DOI 10.31891/2307-5732-2025-347-42
УДК 004.4

ПРОХОРОВ ГЕОРГІЙ
Чернівецький національний університет ім. Ю. Федьковича
https://orcid.org/0000-0001-7810-2785
e-mail: g.prokhorov@chnu.edu.ua
ТРЕМБАЧ ДЕНИС
Чернівецький національний університет ім. Ю. Федьковича
https://orcid.org/0000-0001-8095-4186
e-mail: trembach.denys@chnu.edu.ua

ПІДВИЩЕННЯ ЯКОСТІ СТАТИСТИЧНИХ ХАРАКТЕРИСТИК ЧИСЛОВОЇ ВИПАДКОВОЇ ПОСЛІДОВНОСТІ

В роботі наведено результати досліджень обробки числової випадкової послідовності чисел, що була одержана з одного кадру вебкамери, на відповідність однієї з вимог криптозахисту інформації: рівномірність розподілення. Розроблено високошвидкісний функціонал обробки послідовності на основі клітинних автоматів. В результаті виявлено, що обробка функціоналом лінійних клітинних автоматів хаотичних правил (правила 30, 90, 105, 150) підвищує якість розподілу елементів послідовності випадкових чисел майже до ідеальної. Проведено вимірювання швидкодії кожного правила, виявлено, що правило 30 демонструє найвищу швидкість обробки, проте не набагато відрізняється від інших правил. Виявлено, що підбором кількості ітерацій можливо одержати необхідний рівень рівномірності розподілу елементів послідовності. Результати дослідження можуть лягти в основу проектуванні доступного високошвидкісного надійного апаратно-програмного генератора послідовності випадкових чисел.

Ключові слова: програмна інженерія, вебкамера, генератор випадкових чисел, клітинні автомати.

PROKHOROV HEORHII, TREMBACH DENIS
Chernivtsi National University

IMPROVEMENT OF STATISTICAL CHARACTERISTICS OF NUMERICAL RANDOM SEQUENCE

At the present stage, a program generation of random numbers is at risk of being hacked due to the increasing computing power of modern systems. Hardware generation is based on reliable stochastic physical phenomena but provides low productivity or bad statistical parameters. The work proposes to use a linear cellular automata processing for improvement of some statistical parameters of random number sequences, generated by web camera.

The investigation presents the results of improvement of the statistical characteristics of a sequence of numbers obtained from an ordinary webcam, in terms of compliance with one of the requirements of crypto resistance: uniform distribution of elements by value. Previously it was found that stochastic processes occurring in the webcam matrix cause a chaotic distribution of the values in a generated sequence of random numbers. This obstacle can be overcome utilizing the processing power of linear cellular automata, especially rules 30, 90, 105. 150. These cryptoprimitives are known as chaotic ones that consume low computing power.

The success of the application was assessed in comparison with the generation of random numbers by a software method, in particular a class SecureRandom of Java programming language. It is shown that by choosing several iterations, it is possible to obtain the required level of uniformity of distribution of sequence elements by value.

Estimation the level of uniformity of a 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, cellular automata, random number generator, webcam.

Постановка проблеми

Генерація випадкових чисел має важливу роль у забезпеченні криптостійкості, наприклад, генерація ключів, пін-коду, паролів. До такої генерації стандартно висувається ряд вимог затверджених протоколами NIST та BSI. Але при генерації саме послідовності випадкових чисел (ПВЧ) ці вимоги виходять на новий рівень.

На вістрі сучасних вимог стоїть актуальна задача програмної інженерії — генерація ПВЧ з продуктивністю не менше 100 Мбіт/сек, а бажано довести швидкість до 1 Гбіт/сек. Така вимога лежить в основі стеганографії - створення захищених каналів передачі даних, де захищаються не тільки дані, що передаються, але і сам факт передачі [1]. У цьому випадку зашифровані дані «розчиняються» у послідовності випадкових чисел і у такому вигляді передаються споживачу. Якщо таку передачу перехопити, то буде вкрай важко вичислити, чи містить пакет інформативну складову, чи це просто випадковий контент для перенавантаження каналів перехоплення.

