним

|     |      |          |        |      |                    |      |
|-----|------|----------|--------|------|--------------------|------|
|     |      |          |        |      | ІАЛЦ.467200.003 ПЗ | Арк. |
|     |      |          |        |      |                    | 18   |
| Зм. | Арк. | № докум. | Підпис | Дата |                    |      |

засобом захисту від загрози атак типу «людина посередині» (Man-in-the-Middle), фішингу та підробки електронних транзакцій.

Сучасні фінансові технології (FinTech) активно інтегрують криптографічні інструменти нового покоління, зокрема для побудови протоколів перевірки особи, видачі цифрових сертифікатів, а також для динамічного управління доступом. Наприклад, у випадку мультифакторної автентифікації, ZKP може виступати додатковим фактором, що підтверджує знання приватного ключа, при цьому не передаючи його або його похідні значення по мережі.

Не менш актуальним є застосування ZKP у системах електронного голосування, особливо у демократичних процесах, де критично важливим є збереження анонімності виборця при гарантуванні достовірності голосу. Завдяки схемам нульових знань стає можливим підтвердити факт голосування, не розкриваючи конкретного вибору, що особливо важливо у політично чутливих регіонах або в умовах воєнного стану.

У галузі охорони здоров'я застосування ZKP дозволяє реалізувати безпечний обмін медичними даними між установами без розкриття повної інформації про пацієнта. Таким чином, одна лікарня може переконатись у наявності певного медичного показника або діагнозу без доступу до повного медичного досьє, що відповідає вимогам законодавства про захист персональних даних.

У сфері правосуддя ZKP дозволяють адвокату або представнику сторони підтвердити легітимність певного документа або факту без розголошення його змісту. Це відкриває нові можливості для дистанційного судочинства, де важлива не лише автентичність, а й дотримання конфіденційності.

Варто також згадати про використання технологій нульових знань у проєктах з відкритим кодом, де спільноти розробників експериментують із застосуванням ZKP у нових галузях, таких як децентралізовані автономні організації (DAO), обмін цифровими активами, створення динамічних токенів або автентифікація у метавсесвітах.

|     |      |          |        |      |                    |      |
|-----|------|----------|--------|------|--------------------|------|
|     |      |          |        |      | ІАЛЦ.467200.003 ПЗ | Арк. |
|     |      |          |        |      |                    | 19   |
| Зм. | Арк. | № докум. | Підпис | Дата |                    |      |

Із технологічного погляду одним з найважливіших викликів при реалізації ZKP-протоколів залишається компроміс між рівнем безпеки, швидкістю виконання та обсягом необхідних обчислень. Для подолання цього бар'єру дедалі частіше використовуються спеціалізовані апаратні прискорювачі - криптографічні сопроцесори, платформи FPGA, апаратні модулі довіри (TPM), що дає змогу реалізувати ZKP навіть у малопотужних пристроях.

З наукової точки зору перспективними напрямками є розробка так званих recursive zero-knowledge proofs - доказів, які інкапсулюють попередні докази та дозволяють суттєво зменшити обсяг перевірки [15][16]. Також активно розвивається підхід PLONK (Permutations over Lagrange-bases for Oecumenical Noninteractive arguments of Knowledge), який дозволяє створювати універсальні та ефективні докази для широкого спектра обчислювальних задач.

Загалом, нині спостерігається перехід ZKP-технологій із сфери фундаментальної криптографії до прикладного програмування, де вони вже є частиною програмних фреймворків, платформ, API та стандартів, таких як zkSync, Aztec, StarkNet тощо. Це свідчить про те, що нульові знання стають не лише інструментом захисту, а й рушієм інновацій у цифровій трансформації.

|     |      |          |        |      |                    |      |
|-----|------|----------|--------|------|--------------------|------|
|     |      |          |        |      | ІАЛЦ.467200.003 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата |                    | 20   |

# ВИСНОВОК ДО РОЗДІЛУ 1

У результаті проведеного комплексу аналітичних та дослідницьких заходів, що охоплюють перший розділ бакалаврського проєкту, вдалося сформулювати узагальнене уявлення про сучасний стан безпекових аспектів систем віддаленої взаємодії. Дослідження включало в себе вивчення актуальних тенденцій розвитку таких систем, виявлення основних векторів еволюції загроз інформаційній безпеці, а також критичний аналіз існуючих підходів до ідентифікації учасників інформаційного обміну. На основі отриманих результатів можна сформулювати наступні висновки:

Інтенсивний розвиток технологій цифрової взаємодії на відстані породжує нові виклики для засобів автентифікації, зокрема - необхідність забезпечення захисту від нових типів атак, таких як витіснення законного користувача після початку сесії, або перехоплення даних через відкриті канали зв'язку. У зв'язку з цим, сучасні засоби ідентифікації повинні забезпечувати не лише криптографічну стійкість, а й високу швидкодію, адаптовану до реального часу.

Аналіз загроз і специфіки віддаленої взаємодії показав, що найбільш ефективним засобом захисту від згаданих атак є використання криптографічно строгих методів, які базуються на парадигмі нульових знань (zero-knowledge proofs). Такі методи дозволяють виконувати перевірку достовірності без розкриття секретної інформації навіть у відкритому середовищі.

Проведене зіставлення сучасних схем ідентифікації довело, що алгоритми, побудовані на класичних незворотних перетвореннях із теорії чисел (наприклад, факторизація чи дискретне логарифмування), не відповідають вимогам високої продуктивності в умовах обмежених ресурсів. Їхня апаратна реалізація на малопотужних пристроях, таких як вбудовані системи, смарт-

|     |      |          |        |      |                    |      |
|-----|------|----------|--------|------|--------------------|------|
|     |      |          |        |      | ІАЛЦ.467200.003 ПЗ | Арк. |
|     |      |          |        |      |                    | 21   |
| Зм. | Арк. | № докум. | Підпис | Дата |                    |      |

карти чи мікроконтролери, є малопридатною через високу обчислювальну складність.

Найбільш перспективним напрямом розвитку є криптографічні схеми, які використовують математичну основу у вигляді незворотних булевих перетворень. До таких належать схеми, де автентифікаційні паролі формуються у вигляді ланцюжків, кожен елемент яких отримується за допомогою односторонніх перетворень булевої алгебри. Як правило, у таких схемах застосовуються хеш-функції або блочні шифри, що забезпечують ефективне формування сеансових паролів. Поглиблений аналіз таких підходів виявив, що застосування генераторів псевдовипадкових двійкових послідовностей надає змогу значно покращити швидкодію процесу ідентифікації без втрати стійкості до криптоаналітичних атак.

Загалом, результати проведеного дослідження свідчать про доцільність подальшого розвитку і впровадження схем ідентифікації, побудованих на булевих перетвореннях, з орієнтацією на їх реалізацію у реальних технічних умовах із врахуванням обмежень обчислювальних ресурсів.

|     |      |          |        |      |                    |      |
|-----|------|----------|--------|------|--------------------|------|
|     |      |          |        |      | ІАЛЦ.467200.003 ПЗ | Арк. |
|     |      |          |        |      |                    | 22   |
| Зм. | Арк. | № докум. | Підпис | Дата |                    |      |

## РОЗДІЛ 2

# РОЗРОБКА МЕТОДУ КРИПТОГРАФІЧНО-СТРОГОЇ ІДЕНТИФІКАЦІЇ З ВИКОРИСТАННЯМ ГЕНЕРАТОРІВ ПСЕВДОВИПАДКОВИХ ПОСЛІДОВНОСТЕЙ

У оглядовому розділі даного бакалаврського дослідження було встановлено, що в сучасному інформаційному середовищі особливо небезпечним і поширеним видом атак залишаються так звані атаки типу «людина посередині» (англ. man-in-the-middle attacks) [12]. Значення таких атак базується на втручанні зловмисника в комунікаційний канал між автентичним користувачем і системою вже після проходження процедури початкової ідентифікації. Таким чином, легітимний користувач фактично усувається з активної взаємодії з системою, а сеанс перехоплюється зловмисником. Одним з найефективніших способів реалізації такого втручання є блокування користувача на одному з вузлів маршрутизації чи мережевої комутації [13], що дозволяє створити видимість безперервності сесії зі сторони системи.

З урахуванням вищезазначеного, основним завданням дослідження є створення інноваційного методу ідентифікації користувача, що базується на криптографічно строгих принципах. Такий метод має забезпечувати ефективну протидію атакам на втручання в інформаційну сесію шляхом впровадження механізму регулярного підтвердження наявності активного контакту між системою та автентичним користувачем протягом усього періоду взаємодії. Інакше кажучи, необхідно передбачити первинну автентифікацію при встановленні з'єднання та подальші численні вторинні цикли перевірки автентичності, які ініціюються з певною періодичністю або

|     |      |          |        |      |                    |      |
|-----|------|----------|--------|------|--------------------|------|
|     |      |          |        |      | ІАЛЦ.467200.003 ПЗ | Арк. |
|     |      |          |        |      |                    | 23   |
| Зм. | Арк. | № докум. | Підпис | Дата |                    |      |

випадковим чином з боку системи. Завдяки цьому унеможлиблюється збереження зловмисником сеансу після витіснення легітимного учасника.

Крім функціональної ефективності, новий метод має задовольняти вимоги до продуктивності - тобто забезпечувати мінімальний час реакції при виконанні процедур автентифікації, навіть у системах з великою кількістю одночасних користувачів. Водночас, критично важливою вимогою є дотримання принципів криптографічної строгості, що передбачає використання концепції “доказів з нульовим розголошенням” (zero-knowledge proofs). Цей підхід дозволяє досягти високого рівня захищеності без необхідності передавати чи зберігати секретні дані у відкритому вигляді.

У межах другого розділу даного дослідження передбачається здійснити теоретично обґрунтований вибір типу незворотного перетворення, який забезпечить і достатній рівень криптостійкості, і прийнятну швидкодію. До того ж, має бути розроблено загальну архітектуру та організаційну структуру ідентифікаційної системи, а також детально описані етапи функціонування авторизаційного механізму. Особлива увага приділяється побудові послідовних процедур первинної та вторинної автентифікації, їх внутрішній логіці, параметрам налаштування та варіантам оптимізації. Важливо також провести верифікацію функціональної правильності запропонованої моделі, довести її ефективність за допомогою об'єктивних критеріїв оцінки захищеності та швидкодії, а також запропонувати можливі шляхи подальшої еволюції методики відповідно до розвитку апаратного забезпечення та загроз безпеці в майбутньому.

## 2.1 Обґрунтування вибору методу незворотного перетворення

У першому розділі дипломної роботи виконано детальний аналіз сучасні загрози, пов'язані з автентифікацією користувачів у відкритих цифрових

|     |      |          |        |      |                    |      |
|-----|------|----------|--------|------|--------------------|------|
|     |      |          |        |      | ІАЛЦ.467200.003 ПЗ | Арк. |
|     |      |          |        |      |                    | 24   |
| Зм. | Арк. | № докум. | Підпис | Дата |                    |      |

середовищах. Як результат, сформовано висновок про доцільність використання методів, що базуються на парадигмі «нульових знань» (Zero-Knowledge Proof), які дозволяють досягати високого ступеня безпеки без необхідності розкриття критичних даних під час обміну. Особливої актуальності це набуває у випадках дистанційної взаємодії, де ризик несанкціонованого втручання зловмисника суттєво зростає.

Ключовим технічним компонентом таких схем виступає незворотне перетворення - математична функція, що забезпечує складність обчислення зворотного образу. Особливо ефективними вважаються перетворення, що базуються на використанні складних булевих функцій, які демонструють істотну обчислювальну складність при спробах інверсії. Для них характерна відсутність аналітичних методів обчислення оберненої функції, через що єдиною можливою стратегією отримання зворотного значення залишається метод повного перебору всіх варіантів вхідних даних, що практично нереалізовано при значній розрядності вхідного вектора.

