

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ ВАДИМА ГЕТЬМАНА**

ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ В ЕКОНОМІЦІ

Кафедра системного аналізу та кібербезпеки

ОСВІТНЬО-ПРОФЕСІЙНА ПРОГРАМА «КІБЕРБЕЗПЕКА»

Галузь знань 12 «Інформаційні технології»

Спеціальність 125 «Кібербезпека»

Форма навчання: очна (денна)

КВАЛІФІКАЦІЙНА БАКАЛАВРСЬКА РОБОТА

на тему «Система генерації псевдовипадкових ключових послідовностей»

здобувача Мацкової Діани Валентинівни

Науковий керівник: к.т.н., доцент, Кулініч Олег Миколайович

**Робота допущена до захисту перед екзаменаційною
комісією з атестації здобувачів вищої освіти (ЕК)**

Завідувач кафедри: д.ф.-м.н., професор Джалладова І.А.

Київ 2024

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ ВАДИМА ГЕТЬМАНА**

ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ В ЕКОНОМІЦІ

Кафедра системного аналізу та кібербезпеки

ОСВІТНЬО-ПРОФЕСІЙНА ПРОГРАМА «КІБЕРБЕЗПЕКА»

Галузь знань 12 «Інформаційні технології»

Спеціальність 125 «Кібербезпека»

Форма навчання: очна (денна)

ПОГОДЖЕНО

Керівник проектної групи (гарант)
освітньо-професійної програми
«Кібербезпека»
к.ф.-м.н., доцент Г.В. Мамонова

_____ 2024 р.

ЗАТВЕРДЖУЮ:

Завідувач кафедри системного
аналізу та кібербезпеки

д.ф.-м.н., проф. І.А. Джалладова

_____ 2024 р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ

здобувача вищої освіти Мацкової Діани Валентинівни

денної форми навчання

на підготовку кваліфікаційної бакалаврської роботи

***на тему «Система генерації псевдовипадкових ключових
послідовностей»***

Тему затверджено наказом ректора Університету 30.04.2024 р. № 726-ст

План кваліфікаційної бакалаврської роботи

Розділ 1	Загальні поняття захисту інформації
Розділ 2	Принципи побудови і функціонування різноманітних генераторів.
Розділ 3	Алгоритмізація апаратного та математичного генераторів
Розділ 4	Система генерації псевдовипадкових ключових послідовностей
Об'єкт дослідження:	процес генерування псевдовипадкових ключових послідовностей.
Предмет дослідження: технологія впровадження генератора псевдовипадкових чисел для надійного шифрування в інформаційну систему органу місцевого самоврядування.	
Мета кваліфікаційної бакалаврської роботи: розробка елементів системи захисту інформації на основі впровадження генератора псевдовипадкових ключових послідовностей, детальний опис етапів впровадження на прикладному рівні, що є важливим етапом елемента захисту для підвищення рівня безпеки, при генеруванні інших криптографічних елементів, таких як ключі шифрування та електронні підписи для органу місцевого самоврядування.	
Конкретні завдання, які здобувач повинен виконати для досягнення поставленої мети: 1. Сформувані основні поняття інформаційної безпеки. 2. Проаналізувати існуючі методи ГПВЧ. Дослідити криптографічні та математичні властивості генерації. 3. Визначити основні методи для реалізації гібридного ГПВЧ. 4. Розробити та впровадити ГПВЧ у інформаційну систему органу місцевого самоврядування.	
<u>У розділі 1</u>	Здійснити дослідження предметної галузі, описати ключові терміни інформаційної безпеки. Охарактеризувати криптографічні та технічні засоби захисту інформації, а також захист від витоку інформації різними способами.
<u>У розділі 2</u>	Описати принципи побудови різноманітних генераторів чисел. Надати математичне представлення генератора на основі послідовності Фібоначчі, а також охарактеризувати генератори шуму, зокрема, «кольорові шуми».
<u>У розділі 3</u>	Сформувані два алгоритми генераторів для подальшої програмної реалізації. Надати переваги та недоліки кожного з них, а також порівняльну характеристику задля виокремлення сильних сторін.
<u>У розділі 4</u>	Здійснити на прикладному рівні програмну реалізацію, в основі якої полягає об'єднання двох існуючих генераторів, враховуючи їх сильні сторони, задля отримання більшої випадковості ключів шифрування.

Завдання
підготував
науковий
керівник

_____ к.т.н., доцент, Кулініч О.М.

Завдання
одержав
здобувач

_____ Мацкова Д.В.

РЕФЕРАТ

Кваліфікаційна бакалаврська робота складається зі вступу, чотирьох розділів, загальних висновків, списку використаних джерел, додатку і містить 84 сторінки основного тексту, 21 рисунок, 2 сторінки додатків. Перелік використаних джерел містить 39 найменувань і займає 5 сторінок.

Перелік ключових слів: генератор, псевдовипадковість, генерація, шифрування, ключова послідовність, захист інформації.

Предмет дослідження: технологія впровадження генератора псевдовипадкових чисел для надійного шифрування в інформаційну систему органу місцевого самоврядування.

Об'єкт дослідження: процес генерування псевдовипадкових ключових послідовностей.

Мета кваліфікаційної бакалаврської роботи полягає в розробці елементів системи захисту інформації на основі впровадження генератора псевдовипадкових ключових послідовностей, детальний опис етапів впровадження на прикладному рівні, що є важливим етапом елемента захисту для підвищення рівня безпеки, при генеруванні інших криптографічних елементів, таких як ключі шифрування та електронні підписи органу місцевого самоврядування.

Завданням кваліфікаційної бакалаврської роботи є впровадження написаного коду, який виконує функції генерування чисел для використання у створенні криптографічного захисту під виглядом ключів при передачі конфіденційної інформації та електронних підписів.

Апаратні та програмні засоби, що використовувались при проектуванні – Visual Studio Code та мова програмування Java Script.

Продемонстрований вихідний код, може бути використаний у інших інформаційних системах, де впровадження генератору є раціональним.

Рік виконання кваліфікаційної бакалаврської роботи 2024.

Рік захисту кваліфікаційної бакалаврської роботи 2024.

ВІДГУК
керівника бакалаврської кваліфікаційної роботи
кафедри системного аналізу та кібербезпеки
Інституту інформаційних технологій в економіці
МАЦКОВОЇ ДІАНИ ВАЛЕНТИНІВНИ

Актуальність теми Тема кваліфікаційної роботи Мацкової Діани «Система генерації псевдовипадкової ключової послідовності» є актуальною оскільки зростання кіберзагроз та вимоги до стійкості криптографії роблять системи ГПВЧ все більш важливими.

Повнота розкриття теми. Тема бакалаврської кваліфікаційної роботи розкрита Мацковою Д.В. в повному обсязі, про що свідчать результати приведені в пояснювальній записці, новизна отриманих результатів та їх практична цінність.

Теоретичний рівень. Студентка Мацкова Д.В. в своїй кваліфікаційній роботі самостійно виконала аналіз існуючих аспектів захисту інформації, збір та попередньо підготовку даних. В процесі роботи над кваліфікаційною роботою студенткою Мацковою Д.В. були використані сучасні методи досліджень, зокрема виконана розробка програмної реалізації запропонованих рішень.

Практична значущість. Отримані в кваліфікаційній роботі результати можуть бути використані для подальшої розробки КСЗІ.

Самостійність виконання роботи.

Кваліфікаційна робота студентки Мацкової Д.В. виконана у повному обсязі, самостійно в визначені строки та у відповідності до завдання. Оформлена у відповідності до нормативних вимог.

Загальна оцінка роботи та висновок щодо рекомендації до захисту в ЕК. Результати представленої до захисту кваліфікаційної роботи свідчать про високий рівень підготовки її автора. Кваліфікаційна робота студентки Мацкової Д.В. повністю відповідає вимогам, що висуваються до кваліфікаційних робіт освітнього ступеня бакалавр і заслуговує оцінки «добре», а її автор Мацкова Д.В. заслуговує присвоєння освітньої кваліфікації «бакалавр кібербезпеки».

Керівник бакалаврської роботи
к.т.н, доцент кафедри САтаК

Кулініч О.М.

« » червня 2024 р.

РЕЦЕНЗІЯ

на кваліфікаційну бакалаврську роботу
студентки Мацкової Діани Валентинівни гр. ІК - 401
на тему: «Система генерації псевдовипадкових ключових послідовностей»

Актуальність теми

Робота присвячена створенню нової системи генерації псевдовипадкових ключових послідовностей (ГПВЧ). Ця тема є надзвичайно актуальною в сучасних умовах, коли інформаційна безпека та захист даних стають ключовими аспектами в діяльності будь-якої організації.

Наукова новизна

Полягає в розробці нового ГПВЧ, який має покращені характеристики, порівняно з існуючими.

Якість проведеного аналізу

Аналіз проведено на високому рівні, із застосуванням сучасних методів та підходів. Автор детально розглянув тему ГПВЧ, їх особливості та переваги, а також здійснив порівняння, на основі якого було виокремлено сильні сторони досліджуваних генераторів.

Уміння користуватися літературними джерелами

Робота демонструє хороше вміння автора користуватися літературними джерелами. Використано широкий спектр наукових праць, статей та інших ресурсів, що свідчить про систематичний підхід до дослідження.

Практична цінність висновків і рекомендацій

Практична цінність дослідження полягає у можливості реального впровадження розробленої системи генерації псевдовипадкових числових послідовностей. Запропонована програмна реалізація може значно покращити випадковість ключів шифрування в організаціях, забезпечуючи високий рівень захисту конфіденційних даних.

Переваги та недоліки

Серед переваг роботи слід відзначити глибокий аналіз проблеми, детальний огляд сучасних технологій та практичну спрямованість дослідження. Однак, деякі аспекти можуть бути вдосконалені. Наприклад, у роботі відсутній детальний розгляд можливих ризиків та перешкод при впровадженні системи генерації псевдовипадкових числових послідовностей.

Загальний висновок і оцінка роботи

Загалом, дипломна робота справляє позитивне враження. Вона є актуальною, науково новою та має високу практичну цінність. Автор демонструє глибокі знання предмету та здатність до самостійного наукового дослідження.

в.о. начальника відділу
інформаційно-комп'ютерного
забезпечення
Сарненської міської ради



Поручиков С.В.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ	4
ВСТУП	6
РОЗДІЛ 1. ЗАГАЛЬНІ ПОНЯТТЯ ЗАХИСТУ ІНФОРМАЦІЇ	7
1.1 Основні визначення, які формують захист інформації	7
1.1.1 Ключові терміни	8
1.1.1.1 Інформаційна безпека	8
1.1.1.2 Модель CIA	9
1.1.1.3 Ідентифікація та автентифікація	10
1.1.2 Види захисту інформації	11
1.1.2.1 Криптографічний метод	13
1.1.2.2 Технічний метод	17
1.1.3 Поєднання методів для створення КСЗІ	18
1.2 Поняття криптосистеми	19
1.2.1 Симетричні криптосистеми	20
1.2.2 Асиметричні криптосистеми	22
1.3 Технічний захист інформації	23
1.3.1 ТЗІ від НСД	24
1.3.2 Захист інформації від витоку технічними каналами	27
Висновки до розділу 1	29
РОЗДІЛ 2. ПРИНЦИПИ ПОБУДОВИ І ФУНКЦІОНУВАННЯ РІЗНОМАНІТНИХ ГЕНЕРАТОРІВ	31
2.1 Генератори числових послідовностей	31
2.1.1 Загальне поняття генератора	32
2.1.2 Псевдовипадкові криптографічні генератори	33
2.1.3 Шумові технічні генератори	34
2.2 Побудова математичного генератора чисел	36
2.2.1 Детермінований генератор Фібоначчі	37
2.2.1.2 Методи ініціалізації та налаштування генератора	39
2.2.2 Недетерміновані генератори випадкових бітів	41

2.2.2.1 Фізичні процеси для створення випадковості.....	42
2.2.2.2 Вимоги до стійкості та непередбачуваності випадкових бітів	44
2.3 Побудова генератора шумових сигналів	45
2.3.1 Характеристики шумового генератора	46
2.3.1.1 Білий шум.....	46
2.3.1.2 Рожевий шум.....	49
2.3.2 Інтенсивність та розподіл амплітуд шуму	50
2.3.3 Кореляційні властивості шуму	51
Висновки до розділу 2	53
РОЗДІЛ 3. АЛГОРИТМІЗАЦІЯ АПАРАТНОГО ТА МАТЕМАТИЧНОГО ГЕНЕРАТОРІВ	55
3.1 Апаратний генератор: генератор лінійного рекурентного регістру	55
3.1.1 Принцип роботи	55
3.1.2 Застосування ГЛРР.....	60
3.1.3 Переваги та недоліки ГЛРР	61
3.2 ГПВЧ на основі математичного алгоритму: вихор Мерсенна.	63
3.2.1 Принцип роботи	63
3.2.2 Застосування вихору Мерсенна	65
3.2.3 Переваги та недоліки вихору Мерсенна	67
3.3 Порівняльна характеристика двох генераторів	68
Висновки до розділу 3	70
РОЗДІЛ 4. СИСТЕМА ГЕНЕРАЦІЇ ПСЕВДОВИПАДКОВИХ КЛЮЧОВИХ ПОСЛІДОВНОСТЕЙ.....	72
Висновки до розділу 4	77
ВИСНОВКИ.....	79
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ	80
ДОДАТКИ	
ДОДАТОК А	

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

ГПВЧ – Генератор псевдовипадкових чисел.

CIA – Керівна модель для розробки правил інформаційної безпеки(від англ. Confidentiality, Integrity, Availability).

TLS – Протокол транспортного рівня (від англ. Transport Layer Security).

HTTPS – Протокол безпеки гіпертекстового передачі (від англ. Hypertext Transfer Protocol Secure).

ТЗІ – Технічний захист інформації.

КСЗІ – Комплексна система захисту інформації.

AES – Симетричний блоковий шифр (від англ. Advanced Encryption Standard).

DES – Симетричний алгоритм шифрування даних (від англ. Data Encryption Standard).

VPN – Віртуальні приватні мережі (від англ. virtual private network).

RSA – Алгоритм шифрування з відкритим ключем Рівеста-Шаміра-Адлемана (від англ. Rivest-Shamir-Adleman).

SSL – Протокол безпеки розподіленого рівня (від англ. Secure Sockets Layer).

SSH – Протокол безпеки прикладного рівня (від англ. Secure Shell).

ECC – Розділ криптографії (від англ. Elliptic Curve Cryptography).

IoT – Концепція мережі, що складається із взаємозв'язаних фізичних пристроїв (від англ. internet of things).

НСД – Несанкціонований доступ.

ТЗІвідНСД – Технічний захист інформації від несанкціонованого доступу.

ІТС – Інформаційно-телекомунікаційна система.

КЗЗ – Комплекс засобів захисту.

ПЕМВ – Побічні електромагнітні випромінювання.

ГВЧ – Генератор випадкових чисел.

КГПВЧ – Криптографічні генератори псевдовипадкових чисел.

ГВБ – Генератор випадкових бітів.

НГВБ – Недетермінований генератор випадкових бітів.

ВІ – Вектор ініціалізації.

NIST – Національний інститут стандартів і технології (від англ. The National Institute of Standards and Technology).

ГЛРР – Генератор лінійного рекурентного регістру.

НСК – Найменше спільне кратне.

XOR – Виняткова диз'юнкція (від англ. eXclusive OR).

МТ – Вихор Мерсенна (від англ. Mersenne Twister).

MTGP – Вихор Мерсенна для графічних процесорів (від англ. Twister for Graphic Processors).

SFMT – Швидкий вихор Мерсенна (від англ. SIMD-oriented Fast Mersenne Twister).

ВСТУП

Актуальність і перспективність тематики роботи полягає в необхідності забезпечення надійного захисту інформації у сучасному цифровому світі. Важливу роль у цьому відіграють генератори псевдовипадкових чисел. Ці послідовності використовуються для шифрування та дешифрування інформації, а також для створення електронних підписів.

Метою кваліфікаційної бакалаврської роботи є розробка елементів системи захисту інформації на основі впровадження генератора псевдовипадкових ключових послідовностей, детальний опис етапів впровадження на прикладному рівні, що є важливим етапом елемента захисту для підвищення рівня безпеки, при генеруванні інших криптографічних елементів, таких як ключі шифрування та електронні підписи органу місцевого самоврядування.

Предметом дослідження є технологія впровадження генератора псевдовипадкових чисел для надійного шифрування в інформаційну систему органу місцевого самоврядування.

Об'єктом дослідження є процес генерування псевдовипадкових ключових послідовностей.

Методами дослідження теоретичний аналіз наукової літератури, порівняльний аналіз методів, а також практичне впровадження.

Очікується, що результати дослідження дозволять розробити та реалізувати власну систему, яка відповідає вимогам криптографічної стійкості та швидкості генерації ключів, а також впровадити систему до органів місцевого самоврядування.

Практична значущість роботи полягає у підвищенні безпеки інформаційних систем через розробку надійних генераторів псевдовипадкових чисел.

Основні наукові положення і результати доповідалися на науково-практичній конференції «Інтелектуальні методи аналізу загроз кібербезпеки»(Київ, 2023).

РОЗДІЛ 1. ЗАГАЛЬНІ ПОНЯТТЯ ЗАХИСТУ ІНФОРМАЦІЇ

1.1 Основні визначення, які формують захист інформації

Захист інформації став визначальним фактором для успіху не лише в сфері бізнесу, але й у нашому повсякденному житті. Необхідно визнати, що сучасне суспільство пов'язане з безпекою інформації, і ті, хто зможе ефективно захищати цю інформацію, матимуть ключову перевагу в умовах постійних загроз та викликів. У цьому розділі будуть докладені зусилля для розуміння, впровадження та вдосконалення стратегій захисту інформації, щоб створити стійке та безпечне цифрове середовище для всіх.

Інформаційна безпека - це практика захисту інформації шляхом зменшення інформаційних ризиків. Вона передбачає захист інформаційних систем та інформації, що обробляється, зберігається та передається цими системами, від несанкціонованого доступу, використання, розкриття, перекручення, модифікації або знищення.

Ефективна інформаційна безпека вимагає комплексного підходу, який враховує всі аспекти інформаційного середовища, включаючи технології, політику і процедури, а також людей. Вона також вимагає постійного моніторингу, оцінки та адаптації для усунення нових загроз і вразливостей.

Перед тим, як поглиблюватися в основну проблему роботи, варто ознайомитися з основними поняттями захисту інформації, а також її складовими.

Для початку, важливо визначити поняття конфіденційності, цілісності та доступності, які є основними складовими трьохсторонньої моделі CIA[1]. Конфіденційність визначає те, наскільки ефективно інформація захищена від несанкціонованого доступу; цілісність вказує на стан непошкодженості інформації та її невідмінності; доступність визначає, наскільки легко користувачі можуть отримати доступ до необхідної інформації у відповідний момент часу, що демонструє рис 1.1.



Рисунок 1.1 - Трьохстороння модель CIA

Окрім того, у цьому розділі розглянуть ключові аспекти ідентифікації та аутентифікації, які виступають фундаментом будь-якої системи захисту інформації. Таким чином, цей розділ буде важливим кроком для глибокого розуміння загальних понять захисту інформації та створення ефективних стратегій, які забезпечать високий рівень безпеки в умовах постійно змінюючогося цифрового середовища.

1.1.1 Ключові терміни

1.1.1.1 Інформаційна безпека

Якщо звернутися до офіційних джерел, зокрема до законодавства України, визначення «інформаційної безпеки» є вузьконаправленим, оскільки безпосередньо стосується безпеки інформації нашої держави.

Інформаційна безпека — стан захищеності життєво важливих інтересів людини і громадянина, суспільства і держави, при якому запобігається завдання шкоди через неповноту, несвоєчасність та недостовірність поширюваної інформації, порушення цілісності та доступності інформації, несанкціонований обіг інформації з обмеженим доступом, а також через негативний інформаційно-

психологічний вплив та умисне спричинення негативних наслідків застосування інформаційних технологій[2].

Інший ресурс, такий як Microsoft, теж має своє визначення, яке більше стосується корпоративної діяльності.

Інформаційна безпека – це набір процедур та інструментів, які захищають усю делікатну корпоративну інформацію від неправомірного використання, несанкціонованого доступу, псування або знищення[3].

Проаналізувавши велику кількість інформаційних ресурсів, можна сформулювати визначення, яке узагальнить та охопить усі сфери, які мають потребу в захисті інформації.