Щодо самих ПВЧ , що згенеровані, вони мають відповідати усім вимогам криптостійкості: рівень хаосу (випадковість), довгий період зациклювання, рівномірність розподілу елементів. Великою проблемою на цей час є саме балансування всіх цих вимог в одному пристрої — високопродуктивному генераторі ПВЧ.

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 [2] наведено

математичні докази незадовільної криптостійкості псевдовипадкових послідовностей. Алгоритм генерації псевдовипадкової послідовності знаходиться у відкритому доступі, наприклад, для мови Java [3, 4], що робить теоретично можливим атакувати алгоритм шифрування.

Таким чином, можна сказати, програмний спосіб генерації випадкових в силу його передбачуваності не є повністю криптостійким хоча і повністю задовольняє всі інші вимоги криптозахисту інформації (КЗІ).

Другий підхід — апаратний — побудований з застосуванням фізичних пристроїв, які використовують будь-які стохастичні джерела шуму. Так у роботі [5,6] для генерації випадкових чисел використовується лічильник бета-випромінювання. Такий підхід, хоча і є повністю криптостійким, проте вимагає додаткове дороге та екзотичне обладнання.

Вище приведені обмеження приводять до висновку про необхідність дослідження можливості використання простої доступної вебкамери як основу надійного високопродуктивного генератора ПВЧ. Аналогічна ідея вже розглядалась у роботі [7], проте на той час (2014 рік) теоретична можлива швидкість обмежувалась 200 Мбіт/сек, а максимальний режим вебкамери не перевищував VGA (640×480). Але сучасні вимоги рекомендують мати швидкодію як мінімум 100 Кбіт/сек, а бажано 1 Гбіт/сек.

У роботах [8, 9] було розглянуто статистичні характеристики ПВЧ, що одержана з вебкамери, і виявилось що при всіх позитивних моментах така характеристика, як рівномірний розподіл елементів по значенню, залишається не задовільною, що є не припустимим з точки зору вимог КЗІ.

Метою роботи є вивчення можливості поліпшення характеристик ПВЧ, а саме характеристику розподілу елементів.

Задачі дослідження полягають у наступному:

- дослідити характер розподілу оригінальної ПВЧ, що одержана з кадру вебкамери;
- дослідити результати обробки ПВЧ клітинним автоматом (КА) хаотичного правила (30, 90, 105, 150);
- визначити інтенсивність використання КА для поліпшення статистичних характеристик ПВЧ.

Генератор випадкових чисел, реалізований у цій роботі, розроблявся як частина криптографічної системи захисту каналу передачі інформації методом стеганографії.

Виклад основного матеріалу

Обладнання дослідження:

- Десктоп:
  - CPU: AMD Ryzen 5 5600 4.4ghz,
  - RAM: 16gb 3200mhz,
  - SSD: Kingston NV1 250gb,
  - GPU: Nvidia GeForce GTX 1660ti
- Anker Powerconf C200. QQVGA (176×144); QVGA (320 × 240); VGA (640× 480); SVGA (800 × 600); HD (1280 x 720); Full HD (1920 x 1080); Quad HD (2560 x 1440) ;
- Програмне забезпечення: OS Linux 22.4 LTS, 64 bit; Java Amazon Corretto 21.0.3; IntelliJ IDEA 2024.3; бібліотека для роботи з вебкамерою com.github.sarxos.webcam версії 0.3.12

Метод дослідження.

В основу дослідження було покладено результати попередніх робіт [8,9], де детально описано метод вилучення ПВЧ з кадру вебкамери. Було дороблено функціонал по обчисленню і візуалізації статистичних характеристик ПВЧ.

Щодо методів поліпшення статистичних характеристик ПВЧ, то було розглянуто криптопримітиви — лінійні кліткові автомати (КА) [10]. На рис.1 графічно проілюстровано основи функціоналу КА хаотичного типу — правила 30, 90, 105, 150.

