https://doi.org/10.31891/2219-9365-2023-73-1-22
УДК 004.421.5.052.2:004.77(045)

Ольга САЛІЄВА
Вінницький національний технічний університет
https://orcid.org/0000-0003-2388-7321
salieva8257@gmail.com
Василь КАРПІНЕЦЬ
Вінницький національний технічний університет
https://orcid.org/0000-0001-8148-2002
karpinets@gmail.com
Анатолій ГРИЦАК
Вінницький національний технічний університет
https://orcid.org/0000-0002-0776-9889
grytsak.a.v@gmail.com
Павло ПАВЛОВСЬКИЙ
Вінницький національний технічний університет
https://orcid.org/0009-0001-1730-4102
prepod@vntu.net
Ірина БОНДАРЕНКО
Вінницький національний технічний університет
https://orcid.org/0000-0003-2104-657X
fm.ub15b.bondarenko@gmail.com

# ПІДВИЩЕННЯ СТІЙКОСТІ КРИПТОГРАФІЧНИХ АЛГОРИТМІВ У БАГАТОКОРИСТУВАЦЬКИХ WEB-РЕСУРСАХ НА ОСНОВІ ГЕНЕРАТОРІВ ВИПАДКОВИХ ЧИСЕЛ, ЩО ВРАХОВУЮТЬ ЕНТРОПІЮ ПОВЕДІНКИ КОРИСТУВАЧА

*У роботі розглянуто проблему підвищення захищеності Web-ресурсів стійкими криптоалгоритмами на основі генераторів випадкових чисел, що враховують ентропію поведінки користувача у багатокористувацькому середовищі. Під час дослідження розроблено алгоритм надійного джерела ентропії істино випадкових чисел для табличного генератора, який використовує заздалегідь підготовлені таблиці, що містять перевірені нерелевантні числа. Для програмної реалізації підсистеми шифрування розроблено алгоритм роботи табличного генератора, при цьому здійснено об'єднання окремих модулів генератора випадкових чисел, а саме алгоритму заповнення буфера табличного генератора та алгоритму вибору з нього випадкового числа. Вихідні дані генератора збережено у текстовий файл, який використовувався в якості вхідних даних, необхідних для проведення тестування на випадковість. При цьому використовувалася методика тестування, запропонована Національним інститутом стандартизації і технологій США – NIST на основі пакету статистичних тестів, спрямованих на визначення міри випадковості двійкових послідовностей, утворених генераторами випадкових чисел. Основною перевагою є те, що ці тести засновані на різних статистичних властивостях, які належать лише випадковим послідовностям. З отриманих результатів було зроблено висновок, що послідовність повністю задовольняє критеріям випадковості. Крім того, проведено порівняльний аналіз характеристик розробленого генератора випадкових чисел з класичними реалізаціями, на основі якого визначено його переваги.*

*Ключові слова: Web-ресурс, табличний генератор випадкових чисел, ентропія, випадкова послідовність чисел, NIST.*

Olga SALIIEVA, Vasyl KARPINETS,
Anatoliy HRYTSAK, Pavlo PAVLOVSKYI, Iryna BONDARENKO
Vinnytsia National Technical University

# INCREASING THE STABILITY OF CRYPTOGRAPHIC ALGORITHMS IN MULTI-USER WEB RESOURCES BASED ON RANDOM NUMBER GENERATORS THAT TAKE INTO ACCOUNT THE ENTROPY OF USER BEHAVIOR

*The paper considers the problem of increasing the security of Web resources by stable crypto-algorithms based on random number generators that take into account the entropy of user behavior in a multi-user environment. The research developed an algorithm for a reliable entropy source of truly random numbers for a table generator that uses pre-prepared tables containing verified irrelevant numbers. For the software implementation of the encryption subsystem, an algorithm for the work of the tabular generator was developed, while separate modules of the random number generator were combined, namely, the algorithm for filling the buffer of the tabular generator and the algorithm for selecting a random number from it. The output data of the generator was saved in a text file, which was used as input data necessary for randomness testing. At the same time, the testing method proposed by the US National Institute of Standards and Technology - NIST based on a NIST package of statistical tests aimed at determining the degree of randomness of binary sequences generated by random number generators. The main advantage is that these tests are based on various statistical properties belonging only to random sequences. From the obtained results, it was concluded that the sequence fully satisfies the criteria of randomness. In addition, we will conduct a comparative*