На практиці використовуються різні типи незворотних функцій, серед яких найбільш поширені:

- криптографічні хеш-функції (SHA-256, SHA-3 тощо), які широко застосовуються у створенні цифрових підписів, перевірці цілісності повідомлень, збереженні паролів;
- блочні шифри (AES, DES), що працюють за принципом обробки даних блоками та забезпечують їх захист у зашифрованому вигляді [25][27];
- генератори псевдовипадкових послідовностей (ПВДП), що функціонують на основі математичних алгоритмів та здатні забезпечити високу швидкодію, що критично важливо у реальному часі, наприклад, для потокового шифрування.

Особливо актуальними генератори ПВДП стають у випадках, коли потрібна динамічна генерація ключів або сеансів автентифікації, які неможливо передбачити або повторити без знання початкових параметрів. Наприклад, у

|     |      |          |        |      |                    |      |
|-----|------|----------|--------|------|--------------------|------|
|     |      |          |        |      | ІАЛЦ.467200.003 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата |                    | 25   |

мобільних додатках з біометричною автентифікацією або в банківських системах, де безперервно генеруються одноразові коди доступу.

Генератори ПВДП моделюються як скінченні автомати, що складаються з:

- вектора поточного стану  $Q$  (пам'ять);
- функції переходу  $\Phi$ , яка обчислює новий стан на основі попереднього;
- функції виходу  $\Psi$ , яка формує вихідний сигнал на основі внутрішнього стану.

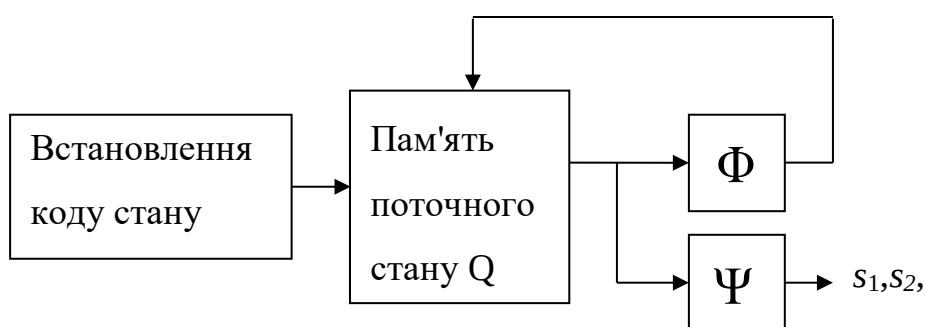


Рис.2.1. Абстрактна модель генератора ПВДП автомату

Для забезпечення криптостійкості та практичної ефективності, необхідно, щоб період вихідної послідовності був максимально довгим - наближеним до значення  $2^n$ , де  $n$  - кількість бітів у векторі стану. Окрім того, функція виходу  $\Psi$  повинна бути незворотною, щоб запобігти можливості аналітичного визначення внутрішнього стану генератора, навіть за умови знання великої кількості вихідних даних.

Одним із найефективніших рішень у даній галузі є використання комбінованих архітектур, які об'єднують сильні сторони різних типів генераторів. Наприклад, поєднання лінійної бази у вигляді LFSR із додатковим нелінійним модулятором значно підвищує криптостійкість без істотного зниження продуктивності.

Найбільш поширеним базовим елементом генераторів ПВДП є зсувні регістри з лінійною функцією зворотного зв'язку (LFSR - Linear Feedback Shift Register), які демонструють високі показники продуктивності при апаратній реалізації. Вони функціонують на основі лінійних булевих функцій,

побудованих за допомогою непривідних поліномів у полях Галуа, що дозволяє досягати максимального періоду у  $2^n - 1$  [8][17][18].

Незважаючи на це, класичні LFSR мають один суттєвий недолік - лінійність структури дає змогу відтворити внутрішній стан генератора при знанні обмеженої кількості бітів вихідної послідовності. Для усунення цього недоліку застосовуються додаткові компоненти - нелінійні перетворювачі, які ускладнюють залежність між внутрішнім станом і вихідним сигналом, тим самим значно підвищуючи стійкість системи до криптоаналізу.

У практичній реалізації це означає впровадження булевих функцій високого порядку, які складно представити у вигляді простих логічних виразів. Вибір таких функцій зазвичай відбувається на основі емпіричного аналізу або генетичних алгоритмів, здатних знаходити наближено оптимальні комбінації.

Процедура генерації послідовності передбачає використання вхідного бінарного вектора  $X = (x_1, x_2, \dots, x_n)$ , де кожен  $x_i$  - це двійковий біт. Над ним виконується функція  $F(X)$ , яка формує вихід  $Y$ . Для додаткової варіативності та підвищення захищеності застосовується ключовий модифікатор, який змінює структуру або параметри функції  $F$ .

Генератори ПВДП використовуються для створення послідовностей заданої довжини. У випадку, коли результат виходить за межі допустимого діапазону (наприклад, при генерації випадкових чисел), значення відкидається і генерується нове. Таким чином забезпечується рівномірність розподілу значень у заданому інтервалі.

До основних вимог до генераторів ПВДП, придатних для використання в захищених системах, належать:

- забезпечення великого періоду повторення (близько  $2^n$ );
- відповідність критеріям статистичної випадковості (тестування NIST, DieHard тощо);
- стійкість до прогнозування (наступний елемент послідовності - непередбачуваний);

|     |      |          |        |      |                    |      |
|-----|------|----------|--------|------|--------------------|------|
|     |      |          |        |      | ІАЛЦ.467200.003 ПЗ | Арк. |
|     |      |          |        |      |                    | 27   |
| Зм. | Арк. | № докум. | Підпис | Дата |                    |      |

- висока швидкодія та низьке споживання обчислювальних ресурсів.

Застосування ПВДП у сфері безпеки охоплює:

- генерацію одноразових паролів (ОТР);
- формування одноразових токенів доступу;
- побудову ключових пар для асиметричного шифрування;
- реалізацію поточкових шифрувальників для реального часу;
- підтримку процедур автентифікації на основі концепції «нульових знань».

Одним із критичних параметрів таких генераторів є їх здатність до визначеного відтворення при однакових початкових умовах - це дозволяє застосовувати їх у синхронізованих схемах автентифікації та передачі даних між віддаленими вузлами [19][20].

Для підвищення криптостійкості поширено використовують двокаскадні структури, у яких перший каскад складається з декількох незалежних LFSR, а другий - з нелінійної булевої функції  $f(x_1, x_2, \dots, x_h)$ , що приймає значення з кожного регістра.

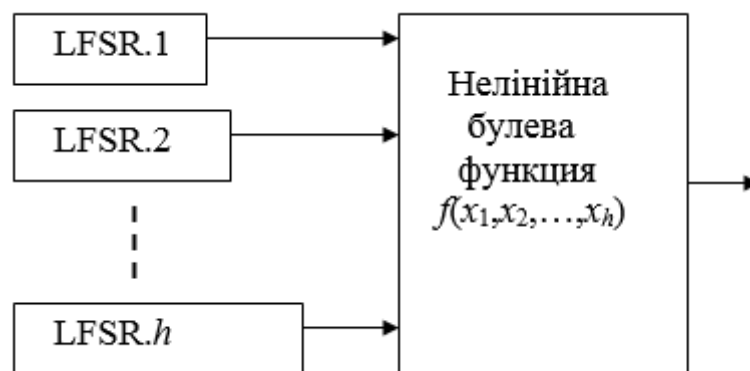


Рис.2.2. Двокаскадна структура генератора ПВДП

Такий підхід дозволяє мінімізувати автокореляційні зв'язки між елементами послідовності, забезпечити рівномірність її розподілу та захист від загрози атак виду known-output attack.

Іншим ефективним методом є використання регістрів із нелінійною функцією зворотного зв'язку (NFSR - Nonlinear Feedback Shift Register), які

об'єднують переваги LFSR із вищою стійкістю до аналітичного аналізу. Їх особливістю є складність побудови функції  $f(x_1, x_2, \dots, x_n)$ , яка гарантує період повторення  $2^n$ , але при цьому значно ускладнює можливість криптоаналізу.

Кількість можливих нелінійних функцій для генерації ПВДП визначається виразом:

$$N = 2^{2^n}$$

Навіть при значенні  $n = 10$ , кількість усіх можливих функцій становить  $2^{1024}$ , однак лише мізерна частина з них є стійкими до атак. Це обумовлює необхідність ретельного підбору булевих функцій із високою нелінійністю, великим алгебраїчним степенем і низькою кореляцією.

Саме тому розробка таких перетворень є складною, але необхідною умовою для гарантування незворотності.

Отже, у рамках поставленої задачі - побудови ефективного механізму ідентифікації на основі «нульових знань» - доцільним є застосування генераторів ПВДП з розширеною структурою (NFSR або комбінованих LFSR). Це дозволяє одночасно досягти високої швидкодії, забезпечити криптостійкість та відповідати вимогам апаратної реалізації для використання у вбудованих або мобільних системах. Саме такий підхід забезпечує баланс між складністю обчислень та надійністю, що робить його практично доцільним для широкого кола безпекових задач.

Варто зазначити, що генератори псевдовипадкових послідовностей є не лише як технічна реалізація незворотних перетворень у схемах автентифікації, але й як фундаментальний елемент широкого спектру криптографічних механізмів. Зокрема, вони активно застосовуються у протоколах симетричного та асиметричного шифрування, створенні ініціалізаційних векторів, генерації сольових значень (salt) для функцій хешування паролів, а також у механізмах генерації випадкових викликів (challenge) у протоколах ідентифікації на основі нульових знань.

У цьому контексті особливої уваги заслуговують так звані CSPRNG (Cryptographically Secure Pseudorandom Number Generators) - криптографічно-стійкі генератори псевдовипадкових чисел. Їх визначальною характеристикою є здатність витримувати криптоаналітичні атаки, що спрямовані на відновлення внутрішнього стану або прогнозування слідуючих значень у послідовності. У випадках реалізації схем «нульових знань» це надзвичайно важливо, оскільки будь-який витік чи слабкість у генераторі може потенційно призвести до компрометації всієї системи.

Додаткову складність забезпечення стійкості ПВДП створює необхідність їхньої ефективної інтеграції з апаратними платформами. У вбудованих системах та пристроях Інтернету речей (IoT), де обчислювальні ресурси обмежені, виникає потреба в оптимізації архітектури генератора чином, який зміг би забезпечити компроміс між швидкодією, розмірами апаратної реалізації та рівнем криптографічного захисту. За таких умов перевагу віддають генераторам на основі LFSR/NFSR, які реалізуються на логічному рівні (FPGA, ASIC) з мінімальними витратами апаратних ресурсів.

Зростаючий попит на генератори ПВДП також спостерігається в сучасних протоколах мультифакторної автентифікації, де окремі сеанси захисту будуються динамічно на основі унікальних, випадково згенерованих даних. Наприклад, при реалізації алгоритмів Time-based OTP (TOTP) або HMAC-based OTP (HOTP), генератори виконують роль джерела ентропії, що забезпечує унікальність кожного коду доступу.

Важливо зазначити, що надійність генератора ПВДП у схемі «нульових знань» не лише визначає його стійкість до атак, але й напряду впливає на коректність самого протоколу. У класичних ZKP-протоколах типу  $\Sigma$ -протоколів, де використовуються випадкові виклики (challenges), які надсилаються від перевіряча до доводячого, генератор повинен забезпечити повну непередбачуваність та рівномірність вибору. У протилежному випадку зломисник може скористатися статистичними відхиленнями для побудови

|     |      |          |        |      |                    |      |
|-----|------|----------|--------|------|--------------------|------|
|     |      |          |        |      | ІАЛЦ.467200.003 ПЗ | Арк. |
|     |      |          |        |      |                    | 30   |
| Зм. | Арк. | № докум. | Підпис | Дата |                    |      |

атаки повторного відтворення (replay attack) або визначення шаблонів у послідовності.

Задля отримання більшої довіри до використовуваних генераторів, підходи сьогодення передбачають їх сертифікацію відповідно до міжнародних стандартів, зокрема NIST SP 800-90A/B/C, а також використання незалежного тестування за допомогою статистичних пакетів - наприклад, TestU01, Dieharder або Ent. Це дозволяє виявити наявність потенційних вразливостей ще на етапі проектування криптографічної системи.