Інформаційна безпека - це комплекс заходів, стратегій та технологій, спрямованих на забезпечення конфіденційності, цілісності та доступності інформації в обличчі різноманітних загроз та ризиків, а також основною метою яких є забезпечення стійкості та захищеності інформаційних ресурсів, систем та мереж від несанкціонованого доступу, руйнівних атак, втрати чи порушення конфіденційності.

1.1.1.2 Модель CIA

Три літери в "тріаді CIA" розшифровуються як конфіденційність, цілісність і доступність. Тріада CIA - це поширена модель, яка лежить в основі розробки систем безпеки. Їх використовують для пошуку вразливостей і методів створення рішень. В ідеалі, коли всі три стандарти дотримані, профіль безпеки організації стає сильнішим і краще підготовленим до протидії інцидентам, що загрожують.

Положення про технічний захист інформації в Україні[4] теж має у своєму контексті дану модель. Однак, як такого поняття «модель CIA» не використовується, тому вони представлені як властивості інформації. В свою чергу, перед тим як перейти до основних термінів, варто зазначити, що таке «властивість інформації»:

Властивість інформації - це характеристика, яка визначає рівень важливості, якість чи стан інформаційного ресурсу.

Властивості інформації визначають, наскільки ця інформація є цінною, достовірною, конфіденційною, цілісною та доступною. Зазвичай властивості інформації визначаються з урахуванням конкретних потреб і вимог користувачів, а також контексту використання цієї інформації. Зокрема, ужиті в вищезгаданому положенні терміни мають такі значення:

Конфіденційність – властивість інформації бути захищеною від несанкціонованого ознайомлення.

Цілісність – властивість інформації бути захищеною від несанкціонованого спотворення, руйнування або знищення.

Доступність – властивість інформації бути захищеною від несанкціонованого блокування[4].

1.1.1.3 Ідентифікація та автентифікація

Ідентифікація та автентифікація є основою інформаційної безпеки, забезпечуючи комплексну основу для перевірки легітимності користувачів, які отримують доступ до цифрових ресурсів.

Ідентифікація є процесом встановлення унікальної ідентичності для фізичної або юридичної особи в системі. Основна мета ідентифікації полягає в унікальному визначенні об'єкта чи особи, встановленні їхньої ідентичності в системі. Цей процес дозволяє надавати унікальні ідентифікатори, такі як імена користувачів чи номери, для однозначної ідентифікації суб'єктів. Автентифікація, з іншого боку, передбачає перевірку заявленої ідентичності, гарантуючи, що лише авторизовані користувачі отримують доступ до конфіденційної інформації або функціональних можливостей. Цей процес включає в себе перевірку різних факторів, таких як паролі, біометричні дані чи криптографічні ключі, щоб забезпечити відповідність пред'явленого ідентифікатора заздалегідь встановленим автентифікаційним даним.

Якщо звернутися до Закону України про електронну ідентифікацію та електронні довірчі послуги[5], можна також знайти термін, такі як «автентифікація», «електронна ідентифікація», а також «ідентифікація особи». В цьому законі терміни вживаються в такому значенні:

Автентифікація - електронний процес, що дає змогу підтвердити електронну ідентифікацію фізичної, юридичної особи, інформаційної або інформаційно-комунікаційної системи та/або походження та цілісність електронних даних.

Електронна ідентифікація - процес використання ідентифікаційних даних особи в електронній формі, які однозначно визначають фізичну, юридичну особу або уповноваженого представника юридичної особи.

Ідентифікація особи - процес використання ідентифікаційних даних особи з документів, створених на матеріальних носіях, та/або електронних даних, у результаті якого забезпечується однозначне встановлення фізичної, юридичної особи або уповноваженого представника юридичної особи та перевірка належності особі таких даних[6].

1.1.2 Види захисту інформації

Захист даних є ключовим питанням для приватних осіб, підприємств та організацій будь-якого розміру. Зі зростанням залежності від технологій для зберігання та передачі конфіденційної інформації, загроза витоку та крадіжки даних є головною проблемою.

Щоб зменшити ризики, було розроблено кілька методів захисту даних від несанкціонованого доступу та маніпуляцій. Залежно від загрози можливого порушення роботи системи та несанкціонованого доступу до інформації, багато видів захисту можна розділити на такі групи: морально-етичні, правові, адміністративні (організаційні), технічні (фізичні) та криптографічні.

Морально-етичний захист інформації є критичним аспектом у нашому взаємопов'язаному та оцифрованому світі. Справедливість є етичною вимогою при

роботі з інформацією. Дані слід використовувати чесно та справедливо, уникаючи дискримінації та упередженості.

Не менш важливим методом захисту інформації є встановлення правових норм і стандартів, які регулюють збір, обробку, зберігання та передачу інформації. Ці норми визначають права та обов'язки суб'єктів, які працюють з інформацією, і встановлюють відповідальність за порушення цих норм.

Україна використовує різноманітні правові методи для захисту інформації. Один із основних законів, що регулює це питання, - « Закон України про захист інформації в інформаційно-телекомунікаційних системах»[7]. Цей документ визначає основні принципи та заходи захисту інформації, права та обов'язки суб'єктів обробки інформації, а також встановлює відповідальність за порушення норм безпеки.

Ще одним важливим законом є « Закон України про захист персональних даних», який регулює збір, обробку та зберігання персональних даних громадян[8]. Цей закон має на меті забезпечення конфіденційності та безпеки особистої інформації.

Для боротьби з кіберзлочинністю та захисту критично важливих об'єктів інформаційної інфраструктури використовується "Закон України про основні засади забезпечення кібербезпеки України "[9]. Цей закон визначає основні принципи організації кіберзахисту та встановлює механізми реагування на кіберінциденти.

Важливо враховувати також міжнародний аспект захисту інформації, оскільки віртуальний світ не обмежується національними кордонами. Участь у міжнародних конвенціях та угодах може забезпечити взаємодію та обмін досвідом з іншими країнами для ефективного вирішення спільних проблем. Україна приєднана до міжнародних угод і конвенцій, таких як « Конвенція про захист осіб у зв'язку з автоматизованою обробкою персональних даних» (Конвенція 108)[10].

Одним з ключових методів з захисту інформації є криптографічний метод та криптографія як наука загалом. Криптографічний метод захисту інформації також детальніше розглядатиметься далі. В загальному, він використовує математичні

алгоритми та ключі для шифрування та розшифрування даних, забезпечуючи конфіденційність та цілісність інформації.

Однією з основних складових криптографії є використання симетричних та асиметричних ключів, які також розглядатимуться у наступних пунктах.

1.1.2.1 Криптографічний метод

Перш ніж розглянути безпосередньо сам криптографічний метод, варто зазначити певні визначення, які допоможуть краще зрозуміти його. Зокрема, що таке криптологія, криптографія та криптоаналіз.

Криптологія зосереджена на побудові та аналізі протоколів, які не дають жодній людині чи системі розшифрувати повідомлення без дозволу. Криптологія допомагає забезпечити основні принципи кібербезпеки, забезпечуючи конфіденційність, цілісність, автентифікацію та неможливість спростування.

Зокрема, криптографія - це наука про таємне письмо з метою приховування сенс повідомлення. Криптографія також поділяється на криптографічні протоколи, симетричні алгоритми та асиметричні (або з відкритим ключем) алгоритми.

Криптографічні протоколи, також відомі як протоколи безпеки або шифрування, складаються з чітко визначеної серії кроків і обмінів повідомленнями між кількома обмінами повідомленнями між кількома суб'єктами для досягнення певної мети безпеки.

Криптографічні протоколи використовуються для безпечної передачі даних на рівні додатків наприклад TLS, який використовується для захисту веб-з'єднань HTTPS. Ще одним відомим протоколом є обмін ключами Діффі-Хеллмана, який також використовується TLS у захищених з'єднаннях.

Проаналізувавши джерела, можна узагальнити такі поняття та терміни:

Криптологія - це наука про перетворення інформації для забезпечення її таємності, що складається з двох гілок: криптографії та криптоаналізу.

Криптографія - наука про способи перетворення (шифрування) інформації з метою її захисту від незаконних користувачів. [11]

Криптоаналіз - наука про методи отримання вихідного значення зашифрованої інформації, не маючи доступу до секретної інформації (ключа), необхідної для цього[12].

Криптографічне шифрування даних - це процес перетворення інформації за допомогою кодування. Прототипом цього методу був обмін секретними повідомленнями між державними чиновниками. Повідомлення шифрувалося за допомогою спеціального алгоритму (ключа) і відправлялося одержувачу. Одержувач використовував аналогічний алгоритм розшифровки.

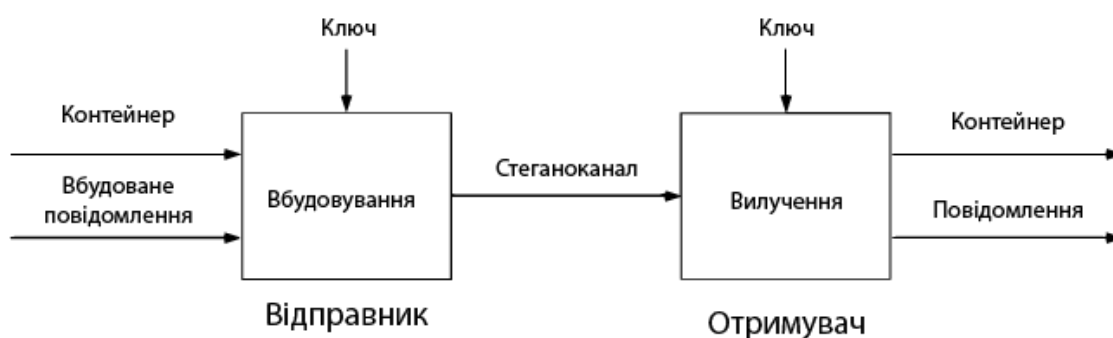
Інші методи захисту включають кодування, стеганографію та стиснення. Вони так само ефективні, як і традиційне шифрування.

Кодування - це простий метод захисту повідомлення шляхом використання різних наборів символів для заміни слів або фраз у повідомленні, що кодується. Кодування, як і шифрування, використовується протягом століть. Різниця між цим методом захисту інформації полягає в тому, що код легше розгадати, ніж шифр.

До винаходу комп'ютерів для захисту повідомлень кодувальники використовували цілі словники, в яких значення кожного слова відображалось у вигляді набору кодових символів. Зараз кодування і зворотне перетворення здійснюється за допомогою спеціальних програм або апаратних засобів. Кодування є популярним способом захисту інформації у військових частинах та інших підрозділах збройних сил.

Стеганографія - спосіб маскування секретної інформації. Застосовується лише до електронних носіїв. Принцип стеганографії полягає в наступному: текстові файли, відео або зображення за допомогою певної програми змінюються на рівні програмного коду. Оскільки будь-який електронний файл являє собою чергування символів у певній послідовності, заміна деяких з цих символів призводить до зміни вмісту файлу. Відповідно, користувач, який не має ключа для розшифровки файлів,

що містяться на пристрої, не зможе знайти і зрозуміти приховану інформацію.



Детальну схему роботи стеганографії зображено на рис 1.2 .

Рисунок 1.2 – Схема роботи стеганографії

Стиснення вважається одним із засобів криптографічного захисту даних лише опосередковано. Це пов'язано з тим, що спочатку стиснення електронних файлів здійснювалося не для захисту інформації, а для зменшення її обсягу. Однак, оскільки стиснуту інформацію неможливо прочитати, стиснення стало засобом криптографічного захисту даних.

З усіх методів захисту інформаційних активів, шифрування є найпоширенішим. Це пов'язано з тим, що воно підходить для будь-яких цілей. При використанні цього методу на сучасних гаджетах не потрібно використовувати всю потужність пристрою. Тому більшість месенджерів, встановлених на смартфонах і планшетах, шифрують всі повідомлення користувачів.

Існує багато методів шифрування текстових повідомлень і аудіофайлів. Однак не всі вони використовуються через різний ступінь надійності.

Кожен метод шифрування оцінюється з точки зору наступних факторів:

- Криптостійкість. У програмуванні є такий термін, як криптоатака.

Це поняття, яке визначає процес дешифрування повідомлення шляхом вгадування ключів. Відповідно, криптостійкість - це рівень надійності шифру, який визначається складністю підбору ключів.

- Обсяг зашифрованого повідомлення. Оскільки для швидкої передачі інформації важливо зберегти її початковий обсяг, перевага віддається методам, в

яких обсяг зашифрованого тексту дорівнює обсягу вихідного повідомлення або трохи перевищує його.

- Відсутність помилок. Деякі методи дають збої, через що зміст зашифрованих повідомлень частково або повністю втрачається. Тому в загальній практиці обраний метод шифрування перевіряється на наявність помилок перед реалізацією.

- Швидкість шифрування та розшифрування. Чим швидше шифрується, передається і розшифровується повідомлення, тим вища популярність методу.

- Цінова доступність використовуваних алгоритмів. Для оцінки їхньої ефективності вартість порівнюється з цінністю інформації та фінансовими наслідками її витоку.

Виходячи з перерахованих факторів, можна зробити висновок, що найбільш популярними і ефективними методами шифрування є надійні, швидкі, недорогі алгоритми, які не призводять до втрати даних або збільшення їх обсягу.

Що стосується використання криптографічного методу захисту інформації в межах держави, є затверджене «Положення про порядок здійснення криптографічного захисту інформації в Україні»[13]. Це Положення визначає порядок здійснення криптографічного захисту інформації з обмеженим доступом, розголошення якої завдає (може завдати) шкоди державі, суспільству або особі.

Зокрема, Положення містить термін, який дає визначення «криптографічному захисту»:

Криптографічний захист — вид захисту, що реалізується за допомогою перетворень інформації з використанням спеціальних даних (ключових даних) з метою приховування (або відновлення) змісту інформації, підтвердження її справжності, цілісності, авторства тощо[13].

1.1.2.2 Технічний метод

Технічний метод захисту інформації має на меті захистити людей, майно та матеріальні активи від будь-яких дій чи подій, що можуть призвести до їхньої втрати або пошкодження.

Технічний захист інформації (ТЗІ) – сукупність організаційних та технічних заходів і прав, норм, спрямованих на забезпечення порядку доступу і цілісності закритої (таємної, конфіденційної) або відкритої інформації, а також на запобігання заподіяння шкоди суб'єктам інформаційних відносин. Об'єктом захисту є будь-яка інформація, що обробляється в автоматизованих системах, права власників цієї інформації та власників систем, а також права користувачів.

Важливо розуміти, що фізична безпека стосується всього простору, і вона не повинна обмежуватися лише вхідними дверима, а поширюватися на всю будівлю. Експерти з безпеки називають таку форму захисту глибоким або багаторівневим захистом, оскільки у фізичній інфраструктурі є кілька контрольних точок. Фізичні пошкодження так само шкідливі, як і цифрові втрати, тому необхідно вживати суворих заходів фізичної безпеки.

Наразі існує багато векторів атак, і вони можуть бути спрямовані не лише з фізичної та технологічної точки зору, але й на вивчення слабких місць, притаманних людській природі, так звана соціальна інженерія[14]. Атаки соціальної інженерії бувають різних форм і можуть здійснюватися будь-де, де задіяна людська взаємодія.

Що стосується використання технічного методу захисту інформації в межах держави, ТЗІ забезпечується шляхом проведення єдиної технічної політики у цій сфері, дотримання зазнач, суб'єктами відповідних норм і правил, використання програм, забезпечення, систем зв'язку, засобів обчислювальної техніки, захисту інформації тощо, які мають відповідний сертифікат. Суб'єкти права власності, визначені автор, правом або договір, відносинами, мають право на юридичний захист від заподіяння шкоди власнику інформації внаслідок навмисної чи

ненавмисної втрати, знищення, підроблення, спотворення і блокування інформації та інші неправомірних дій.

Державна політика ТЗІ формується на основі актів і реалізується департаментом спеціальних телекомунікаційних систем та захисту інформації Служби безпеки України, основними завданнями якої є створення і розвиток системи ТЗІ в Україні та здійснення контролю за її функціонуванням. Він також встановлює вимоги і правила щодо захисту інформації, що є власністю держави, або інформації, захист якої гарантується державою. Особи, винні в порушенні порядку ТЗІ, несуть дисциплінарну, матеріальну, адміністративну чи кримінальну відповідальність[4].

1.1.3 Поєднання методів для створення КСЗІ

Поєднання криптографічних і технічних методів у сфері інформаційної безпеки є надійним підходом до захисту конфіденційних даних і систем. Криптографічні методи, засновані на математичних алгоритмах і ключах, відіграють ключову роль у забезпеченні конфіденційності та цілісності інформації. Ці методи передбачають шифрування даних, щоб зробити їх нечитабельними без відповідного ключа для розшифрування, тим самим запобігаючи несанкціонованому доступу.

У тандемі з криптографічними заходами, технічні методи зміцнюють механізми захисту, усуваючи вразливості в апаратних і програмних системах. Однією з важливих переваг такого комбінованого підходу є стійкість. Інтеграція криптографічних і технічних методів створює стійкий механізм захисту, здатний протистояти кіберзагрозам, що постійно розвиваються. Оскільки зловмисники розробляють більш витончені методи, ця синергія гарантує, що заходи безпеки залишаються проактивними та адаптивними.

Організації часто використовують комбінацію криптографічних і технічних методів для створення надійного і багаторівневого захисту від кіберзагроз. Одним з яскравих прикладів є інтеграція протоколів шифрування із заходами мережевої

безпеки. У цьому сценарії криптографічні методи використовуються для шифрування даних як у стані спокою, так і під час передачі, гарантуючи, що навіть у разі несанкціонованого доступу інформація залишиться нерозбірливою без відповідних ключів розшифрування.

Інший приклад - поєднання криптографічних методів і захисту кінцевих точок. Організації часто розгортають повнодискове шифрування на окремих пристроях для захисту даних, що зберігаються на них[15]. Цей криптографічний метод гарантує, що в разі втрати або крадіжки пристрою дані залишаться недоступними без ключа шифрування.

1.2 Поняття криптосистеми

Термін "криптосистема" охоплює складний набір інструментів, методів і математичних принципів, призначених для захисту даних від несанкціонованого доступу та забезпечення конфіденційності, цілісності й автентичності інформації.

Криптографічна система (криптосистема) - це сукупність засобів криптографічного захисту інформації, необхідної нормативної, експлуатаційної та іншої документації, які складають єдину систему з метою роз'язання конкретної задачі захисту інформації. Криптосистема складається з таких базових підсистем: шифрування, ідентифікація, забезпечення цілісності (імітозахисту), цифрового підпису тощо; кожна з них має свою підсистему ключів[16].

Центральне місце в її архітектурі займають криптографічні алгоритми, які визначають математичні операції, що виконуються над даними для шифрування та розшифрування. В рамках цієї криптографічної сфери виникають дві основні парадигми - симетричні та асиметричні (з відкритим ключем) алгоритми, які будуть розглянуті згодом.

Що стосується ключів, то їхня ключова роль в управлінні процесами шифрування і дешифрування не викликає сумнівів. Симетричні криптосистеми ґрунтуються на єдиному секретному ключі, тоді як асиметричні сценарії

передбачають подвійність публічних і приватних ключів. На рис 1.3 наглядно показано різницю між ними.

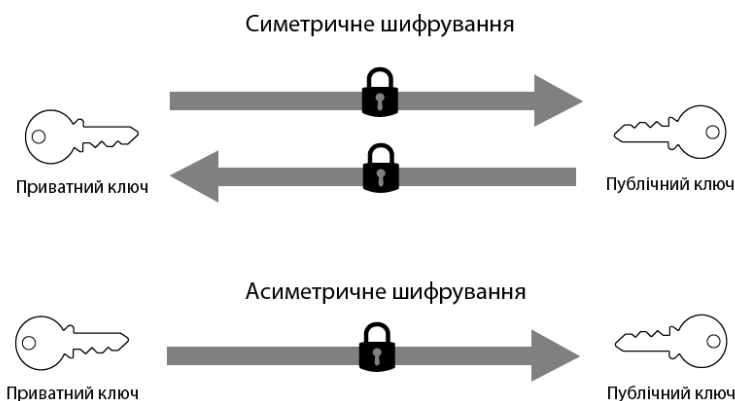


Рисунок 1.3 – Різниця між симетричним та асиметричним шифруванням

Окрім звичайного шифрування, деякі криптосистеми інтегрують вже згадану стеганографію, приховуючи інформацію в начебто нешкідливих даних, щоб завуалювати комунікацію. Криптоаналіз, що протистоїть криптографічним зусиллям, займається аналізом і потенційним зламом криптографічних систем. Криптографи та експерти з безпеки постійно вдосконалюють ці системи, щоб вони могли протистояти різноманітним атакам та адаптуватися до нових загроз.

