ISSN 2707-1898 (print)

Український журнал інформаційних технологій
Ukrainian Journal of Information Technology
http://science.lpnu.ua/uk/ujit
https://doi.org/10.23939/ujit2023.01.067
Article received 20.03.2023 p.
Article accepted 02.05.2023 p.
UDK 004.056

Correspondence author
O. V. Isakov
oleksandr.v.isakov@lpnu.ua

О. В. Ісаков, С. С. Войтусік
Національний університет "Львівська політехніка", м. Львів, Україна

ПОРІВНЯЛЬНИЙ АНАЛІЗ ЦИФРОВИХ ШУМІВ, ЗГЕНЕРОВАНИХ АДИТИВНИМИ ГЕНЕРАТОРАМИ ФІБОНАЧЧІ

Генератори шумів та генератори псевдовипадкових чисел (ГПВЧ) широко використовуються у сфері інформаційних технологій, зокрема кібербезпеці, для моделювання, генерування ключів авторизації та технічного захисту інформації. З'ясовано, що характеристики цифрового шуму напряму залежать від обраного алгоритму ГПВЧ. Для визначення якості згенерованого шуму проводять спеціальні тести, які передусім застосовуються до згенерованої за допомогою ГПВЧ послідовності. Досліджено результати цифрових шумів, згенерованих за допомогою ГПВЧ на підставі чотирьох різних алгоритмів адитивних генераторів Фібоначі (АГФ). Вибір генераторів одного типу дав змогу проаналізувати вплив різних модифікацій на остаточний результат згенерованих послідовностей, щоб визначити їхні переваги та недоліки. Для тестування шуму і згенерованих послідовностей використано техніки цифрового оброблення сигналів, такі як: частотний, автокореляційний та візуальний аналіз, співвідношення сигнал/шум і статистичні тести пакету NIST. Розроблено функції для інтерпретації отриманих даних за допомогою пакету прикладних програм MATLAB (DSP System Toolbox) та мови програмування C для автоматизації тестів NIST. З'ясовано, що для ефективного тестування варто визначити конкретні етапи і їх послідовність: визначення періоду ГПВЧ, статистичні тести пакету NIST, обчислення автокореляційної функції, інші методи цифрового оброблення сигналів. Встановлено, що модифікація одного АГФ за допомогою використання біту переносу (МАГФ2) не покращує результати згенерованої послідовності, на відміну від алгоритму PIKE, який складається із трьох АГФ. Алгоритм МАГФ показав кращі результати при тестуванні періоду і водночас пройшов тести NIST, на відміну від немодифікованої версії. Виявлено залежність між порядком згенерованих послідовностей та результатами їхньої автокореляційної функції. Запропоновано, окрім загальних статистичних тестів, проводити прикладні. При виборі чи під час розроблення нового генератора, варто перевірити його ефективність в умовах, які вимагаються згідно з наявними стандартами та вимогами. Встановлено відповідність згенерованих цифрових шумів із вимогами до пристроїв технічного захисту інформації, а саме – захисту мовної інформації.

Ключові слова: генератор псевдовипадкових чисел, генератор шуму, NIST(National Institute of Standards and Technology), цифрова обробка сигналів.

Вступ / Introduction

ГПВЧ відіграють велику роль у сучасному технологічному світі і застосовуються в багатьох сферах, фізичних дослідженнях, моделюванні, цифровій обробці сигналів, забезпеченні інформаційної безпеки, де часто приходиться генерувати або ж стикатись із випадковими послідовностями. З кожним роком ГПВЧ стають все більш актуальними для використання у сфері інформаційної безпеки [1]. До того ж, саме через потребу в послідовностях, які не тільки проходять статистичні тести, й такі, результати яких не можуть бути передбачені, без знання внутрішнього стану генератора і його початкової конфігурації, виникла окрема група криптостійких ГПВЧ.

У сфері кібербезпеки ГПВЧ використовують як на апаратному, так і на програмному рівні. Так, генератори часто є складовою протоколів автентифікації та генерування для ключів шифрування. Окрім цього, ГПВЧ може використовуватися в якості джерела ентропії для генератора шуму, який водночас часто застосовується для захисту каналів зв'язку. Оскільки використання ГПВЧ у сфері інформаційної безпеки є більш вибагливим і вимагає додаткових тестів та умов використання, дослідження таких генераторів дасть змогу оптимізувати їх роботу та удосконалити наявні підходи до їх побудови.

Для того, щоб отримати більше інформації про алгоритм, на підставі якого планується будувати ГПВЧ, доцільно протестувати його на практиці. Наявні пакети тестування, такі як: NIST, Diehard, TestU01, які дають змогу визначити статистичні характеристики згенерованої послідовності, допомагають одразу знайти недоліки в генераторі, але в такому випадку не враховується сфера та умови застосування генератора. Залежно від потреб, критичними можуть бути вимоги до швидкодії,

пам'яті, періоду згенерованої послідовності та її достатньої випадковості [2]. В даній роботі ГПВЧ використовується як основа для цифрового генератора шуму. Саме тому, окрім результатів статистичних тестів пакету NIST [3], проведено експеременти в контексті цифрового оброблення сигналів.

Об'єкт дослідження – процес порівняльного аналізу основних обчислювальних характеристик цифрових шумів.

Предмет дослідження – методи, алгоритми та програмна реалізація ГПВЧ.

Мета роботи – порівняльний аналіз досліджених обчислювальних характеристик цифрових шумів. Це дасть змогу оптимального вибору алгоритму ГПВЧ для використання в якості генератора цифрового шуму.

Для досягнення зазначеної мети визначено такі основні *завдання дослідження*:

- аналіз та дослідження методів тестування ГПВЧ;
- визначення залежності між результатами статистичних тестів та цифрового оброблення шуму;
- дослідження впливу на результат згенерованої послідовності різних методів модифікації ГПВЧ;
- визначення відповідності згенерованих шумів із вимогами до захисту мовної інформації.

Матеріали і методи дослідження. Усі ГПВЧ імплементовано на базі адитивного алгоритму Фібоначі. Обрано саме цей тип генераторів через простоту їх реалізації та відому криптостійку версію алгоритму. Отже, максимальна подібність цих алгоритмів дає змогу дослідити наявність чи відсутність зв'язку між якістю згенерованої послідовності та версією алгоритму. Також важливим питанням є доцільність використання криптостійкої версії алгоритму в певних задачах. Хоча результати криптостійких алгоритмів неможливо передбачити без знання внутрішнього стану системи, такі алгоритми повільніші, вимагають більше пам'яті і їх складніше реалізовувати на практиці.

Аналіз останніх досліджень та публікацій. Останні дослідження показують інтерес до пошуку ефективних ГПВЧ, їх характеристик та методів модифікацій наявних алгоритмів [2; 4, 5]. Ці роботи демонструють тестування наявних алгоритмів та дослідження впливу початкових значень та різних характеристик генератора на остаточний результат. Так, у роботі Мохамеда та Аванга [4] описані результати шести популярних ГПВЧ, протестовано на критерій узгодженості Пірсона та їх швидкодію. В дослідженнях Камаліки та Суканти [5] глибше досліджуються результати емпіричних і графічних тестів, зокрема проблема вибору полінома для роботи алгоритму з максимально можливим періодом.

