DOI 10.31891/2307-5732-2024-337-3-6
УДК 004.4

ГАНЖЕЛО ДМИТРО
Чернівецький національний університет ім. Ю. Федьковича
https://orcid.org/0000-0002-0836-4568
e-mail: hanzhelo.dmytro@chnu.edu.ua
ПРОХОРОВ ГЕОРГІЙ
Чернівецький національний університет ім. Ю. Федьковича
https://orcid.org/0000-0001-7810-2785
e-mail: g.prokhorov@chnu.edu.ua

ДОСЛІДЖЕННЯ СТАТИСТИЧНИХ ХАРАКТЕРИСТИК ЧИСЛОВОЇ ВИПАДКОВОЇ ПОСЛІДОВНОСТІ, ОДЕРЖАНОЇ З КАДРА ВЕБ-КАМЕРИ

В роботі наведено результати досліджень статистичної характеристики числової випадкової послідовності чисел, що була одержана з одного кадру веб-камери, на відповідність однієї з вимог криптозахисту інформації: рівномірність розподілення по значенню. В результаті виявлено, що навіть в повній темряві стохастичні теплові процеси, що мають місце у матриці веб-камери, обумовлюють нерівномірний проте хаотичний розподіл по значенню чисел, що були згенеровані Проведено порівняння з аналогічною характеристикою псевдовипадковою послідовністю, що була згенерована програмним методом. Продемонстровано, що підбором кадру для зйомки цілком можливо одержати необхідний рівень рівномірного розподілу елементів послідовності по значенню. Результати дослідження можуть бути використані при проектуванні апаратного генератора послідовності випадкових чисел.

Ключові слова: програмна інженерія, теорія хаосу, криптостійкість, генератор послідовності випадкових чисел, веб-камера.

HANZHELO DMYTRO, PROKHOROV HEORHII
Chernivtsi National University

INVESTIGATION OF STATISTICAL CHARACTERISTICS OF NUMERICAL RANDOM SEQUENCE OBTAINED FROM A WEB CAMERA FRAME

At the present stage, the generation of random numbers programmatically (pseudo-random sequences) are to a certain extent predictable and susceptible to hacking. Hardware generation in modern conditions is either low-speed or expensive.

The work proposes to use a regular web camera as a source of stochastic chaos. In this case, from a frame of VGA resolution (800 × 600) it is possible to obtain a sequence with a length of 1,440,000 numbers.

The paper presents the results of a study of the statistical characteristics of a sequence of numbers obtained from one frame of a regular webcam, in terms of compliance with one of the requirements of cryptographic information protection: uniform distribution of values. As a result, it was found that even in complete darkness, stochastic thermal processes occurring in the webcam matrix cause an uneven but chaotic distribution of the values of the generated numbers.

It was experimentally established that an absolutely white illuminated surface provides a normal (Gaussian) distribution, while more than 30% of the values are missing, which does not meet the requirements of cryptographic protection and cannot be used for generation. A comparison is made with a similar characteristic of a pseudorandom sequence generated by a software method.

It is shown that by choosing a frame for shooting, it is quite possible to obtain the required level of uniform distribution of sequence elements by value. Determining the level of uniformity of the distribution is carried out quickly using the statistical library of the Java programming language and can be implemented on a regular smartphone, the Android operating system, without the use of cumbersome statistical packages. The results of the study can be used in the design of a hardware random number sequence generator.

Keywords: software engineering, chaos, crypto-resistant, random number sequence generator, webcam.

Постановка проблеми

На сучасному етапі використання інформаційних технологій проблеми безпеки національних інтересів України у цифровому просторі з точки зору кіберзахисту набувають визначального значення [1].

Генератори послідовності випадкових чисел (ПВЧ) є одними з критичних складових компонентів криптосистем. При використанні даних, сформованих із ПВЧ, можуть досягатися необхідні рівні надійності криптографічного захисту інформації (КЗІ). ПВЧ використовуються для генерації криптографічних ключів, при встановленні захищених з'єднань у мережах, для балансування навантаження, контролю цілісності даних, створення PIN-кодів і ще для багатьох застосувань [2].

ПВЧ, що згенеровані обчислювальними засобами називають псевдовипадковими, оскільки вони мають в основі наперед відомий алгоритм. Апаратні генератори випадкових чисел формують випадкові числа з хаосу справжніх імовірнісних процесів, наприклад, на основі використання теплових процесів резисторів та діодів. Для генерації дійсно криптостійких ПВЧ необхідні джерела хаосу, які забезпечують необхідний рівень непередбачуваності значень послідовності у великих діапазонах.