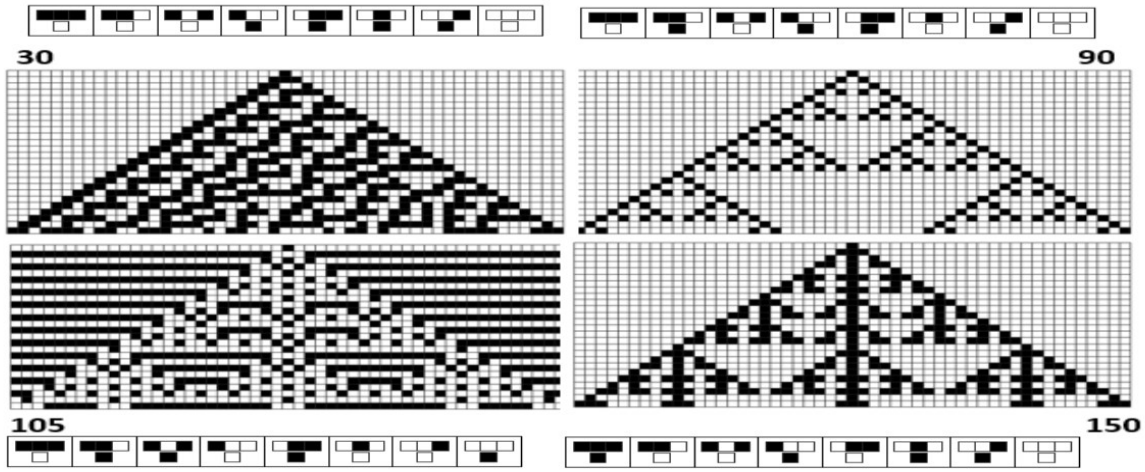

Рис. 1. Графічна ілюстрація роботи клітинних автоматів (КА) хаотичного типу з правилами 30, 90, 105, 150 на протязі 20 ітерацій

На рис.1 показано проілюстровано результати роботи КА та результати роботи на 20 ітераціях. На початкових умовах дана лише одна чорна клітина (логічний 0) в центрі вхідної горизонтальної послідовності (верхній горизонтальний ряд). Всі чотири правила продемонстрували хаос (непередбачуваність) на 20й ітерації, проте візуально помітно, що співвідношення одиниць та нулів (50:50) притаманно більше саме правилу 30. Це одна з вимог КЗІ до послідовностей випадкових чисел.

Ці правила чітко описані апаратом дискретної математики та булевої алгебри, що продемонстровано у Таблиці 1 .

Таблиця 1

Реалізація правил кліткових автоматів у термінах булевої алгебри та мови Java

Правило	**Логічний вираз**	**Java**
30	left XOR (center OR right)	p ^ (q || r)
90	left XOR right	p ^ r
105	left XOR center XOR (NOT right)	p ^ q ^ (!r)
150	left XOR center XOR right	p ^ q ^ r

В наведеній таблиці показано, як логічні правила КА легко імплементуються мовою Java у програмний код. Це бітові операції, тому виконуються миттєво і не споживають великого ресурсу.

Дослідження послідовності на рівномірність при обробці КА.

Клас SecureRandom розроблений Oracle спеціально для генерації криптостійких ПВЧ. Він гарантує відповідність всім вимогам криптостійкості - швидкість, розподілення, випадковість — крім одного: на великих послідовностях простежується його періодичність і передбачуваність.

Елементи ідеальної послідовності приймають цілі значення в діапазоні [-128 .. +127] — всього 256 значень , що відповідає типу даних **byte** мови програмування Java. То ж в ідеально згенерованій ПВЧ кожне значення має присутність рівну 1/256, або 0.385%. На практиці такий випадок малоймовірний. Якщо згенерована послідовність , наприклад, на 100 тис елементів, кожне значення мусить зустрітися у послідовності рівно 390 раз (100 000 : 256 = 390.625). Витримати на практиці таке значення — вкрай важко. То ж допускаються певні відхилення від ідеалу. Дослідимо ці практично допустимі відхилення.

Дослідимо для початку послідовність, згенеровану за допомогою класа **SecureRandom** мови програмування Java, та вияснимо яке відхилення у відсотках воно має, та як виглядає спектр розподілу, рис. 2.