В своїй роботі Рос Андерсон [6] запропонував заміну криптостійкому генератору Fish, який був ним зламаний. Додавши ще один АГФ і удосконаливши умову генерування наступного значення, автор підсумував щодо позитивного впливу модифікації за рахунок біту переносу і представив генератор PIKE. Більше способів та підходів до модифікації наявних алгоритмів ГПВЧ описані в фундаментальній праці Фергюсона та Шнайера [2].

Значну увагу приділяють пошукам алгоритмів, які одночасно задовільняли б такі вимоги: рівномірний розподіл значень, великий період, нерозривні послідовності, швидкодія. Для генерування різного типу ключів автентифікації, додатковою вимогою буде криптостійкість алгоритму.

У роботі Роберта Мінгеза [7] досліджено вплив характеристик ГПВЧ на генератор цифрового шуму і підхід до вибору алгоритму, базуючись на вимозі достатнього періоду і високої швидкодії.

Проаналізувавши наявні дослідження, можна зробити висновок, що на сьогодні немає єдиного алгоритму, який би повною мірою задовільнив усі вимоги, адже в усіх випадках доведеться шукати компроміс між періодом, статистичними характеристиками та швидкодією алгоритму. Через відсутність формули, яка б дала можливість моментально знаходити оптимальні початкові значення та константи ГПВЧ, необхідно проводити більше порівняльних аналізів та досліджень для пошуку оптимальних алгоритмів.

Результати дослідження та їх обговорення/ Research results and their discussion

Адитивний генератор Фібоначі (АГФ) [2] був запропонований в якості заміни та покращеної версії генераторів на підставі лінійного конгруентного методу і має загальну формулу:

$$X_{n+1} = (X_n + X_{n-1}) \bmod 2^m . \qquad (1)$$

Структурна схема АГФ (рис. 1) складається з трьох регістрів X_{n-1}, X_n, X_{n+1}, суматора ADD і мультиплексора MUL. у даному випадку мультиплексор використовується для обчислення операції $\bmod$, як:

$$x \bmod (2^m) = x \,\&\, (m-1) .$$

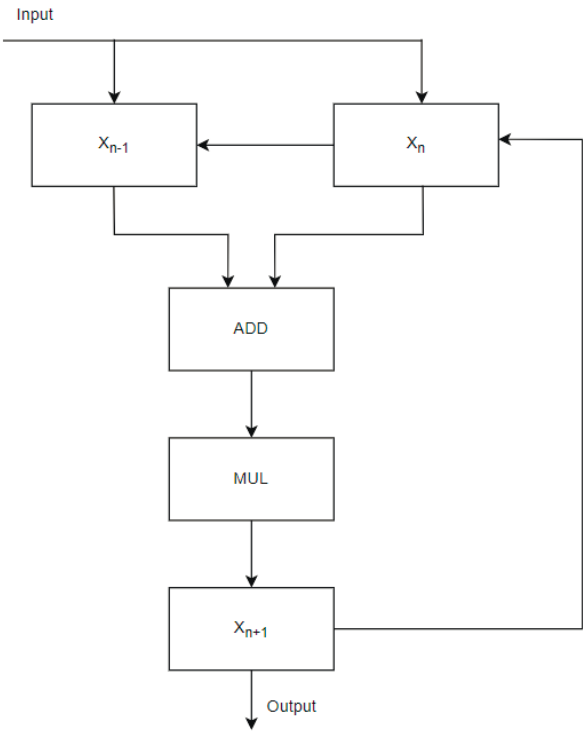

Рис. 1. Структурна схема АГФ / Block diagram of the AFG

Число m дає змогу коригувати розрядність згенерованої послідовності, n – індекс. Результати такого генератора залежать від його початкової конфігурації, тому для покращення статистичних характеристик можна використовувати не два попередні значення, які йдуть одне за одним, а з додатковою затримкою a, b. У такому випадку потрібно передбачити потребу більшого використання пам'яті, замість двох значень у загальному варіанті до:

$n = \max(a,b)$, якщо індекс починається з 0;

$n = \max(a,b)+1$, якщо індекс починається з 1.

Загалом, для цього алгоритму характерними є висока швидкодія і невелике використання пам'яті. Однак для отримання послідовності, яка може вважатись достатньо випадковою, необхідно використовувати модифіковані версії цього алгоритму з рекомендованими коефіцієнтами затримки, хорошою хеш функцією або генератором випадкових чисел для ініціалізації початкових значень.

Опис використаних алгоритмів. АГФ – описаний у формулі (1).

Модифікований генератор АГФ (МАГФ) [8] =

$$= X_{n+1} = (X_n + X_{n-1} + a) \bmod 2^m, \quad (2)$$

де m – *розрядність;* $a = xor\left(dec2bin\left(X_n\right)\right)$.

Модифікований АГФ (МАГФ2) =

$$= X_{n+1} = (X_n + X_{n-1} + a) \bmod 2^m, \quad (3)$$

де m – *розрядність;* $a = X_{i-2} + X_{i-1} > 2^n \,?1:0$.

Генератор PIKE:

$$\begin{aligned} a1_i &= a1_{i-55} + a1_{i-24} (\bmod 2^m) \\ a2_i &= a2_{i-57} + a2_{i-7} (\bmod 2^m) \\ a3_i &= a3_{i-58} + a2_{i-19} (\bmod 2^m) \\ res_i &= a1_{end} \,\hat{}\, a2_{end} \,\hat{}\, a3_{end}, (\hat{} - xor) \end{aligned} \quad (4)$$

АГФ – звичайний генератор, який не проходить статистичні тести з пакету NIST.

МАГФ – модифікована версія АГФ, у якій додатковим кроком є додавання суми по модулю усіх бітів попереднього згенерованого числа (X_n). Проходить тести NIST.

МАГФ2 – модифікована версія АГФ, у якій додатковим кроком є додавання біту переносу. Проходить тести NIST.

PIKE – криптостійкий генератор псевдовипадкових чисел, який складається з трьох адитивних генераторів Фібоначі. Проходить тести NIST. Залежно від значення біта переносу кожного з генераторів, наступне значення обчислюється, згідно з формули [6; 9]:

$$move = (carryBit1 \,\&\, carryBit2) \,|\, (carryBit2 \,\&\, carryBit3) \,|\, (carryBit1 \,\&\, carryBit3), \quad (5)$$

де & – логічне І; | – логічне АБО.

Після імплементації вказаних алгоритмів проведено:

1) тестування результатів генераторів псевдовипадкових чисел за допомогою набору тестів NIST;

2) створення середовища для тестування цифрових сигналів;

3) порівняння результатів автокореляційної функції з різним набором:

а) вхідних коефіцієнтів;

б) розрядів згенерованих чисел (m).

За допомогою тестового пакету NIST опрацьовано результати імплементованих алгоритмів. Для кожного тесту використовувались однакові конфігурації [3], а саме:

- розмір блоку: 1000000 біт;
- кількість блоків: 100.

Інші параметри були використані за замовчуванням. Внаслідок цього (рис. 2–5) алгоритм АГФ не пройшов тест, решта алгоритмів пройшли тест. МАГФ2 має дещо кращі результати, порівняно з МАГФ, при цьому швидкодія і використання пам'яті для цих алгоритмів є однаковими.