*analysis of the characteristics of the developed random number generator with classical implementations, on the basis of which its advantages are determined.*

*Keywords: Web-resource, tabular generator of random numbers, entropy, random sequence of numbers.*

**Постановка проблеми у загальному вигляді**
**та її зв'язок із важливими науковими чи практичними завданнями**

Швидкі темпи розвитку та впровадження Web-технологій в різноманітні сфери суспільної діяльності зумовили необхідність удосконалення алгоритмів їхнього захисту. Адже з кожним роком зростає не тільки кількість Web-додатків, але й збільшується кількість нових кіберзагроз та вразливостей Web-ресурсів до атак. Для підвищення захищеності Web-інфраструктури використовують різні методи і засоби, зокрема стійкі криптографічні алгоритми на основі генераторів випадкових чисел (ГВЧ). Для генерації випадкового потоку застосовується джерело ентропії, що ґрунтується на фізично випадкових явищах, таких як шум звукової карти, лічильник тактів процесора, рух миші і т. п.

У даній роботі, щоб підвищити стійкість криптографічних алгоритмів доцільно розробити ГВЧ, який використовує швидкодіюче джерело з метою отримання істино випадкових послідовностей для їхньої ініціалізації. Для досягнення поставленої мети необхідно провести аналіз предметної області, на основі якого обрати джерело ентропії та вид ГВЧ; розробити алгоритм роботи ГВЧ і здійснити його програмну реалізацію; провести статистичне тестування розробленого генератора.

**Аналіз досліджень та публікацій**

На сьогодні багато веб-застосунків мають загальновідомі вразливості, використання яких дозволяє успішно проводити атаки на веб-ресурси. Одним із варіантів вирішення даної проблеми є підвищення рівня захищеності Web-інфраструктури за допомогою використання стійких криптографічних алгоритмів на основі ГВЧ. Удосконаленню властивостей різних типів генераторів присвячено багато наукових праць. Так, для покращення якісних характеристик двійкової вибірки, автори роботи [1] пропонують використовувати програмні генератори хаосу, які на відміну від лінійно-конгруентних генераторів мають набагато більший період. У [2] було здійснено криптоаналіз машинного навчання квантового генератора випадкових чисел. З рекомендаціями щодо джерел ентропії, які використовуються для генерації випадкових бітів можна ознайомитися в [3]. Автори праці [4] пропонують новий алгоритм генерації псевдовипадкових чисел, у якому в якості псевдовипадкової функції використовується решітка відображення із затримкою. Роботу генератора псевдовипадкових чисел малої потужності на основі хаотичної карти Lemniscate, що забезпечує широкий діапазон контрольних параметрів та надає чудову продуктивність, описано в [5]. Адаптивна карта Чирікова для генерації псевдовипадкових чисел у потоковому шифруванні на основі хаосу пропонується у роботі [6]. Ознайомитися з дослідженнями, пов'язаними з роботою апаратного генератора псевдовипадкових чисел, що використовує стохастичні обчислення та логістичну карту можна в [7]. Швидка криптосистема гібридного зображення на основі генератора випадкових змін і модифікованої логістичної карти відображена у праці [8].

**Виділення невирішених раніше частин загальної проблеми**

Стійкість криптоалгоритмів напряму залежить від криптографічної якості ГВЧ, які бувають трьох типів: апаратні, програмні та табличні. Зазначимо, що апаратним ГВЧ притаманні потенційно високі часові та матеріальні витрати на їх конструювання. Крім того, швидкість генерації випадкових чисел апаратних ГВЧ нижча, ніж у програмних. У свою чергу, програмні генератори повністю детерміновані. Зазвичай, вони використовують різні складні функції для обчислення псевдовипадкових чисел. Відповідно, послідовності, отримані внаслідок роботи таких генераторів, є в тій чи іншій мірі передбачуваними та відтворюваними і не підходять, наприклад, для використання у криптографічних додатках. Враховуючи зазначені недоліки апаратних та програмних ГВЧ, варто звернути увагу на табличні ГВЧ, які видають істино випадкову послідовність та не потребують фізичної наявності модуля генерації в системі. Проте для зберігання таблиць, в які заносяться випадкові числа, потрібний великий об'єм пам'яті ЕОМ. Для вирішення даної проблеми пропонується використати ентропію поведінки користувача в якості Seed даних табличного ГВЧ, оскільки таке джерело ентропії є швидкодіючим та фактично нескінченим. Таблицю можна буде заповнювати під час роботи системи, що надасть можливість використовувати менші об'єми пам'яті.