1.2.1 Симетричні криптосистеми

Симетрична криптографія, також відома як криптографія з секретним ключем, є фундаментальним аспектом сучасної криптографії, який передбачає використання одного секретного ключа для шифрування і дешифрування повідомлень.

Симетричні алгоритми шифрування – ключ, який використовується для шифрування повідомлень, є похідним від ключа розшифрування, і навпаки, такі криптографічні алгоритми називаються симетричними. Надійність алгоритмів з одним ключем визначається вибором ключа. Якщо злоумисник знає ключ, він

зможе розшифрувати всі вилучені дані без будь-якого опору. Це означає, що обраний ключ необхідно тримати в таємниці від сторонніх.

Існує два типи алгоритмів симетричного шифрування. Один з них працює з бітами простого тексту. Вони називаються потоковими алгоритмами або потоковими шифрами. У другому випадку відкритий текст розбивається на блоки по кілька біт. Такі алгоритми називаються блоковими алгоритмами або блоковими шифрами. У сучасних комп'ютерних алгоритмах блочного шифрування довжина блоку зазвичай становить 64 біта.

Незважаючи на свою ефективність і швидкість, симетрична криптосистема має обмеження, особливо в тих випадках, коли безпечний розподіл ключів є складним або недоцільним. У таких випадках рішенням є асиметрична криптографія, яка використовує пару ключів (відкритий і закритий) для шифрування і дешифрування. Гібридні криптографічні системи часто поєднують сильні сторони як симетричної, так і асиметричної криптографії для досягнення балансу між продуктивністю і безпекою в різних додатках, таких як протоколи безпечного зв'язку і цифрові підписи.

До поширених алгоритмів симетричного шифрування відносяться :

- AES — це широко використовуваний симетричний алгоритм шифрування, прийнятий урядами та організаціями по всьому світу. Він підтримує довжину ключа 128, 192 або 256 біт і працює з блоками даних фіксованого розміру.
- DES — один з перших симетричних алгоритмів шифрування, розроблених у 1970-х роках. Хоча він вважається застарілим через малу довжину ключа (56 біт), він заклав основу для сучасних методів симетричного шифрування.
- Blowfish — симетричний блоковий шифр. Він підтримує змінну довжину ключа (до 448 біт) і відомий своєю простотою і швидкістю. Blowfish використовується в різних додатках, включаючи системи електронної комерції, VPN та хешування паролів. Ці алгоритми відрізняються за довжиною ключа, розміром блоку та іншими параметрами, але всі вони працюють за одним і тим же принципом - використання одного спільного секретного ключа для шифрування і розшифрування.

1.2.2 Асиметричні криптосистеми

На відміну від симетричної криптосистеми, у асиметричній криптосистемі (або криптосистемі з відкритим ключем) є два різні ключі, які використовуються для шифрування та розшифрування даних. Ключ, який використовується для шифрування, є загальнодоступним і тому називається відкритим ключем, а ключ для розшифрування зберігається в таємниці і називається приватним ключем. Ключі генеруються таким чином, що неможливо отримати приватний ключ з відкритого ключа.

В асиметричній системі передавач і приймач мають по два ключі. Однак приватний ключ зберігається в таємниці і не надсилається разом з повідомленням одержувачу, на відміну від публічного ключа, який надсилається.

Асиметрична криптографія має низку переваг над симетричною криптографією, особливо в тих випадках, коли безпечний обмін ключами або їх розповсюдження є складним завданням. Оскільки відкритий ключ може вільно розповсюджуватися, асиметрична криптографія спрощує процес встановлення безпечних каналів зв'язку між сторонами, які ніколи раніше не спілкувалися. Вона також полегшує використання цифрових підписів, коли відправник може використовувати свій приватний ключ для підписання повідомлення, надаючи гарантії цілісності та автентичності одержувачу, який може перевірити підпис за допомогою відкритого ключа відправника.

Однак асиметрична криптографія, як правило, працює повільніше і вимагає більших обчислень порівняно з симетричною криптографією. Як наслідок, її часто використовують у поєднанні з симетричною криптографією в гібридних криптографічних системах, щоб використовувати сильні сторони обох підходів. Наприклад, у захищених протоколах зв'язку, таких як TLS, асиметрична криптографія використовується для обміну ключами та автентифікації, в той час як симетрична криптографія застосовується для шифрування власне передачі даних, досягаючи балансу між безпекою та ефективністю.

До поширених алгоритмів асиметричного шифрування відносяться

- Алгоритм RSA – один з найпоширеніших алгоритмів асиметричного шифрування. Він передбачає генерування пари відкритих і закритих ключів на основі математичної складності факторизації великих складених чисел. RSA використовується в цифрових підписах, захищеному електронному листуванні, безпечному перегляді веб-сторінок і системах управління ключами.
- Обмін ключами Діффі-Хеллмана – це протокол обміну ключами, який дозволяє двом сторонам створити спільний секретний ключ через незахищений канал зв'язку. Він не передбачає шифрування, а скоріше забезпечує безпечне узгодження ключів. Діффі-Хеллмана використовується в різних протоколах захищеного зв'язку, включаючи TLS/SSL, SSH.
- Криптографія еліптичних кривих ECC – це асиметричний метод шифрування, заснований на теорії еліптичних кривих. Він забезпечує безпеку, порівнянну з традиційними криптосистемами з відкритим ключем, але з меншою довжиною ключа, що робить його особливо придатним для обмежених середовищ, таких як мобільні пристрої та пристрої IoT. ECC використовується в TLS/SSL, цифрових підписах і криптовалютних системах, таких як Bitcoin.

1.3 Технічний захист інформації

ТЗІ є дуже поширеним видом безпеки, що використовується в організаціях, які використовують комп'ютери або майже будь-які інші технології. Він полягає у виявленні прогалин у системі безпеки та пошуку адекватних рішень для усунення ризику технічного збою або злому. Оскільки більшість даних існує в нефізичній формі, з переміщенням даних на хмарні накопичувачі та портативні гаджети, важко забезпечити безпечний сеанс зв'язку та передачу інформації. У ненадійних мережах і пристроях, а також при взаємодії з неавторизованими системами відсутні засоби контролю безпеки. Це деякі з основних причин, що зумовлюють підвищену потребу в технічній безпеці між пристроями та в мережах.

У Положенні про технічний захист інформації в Україні, яке вже згадувалося у попередніх підрозділах, є термін, який безпосередньо дає визначення «технічному захисту інформації»:

ТЗІ – діяльність, спрямована на забезпечення інженерно-технічними заходами конфіденційності, цілісності та доступності інформації[4].

Метою ТЗІ є запобігання:

- витоку інформації технічними каналами;
- витоку інформації за рахунок несанкціонованих дій щодо неї;
- несанкціонованим діям щодо інформації, які призводять до її знищення, порушення цілісності або блокування.

Напрями розвитку ТЗІ регулюються необхідністю оперативного здійснення заходів, що відповідають масштабам інформаційних загроз та виходячи з принципів правової врегульованої демократичної держави, відповідно до права суб'єктів інформаційних відносин на отримати доступ до інформації та захистити її.

Приведення інформаційних відносин у сфері ТЗІ у відповідність до міжнародних стандартів сприятиме перетворенню України на демократичну правову країну у світі.

У розділі вже згадувалося про технічний захист інформації, однак, лише з теоретичних аспектів, а також як узагальнений метод захисту інформації. У цьому ж підрозділі ТЗІ детально розглядатиметься як елемент діяльності. Зокрема, захист від несанкціонованого доступу та від витоку технічними каналами.

1.3.1 Технічний захист інформації від несанкціонованого доступу

У сфері кібербезпеки несанкціонований доступ означає порушення комп'ютерних систем, мереж або баз даних. Ці порушення, як правило, пов'язані з проникненням хакерів у систему з метою викрадення, зміни або знищення інформації. Однак важливо зазначити, що несанкціонований доступ не обмежується атаками зовнішніх хакерів. Це також може бути доступ співробітників до файлів або інформації за межами їхнього рівня авторизації.

Несанкціонований доступ(НСД) – це процес входу або доступу до системи, фізичної або електронної, без дозволу власника або адміністратора. Такий доступ може бути отриманий шляхом обходу заходів безпеки, використання вразливостей системи або за допомогою викрадених облікових даних. НСД є серйозним порушенням законів про конфіденційність і може призвести до серйозних наслідків, включно з судовими позовами.

Коли неавторизовані особи отримують доступ до системи, вони часто націлені на конфіденційні дані, такі як фінансові записи, особиста ідентифікаційна інформація, комерційна таємниця або інтелектуальна власність. Таке вторгнення може призвести до значних фінансових втрат, шкоди репутації компанії та потенційних юридичних наслідків.

Крім того, в деяких випадках зловмисник може не лише викрасти дані, але й пошкодити, знищити або зашифрувати їх. Такий акт саботажу може завдати катастрофічної шкоди, особливо компаніям, які значною мірою покладаються на свої дані. Наслідки таких випадків можуть бути руйнівними - від зупинки бізнес-операцій до значної втрати довіри серед клієнтів і замовників.

Існує безліч способів здійснення НСД, зокрема:

- Ініціативна співпраця при несанкціонованому доступі є одним зі зловживань, які можуть спричинити значні шкоди підприємству. Ініціатор такої співпраці може діяти з різних мотивів, включаючи особисту вигоду, шантаж, або співпрацю з конкурентами. Це може призвести до витоку конфіденційної інформації, порушення безпеки даних, втрати репутації підприємства та фінансових збитків.

- Схилення до співпраці – це, як правило, насильницька дія з боку зловмисників. Схилення або вербування може здійснюватися шляхом підкупу, залякування, шантажу. Схилення до співпраці реалізується у вигляді реальних загроз, переслідування та інших дій, що виражаються в переслідуванні, образі і т.д.

- Підслуховування є ще одним способом ведення розвідки і промислового шпигунства, вживаний агентами, спостерігачами, інформаторами, спеціальними постами підслуховування. Підслуховування може здійснюватися

безпосереднім сприйняттям акустичних коливань особою при прямому сприйнятті мовної інформації або за допомогою технічних засобів.

- Спостереження як візуальний спосіб ведення розвідки про стан і діяльність супротивника за допомогою оптичних приладів. Як правило, ведеться цілеспрямовано в певний час і в потрібному місці спеціально підготовленими людьми, ведеться потайно. До технічних засобів спостереження відносяться оптичні прилади (біноклі, труби, перископи), телевізійні системи (для звичайної освітленості і низькорівневої), прилади спостереження вночі і при обмеженій видимості.

- Перехоплення дозволяє отримати розвідувальну інформацію за рахунок прийому сигналів електромагнітної енергії пасивними засобами прийому, розташованими, як правило, на достатній відстані від джерела конфіденційної інформації. До перехоплення переговорів схильні будь-які системи радіозв'язку, переговори, що ведуться з рухливих засобів телефонного зв'язку, переговори усередині приміщення за допомогою безпроводних систем установського зв'язку та інші[17].

Задля збереження конфіденційності, цілісності та доступності інформації, впроваджується ТЗІ від несанкціонованого доступу (ТЗІвідНСД).

ТЗІвідНСД – це діяльність, спрямована на забезпечення дотримання правил розмежування доступу шляхом створення і підтримки в дієздатному стану системи заходів із захисту інформації[17].

Основними принципами забезпечення ТЗІвідНСД є планування захисту і керування системою захисту, керування системою доступу, а також забезпечення послуг безпеки і гарантій.

Для забезпечення безпеки інформації під час її обробки в ІТС створюються КСЗІ, процес управління якою повинен підтримуватись протягом всього життєвого циклу ІТС.

При плануванні дій слід враховувати порядок створення КСЗІ:

- аналіз об'єкта захисту і можливих загроз – визначення ресурсів ІТС, що підлягають захисту, при цьому загрози визначаються в термінах ймовірності їх реалізації і величини можливих збитків;
- оцінка ризиків (на підставі аналізу загроз, існуючих в системі вразливостей, ефективності вже реалізованих заходів захисту);
- вироблення заходів захисту, перетворення яких в житті дозволило б знизити рівень остаточного ризику до прийняттого рівня;
- формулювання або коригування політики безпеки;
- розробка плану захисту (опис послідовності і змісту всіх стадій і етапів життєвого циклу КСЗІ)[18].

Захист інформації повинен забезпечуватись протягом всього періоду її існування – з моменту створення об'єкта ІТС або його імпорту до системи і до його знищення або експорту з системи всі запити на доступ до об'єкта і об'єкта на доступ до інших об'єктів мають контролюватися КЗЗ.

Необхідно, щоб абсолютно всі запити на доступ до об'єктів контролювались КЗЗ і не існувало можливості обминути цей контроль (для захисту об'єктів КЗЗ повинен, в першу чергу, забезпечувати свою цілісність і керованість). Особливе значення набуває визначення діючих за умовчанням правил, які визначають початкові умови, за яких починається існування об'єкта всередині ІТС[19].

1.3.2 Захист інформації від витоку технічними каналами

У сучасних умовах основною загрозою безпеці інформації, яка має обмежений доступ, є витік інформації технічними каналами. Витоком інформації з технічних каналів є поширення інформативного сигналу, який піддається контролю, від джерела інформації через певне фізичне середовище до технічного засобу, з якого здійснюється прийом цієї інформації.

У Законі України Про захист інформації в інформаційно-комунікаційних системах також використовується такий термін, як «витік інформації» і має визначення:

Витік інформації - результат дій, внаслідок яких інформація в системі стає відомою чи доступною фізичним та/або юридичним особам, що не мають права доступу до неї[7].

Основними каналами витоку інформації є акустичний, віброакустичний, акустоелектричний, ПЕМВ та параметричний.

Акустичний канал - це канал передачі інформації за допомогою звукових хвиль. Це може бути звичайний голосовий розмовний канал, де основними елементами є звукові коливання, передавані через повітря, воду або інший середовища. Акустичні канали використовуються в різних сферах, таких як телефонія, радіозв'язок, сонари, музикальні системи, вимірювальні та діагностичні пристрої та багато інших.

Акустичний канал може бути використаний як для односторонньої, так і для двосторонньої передачі інформації. Він може бути бездротовим, як у випадку радіозв'язку, або може використовувати провідне середовище, таке як кабель. Акустичний канал може мати різні характеристики передачі звуку, такі як частотний діапазон, шум та затримка.

Генератор білого шуму може використовуватися як захистний захід проти акустичного витоку інформації.

Віброакустичний канал витоку інформації - це канал, через який інформація може витікати шляхом передачі вібраційних сигналів через тверді поверхні, такі як стіни, стелі або меблі. Цей тип витоку інформації може бути використаний для прослуховування або видалення конфіденційної інформації, що обговорюється в приміщенні.

Захист від витоку інформації через віброакустичний канал може включати застосування звукопоглинаючих матеріалів для зменшення вібрацій твердих поверхонь, використання шумозаглушуючих пристроїв або створення штучного шуму, що перекриває віброакустичні сигнали. Така система складається, як правило, з генератора шуму та комплекту акустичних і вібраційних випромінювачів[20].

Акустоелектричний канал витоку інформації - це канал, через який інформація може витікати за допомогою модуляції електричних сигналів відповідно до акустичних сигналів. Цей тип витоку інформації може виникнути, наприклад, внаслідок акустичного перетворення звукових сигналів у вібрації або коливання електричних провідників, таких як кабелі або електричні мережі.

Захист від витоку інформації через акустоелектричний канал може включати застосування захисних пристроїв, таких як фільтри або шумоподавляючі пристрої, які зменшують вплив звукових коливань на електричні сигнали.

ПЕМВ можуть бути використані для незаконного збирання інформації, так як вони можуть передавати сигнали, що виникають в результаті обробки конфіденційної інформації в електронних пристроях. Наприклад, електромагнітні випромінювання можуть включати електромагнітні сигнали, які виникають під час роботи процесорів або інших електронних компонентів, і які можуть містити інформацію про виконувані обчислення або оброблювані дані.

Параметричний канал витоку інформації - це канал, через який інформація може витікати внаслідок витікання параметрів електронної системи або пристрою. Цей тип витоку інформації може бути використаний для незаконного збирання конфіденційної інформації, що обробляється або передається через таку систему або пристрій.

Висновки до розділу 1

У цьому розділі були розглянуті фундаментальні поняття, що лежать в основі захисту інформації. Заглибившись у базові визначення, які є невід'ємною частиною цієї сфери, створено міцний фундамент, на якому можна будувати подальше розуміння усього матеріалу у роботі. Технічні та криптографічні механізми захисту були визначені як найважливіші компоненти у забезпеченні цілісності, конфіденційності та доступності даних.

Підсумовуючи все, можна сказати, що технічні механізми захисту слугують для запобігання несанкціонованому доступу, маніпуляціям або перехопленню

даних. Комплексні системи формують захист від витоку інформації різними каналами, тим самим надаючи гарантію збереження всіх властивостей інформації.

Паралельно з цим, криптографічний захист формує важливий рівень захисту від зловмисників. Криптографічні методи, такі як симетричне та асиметричне шифрування, зміцнюють цілісність даних, автентифікацію та неможливість спростування. Використовуючи складні математичні алгоритми, криптографічні механізми кодують дані таким чином, щоб зробити їх нерозбірливими для неавторизованих суб'єктів, забезпечуючи їх конфіденційність і цілісність під час передачі та зберігання.

Задля забезпечення усіх властивостей інформації, як зазначалося у розділі, застосовуються технічні та криптографічні методи захисту. Одними з таких являються генератори шуму та криптографічні генератори, зокрема випадкових та псевдовипадкових послідовностей, які розглядатимуться у наступних розділах.

У сфері механізмів захисту генератори шуму стають важливим інструментом для зміцнення інформаційної безпеки. Шум, що характеризується притаманною йому випадковістю та непередбачуваністю, слугує потужним джерелом для генерування випадкових чисел. Генератори шуму, як апаратні, так і програмні, використовують шум навколишнього середовища або електронні флуктуації для створення випадкових потоків даних. Ці потоки можна використовувати для покращення генерації криптографічних ключів, вдосконалення криптографічних протоколів та посилення механізмів безпеки.

По суті, поєднання генерації шуму і криптографічних методів утворює симбіоз, де кожен компонент посилює ефективність іншого в захисті інформації. У той час як криптографічні генератори створюють основу для надійного шифрування і захисту даних, генерація шуму посилює цю основу, додаючи непередбачуваності і складності. Разом вони створюють потужний бар'єр проти несанкціонованого доступу, забезпечуючи конфіденційність, цілісність та автентичність конфіденційної інформації.

РОЗДІЛ 2. ПРИНЦИПИ ПОБУДОВИ І ФУНКЦІОНУВАННЯ РІЗНОМАНІТНИХ ГЕНЕРАТОРІВ

У сфері обчислювальних і статистичних програм генерація випадкових чисел є важливим інструментом, що має різноманітне застосування в таких сферах, як криптографія, симуляції, ігри та наукові дослідження. В основі цього важливого процесу лежить розуміння основ і принципів побудови різних генераторів чисел.

У цьому розділі розглядатиметься саме поняття генератора, похідних від нього. Також, прицільна увага приділятиметься побудові, функціям та принципам, що лежать в основі створення генераторів випадкових чисел та їх різноманітних різновидів. Знаючи ці основоположні принципи, можна отримати уявлення про внутрішню роботу генераторів і фактори, що впливають на їхню надійність та ефективність.

2.1 Генератори числових послідовностей

Більшість криптографічних протоколів використовують випадкові числа як вхідні дані на певних етапах або використовують їх як ключ у потокових системах шифрування.

Однак проблема отримання випадкових послідовностей аж ніяк не є тривіальною з обчислювальної точки зору. Оскільки комп'ютери є детермінованими машинами, керованими інструкціями, важко створити алгоритми, які "продукують" випадковість. При цьому виконання операції можуть виглядати випадковими, за ними стоїть заздалегідь визначена закономірність. Тому ми вдаємося до фізичних явищ, які мають внутрішню випадковість, щоб отримати "справді" випадкові дані: радіоактивний розпад, теплові шуми, атмосферні явища тощо.

Отримання даних з фізичних явищ, однак, є відносно дорогим, тому не можливо отримати необхідну кількість випадкових чисел за короткий час. Іноді, або через те, що обчислювальна потужність досить низька, або через те, що

потрібно отримати багато чисел за дуже короткий час, як у потоковому шифруванні, представляє інтерес створення швидких і обчислювально прості в обчислювальному плані генератори, які видають на перший погляд випадкові послідовності[21].