Згенеровані числа були подані у вигляді шуму (рис. 6) та їхніх автокореляційних характеристик (рис. 7).

```
-------------------------------------------------------------------------------
 C1  C2  C3  C4  C5  C6  C7  C8  C9 C10  P-VALUE  PROPORTION  STATISTICAL TEST
-------------------------------------------------------------------------------
100   0   0   0   0   0   0   0   0   0  0.000000 *    0/100  *  Frequency
 21   5   7   5   4   3   9  10   9  27  0.000000 *   91/100  *  BlockFrequency
100   0   0   0   0   0   0   0   0   0  0.000000 *    0/100  *  CumulativeSums
100   0   0   0   0   0   0   0   0   0  0.000000 *    0/100  *  CumulativeSums
100   0   0   0   0   0   0   0   0   0  0.000000 *    0/100  *  Runs
 87   6   0   5   1   0   1   0   0   0  0.000000 *   25/100  *  LongestRun
  8  14  10   6  11   5  18   6  12  10  0.102526    100/100     Rank
100   0   0   0   0   0   0   0   0   0  0.000000 *    0/100  *  FFT
 57   8  11  11   2   3   3   2   2   1  0.000000 *   81/100  *  OverlappingTemplate
  0   0 100   0   0   0   0   0   0   0  0.000000 *  100/100     Universal
100   0   0   0   0   0   0   0   0   0  0.000000 *    0/100  *  ApproximateEntropy
 42  16  14   4   6   3   2   5   4   4  0.000000 *   85/100  *  Serial
  7  10   9  11  10   8  10  11  14  10  0.955835    100/100     Serial
 11   9  10  13   7  10  12   9  12   7  0.924076     98/100     LinearComplexity
```

Рис. 2. Результати АГФ / AFG results

C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	P-VALUE	PROPORTION	STATISTICAL TEST
11	5	11	6	9	11	14	12	10	11	0.678686	98/100	Frequency
9	9	7	11	10	8	15	8	9	14	0.719747	98/100	BlockFrequency
12	7	9	11	5	8	7	13	13	15	0.383827	98/100	CumulativeSums
10	6	8	11	9	12	8	11	13	12	0.883171	98/100	CumulativeSums
11	11	13	13	11	10	9	11	3	8	0.574903	97/100	Runs
4	10	16	8	10	12	14	10	7	9	0.304126	99/100	LongestRun
12	14	13	6	11	9	13	6	7	9	0.514124	97/100	Rank
9	14	10	6	9	7	10	10	15	10	0.657933	98/100	FFT
13	11	10	10	9	8	9	7	14	9	0.897763	99/100	OverlappingTemplate
100	0	0	0	0	0	0	0	0	0	0.000000 *	100/100	Universal
13	12	9	9	7	7	11	15	7	10	0.657933	97/100	ApproximateEntropy
1	1	1	1	2	2	1	0	2	2	0.834308	13/13	RandomExcursions
0	1	2	1	1	1	1	2	2	2	0.834308	13/13	RandomExcursions
2	3	2	0	2	2	1	0	0	1	0.275709	13/13	RandomExcursions
3	0	1	6	0	2	0	1	0	0	0.000060 *	13/13	RandomExcursions
1	0	4	0	4	1	1	0	0	2	0.006196	13/13	RandomExcursions
2	2	1	2	2	0	1	0	2	1	0.637119	12/13	RandomExcursions
1	2	1	2	2	1	0	0	1	3	0.437274	13/13	RandomExcursions
2	1	2	1	2	1	0	1	2	1	0.834308	12/13	RandomExcursions
1	3	0	0	1	1	4	0	1	2	0.048716	13/13	RandomExcursionsVariant
2	2	0	1	1	0	1	1	2	3	0.437274	13/13	RandomExcursionsVariant
1	3	1	0	0	1	0	4	1	2	0.048716	13/13	RandomExcursionsVariant
1	2	2	0	0	2	1	1	3	1	0.437274	13/13	RandomExcursionsVariant
1	1	1	2	0	5	0	1	0	2	0.012650	13/13	RandomExcursionsVariant
1	0	2	0	4	1	2	1	2	0	0.090936	13/13	RandomExcursionsVariant
0	1	2	1	4	1	0	0	3	1	0.048716	13/13	RandomExcursionsVariant
0	2	4	2	2	0	0	1	1	1	0.090936	13/13	RandomExcursionsVariant
1	2	2	3	1	0	0	1	1	2	0.437274	13/13	RandomExcursionsVariant
2	2	1	1	2	1	2	1	1	0	0.834308	12/13	RandomExcursionsVariant
2	1	1	3	0	0	1	2	0	3	0.162606	12/13	RandomExcursionsVariant
2	0	1	2	2	2	1	0	2	1	0.637119	12/13	RandomExcursionsVariant
2	1	1	1	1	2	1	1	2	1	0.964295	13/13	RandomExcursionsVariant
2	2	1	1	1	2	0	0	2	2	0.637119	13/13	RandomExcursionsVariant
2	2	0	3	2	0	1	0	2	1	0.275709	13/13	RandomExcursionsVariant
2	1	2	2	1	1	3	0	0	1	0.437274	13/13	RandomExcursionsVariant
1	3	2	2	0	1	1	1	1	1	0.637119	13/13	RandomExcursionsVariant
2	3	1	0	0	3	3	0	0	1	0.048716	13/13	RandomExcursionsVariant
8	10	7	7	9	12	10	19	10	8	0.262249	97/100	Serial
7	6	11	10	10	10	18	11	6	11	0.289667	99/100	Serial
11	2	12	9	12	10	12	7	16	9	0.191687	99/100	LinearComplexity

Рис. 3. Результати МАГФ / MAFG results

C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	P-VALUE	PROPORTION	STATISTICAL TEST
11	5	18	6	7	10	11	8	16	8	0.066882	99/100	Frequency
8	8	8	6	14	12	8	9	15	12	0.514124	100/100	BlockFrequency
15	6	12	9	9	8	12	10	6	13	0.534146	100/100	CumulativeSums
12	5	10	14	7	14	6	8	13	11	0.350485	99/100	CumulativeSums
5	13	19	7	9	10	10	8	8	11	0.145326	99/100	Runs
15	6	17	17	3	11	9	5	14	3	0.000954	100/100	LongestRun
10	14	5	14	13	9	14	8	6	7	0.262249	97/100	Rank
10	8	9	10	9	10	9	7	12	16	0.779188	100/100	FFT
10	9	8	12	14	9	12	10	10	6	0.867692	99/100	OverlappingTemplate
0	0	0	0	0	0	0	0	0	100	0.000000 *	100/100	Universal
10	13	12	11	12	8	10	9	5	10	0.851383	100/100	ApproximateEntropy
1	4	0	0	0	1	2	0	0	0	----	8/8	RandomExcursions
1	1	1	1	0	0	1	1	1	1	----	8/8	RandomExcursions
2	1	1	0	0	0	1	1	2	0	----	8/8	RandomExcursions
1	0	3	0	0	2	0	0	2	0	----	8/8	RandomExcursions
0	0	0	0	1	1	1	2	1	2	----	8/8	RandomExcursions
1	0	1	0	2	2	0	0	0	2	----	8/8	RandomExcursions
1	0	0	0	2	0	1	1	2	1	----	7/8	RandomExcursions
1	1	0	1	1	1	0	2	1	0	----	8/8	RandomExcursions
1	0	0	1	0	2	0	0	2	2	----	8/8	RandomExcursionsVariant
1	0	0	1	1	0	1	1	2	1	----	8/8	RandomExcursionsVariant
1	0	1	0	0	0	3	1	1	1	----	8/8	RandomExcursionsVariant
1	1	0	1	0	0	0	1	1	3	----	8/8	RandomExcursionsVariant
1	1	1	0	0	0	0	0	3	2	----	8/8	RandomExcursionsVariant
2	0	1	0	0	0	0	0	2	3	----	8/8	RandomExcursionsVariant
1	2	0	0	0	1	1	2	0	1	----	8/8	RandomExcursionsVariant
1	1	1	1	0	1	1	1	1	0	----	8/8	RandomExcursionsVariant
0	2	1	1	0	0	1	3	0	0	----	8/8	RandomExcursionsVariant
0	1	0	0	0	1	3	1	1	1	----	8/8	RandomExcursionsVariant
0	0	0	2	0	3	0	0	0	3	----	8/8	RandomExcursionsVariant
0	0	2	0	1	2	0	2	0	1	----	8/8	RandomExcursionsVariant
1	0	2	0	0	0	1	1	3	0	----	8/8	RandomExcursionsVariant
1	1	1	1	0	2	0	1	1	0	----	8/8	RandomExcursionsVariant
0	2	3	0	0	0	1	0	1	1	----	8/8	RandomExcursionsVariant
1	2	1	1	0	0	2	0	1	0	----	8/8	RandomExcursionsVariant
1	1	2	2	0	0	2	0	0	0	----	8/8	RandomExcursionsVariant
1	4	0	1	0	1	0	0	1	0	----	8/8	RandomExcursionsVariant
7	13	6	10	12	11	7	10	13	11	0.759756	100/100	Serial
8	13	9	8	9	10	12	12	9	10	0.971699	98/100	Serial
15	10	8	5	10	12	13	9	9	9	0.637119	99/100	LinearComplexity