**Формулювання цілей статті**

Метою дослідження є підвищення стійкості криптографічних алгоритмів у багатокористувацьких Web-ресурсах на основі табличних ГВЧ, що враховують ентропію поведінки користувача.

**Виклад основного матеріалу**

Розробка алгоритму надійного джерела ентропії на основі поведінки користувача Web-ресурсу.

Якщо розглядати випадок використання ГВЧ як складову криптографічних алгоритмів у роботів Web-

ресурсу, наприклад соціальної мережі, то поведінка користувачів даної системи може слугувати в якості джерела ентропії істино випадкових чисел для табличного ГВЧ.

Розглянемо отримання випадкових значень на прикладі популярної соціальної мережі Facebook.

Гортаючи стрічку новин, користувач підсвідомо здійснює випадкові дії, які можна відслідкувати та використати в якості ентропії для ГВЧ.

Для аналізу випадковості таких дій, проведемо експеримент, в якості випадкового значення будемо використовувати значення координат курсору користувача.

Відкривши консоль розробника у веб-браузері Chrome підпишемось на подію «onmousemove», після кожного спрацювання події будемо отримувати об'єкт інтерфейсу MouseEvent, який представляє події, що відбуваються внаслідок взаємодії користувача з вказівним пристроєм (наприклад, мишею).

Об'єкт MouseEvent має властивості, що наведені в табл. 1.

Таблиця 1

**Властивості об'єкту MouseEvent**

| Назва | Опис |
|---|---|
| MouseEvent.altKey | Повертає true, якщо клавіша alt була натиснута під час запуску події миші. |
| MouseEvent.button | Номер кнопки, яка була натиснута (якщо є) під час запуску події миші. |
| MouseEvent.buttons | Кнопки, які натискаються (якщо є) під час запуску події миші. |
| MouseEvent.clientX | Координата X вказівника миші в локальних координатах (вміст DOM). |
| MouseEvent.clientY | Координата Y вказівника миші в локальних координатах (вміст DOM). |
| MouseEvent.ctrlKey | Повертає істину, якщо клавіша керування була натиснутою під час запуску події миші. |
| MouseEvent.movementX | Координата X вказівника миші відносно позиції останньої події mousemove. |
| MouseEvent.movementY | Координата Y вказівника миші відносно позиції останньої події переміщення миші. |
| MouseEvent.pageX | Координата X вказівника миші відносно всього документа. |
| MouseEvent.pageY | Координата Y вказівника миші відносно всього документа |
| MouseEvent.screenX | Координата X вказівника миші в глобальних (екранних) координатах. |
| MouseEvent.screenY | Координата Y вказівника миші в глобальних (екранних) координатах |

Для отримання випадкових значень можна обирати одне значення властивості, або комбінацію з декількох значень.

Під час кожного спрацювання події «onmousemove» будемо брати значення властивостей MouseEvent.clientX та MouseEvent.clientY та записувати їх в глобальні зміні **X** та **У**.

Також запустимо функцію setInterval, з якої кожні 500 мс обиратимемо значення змінних **X** і **У** та записувати їх в масив ArrayX та ArrayY.

На рис. 1 відображено вигляд консолі розробника веб-браузера Chrome.

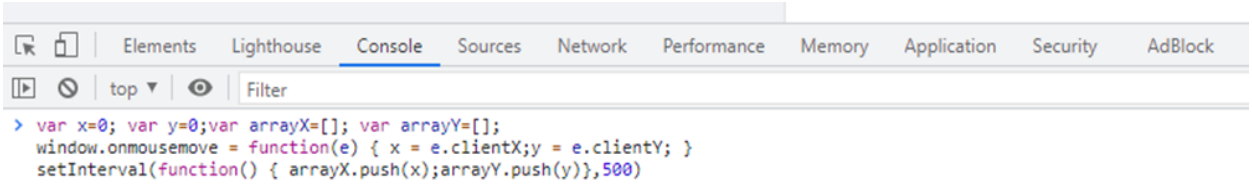


**Рис. 1. Вигляд консолі розробника веб-браузера Chrome**

Для збору даних двом користувачам було запропоновано переглянути стрічку новин, не оголошуючи їм суті експерименту. При цьому враховано той факт, що при нерухомому стані мишки, таблиця заповниться однаковими даними, тому вибірки такого роду будуть відкидатися.