Таким чином, з точки зору програмної інженерії є актуальними дослідження, які присвячені вивченню нових підходів до генерації випадкових послідовностей саме на апаратному рівні. Одним із самих поширених і доступних периферійних пристроїв (апаратів) комп'ютера є веб-камера. Її властивість генерувати зображення зі швидкістю 25 кадрів/сек і роздільна здатність одного кадру в середньому 800*600 пікселів — це вагомий аргумент дослідити веб-камеру як джерело генерації ПВЧ, а статистичні

характеристики згенерованих послідовностей вимагають додаткових досліджень.

Аналіз останніх джерел

Для генерації випадкових чисел використовують три підходи.

Перший підхід — програмний — ґрунтується на спеціалізованих математичних алгоритмах програмної інженерії. На жаль програмні генератори до певної міри передбачувані. В роботі [3] наведено математичні докази незадовільної криптостійкості псевдовипадкових послідовностей. Алгоритм генерації псевдовипадкової послідовності знаходиться у відкритому доступі, наприклад, для мови Java [4], що робить теоретично можливим атакувати алгоритм шифрування. А у роботі [5] автор детально розбирає технологію зламу, хоча із залученням великих обчислювальних потужностей. Проте з розвитком обчислювальних потужностей, наприклад, квантовий комп'ютінг [6], де висока швидкість обчислень схиляє можливість атаки у практичну площину. У грудні 2022 року з'явилась публікація групи китайських вчених, яка продемонструвала можливість злому довгих RSA-ключів за допомогою сучасних квантових комп'ютерів. У роботі [7] розказано про перший в історії злом 48-бітного ключа.

Таким чином, можна сказати, програмний спосіб генерації випадкових не є повністю криптостійким.

Другий підхід — апаратний — пов'язаний з створенням та застосуванням спеціальних пристроїв, які використовують будь-які фізичних джерел шуму. Так у роботі [8] для генерації випадкових значень використовується лічильник бета-випромінювання, що робить відповідні дослідження залежними від додаткового обладнання. Такий підхід, хоча і є повністю криптостійким, проте вимагає додаткове дороге та екзотичне обладнання.

Враховуючи це, найчастіше більш актуальним є третій підхід, пов'язаний із використанням подій від стандартних пристроїв комп'ютера. Найбільш поширеним методом генерації випадкових чисел, що використовують цей підхід, є генерація випадкових чисел з використанням лічильника тактів процесора. Проте у роботі [9] було досліджено чутливість фазового шуму генераторів частоти до зовнішнього впливу, а значить, можливість впливати на генератор випадкових чисел ззовні.

У роботі [10] запропонований спосіб генерації за допомогою оптичного маніпулятора «миша», що дозволяє отримувати нерівномірно розподілені випадкові числа. Недоліком цього методу є те, що швидкість генерації випадкових чисел складає не більше 1 Кбіт/с, що не дозволяє створити на його основі високошвидкісну систему шифрування.

Систематизація вищенаведених недоліків існуючих підходів дозволяє сформулювати загальну проблему відсутності апаратного криптостійкого доступного швидкісного генератора ПВЧ :

Вище приведені недоліки приводять до висновку про необхідність проведення дослідження можливості використання зображення простої доступної веб-камери як основу генератора ПВЧ.

Слід зазначити, що подібна проблема вже розглядалась у роботі [11], проте на той час (2014 рік) теоретична можлива швидкість наближалась до 200 Мбіт/сек, а максимальна роздільна здатність веб-камери не перевищувала VGA. Але на сучасному етапі рекомендовано мати швидкодію в 1 Гбіт/сек.

У роботі [12] було розглянуто такі характеристики ПВЧ, що одержані з кадру веб-камери, як швидкодія та рівномірність розподілу по об'єму. Проте не визначеною лишилась характеристика розподілу елементів по значенню, що є однією із чотирьох критичних вимог до ПВЧ з боку КЗІ.

Метою роботи є вивчення можливості використання кадрів веб-камери для генерації послідовностей випадкових чисел. А саме оцінка розподілу по значенню чисел послідовності, що одержані в результаті екстракції значень пікселів матриці. Це дасть можливість стверджувати про відповідність вимогам КЗІ апаратного генератора випадкових послідовностей без залучення додаткового обладнання і коштів.