Рис. 4. Результати МАГФ2 / MAFG2 results

C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	P-VALUE	PROPORTION	STATISTICAL TEST
8	12	9	12	10	7	11	6	10	15	0.699313	99/100	Frequency
11	10	11	14	7	11	11	4	13	8	0.554420	99/100	BlockFrequency
7	15	11	11	5	8	8	7	13	15	0.262249	100/100	CumulativeSums
12	6	14	12	8	5	6	12	12	13	0.334538	99/100	CumulativeSums
13	8	10	9	10	17	5	10	9	9	0.437274	100/100	Runs
12	9	13	8	7	7	12	9	8	15	0.637119	96/100	LongestRun
10	9	7	16	14	6	8	13	6	11	0.289667	99/100	Rank
14	12	11	12	9	6	10	10	8	8	0.834308	99/100	FFT
8	19	8	7	9	6	7	16	8	12	0.051942	100/100	OverlappingTemplate
0	0	0	0	0	0	0	100	0	0	0.000000 *	100/100	Universal
19	10	11	10	9	12	8	6	9	6	0.191687	100/100	ApproximateEntropy
1	1	0	0	2	3	4	3	1	0	0.012650	14/15	RandomExcursions
1	0	1	2	2	0	2	1	4	2	0.090936	15/15	RandomExcursions
5	1	0	2	1	2	1	2	1	0	0.012650	15/15	RandomExcursions
3	0	2	1	1	3	0	3	2	0	0.048716	15/15	RandomExcursions
3	0	1	0	2	1	1	2	3	2	0.162606	13/15	RandomExcursions
3	2	0	0	3	2	0	1	4	0	0.006196	15/15	RandomExcursions
1	0	2	1	2	2	1	3	3	0	0.162606	15/15	RandomExcursions
1	1	1	1	2	2	3	3	0	1	0.275709	15/15	RandomExcursions
2	1	3	1	1	2	2	0	1	2	0.437274	15/15	RandomExcursionsVariant
1	3	1	2	3	2	1	1	1	0	0.275709	15/15	RandomExcursionsVariant
1	2	3	1	1	4	1	0	0	2	0.048716	15/15	RandomExcursionsVariant
3	1	0	2	1	4	1	1	1	1	0.090936	15/15	RandomExcursionsVariant
1	3	1	1	2	1	0	3	1	2	0.275709	15/15	RandomExcursionsVariant
1	3	2	1	0	2	0	2	1	3	0.162606	15/15	RandomExcursionsVariant
1	2	3	2	1	0	2	1	1	2	0.437274	15/15	RandomExcursionsVariant
1	2	1	4	0	1	2	1	3	0	0.048716	15/15	RandomExcursionsVariant
2	1	0	4	2	0	2	3	0	1	0.025193	15/15	RandomExcursionsVariant
1	1	3	2	2	1	1	1	1	2	0.637119	15/15	RandomExcursionsVariant
0	2	2	3	3	2	0	1	2	0	0.090936	15/15	RandomExcursionsVariant
1	1	0	4	3	2	1	1	1	1	0.090936	15/15	RandomExcursionsVariant
0	2	2	0	4	2	1	1	2	1	0.090936	15/15	RandomExcursionsVariant
0	2	0	5	3	2	1	0	0	2	0.001399	15/15	RandomExcursionsVariant
0	3	1	5	1	2	1	0	1	1	0.006196	15/15	RandomExcursionsVariant
0	3	3	2	3	1	0	1	1	1	0.090936	15/15	RandomExcursionsVariant
0	2	5	4	0	0	0	0	4	0	0.000005 *	15/15	RandomExcursionsVariant
0	2	4	3	1	1	2	1	0	1	0.048716	15/15	RandomExcursionsVariant
5	10	9	9	12	7	15	10	14	9	0.514124	100/100	Serial
7	10	10	8	6	13	12	11	9	14	0.739918	98/100	Serial
13	17	6	5	9	7	12	5	13	13	0.075719	99/100	LinearComplexity