2.1.1 Загальне поняття генератора

Генератор чисел, також відомий як генератор випадкових чисел (ГВЧ), - це обчислювальний або фізичний процес чи пристрій, який генерує послідовності чисел або символів, що не мають жодної закономірності чи передбачуваності[21].

Існує два основних типи генераторів чисел:

Генератори справжніх випадкових чисел: Ці генератори генерують числа на основі дійсно випадкових фізичних процесів, які неможливо передбачити, таких як радіоактивний розпад, атмосферні шуми або теплові шуми. ГВЧ вважаються справді випадковими, оскільки їхні результати не визначаються жодним алгоритмом чи математичною формулою. Однак вони часто вимагають спеціалізованого обладнання і можуть працювати повільніше порівняно з генераторами псевдовипадкових чисел.

Генератори псевдовипадкових чисел: Ці генератори створюють послідовності чисел, які здаються випадковими, але насправді визначаються алгоритмом або математичною формулою. ГПВЧ починають з початкового значення, і використовують детерміновані алгоритми для генерації послідовностей чисел. Хоча їхні результати можуть демонструвати статистичну випадковість і можуть бути придатними для багатьох застосувань, вони не є справді випадковими. ГПВЧ широко використовуються в програмному забезпеченні завдяки своїй ефективності та простоті реалізації.

ГВЧ є фундаментальним інструментом у багатьох різних областях. Дві основні сфери застосування - це стохастичне моделювання та криптографія. У стохастичному моделюванні ГВЧ використовуються для імітації поведінки випадкової величини із заданим розподілом ймовірностей. У криптографії ці

генератори використовуються для створення секретних ключів, шифрування повідомлень або маскування вміст певних протоколів, комбінуючи його з випадковою послідовністю..

Незалежно від того, чи використовується ГВЧ для стохастичного моделювання, чи для криптографічних застосувань, він повинен задовольняти певні умови. Перш за все, вихідні дані повинні імітувати реалізацію послідовності незалежних рівномірно розподілених випадкових величин.

Випадкові величини, які не є рівномірно розподіленими, можна імітувати, застосовуючи певні перетворень на виході рівномірно розподілених генераторів,

У двійковій послідовності, яка була згенерована незалежними та однаково рівномірно розподіленими випадковими величинами випадковими величинами, одиниці та нулі, а також всі двійкові n -парні числа для $n \geq 1$ розподілені рівномірно розподілені у n -вимірному просторі. Крім того, не існує кореляції між окремими бітами або n -парами, відповідно. Від ГПВЧ ми очікуємо такої ж поведінки.

Крім того, хороший ГВЧ повинен працювати ефективно, що означає, що він повинен бути здатний генерувати велику кількість випадкових чисел за короткий проміжок часу. Для таких додатків, як стохастичне моделювання, потокові шифри, маскування протоколів або азартні ігри в Інтернеті, необхідна величезна кількість випадкових чисел, а отже, потрібні швидкі ГВЧ.

На додаток до перерахованих вище умов, ГВЧ для криптографічних додатків повинні бути стійкими до атак, що не є актуальним для стохастичного моделювання. Це означає, що супротивник не повинен мати можливості вгадати жодного поточного, майбутнього, навіть якщо він або вона має деяку інформацію про вхідні дані, внутрішній стан або про поточний чи попередній вихід ГВЧ.

2.1.2 Псевдовипадкові криптографічні генератори

Криптографічні генератори псевдовипадкових чисел (КГПВЧ) - це генератори випадкових чисел, які гарантують, що випадкові числа, які вони генерують, є абсолютно непередбачуваними. КГПВЧ задовольняють тести

наступних бітів і витримують розширення компрометації стану і, як правило, є частиною операційної системи або надходять із захищеного зовнішнього джерела.

Псевдовипадкові криптографічні генератори є важливими компонентами сучасної криптографії, що слугують для генерування непередбачуваних послідовностей чисел, які імітують справжню випадковість. Ці генератори мають вирішальне значення для різних криптографічних додатків, включаючи шифрування, генерацію ключів і цифрових підписів.

За своєю суттю, псевдовипадкові криптографічні генератори починають роботу з початкового значення, як правило, відносно невеликого фрагмента даних. За допомогою складних математичних алгоритмів ці генератори маніпулюють початковим значенням, щоб створити потік начебто випадкових вихідних даних. Згенерована послідовність здається випадковою для спостерігача без знання початкового значення, що робить її придатною для криптографічних цілей.

Однак, на відміну від справжніх генераторів випадкових чисел, які покладаються на непередбачувані за своєю природою фізичні процеси, псевдовипадкові генератори є детермінованими. При однаковому початковому значенні псевдовипадковий генератор завжди буде генерувати одну і ту ж послідовність чисел.

Незважаючи на свою детерміновану природу, генератори псевдовипадкових чисел можуть забезпечити достатній рівень безпеки для багатьох криптографічних додатків при правильній реалізації. Однак, дуже важливо періодично оновлювати початкове значення, щоб запобігти використанню зловмисниками будь-яких потенційних слабких місць в алгоритмі або в самому початковому значенні.

2.1.3 Шумові технічні генератори

Генератори шуму - це пристрої або програми, призначені для створення широкого спектру звуків, які маскують неприємні шуми навколишнього середовища, допомагають розслабитися, полегшують концентрацію уваги або підтримують експериментальну роботу в галузі звукотехніки та акустики.

У наукових дослідженнях генератори шуму використовуються в експериментах, що вимагають стохастичних процесів або випадкових вхідних даних. Наприклад, у фізиці та біології шум притаманний багатьом природним системам, і дослідники можуть використовувати генератори шуму для імітації цих умов у контрольованих експериментах. Це дозволяє вченим вивчати такі явища, як броунівський рух, нейронну активність або хаотичні системи.

Принцип дії генераторів шуму заснований на випромінюванні шуму, функцією якого є інтерференція[22]. Це гарантує, що шпигун не зможе зрозуміти записаний матеріал.

Генератори шуму є легальними і можуть використовуватися будь-ким - установами, підприємствами та приватними особами. Генератори білого шуму – це саме ті пристрої, які забезпечують конфіденційність під час дзвінків, захищаючи від витoku важливої інформації. Прослуховуючий пристрій або диктофон можна легко захопити в будь-якому приміщенні (наприклад, у звичайному предметі, такому як будильник або ліпнина). Таким чином, це простий у використанні превентивний інструмент.

На ринку все ще існує категорія пристроїв, які зазвичай називають "глушилки", що асоціюються зі стандартними генераторами шуму. Глушилка, джаммер або пригнічувач радіосигналів - все перераховане відноситься до одного пристрою, призначення якого - створити перешкоди для радіоканалу. Інша техніка, яка працює на заглушеній частоті, перестає передавати і приймати сигнали в радіусі дії джаммера. Цей тип пристроїв є незаконним. Основне призначення глушилок - створювати перешкоди на частотах мобільного зв'язку. На джаммери не потрібна ліцензія. Їх можна вільно ввозити в Україну, продавати і купувати.

Стаття 361 Кримінального кодексу України передбачає відповідальність за несанкціоноване втручання в роботу комп'ютерів, автоматизованих систем, комп'ютерних мереж або мереж електрозв'язку. Мобільний зв'язок, домашній і офісний Wi-Fi входять в поняття подібних систем.

Закон не забороняє глушити мобільний зв'язок в своєму автомобілі, офісі, якщо попереджені співробітники. Але не можна встановити глушилку в своєму або орендованому приміщенні і накрити весь поверх центру - це злочин[23].

2.2 Побудова математичного генератора чисел

Математика з її заплутаною павутиною понять і застосувань слугує основою для багатьох наукових дисциплін, пронизуючи своїми нитками такі різноманітні галузі, як фізика, економіка та інформатика. Серед багатьох її відгалужень теорія ймовірностей займає не останнє місце в цьому списку.

Теорія ймовірностей слугує фундаментальною основою для побудови математичних генераторів. Йдеться не просто про підрахунок чисел, а про розуміння невизначеності, притаманної будь-якій системі.

Більше того, теорія ймовірностей допомагає управляти ризиками та приймати рішення в умовах невизначеності. Йдеться не лише про генерування даних, а й про їх осмислення та використання для обґрунтування наших дій.

В свою чергу, комбінаторика відіграє фундаментальну роль у теорії ймовірностей, зокрема в побудові математичних ГВЧ. Оскільки справжньої випадковості важко досягти в детермінованій системі, тому ГВЧ покладаються на різні математичні методи, включаючи комбінаторику, щоб ефективно імітувати випадковість.

Комбінаторика забезпечує математичну основу для розуміння та маніпулювання кінцевими дискретними структурами, такими як перестановки, комбінації та розташування, які є важливими при розробці алгоритмів ГВЧ.

Один з поширених підходів до побудови ГВЧ полягає у використанні комбінаторних методів для створення послідовностей чисел, які мають властивості, подібні до властивостей випадкових послідовностей. Наприклад, відомий лінійний конгруентний генератор спирається на властивості модульної арифметики, які можна проаналізувати за допомогою комбінаторних методів[24].

Іншим прикладом застосування комбінаторики в побудові ГВЧ є використання алгоритмів на основі перестановок. Ці алгоритми покладаються на перетасування або перестановку кінцевого набору чисел для створення випадковості. Наприклад, алгоритм перемішування Фішера-Йейтса використовує комбінаторні принципи для створення випадкової перестановки заданої послідовності.

Комбінаторика також відіграє важливу роль в аналізі статистичних властивостей ГВЧ. Такі методи, як підрахунок кількості можливих результатів, обчислення ймовірностей та оцінка очікуваних значень певних властивостей, значною мірою покладаються на комбінаторні методи.

2.2.1 Детермінований генератор Фібоначчі

2.2.1.1 Математичне представлення послідовності Фібоначчі

Одним з прикладів математичних генераторів є детермінований генератор Фібоначчі. Перед тим, як заглиблюватися у теорію самого генератора, варто дати визначення терміну «детермінованість».

Детермінованість – це властивість процесу або явища, що характеризується тим, що його результати або стан системи визначаються чіткою послідовністю вхідних умов або подій. У детермінованому середовищі кожна дія або подія має однозначно визначений результат, і немає випадкових елементів або непередбачуваних факторів, які б впливали на процес.

Переважно, термін детермінованості використовується у теорії алгоритмів[25]. І саме детермінованість є ключовою властивістю більшості існуючих алгоритмів. Отже, це означає, що для заданого набору значень початкових (вхідних) даних алгоритм однозначно визначає порядок дій виконавця і результат цих дій[26].

Маючи розуміння, що таке детермінованість та детермінований алгоритм, можна надати визначення детермінованому генератору Фібоначчі, однак, спершу

варто зазначити, що таке «послідовність Фібоначчі», на основі якої працює генератор.

Послідовність Фібоначчі(відома також як числа Фібоначчі) - це цілі натуральні числа, розташовані в числовій послідовності таким чином, що кожне наступне число є сумою двох попередніх чисел, при цьому в цьому числовому ряді проявляються унікальні цікаві властивості, виражені в постійних стосунках між окремими членами послідовності і формуванні деяких постійних коефіцієнтів[27].

Детермінований генератор Фібоначчі – це математична функція або алгоритм, який створює послідовність чисел, відому як послідовність Фібоначчі. У детермінованому генераторі послідовність повністю визначається початковими умовами, що робить її передбачуваною і відтворюваною.

Щоб створити детермінований генератор Фібоначчі, необхідно встановити початкові умови, як правило, перші два числа послідовності, а потім використовувати рекурсивний алгоритм або ітераційний процес для генерації наступних чисел. Найпоширенішими початковими умовами є 0 і 1.

Математично послідовність Фібоначчі можна визначити рекурсивно наступним чином:

$$F(n) = F(n - 1) + F(n - 2) \quad (2.1)$$

де n – порядковий номер.

Числа Фібоначчі можна генерувати за допомогою цієї рекурсивної формули або за допомогою ітеративних методів, таких як динамічне програмування, щоб уникнути надлишкових обчислень.

Далі покроково розглянемо те, як обчислити числа Фібоначчі вручну за допомогою цієї рекурсивної формули. Наприклад, обчислимо числа Фібоначчі до 8-го члена.

Використаємо ті ж самі $F(0) = 0$ та $F(1) = 1$, як початкові значення. Щоб знайти третє число Фібоначчі $F(2)$, ми додаємо два попередніх числа $F(1)$ та $F(0)$, які дорівнюють 1 та 0 відповідно. Щоб знайти четверте число Фібоначчі $F(3)$, додаємо два попередніх числа $F(2)$ і $F(1)$, які дорівнюють: $F(3)=1+1=2$. По цьому ж

принципу розраховуємо до 8 числа, і отримуємо таку послідовність: $F(4)=3$, $F(5)=5$, $F(6)=8$, $F(7)=13$, $F(8)=21$

Можна продовжити цей процес, щоб згенерувати стільки чисел Фібоначчі, скільки потрібно. Кожне число в послідовності є сумою двох попередніх чисел, схема зображена на рис 2.1:

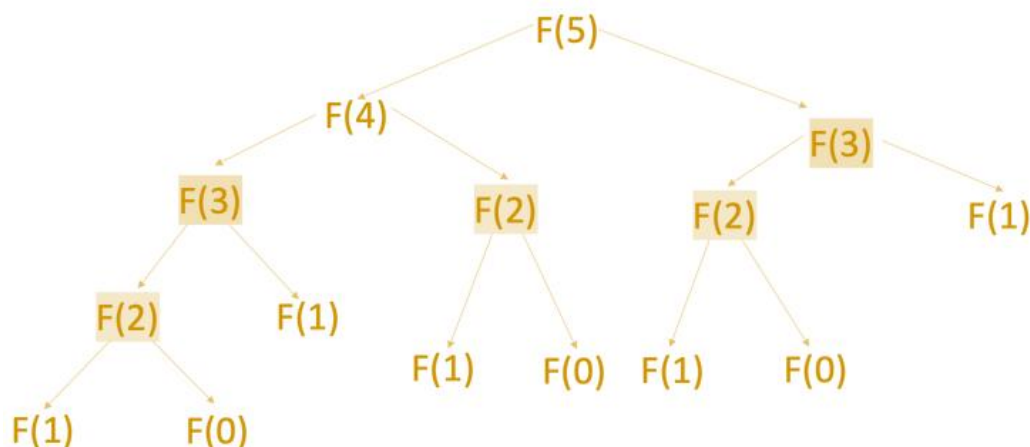


Рисунок 2.1 – Послідовність Фібоначчі

Кольорові частини діаграми представляють повторювані рішення однієї і тієї ж задачі. Якщо ви підніметесь вище по дереву, то знайдете ще більше таких повторюваних розв'язків. Це означає, що для рекурсивної генерації послідовності Фібоначчі вам доведеться обчислювати багато проміжних чисел знову і знову.

Крім того, відношення послідовних чисел Фібоначчі наближається до золотого перетину ($\phi \approx 1,618$), коли ми рухаємося далі вздовж послідовності[28].

2.2.1.2 Методи ініціалізації та налаштування генератора

Щоб ініціалізувати та налаштувати генератор Фібоначчі, починати потрібно з визначення функції або класу у вибраній вами мові програмування. Ця функція або клас буде генерувати числа Фібоначчі відповідно до бажаних специфікацій.

Варто розпочати з визначення змінних для зберігання початкових значень послідовності Фібоначчі. Зазвичай першими двома числами в послідовності є 0 і 1.

Далі визначаються будь-які додаткові параметри для налаштування генератора. Наприклад, можна дозволити користувачеві вказати початкові значення, кількість згенерованих чисел Фібоначчі або вказати, чи генерувати послідовність до нескінченності.

Після того, як визначено необхідні змінні та параметри, реалізовується логіка генерації чисел Фібоначчі. Зазвичай це передбачає використання циклу або рекурсії для обчислення кожного наступного числа в послідовності на основі двох попередніх чисел.

У JavaScript можна створити генератор чисел Фібоначчі, визначивши клас з назвою `FibonacciGenerator`. Цей клас матиме метод конструктора, який приймає необов'язкові параметри для кількості чисел Фібоначчі, які потрібно згенерувати (`n`), а також початкові значення (`start1` і `start2`). За замовчуванням ці параметри дорівнюють 10, 0 і 1 відповідно. Приклад генератора зображено на рис 2.2.

```
class FibonacciGenerator {
  constructor(n = 10, start1 = 0, start2 = 1) {
    this.n = n;
    this.start1 = start1;
    this.start2 = start2;
  }

  generateSequence() {
    let sequence = [this.start1, this.start2];
    for (let i = 2; i < this.n; i++) {
      let nextNumber = sequence[i - 1] + sequence[i - 2];
      sequence.push(nextNumber);
    }
    return sequence;
  }
}

// Example usage:
const fibGen = new FibonacciGenerator(5, 3, 4);
const fibSequence = fibGen.generateSequence();
console.log(fibSequence); // Output: [3, 4, 7, 11, 18]
```

Рисунок 2.2 – Приклад застосування у JavaScript

Усередині конструктора ці параметри зберігаються як змінні екземпляра (`this.n`, `this.start1` і `this.start2`).

Крім того, клас містить метод `generateSequence()`. Цей метод генерує послідовність Фібоначчі на основі заданих параметрів. Він ініціалізує масив з назвою `sequence` з початковими значеннями `start1` і `start2`. Потім він використовує цикл для обчислення кожного наступного числа Фібоначчі і додає його до масиву `sequence`. Цикл виконується з індексу 2 до `n - 1` включно.

Після завершення циклу повертається масив послідовностей, що містить згенеровану послідовність Фібоначчі.

Для використання генератора Фібоначчі необхідно створити екземпляр об'єкта класу `FibonacciGenerator` з потрібними параметрами. Наприклад, ви можете створити екземпляр `fibGen` з `n = 5`, `start1 = 3` і `start2 = 4`.

Нарешті, викликається метод `generateSequence()` на об'єкті `fibGen`, щоб отримати згенеровану послідовність Фібоначчі. Ця послідовність доступна для подальшої обробки або відображення.

2.2.2 Недетерміновані генератори випадкових бітів

Генератор випадкових бітів (ГВБ) – це система, вихід якої складається з повністю непередбачуваних (тобто статистично незалежних і незміщених) бітів.

ГВБ принципово відрізняється від генератора псевдовипадкових чисел тим, що повне знання структури генератора і будь-якої раніше згенерованої послідовності не призводить до знання будь-якого іншого біта. Це означає, що ентропія послідовності з `n` вихідних бітів в ідеалі має дорівнювати `n`.

Недетерміновані генератори випадкових бітів (НГВБ) - це алгоритми або пристрої, які генерують випадкові біти, не покладаючись на заздалегідь визначену послідовність або шаблон. На відміну від детермінованих генераторів, які обговорювалися вище, та котрі генерують однакову послідовність випадкових чисел за однакових початкових умов, недетерміновані генератори використовують

непередбачувані за своєю природою фізичні процеси або явища для генерування випадковості.

Побудова недетермінованого генератора випадкових бітів зазвичай передбачає використання фізичних явищ, які за своєю природою є непередбачуваними, таких як тепловий шум, радіоактивний розпад або атмосферний шум.

Тепловий шум виникає внаслідок хаотичного руху електронів у провіднику під впливом температури. Для створення теплового шуму можна використати резистор. Коли через резистор подається напруга, результуючий струм містить випадкові коливання, зумовлені тепловим рухом електронів.

Радіоактивний розпад за своєю природою є випадковим і непередбачуваним. Ви можете використовувати радіоактивний ізотоп, такий як америцій-241, що міститься в деяких детекторах диму, для генерування випадкових бітових послідовностей, заснованих на синхронізації подій атомного розпаду.

Атмосферний шум, також відомий як "білий шум", спричиняється природними атмосферними явищами, такими як грозові розряди. Можа використовувати простий радіоприймач, налаштований на невикористовувану частоту, щоб вловити атмосферний шум. Отриманий сигнал міститиме випадкові коливання, спричинені атмосферними збуреннями.

2.2.2.1 Фізичні процеси для створення випадковості

Існує багато фізичних процесів, які здаються випадковими або хаотичними. Здається, що вони не підпорядковуються жодним певним правилам, а керуються лише ймовірностями. Але всі фундаментальні фізичні закони, принаймні за межами квантової механіки, вважаються детермінованими. Однак, постає питання як саме виникає очевидна випадковість.

Одна з версій полягає в тому, що її кінцевим джерелом є зовнішній шум, часто від теплової ванни. Коли еволюція системи нестабільна, тобто збурення зростають,

будь-яка випадковість, введена через початкові та граничні умови передається або посилюється з часом, і зрештою впливає на багато компонентів системи.

Одним з широко використовуваних є НГВБ з використанням теплового шуму, який згадувався в попередньому підпункті.