Задачі дослідження полягають у наступному:

- дослідити статистично розподіл елементів ПВЧ по значенню;
- дослідити аналогічну характеристику псевдовипадкової еталонної послідовності, що одержана програмним методом;
- порівняти дві характеристики та зробити рекомендації;

Генератор випадкових чисел, реалізований у цій роботі, розроблявся як частина криптографічної системи захисту інформації на основі числових випадкових послідовностей.

Виклад основного матеріалу

Обладнання дослідження:

- десктоп комп'ютер ASUS Z97K; CPU Intel® Core™ i3-4170 CPU @ 3.70GHz × 4; 32 Gb RAM; SDD Kingston 240 Gb;
- Web Digital Camera FULL HD 1080P, TrueColor. QQVGA (176×144); QVGA (320 × 240); VGA (640× 480); SVGA (800 × 600). У обраній вебкамері верхня межа роздільної здатності згідно специфікації становить 800×600 пікселів (VGA), а дефолтним – режим 176 × 144 (QQVGA), the Quarter-QVGA resolution. При бажані цей розмір можна розширити до WebcamResolution.HXGA (4096x3072) – все залежить від роздільної здатності обраної камери [21];
- програмне забезпечення:: ОС Ubuntu 22 LTS, 64 bit; Java Amazon Corretto 17.0.5; IntelliJ IDEA 2023.3.4 (Ultimate Edition); пакет com.github.sarxos.webcam версія 0.3.12 – захоплення кадру; пакет

javax.imageio – обробка відео зображення; пакет java.security.SecureRandom – програмна генерація випадкової послідовності.

Метод дослідження

Обчислювальні потужності, швидкодія та гнучкі методи класу Webcam та BufferedImage мови програмування Java дозволили швидко і оптимально вилучити з об'єкта кадра веб-камери одразу ж саму послідовність чисел без складних матричних перетворень. Сама послідовність була розміщена у просту структуру даних типу Array і готова для подальшого дослідження.

Базовий програмний код програми:

```
Webcam webcam = Webcam.getWebcams().get(0);
Dimension dimension = WebcamResolution.VGA.getSize();
webcam.setViewSize(dimension);
webcam.open();
```

Спочатку ініціалізується веб-камера системи. При такому налаштуванні можна вибрати і саму камеру (якщо їх декілька) і режим захоплення кадру (роздільну здатність знімка)

Наступний код витягає з камери об'єкт моментального знімку, перетворює зображення у потік байтів формату TIFF і зберігає його у одномірному масиві байтів.

```
BufferedImage image = webcam.getImage();
ByteArrayOutputStream stream=new ByteArrayOutputStream()
ImageIO.write(image, "tiff", stream);
byte[] bytes = stream.toByteArray();
```

Згідно досліджень у роботі [11] вилучений масив байтів є по суті послідовністю випадкових чисел і готовий для подальших досліджень статистичних характеристик.

Дослідження послідовності на рівномірність розподілу чисел по значенню

Одна із вимог до послідовності випадкових чисел — це рівномірність розподілу по значенню. Значення всіх чисел типу byte знаходиться у проміжку [-128... 127]. Тож вимога полягає у тому, що всі значення однаково мати бути присутні у згенерованій послідовності. Всього чисел 256, тоді частка кожного — 0.00390625 або приблизно 0.4%.

Завдання зводиться до того, що треба підрахувати кількість кожного числа у послідовності і вирахувати його частку. Вивести ці величини зовсім не складно, це просте групування величина-кількість. Мовою **Java** це робиться елементарно:

```
val map = bytes.stream()
        .collect(Collectors.groupingBy(Function.identity(),                    Collectors.counting()));
```

В результаті маємо стандартну структуру даних **Map<key, value>**, де **key** – це значення кожного числа (елемента) у проміжку [-128 … 127], а от **value** буде кількість цього числа (елемента) у послідовності. Ця кількість ділиться на кількість всіх чисел у послідовності і одержуємо частку цього значення або ж перерахунок у проценти.

Щоб вирахувати статистичні данні розподілу трансформуємо цю структуру даних у простий список часток значень у процентах без різниці , якому числу ця частка належить. Одержані значення завантажуються у об'єкт класу **DescriptiveStatistics** мови програмування Java, який надалі вираховує всю необхідно статистику. Приклад на Java:

```
DescriptiveStatistics stats = new DescriptiveStatistics();
map.values().stream().forEach(el -> stats.addValue(el));
```

Результати порівняння були занесені у нижченаведену табл. 1.