На основі отриманої інформації побудовано графіки розподілу координат курсору користувачів А та Б (рис. 2).

**ISSN 2219-9365**

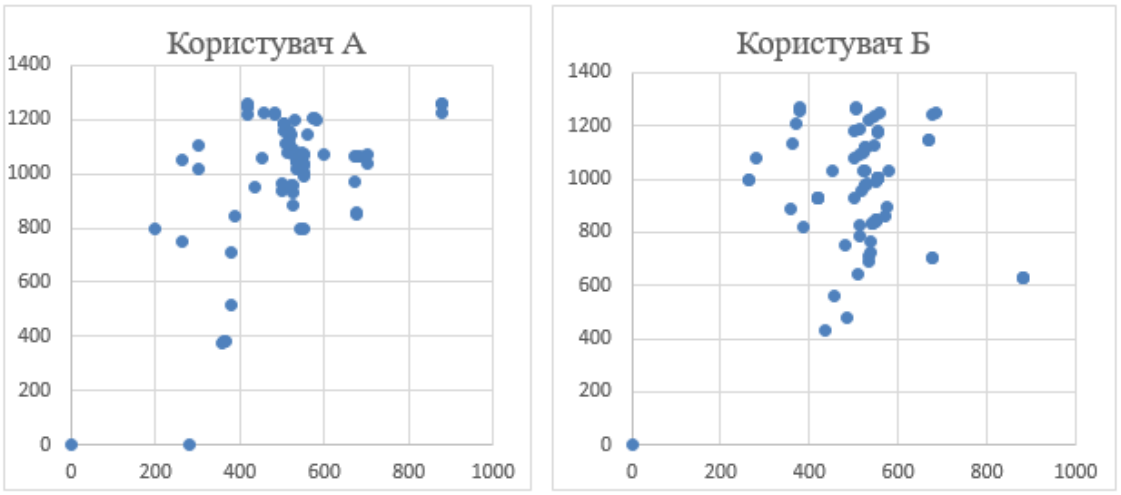


**Рис. 2. Графіки розподілу координат курсору користувачів А та Б**

З графіків видно, що між точками, які є координатами вказівника миші в локальних координатах, немає ніякої залежності.

Результуюча послідовність бітів повинна мати розподіл максимально близький до рівномірного. Для досягнення бажаного результату потрібен окремий важливий етап, званий постобробкою та реалізований шляхом застосування різноманітних спеціальних алгоритмів (рис. 3).

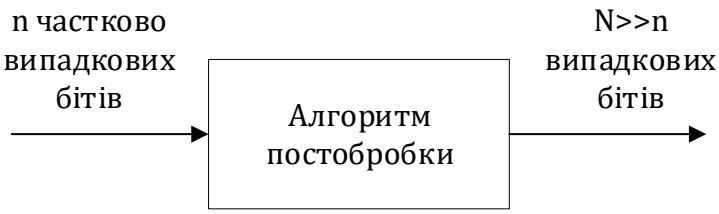


**Рис. 3. Схема процесу постобробки випадкової послідовності**

В якості методу постобробки використаємо один з варіантів спеціальних простих коректорів, а саме застосування операції XOR до бітів послідовностей, отриманих від двох або більше паралельно працюючих генераторів. Застосуємо XOR для послідовностей отриманих від користувача А та користувача Б (рис. 4).