На завершення, варто підкреслити, що у контексті реалізації методу криптографічно-строкої ідентифікації базуючись на парадигмі нульових знань, генератори ПВДП виступають критичним компонентом. Від їхньої якості залежить не лише успішність проходження процедури автентифікації, але й загальна стійкість системи до широкого спектру атак, зокрема атак прогнозування, синхронізації та компрометації сеансу. Тому належне конструювання, верифікація та тестування таких генераторів має бути невід'ємною частиною будь-якої архітектури систем захисту інформації.

## **2.2 Розробка методу криптографічно-строкої ідентифікації за використання генераторів псевдовипадкових послідовностей**

Задля підвищення ефективності криптографічно-строкої ідентифікації віддалених користувачів в інтегрованих цифрових системах, з урахуванням потреби в покращенні продуктивності реалізації та захищеності від атак типу перехоплення сеансу, пропонується метод автентифікації, заснований на використанні генераторів псевдовипадкових бінарних послідовностей (ГПВП).

Переваги такого підходу включають:

- високу обчислювальну ефективність завдяки лаконічності архітектури генераторів у порівнянні з традиційними

|     |      |          |        |      |                    |      |
|-----|------|----------|--------|------|--------------------|------|
|     |      |          |        |      | ІАЛЦ.467200.003 ПЗ | Арк. |
|     |      |          |        |      |                    | 31   |
| Зм. | Арк. | № докум. | Підпис | Дата |                    |      |

криптографічними механізмами, такими як блочні шифри чи хеш-функції;

- можливість реалізації механізму взаємної автентифікації, що забезпечує користувачеві впевненість у справжності системи;
- підтримку механізмів вторинної ідентифікації без значних обчислювальних витрат, що важливо для підтримки безпеки під час довготривалих сесій;
- універсальність застосування: генератори можуть слугувати як основа схеми автентифікації за принципом "нульових знань", так і як джерело ключів у потоковому шифруванні.

У запропонованій схемі кожна сторона - як клієнт, так і серверна система - має у своєму розпорядженні власний ГПВП. Типова структура генератора включає регістр пам'яті для поточного стану  $Q$  та логічну схему, що формує вихідний біт і змінює стан генератора відповідно до внутрішньої логіки. Значення  $Q$  має довжину  $m$  біт і повністю визначає поведінку генератора в наступні такти.

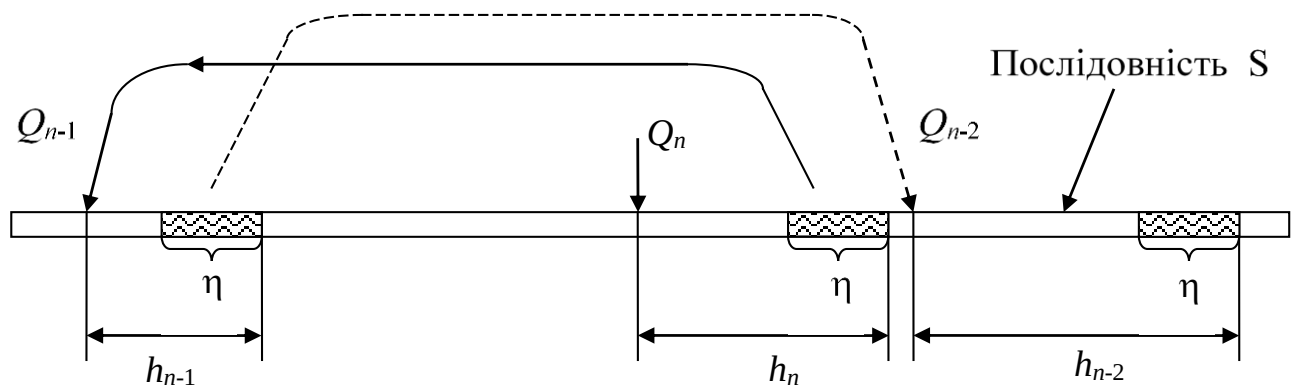


Рис.2.3. Приклад побудови послідовності паролів  $Q_n$ ,  $Q_{n-1}$ ,  $Q_{n-2}$  відповідно до алгоритму ініціалізації

Реалізація базового протоколу автентифікації містить дві основні фази: етап ініціалізації користувача та процедури первинної перевірки особи на початку кожної сесії. На першому етапі створюється набір унікальних паролів

для кожної окремої сесії, які зберігаються у внутрішній пам'яті користувача в оберненому порядку формування.

Алгоритм ініціалізації виглядає наступним чином:

1. Визначається кількість сесій  $n$ . Генерується випадковий початковий вектор  $Q_n$ . Встановлюється лічильник  $j = n$ .
2. Для кожного значення  $j$  генерується випадкова кількість кроків  $h_j$ , після виконання яких отримується новий стан  $Q_{j-1}$ . Сеансовий пароль має вигляд:  $P_j = \langle Q_j, h_j \rangle$ .
3. Значення  $j$  зменшується на одиницю. Якщо  $j > 0$ , повертаємося до кроку 2.
4. Початковий стан  $Q_0$  передається системі і зберігається як значення ідентифікатора  $D$ .

На етапі автентифікації під час кожного нового сеансу:

1. Користувач надає системі пару  $\langle Q_j, h_j \rangle$ .
2. Система ініціалізує генератор  $Q_j$  і виконує  $h_j$  кроків генерації псевдовипадкових бітів.
3. Якщо отриманий стан  $Q_a$  збігається з збереженим  $D$ , автентифікація вважається успішною, після чого  $D$  оновлюється на  $Q_j$ .

Код ініціалізації ГПВДП на  $i$ -тому сеансі

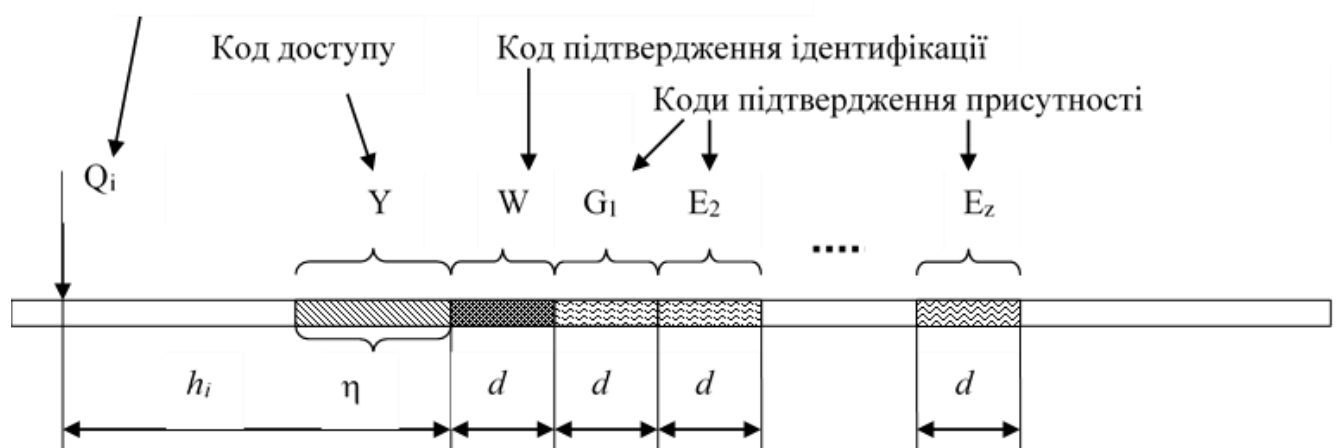


Рис.2.4. Структура псевдовипадкової бітової послідовності, що використовується для автентифікації в межах і-го сеансу обміну.

З метою недопущення атак із захопленням сесії впроваджується процедура підтвердження присутності, яка здійснюється періодично в ході активної сесії. Цей процес не потребує повної переініціалізації ГПВП та забезпечує двостороннє підтвердження безперервного з'єднання.

Послідовність дій виглядає наступним чином:

1. Користувач генерує два  $k$ -бітові коди  $G1$  та  $E1$ , при цьому  $E1$  надсилається до системи.
2. Система генерує послідовність довжиною  $2k$  бітів, перші  $k$  з яких утворюють  $G1'$ , а решта -  $E1'$ .
3. Якщо  $E1$  і  $E1'$  збігаються, система верифікує користувача і надсилає йому  $G1'$ .
4. Користувач перевіряє збіг отриманого  $G1'$  зі збереженим  $G1$ . У разі відповідності - автентичність системи підтверджено.

Таким чином реалізується механізм автентифікації за концепцією "нульових знань": жодна зі сторін не розкриває секретні дані, але має змогу достовірно верифікувати іншу. Завдяки одноразовим сеансовим паролем, сформованим генератором псевдовипадкових послідовностей, злоумисник не може повторити автентифікацію навіть за повного доступу до каналу зв'язку. Це забезпечує стійкість до атак повторного відтворення та підвищує загальний рівень безпеки.

Надійність такого рішення базується на властивості незворотності ГПВП, яка пов'язана із неможливістю ефективного розв'язання системи нелінійних булевих рівнянь. Оцінка обчислювальної складності такої задачі зростає експоненційно і наближається до  $2^{k-1}$ , що значно перевищує потужність навіть найсучасніших обчислювальних засобів.

Завдяки описаному підходу забезпечується надійний захист персональних даних, стійкість до активних атак, а також гнучкість реалізації в різних

|     |      |          |        |      |                    |      |
|-----|------|----------|--------|------|--------------------|------|
|     |      |          |        |      | ІАЛЦ.467200.003 ПЗ | Арк. |
|     |      |          |        |      |                    | 34   |
| Зм. | Арк. | № докум. | Підпис | Дата |                    |      |

цифрових середовищах - від мобільних пристроїв до розподілених інформаційних систем корпоративного рівня.

### 2.3 Оцінка ефективності

Одна з ключових переваг нової схеми ідентифікації, розробленої в рамках концепції «нульового розголошення», полягає у значному скороченні часу, необхідного для виконання обчислювальних процедур, що супроводжують автентифікацію користувача. Традиційні підходи, описані в роботах [23,24], спираються на побудову ланцюгів сеансових паролів із застосуванням поширених хеш-функцій або блочних шифрів. Проте ці методи мають високу обчислювальну складність, що обмежує їхню ефективність при реалізації у пристроях з обмеженими ресурсами.

Розглянемо, для прикладу, найбільш використовуване на практиці хеш-перетворення - алгоритм SHA-256. Його реалізація передбачає 64 раунди, кожен з яких включає 6 зсувів, 15 операцій додавання та понад десяток логічних операцій, що у сумі дає приблизно 2048 процесорних операцій на одне хешування. У випадку блочного шифру AES (наприклад, у конфігурації з 10 раундами), кожен раунд включає операції XOR з ключем, байтову заміну, зсув рядків і мікшування стовпців, у результаті чого загальна кількість операцій становить приблизно 310 [25][27].

У той же час, застосування генераторів псевдовипадкових двійкових послідовностей (ГПВДП), зокрема табличних генераторів на основі LFSR (linear feedback shift register), дозволяє знизити обчислювальне навантаження. Наприклад, генератор із 12 регістрами зсуву та сумарною довжиною 210 біт вимагає лише 25 операцій для генерації одного біта, що дає 5250 операцій для 210 бітів. Інша конфігурація - з використанням одного 128-бітного LFSR - дозволяє скоротити цей показник до 640 операцій, що демонструє високу продуктивність подібних схем.

|     |      |          |        |      |                    |      |
|-----|------|----------|--------|------|--------------------|------|
|     |      |          |        |      | ІАЛЦ.467200.003 ПЗ | Арк. |
|     |      |          |        |      |                    | 35   |
| Зм. | Арк. | № докум. | Підпис | Дата |                    |      |

Під час апаратної реалізації на криптопроцесорі 7955 хеш-функція працює з пропускнуою здатністю до 325 Мбіт/с. Це означає, що на обробку 128 бітів потрібно лише  $1.6 \times 10^{-6}$  секунд. Вбудований генератор псевдовипадкових послідовностей цього процесора генерує 1 біт за 28 наносекунд, тобто 128 бітів - за  $3.5 \times 10^{-6}$  секунд. Такі характеристики є надзвичайно привабливими для створення схем, що потребують швидкої генерації даних для сеансових паролів.