Статистика пояснюється на прикладі режиму захоплення кадру QQVGA при освітленні 100 люкс (офісне приміщення).

Мінімальна частка у розподілі по величині значень — **min**: 0.05, це означає, що якийсь елемент з числового проміжку [-128..+127] у вилученій послідовності зустрічається у 0.05% випадків, а саме 38 раз із 76032 можливих. І це мінімальне значення.

Максимальне значення **max**: 1.41 — якесь число зустрічається у 1.41% випадків у послідовності (не важливо , яке саме число), а саме 1072 рази.

Середнє арифметичне (**mean**: 0.39) воно завжди стандартне для любих умов, бо 1/256 = 0.00390625, або 0.39%. Це відповідає значенню 297 зустрічей для кожного числа — ідеал.

Медіана **median**: 0.37 разом із середньо квадратичним відхиленням (**std dev**: 0.22) визначає характерний проміжок 0.37 +/- 0.22 (або [0.15 .. 0.59]), що покриває 67% значень розподілу.

Величина **absent**: 0.0 означає, що всі числа ряда [-128..+127] присутні у згенерованій послідовності.

Таблиця 1

Залежність рівномірного розподілу по значенню від режиму захоплення веб-камери (роздільної здатності кадру) для різних значень освітлення

Роздільна здатність освітлення	**QQVGA(176 * 144)**	**QVGA(320*240)**	**VGA (640*480)**
Темрява, 10^{-4} люкс	min: 0.0 max: 39.73 mean: 0.39 std dev: 2.59 median: 0.0 absent: 59.4	min: 0.0 max: 38.71 mean: 0.39 std dev: 2.52 median: 0.00005 absent: 46.8	min: 0.0 max: 35.42 mean: 0.39 std dev: 2.53 median: 0.00001 absent: 40.5
Світла біла поверхня, 100 люкс,	min: 0.0 max: 2.16 mean: 0.39 std dev: 0.61 median: 0.01 absent: 32.0	min: 0.0 max: 2.70 mean: 0.39 std dev: 0.73 median: 9.79 absent: 35.1	min: 0.0 max: 3.26 mean: 0.39 std dev: 0.75 median: 0.00 absent: 27.3
Офіс, 100 люкс	min: 0.05 max: 1.41 mean: 0.39 std dev: 0.22 median: 0.37 absent: 0.0	min: 0.01 max: 1.25 mean: 0.39 std dev: 0.24 median: 0.44 absent: 0.0	min: 0.03 max: 1.02 mean: 0.39 std dev: 0.21 median: 0.45 absent: 0.0

Дослідження рівномірності розподілу у програмно згенерованій послідовності

Для дослідження була згенерована ПВЧ аналогічної довжини використовуючи стандартну бібліотеку Java SecureRandom

```
SecureRandom secureRandom = new SecureRandom();
byte[] randoms = new byte[size];
secureRandom.nextBytes(randoms);
```

Об’єкт DescriptiveStatistics обрахував статистику аналогічно до попереднього експерименту:

min: 0.377, **max**: 0.401, **mean**: 0.396, **std dev:** 0.004, **median**: 0.390, **absent**: 0.0

Порівнюючи ці результати з аналогічними, що одержані з веб-камери, можна сказати, що для абсолютної темряви і однорідної білої поверхні розподілення носить чітко виражений нормальний (гауссівський) характер. А програмно згенерована послідовність у великій степені подібна до рівномірного розподілу.

Порівняння на розподілення по простору і рівень хаосу у програмно згенерованої послідовності у контексті нашого експерименту сенсу не мають. В цьому випадку нема чітко вираженого періоду (кадру), то ж дослідження на довжину період повторення мають проводитись іншим методом, наприклад, тести NIST.

Обговорення результатів дослідження розподілу елементів ПВЧ по значенню

В ідеалі рівномірного розподілу частка (процент) присутності у послідовності для кожного елемента має бути однаковий.