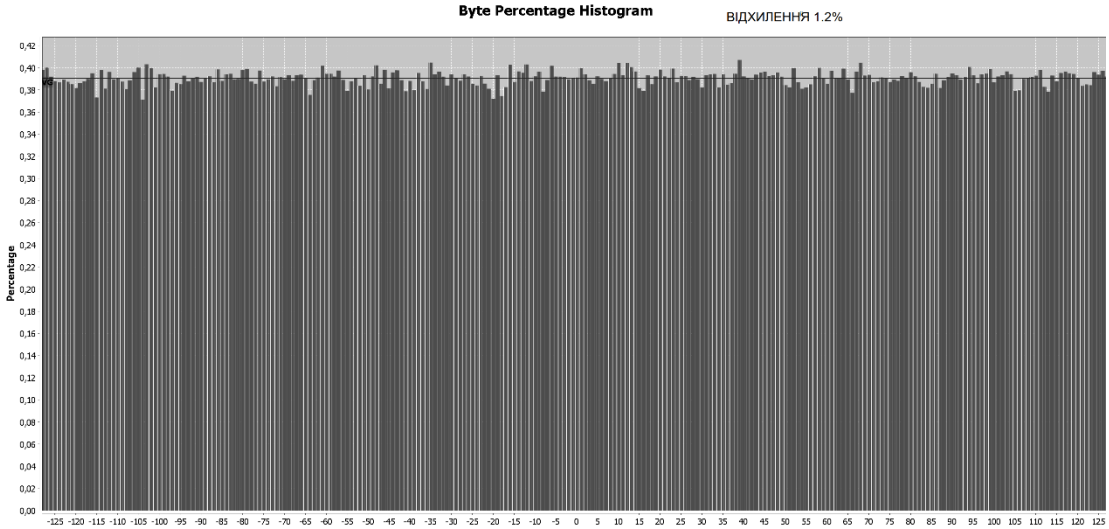

Рис. 2. Гістограма розподілення елементів по значеною ПВЧ, що згенерована класом SecureRandom

На рис.2 зображено спектр ПВЧ, що згенерована SecureRandom. По осі абсцис (горизонтальна вісь) відкладено діапазон можливих значень. Для Джави цей діапазон становить [-128 .. +127]. По вертикальній осі відкладена «частка присутності» - кількість байтів певної величини у процентах.

Це майже ідеальний розподіл, оскільки графік має вигляд прямокутника. Також бачимо, що середньо квадратичне відхилення значення від середнього (тонка лінія на рівні приблизно 0.39) складає лише 0.0005 (приблизно 1.2%), що є насправді майже ідеальним результатом або практично ідеальним.

Тепер дослідимо спектр випадкового кадру, що зображений на рис. 3.

а)

б)

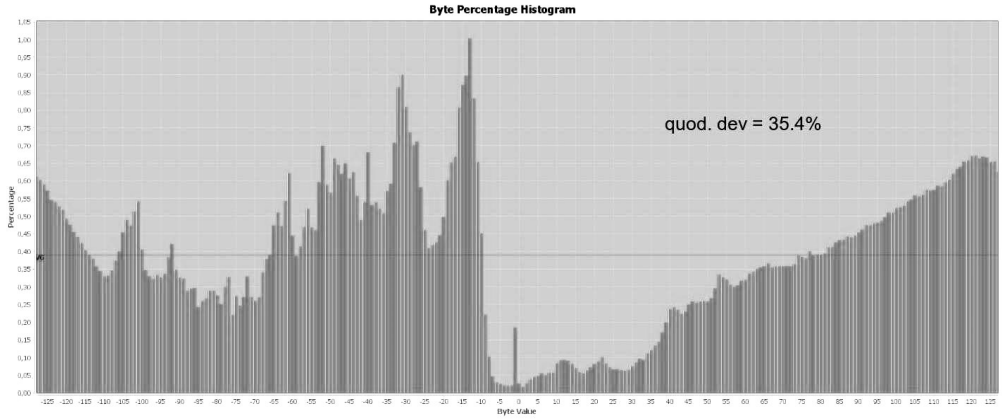

Рис. 3. Випадкове зображення: а) кадр оцифрований у форматі BMP; б) гістограма розподілення елементів по значенню (спектр)

На рис.3а представлено зображення з вікна ввечері 17 липня о 20.27, напрямок на північ. Кадр не вибирався спеціально, але бажано, щоб він містив побільше кольорової палітри [9]. Рисунок 3б містить спектр ПВЧ, що була екстрагована з цього кадру.