Надійність криптографічної системи базується також на стійкості до атак типу перебору. Припустимо, що зломисник має доступ до всієї інформації користувача, яка зберігається в системі. Якщо генератор працює на основі лінійної функції зворотного зв'язку, то знання початкового стану дозволяє передбачити всю наступну послідовність. Але навіть у такому випадку зломисник не має доступу до поточної конфігурації генератора, а кількість можливих варіантів станів зростає експоненційно зі збільшенням розрядності.

Значну захищеність забезпечує також структура сеансового пароля, який складається з двох частин: відкритої компоненти  $q$  та прихованої  $U$ . Їхня взаємозалежність через хеш-функцію унеможливорює зворотне відновлення без виконання повного перебору. Для алгоритму SHA-1, наприклад, кількість необхідних операцій для інверсії хешу сягає  $2^{159}$ . Відповідно, навіть за наявності попереднього пароля відновити наступний надзвичайно складно.

Важливо також, що запропонована схема дозволяє реалізувати так звану вторинну ідентифікацію під час активного сеансу взаємодії, що неможливо у більшості традиційних схем. Це забезпечує стійкість до атак типу «відтиснення» користувача після первинного входу в систему. Фактично, такий механізм дозволяє системі постійно перевіряти, чи справжній користувач досі на зв'язку.

На додачу до цього, використання логічних операцій у реалізації шифрування та хешування робить цю систему надзвичайно зручною для реалізації на програмованих логічних інтегральних схемах (ПЛІС), що підвищує як продуктивність, так і безпеку. У порівнянні з методами, що

|     |      |          |        |      |                    |      |
|-----|------|----------|--------|------|--------------------|------|
|     |      |          |        |      | ІАЛЦ.467200.003 ПЗ | Арк. |
|     |      |          |        |      |                    | 36   |
| Зм. | Арк. | № докум. | Підпис | Дата |                    |      |

базуються на модульній арифметиці великої розрядності (наприклад, у схемах Fiat-Shamir або Schnorr), нова система вимагає на порядок менше обчислень і відповідно забезпечує на порядок більшу швидкодію.

Таким чином, запропонована схема криптографічно строгої ідентифікації на основі генераторів ПВП у поєднанні з концепцією «нульового розголошення» дозволяє одночасно досягти високого рівня захищеності, знизити обчислювальне навантаження та забезпечити можливість апаратної реалізації, що робить її оптимальним рішенням для сучасних систем автентифікації з віддаленим доступом.

Окрім суто технічних переваг, запропонована схема має і прикладні аспекти, які забезпечують її універсальність. Наприклад, вона може бути легко адаптована до різних архітектурних платформ, включаючи мобільні пристрої, вбудовані системи або серверні інфраструктури. Завдяки малій кількості обчислювальних операцій і відсутності потреби в складних числових обчисленнях, така ідентифікація може ефективно функціонувати навіть у середовищах із обмеженими ресурсами.

Аналіз безпеки також показує, що на відміну від традиційних підходів, які можуть бути чутливими до атак повторного використання або атак типу «man-in-the-middle», система, що ґрунтується на властивостях непередбачуваності псевдовипадкових послідовностей, не дозволяє зломиснику відтворити жоден з компонентів пароля без повного знання початкових параметрів, що є практично недосяжним [21].

Загалом, стійкість методу можна формалізувати через ймовірнісну оцінку успішного відновлення компонент пароля при повному або частковому знанні попередніх сеансів. Якщо навіть зломисник перехопить декілька паролів поспіль, у нього не буде достатньо інформації для побудови наступного пароля без обчислення зворотного хешу та дешифрування за невідомим ключем, що, в умовах використання сучасних криптографічних стандартів, є практично неможливим.

|     |      |          |        |      |                    |      |
|-----|------|----------|--------|------|--------------------|------|
|     |      |          |        |      | ІАЛЦ.467200.003 ПЗ | Арк. |
|     |      |          |        |      |                    | 37   |
| Зм. | Арк. | № докум. | Підпис | Дата |                    |      |

Ще однією важливою перевагою є масштабованість запропонованого підходу. В міру зростання складності обчислень або рівня загроз можливо легко змінювати розрядність ключів, довжину регістрів LFSR або обирати більш складні булеві функції. Це дозволяє підтримувати високий рівень безпеки без радикальних змін в архітектурі системи.

Враховуючи наведене, запропонований метод відкриватиме нові можливості у побудові адаптивних систем безпеки, які поєднують у собі високу продуктивність, надійність, криптостійкість та простоту апаратної реалізації. В умовах зростання загроз у цифровому середовищі такі властивості набувають ключового значення для захисту конфіденційних даних у процесах віддаленої ідентифікації.

Крім наведених технічних характеристик, запропонована система може бути успішно інтегрована у системи багатфакторної автентифікації, де поряд з паролями або біометричними даними виступає як незалежний механізм підтвердження особи. Це дозволяє посилити загальний рівень безпеки без втрати продуктивності.

Щодо апаратної реалізації, важливо зазначити, що використання регістрів зсуву з лінійним зворотним зв'язком не лише знижує енергоспоживання, а й забезпечує високу швидкодію. Наприклад, реалізація LFSR на ПЛІС дозволяє досягати пропускну здатності понад 1 Гбіт/с, що робить цю технологію придатною навіть для високошвидкісних мереж.

У контексті кіберзагроз, які постійно еволюціонують, зокрема таких як фішинг, атаки зі сторони «внутрішнього» користувача або ін'єкція шкідливого коду, запропонована система демонструє високу стійкість до цих типів загроз завдяки ізолюваності ідентифікаційних даних та їхній мінливості. Навіть якщо попередній пароль буде скомпрометовано, наступний неможливо вирахувати без повного перебору.

Ці переваги роблять запропонований метод перспективним не лише для застосування у комерційних інформаційних системах, а й у критичних

|     |      |          |        |      |                    |      |
|-----|------|----------|--------|------|--------------------|------|
|     |      |          |        |      | ІАЛЦ.467200.003 ПЗ | Арк. |
|     |      |          |        |      |                    | 38   |
| Зм. | Арк. | № докум. | Підпис | Дата |                    |      |

інфраструктурах - таких як телекомунікації, енергетика, системи управління доступом до державних реєстрів, де необхідна максимальна надійність автентифікації.

Зважаючи на вказане, можна підсумувати, що розроблений підхід до криптографічно строгої ідентифікації не лише забезпечує фундаментальну безпеку на рівні протоколу, але й створює платформу для гнучкого масштабування, інтеграції в інші криптографічні інструменти та гармонійного поєднання із сучасними стандартами інформаційної безпеки.

У перспективі можливе вдосконалення даної схеми шляхом впровадження квантово-стійких компонентів, що дозволить ще більше посилити її захищеність у світлі майбутніх викликів, пов'язаних з розвитком квантових обчислювальних систем. Також доцільно розглядати створення модульних рішень на її основі, які можуть бути адаптовані до специфіки кожного середовища застосування.

Ще один напрямок розвитку системи - розробка механізмів динамічного регулювання складності генератора у відповідь на виявлення потенційної загрози. Наприклад, у разі виявлення аномальної активності в каналі зв'язку, система може автоматично збільшувати кількість регістрів зсуву або змінювати параметри хеш-функції. Це створює додатковий бар'єр для атак в реальному часі.

Формально, генерація сеансового пароля описується наступною системою рівнянь:

$$q_i = H(q_{i+1})$$

$$U_{i+1} = \varphi^{-1}(R, K_{i+1}),$$

де  $H$  - хеш-функція,  $\varphi^{-1}$  - обернене функціональне перетворення,  $R$  - відомий системі параметр,  $K_{i+1}$  - ключ, що змінюється в кожному сеансі. Така формалізація дозволяє проводити формальний аналіз криптостійкості та моделювання сценаріїв атаки.

|     |      |          |        |      |                    |      |
|-----|------|----------|--------|------|--------------------|------|
|     |      |          |        |      | ІАЛЦ.467200.003 ПЗ | Арк. |
|     |      |          |        |      |                    | 39   |
| Зм. | Арк. | № докум. | Підпис | Дата |                    |      |

Аналізуючи ентропійні характеристики псевдовипадкових послідовностей, можна стверджувати, що рівень ентропії на виході генератора LFSR з правильною конфігурацією наближається до максимальної, що є ознакою високої криптографічної якості. При цьому важливо забезпечити, щоб початкові стани регістрів не повторювалися, тобто були гарантовано випадковими.

У загальному вигляді, складність підбору компонент сеансового пароля зловмисником можна оцінити як:

$$C_{\text{total}} = 2^{h-1} + 2^{(m+d)} \approx 2^{h-1} + 2^{(2m-h)},$$

де  $h$  - розрядність хеш-коду,  $m$  - розрядність відкритої частини пароля,  $d$  - довжина ключа. Така комбінована складність показує, що навіть за умов зламу одного компонента (наприклад,  $q_i$ ), зловмисник не має прямого доступу до  $U_i$  без повного перебору.

Таким чином, запропонована методологія забезпечує не лише криптографічну надійність на фундаментальному рівні, а й гнучкість, масштабованість і адаптивність - як у контексті технічної реалізації, так і у сфері її застосування. В умовах швидкозмінного цифрового середовища така система стає однією з ключових для реалізації безпечної взаємодії на відстані.

## ВИСНОВОК ДО РОЗДІЛУ 2

У другому розділі дипломної роботи здійснено комплексне дослідження методологічних та технічних засад створення механізму криптографічно строгої ідентифікації, що ґрунтується на концепції "нульових знань" і передбачає застосування генераторів псевдовипадкових бітових послідовностей. За підсумками проведеної роботи отримано наступні результати:

1. На теоретичному рівні обґрунтовано вибір незворотного криптографічного перетворення для побудови ланцюга сеансових паролів. У якості основи для цього перетворення обрано генератори псевдовипадкових бітових послідовностей (ГПВДП), що демонструють високу продуктивність та криптостійкість при реалізації в реальному часі. Порівнюючи з традиційними хеш-функціями та шифрувальними блоками, ГПВДП забезпечують істотне скорочення обчислювальних витрат і дозволяють інтегрувати функціонал як первинної, так і вторинної ідентифікації в єдину схему, зберігаючи властивість незворотності.
2. Розроблено метод ідентифікації віддаленого користувача, який відповідає концептуальним принципам доказів з нульовим розголошенням знань. Особливістю методу є реалізація багаторазового підтвердження факту присутності користувача у процесі активного сеансу, що досягається шляхом застосування спеціальної криптографічної процедури – вторинної взаємної автентифікації. Метод базується на непередбачуваній еволюції внутрішнього стану генераторів ПВДП, що забезпечує відсутність можливості інверсії або підміни автентичного користувача.
3. Сформовано технологічну модель криптографічно строгої автентифікації, яка демонструє високу швидкодію та стійкість до атак типу "man-in-the-middle", а також перехоплення сеансів. Завдяки

|     |      |          |        |      |                    |      |
|-----|------|----------|--------|------|--------------------|------|
|     |      |          |        |      | ІАЛЦ.467200.003 ПЗ | Арк. |
|     |      |          |        |      |                    | 41   |
| Зм. | Арк. | № докум. | Підпис | Дата |                    |      |

використанню генераторів з апаратною ефективністю та механізму постійного підтвердження зв'язку між користувачем і системою, було досягнуто посилення безпеки при віддаленій взаємодії. Крім того, метод може бути масштабовано інтегрований у схеми потокового шифрування, тим самим забезпечуючи одночасну автентифікацію та конфіденційність переданих даних.