Ідеальні статистичні характеристики для рівномірного розподілу мали би виглядати так:

min: 0.39, **max**: 0.39, **mean**: 0.39, **std dev**: 0.0, **median**: 0.39, **absent**: 0.0

Це значить, що всі числа ПВЧ мають однаковий процент присутності (0.39%), відсутніх нема, відхилень нема — ідеально рівномірний розподіл. Аналізуючи одержану статистику для різних режимів приходимо до висновку, що розподілення по значенню :

- для темряви 10^{-4} люкс — хаотичне, 50% чисел взагалі відсутні;
- для білої рівномірно освітленої (200 люкс) поверхні — ближче до нормального, але з елементами хаосу, 10% чисел відсутні;
- для звичайного офісного зображення і освітлення 150-200 люкс — наближається до рівномірного, але з елементами хаосу.

Діаграма розподілення елементів по значенню величини для кадру офісного приміщення при освітленні 150-200 люкс зображена на рис. 1.

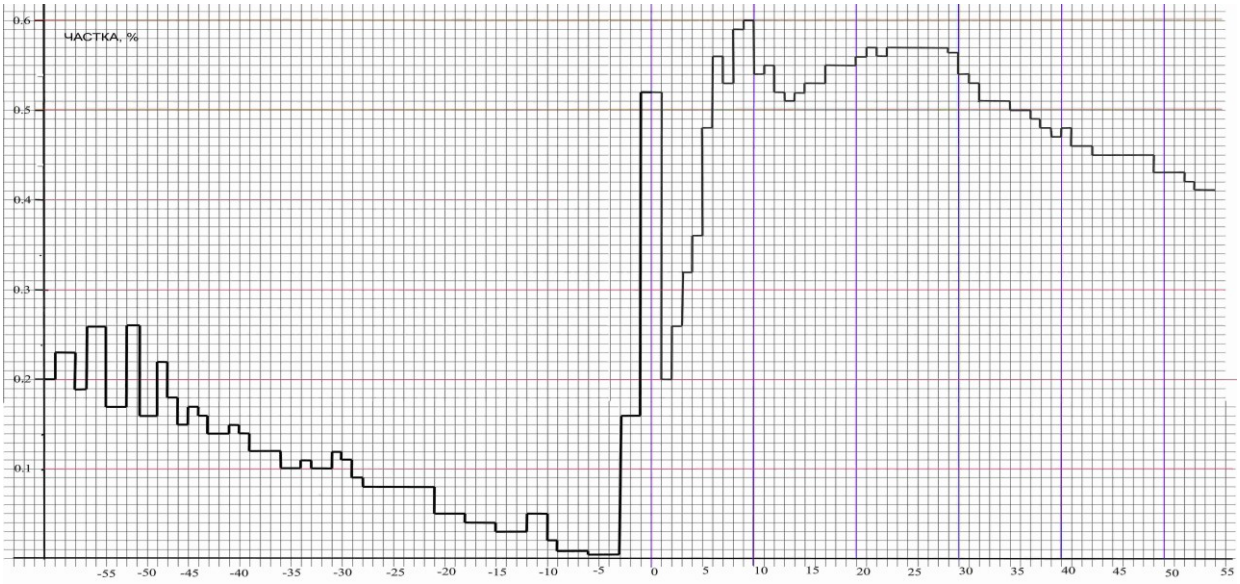

Рис. 1. Фрагмент діаграми розподілення елементів по значенню у процентах. Офіс, 150-200 люкс

На рис. 1 зображений фрагмент ([-55, +55]) розподілу значень ПВЧ. Весь діапазон, як зазначалось, [-128, +127], такий довгий ряд важко привести весь у зручному вигляді.

Пояснити його можна наступним чином. Число 0 зустрічається у послідовності у 0.52% випадків, число 10 — 0.61%, число -10 — 0.02%.

Згідно обрахованої статистики медіана становить 0.37, середньо-квадратичне відхилення – 0.22. Це значить, що у діапазоні [0.15 .. 0.59] знаходиться 68% всіх можливих чисел. При рівномірному розподілі середньо-квадратичне відхилення прямує до нуля. То ж приведена характеристика характеризує розподіл як не рівномірний, а як хаотичний. Проте у порівнянні з граничними експериментами (повна темрява) можна побачити, що відхилення у порівнянні не таке вже і велике, і, що найважливіше, піддається коригуванню.

Аналогічно до апаратного метода програмні ПВЧ були оброблені за допомогою функціонала Java, що дало змогу об'єктивно порівняти аналогічні статистичні характеристики. Якість програмного метода теж не ідеальна, проте все ж таки знаходиться у межах похибки 1.0%. Без додаткового опрацювання було помітно переваги програмного метода у питанні рівномірності розподілу по значенню елементів. Проте спеціальним підбором кадра і освітлення вдалось наблизитись до якості програмного метода.