Рис. 5. Результати Pike / Pike results

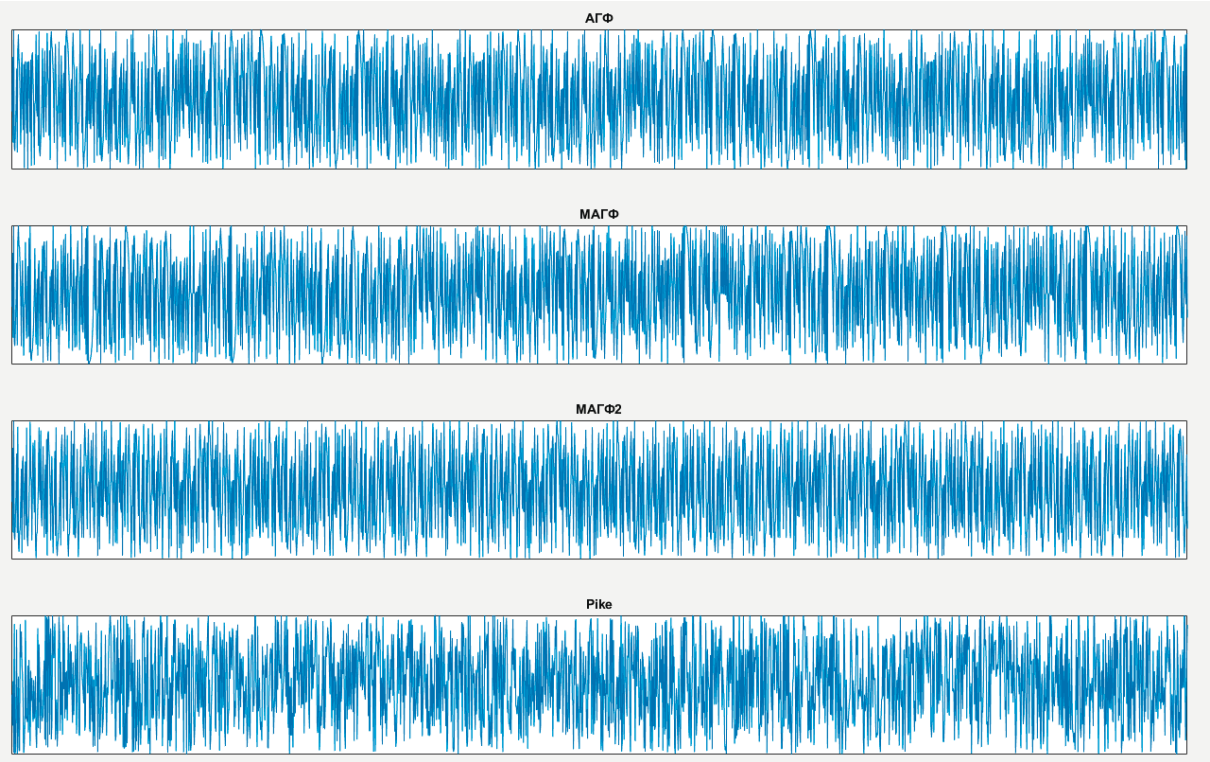

Рис. 6. Шуми, згенеровані різними алгоритмами / Noise generated by different algorithms

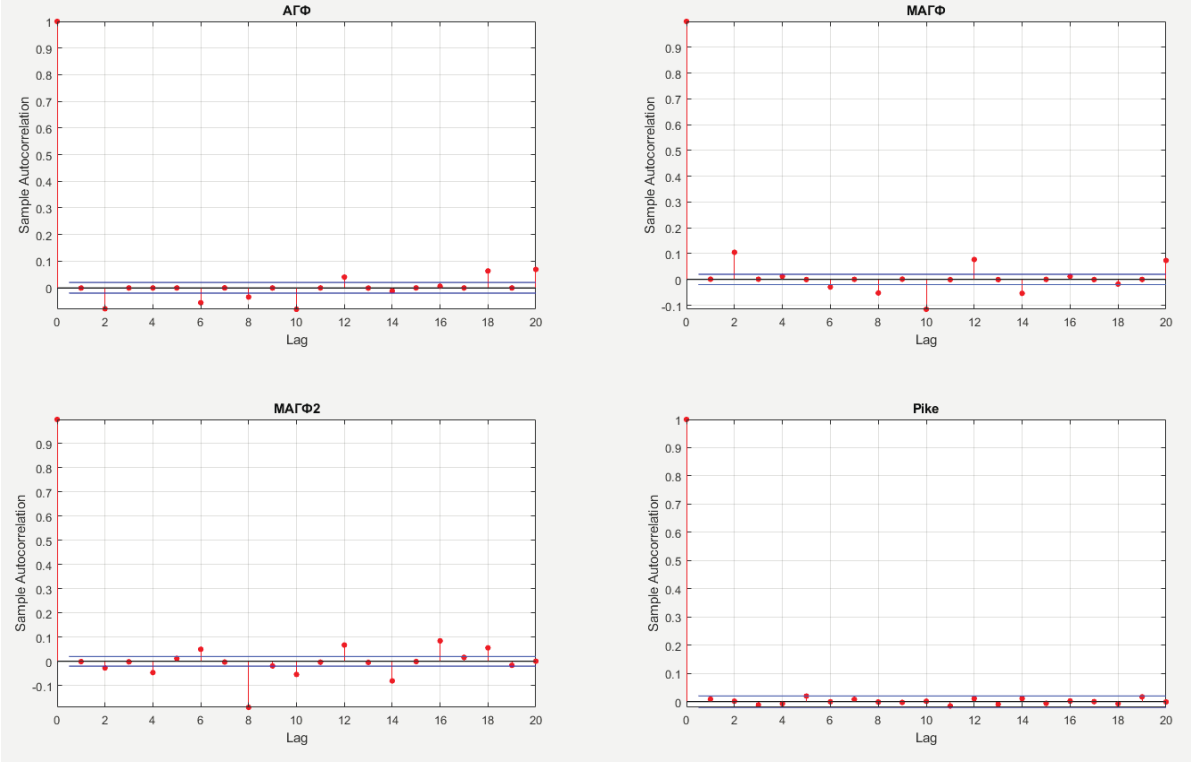

Рис. 7. Результати автокореляційної функції при m=8 / Results of the autocorrelation function at m=8

При порівнянні імплементованих алгоритмів обчислено довірчі інтервали автокореляційної функції за формулою [10, 11]:

$$sigmaNMA = \frac{1}{\sqrt{N}}, \qquad (6)$$

де N – кількість значень; $bounds = sigmaNMA \cdot [-2; +2]$ – межі довірчих інтервалів.

Для різних наборів коефіцієнтів обчислено кількість точок, які знаходяться за межами довірчих інтервалів. Коефіцієнти для кожного набору тестів були ідентичними у АГФ, МАГФ та МАГФ2. Для алгоритму Pike використано інші коефіцієнти, через більше використання пам'яті для ініціалізації. Оскільки кожне наступне значення АГФ може відрізнятись від значення МАГФ та МАГФ2 тільки на 1 (при однакових коефіцієнтах), більшу різницю між довірчими інтервалами можна спостерігати при невеликому m < 13. Нижче зображено результати автокореляційної функції для усіх сигналів із однаковими початковими коефіцієнтами, при m = 8 (рис. 8, 9).

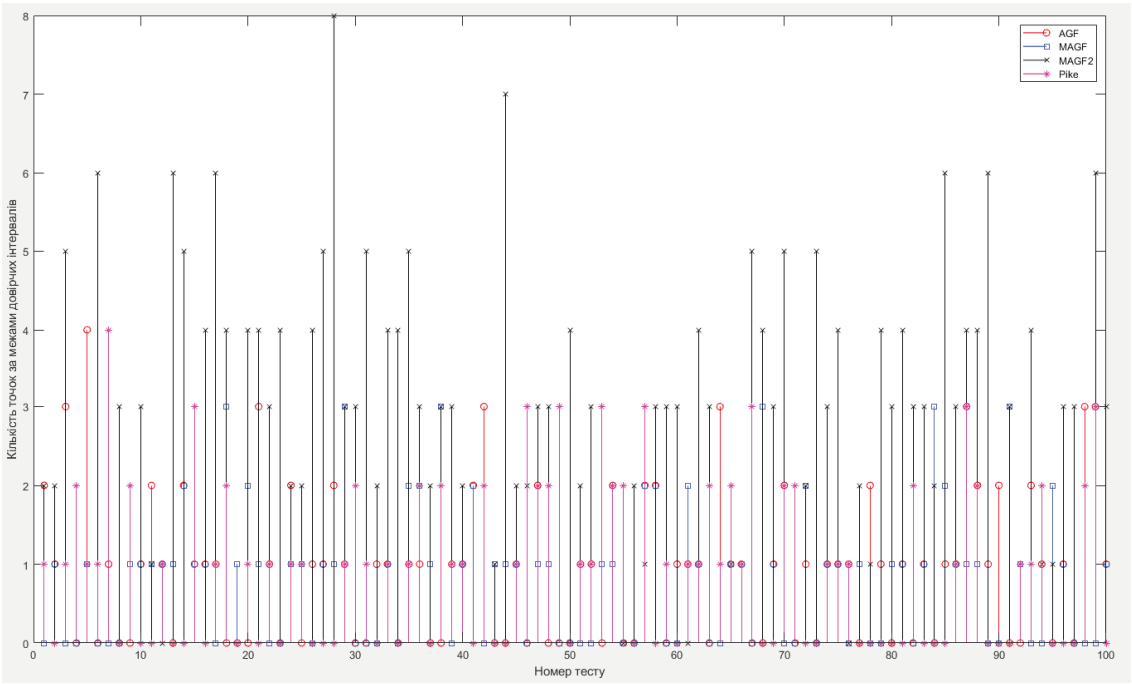

Рис. 8. Результати виходу за межі довірчих інтервалів автокореляційної функції / Results of going beyond the confidence intervals of the autocorrelation function