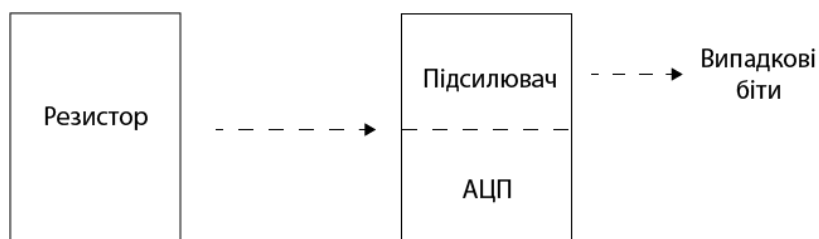


Рисунок 2.3 - Базова схема недетермінованого генератора випадкових бітів з використанням теплового шуму

У схемі на рис. 2.3 зображені такі процеси:

- резистор генерує тепловий шум;
- підсилювач підсилює слабкий сигнал теплового шуму;
- аналого-цифровий перетворювач робить вибірку посиленого шумового сигналу і перетворює його в цифрову форму;
- цифровий вихід являє собою випадкові біти, згенеровані системою.

Генератори на основі теплового шуму відіграють життєво важливу роль в інформаційній безпеці. Вони забезпечують випадковість, необхідну для генерації криптографічних ключів, векторів ініціалізації (VI), нонсів(номери, що використовуються один раз) та інших криптографічних операцій, що вимагають непередбачуваності. Ці генератори підвищують безпеку, створюючи ключі з високою непередбачуваністю, забезпечуючи унікальність VI, запобігаючи атакам з використанням нонсів і забезпечуючи безпечну автентифікацію за принципом "виклик-відповідь". Крім того, вони підтримують рандомізовані алгоритми і забезпечують випадковість для тестування надійності систем безпеки.

Однак, більш доречними в захисті інформації є білий шум.

Білий шум — будь-який послідовний звук; постійний шум, спектральна інтенсивність якого не залежить від частоти[29].

Оскільки він охоплює всі частоти, він був би надзвичайно дратівливим, якби його посилили до рівня гучності, який був би ефективним для маскування людської мови. Він приховує людську мову, але також створює некомфортне робоче середовище.

2.2.2.2 Вимоги до стійкості та непередбачуваності випадкових бітів

Стабільність і непередбачуваність випадкових бітів є критично важливими аспектами в різних галузях, таких як криптографія, моделювання та генерація випадкових чисел. Забезпечення цих характеристик вимагає ретельної уваги до кількох ключових вимог.

Перш за все, стабільність означає узгодженість згенерованих випадкових бітів у часі та в різних системах або середовищах. Щоб досягти стабільності, основний процес генерації випадкових бітів повинен бути стійким до зовнішніх факторів, які потенційно можуть вплинути на результат.

Забезпечення стабільності та непередбачуваності випадкових бітів вимагає багатогранного підходу, що спирається на різні дисципліни та методології. Наприклад, розглянемо сценарій, коли криптографічна програма вимагає постійного потоку випадкових бітів для генерації безпечних ключів. У такому випадку апаратні компоненти, відповідальні за генерацію випадковостей, такі як апаратний ГВЧ, повинні бути ретельно підібрані. Прикладом може бути фізичне джерело ентропії, таке як датчик радіоактивного розпаду або діодний генератор шуму, обидва з яких створюють випадкові флуктуації, які за своєю природою є непередбачуваними. Після вибору відповідного джерела ентропії наступним кроком є розробка процесу вилучення, який перетворює сиру випадковість на придатну для використання послідовність випадкових бітів.

Крім того, для підтримки стабільності необхідно відстежувати продуктивність процесу генерації випадкових бітів у часі. Це може включати періодичне тестування та аналіз для виявлення будь-яких відхилень від очікуваної поведінки. Наприклад, статистичний аналіз згенерованої випадкової послідовності

бітів за допомогою таких тестів, як NIST Statistical Test Suite, може виявити закономірності або зсуви, які можуть поставити під загрозу непередбачуваність.

Крім апаратних і алгоритмічних міркувань, на стабільність і непередбачуваність генерації випадкових бітів можуть впливати фактори навколишнього середовища. Наприклад, коливання температури або електромагнітні перешкоди можуть вносити зсуви або кореляції в джерело випадковості, підбиваючи непередбачуваність.

2.3 Побудова генератора шумових сигналів

Незважаючи на те, що шумові генератори вже згадувалися в попередньому пункті, все ж таки варто надати більш детальний опис усім процесам, пов'язаних з ними.

Генератор шумових сигналів - це електронний пристрій, який використовується для створення випадкових сигналів у діапазоні частот

В основі генератора шумових сигналів лежить джерело шуму, яке часто є спеціалізованою електронною схемою, призначеною для створення випадкових коливань напруги або струму. Це джерело шуму може мати різні форми, наприклад, діод зі зворотним зміщенням, вакуумна лампа або транзистор, що працює в області лавинного або дробового шуму.

Вихідний сигнал джерела шуму потім підсилюється для досягнення бажаного рівня сигналу. Після підсилення шумовий сигнал може піддаватися фільтрації для формування його частотних характеристик. Фільтри можуть застосовуватися для обмеження смуги пропускання шумового сигналу або для виділення певних частотних діапазонів.

Крім того, деякі генератори шумових сигналів мають можливість модуляції, що дозволяє модулювати шумовий сигнал зовнішнім сигналом або керуючою напругою. Ця можливість модуляції додає універсальності генератору, дозволяючи йому створювати більш складні форми сигналів, змінюючи характеристики шумового сигналу в часі.

2.3.1 Характеристики шумового генератора

Переважно, генератори шуму демонструють різний рівень випадковості або контролю випадковості. Деякі генератори генерують дійсно випадкові сигнали, такі як білий шум, де кожна вибірка є незалежною і однаково розподіленою. Інші забезпечують контрольовану випадковість, дозволяючи користувачам налаштовувати такі параметри, як форма розподілу або кореляція між послідовними відліками.

Іншою характеристикою генераторів шуму є їхня вихідна амплітуда, яка може бути фіксованою або регульованою. Генератори з фіксованою амплітудою генерують шумові сигнали на заздалегідь визначеному рівні, забезпечуючи стабільність у тестуванні або експерименті. І навпаки, генератори з регульованою амплітудою пропонують гнучкість, дозволяючи користувачам змінювати рівень сигналу відповідно до їхніх конкретних вимог.

Крім того, генератори шуму можуть мати різні типи спектральних характеристик, що впливають на розподіл енергії сигналу по частотах. Наприклад, білий шум має плаский спектр, що означає, що він має однакову потужність на всіх частотах у межах своєї смуги пропускання. На відміну від нього, кольоровий шум, наприклад, рожевий або коричневий, відхиляється від плоского спектра, демонструючи специфічні частотні залежності.

Також, генератори шуму можуть містити функції для формування або фільтрації згенерованого шумового сигналу. Ці функції дозволяють користувачам пристосовувати характеристики шуму до конкретних застосувань або імітувати певні умови навколишнього середовища.

2.3.1.1 Білий шум

Генератор білого шуму в інформаційній безпеці - це пристрій або програмне забезпечення, призначене для створення випадкового, неповторюваного потоку даних.

Метою генератора білого шуму в інформаційній безпеці є підвищення рівня конфіденційності та безпеки шляхом маскування або приховування конфіденційної інформації або сигналів зв'язку.

На практиці генератори білого шуму можуть бути реалізовані в різних формах, включаючи апаратні пристрої, програмні додатки або алгоритми, вбудовані в комунікаційні протоколи або криптографічні системи. Вони широко використовуються в захищених системах зв'язку, генерації криптографічних ключів, шифруванні даних та електронних засобах протидії стеженню або перехопленню сигналів.

Генератор білого шуму працює, генеруючи випадкові зразки шуму і відтворюючи їх через аудіопристрій, наприклад, динаміки або навушники. Процес включає генерування випадкових чисел, їх квантування, щоб вони вкладалися в потрібний діапазон, а потім відтворення у вигляді звуку.

Генератор випадкових чисел виробляє послідовність чисел, які представляють амплітуду шумового сигналу в різні моменти часу. Ці випадкові числа зазвичай генеруються за допомогою алгоритмів, які створюють числа, розподілені рівномірно або відповідно до гаусівського (нормального) розподілу, що представлений у гаусівському білому шумі, приклад якого зображено на рис. 2.4 [30].

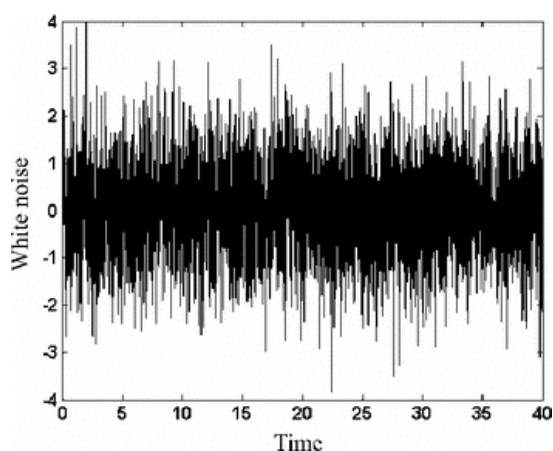


Рисунок 2.4 – Гаусівський білий шум

Після того, як випадкові числа згенеровані, вони квантуються відповідно до розрядності аудіосистеми. Наприклад, якщо аудіосистема має розрядність 16 біт, випадкові числа масштабуються і округлюються, щоб вписатися в діапазон від -32768 до 32767 (для знакових 16-бітних цілих чисел). Потім квантовані випадкові числа відтворюються через аудіопристрій.

Нарешті, аудіопристрій перетворює електричні сигнали на звукові хвилі, які слухач чує як білий шум. Швидкість, з якою відтворюються семпли, визначає частотний спектр і висоту звуку, що сприймається як білий шум.

На схемі на рис. 2.5 зображено, як ГВЧ генерує білий шум, створюючи випадкові вибірки. Потім ці випадкові зразки квантуються, щоб вписатися в потрібний діапазон. Квантовані зразки надсилаються на аудіопристрій (динаміки або навушники) для відтворення.



Рисунок 2.5 – Схема описаного генератора

В інформаційній безпеці генератор білого шуму використовується як механізм підвищення конфіденційності та безпеки шляхом введення випадкових даних або безглузвих сигналів у різні аспекти інформаційних систем. Він часто використовується для генерації криптографічних ключів та ВІ, що підвищує безпеку алгоритмів і протоколів шифрування.

2.3.1.2 Рожевий шум

Рожевий шум містить усі звукові частоти, які може чути людина. На відміну від білого шуму, який представляє всі частоти однаково, вищі частоти в рожевому шумі менш інтенсивні.

Рожевий шум - це "колір" шуму, як і білий шум. Кольори використовуються для опису частот, що містяться в шумі. Приклад виокристання зображено на рис. 2.6. Рожевий шум є плоским за логарифмічною шкалою. Це означає, що рожевий шум має відношення до акустики, оскільки він краще відображає те, як людина сприймає звук[31].

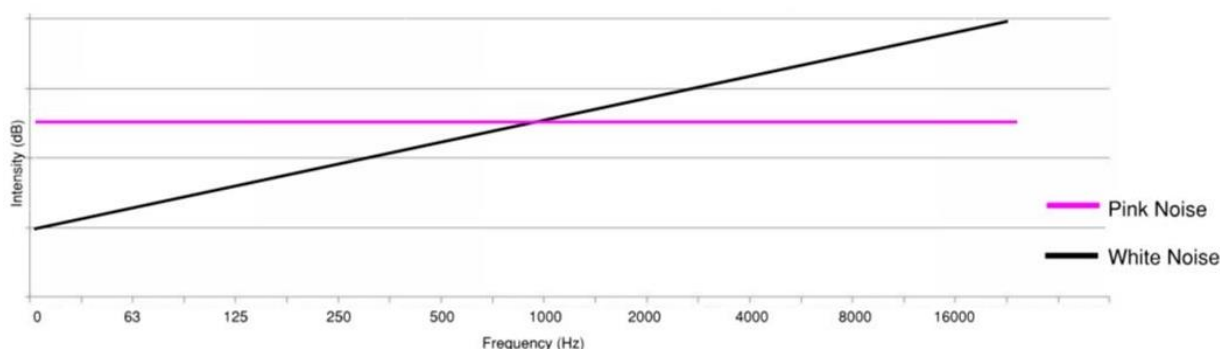


Рисунок 2.6 – Представлення білого та рожевого шумів

Деякі поширені приклади білого шуму - це пылесос, що працює, і телевізійні перешкоди. Ці звуки можна використовувати для блокування небажаних шумів, коли люди намагаються зосередитися або заснути. Проблема білого шуму полягає в тому, що всі частоти проникають з однаковою інтенсивністю. Рожевий шум намагається вирішити цю проблему. У рожевому шумі залишаються всі чутні звуки, але нижчі частоти посилюються, а вищі - послаблюються.

Рожевий шум визначається як випадковий сигнал, який має однакову енергію в кожній частоті октави. Іншими словами, рожевий шум має більше низькочастотних компонентів, ніж білий шум, і є фоновим шумом для багатьох звуків, які ми чуємо в житті.

У контексті інформаційної безпеки рожевий шум можна порівняти з комплексним і рівномірно розподіленим підходом до захисту даних. Він передбачає

впровадження заходів безпеки в усіх аспектах управління інформацією, від мережевої безпеки до протоколів шифрування, що гарантує, що захист буде цілісним і надійним. Подібно до того, як рожевий шум у звукотехніці допомагає маскувати перешкоди і покращує загальну якість звуку, підхід рожевого шуму в інформаційній безпеці спрямований на зменшення вразливостей і створення гармонійного середовища для безпечної роботи з даними.

2.3.2 Інтенсивність та розподіл амплітуд шуму

Інтенсивність амплітуд шуму відноситься до сили або величини шумових сигналів. У різних контекстах шум може проявлятися як небажані або випадкові коливання сигналу, наприклад, в аудіозаписах, електронних схемах або системах зв'язку.

Загалом, амплітуда є типовою змінною, що використовується для опису характеристик хвилі. Вона говорить про кількість енергії, що передається хвилею, і може розглядатися як міра того, наскільки сильною або великою є хвиля.

Амплітуда подається в безрозмірних одиницях децибелах (дБ) або як відношення до рівня сигналу (відношення сигнал/шум). Співвідношення між цими двома одиницями може дещо відрізнятись в різних приладах, але перерахунок на 0,5 дБ/відліки є гарною оцінкою.

Вища інтенсивність амплітуд шуму зазвичай означає більшу інтерференцію з потрібним сигналом і може погіршити якість або надійність сигналу. Тому в багатьох застосуваннях мінімізація інтенсивності шуму або поліпшення відношення має вирішальне значення для оптимальної продуктивності.

Інтенсивність амплітуди шуму в інформаційній безпеці відноситься до рівня небажаних сигналів або перешкод, які можуть порушити або затуманити законний зв'язок або передачу даних. Шум може приймати різні форми, такі як електромагнітні перешкоди, радіочастотні перешкоди або навіть навмисне глушіння зловмисниками.

Більше того, в цифровому зв'язку шум може бути введений злоумисниками з метою приховати конфіденційну інформацію або порушити роботу каналів зв'язку. Ця тактика часто використовується в стеганографії, де приховані повідомлення ховаються в нешкідливих на перший погляд даних, щоб уникнути виявлення заходами безпеки. Маніпулюючи амплітудою шуму, злоумисники можуть ускладнити алгоритмам безпеки розрізнення легітимних і шкідливих даних.

Білий шум, будучи неперервним сигналом, має неперервний розподіл як в часовій, так і в частотній областях. Спектральна щільність потужності білого шуму є постійною у всьому діапазоні частот, які він займає. Це означає, що енергія сигналу білого шуму рівномірно розподілена по всіх частотах, що дає підставу називати його "білим" шумом (оскільки біле світло містить всі частоти видимого спектру).

2.3.3 Кореляційні властивості шуму

Шум, у його незліченних формах, демонструє різноманітні кореляційні властивості, які відіграють вирішальну роль у різних галузях, починаючи від телекомунікацій і закінчуючи обробкою сигналів. Розуміння цих властивостей має першорядне значення для ефективного опису і зменшення шуму в практичних застосуваннях.

Кореляція - це статистична міра подібної поведінки між двома спостережуваними величинами, яка в нашому випадку показує, наскільки схожий переданий сигнал на сам себе через деякий час (коли сигнал отримано після відбиття). Кореляція виводиться у %, де 100% означає ідеальну кореляцію, а 0% - відсутність подібності. Величина кореляції, таким чином, є мірою якості даних про швидкість, і зі зменшенням кореляції зменшується і точність даних. Загальноприйнятий поріг дальності при розгляді кореляційних даних становить 50%. На рис. 2.7 показано, як можна обґрунтувати поріг у 50%, оскільки стандартне відхилення збільшується зі зменшенням кореляції.

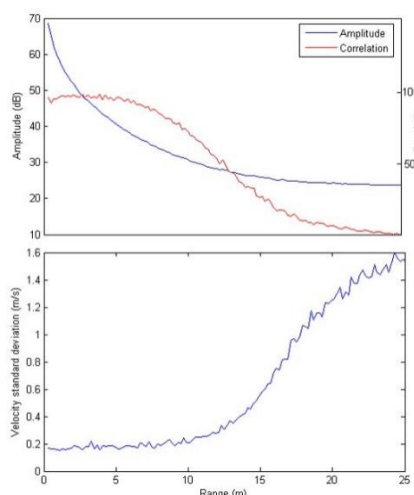


Рисунок 2.7 – Амплітуда та кореляція

На верхньому зображенні показано взаємодію між амплітудою та кореляцією. Нижче, стандартне відхилення. Варто звернути увагу, як збільшується стандартне відхилення зі зменшенням кореляції.

Як амплітуда, так і кореляція використовуються як параметри якості даних про швидкість, і обидва можуть бути використані для встановлення максимального діапазону. Оскільки вони базуються на різних речах, вони також чутливі до різних відхилень. Кореляція повинна впасти до 50% приблизно там, де відношення досягає порогового значення 3 дБ. Якщо це не так, вважайте це ознакою того, що дані потребують ретельнішого аналізу. Однак, не всі прилади вимірюють кореляцію, для них основна увага повинна бути зосереджена на амплітуді.

Часова кореляція, що спостерігається в даних часових рядів, означає зв'язок між точками даних, розділеними різними часовими інтервалами. Ця кореляція може бути кількісно оцінена за допомогою таких метрик, як автокореляційні функції, що з'ясовують, як минулі значення сигналу впливають на його майбутню поведінку.

Крім того, кореляція шуму може демонструвати частотно-залежні характеристики, особливо в сценаріях, що включають сигнали з різним спектральним розподілом.

Таким чином, кореляційні властивості шуму охоплюють широкий спектр, що включає різні часові, просторові і частотні області. Розуміння цих властивостей дозволяє фахівцям розробляти надійні системи, оптимізувати алгоритми обробки

сигналів і підвищувати надійність і продуктивність широкого спектру технологічних застосувань.

Висновки до розділу 2

Досліджуючи принципи побудови та функціонування різних генераторів, ми заглиблюємося в складні конструкції та операційні структури, які лежать в основі їхньої функціональності. У цьому розділі було розглянуто два різних типи генераторів: математичних генераторів і генераторів шумових сигналів.

Математичні генератори, як випливає з назви, ґрунтуються на математичних принципах, використовуючи алгоритми для генерації послідовностей чисел або математичних функцій. Ці генератори слугують для широкого спектру цілей у різних галузях, від криптографії до наукового моделювання. Побудова математичного генератора передбачає ретельний відбір і реалізацію алгоритмів, пристосованих до конкретних вимог. Розуміючи основні математичні концепції та алгоритми, можна розробляти генератори, здатні створювати послідовності з бажаними властивостями, такими як випадковість та повторюваність.

З іншого боку, генератори шумових сигналів відіграють вирішальну роль у різних сферах застосування, включаючи телекомунікації, аудіотехніку та обробку сигналів. Ці генератори виробляють сигнали, що характеризуються випадковими коливаннями, імітуючи природні явища, такі як тепловий шум або хаотичні процеси. Побудова генераторів шумових сигналів вимагає складної схемотехніки, здатної генерувати випадкові або псевдовипадкові сигнали в межах заданих параметрів. Ретельно розробляючи схему та підбираючи відповідні компоненти, можна створювати генератори, які точно імітують реальні джерела шуму, пропонуючи при цьому контроль над ключовими параметрами, такими як амплітуда.