Отже, результати розробки та аналізу свідчать про доцільність впровадження описаного підходу у системи інформаційної безпеки з вимогами до високої надійності і низької затримки. Запропонований метод дозволяє реалізувати ефективний захист інформаційної взаємодії з гарантією автентичності та захистом від перехоплення в рамках єдиного криптографічного протоколу, узгодженого з принципами нульових знань.

|     |      |          |        |      |                    |      |
|-----|------|----------|--------|------|--------------------|------|
|     |      |          |        |      | ІАЛЦ.467200.003 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата |                    | 42   |

# РОЗДІЛ 3

## РОЗРОБКА ПРОГРАМИ ДЛЯ ДОСЛІДЖЕННЯ

### ПВП

У межах даного розділу розглядається практична частина дослідження, що полягає в реалізації програмного засобу для моделювання процесів криптографічно строгої ідентифікації на основі генераторів псевдовипадкових послідовностей. Розробка програмного забезпечення є невіддільною складовою запропонованого методу, оскільки дозволяє в повному обсязі перевірити його працездатність, адаптивність до різних початкових умов та ефективність обчислювальної реалізації.

Програмний код створено з урахуванням принципів модульності, розширюваності та супроводжуваності. В основі реалізації лежить концепція зсувного регістру з лінійним зворотним зв'язком (LFSR), який забезпечує генерацію псевдовипадкових бітів. Для підвищення криптографічної стійкості та складності передбачення результату, використано комбінацію декількох незалежно ініціалізованих регістрів. Такий підхід дозволяє сформувати генератор псевдовипадкових чисел, що відповідатиме вимогам сьогодення до безпеки в цифрових системах.

Реалізоване програмне рішення дозволяє не лише емулювати поведінку окремих LFSR-компонентів, але й забезпечує їх координацію для отримання результатів у вигляді адрес доступу до масиву ключових значень. Вихідні біти, які формуються у процесі генерації, використовуються як основа для визначення індексу, за яким зчитується псевдовипадкове значення - імітуючи тим самим механізм автентифікації або шифрування, де рішення залежить від внутрішнього стану криптографічного ядра.

Розроблена система має універсальний характер і може бути використана як для подальших досліджень у сфері криптографічного аналізу, так і як основа для впровадження методів безпечної ідентифікації в прикладні ІТ-рішення. У

|     |      |          |        |      |                    |      |
|-----|------|----------|--------|------|--------------------|------|
|     |      |          |        |      | ІАЛЦ.467200.003 ПЗ | Арк. |
|     |      |          |        |      |                    | 43   |
| Зм. | Арк. | № докум. | Підпис | Дата |                    |      |

наступних підрозділах буде представлено детальний опис реалізованих класів, алгоритмів, їх функціональних обов'язків, а також фрагменти коду, що ілюструють основні етапи роботи генератора псевдовипадкових послідовностей.

### 3.1 Розробка компонентів програми

Розроблене програмне забезпечення реалізоване на мові програмування C++ із застосуванням лише стандартних можливостей, без використання сторонніх бібліотек або фреймворків. Такий підхід забезпечує повну контрольованість усіх етапів виконання програми, а також дає змогу досягти високої прозорості та передбачуваності поведінки коду. Уся логіка функціонування системи побудована на використанні користувацьких структур, функцій та класів, що були створені спеціально під вимоги поставленої задачі.

Архітектура програмної системи має модульну будову, що передбачає чіткий поділ функціональності між окремими компонентами. Основу системи становить реалізація зсувного регістру з лінійним зворотним зв'язком (LFSR), який відповідає за генерацію псевдовипадкових бітів. Крім того, сформовано набір незалежних генераторів на базі LFSR-модулів, об'єднаних у масив для забезпечення комбінованої генерації, що підвищує криптографічну складність вихідної послідовності.

Особливу роль відіграє головна функція `main()`, яка ініціалізує всі компоненти, задає початкові умови, координує обчислювальні процеси та виконує підсумковий вивід результатів у консоль. Завдяки цьому вона виконує роль центрального елемента, що узгоджує взаємодію між генераторами, ключовим масивом і алгоритмами генерації бітів.

У наступних підрозділах буде здійснено поетапний розбір усіх складових програмної системи. Зокрема, буде представлено детальний опис структури та логіки роботи таких основних компонентів:

- структур та класів, що реалізують LFSR-ядро;
- масиву генераторів, який використовується для комбінованої обробки даних;
- допоміжних змінних, що відповідають за зберігання проміжних і кінцевих результатів;
- функціональних блоків генерації та вибору випадкових бітів;
- а також головної функції, яка формує логіку запуску та завершення сеансу генерації.

Для кожного компоненту буде надано технічну специфікацію, у якій детально описуються внутрішні змінні, методи, а також їхнє функціональне призначення у контексті загальної системи. Окрім текстового опису, будуть додані фрагменти коду у вигляді зображень для кращої наочності та зручності сприйняття технічного змісту.

### 3.1.1 Клас InitialStateConfig

Клас InitialStateConfig призначений для збереження конфігурації початкових станів (seed) і маски зворотного зв'язку (tap mask) для кожного окремого LFSR у генераторі псевдовипадкових послідовностей. Він забезпечує централізоване зберігання параметрів конфігурації, що дозволяє гнучко налаштовувати роботу системи генераторів.

Як основа у даному класі наявний конструктор, який ініціалізує об'єкт з переданими початковими значеннями та маскою, зокрема:

- Зберігає в поля seedBits і tapMask відповідні вхідні вектори.
- Обчислює registerLength як seed.size().

|     |      |          |        |      |                    |      |
|-----|------|----------|--------|------|--------------------|------|
|     |      |          |        |      | ІАЛЦ.467200.003 ПЗ | Арк. |
|     |      |          |        |      |                    | 45   |
| Зм. | Арк. | № докум. | Підпис | Дата |                    |      |

```
InitialStateConfig(std::vector<int> seed, std::vector<int> mask)
: seedBits(seed), tapMask(mask), registerLength(seed.size()) {}
```

Рис.3.1. Реалізація конструктора класу InitialStateConfig

Переглянемо решту наявних методів у класі:

- void setSeed змінює значення початкового стану на нове.
- void setMask змінює маску зворотного зв'язку.
- getSeed() повертає поточний вектор - seedBits.
- getMask() повертає поточну маску - tapMask.
- getLength() повертає довжину регістру - registerLength.

```
void setSeed(const std::vector<int>& newSeed) {
    seedBits = newSeed;
    registerLength = newSeed.size();
}

void setMask(const std::vector<int>& newMask) {
    tapMask = newMask;
}

std::vector<int> getSeed() const { return seedBits; }
std::vector<int> getMask() const { return tapMask; }
int getLength() const { return registerLength; }
```

Рис.3.2. Реалізація методів класу InitialStateConfig

Окрім того у класі було ініціалізовано наступні атрибути кожне з яких має своє призначення у процесах класу, зокрема:

- seedBits зберігає початкові значення регістру зсуву (seed), які визначають, з якого стану починається генерація.
- tapMask є бітовою маскою, що визначає, які біти беруть участь в обчисленні нового біту регістру (через XOR).
- registerLength це довжина регістру зсуву, яка визначається автоматично на основі довжини seedBits.

Клас `InitialStateConfig` є допоміжним конфігураційним класом для генераторів типу `LFSRUnit` або `BitStreamGenerator`. Він дозволяє централізовано визначати вхідні дані та гарантує відокремлення логіки конфігурації від логіки генерації.

### 3.1.2 Клас `LFSRUnit`

Клас `LFSRUnit` моделює поведінку одного LFSR-блоку. Цей блок зберігає власний стан, має набір позицій, які використовуються для обчислення нового біту зворотного зв'язку, та виконує зсув зі вставкою нового біту в кінець послідовності. Його основна роль - генерування бітів псевдовипадкової послідовності.

Розглянемо конструктор класу. Під час виклику він виконує наступні дії:

- ініціалізує `registerState` початковим значенням.
- ініціалізує `feedbackPositions` (тобто які біти зсумовуються за XOR для нового біта).
- встановлює `registerLength`.

```
public:
    LFSRUnit(const std::vector<int>& seed, const std::vector<int>& mask)
        : state(seed), feedbackMask(mask) {}
```

Рис.3.3. Реалізація конструктора класу `LFSRUnit`

Також у класі наявний ще один метод `int shift()`. Він призначений виконанню зсуву бітів, з якого можна виділити наступні кроки:

1. виконання однієї операції зсуву в LFSR;
2. повернення вихідного біту (найстарший біт до зсуву);
3. обчислення нового біту через XOR елементів з `feedbackPositions`;
4. вставлення нового біту в кінець.

```
// Метод shift() здійснює зсув регістру з використанням XOR
int shift() {
    int newBit = 0;
    for (size_t i = 0; i < state.size(); ++i) {
        if (feedbackMask[i]) newBit ^= state[i];
    }

    int outputBit = state[0];
    state.erase(state.begin());
    state.push_back(newBit);
    return outputBit;
}
```

Рис.3.4. Реалізація методу shift()

Окрім того в класі можна виділити наступні атрибути:

- registerState - це вектор, що представляє двійковий код стану. Наприклад, для 4-бітного регістру: [1, 0, 1, 1].
- feedbackPositions - позиції, на основі яких обчислюється новий біт (через XOR).
- registerLength - довжина регістру, визначає кількість кроків та пам'яті.

Багато екземплярів LFSRUnit можуть використовуватись в CompositeLFSRGenerator, який зчитує біти з кількох LFSRUnit, щоб створити складну адресу для доступу до масиву ключових значень або для генерації складніших послідовностей.

### 3.1.3 Клас BitStreamGenerator

BitStreamGenerator — це клас-агрегатор, що поєднує декілька об'єктів LFSRUnit та формує на їх основі адресу, за якою вибирається біт із таблиці ключових значень (keyTable). Це і є результат генерації одного псевдовипадкового біта. Він реалізує композиційний принцип, у якому кожен LFSR виконує зсув і впливає на фінальний результат.

Конструктор цього класу виконує наступні дії:

|     |      |          |        |      |                    |      |
|-----|------|----------|--------|------|--------------------|------|
|     |      |          |        |      | ІАЛЦ.467200.003 ПЗ | Арк. |
|     |      |          |        |      |                    | 48   |
| Зм. | Арк. | № докум. | Підпис | Дата |                    |      |

- Ініціалізує внутрішній вектор `lfsrUnits` переданими LFSR-екземплярами.
- Зберігає таблицю `keyTable`.
- Визначає `addressBits` як кількість переданих LFSR.

```
public:
    BitStreamGenerator(const std::vector<InitialStateConfig>& configs, const std::vector<int>& keyTable) {
        for (const auto& cfg : configs) {
            lfsrUnits.emplace_back(cfg.getSeed(), cfg.getMask());
        }
        lookupTable = keyTable;
    }
```

Рис.3.5. Реалізація конструктора класу BitStreamGenerator

Окрім того у класі наявний метод `int generate()` який виконує основну роль у цьому класі:

1. Генерує один псевдовипадковий біт, використовуючи LFSR-блоки для побудови адреси.
2. Кожен LFSRUnit виконує `shift()`, а його результат стає одним із бітів адреси.
3. Після побудови адреси - витягується значення з `keyTable` за цим індексом.

```
// Метод generate() виконує генерацію одного біта на основі адреси
int generate() {
    int address = 0;
    for (size_t i = 0; i < lfsrUnits.size(); ++i) {
        address |= (lfsrUnits[i].shift() << i);
    }
    return lookupTable[address % lookupTable.size()];
}
```

Рис.3.6. Реалізація методу generate()

До того ж у класі наявні наступні атрибути:

- Вектор `lfsrUnits`, який містить усі регістри LFSR (одиниці типу `LFSRUnit`), що використовуються генератором.
- `keyTable` - масив із  $2^n$  елементів (наприклад, 4096), у якому кожен елемент має значення 0 або 1.
- `addressBits` - кількість бітів, які використовуються для побудови адреси (визначається кількістю LFSR або фіксовано, наприклад, 12).

BitStreamGenerator є центральною точкою криптографічно строгої ідентифікації, адже на його основі:

- будується сеансовий пароль;
- реалізується перевірка відповідності стану генератора;
- створюються ідентифікатори у схемах «нульових знань»;
- забезпечується непередбачуваність та стійкість до атак.