На гістограмі рис 3б видно, що чітко виражені локальні мінімуми у розподіленні мають елементи (байти) значення -2 та +2. Максимальну присутність продемонструвало значення -13. Такі локальні екстремуми можуть грати роль «відбитка пальця» для генератора, що не бажано , проте не категорично. Розподілення носить не рівномірний, а хаотичний характер, середньо квадратичне відхилення значення від середнього (тонка синя лінія на рівні приблизно 0.39) складає 35.4%. Це не задовольняє вимоги КЗІ, то ж згенерована ПВЧ потребує обробки. Перш за все заміряли час обробки ПВЧ кожним правилом КА, щоб вибрати найбільш швидкий метод. Зафіксували наступні результати: правило КА-30 — 135 Мбіт/сек, правило КА-90 — 147 Мбіт/сек, правило КА-105 — 147 Мбіт/сек, правило КА-150 — 154 Мбіт/сек. Виявилось, що правило 30 швидше за всіх обробляє ПВЧ, хоча і не набагато. Сама генерація ПВЧ з вебкамери у режимі SVGA (800 × 600) характеризується продуктивністю 144 Mbit/s, то ж час обробки цієї ПВЧ засобами КА приблизно такий же самий. На рис.4 подано спектр оригінального ПВЧ пісяла першого циклу обробки КА-30.

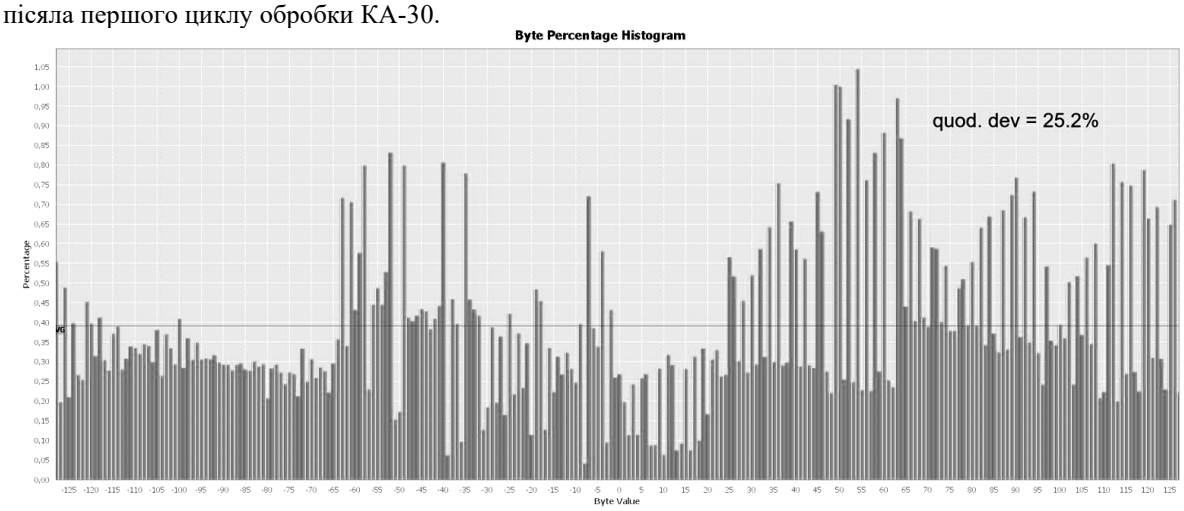

Рис. 4. Гістограма розподілення значень після обробки КА-30. 1 ітерація

На рис. 4 видно суттєве вирівнювання гістограми в сторону рівномірності. Квадратичне відхилення знизилось з 35% до 25%. Проте це все ще задовільно згідно вимог КЗІ, згідно практичному ідеалу 1.2%. Наступний рис. 5 демонструє спектр після 10ти ітерацій обробки КА-30.