Поєднуючи знання, отримані з принципів побудови та функціонування як математичних, так і шумових генераторів сигналів, ми отримуємо всебічне розуміння основних механізмів, що керують їхньою роботою. Наступний розділ, у

якому будуть описані алгоритми, що керують цими генераторами, дасть змогу зрозуміти на практиці та прояснити конкретні обчислювальні процеси, що беруть участь у створенні послідовностей або сигналів. Поєднуючи теорію з практичною реалізацією, цей цілісний підхід надає інструменти, необхідні для проектування, оптимізації та використання генераторів, пристосованих до їхніх конкретних потреб і застосувань, зокрема і в інформаційній безпеці.

РОЗДІЛ 3. АЛГОРИТМІЗАЦІЯ АПАРАТНОГО ТА МАТЕМАТИЧНОГО ГЕНЕРАТОРІВ

3.1 Апаратний генератор: генератор лінійного рекурентного регістру

У попередньому розділі розглядався такий вид генераторів, як НГВБ. Зазначалося, що це алгоритми або пристрої, які генерують випадкові біти, не покладаючись на заздалегідь визначену послідовність або шаблон. У свою чергу, генератор лінійного рекурентного регістру (ГЛРР) є безпосереднім прикладом НГВБ.

ГЛРР - це тип цифрової схеми, що використовується для генерування псевдовипадкових послідовностей. Він широко використовується в криптографії, цифровій обробці сигналів і моделюванні. ГЛРР складаються з регістрів зсуву, вхідний біт яких є лінійною функцією його попереднього стану, що показує рис. 3.1. Цей механізм лінійного зворотного зв'язку створює послідовність бітів, яка здається випадковою, але в кінцевому підсумку періодично повторюється через обмежену довжину регістра зсуву[32].

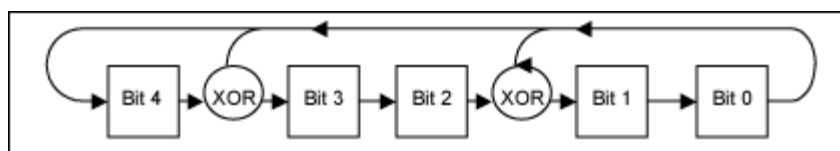


Рисунок 3.1 - Спрощене зображення ГЛРР[33].

3.1.1 Принцип роботи

ГЛРР повністю задається своїм поліномом. Наприклад, многочлен 6-го степеня з усіма присутніми членами описується рівнянням $x^6 + x^5 + x^4 + x^3 + x^2 + x + 1$. Існує $2^{6-1} = 32$ різних можливих многочленів цього розміру. Як і у випадку з числами, деякі многочлени є простими або примітивними. Нас цікавлять примітивні поліноми, тому що вони дають нам періоди максимальної довжини при

зсуві. Поліном максимальної довжини степеня n матиме $2n - 1$ різних станів. Після кожного зсуву відбувається перехід у новий стан. Отже, поліном 6-го степеня матиме 31 стан. Кожне число від 1 до 31 буде зустрічатися у регістрі зсуву перед тим, як воно повториться. У випадку примітивних поліномів 6-го степеня їх лише шість. У свою чергу, маска полінома створюється шляхом взяття двійкового представлення полінома та відсікання крайнього правого біта. Маска використовується у коді, який реалізує многочлен. Для реалізації маски многочлена для полінома n -го степеня потрібно n бітів.

Кожен примітивний поліном матиме непарну кількість членів, що означає, що кожна маска для примітивного полінома матиме парну кількість 1 біт. Кожен примітивний поліном також визначає другий примітивний поліном, який є його дуалом. Дуал можна знайти, віднявши від степеня полінома показник степеня для кожного члена. Наприклад, для многочлена 6-го степеня, $x^6 + x + 1$, його дуал буде $x^6 - 6 + x^6 - 1 + x^6 - 0$, що дорівнює $x^6 + x^5 + 1$.

ГЛРР ніколи не можуть мати нульового значення, оскільки кожне зміщення обнуленого ГЛРР залишатиме його нульовим. ГЛРР повинен бути ініціалізований, тобто, заповнений ненульовим значенням. Коли ГЛРР містить 1 і зсувається один раз, його значення завжди буде дорівнювати значенню поліноміальної маски. Якщо у регістрі всі нулі, окрім старшого біта, то наступні декілька зсувів покажуть зсув старшого біта до молодшого з заповненням нулем. Наприклад, будь-який 8-бітний зсувний регістр з примітивним поліномом зрештою згенерує послідовність $0 \times 80, 0 \times 40, 0 \times 20, 0 \times 10, 8, 4, 2, 1$, а потім маску полінома.

ГЛРР характеризуються поліномом зворотного зв'язку, який визначає позиції перемикавання, де вихід регістра подається назад на вхід. Довжина регістра зсуву і положення відводів визначають період і статистичні властивості згенерованої послідовності. ГЛРР відносно прості в апаратній реалізації і ефективні з точки зору як площі, так і енергоспоживання, що робить їх популярним вибором для генерації псевдовипадкових послідовностей в різних додатках.

Загалом, базова ГЛРР не дає дуже хороших випадкових чисел. Кращу послідовність чисел можна покращити, вибравши більший ГЛРР і використавши

молодші біти для випадкового числа. Наприклад, якщо ви маєте 10-бітовий ГЛРР і хочете отримати 8-бітне число, ви можете взяти молодші 8 біт регістра для вашого числа. За допомогою цього методу ви побачите кожне 8-бітне число чотири рази, а нуль - три рази, перш ніж ГЛРР закінчить один період і повториться. Це вирішує проблему отримання нулів, але все одно числа мають не дуже хороші статистичні властивості. Замість цього можна використати підмножину ГЛРР для випадкового числа, щоб збільшити кількість перестановок чисел і покращити випадкові властивості виходу ГЛРР.

Зсув ГЛРР більше одного разу перед отриманням випадкового числа також покращує його статистичні властивості. Зсув ГЛРР на фактор його періоду зменшує загальну довжину періоду на цей фактор.

Відносно короткі періоди ГЛРР можна вирішити, об'єднавши значення двох або більше різних за розміром ГЛРР разом. Новий період цих ГЛРР, об'єднаних за допомогою операції XOR, буде НСК періодів. Наприклад, НСК примітивного 4-бітового та примітивного 6-бітового ГЛРР дорівнює НСК(15, 63), що дорівнює 315. При об'єднанні ГЛРР таким чином, переконайтеся, що ви використовуєте лише мінімальну кількість бітів ГЛРР; краще використовувати менше. Для 4- та 6-бітових ГЛРР слід використовувати не більше молодших 4 бітів.

Непередбачуваність ГЛРР можна збільшити, якщо додати до періоду зворотного зв'язку трохи "ентропії". При цьому слід бути обережним - існує невелика ймовірність того, що ГЛРР обнулиться до нуля з додаванням біта ентропії. Обнулення ГЛРР виправиться, якщо періодично додавати ентропію. Цей метод XOR біта зі зворотним зв'язком є способом обчислення циклічних перевірок на надлишковість.

Поліноми не створюються рівними. Деякі поліноми однозначно будуть кращими за інші.

Найчастіше ГЛРР використовується у криптографії. Розглянемо алгоритм генератора на прикладі.

Алгоритм ГЛРР, в основному, містить такі основні етапи: ініціалізація, операція зсуву, повернення.

Опис алгоритму:

1. Ініціалізація:

Алгоритм починається з ініціалізації регістра зсуву початковим значенням, зазвичай ненульовим двійковим числом, зображене на рис. 3.2[34]. Це початкове значення представляє початковий стан ГЛРР.

Далі алгоритм визначає поліном зворотного зв'язку, який визначає відгалуження (біти, які потрібно об'єднати за допомогою XOR) у регістрі зсуву. Поліном зворотного зв'язку визначає, які біти зсувного регістра об'єднуються для отримання наступного біта.

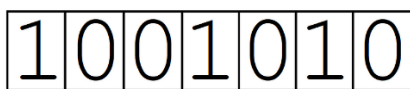


Рисунок 3.2 – Початкове значення регістру зсуву

2. Операція зсуву:

З технічної точки зору, зсувні регістри відносяться до області цифрових схем і найкраще пояснюються в цьому контексті, однак, в даному випадку надається розуміння в рамках криптографії.

Під час кожної ітерації алгоритм зсуває всі біти в регістрі вправо (або вліво) на одну позицію. Цей процес зсуву переміщує біти вздовж регістра.

Після зсуву алгоритм обчислює наступний біт, який повинен увійти в регістр. Цей розрахунок базується на поліномі зворотного зв'язку, який визначає, які біти регістру об'єднуються в операцію XOR. Результатом операції XOR є наступний біт, який буде занесено в регістр.

Як тільки наступний біт обчислено, алгоритм вставляє його в крайню ліву (або крайню праву) позицію регістра. Цей новий біт стає частиною стану регістра для наступної ітерації.

Візьмемо 7-бітний регістр, приклад якого наводився вище, і уявімо його як бітовий масив. Тепер зсуваємо значення в регістрі вправо. Крайній лівий біт переміститься вправо і стане другим зліва. Другий зліва переміститься і стане

третім зліва. Це відбувається до тих пір, поки ми не досягнемо останнього значення в нашому регістрі.

Це останнє значення "зсувається" з регістру і може бути поглинуте зовнішнім процесом. Уявімо, що це використовується для генерування бітового потоку. У наведеному нижче прикладі спочатку отримали 0, потім 1, потім 0 і так далі. У прикладі також просто припускається, що нулі заповняють порожнє місце, яке з'явиться зліва після зсуву, що наглядно показано на рис. 3.3:

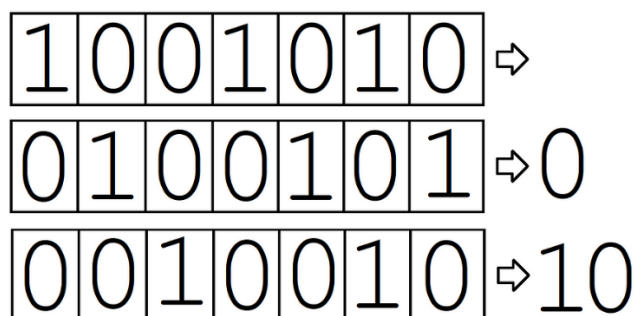


Рисунок 3.3 – Зсув з регістру

3. Виведення:

На даному етапі впроваджується механізм зворотного зв'язку. Вихід ГЛРР зазвичай береться з одного або декількох бітів у регістрі, залежно від вимог програми. Ці вихідні біти можна використовувати як псевдовипадкові числа в різних контекстах.

Для цього виконується XOR деяких значень у нашому зсувному регістрі і використовується результат як вхідний сигнал, щоб не завжди записувалися нулі. У середовищі ГЛРР місця в регістрі зсуву, які будуть використовуватися для XOR, називаються тапами ("taps" - перемикачі). У прикладі нижче показано, що ми "тапнули" чотири біти, читаючи зліва направо, ми використовуємо біти з індексами 1, 2, 4 і 6 відповідно до рис. 3.4.

І як можна помітити з прикладу, перший біт заміни - це нуль, а другий біт - це одиниця. Таким чином, ми отримаємо принаймні трохи цікавіший потік бітів, ніж раніше.

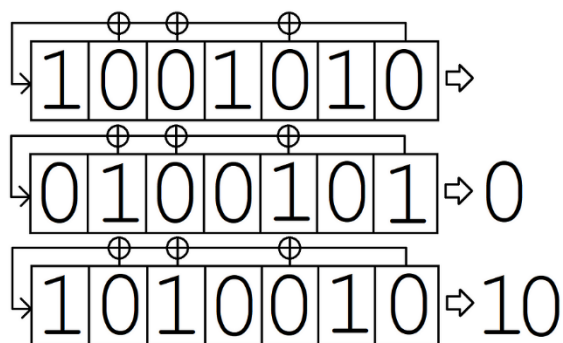


Рисунок 3.4 – Процес виведення

ГЛРР є детермінованими. Це означає, що, маючи початковий потік і статичний набір натискань, ви завжди отримаєте один і той самий бітовий потік. Іншим наслідком цього факту є те, що ми врешті-решт досягнемо повторюваного циклу внутрішніх станів. Коли ми повернемося до значення регістра, яке ми вже бачили, ми будемо повторювати ту ж саму серію зсувів і обчислень зворотного зв'язку. Період LFSR - це кількість унікальних внутрішніх станів, які ми можемо відвідати перед повторенням.

У криптографії зазвичай ГЛРР використовується для генерації саме псевдовипадкових бітових потоків. Через це, в ідеалі, не хотілося, щоб така поведінка повторюваного циклу мала місце.

3.1.2 Застосування ГЛРР

Лінійні регістри зсуву зі зворотним зв'язком знаходять застосування в різних галузях, включаючи уже згадану криптографію для шифрування, генерацію послідовностей для генерації ключів, а також виявлення та виправлення помилок при передачі даних.

Ці регістри широко використовуються в криптографії для генерації псевдовипадкових послідовностей, які відіграють вирішальну роль у забезпеченні безпечної передачі даних. У контексті генерації ключів ГЛРР сприяють створенню унікальних і непередбачуваних криптографічних ключів, які необхідні для шифрування конфіденційної інформації.

У передачі даних ГЛРР допомагають виявляти і виправляти помилки, які можуть виникнути в процесі надсилання та отримання даних, підвищуючи тим самим надійність і точність інформації, що передається.

У сфері кібербезпеки реєстри зсуву зі зворотним зв'язком відіграють вирішальну роль у процесах шифрування, генеруючи псевдовипадкові послідовності, які є основою криптографічних алгоритмів і захищених протоколів зв'язку.

Ці псевдовипадкові послідовності, що генеруються реєстрами зсуву зі зворотним зв'язком, використовуються як ключі для перетворення відкритих даних у зашифрований текст, роблячи його нечитабельним для сторонніх осіб. Цей процес скремблювання, відомий як шифрування, є фундаментальним для захисту конфіденційної інформації під час передачі або зберігання.

Криптографічні алгоритми використовують непередбачувану природу цих послідовностей для підвищення безпеки комунікацій, забезпечуючи конфіденційність і цілісність. Інтегруючи реєстри зсуву зі зворотним зв'язком у криптографічні примітиви, організації можуть створити надійні механізми шифрування, які протистоять різним кіберзагрозам і запобігають потенційним атакам на конфіденційність і безпеку даних.

3.1.3 Переваги та недоліки ГЛРР

Використання реєстрів зсуву зі зворотним зв'язком у кібербезпеці має такі переваги, як швидка обробка, простота конструкції та універсальність, але також створює проблеми, такі як вразливість до атак та обмежена періодичність генерації послідовностей.

Швидкість, з якою працюють реєстри зсуву лінійного зворотного зв'язку, є ключовою перевагою, що забезпечує швидкі процеси шифрування та дешифрування, необхідні для безпечного зв'язку. Їх проста структура полегшує реалізацію порівняно з більш складними алгоритмами, що може бути перевагою, особливо в сценаріях, де ефективність є пріоритетом.

Вразливість до атак становить значний ризик, оскільки простота ГЛРР може бути використана кіберзлочинцями. Обмежена періодичність генерації послідовностей може призвести до передбачуваності, що робить шифрування вразливим до атак на основі шаблонів.

Переваги:

Лінійні регістри зсуву зі зворотним зв'язком мають переваги в кібербезпеці завдяки швидкій та ефективній обробці даних, спрощеній конструкції, що полегшує реалізацію, та універсальності, придатній для різних криптографічних застосувань.

Ці регістри працюють на високих швидкостях, що робить їх ідеальними для криптографічних завдань у реальному часі, таких як шифрування та дешифрування. Їх ефективна обробка дозволяє швидко і точно обробляти великі обсяги даних, що має вирішальне значення для захисту конфіденційної інформації. Простота конструкції лінійних регістрів зсуву зі зворотним зв'язком спрощує інтеграцію в існуючі системи, зменшуючи складність криптографічних реалізацій. Їх універсальність дозволяє адаптувати їх для широкого спектру криптографічних функцій, підвищуючи загальну безпеку системи.

Недоліки:

Незважаючи на свої переваги, лінійні регістри зсуву зі зворотним зв'язком схильні до вразливостей, які можуть призвести до атак, і вони мають обмеження в генерації послідовностей з довгою періодичністю, що створює проблеми з кіберзагрозами.

Одним з ключових недоліків регістрів зсуву лінійного зворотного зв'язку в кібербезпеці є їхня вразливість до атак через детермінований характер їхнього виходу. Ця передбачуваність робить їх вразливими до методів криптоаналізу, де зловмисники можуть потенційно використовувати закономірності у згенерованих послідовностях, щоб порушити безпеку системи. Обмежена періодичність генерації послідовностей регістрів зсуву лінійного зворотного зв'язку може обмежити складність і випадковість вихідних даних, що полегшує зловмисникам прогнозування і маніпулювання згенерованими послідовностями в зловмисних цілях. Ці фактори в сукупності призводять до ослаблення механізмів захисту від

кіберзагроз у системах, що використовують регістри зсуву лінійного зворотного зв'язку.

3.2 ГПВЧ на основі математичного алгоритму: вихор Мерсенна.

Вихор Мерсенна - один з найпоширеніших ГПВЧ. Його розробили Такудзі Нішімура та Макото Мацумото у 1997 році.

У ГПВЧ Мерсенна використовується початкове значення довжиною 19 937 біт, яке зберігається в масиві з 624 елементів, що містять 32-розрядні значення. Таку назву він отримав через те, що має період $2^{19937}-1$, який називається простим числом Мерсенна.

Просте число Мерсенна - це будь-яке просте число, що визначається як $M_n = 2^n - 1$. Його назва походить від імені Маріна Мерсенна, французького монаха-мініма, який вивчав їх на початку 17-го століття.

3.2.1 Принцип роботи

Алгоритм обертається навколо двох основних функцій, які описані нижче:

1. Спершу ініціалізується стан відповідно до початкового значення.
2. Функція генерування випадкових чисел витягує одне ціле число з попередньо визначеного стану, робить деякий бітовий зсув і генерує число. Після використання всіх цілих числових станів (623) алгоритм виконує операцію "перезавантаження", перекручуючи існуючі цілі числа і генеруючи новий набір числових станів.

Цілочисельна частина алгоритму вихору Мерсенна не передбачає жодних арифметичних дій, таких як додавання, віднімання, множення або ділення. Всі операції - це зсуви, and, or та xor.

Всі елементи стану, окрім останнього, є беззнаковими 32-бітними випадковими цілими числами, які утворюють кеш, що ретельно генерується при

запуску. Ця генерація запускається за допомогою seed - єдиного цілого числа, яке ініціює весь процес.

Останній елемент стану - це вказівник на кеш. Кожен запит випадкового цілого числа призводить до вилучення елемента з кешу та збільшення вказівника. Елемент "загартується" додатковими логічними операціями для покращення випадковості. Коли вказівник досягає кінця кешу, кеш поповнюється ще 624 елементами.

Алгоритм проаналізовано за допомогою дослідження теоретико-групових властивостей операцій перестановки та темперування. Параметри підбрано так, щоб період був простим числом Мерсенна $2^{19937}-1$. Цей період набагато більший, ніж у будь-якого іншого генератора випадкових чисел, запропонованого до або після нього, і є однією з причин популярності МТ.

За задумом, результати, що генеруються, задовольняють властивості рівномірного розподілу у 623-вимірному кубі.

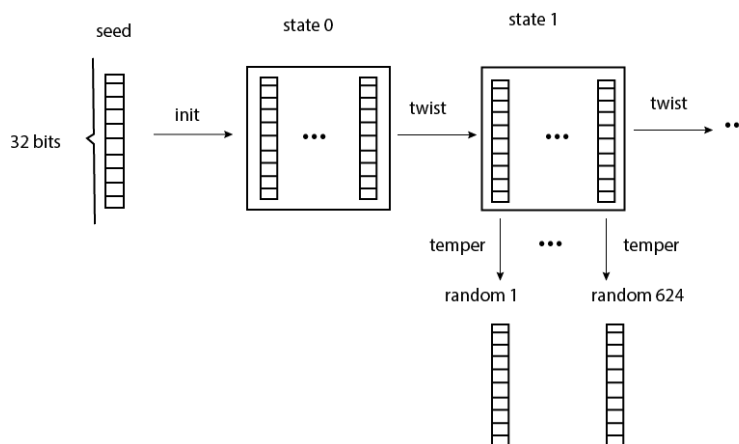


Рисунок 3.5 – Графічне представлення роботи алгоритму вихору Мерсенна

На схемі рис. 3.5 показано, що алгоритм починає роботу з seed (при повторному використанні того ж самого seed ви отримаєте ті ж самі випадкові числа), ви ініціалізуєте його в state (стан). Потім, кожного разу, коли ви хочете отримати випадкове число, ви перетворюєте цей state за допомогою односторонньої функції g. Це робиться тому, що ви не хочете, щоб люди дізналися про стан за випадковим виходом.