### 3.1.4 Виконавчий метод `main()`

Метод `main()` є головною функцією запуску у C++ програмі. У цьому методі наявні наступні кроки виконання:

1. `const int lfsrCount` - визначає кількість незалежних LFSR-блоків, що будуть використовуватись у генераторі. Це число визначає також довжину бітової адреси (логарифмічно — довжину `keyTable`).
2. Створення `InitialStateConfig` для кожного LFSR, тобто ініціалізується конфігурація для кожного регістра зсуву із відповідно вказаними параметрами.
3. Формування `keyTable` (таблиці псевдовипадкових значень)
4. Створення та запуск `LFSRUnit`. Для кожної конфігурації створюється об'єкт `LFSRUnit`. Ці об'єкти моделюють реальні регістри зсуву з заданими початковими станами та позиціями XOR.
5. Ініціалізація головного генератора `BitStreamGenerator` - створюється генератор, який агрегує всі LFSR та використовує їх вихід для формування адреси, за якою читається біт із `keyTable`.
6. Виклик методу `generateBit()`, метод викликає `.shift()` для кожного LFSR, формує бітову адресу та повертає відповідне значення з таблиці.
7. Виведення результату

```
// =====
// Головна функція: створює LFSR, ініціалізує ключі, виконує генерацію біта
// =====
int main() {
    // Підготовка таблиці ключів: чергування 0 і 1
    std::vector<int> keyTable(4096);
    for (int i = 0; i < 4096; ++i) {
        keyTable[i] = i % 2;
    }

    // Формування початкових конфігурацій LFSR
    std::vector<InitialStateConfig> configs;
    for (int i = 0; i < 12; ++i) {
        configs.emplace_back(std::vector<int>{0, 1, 0, 1}, std::vector<int>{1, 0, 0, 1});
    }

    // Ініціалізація генератора та отримання результату
    BitStreamGenerator generator(configs, keyTable);
    int result = generator.generate();

    // Виведення результату генерації
    std::cout << "Згенерований біт: " << result << std::endl;
    return 0;
}
```

Рис.3.7. Реалізація методу main()

## 3.2 Розробка тестового модулю

Одним із ключових аспектів якісної розробки програмного забезпечення є організація процесу тестування, що дозволяє виявити та усунути потенційні помилки ще на етапі розробки. Для забезпечення стабільності, передбачуваності та правильності роботи програмної системи, що реалізує генерацію псевдовипадкових бітових послідовностей базуючись на зсувних регістрах з лінійним зворотним зв'язком (LFSR), було реалізовано спеціалізований тестовий модуль.

Головною метою створення цього модуля є автоматизована перевірка функціональної коректності реалізованих алгоритмів, а також виявлення потенційно вразливих або нестабільних фрагментів коду. Основна увага приділяється перевірці роботи ключових компонентів системи, таких як:

- правильність функціонування окремого регістру LFSR, зокрема процесу зсуву та генерації нового біту на основі операції XOR;
- валідація коректності формування адреси доступу в генераторі псевдовипадкових чисел (BitStreamGenerator);
- перевірка точності вибору значення з таблиці ключів (keyArray) за згенерованою адресою;
- узгодженість вихідного результату з очікуваними значеннями в рамках тестових сценаріїв.

Для досягнення цих цілей у рамках модуля реалізовано набір юніт-тестів, кожен з яких відповідає за перевірку окремого елементу логіки системи. Усі тести структуровані за принципом модульного тестування: кожна функція або клас перевіряється незалежно, з використанням контрольованих вхідних даних і перевіркою очікуваного виходу.

Тестовий модуль включає такі основні тести:

1. Тест ініціалізації LFSR: перевіряє, чи правильно зберігаються початкові значення регістру та індекси бітів, що беруть участь в обчисленні нового біта.

```
void testLFSRInitialization() {
    std::vector<int> seed = {1, 0, 1, 1};
    std::vector<int> mask = {1, 0, 1, 0};
    LFSRUnit lfsr(seed, mask);
    std::cout << "testLFSRInitialization passed." << std::endl;
}
```

Рис.3.8. Реалізація юніт-тесту testLFSRInitialization()

2. Тест зсуву регістру: перевіряє правильність алгоритму зсуву та додавання нового біта згідно з алгоритмом лінійного зворотного зв'язку.

```

void testLFSRShift() {
    std::vector<int> seed = {1, 0, 0, 1};
    std::vector<int> mask = {1, 1, 0, 0};
    LFSRUnit lfsr(seed, mask);
    int output = lfsr.shift();
    assert((output == 1 || output == 0) && "Shift output must be binary.");
    std::cout << "testLFSRShift passed." << std::endl;
}

```

Рис.3.9. Реалізація юніт-тесту testLFSRShift()

3. Тест генерації адреси: перевіряє, чи правильно агрегуються біти з усіх регістрів для формування індексу доступу до ключового масиву.

```

void testAddressGeneration() {
    std::vector<InitialStateConfig> configs;
    for (int i = 0; i < 4; ++i) {
        configs.emplace_back(std::vector<int>{1, 0, 1, 1}, std::vector<int>{1, 0, 0, 1});
    }
    std::vector<int> keyTable(16, 1);
    BitStreamGenerator generator(configs, keyTable);
    int result = generator.generate();
    assert(result == 1);
    std::cout << "testAddressGeneration passed." << std::endl;
}

```

Рис.3.10. Реалізація юніт-тесту testAddressGeneration()

4. Тест вибору ключового значення: перевіряє, чи коректно програма обирає значення з масиву keyArray згідно з обчисленою адресою.

```

void testKeyValueSelection() {
    std::vector<InitialStateConfig> configs;
    for (int i = 0; i < 4; ++i) {
        configs.emplace_back(std::vector<int>{0, 1, 0, 1}, std::vector<int>{1, 1, 0, 0});
    }
    std::vector<int> keyTable(16);
    for (int i = 0; i < 16; ++i) keyTable[i] = i % 2;
    BitStreamGenerator generator(configs, keyTable);
    int result = generator.generate();
    assert(result == 0 || result == 1);
    std::cout << "testKeyValueSelection passed." << std::endl;
}

```

Рис.3.11. Реалізація юніт-тесту testKeyValueSelection()

Усі тести реалізовано з використанням базових можливостей мови C++, без залучення зовнішніх фреймворків, що дозволяє зберегти контроль над кожною фазою виконання перевірки. Логіка тестів викладена у вигляді функцій із заданими вхідними параметрами та фіксованими очікуваними результатами. Протягом виконання програми результати тестів виводяться у консоль, що дає

змогу швидко визначити успішність або помилку в роботі відповідного компонента.

Завдяки впровадженню такого тестового модуля забезпечується можливість завчасного виявлення помилок, що підвищує надійність розробленого програмного забезпечення, а також створюється основа для подальшого розширення та підтримки коду. Тестування є невід'ємною складовою життєвого циклу програмного продукту, особливо у сфері криптографії, де навіть незначна похибка у генерації випадкових значень може призвести до критичних наслідків.

|     |      |          |        |      |                    |      |
|-----|------|----------|--------|------|--------------------|------|
|     |      |          |        |      | ІАЛЦ.467200.003 ПЗ | Арк. |
|     |      |          |        |      |                    | 54   |
| Зм. | Арк. | № докум. | Підпис | Дата |                    |      |

## ВИСНОВОК ДО РОЗДІЛУ 3

У результаті реалізації третього розділу було повністю спроектовано, реалізовано та задокументовано програмну систему, яка моделює функціонування генератора псевдовипадкових бітових послідовностей на основі зсувних регістрів з лінійним зворотним зв'язком (LFSR). Розроблена система стала практичним втіленням теоретичних засад, викладених у другому розділі, зокрема ідей криптографічно строгої ідентифікації, реалізованих через використання LFSR-генераторів.

Зокрема, у цьому розділі:

- Реалізовано програмний код, який повністю відповідає запропонованій архітектурі. Створено окремі класи для моделювання LFSR-логіки (наприклад, LFSRUnit, BitStreamGenerator, InitialStateConfig) та центральний координуючий механізм - головну функцію main(), яка забезпечує запуск і узгоджену взаємодію всіх компонентів.
- Забезпечено унікальність і наочність реалізації: розробка здійснена виключно засобами мови програмування C++ без залучення зовнішніх бібліотек. Це гарантує прозорість роботи алгоритмів і дозволяє гнучко контролювати кожен етап обробки даних.
- Проведено деталізований опис усіх структур і класів, що використовуються системою: включено пояснення логіки методів, їх ролі у загальній схемі генерації псевдовипадкових бітів, опис змінних та логіки обчислень.
- Виконано візуалізацію коду у вигляді фрагментів (зображень), що дозволяє легше аналізувати роботу програми та використовувати її у навчальних або дослідницьких цілях.
- Розроблено модуль автоматизованого тестування, який забезпечує перевірку функціональної коректності основних складових програмного коду. Зокрема, верифікуються: правильність ініціалізації LFSR,

|     |      |          |        |      |                    |      |
|-----|------|----------|--------|------|--------------------|------|
|     |      |          |        |      | ІАЛЦ.467200.003 ПЗ | Арк. |
|     |      |          |        |      |                    | 55   |
| Зм. | Арк. | № докум. | Підпис | Дата |                    |      |

коректність зсуву регістрів, генерація адреси доступу до масиву ключових значень, а також вибірка даних згідно з обчисленою адресою.

- У тестовому модулі реалізовано юніт-тести для всіх критичних ділянок, що дозволяє впевнено оцінити стійкість програми до помилок і підтвердити її працездатність при різних сценаріях використання.

Таким чином, результати, отримані в цьому розділі, не лише підтверджують можливість реалізації криптографічно строгої ідентифікації на основі LFSR, але й демонструють практичну ефективність обраного підходу. Високий рівень деталізації коду, структурна логіка реалізації, відповідність теоретичній моделі та наявність тестів забезпечують надійність і масштабованість створеної системи. Одержані результати можуть бути використані як основа для подальших досліджень та вдосконалення наявних криптографічних протоколів автентифікації з використанням псевдовипадкових послідовностей.

|     |      |          |        |      |                    |      |
|-----|------|----------|--------|------|--------------------|------|
|     |      |          |        |      | ІАЛЦ.467200.003 ПЗ | Арк. |
|     |      |          |        |      |                    | 56   |
| Зм. | Арк. | № докум. | Підпис | Дата |                    |      |

# ВИСНОВКИ

Як результат виконання дипломного проєкту, присвяченого проблемам ідентифікації користувачів у системах віддаленої інформаційної взаємодії, отримано низку важливих практичних та наукових результатів, що підтверджують актуальність та ефективність обраного підходу.

1. Проведено всебічний аналіз сучасних викликів у сфері безпеки віддаленої взаємодії, зокрема загроз, пов'язаних із несанкціонованим втручанням у сеанси зв'язку, підміною користувачів або витісненням легітимного абонента з інформаційного каналу. Показано, що традиційні методи ідентифікації є недостатньо ефективними в умовах динамічного середовища з відкритими каналами зв'язку.
2. Встановлено, що найбільш перспективним підходом до вирішення проблем автентифікації є використання криптографічно строгих методів, заснованих на концепції "нульових знань", які дозволяють перевірити особу користувача без розкриття будь-якої конфіденційної інформації.
3. Проаналізовано існуючі алгоритми, що реалізують дану концепцію, та виявлено їхню обмежену придатність до використання у системах із підвищеними вимогами до швидкодії або вбудованих пристроях з обмеженими ресурсами. Зокрема, методи на основі теорії чисел виявились надто ресурсомісткими.
4. Обґрунтовано доцільність використання незворотних булевих перетворень, які реалізуються на базі генераторів псевдовипадкових бітових послідовностей (ПВП), зокрема LFSR (регістрів зсуву з лінійним зворотним зв'язком). Доведено, що така реалізація дозволяє забезпечити як високу швидкодію, так і стійкість до криптографічного аналізу.
5. Запропоновано ефективний метод криптографічно строгої ідентифікації, який передбачає як первинний етап автентифікації, так і періодичні

|     |      |          |        |      |                    |      |
|-----|------|----------|--------|------|--------------------|------|
|     |      |          |        |      | ІАЛЦ.467200.003 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата |                    | 57   |

вторинні перевірки під час активної сесії. Це дозволяє гарантувати безперервність контролю автентичності користувача і запобігти атакам типу “перехоплення сеансу”.