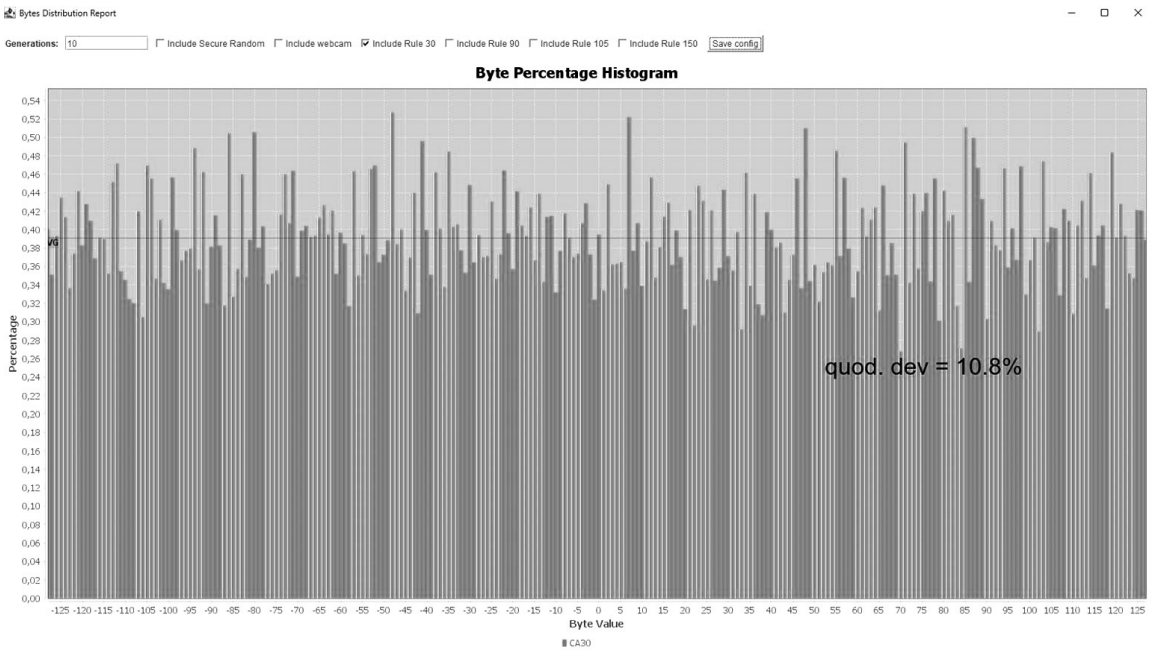

Рис. 5. Гістограма розподілення (спектр) після обробки КА-30. 10 ітерацій

На рис. 5 помітно, що хаотичність розподілу зменшується. «Відбиток пальця» повністю зник. Середньоквадратичне відхилення скоротилось до величини 10.8%. Можливо, що такого відхилення буде достатньо, щоб визнати рівномірність розподілення задовільним.

Наступний рис. 6 демонструє спектр після 20-и циклів обробки КА-30.