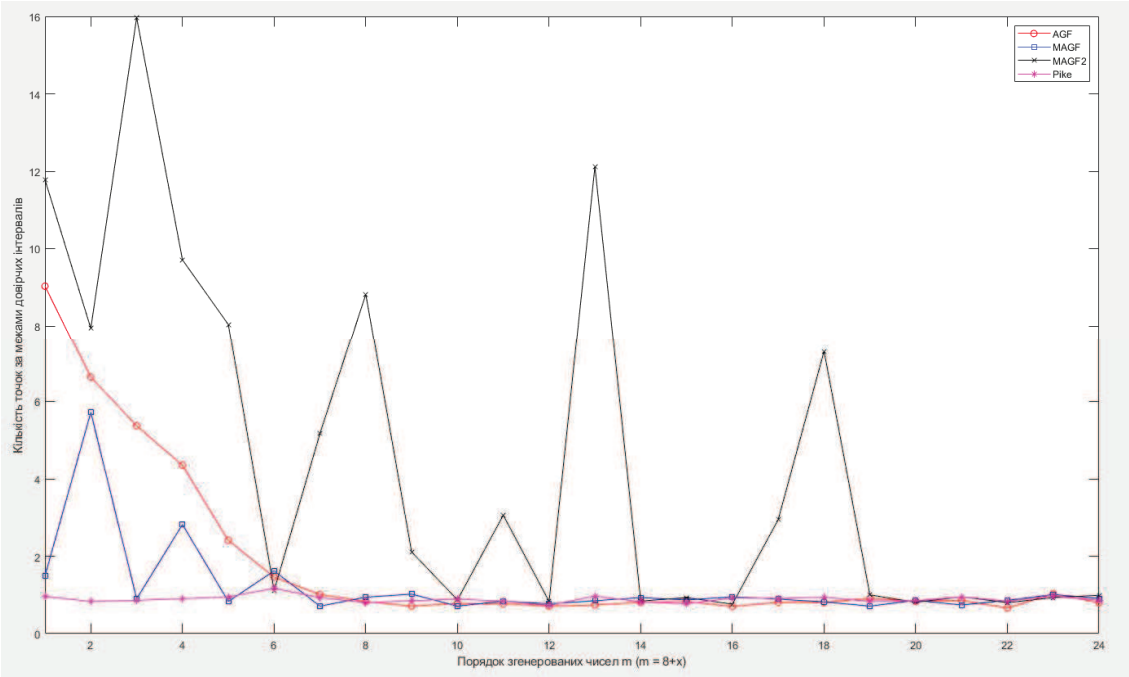

Рис. 9. Результати виходу за межі довірчих інтервалів автокореляційної функції при різних порядках m / Results of going beyond the confidence intervals of the autocorrelation function at different orders of m

Для того, щоб отримати дані з різним набором параметрів, конфігурація перевірки для кожного алгоритму була така:

- 24 ітерації для зміни періоду m від 8(byte) до 32(int32);
- 100 тестів на кожне значення m. Під час кожної ітерації змінювались початкові коефіцієнти;
- 1000000 згенерованих значень на тест.

Із отриманих результатів можна зробити кілька висновків. Результати МАГФ2 менш стабільні, ніж в інших алгоритмах. Якщо в інших алгоритмах зі збільшенням порядку згенерованих чисел, результати автокореляційної функції покращуються, то в МАГФ2 видно різкі зміни в результатах.

На межі від 0 до 8 (8–16 біт) МАГФ, АГФ та Pike відрізняються найбільше, і після цього відрізку стають на один рівень. Це означає що при генерування невеликих чисел (типу byte, short) Pike має значну перевагу над МАГФ, який водночас має кращі характеристики від АГФ.

Ті ж самі результати але в числовій формі зображено у табл. 1. Перший рядок відображає середнє арифметичне кількості точок за межами довірчих інтервалів на усіх тестах від 8 до 32 біт. У наступних рядках результати розбиті на менші інтервали, з яких також можна побачити, що при збільшенні порядку, характеристики АГФ та МАГФ покращуються. Алгоритм Pike має хороші результати на усіх інтервалах, тоді як результати МАГФ2 – значно гірші.

Також обчислено середнє арифметичне співвідношення сигнал/шум (SNR) для кожного тесту. Обчислення за формулою [12, 13]:

$$SNR = \frac{avg(signal)}{std(signal)}, \quad (7)$$

де avg() – середнє арифметичне; а std() – стандартне відхилення.

Із отриманих результатів (табл. 2) видно, що Pike та МАГФ мають кращі значення SNR, проте різниця між усіма варіантами невелика.

Табл. 1. Співвідношення кількості точок (у відсотках) за межами довірчих інтервалів до розрядності / The ratio of the number of points (in percentage) outside the confidence intervals to the bit depth

АГФ	МАГФ	МАГФ2	Pike	Тест
1,8275	1,1988	4,4079	0,8892	8–32
3,82	1,9162	8,7387	0,8837	8–16
0,8363	0.875	2,7163	0,88	16–24
0,8938	0,8	2,0325	0,9	24–32

Табл. 2. Середні значення SNR / Average SNR values

АГФ	МАГФ	МАГФ2	PIKE	Тест
1,7317	1,7316	1,7366	1,7315	8–32
1,7314	1,7298	1,7456	1,7305	8–16
1,7315	1,7323	1,7311	1,7319	16–24
1,7321	1,7327	1,7331	1,7318	24–32

Для того, щоб переконатись, чи підходить конкретний алгоритм для генерування шуму, також необхідно дослідити його порядок та спектральні характеристики.

В таблицях вище наведені значення частот, які використовуються для різних форм комунікацій. Зі збільшенням частоти, збільшуються вимоги до швидкодії та періоду [14], що повинен забезпечити вибраний алгоритм (табл. 3). Відповідно до рекомендацій захисту мовної інформації, генератор повинен забезпечити достатній період для безперервної роботи від 8 годин і далі з кроком 8 годин. При дослідженні періоду чотирьох обраних алгоритмів (до 2^{32}), виявлено що найкращі характеристики у МАГФ та PIKE. Здебільшого МАГФ2 зациклюється на кілька порядків раніше, ніж обране значення *m*. Окрім цього, період у МАГФ2 менший, ніж у АГФ. При меншому значенні m = 16, PIKE та МАГФ все ще працюють із періодом > 2^{16}. При тих же ж порядках АГФ із ймовірністю 23,5 % зациклюється раніше, МАГФ2 із ймовірністю 84 %. З чого можна зробити висновок, що найкращим вибором будуть алгоритми МАГФ та PIKE.