6. У межах проєкту реалізовано програмне забезпечення, що моделює функціонування LFSR і генераторів на базі них. Програма побудована із застосуванням стандартного функціоналу мови C++, із чітким розподілом логіки по класах, зокрема LFSRUnit, BitStreamGenerator, InitialStateConfig, main() та інших.
7. Описано структуру програмної реалізації, детально розглянуто змінні, методи, їх функціональне призначення та міжкомпонентну взаємодію. Представлено численні фрагменти коду для зручного аналізу та перевірки.
8. Розроблено тестовий модуль, що включає набір юніт-тестів для перевірки критично важливих аспектів функціонування системи: правильність зсуву бітів, генерації адреси доступу до таблиці ключів, вибору значень, ініціалізації регістрів тощо. Це дозволяє підтвердити коректність реалізації та забезпечити стабільність роботи всієї системи.
9. Отримані результати мають як теоретичне, так і прикладне значення. Запропонована система може бути інтегрована в реальні системи автентифікації користувачів у сфері інформаційної безпеки, електронного документообігу, IoT-платформах, системах захищеної комунікації тощо.
10. У перспективі можлива оптимізація реалізованого коду, розширення функціоналу за рахунок підтримки інших типів регістрів або джерел ентропії, а також інтеграція з апаратними криптомодулями для підвищення надійності та масштабованості розробленого рішення.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Chernyshevych M. Zero-knowledge identification of remote users by utilization of pseudorandom sequences / M. Chernyshevych, I. Daiko, V. Selivanov, O. Markovskyi // Information, Computing and Intelligent systems. – 2022. – № 3.- P.42-48. DOI: 10.20535/2708-4930.3.2022
2. Васильєва М.Д.. Метод швидкої ідентифікації віддалених абонентів на основі концепції нульових знань / Васильєва М.Д., Дайко І.В., Саницький А.П. // Наука і техніка сьогодні. 2024.-№ 3(31).-С.791-803. [https://DOI.org/10.52058/2786-6025-2024-3\(31\)-791-804](https://DOI.org/10.52058/2786-6025-2024-3(31)-791-804)
3. Марковський О.П. Метод криптографічно строгої ідентифікації на основі незворотних перетворень на основі табличних перетворень на полях Галуа / О.П. Марковський, І.В. Дайко // Адаптивні системи автоматичного управління.- 2024.- Том.1 - № 44.- С.127-141. DOI: 10.20535/1560-8956.44.2024.302429
4. Schneier B. Applied Cryptography. Protocols. Algorithms and Source codes in C. Ed.John Wiley, 1996 - 758 p.
5. Menezes Alfred, Handbook of Applied Cryptography. / Alfred Menezes, Paul C. van Oorschot, and Scott A. Vanstone // CRC Press. – 2001. – 780 p.
6. Kittichokenai K. Secret Key-based Identification and Authentication with a Privacy Constraint. / Kittichokenai K., Care G. // IEEE Trans. Inf. Theory.- Vol. 62.- 2016.- № 11.- P. 6189-6203.
7. Asimi Y. Strong zero-knowledge authentication based on the session keys (SASK) / Asimi Y., Amghar A., Asimi A., Sadgi Y. // International Journal of Network Security & Its Applications (IJNSA).-2015.- Vol.7, - No.1,- P.51-66.

8. Aquilar C., Gaborit P., Schrec J. A new zero-knowledge code based identification scheme with reduced communication. IEEE Information Theory Workshop 16-20 Oct. 2011 .-2011.-P.648-652. DOI: 10.1109/ITW.2011.6089577
9. Bardis N. Fast subscriber identification based on the zero knowledge principle for multimedia content distribution./ Bardis N., Doukas N., Markovskyi O. // International Journal of Multimedia Intelligence and Security.-2010.- No.4,- P. 363-377.
- 10.Марковський О.П. Метод строгої ідентифікації віддалених абонентів на основі стандартизованих шифроблоків та хеш-перетворень / О.П. Марковський, Захаріудакіс Лефтеріс., М.Ф. Федотов // Вісник Національного технічного університету України “КПІ” Інформатика, управління та обчислювальна техніка. К.: ТОО „ВЕК+”. 2016.- № 64.- С. 161-165.
- 11.Марковський О. П. Використання алгебри полів Галуа для реалізації концепції нульових знань при ідентифікації та автентифікації віддалених користувачів /О.П.Марковський, Захаріудакіс Лефтеріс., В.Р.Максимук // Электронное моделирование. 2017.- № 6.- С.96-110.
- 12.Mu Han, Zhikun Yin, Pengzhou Chen, Xing Zhang, Shidian MA. Zero-knowledge identity authentication for internet of vehicles: Improvement and application. PLoS ONE.- 2020.- Vol.15 –No.9.-P.217-247.
- 13.Guillou L.C., Quisquater J.J. A Practical Zero-Knowledge Protocol Fitted to Security Microprocessor Minimizing Both Transmission and Memore. Proceeding of Advances of Cryptology – Eurocrypt-88.- Springer-Verlag.- 1988.- P.123-128.
- 14.Schnorr C.P. Method for Identification Subscribers and for Generating and Verificating Electronic Signatures in data Exchange System.- US Patent #4995,083.19- 1991.
- 15.Feige U.,Fiat A.,Shamir A. Zero knowledge proofs of identity. Journal of Cryptology, - 1988.- Vol.1.- №.2 . – P.77-94.

- 16.Rafaël Del Pino. Efficient lattice-based zero-knowledge proofs and applications. Cryptography and Security. Université Paris sciences et lettres,-2018.- 110 p.
- 17.Bardis N., Doukas N. Markovskyi O. A Method for strict remote user authentication using non-reversible Galois field transformations // MATEC Web of Conferences 125, 05017 (2017)- P.243-249.
- 18.Bardis N.G., Markovskyi O.P., Doukas N., Drigas F. Fast implementation zero knowledge identification schemes using the Galois Fields arithmetic, Proceeding of IX. International Symposium IEEE on Telecommunications - BIHTEL-2012, October 25-27, 2012, Sarajevo, Bosnia and Herzegovina. – P.54-62.Stavroulakis P., Markovskyi O., Bardis N., Doucas N. Efficient Zero Knowledge identification based on one way Boolean transformations. IEEE Globecom Workshops. 5-9 Dec. 2011.- 2011.-P.275-280. DOI: 10.1109/GLOCOMW.2011.6162452.
- 19.Lamport L.Password Authentication with Insecure Communication. Communications of the ACM. -1981.- Vol.24.- № 11,- P.770-772.
- 20.Haller N., Atkinson R. On Internet Authentication. RFC 1704, Bell Communications Research and Naval Research Laboratory, October 1994.
- 21.Conti, M., Dragoni, N., Lesyk, V. A Survey of Man in the Middle Attacks. IEEE Communications Surveys and Tutorials.- 2016. -Vol.18.- № 3.- P.2027-2051.
- 22.Soo Yun Hwang, Gi Yoon Parkm Dae Ho Kim, Kyong Son Jhang. Efficient Implementation of a Pseudorandom Sequence Generator for High-Speed Data Communications. ETRI Journal,-2010.- Vol.32,- № 2,-P.222-229.
- 23.Wang X., Wu Y., Caron B. Transmitted identification using embedded pseudorandom sequences. IEEE Transaction on Broadcasting.-2004. Vol.50. - № 3.- P.244-252.
- 24.Самофалов К.Г., Тубольцев А.А., Ияд Мохд Маджид Ахмад Шахрури. Повышение эффективности идентификации удаленных абонентов

многопользовательских систем // Проблеми інформатизації та управління.  
Збірник наукових праць: Випуск 3(14).-К.,НАУ.- 2008.- С.129-136.

- 25.12. Léo Ducas, Vadim Lyubashevsky, and Thomas Prest. Efficient Identity-Based Encryption over NTRU Lattices. In: ASIACRYPT 2014, Part II. Ed. by Palash Sarkar and Tetsu Iwata. Vol. 8874. LNCS. Springer, Heidelberg, Dec. 2014, pp. 22–41.
- 26.Zacharioudakis Eleftherios. Method for high speed remote user authentication based on zeroknowledge conception. Scientific notes of the Ukrainian Research Institute of Communication. -2017.- Vol.45.-no.1- P.109-116.
- 27.Зензин О. С., Иванов М. А., Стандарт криптографической защиты XXI века – AES. Конечные поля. М.: КУДИЦ-ОБРАЗ.- 2002 – 174 с.
- 28.Мухін В.Є. Метод ідентифікації віддалених абонентів на основі концепції “нульових знань” / В.Є. Мухін, Лефтеріс Захаріудакіс, Ю.Н. Герасименко, М.С. Козерацький // Телекомунікаційні та інформаційні технології, 2017 – №1.– С.50-57. рг.-2005.- 286 с.

|     |      |          |        |      |                    |      |
|-----|------|----------|--------|------|--------------------|------|
|     |      |          |        |      | ІАЛЦ.467200.003 ПЗ | Арк. |
|     |      |          |        |      |                    | 62   |
| Зм. | Арк. | № докум. | Підпис | Дата |                    |      |

# **ДОДАТОК 1**

Метод та програмні засоби криптографічно-строкої  
ідентифікації на базі генераторів псевдовипадкових послідовностей

**Блок-схема алгоритму шифрування**  
**ІАЛЦ.467200.004 Д1**

Аркушів 1

**Київ 2025 р**



|           |                  |          |        |      |                                                                                                      |                                           |       |
|-----------|------------------|----------|--------|------|------------------------------------------------------------------------------------------------------|-------------------------------------------|-------|
|           |                  |          |        |      | ІАЛЦ.467200.004 Д1                                                                                   |                                           |       |
|           |                  | № докум. | Підпис | Дата |                                                                                                      |                                           |       |
| Розробив  | Чернишевич М.С.  |          |        |      | Метод та програмні засоби<br>криптографічно-строкої ідентифікації<br>Блок-схема алгоритму шифрування | Літ.                                      | Аркуш |
| Перевірів | Марковський О.П. |          |        |      |                                                                                                      | 1                                         | 1     |
| Н. Контр. | Міщенко Л.Д.     |          |        |      |                                                                                                      | КПІ ім. Ігоря<br>Сікорського, ФІОТ, ІО-11 |       |
| Затвердив |                  |          |        |      |                                                                                                      |                                           |       |

## **ДОДАТОК 2**

**Метод та програмні засоби криптографічно-строкої ідентифікації на базі  
генераторів псевдовипадкових послідовностей**

**Схема електрична функціональна**

**ІАЛЦ.467200.005 Д2**

**Аркушів 1**

**Київ 2025 р**



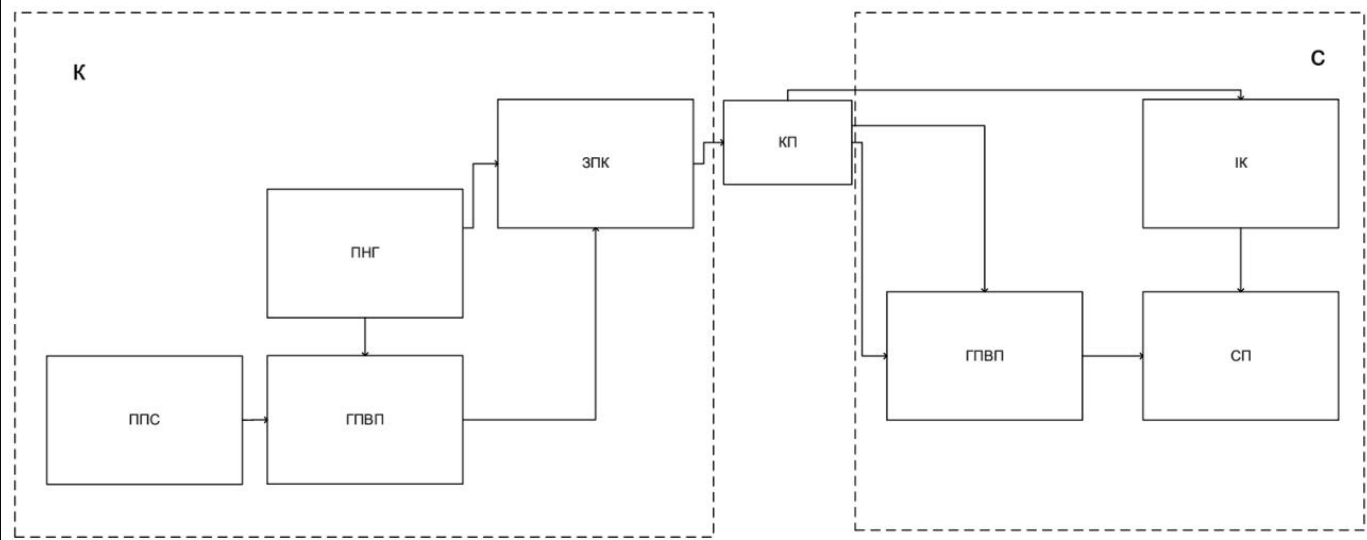
## **ДОДАТОК 3**

Метод та програмні засоби криптографічно-строкої ідентифікації на базі  
генераторів псевдовипадкових послідовностей