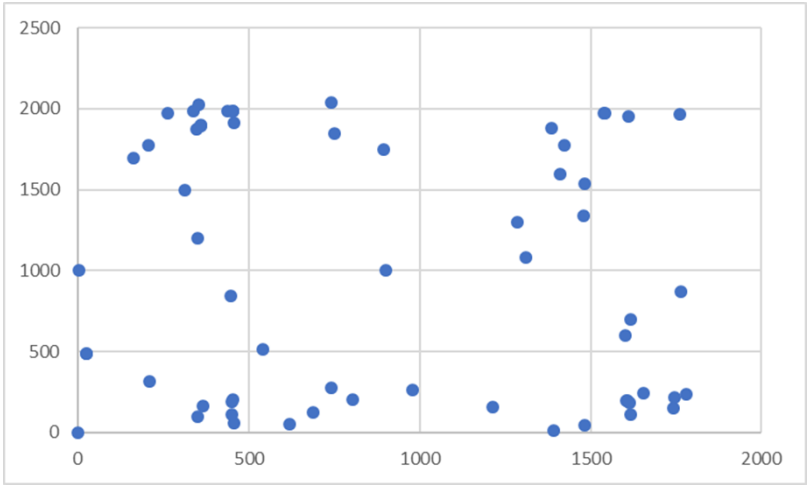


**Рис. 4. Графік розподілу після постобробки**

Розглянувши графік зображений на рис. 4 можна зробити висновок про позитивний вплив функції постобробки та наближення розподілу до рівномірного.

Враховуючи кількість користувачів, які одночасно активно використовують систему, кількість

ентропії, яку можна отримати розробленим методом є нескінченою. Крім того, в алгоритмі немає жодних складних математичних операцій, тому можна зробити висновок про швидкодію такого джерела ентропії.

Використовуючи поведінку користувачів як джерело ентропії для табличного ГВЧ, можна позбутись основного його недоліку – тримати у пам'яті таблицю великих розмірів, оскільки таке джерело ентропії є швидкодіючим та фактично нескінченим. Таблицю можна буде заповнювати під час роботи системи, що дасть можливість використовувати менші об'єми пам'яті.

**Розробка алгоритму роботи табличного ГВЧ.**

Проаналізувавши отриману інформацію можна об'єднати окремі модулі ГВЧ, а саме алгоритм заповнення буфера табличного ГВЧ та алгоритм вибору з нього випадкового числа, для подальшої програмної реалізації підсистеми шифрування.

Розроблений алгоритм можна розділити на два етапи.

**Етап 1.** Заповнення буферу табличного ГВЧ, який складається з таких кроків:

Крок 1 Обираються деякі величини, які характеризують стан системи, джерелом зміни якої є користувач. У даній реалізації в якості показників ентропії будуть використовуватись координати курсора користувача X та У.

Крок 2. Вираховується $Z_n = X_n \oplus Y_n$, яке є результатом ентропії поведінки користувача.

Крок 3. Перевіряється рівність $Z_n$ та $Z_{n-1}$. Дана перевірка необхідна для недопущення заповнення буферу табличного ГВЧ однаковими бітами. Такий сценарій можливий коли користувач не здійснює жодних дій, в такому випадку користувач перестає бути джерелом ентропії.

Крок 4. Надсилається $Z_n$ з клієнтської частина на сервер.

Крок 5. На стороні сервера розраховується $Q = Z_{An} \oplus Z_{Bn}$, де $Z_{An}$ – значення ентропії користувача А, а $Z_{Bn}$ значення ентропії користувача Б, отримане аналогічним чином.

Крок 6. У комірку таблиці з індексом $i$ записується Q.

Крок 7. Інкрементується та записується $i$.

Крок 8. Здійснюється перевірка: якщо $i$ більше розміру таблиці, то $i$ присвоюється значення 0.

**Етап 2.** Вибір випадкового числа, який складається з таких кроків:

Крок 1. У модуль передається бажана довжина послідовності $n$.

Крок 2. З комірки таблиці з індексом $i$ вибирається число $X_i$.

Крок 3. Інкрементується та записується $i$.

Крок 4. Перевіряється чи довжина числа $X_i$ задовільняє бажаному $n$.

Крок 5. Якщо довжина $X_i$ менша $n$, то вибирається наступне число $X_{i+1}$ (повторюються кроки 2, 3) та конкатинується до $X_i$. Числа вибираються та конкатинуються до тих пір, доки довжина числа $X_i$ не задовольняє бажаному $n$.

Реалізувавши алгоритм роботи ГВЧ, було розроблено пакет програм криптографічної підсистеми захисту інформації, зокрема, здійснено програмну реалізацію серверної частини алгоритму табличного ГВЧ для платформи ASP.NET Core, та клієнтської для Angular2.

**Аналіз підвищення стійкості криптографічних алгоритмів за рахунок використання в них розробленого ГВЧ.**

Оскільки є пряма залежність стійкості криптографічних алгоритмів від якості та стійкості до криптоаналізу ГВЧ, які в них використовуються, то можна стверджувати, що мета даної роботи буде досягнута у разі якщо характеристики розробленого ГВЧ, а також його загальна стійкість до криптоаналізу буде перевершувати класичні ГВЧ.