Слід окремо зазначити, що запропонований метод Java обробки згенерованої послідовності дає миттєву характеристику розподілу значень на відміну від [11], де для цього застосовується громіздке програмне забезпечення. Це дає змогу використовувати звичайний смартфон для вибору необхідного кадру.

Висновки

1. Зафіксовано незадовільний рівень однорідності послідовності по значенню на граничних умовах: у темряві (10^{-4} люкса) — хаотичний і гомогенному освітленні 200 люкс - нормальний.

2. Експериментальним шляхом виявлено у ПВЧ, що згенерованих програмним шляхом, розбіжність з ідеальним зразком становить на рівні 1.2%, що дає змогу зафіксувати рівень граничного достатнього значення розподілу елементів по значенню.

3.Порівняння статистичних характеристик ПВЧ, згенерованих апаратним шляхом, із аналогічними , згенерованими програмним методом, висвітило, що підбором кадру та коригування освітленості можливо наблизити цей розподіл до задовільного рівня вимог КЗІ.

Загальний висновок: послідовності випадкових чисел, що згенеровані за допомогою веб-камери можуть слугувати основою для апаратного генератора ПВЧ.

Література

1. Про рішення Ради національної безпеки і оборони України від 30 грудня 2021 року "Про План реалізації Стратегії кібербезпеки України: указ Президента України № 37/2022. URL: https://www.president.gov.ua/documents/372022-41289

2. Asia Othman Aljahdal, "Random Number Generators Survey" International Journal of Computer Science and Information Security (IJCSIS),Vol. 18, No. 10, October 2020. https://zenodo.org/records/4249407

3. Martinez, F. (2022). Attacks on Pseudo Random Number Generators Hiding a Linear Structure. In: Galbraith, S.D. (eds) Topics in Cryptology – CT-RSA 2022. CT-RSA 2022. Lecture Notes in Computer Science, vol. 13161. Springer, Cham. https://doi.org/10.1007/978-3-030-95312-6_7

4. Class SecureRandom. All Implemented Interfaces. URL: https://docs.oracle.com/javase/8/docs/api/java/security/SecureRandom.html

5. M. Cornejo, S. Ruhault, "(In)Security of Java SecureRandom Implementations", Journées Codage et Cryptographie, 2014. https://www-fourier.ujf-grenoble.fr/JC2/exposes/ruhault.pdf

6. Остапов С.Е., Добровольський Ю.Г. Квантова інформатика та квантові обчислення. Чернівці: ЧНУ, 2021. 99 с. https://archer.chnu.edu.ua/jspui/bitstream/123456789/2830/1/QuantumComputing.pdf

7. Bao Yan, Ziqi Tan, Shijie Wei, Haocong Jiang, Weilong Wang, Hong Wang, et al. Factoring integers with sublinear resources on a superconducting quantum processor. arXiv:2212.12372v1 [quant-ph] 23 Dec 2022 https://arxiv.org/pdf/2212.12372.pdf

8. Seongmo Park, Byoung Gun Choi, Taewook Kang, Kyunghwan Park, Youngsu Kwon, Jongbum Kim, "Efficient hardware implementation and analysis of true random-number generator based on beta source." ETRI Volume 42, Issue4 ,Special Issue on SoC and AI processors, August 2020, Pages 518-526, https://onlinelibrary.wiley.com/doi/full/10.4218/etrij.2020-0083

9. Agata KAŹMIERCZYK, Andrzej Ł. CHOJNACKI, Kornelia BANASIK (2022). Pseudorandom number generators as applied in reliability analysis. Kielce University of Technology, Faculty of Electrical Engineering, Automatic Control and Computer Science, Department of Power Engineering, Power Electronics and Electrical Machines, doi:10.15199/48.2022.12.44. http://pe.org.pl/articles/2022/12/44.pdf

10. Ostapov S., Diakonenko B., Fylypiuk M., Hazdiuk K., Shumyliak L. and Tarnovetska O. 2023. Symmetrical Cryptosystems based on Cellular Automata. International Journal of Computing. 22, 1 (Mar. 2023), 15-20. https://doi.org/10.47839/ijc.22.1.2874.