**Структурна схема**  
ІАЛЦ.467200.006 ДЗ

Аркушів 1

Київ 2025 р



*К – користувач*

*ЗПК – захищена пам'ять користувача*

*ППС – пам'ять початкових станів генератора*

*ГПВП – Генератор псевдовипадкових послідовностей*

*ПНГ – пам'ять налаштувань генератора*

*КП – канал передачі*

*С – система*

*ІК – ідентифікуючий код*

*СП – схема порівняння*

|           |                  |          |        |      |                                                                                                                                                 |  |                                           |         |
|-----------|------------------|----------|--------|------|-------------------------------------------------------------------------------------------------------------------------------------------------|--|-------------------------------------------|---------|
|           |                  |          |        |      | ІАЛЦ.467200.006 ДЗ                                                                                                                              |  |                                           |         |
|           |                  |          |        |      |                                                                                                                                                 |  |                                           |         |
|           |                  | № докум. | Підпис | Дата |                                                                                                                                                 |  |                                           |         |
| Розробив  | Чернишевич М.С.  |          |        |      | Метод та програмні засоби<br>криптографічно-строгої ідентифікації<br>на базі генераторів псевдовипадкових<br>послідовностей<br>Структурна схема |  | Літ.                                      | Аркуш   |
| Перевірив | Марковський О.П. |          |        |      |                                                                                                                                                 |  |                                           | Аркушів |
|           |                  |          |        |      |                                                                                                                                                 |  | 1                                         | 1       |
| Н. Контр. | Міщенко Л.Д.     |          |        |      |                                                                                                                                                 |  | КПІ ім. Ігоря<br>Сікорського, ФІОТ, ІО-11 |         |
| Затвердив |                  |          |        |      |                                                                                                                                                 |  |                                           |         |

## **ДОДАТОК 4**

Метод та програмні засоби криптографічно-строкої  
ідентифікації на базі генераторів псевдовипадкових послідовностей

Текст програмного коду  
ІАЛЦ.467200.007 Д4

Аркушів 10

Київ 2025 р

```
#include <iostream>
#include <vector>
#include <bitset>
#include <cmath>
#include <string>
#include <fstream>
#include <sstream>
#include <numeric>
#include <random>
#include <chrono>
#include <unordered_set>
#include <openssl/sha.h>
#include <openssl/evp.h>
#include <openssl/buffer.h>
#include <iomanip>
class LFSRUnit {
private:
    std::vector<int> state;
    std::vector<int> feedbackMask;
public:
    LFSRUnit(const std::vector<int>& seed, const std::vector<int>&
mask)
        : state(seed), feedbackMask(mask) {}
    int shift() {
        int newBit = 0;
        for (size_t i = 0; i < state.size(); ++i) {
            if (feedbackMask[i]) newBit ^= state[i];
        }
        int outputBit = state[0];
```

```

        state.erase(state.begin());
        state.push_back(newBit);
        return outputBit;
    }

    std::string getStateString() const {
        std::string s;
        for (int bit : state) s += std::to_string(bit);
        return s;
    }

    void randomizeState() {
        std::random_device rd;
        std::mt19937 gen(rd());
        std::uniform_int_distribution<> dis(0, 1);
        for (int& bit : state) {
            bit = dis(gen);
        }
    }

    std::string getInitialStateString() const {
        return getStateString();
    }
};

class BitStreamGenerator {
private:
    std::vector<LFSRUnit> lfsrUnits;
    std::vector<int> lookupTable;
public:
    BitStreamGenerator(const std::vector<std::vector<int>>& seeds,
        const std::vector<std::vector<int>>& masks, const std::vector<int>&
        keyTable) {

```

```

        for (size_t i = 0; i < seeds.size(); ++i) {
            lfsrUnits.emplace_back(seeds[i], masks[i]);
        }
        lookupTable = keyTable;
    }
    int generate() {
        int address = 0;
        for (size_t i = 0; i < lfsrUnits.size(); ++i) {
            address |= (lfsrUnits[i].shift() << i);
        }
        return lookupTable[address % lookupTable.size()];
    }
    std::vector<int> generateMultiple(int count) {
        std::vector<int> output;
        for (int i = 0; i < count; ++i) {
            output.push_back(generate());
        }
        return output;
    }

    std::vector<std::string> getLfsrStates() const {
        std::vector<std::string> states;
        for (const auto& lfsr : lfsrUnits) {
            states.push_back(lfsr.getStateString());
        }
        return states;
    }
    void randomizeAllStates() {
        for (auto& lfsr : lfsrUnits) {

```

```

        lfsr.randomizeState();
    }
}

int detectCycleLength(int maxIterations = 100000) {
    std::unordered_set<std::string> seen;
    std::string initial;
    for (const auto& lfsr : lfsrUnits) {
        initial += lfsr.getInitialStateString();
    }
    seen.insert(initial);
    for (int i = 1; i <= maxIterations; ++i) {
        generate();
        std::string current;
        for (const auto& lfsr : lfsrUnits) {
            current += lfsr.getStateString();
        }
        if (seen.count(current)) {
            return i;
        }
        seen.insert(current);
    }
    return -1;
}

};

std::string computeSHA256(const std::vector<int>& bits) {
    std::string bitString;
    for (int bit : bits) bitString += (bit ? '1' : '0');
    unsigned char hash[SHA256_DIGEST_LENGTH];

```

```

        SHA256((const unsigned char*)bitString.c_str(),
bitString.size(), hash);
        std::stringstream ss;
        for (int i = 0; i < SHA256_DIGEST_LENGTH; ++i) {
            ss << std::hex << std::setw(2) << std::setfill('0') <<
(int)hash[i];
        }
        return ss.str();
    }

std::string sha256ToBase64(const std::vector<int>& bits) {
    std::string bitString;
    for (int bit : bits) bitString += (bit ? '1' : '0');
    unsigned char hash[SHA256_DIGEST_LENGTH];
    SHA256((const unsigned char*)bitString.c_str(),
bitString.size(), hash);
    BIO* bio, * b64;
    BUF_MEM* bufferPtr;
    b64 = BIO_new(BIO_f_base64());
    bio = BIO_new(BIO_s_mem());
    bio = BIO_push(b64, bio);
    BIO_set_flags(bio, BIO_FLAGS_BASE64_NO_NL);
    BIO_write(bio, hash, SHA256_DIGEST_LENGTH);
    BIO_flush(bio);
    BIO_get_mem_ptr(bio, &bufferPtr);
    std::string base64Hash(bufferPtr->data, bufferPtr->length);
    BIO_free_all(bio);
    return base64Hash;
}

```

```
void saveHashToFile(const std::string& hash, const std::string&
filename) {
    std::ofstream file(filename);
    file << hash;
    file.close();
    std::cout << "Хеш збережено до файлу: " << filename <<
std::endl;
}
```

```
void displayMenu() {
    std::cout << "\nМеню:\n";
    std::cout << "1 – Згенерувати біти\n";
    std::cout << "2 – Зберегти у файл\n";
    std::cout << "3 – Переглянути стан LFSR\n";
    std::cout << "4 – Показати статистику\n";
    std::cout << "5 – Рандомізувати стан LFSR\n";
    std::cout << "6 – Визначити довжину циклу\n";
    std::cout << "7 – Обчислити SHA256 хеш (hex)\n";
    std::cout << "8 – Зберегти SHA256 хеш (hex) у файл\n";
    std::cout << "9 – Обчислити SHA256 хеш (Base64)\n";
    std::cout << "0 – Вийти\n";
}
```

```
int main() {
    int lfsrCount, registerLength;
    std::cout << "Введіть кількість LFSR: ";
    std::cin >> lfsrCount;
    std::cout << "Введіть довжину регістру: ";
    std::cin >> registerLength;
    std::vector<std::vector<int>> seeds, masks;
```

```

    for (int i = 0; i < lfsrCount; ++i) {
        std::vector<int> seed(registerLength),
mask(registerLength);
        std::cout << "Seed для LFSR #" << i + 1 << ": ";
        for (int& bit : seed) std::cin >> bit;
        std::cout << "Mask для LFSR #" << i + 1 << ": ";
        for (int& bit : mask) std::cin >> bit;
        seeds.push_back(seed);
        masks.push_back(mask);
    }
    std::vector<int> keyTable(4096);
    for (int i = 0; i < 4096; ++i) keyTable[i] = i % 2;
    BitStreamGenerator generator(seeds, masks, keyTable);
    std::vector<int> lastBits;
    while (true) {
        displayMenu();
        int option;
        std::cin >> option;
        if (option == 0) break;
        switch (option) {
            case 1: {
                int count;
                std::cout << "Скільки бітів згенерувати: ";
                std::cin >> count;
                lastBits = generator.generateMultiple(count);
                std::cout << "Результат: ";
                for (int b : lastBits) std::cout << b;
                std::cout << std::endl;
                break;
            }
        }
    }

```

```

    }
    case 2: {
        std::string filename;
        std::cout << "Ім'я файлу: ";
        std::cin >> filename;
        saveHashToFile(std::string(lastBits.begin(),
lastBits.end()), filename);
        break;
    }
    case 3: {
        auto states = generator.getLfsrStates();
        for (size_t i = 0; i < states.size(); ++i) {
            std::cout << "LFSR #" << i + 1 << " стан: " <<
states[i] << std::endl;
        }
        break;
    }
    case 4: {
        int ones = std::accumulate(lastBits.begin(),
lastBits.end(), 0);
        int zeros = lastBits.size() - ones;
        double ratio = lastBits.empty() ? 0 :
static_cast<double>(ones) / lastBits.size();
        std::cout << "Кількість 1: " << ones << "\n";
        std::cout << "Кількість 0: " << zeros << "\n";
        std::cout << "Співвідношення 1/загалом: " << ratio
<< std::endl;
        break;
    }
}

```

```

case 5: {
    generator.randomizeAllStates();
    std::cout << "Рандомізація завершена.\n";
    break;
}
case 6: {
    int cycleLen = generator.detectCycleLength();
    if (cycleLen == -1)
        std::cout << "Цикл не знайдено у межах
обмеження.\n";
    else
        std::cout << "Довжина циклу: " << cycleLen <<
std::endl;
    break;
}
case 7: {
    if (lastBits.empty()) {
        std::cout << "Спершу згенеруйте бітовий
потік.\n";
    } else {
        std::string hash = computeSHA256(lastBits);
        std::cout << "SHA256 хеш (hex): " << hash <<
std::endl;
    }
    break;
}
case 8: {
    if (lastBits.empty()) {

```

```

        std::cout << "Спершу згенеруйте бітовий
потік.\n";

        } else {
            std::string hash = computeSHA256(lastBits);
            std::string filename;
            std::cout << "Введіть ім'я файлу для збереження
хешу: ";

            std::cin >> filename;
            saveHashToFile(hash, filename);
        }
        break;
    }
    case 9: {
        if (lastBits.empty()) {
            std::cout << "Спершу згенеруйте бітовий
потік.\n";

        } else {
            std::string b64 = sha256ToBase64(lastBits);
            std::cout << "SHA256 хеш (Base64): " << b64 <<

std::endl;

        }
        break;
    }
    default:
        std::cout << "Невідомий вибір.\n";
    }
}
return 0;
}

```