Спочатку здійснимо статистичне тестування за допомогою пакету NIST, що містить тести випадковості. Для їх походження скористаємось офіційним додатком національного інститут стандартів та технологій NIST Statistical Test Suite.

Вказавши шлях до файлу, в якому знаходиться заздалегідь згенерована розробленим генератором послідовність та обравши всі можливі тести, здійснимо тестування послідовності, результати якого будуть збережені у текстовому файлі в кореневій директорії програми.

Фрагмент файлу кінцевого звіту, сформованого пакетом NIST STS, для декількох перших тестів наведено на рис. 7.

| C1 | C2 | C3 | C4 | C5 | C6 | C7 | C8 | C9 | C10 | P-VALUE | PROPORTION | STATISTICAL TEST |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 135 | 149 | 122 | 96 | 117 | 151 | 164 | 120 | 103 | 123 | 0.00014 | 0.993 | frequency |
| 121 | 109 | 134 | 121 | 126 | 133 | 137 | 134 | 137 | 128 | 0.775277 | 0.9906 | block-frequency |
| 126 | 133 | 142 | 143 | 102 | 128 | 83 | 142 | 153 | 128 | 0.000292 | 0.9914 | cumulative-sums |
| 131 | 130 | 114 | 114 | 108 | 146 | 108 | 140 | 154 | 135 | 0.027549 | 0.9922 | cumulative-sums |
| 131 | 130 | 120 | 132 | 140 | 128 | 139 | 131 | 108 | 121 | 0.701879 | 0.9906 | runs |

**Рис. 7. Фрагмент результату проходження тестів**

На рис. 8 представлена діаграма, що характеризує попадання частки послідовностей, що пройшли кожен тест у довірчий інтервал [0,96; 1].

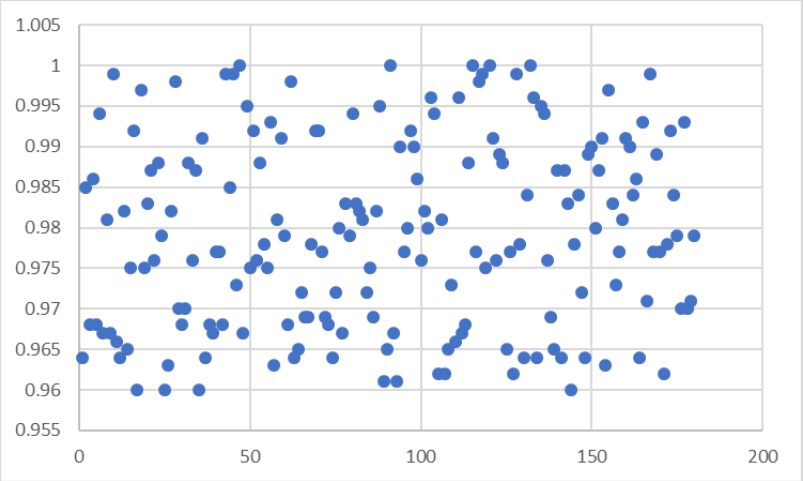


**Рис. 8. Результати тестування послідовності згенерованої розробленим ГВЧ**

З отриманих результатів можна зробити висновок, що послідовність повністю задовольняє критеріям випадковості.

На останок проведемо порівняльний аналіз розробленого ГВЧ з класичними реалізаціями (табл. 2).

Таблиця 2

**Порівняльна характеристика різних типів ГВЧ**

| Характеристика | Апаратні ГВЧ | Програмні ГВЧ | Розроблений ГВЧ |
|---|---|---|---|
| Відсутність періоду | Так | Ні | Так |
| Непередбачуваність | Так | Умовна | Так |
| Незалежність значень | Так | Умовна | Так |
| Рівень крипостійкості | Високий | Умовний | Високий |
| Швидкість генерації | Низька | Висока | Висока |
| Відтворюваність | Ні | Так | Так |
| Простота генерації | Ні | Так | Так |
| Вартість генерації | Висока | Низька | Низька |
| Необхідність апаратної реалізації | Так | Ні | Ні |

Проаналізувавши дані наведені у табл. 2 можна зробити висновок, що розроблений ГВЧ комбінує переваги апаратних та програмних ГВЧ, крім того, в ньому відсутні недоліки обох видів.