Якщо ви хочете отримати інше випадкове число, ви спочатку перетворюєте state за допомогою односторонньої функції f : це тому, що ви не хочете, щоб люди, які дізналися state, могли відновити попередні states (пряма секретність). А потім ви знову використовуєте функцію g , щоб вивести випадкове число.

Вихор Мерсенна схожий на це, за винятком:

1. Ваш перший стан не використовується для виведення випадкових чисел
2. State дозволяє вам вивести не одне, а 624 випадкових числа (хоча це можна вважати одним великим випадковим числом) функція g є оборотною, вона не є односторонньою функцією, тому вихор не є криптографічно безпечним ГПВЧ.

3.2.2 Застосування вихору Мерсенна

CryptMT - це потоковий шифр і криптографічно захищений генератор псевдовипадкових чисел, який використовує вихор Мерсенна. Його розробили Мацумото і Нішімура разом з Маріко Хагіта і Муцуо Сайто. Він був представлений в проєкті eSTREAM мережі eCRYPT. На відміну від вихору Мерсенна або інших його похідних, CryptMT запатентований[35].

MTGP - це варіант вихору Мерсенна, оптимізований для графічних процесорів, опублікований Муцуо Сайто і Макото Мацумото. Базові операції лінійної рекурсії розширені з MT, а параметри підібрані таким чином, щоб дозволити багатьом потокам обчислювати рекурсію паралельно, розділяючи при цьому свій простір станів для зменшення навантаження на пам'ять. У статті стверджується про покращений рівномірний розподіл порівняно з MT і продуктивність на старому графічному процесорі (Nvidia GTX260 з 192 ядрами) у 4,7 мс для 5×10^7 випадкових 32-бітних цілих чисел[36].

SFMT (SIMD-орієнтований Fast Mersenne Twister) - це варіант Mersenne Twister, представлений у 2006 році, розроблений для швидкої роботи на 128-бітній SIMD. Він приблизно вдвічі швидший за вихор Мерсенна, має кращу властивість рівномірного розподілу v -бітної точності, ніж вихор[37].

Також, він швидше відновлюється з початкового стану з нульовим надлишком, ніж вихор. SFMT підтримує процесори Intel SSE2 та PowerPC AltiVec. Також використовується для ігор з Cell BE в PlayStation 3.

TinyMT – це ще один варіант вихору Мерсенна, запропонований Сайто та Мацумото у 2011 р. TinyMT використовує лише 127 біт простору станів, що значно менше порівняно з 2,5 кілобайтами оригіналу. Однак він має період $2^{127} - 1$, набагато коротший за оригінал, тому автори рекомендують його лише у випадках, коли пам'ять на висоті[38].

Хоча сам по собі вихор не є криптографічно безпечним, він все ж може знайти застосування в певних завданнях кібербезпеки.

Одне з таких застосувань - генерація ключів. Криптографічні алгоритми часто вимагають випадкових ключів для шифрування і дешифрування. У цьому випадку вихор Мерсенна може бути використаний для генерації таких ключів. Однак, щоб підвищити безпеку згенерованих ключів, дуже важливо завантажувати генератор дійсно випадковим значенням, наприклад, поточним часом.

Крім того, при хешуванні паролів використовується соління, щоб запобігти атакам типу "райджних таблиць". Вихор Мерсенна може допомогти згенерувати випадкові значення солі, додаючи додатковий рівень безпеки, гарантуючи, що навіть користувачі з однаковими паролями матимуть різні хешовані паролі завдяки унікальним значенням солі.

У криптографічних протоколах, таких як AES, кожна операція шифрування вимагає унікального ВІ. Твістер Мерсенна може слугувати для генерування цих випадкових векторів, підвищуючи безпеку процесу шифрування.

Негативний досвід вихору має його застосування атакою проти України. 23 лютого 2022 року деструктивна кампанія з використанням HermeticWiper була спрямована проти кількох українських організацій. Ця кібератака за кілька годин передувала початку вторгнення в Україну військ російської федерації.

Ці руйнівні атаки використовували принаймні три компоненти:

HermeticWiper : робить систему неприцездатною, пошкоджуючи її дані

HermeticWizard : поширює HermeticWiper у локальній мережі через WMI та SMB

HermeticRansom : програма-вимагач, написана на Go.

Першочергово, зловмисники користувалися так званим IsaacWiper. Він починає з перерахування фізичних дисків і викликає DeviceIoControl за допомогою IOCTL IOCTL_STORAGE_GET_DEVICE_NUMBER, щоб отримати їхні номери пристроїв. Потім він витирає перші 0x10000 байт кожного диска за допомогою генератора псевдовипадкових чисел на основі вихору Мерсенна. Генератор запускається за допомогою значення GetTickCount. Саме у цій атаці вихор Мерсенна перераховує логічні диски і рекурсивно витирає кожен файл кожного диска випадковими байтами. Цікаво відзначити, що вона рекурсивно витирає файли в один потік, а це означає, що для витирання великого диска знадобиться багато часу[39].

3.2.3 Переваги та недоліки вихору Мерсенна

Позитивним моментом є те, що вихор Мерсенна чудово генерує високоякісні випадкові числа. Його статистичні властивості є надійними, що робить його придатним для широкого спектру застосувань, які потребують випадковості. Крім того, його ефективність у швидкому створенні великих партій випадкових даних є перевагою для задач, що вимагають значних обсягів випадкової інформації.

Інша значна перевага полягає в його великому періоді, який зазвичай досягає $2^{19937} - 1$. Такий великий період гарантує, що твістер Мерсенна може генерувати величезну кількість унікальних випадкових послідовностей перед повторенням, що виявляється цінним у різних сценаріях.

Крім того, простота реалізації та сумісність з різними мовами програмування роблять його доступним для розробників з різним досвідом. Також, його детермінований вихід, коли однакові початкові дані завжди призводять до однакової послідовності випадкових чисел, полегшує тестування та налагодження.

Однак, поряд з цими перевагами, вихор Мерсенна має суттєві недоліки. Знаючи початкове число, злоумисник може передбачити всі наступні випадкові числа, що генеруються, що ставить під загрозу безпеку.

Більше того, незважаючи на велику довжину періоду, вихор Мерсенна не позбавлений періодичності. Зрештою, після генерації досить великого набору випадкових чисел, послідовність буде повторюватися, що потенційно обмежує її корисність у сценаріях, які вимагають надзвичайно довгих унікальних послідовностей.

Якість випадковості сильно залежить від початкового значення, обраного для ініціалізації. Поганий вибір початкового значення може призвести до того, що послідовність випадкових чисел буде недостатньо статистично стійкою.

Також, хоча в певних реалізаціях вихору може бути потокобезпечним, у багатопотоковому середовищі необхідно бути обережним, щоб забезпечити належну синхронізацію і запобігти проблемам, що виникають при одночасному доступі до генератора

3.3 Порівняльна характеристика двох генераторів

ГЛРР та вихор Мерсенна - це два широко використовувані, які мають свої переваги та недоліки. Проаналізувавши апаратний та математичний генератор, сформовано порівняльну характеристику.

Обидва генератори мають низку спільних характеристик:

Детермінованість: генерують послідовності на основі початкового значення.

Проста конструкція: легкість реалізації та інтеграції.

Універсальність: підходять для різних криптографічних функцій.

Нестійкість до атак: вразливість до криптоаналізу.

Невелика потреба в пам'яті.

Повна характеристика зазначена у таб.3.1:

Таблиця 3.1 – Порівняльна характеристика двох генераторів

Характеристика	ГЛРР	Вихор Мерсенна
Тип	Апаратний	Математичний
Детермнованість	+	+
Алгоритм	Лінійний регістр зсуву з зворотним зв'язком	Алгоритм Мерсенна
Початкове значення	Масив з 624 32-бітних цілих чисел	19 937-бітове ціле число
Період	Обмежений довжиною регістра зсуву	$2^{19937}-1$
Якість випадкових чисел	Не дуже хороша	Хороша
Швидкість	Дуже швидка	Швидка
Простота	Проста конструкція	Проста конструкція
Універсальність	Може використовуватися для різних криптографічних функцій	Може використовуватися для різних криптографічних функцій
Стійкість до атак	Не стійкий до криптографічних атак	Не стійкий до криптографічних атак
Потокобезпечність	Не потокобезпечний	Може бути потокобезпечним
Пам'ять	Невелика	Невелика

З відмінностей можна зазначити те, що ГЛРР має обмежений період, тоді як вихор Мерсенна має дуже довгий період. Якість випадкових чисел ГЛРР не дуже хороша, тоді як якість випадкових чисел вихору Мерсенна хороша. ГЛРР не потокобезпечний, тоді як вихор Мерсенна може бути потокобезпечним.

ГЛРР і вихор Мерсенна - це два популярні ГПВЧ, які мають свої переваги та недоліки. ГЛРР є більш простим і швидшим, але має обмежений період і не дуже хорошу якість випадкових чисел. Вихор Мерсенна має більш довгий період і кращу якість випадкових чисел, але він більш складний і може бути не потокобезпечним.

На основі спільних характеристик був створений алгоритм для гібридного генератора, який поєднує в собі всі сильні сторони. Алгоритм представлено на рис. 3.6:

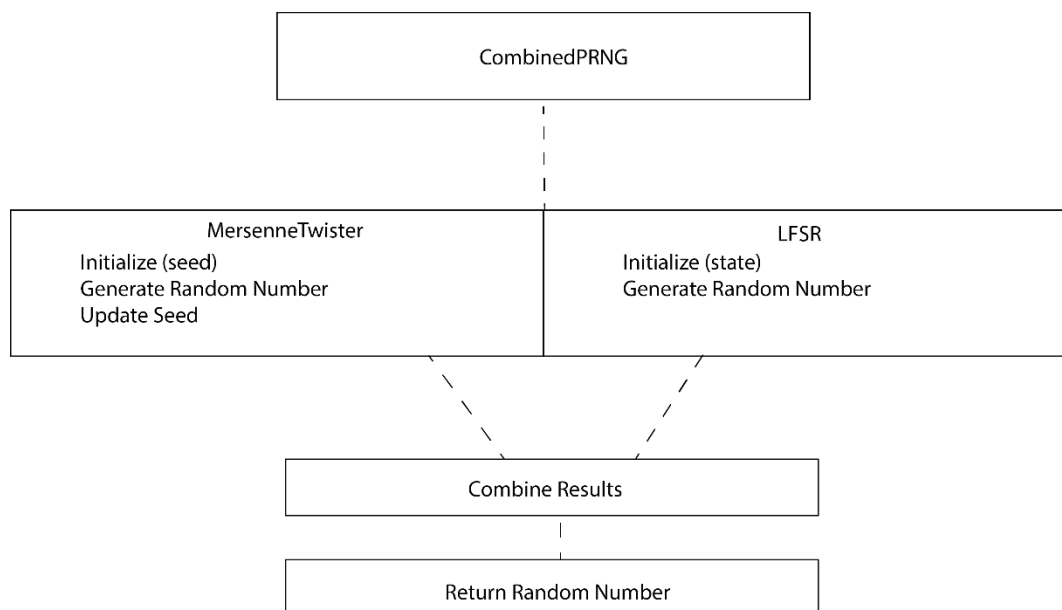


Рисунок 3.6 – Алгоритм гібридного генератора

1. Ініціалізація

Створити екземпляри класів `MersenneTwister` та `LFSR` всередині конструктора `CombinedPRNG`:

`MersenneTwister` використовує поточний час як початкове значення ("seed"). `LFSR` має попередньо визначений початковий стан.

2. Генерація випадкового числа

Метод `generateRandomNumber` викликає методи `random` обох внутрішніх PRNG. Отримає випадкове число від кожного генератора (MT та LFSR). Додає ці два випадкові числа, щоб отримати об'єднаний результат.

Висновки до розділу 3

У розділі було детально розглянуто алгоритм та функціонал генераторів двох видів, а саме апаратний та математичний. Незважаючи на недоліки, зокрема, криптостійкість, генератори мають широкий спектр застосування, і активно використовуються навіть в атаках.

Дослідження алгоритмізації як апаратних, так і математичних генераторів псевдовипадкових чисел дає цінне розуміння складного балансу між ефективністю,

випадковістю та обчислювальною складністю. У той час як апаратні генератори відрізняються швидкістю і низькорівневою інтеграцією, математичні алгоритми забезпечують гнучкість і відтворюваність.

Маючи перелік сильних сторін, теоретично, можна сформуванати генератор-гібрид. Подібна реалізація представлена у наступному розділі. Задумана реалізація поєднання генератора вимагає певного досвіду та ретельної реалізації. У більшості випадків безпечнішим і надійнішим варіантом є використання добре зарекомендованих бібліотек, що пропонують безпечну випадковість. Проте, в експериментальній частині представлено саме генератор-гібрид, який в теорії має місце на існування.

РОЗДІЛ 4. СИСТЕМА ГЕНЕРАЦІЇ ПСЕВДОВИПАДКОВИХ КЛЮЧОВИХ ПОСЛІДОВНОСТЕЙ

Хоча поєднання двох ГПВЧ теоретично може покращити випадковість, воно вимагає обережного застосування, щоб уникнути послаблення загального ефекту. Було розроблено об'єднаний спосіб за допомогою Java Script для поєднання вихору Мерсенна та лінійного рекурентного регістру, який на практиці застосовано як генерування ключів шифрування та електронні підписи для органу місцевого самоврядування.

У коді визначено дві функції, що представляють ГПВЧ, і третю функцію, яка об'єднує їхні результати. Важливо зазначити, що це спрощена програмна реалізація, яка в подальшому може бути модифікована для більш серйозних цілей.

У цій роботі описується комбінований ГПВЧ, який використовує два алгоритми: Mersenne Twister та LFSR. Цей ГПВЧ може бути використаний у різних сферах, де потрібні псевдовипадкові числа, зокрема, і у криптографії.

1. Клас CombinedPRNG

Цей клас представляє комбінований ГПВЧ. Він ініціалізує два об'єкти: `mt` (Mersenne Twister) та `lfsr` (LFSR). Для `mt` використовується поточний час як початкове значення.

Метод `generateRandomNumber` генерує нове псевдовипадкове число, використовуючи комбінацію `mtRandom` та `lfsrRandom`. `mtRandom` генерується за допомогою Mersenne Twister, а `lfsrRandom` - за допомогою LFSR. Потім ці два числа сумуються та повертаються, що демонструє рис. 4.1:

```

class CombinedPRNG {
    /**
     * Конструктор класу CombinedPRNG.
     */
    constructor() {
        this.mt = new MersenneTwister(); // Ініціалізація ГПВЧ вихору Мерсенна
        this.lfsr = new LFSR(); // Ініціалізація ГЛРР
    }

    /**
     * Генерує нове псевдовипадкове число, використовуючи комбінацію Mersenne Twister та LFSR.
     *
     * @returns {number} Нове псевдовипадкове число.
     */
    generateRandomNumber() {
        const mtRandom = this.mt.random(); // Генерація випадкового числа за допомогою Mersenne Twister
        const lfsrRandom = this.lfsr.random(); // Генерація випадкового числа за допомогою LFSR
        return mtRandom + lfsrRandom; // Об'єднання двох чисел
    }
}

```

Рисунок 4.1 – Елемент коду з класом CombinedPRNG

`class CombinedPRNG` – оголошує клас з назвою `CombinedPRNG`. Клас - це зразок коду, який визначає структуру та поведінку об'єктів.

`constructor()` – оголошує конструктор класу `CombinedPRNG`. Конструктор - це спеціальний метод, який викликається при створенні нового об'єкта класу. Він використовується для ініціалізації полів об'єкта.

`this.mt = new MersenneTwister()` – створюється новий екземпляр класу `MersenneTwister` і присвоює його полю `this.mt` поточного об'єкта `CombinedPRNG`. Це поле використовується для зберігання екземпляра генератора псевдовипадкових чисел `Mersenne Twister`.

`this.lfsr = new LFSR()` – створюється новий екземпляр класу `LFSR` і присвоює його полю `this.lfsr` поточного об'єкта `CombinedPRNG`. Це поле використовується для зберігання екземпляра генератора псевдовипадкових чисел `LFSR`.

`generateRandomNumber()` – оголошує метод з назвою `generateRandomNumber`. Метод – це функція, яка пов'язана з класом і може бути викликана з об'єктів цього класу. Цей метод використовується для генерації нового псевдовипадкового числа.

`const mtRandom = this.mt.random()` – цей рядок декларує локальну змінну `mtRandom` і присвоює їй результат виклику методу `random` екземпляра генератора псевдовипадкових чисел `Mersenne Twister`, який зберігається в полі `this.mt`. Цей метод генерує псевдовипадкове число в діапазоні $[0, 1]$.

`const lfsrRandom = this.lfsr.random()` – декларує локальну змінну `lfsrRandom` і присвоює їй результат виклику методу `random` екземпляра генератора псевдовипадкових чисел LFSR, який зберігається в полі `this.lfsr`. Цей метод генерує псевдовипадкове число в діапазоні `[0, 1]`.

`return mtRandom + lfsrRandom` – повертає суму `mtRandom` і `lfsrRandom`. Ці два псевдовипадкові числа додаються, щоб отримати нове псевдовипадкове число.

2. Клас MersenneTwister

Цей клас представляє генератор псевдовипадкових чисел Mersenne Twister. Він ініціалізує початкове значення (`seed`) на основі поточної часової відмітки.

Метод `random` генерує псевдовипадкове число за допомогою алгоритму Mersenne Twister. Використовується спрощена реалізація, яка ґрунтується на множенні, додаванні та залишковому діленні, що показує рис 4.2:

```
class MersenneTwister {
  /**
   * Конструктор для класу MersenneTwister.
   */
  constructor() {
    this.seed = Date.now(); // Початкове значення на основі поточної часової відмітки
  }

  /**
   * Генерує псевдовипадкове число за допомогою алгоритму Mersenne Twister
   *
   * @returns {number} Псевдовипадкове число.
   */
  random() {
    // Реалізація алгоритму Mersenne Twister
    // (це спрощена версія для демонстрації)
    this.seed = (this.seed * 9301 + 49297) % 233280;
    return this.seed / 233280;
  }
}
```

Рисунок 4.2 – Елемент коду з класом Mersenne Twister

`class MersenneTwister` – оголошує клас з назвою `MersenneTwister`.

`constructor()` – оголошує конструктор класу `MersenneTwister`.

`this.seed = Date.now()` – ініціалізує поле `this.seed` поточного об'єкта `MersenneTwister` поточним часом у мілісекундах. Це значення використовується як початкове значення для генератора псевдовипадкових чисел.

`random()` – оголошує метод з назвою `random`. Цей метод використовується для генерації нового псевдовипадкового числа.

`this.seed = (this.seed * 9301 + 49297) % 233280` – оновлює значення поля `this.seed`. Нове значення розраховується як добуток `this.seed` і 9301, доданий до 49297, а потім взятий залишок від ділення на 233280. Це є частиною алгоритму Mersenne Twister для генерації псевдовипадкових чисел.

`return this.seed / 233280` – повертає значення поля `this.seed` ділене на 233280. Це псевдовипадкове число в діапазоні $[0, 1]$.

3. Клас LFSR

Цей клас представляє генератор псевдовипадкових чисел на основі лінійного регістру зсуву зі зворотним зв'язком (LFSR). Він ініціалізує початковий стан (`state`) LFSR.

Метод `random` генерує псевдовипадкове число за алгоритмом LFSR. Використовується спрощена реалізація, яка ґрунтується на зсуві бітів, XOR та нормалізації. Детально цей процес показано на рис 4.3.

```
class LFSR {
    /**
     * Конструктор для класу LFSR.
     */
    constructor() {
        this.state = 0b10101010; // Початковий стан для LFSR
    }

    /**
     * Генерує псевдовипадкове число за алгоритмом LFSR.
     *
     * @returns {number} Псевдовипадкове число.
     */
    random() {
        // Реалізація алгоритму LFSR
        // (спрощена версія для демонстрації)
        this.state = (this.state >> 1) ^ (-(this.state & 1) & 0xB400); // XOR з поліномом зворотного зв'язку
        return this.state / 255; // Нормалізувати до діапазону [0, 1]
    }
}
```

Рисунок 4.3 – Елемент коду з класом LFSR

`class LFSR` – оголошує клас з назвою LFSR.