11. Li R. A True Random Number Generator algorithm from digital camera image noise for varying lighting conditions. SoutheastCon 2015, Fort Lauderdale, FL, USA, 2015, pp. 1-8, doi: 10.1109/SECON.2015.7132901. https://ieeexplore.ieee.org/document/7132901

12. Ганжело Д., Прохоров Г. ДОСЛІДЖЕННЯ ЧИСЛОВОЇ ВИПАДКОВОЇ ПОСЛІДОВНОСТІ, ЩО ОДЕРЖАНА З ВЕБ КАМЕРИ. (2024). Herald of Khmelnytskyi National University. Technical Sciences, 333(2), 120-124. https://doi.org/10.31891/2307-5732-2024-333-2-18

References

1. Pro rishennia Rady natsionalnoi bezpeky i oborony Ukrainy vid 30 hrudnia 2021 roku "Pro Plan realizatsii Stratehii kiberbezpeky Ukrainy : ukaz Prezydenta Ukrainy № 37/2022. URL: https://www.president.gov.ua/documents/372022-41289

2. Asia Othman Aljahdal, "Random Number Generators Survey" International Journal of Computer Science and Information Security (IJCSIS),Vol. 18, No. 10, October 2020. https://zenodo.org/records/4249407

3. Martinez, F. (2022). Attacks on Pseudo Random Number Generators Hiding a Linear Structure. In: Galbraith, S.D. (eds) Topics in Cryptology – CT-RSA 2022. CT-RSA 2022. Lecture Notes in Computer Science, vol. 13161. Springer, Cham. https://doi.org/10.1007/978-3-030-95312-6_7

4. Class SecureRandom. All Implemented Interfaces. URL: https://docs.oracle.com/javase/8/docs/api/java/security/SecureRandom.html

5. M. Cornejo, S. Ruhault, "(In)Security of Java SecureRandom Implementations", Journées Codage et Cryptographie, 2014. https://www-fourier.ujf-grenoble.fr/JC2/exposes/ruhault.pdf

6. Ostapov S.E., Dobrovolskyi Yu.H. Kvantova informatyka ta kvantovi obchyslennia. Chernivtsi: ChNU, 2021. 99 s. https://archer.chnu.edu.ua/jspui/bitstream/123456789/2830/1/QuantumComputing.pdf

7. Bao Yan, Ziqi Tan, Shijie Wei, Haocong Jiang, Weilong Wang, Hong Wang, et al. Factoring integers with sublinear resources on a superconducting quantum processor. arXiv:2212.12372v1 [quant-ph] 23 Dec 2022 https://arxiv.org/pdf/2212.12372.pdf

8. Seongmo Park, Byoung Gun Choi, Taewook Kang, Kyunghwan Park, Youngsu Kwon, Jongbum Kim, "Efficient hardware implementation and analysis of true random-number generator based on beta source." ETRI Volume 42, Issue4 ,Special Issue on SoC and AI processors, August 2020, Pages 518-526, https://onlinelibrary.wiley.com/doi/full/10.4218/etrij.2020-0083

9. Agata KAŹMIERCZYK, Andrzej Ł. CHOJNACKI, Kornelia BANASIK (2022). Pseudorandom number generators as applied in reliability analysis. Kielce University of Technology, Faculty of Electrical Engineering, Automatic Control and Computer Science, Department of Power Engineering, Power Electronics and Electrical Machines, doi:10.15199/48.2022.12.44. http://pe.org.pl/articles/2022/12/44.pdf

10. Ostapov S., Diakonenko B., Fylypiuk M., Hazdiuk K., Shumyliak L. and Tarnovetska O. 2023. Symmetrical Cryptosystems based on Cellular Automata. International Journal of Computing. 22, 1 (Mar. 2023), 15-20. https://doi.org/10.47839/ijc.22.1.2874.

11. Li R. A True Random Number Generator algorithm from digital camera image noise for varying lighting conditions. SoutheastCon 2015, Fort Lauderdale, FL, USA, 2015, pp. 1-8, doi: 10.1109/SECON.2015.7132901. https://ieeexplore.ieee.org/document/7132901

12. Hanzhelo D., Prokhorov H. DOSLIDZhENNIa ChYSLOVOI VYPADKOVOI POSLIDOVNOSTI, ShchO ODERZhANA Z VEB KAMERY. (2024). Herald of Khmelnytskyi National University. Technical Sciences, 333(2), 120-124. https://doi.org/10.31891/2307-5732-2024-333-2-18