Оскільки не завжди є фінансова та фізична змога використовувати апаратні ГВЧ, а вони є більш криптографічно стійкими у порівняні з програмними, використання розробленого табличного ГВЧ надасть змогу отримувати істино випадкові та криптографічно якісні послідовності чисел без апаратної реалізації.

**Обговорення результатів та перспективи подальшого розвитку досліджень**

Отримані результати показують, що за рахунок використання розробленого табличного ГВЧ, який враховує ентропію поведінки користувача, можна підвищити стійкість криптографічних алгоритмів в умовах, коли використання апаратних ГВЧ є неможливим або недоцільним. Адже запропоноване джерело ентропії є швидкодіючим та фактично нескінченним, що надає змогу позбутися основного недоліку табличних ГВЧ – використання великого обсягу пам'яті ЕОМ.

Проте подальших досліджень потребує пошук нових шляхів підвищення криптографічної стійкості існуючих та діючих на практиці криптографічних алгоритмів та систем, враховуючи нові типи загроз та вразливостей Web-ресурсів. Зокрема, варто звернути увагу на дослідження характеристик генераторів хаосу, квантових ГВЧ, які широко використовуються у сучасному світі.

**Висновки з даного дослідження і перспективи подальших розвідок у даному напрямі**

Дослідивши пряму залежність стійкості криптографічних алгоритмів від якості ГВЧ, які в них використовуються, було розроблено та вдосконалено ГВЧ, що є важливою частиною комплексної системи захисту Web-ресурсу. Для вирішення поставленої задачі було розроблено алгоритм отримання ентропії поведінки користувача, яку використано в якості Seed даних для табличного ГВЧ та розроблено повний алгоритм роботи підсистеми генерації випадкових чисел. Також було реалізовано пакет програм криптографічної підсистеми захисту інформації, зокрема здійснено програмну реалізацію серверної частини алгоритму табличного ГВЧ для платформи ASP.NET Core, та клієнтської для Angular2. За допомогою пакету NIST здійснено статистичне тестування, результати якого підтвердили випадкову природу послідовності згенерованої розробленим табличним ГВЧ. Крім того, було продемонстровано той факт, що досліджуваний ГВЧ не має класичних недоліків псевдовипадкових ГВЧ. Таким чином, використання розробленого табличного ГВЧ в криптографічних алгоритмах підвищує їх криптографічну стійкість, що в свою чергу, впливає на рівень захищеності Web-ресурсів.

**References**

1. Korchynskyi V., Kildishev V., Riabukha O., Berdnikov O. The generating random sequences with the increased cryptographic strength, *IAPGOS*, 2020, vol. 1, pp. 20–23.

2. Truong N., Haw J., Assad S., Lam P., Kavehei O. Machine learning cryptanalysis of a quantum random number generator, *IEEE Trans Inf Forensics Secur,* 2018, vol. 14(2), pp. 403 – 414.

3. Turan M. S., Barker E, Kelsey J, McKay K.A., Baish M. L., Boyle M. Recommendation for the Entropy Sources Used for Random Bit Generation, *Gaithersburg: National Institute of Standards and Technology*, 2018, p. 84.

4. Xiupin Lv, Nakun Mu, Xiaofeng Liao. A pseudo-random number generator based on delay coupled map lattice, *Proceedings of the 2018 Eighth International Conference on Information Science and Technology (ICIST). IEEE,* 2018, pp. 377-381.

5. Saber M., Eid M. M. Low power pseudo-random number generator based on lemniscate chaotic map, *Int J Electr Comput Eng*., 2021, vol. 11(1), pp. 863-871.

6. Tutueva A., Pesterev D., Karimov A., Butusov D., Ostrovskii V. Adaptive Chirikov map for pseudo-random number generation in chaos-based stream encryption. *Proceedings of the 2019 25th Conference of Open Innovations Association (FRUCT). IEEE,* 2019, pp. 333-338.

7. Liu J., Liang Z., Luo Y., et al. A hardware pseudo-random number generator using stochastic computing and logistic map. *Micromachines*, 2021, vol. 12(1), – p. 31.

8. Hemdan A. M., Faragallah O. S., Elshakankiry O., Elmhalaway A. A fast hybrid image cryptosystem based on random generator and modified logistic map. *Multimed Tools Appl,* 2019, vol. 78(12), pp. 16177-16193.