Табл. 3. Граничні частоти для захисту мовної інформації / Limiting frequencies for the protection of speech information

Частота			
180 Гц		5600 Гц	
Час (год)	Порядок	Час (год)	Порядок
1	$< 2^{20}$	1	$< 2^{25}$
8	$< 2^{23}$	8	$< 2^{28}$
16	$< 2^{24}$	16	$< 2^{29}$
24	$< 2^{24}$	24	$< 2^{29}$

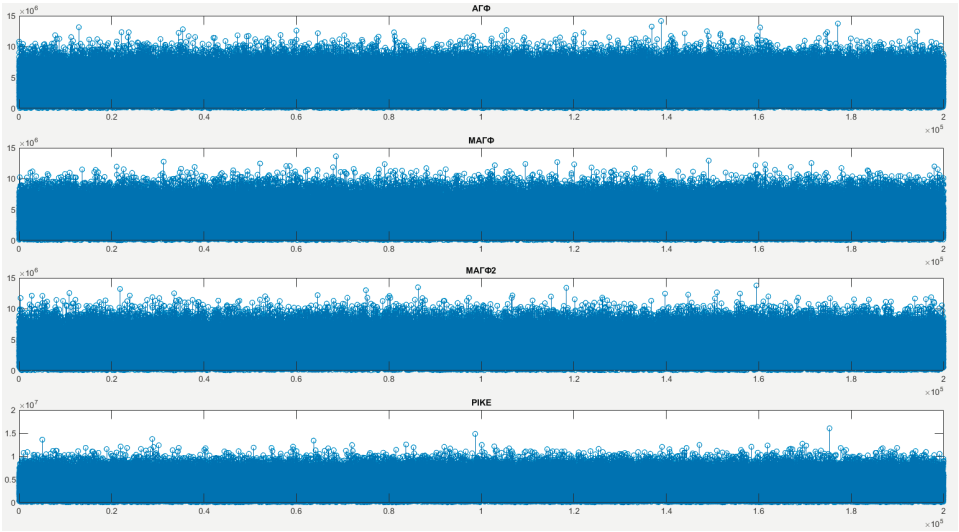

Рис. 10. Частотний розподіл згенерованих шумів / Frequency distribution of generated noise

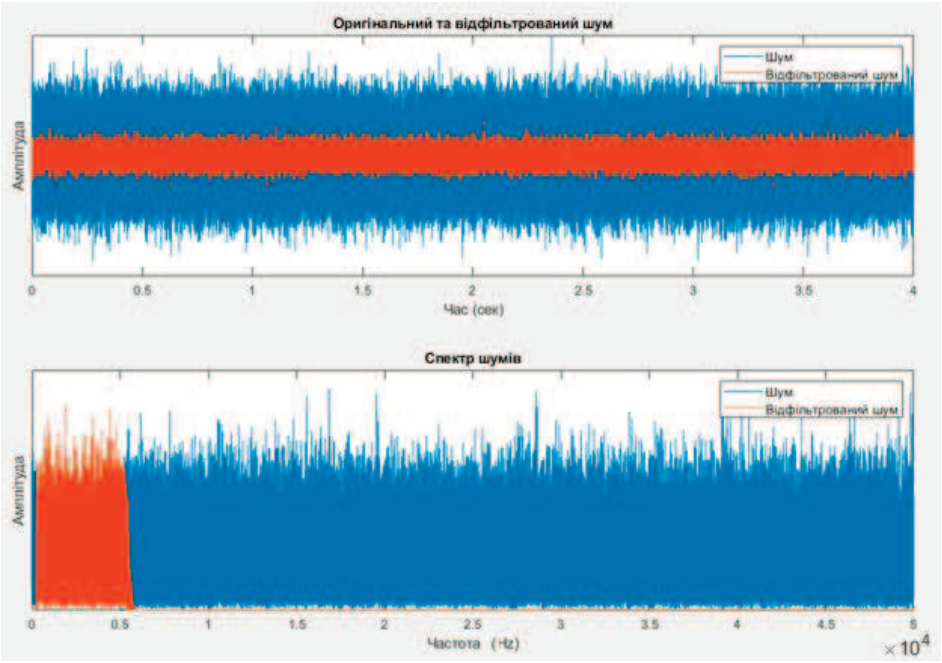

Рис. 11. Шум та його фільтрація в заданих частотах / Noise and its filtering in specified frequencies

Оскільки в ГПВЧ значення рівномірно розподілені (рис. 10), результати потрібно пропустити через фільтр для того, щоб отримати згенерований шум у необхідних межах частот. Згенерований шум на підставі алгоритму МАГФ відповідає частотам 180–5600 Гц (рис. 11).

Обговорення результатів дослідження. У роботі [8] запропоновано модифікацію алгоритму АГФ. МАГФ1 використовує кількість одиниць, у бітовому представленні результату генератора, як ще один доданок до регістрів. Через це різниця між результатами АГФ та МАГФ1 буде відрізнятись не більше, ніж на 1 на кожній ітерації, однак цього достатньо, щоб пройти тести NIST. Аналогічно до представлених у роботі тестів, додатково протестовано більше початкових значень ГПВЧ із різними порядками між 2^8 до 2^{32}, і на кожному порядку обчисленні середні значення автокореляційної функції.

У роботі [6] вперше описано алгоритм PIKE і вплив цієї модифікації на результати генератора із боку криптостійкості. Крім трьох генераторів Фібоначі з запізненням, використання біту переносу покращує статистичні характеристики згенерованих послідовностей [2]. Подібно до цієї модифікації, запропоновано алгоритм МАГФ2, в якому теж використовується біт переносу. На відміну від PIKE, використано один адитивний генератор Фібоначі, щоб він був схожий за будовою з АГФ та МАГФ1. Таким чином, вдалось дослідити саме вплив різних модифікацій на адитивні генератори Фібоначі.

У роботі [5] досліджено результати статистичних тестів для різних типів ГПВЧ із трьома пакетами тестування та двома графічними тестами. Аналогічно до цих тестів, протестовано алгоритми Фібоначі та їх модифікації. На відміну від проведених у роботі графічних тестів, представлено частотний спектр сигналів, які згенеровані відповідними ГПВЧ. Авторами зазначено, що, окрім статистичних тестів, слід проводити додаткові перевірки, залежно від застосування ГПВЧ. Тому додатково представлено результати автокореляційної функції і виходи за межі довірчих інтервалів для кожного алгоритму, які є важливою характеристикою генераторів шуму [14]. Таким чином, виявлено залежність між порядком згенерованих значень та результатами автокореляційної функції.

У роботі [7] описано підхід до розробки цифрових генераторів шуму та залежність між необхідними частотними характеристиками генератора шуму і ГПВЧ. Аналогічно складено табличку з необхідними характеристиками генератора шуму для захисту мовної інформації та протестовано відповідність алгоритмів ГПВЧ цим характеристикам.

Отже, за результатами роботи, можна сформулювати наукову новизну і практичну значущість результатів дослідження.

Наукова новизна отриманих результатів дослідження – досліджено чотири ГПВЧ на підставі алгоритму АГФ та його модифікацій з різним набором вхідних даних і вихідних порядків.

Практична значущість реузультатів дослідження – вибір алгоритму ГПВЧ для побудови генератора цифрових шумів, який може використовуватись для захисту мовної інформації.

