УДК 004.422.5

Вовк В. С., здобувачка 2 курсу
спеціальності 122 Комп'ютерні науки,
Павлюк О. А., канд. техн. наук,
старший викладач кафедри інформаційних технологій

АЛГОРИТМ ШИФРУВАННЯ ІНФОРМАЦІЇ З ВИКОРИСТАННЯМ ПСЕВДОВИПАДКОВИХ ПОСЛІДОВНОСТЕЙ

Донецький національний університет імені Василя Стуса, м. Вінниця

Алгоритм шифрування – це набір інструкцій, що перетворює дані у формат, який неможливо прочитати без спеціального ключа (зашифрований текст), використовуючи ключ шифрування для захисту інформації.

Генератор псевдовипадкових послідовностей у класичному вигляді являє собою деякий повністю математичний алгоритм, що на виході видає числа або двійкові послідовності, які схожі на дійсно випадкові числа та послідовності. Основними елементами у таких генераторах є пам'ять розміром n, яка зберігає стан генератора, і деяке перетворення, за допомогою якого обраховується наступний стан генератора та послідовність, яка подається на вихід. Такий алгоритм завжди може бути представлений як еквівалентний автомат [1].

Генератори псевдовипадкових послідовностей з хорошими статистичними властивостями застосовуються для вирішення багатьох прикладних завдань, як-от генерація криптографічних ключів, реалізація протоколів аутентифікації, створення імітаційних моделей тощо. Для формування якісних послідовностей потрібні хороші статистичні властивості, непередбачуваність, відтворюваність.

У загальному випадку схема генерування ПВП описується виразом [2]:

$$x_{n+1} = (a \cdot x_n + d) \mathrm{mod} N,$$

де x_n, x_{n+1} – значення системи на n-ій та $n+1$-ій ітерації;

$x_0, a, d \in \{0,1, \dots, \mathrm{N}-1\}$ – параметри системи;

N – натуральне число, а «mod» означає арифметичний оператор знаходження залишку від результату ділення цілих чисел.

Алгоритм шифрування базується на використанні в якості ключа генератора ПВП. Схема генерування ключа використовує два незалежні генератори ПВП (лінійний конгурентний генератор та генератор на основі логістичного відображення), що працюють із різними початковими умовами:

$$x_{n+1} = (a \cdot x_n + d) \mathrm{mod} N, x_{n+1} = \lambda \cdot x_n (1 - x_n),$$

де a_1, d_1, λ, x_0 – початкові умови для генерування послідовностей.

Питання безпеки передачі даних з давніх-давен стояло перед суспільством. Злам, перехоплення та зміна даних наносять значну шкоду для інформаційної безпеки та людської життєдіяльності. Ці загрози можуть стосуватись як окремо людини, так і суспільства загалом і призводити до високих економічних втрат та матеріальних збитків.

Для перешкоджання виникненню таких загроз застосовують алгоритми шифрування даних. Одним із таких алгоритмів є алгоритм шифрування RSA. Це криптографічний алгоритм з відкритим ключем, який базується на складності задачі розкладу на прості множники великих цілих чисел [3].

Список використаних джерел

1. Лисенко Д. В. Метод підвищення ефективності генерації псевдовипадкових послідовностей для потокового шифрування даних: магістерська дисертація. Київ, 2022.

2. Knuth D. E. The art of computer programming. Vol. 2, Seminumerical algorithms. 3rd ed. Boston: Addison-Wesley, 1997. 762 p. URL: https://www-cs-faculty.stanford.edu/~knuth/taocp.html

3. Актуальні питання забезпечення цивільного захисту: збірник тез доповідей VIII Всеукраїнської науково-практичної конференції здобувачів вищої освіти і молодих учених (м. Львів, 2021 р.) / Львівський державний університет безпеки життєдіяльності. Львів: ЛДУ БЖД, 2021.

УДК 004.422.5

Бондар А. Б., здобувачка 2 курсу
спеціальності 122 Комп'ютерні науки,
Потапова Н. А., канд. екон. наук,
доцент, доцент кафедри інформаційних технологій

ВИКОРИСТАННЯ АЛГОРИТМІВ ПОШУКУ У СТВОРЕННІ СТРАТЕГІЧНИХ ІГОР РЕАЛЬНОГО ЧАСУ

Донецький національний університет імені Василя Стуса, м. Вінниця

Стратегічні ігри реального часу (RTS) є одним із найбільш вимогливих жанрів щодо алгоритмічних рішень, оскільки у них необхідно забезпечити одночасну взаємодію великої кількості автономних юнітів. Згідно з матеріалами Game Developer, ефективність RTS напряму залежить від здатності ігрового рушія швидко виконувати пошук маршруту, обробляти запити юнітів та відповідати на динамічні зміни середовища [1]. Саме тому алгоритми пошуку стали базовим інструментом для розробників цього жанру.

Однією з ключових задач у RTS є маршрутизація юнітів, яку найчастіше реалізують за допомогою алгоритму A*. Проте класичний A* зазнає значних труднощів на великих і насичених картах. Матеріали MovingAI показують, що для одночасного переміщення сотень юнітів розробники переходять до кооперативних методів, як-от Cooperative Pathfinding та Windowed HCA*, що дають змогу уникати конфліктів між агентами та покращують узгодженість групового руху [2].

Іншим важливим напрямом є Flow Field Pathfinding, описаний у доповіді GDC Vault. Цей підхід забезпечує не окремий маршрут для кожного юніта, а глобальне поле напрямів, яке зчитують усі агенти. Таке рішення було успішно