`constructor()` – оголошує конструктор класу LFSR.

`this.state = 0b10101010` – ініціалізує поле `this.state` поточного об'єкта LFSR значенням `0b10101010`. Це початковий стан регістру зсуву, який використовується для генерації псевдовипадкових чисел.

`random()` – оголошує метод з назвою `random`.

`this.state = (this.state >> 1) ^ (-(this.state & 1) & 0xB400)` – оновлює значення поля `this.state`. Нове значення розраховується як поточне значення `this.state`, зсунуте

на 1 біт вправо, XOR-ине з доповненням до 1 поточного значення `this.state` та побітовим AND з `0xB400`. Це є частиною алгоритму LFSR для генерації псевдовипадкових чисел.

`return this.state / 255` – повертає значення поля `this.state` ділене на 255. Це псевдовипадкове число в діапазоні `[0, 1]`.

У прикладі використання створюється об'єкт `prng`, зображений на рис. 4.4, класу `CombinedPRNG`. Викликається метод `generateRandomNumber` двічі для отримання двох псевдовипадкових чисел `randomValue1` та `randomValue2`. Значення `randomValue1` та `randomValue2` виводяться в консоль.

```
// Приклад використання комбінованого ГВЧ
const prng = new CombinedPRNG();
const randomValue1 = prng.generateRandomNumber();
console.log(`Random Value 1: ${randomValue1}`);

const randomValue2 = prng.generateRandomNumber();
console.log(`Random Value 2: ${randomValue2}`);
```

Рисунок 4.4 – Елемент коду з прикладом

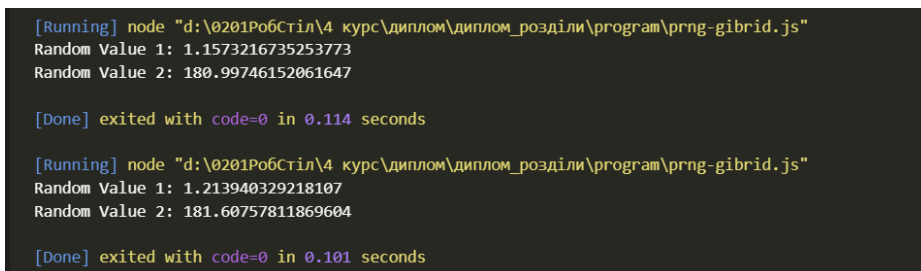
`const prng = new CombinedPRNG()` – створює новий екземпляр класу `CombinedPRNG` і присвоює його змінній `prng`.

`const randomValue1 = prng.generateRandomNumber()` – викликає метод `generateRandomNumber` екземпляра `prng` і присвоює результат змінній `randomValue1`. Це псевдовипадкове число, згенероване комбінованим ГПВЧ.

`console.log(Random Value 1: ${randomValue1})` – виводить текст "Random Value 1:" в консоль, за яким йде значення `randomValue1`. Це дозволяє побачити згенероване псевдовипадкове число.

`const randomValue2 = prng.generateRandomNumber()` – повторно викликає метод `generateRandomNumber` екземпляра `prng` і присвоює результат змінній `randomValue2`. Це ще одне псевдовипадкове число, згенероване комбінованим ГПВЧ.

`console.log(Random Value 2: ${randomValue2})` – виводить текст "Random Value 2:" в консоль, за яким йде значення `randomValue2`. Це дозволяє побачити друге згенероване псевдовипадкове число.



```
[Running] node "d:\0201РобСтіл\4 курс\диплом\диплом_розділи\program\prng-gibrid.js"
Random Value 1: 1.1573216735253773
Random Value 2: 180.99746152061647

[Done] exited with code=0 in 0.114 seconds

[Running] node "d:\0201РобСтіл\4 курс\диплом\диплом_розділи\program\prng-gibrid.js"
Random Value 1: 1.213940329218107
Random Value 2: 181.60757811869604

[Done] exited with code=0 in 0.101 seconds
```

Рисунок 4.5 – Вивід результату

Запустивши код, можна побачити, що він генерує два значення, як і вказано у описі програмної реалізації. При кожному новому запуску відбувається унікальна генерація числа, яку можна побачити на рис 4.5.

Висновки до розділу 4

У цьому розділі дипломної роботи була описана програмна реалізація комбінованого ГПВЧ, який використовує алгоритми Mersenne Twister та LFSR. Цей ГПВЧ може генерувати псевдовипадкові числа з більшою стійкістю та періодом, ніж кожен з алгоритмів окремо.

Комбіновані ГПВЧ можуть мати ряд застосувань у криптографії та кібербезпеці. Наприклад, їх можна використовувати для:

- Генерації криптографічних ключів: Псевдовипадкові числа можуть використовуватися для генерації криптографічних ключів, які є важливими компонентами багатьох криптографічних систем.
- Шифрування та дешифрування даних: Псевдовипадкові числа можуть використовуватися для шифрування та дешифрування даних, роблячи їх стійкими до несанкціонованого доступу.

- Генерації маскувальних кодів: Псевдовипадкові числа можуть використовуватися для генерації маскувальних кодів, які можуть використовуватися для приховування вмісту даних.

- Виявлення атак: Псевдовипадкові числа можуть використовуватися для виявлення атак на криптографічні системи, генеруючи несподівані дані, які злоумисники не зможуть передбачити.

Важливо зазначити, що не всі комбіновані ГПВЧ однаково надійні. Деякі комбінації алгоритмів можуть мати слабкості, які можуть бути використані злоумисниками. Тому важливо ретельно обирати алгоритми та реалізацію ГПВЧ для конкретного застосування.

ВИСНОВКИ

В ході дипломної роботи було проаналізовано основні загрози при впровадженні криптографічного та технічного захисту інформації, а також існуючих методів генерації чисел. В процесі аналізу було обрано два ГПВЧ для подальшого впровадження у інформаційну систему. Для цього було проаналізовано обрані ГПВЧ, та на базі цього визначені сильні сторони для подальшої програмної реалізації гібридного ГПВЧ в органі місцевого самоврядування. А саме, обрано генератор лінійного рекурентного регістру та генератор на основі вихору Мерсенна, після чого за допомогою класу `class CombinedPRNG` вони були об'єднані в генератор-гібрид, що забезпечує більшу стійкість.

На основі порівняльної характеристики, було виокремлено сильні сторони обидвох генераторів, в результаті чого було створено генератор, особливість якого полягає в тому, щоб секретний ключ генерації був доступний користувачеві тільки один раз. Використання двох існуючих об'єднаних генераторів гарантує більшу випадковість, оскільки кожного разу вони генерують різні числа, які не повторюються.

Таким чином, дана дипломна робота має значну наукову та практичну цінність. Результати дослідження можуть бути використані для розробки нових методів та систем криптографії, а також створення КСЗІ.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Hashemi-Pour C. What is the CIA Triad? | Definition from TechTarget [Електронний ресурс] / Cameron Hashemi-Pour, Wesley Chai // WhatIs. – Режим доступу: <https://www.techtarget.com/whatis/definition/Confidentiality-integrity-and-availability-CIA>.
2. Про основні засади розвитку інформаційного суспільства в Україні на 2007 – 2015 роки [Електронний ресурс] : Закон України від 28.05.2014 р. № 4949. – Режим доступу: <https://ips.ligazakon.net/document/JG3TH00A?an=3>.
3. Що таке інформаційна безпека (InfoSec)? | Захисний комплекс Microsoft [Електронний ресурс] – Режим доступу: <https://is.gd/9bMjO4>.
4. Про Положення про технічний захист інформації в Україні [Електронний ресурс] : Указ Президента України від 27.09.1999 р. № 1229/99 : станом на 4 трав. 2008 р. – Режим доступу: <https://zakon.rada.gov.ua/laws/show/1229/99#Text>.
5. Про електронні довірчі послуги [Електронний ресурс] : Закон України від 05.10.2017 р. № 2155-VIII : станом на 1 січ. 2024 р. – Режим доступу: <https://zakon.rada.gov.ua/laws/show/2155-19#Text>.
6. Про схвалення Концепції розвитку системи електронних послуг в Україні [Електронний ресурс] : Розпорядж. Каб. Міністрів України від 16.11.2016 р. № 918-р. – Режим доступу: <https://zakon.rada.gov.ua/laws/show/918-2016-p#Text>.
7. Про захист інформації в інформаційно-телекомунікаційних системах [Електронний ресурс] : Закон України від 05.07.1994 р. № 80/94-ВР : станом на 4 квіт. 2024 р. – Режим доступу: <https://zakon.rada.gov.ua/laws/show/80/94-vp#Text>.
8. Про захист персональних даних [Електронний ресурс] : Закон України від 01.06.2010 р. № 2297-VI : станом на 27 квіт. 2024 р. – Режим доступу: <https://zakon.rada.gov.ua/laws/show/2297-17#Text>.

9. Про основні засади забезпечення кібербезпеки України [Електронний ресурс] : Закон України від 05.10.2017 р. № 2163-VIII : станом на 4 квіт. 2024 р. – Режим доступу: <https://zakon.rada.gov.ua/laws/show/2163-19#Text>.

10. Конвенція про захист осіб у зв'язку з автоматизованою обробкою персональних даних [Електронний ресурс] : Конвенція Ради Європи від 28.01.1981 р. : станом на 6 лип. 2010 р. – Режим доступу: https://zakon.rada.gov.ua/laws/show/994_326#Text.

11. Лекційний матеріал: КRYPTOґРАФІЯ. ШИФРИ, ЇХ ВИДИ І ВЛАСТИВОСТІ [Електронний ресурс]. – Режим доступу: https://view.officeapps.live.com/op/view.aspx?src=https://moodle.znu.edu.ua/pluginfile.php/247176/mod_resource/content/1/%D0%9B_1.doc&wdOrigin=BROWSELINK.

12. Твердун В. ,Голінський І. Сучасні високопродуктивні та розподілені технології криптоаналізу / Голінський І. Твердун В. // ІІІ Всеукраїнська студентська науково - технічна конференція "Природничі та гуманітарні науки. актуальні питання", Тернопіль, 23 квіт. 2010 р. – Тернопіль, 2010. – С. 63.

13. Про Положення про порядок здійснення криптографічного захисту інформації в Україні [Електронний ресурс] : Указ Президента України від 22.05.1998 р. № 505/98 : станом на 12 верес. 2009 р. – Режим доступу: <https://zakon.rada.gov.ua/laws/show/505/98#Text>.

14. Що таке соціальна інженерія [Електронний ресурс] // <https://nadiyno.org>. – Режим доступу: <https://nadiyno.org/shho-take-soczialna-inzheneriya/>.

15. Повнодискове шифрування [Електронний ресурс] // Як?. – Режим доступу: <https://yak.dslua.org/services/windows/povnodyskove-shyfruvannia/>.

16. КRYPTOґРАФІЧНА СИСТЕМА [Електронний ресурс] // Wiki ТНТУ. – Режим доступу: https://wiki.tntu.edu.ua/КRYPTOґРАФІЧНА_СИСТЕМА.

17. Комплексні системи захисту інформації в інформаційно-телекомунікаційних системах : Навч. посіб. / В. Козюра [та ін.]. – 2-е вид. – Ніжин : ФОП Лук'яненко В.В. ТПК «Орхідея», 2019. – 144 с.

18. Нечипорук В. В. Комплексні системи захисту інформаційних систем спеціального призначення / Віталій Володимирович Нечипорук, Світлана Олександрівна Кашкевич, Андрій Володимирович Шишацький // Technical sciences science and development of methods for solving modern problems : Міжнар. наук. конф., Melbourne, 21 черв. 2023 р. – Melbourne, 2023. – с. 215.

19. Савченко А. Інформаційне забезпечення діяльності органів державної фіскальної служби України / А. Савченко // Наше право. – 2015. – № 6. – с. 191.

20. Загальний огляд систем віброакустичного зашумлення [Електронний ресурс] // ТЗІ - інформаційна безпека та захист інформації. – Режим доступу: <https://tzi.com.ua/zagalnij-oglyad-sistem-vbroakustichnogo-zashumlennya.html>.

21. Олефіренко Т., Матвієнко О., Васютіна. Т. Основи організації дистанційного та змішаного навчання у закладах вищої освіти : Навч.-метод. посіб. для студентів / Т. Олефіренко, О. Матвієнко, Т. Васютіна. – 7-ме вид. – Київ : НПУ ім. М.П. Драгоманова, 2022. – 146 с.

22. Інтерференція [Електронний ресурс] // УМІТИ. – Режим доступу: <https://umity.in.ua/concept/?id=876>.

23. Каланча А. Апаратне забезпечення для пригнічення зв'язку ворога у період війни / Андрій Каланча // Кібербезпека в Україні: правові та організаційні питання : Міжнар. науково-практ. конф., Одеса, 23 листоп. 2023 р. – Одеса, 2022. – С. 66–68.

24. LibreTexts. 5.7: модульна арифметика [Електронний ресурс] / LibreTexts // LibreTexts - Ukrayinska. – Режим доступу: <http://surl.li/ugimc>.

25. Теорія алгоритмів і математичні основи представлення знань [Електронний ресурс] // StudFiles. – Режим доступу: <https://studfile.net/preview/5009878/>.

26. Алгоритми та їх властивості – Цікавинки для допитливих, навчальні матеріали [Електронний ресурс] // Математика, логіка, інтелект. – Режим доступу: <https://formula.co.ua/blog/alhorytmy-ta-jih-vlastyvosti/>.

27. Числа фібоначчі: відкриття, властивості та застосування - Валерій У. [Електронний ресурс] // Букі | Buki. – Режим доступу: <https://buki.com.ua/blogs/cisla-fibonacci-vidkrittia-vlastivosti-ta-zastosuvannia/>.
28. Ознайомлення із золотим перетином. [Електронний ресурс] // Adobe. – Режим доступу: <https://www.adobe.com/ua/creativecloud/design/discover/golden-ratio.html>.
29. Білий шум [Електронний ресурс] // ВУЕ. – Режим доступу: https://vue.gov.ua/Білий_шум.
30. Gaussian white noise excitation [Електронний ресурс] // researchgate. – Режим доступу: <https://www.researchgate.net/publication/318715568/figure/fig1/AS:941807122804737@1601555724126/Gaussian-white-noise-excitation.gif>.
31. Szabo A. White noise vs. pink noise vs. the sound masking spectrum (educational focus article) [Електронний ресурс] / Attila Szabo // LinkedIn: Log In or Sign Up. – Режим доступу: <https://www.linkedin.com/pulse/white-noise-vs-pink-sound-masking-spectrum-focus-article-attila-szabo/>.
32. Ippolito R. Linear feedback shift register [Електронний ресурс] / Raffaello Ippolito // Medium. – Режим доступу: <https://medium.com/@raffaelloippolito/linear-feedback-shift-register-1062951786a6>.
33. Pseudo random number generation using linear feedback shift registers | analog devices [Electronic resource] // Mixed-signal and digital signal processing ICs | Analog Devices. – Mode of access: [https://www.analog.com/en/resources/design-notes/random-number-generation-using-lfsr.html#:~:text=LFSRs%20\(linear%20feedback%20shift%20registers,shows%20a%205-bit%20LFSR](https://www.analog.com/en/resources/design-notes/random-number-generation-using-lfsr.html#:~:text=LFSRs%20(linear%20feedback%20shift%20registers,shows%20a%205-bit%20LFSR).
34. Zapfel C. An introduction to lfsrs for cryptography [Електронний ресурс] / Connor Zapfel // Medium. – Режим доступу: <https://medium.com/@czapfel/an-introduction-to-lfsrs-for-cryptography-bf2602640e91>.
35. ESTREAM - the ECRYPT stream cipher project [Електронний ресурс] // Wayback Machine. – Режим доступу:

<https://web.archive.org/web/20120701135329/http://www.ecrypt.eu.org/stream/cryptmtfubuki.html>.

36. Variants of mersenne twister suitable for graphic processors [Электронный ресурс] // arXiv.org. – Режим доступа: <https://arxiv.org/abs/1005.4973v3>.

37. SIMD-oriented fast mersenne twister (SFMT) [Электронный ресурс] // Dept. Math., Hiroshima Univ. – Режим доступа: <http://www.math.sci.hiroshima-u.ac.jp/m-mat/MT/SFMT/index.html>.

38. Tiny mersenne twister (tinymt) [Электронный ресурс] // Dept. Math., Hiroshima Univ. – Режим доступа: <http://www.math.sci.hiroshima-u.ac.jp/m-mat/MT/TINYMT/index.html>.

39. IsaacWiper and HermeticWizard: New wiper and worm targeting Ukraine [Электронный ресурс] // Award-winning news, views, and insight from the ESET security community. – Режим доступа: <https://www.welivesecurity.com/2022/03/01/isaacwiper-hermeticwizard-wiper-worm-targeting-ukraine/>.

ДОДАТОК А

Лістинг програмної реалізації гібридного ГПВЧ

```
/**
 * Клас, що представляє комбінований генератор псевдовипадкових чисел з використанням
 Mersenne Twister та LFSR.
 */
class CombinedPRNG {
    /**
     * Конструктор класу CombinedPRNG.
     */
    constructor() {
        this.mt = new MersenneTwister(); // Ініціалізація ГПВЧ вихору Мерсенна
        this.lfsr = new LFSR(); // Ініціалізація ГЛРР
    }

    /**
     * Генерує нове псевдовипадкове число, використовуючи комбінацію Mersenne
 Twister та LFSR.
     *
     * @returns {number} Нове псевдовипадкове число.
     */
    generateRandomNumber() {
        const mtRandom = this.mt.random(); // Генерація випадкового числа за
 допомогу Mersenne Twister
        const lfsrRandom = this.lfsr.random(); // Генерація випадкового числа за
 допомогу LFSR
        return mtRandom + lfsrRandom; // Об'єднання двох чисел
    }
}

/**
 * Клас, що представляє генератор псевдовипадкових чисел Mersenne Twister.
 */
class MersenneTwister {
    /**
     * Конструктор для класу MersenneTwister.
     */
    constructor() {
        this.seed = Date.now(); // Початкове значення на основі поточної часової
 відмітки
    }

    /**
     * Генерує псевдовипадкове число за допомогою алгоритму Mersenne Twister
     *
     * @returns {number} Псевдовипадкове число.
     */
}
```

```

    random() {
        // Реалізація алгоритму Mersenne Twister
        // (Це спрощена версія для демонстрації)
        this.seed = (this.seed * 9301 + 49297) % 233280;
        return this.seed / 233280;
    }
}

/**
 * Клас, що представляє генератор псевдовипадкових чисел на основі лінійного
регістру зсуву зі зворотним зв'язком (LFSR).
 */
class LFSR {
    /**
     * Конструктор для класу LFSR.
     */
    constructor() {
        this.state = 0b10101010; // Початковий стан для LFSR
    }

    /**
     * Генерує псевдовипадкове число за алгоритмом LFSR.
     *
     * @returns {number} Псевдовипадкове число.
     */
    random() {
        // Реалізація алгоритму LFSR
        // (Спрощена версія для демонстрації)
        this.state = (this.state >> 1) ^ (-(this.state & 1) & 0xB400); // XOR з
поліномом зворотного зв'язку
        return this.state / 255; // Нормалізувати до діапазону [0, 1]
    }
}

// Приклад використання комбінованого ГВЧ
const prng = new CombinedPRNG();
const randomValue1 = prng.generateRandomNumber();
console.log(`Random Value 1: ${randomValue1}`);

const randomValue2 = prng.generateRandomNumber();
console.log(`Random Value 2: ${randomValue2}`);

```

Короткий звіт подібності



Ім'я користувача: Комп'ютерної математики та інформаційної безпеки...	ID перевірки: 1016354237
Дата перевірки: 12.06.2024 22:41:01 EEST	Тип перевірки: Doc vs Internet + Library
Дата звіту: 12.06.2024 23:05:49 EEST	ID користувача: 100005746

Назва документа: Мацкова_КБ_бакалавр.робота

Кількість сторінок: 75 Кількість слів: 15744 Кількість символів: 130305 Розмір файлу: 1.45 MB ID файлу: 1016158167

7.27%
Схожість

Найбільша схожість: 1.98% з Інтернет-джерелом (<http://ir.stu.cn.ua/bitstream/handle/123456789/19247/%d0%9a%d0%b..>)

6.36% Джерела з Інтернету 340 Сторінка 77

5.32% Джерела з Бібліотеки 395 Сторінка 81

0% Цитат

Вилучення цитат вимкнене

Вилучення списку бібліографічних посилань вимкнене

0%
Вилучень

Немає вилучених джерел