Висновки / Conclusions

Проведено порівняльний аналіз досліджених обчислювальних характеристик цифрових шумів для оптимального вибору алгоритму ГПВЧ, використаного в якості генератора цифрового шуму. За результатами виконаної роботи можна зробити основні висновки.

Спроектовано середовище для тестування ГПВЧ у контексті генераторів шуму. Протестовано чотири різних алгоритми на підставі адитивного алгоритму Фібоначі. Досліджено їхні автокореляційні функції та SNR. Знайдено закономірність між збільшенням порядку згенерованих значень і якістю автокореляційної функції для алгоритму АГФ та МАГФ.

З'ясовано, що, незважаючи на кращі результати тестів NIST, порівняно з АГФ та МАГФ, алгоритм МАГФ2 має гірші автокореляційні, SNR та характеристики періоду. Встановлено, що алгоритм Pike має чудові характеристики на усіх досліджених межах і тестах, що особливо важливо при малих порядках ($m < 16$). Однак він повільніший, ніж інші алгоритми, якщо складність алгоритмів АГФ, МАГФ та МАГФ2 – O (n), то для Pike – O (3 n). Визначено, що реальна ж складність алгоритму в середньому – O (2,75 n), проте різниця в 2,75 раза може бути критичною. Якщо від алгоритму не вимагається криптостійкість і порядок згенерованих чисел>2^{16}, тоді МАГФ є кращою альтернативою для генерування шуму, де важлива швидкодія.

З'ясовано, що модифікація одного АГФ за рахунок використання біту переносу (МАГФ2) не покращила якість ГПВЧ у контексті періоду, хоча той самий спосіб використовується на трьох внутрішніх генераторах алгоритму PIKE. Запропоновано використовувати біт переносу для покращення статистичних характеристик тільки після дослідження періоду модифікованої функції.

Подальші дослідження впливу різних підходів до модифікації ГПВЧ дадуть змогу покращити такі критичні властивості ГПВЧ, як швидкодія, статистичні характеристики та період для імплементації ефективніших генераторів шуму.

References

[1] Brent, R. P. (1992). Uniform random number generators for supercomputers.

[2] Ferguson, N., Schneier, B., & Kohno, T. (2010). Cryptography Engineering: Design Principles and Practical Applications.

[3] SP 800-22 Rev. 1a. A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications. (2010). SP 800-22 Rev. 1a. A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications.

[4] Mohamed, M. A., Awang, M. I. (2017). An empirical analysis of six pseudo-random number generators. Far East Journal of Electronics and Communications, 17, 6, 1373–1388. http://dx.doi.org/10.17654/EC017061373

[5] Kamalika, B., Sukanta, D. (2022). A search for good pseudo-random number generators: Survey and empirical studies. Computer Science Review, 45, 100471. https://doi.org/10.1016/j.cosrev.2022.100471
[6] Ross, J. A. (1994). On Fibonacci Keystream Generators. Fast Software Encryption, 346–352. https://doi.org/10.1007/3-540-60590-8_26
[7] Mingesz, R., Farag, D. (2019), Implementing software defined noise generators. 25th International Conference on Noise and Fluctuations. https://doi.org/10.5075/epfl-ICLAB-ICNF-269296
[8] Mandrona, M. M., Maksymovych, V. M., Kostiv, Yu. M., Harasymchuk, O. I. (2014). Modyfikatsiia adytyvnoho heneratora Fibonachchi z zapiznenniam. Suchasnyi zakhyst informatsii, 2, 57. http://nbuv.gov.ua/UJRN/szi_2014_2_10
[9] Schneier, B. (1993). Applied Cryptography Protocols, Algorithms, and Source Code in C.
[10] Solomon, W. G., Guang, G. (2005). Signal design for good correlation: for wireless communication, cryptography, and radar.
[11] Kun II Park (2018). Fundamentals of Probability and Stochastic Processes with Applications to Communications, Springer.
[12] Lyons, R. (2010) Understanding Digital Signal Processing, 3rd Edition.
[13] Sherman, C., Butler, J. (2007). Transducers and Arrays for Underwater Sound. Springer Science & Business Media, 276. https://doi.org/10.1007/978-0-387-33139-3
[14] Zasoby aktyvnoho zakhystu movnoi informatsii z akustychnymy ta vibroakustychnymy dzherelamy vyprominiuvannia. Klasyfikatsiia ta zahalni tekhnichni vymohy. (2020). https://usts.kiev.ua/wp-content/uploads/2020/07/nd-tzi-r-001-2000.pdf

O. V. Isakov, S. S. Voitusik
Lviv Polytechnic National University, Lviv, Ukraine

COMPARATIVE ANALYSIS OF DIGITAL NOISE GENERATED BY ADDITIVE FIBONACCI GENERATORS

Noise generators and pseudorandom number generators (PRNGs) are widely used in the field of information technology, including cybersecurity, for modeling, authorization key generation, and technical protection of information. It has been found that the characteristics of digital noise directly depend on the chosen PRNG algorithm. To determine the quality of the generated noise, special tests are performed, which are primarily applied to the sequence generated by the PRNG. The results of digital noise generated by an PRNG based on four different algorithms of additive Fibonacci generators (AFG) are investigated. The choice of generators of the same type allowed us to analyze the effect of different modifications on the final result of the generated sequences to determine their advantages and disadvantages. Digital signal processing techniques such as frequency, autocorrelation and visual analysis, signal-to-noise ratio, and statistical tests of the NIST package were used to test the noise and generated sequences. Functions for interpreting the obtained data were developed using the MATLAB (DSP System Toolbox) application package and the C programming language for automating NIST tests. It has been found that for effective testing, specific stages and their sequence should be determined: determination of the PRNG period, statistical tests of the NIST package, calculation of the autocorrelation function, and other methods of digital signal processing. It was found that modification of one AFG by using a carry bit (MAFG2) does not improve the results of the generated sequence, unlike the PIKE algorithm, which consists of three AFGs. The MAFG algorithm showed better results during the period testing and at the same time passed NIST tests, unlike the unmodified version. The dependence between the order of the generated sequences and the results of their autocorrelation function was revealed. It is proposed that, in addition to general statistical tests, applied tests should be carried out when choosing or developing a new generator, its effectiveness should be checked under the conditions required by existing standards and requirements. The compliance of the generated digital noise with the requirements for devices for technical protection of information, namely the protection of speech information, has been established.

Keywords: pseudorandom number generator, noise generator, NIST(National Institute of Standards and Technology), digital signal processing, abular functions, formulas of central finite differences, calculation of derivatives.

Інформація про авторів:
Ісаков Олександр Володимирович, аспірант, кафедра безпеки інформаційних технологій.
Email: oleksandr.v.isakov@lpnu.ua; https://orcid.org/0009-0007-4632-9492
Войтусік Степан Степанович, канд. фіз.-мат. наук, доцент, кафедра безпеки інформаційних технологій.
Email: stepan.s.voitusik@lpnu.ua; https://orcid.org/0000-0003-4234-3303

Цитування за ДСТУ: Ісаков О. В., Войтусік С. С. Порівняльний аналіз цифрових шумів, згенерованих адитивними генераторами Фібоначі. *Український журнал інформаційних технологій.* 2023. Т. 5, № 1. С. 67–76.
Citation APA: Isakov, O. V., Voitusik, S. S. (2023). Comparative analysis of digital noise generated by additive fibonacci generators. *Ukrainian Journal of Information Technology*, 5(1), 67–76. https://doi.org/10.23939/ujit2023.01.067