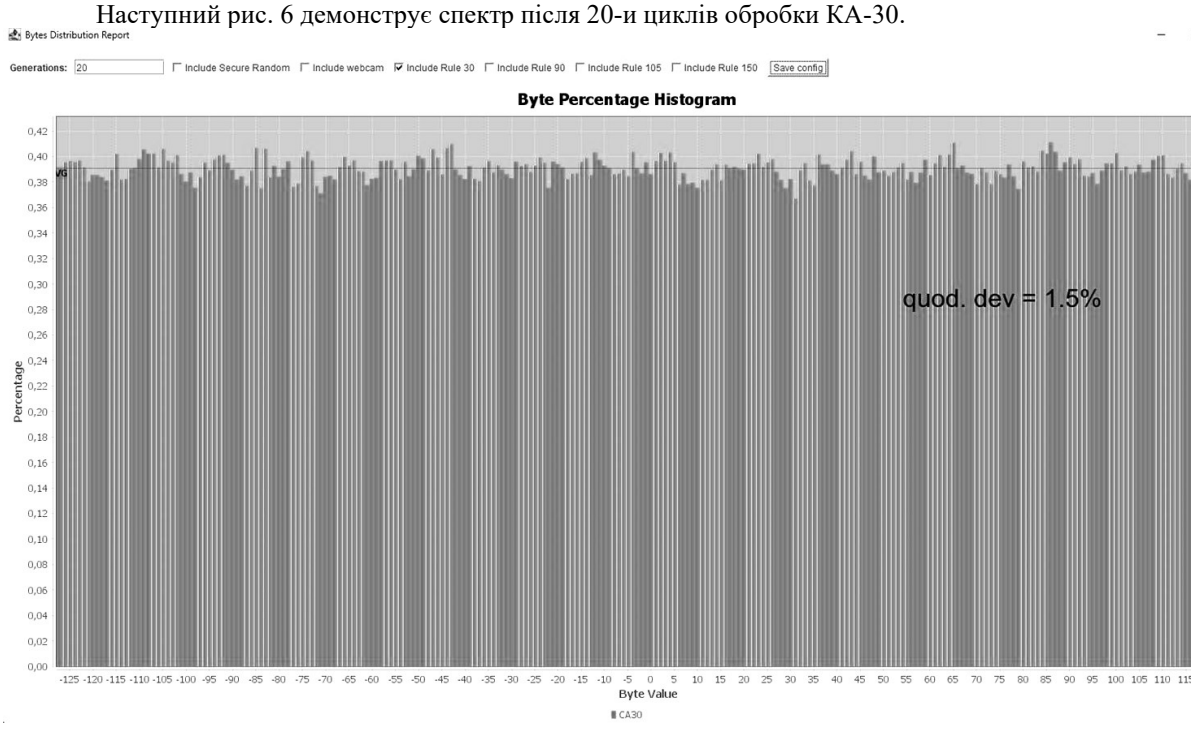

Рис. 6. Гістограма розподілення значень після обробки КА-30. 20 ітерацій

На рис. 6 чисто візуально помітно, що після 20 циклів обробки (ітерацій) спектр близький до практично ідеального, а в числовому значенні відхилення досягло рівня 1.5%, нагадаємо, що у практично ідеального спектра це відхилення дорівнює 1.2%. Після 30 ітерацій відхилення знизилось до 0.8%, проте час обробки досяг 1.5 сек, що нівелює метод генерації (швидкість) майже на два порядки.

Обговорення результатів обробки ПВЧ клітинними автоматами.

Клас *DescriptiveStatistics* мови програмування Java [11] забезпечує обчислення цілого ряду статистичних характеристик ПВЧ, серед яких: мінімальний елемент (min), максимальний елемент (max), середнє значення розподілу (mean), середньоквадратичне відхилення (std dev), медіана (median). Всі

вони являються характерними для статистики. Проте для спрощення матеріалу ми використовували тільки один параметр - середньоквадратичне відхилення. Чим менше його значення, тим більш розподіл схожий на рівномірний. Але практичний задовільний рівень відхилення так і не визначений. Ідеально практичний рівень визначений, він дорівнює 1.2%, а задовільний рівень лишається не визначеним.

КА-30 (правило 30) серед інших хаотичних правил демонструє найвищу швидкість і найкращу якість обробки. Практично встановлено, що 20-30 ітерацій за правилом КА-30 приводять до практично ідеального стану розподілення елементів по значенню, а до задовільного стану можливо стане і 10 ітерацій.

Кількість ітерацій суттєво впливає на продуктивність генератора ПВЧ. Одна ітерація зменшує продуктивність удвічі, а 10 ітерацій – майже у 10 раз. То ж при 10 ітераціях в режимі SVGA реально одержати продуктивність 14.4 Mbit/s. В режимі Quad HD (2560 ×1440), а камера такий режим підтримує, теоретично можливо вийти на рівень 265 Mbit/s.

В процес обговорення не включено розгляд обчислювальних можливостей Java. Теоретично можливо процеси обробки ПВЧ за допомогою КА проводити у паралельних потоках, тоді при 8-ми ядерному процесорі можливо досягти рівня 2 Гбіт/сек.

Слід окремо зазначити, що запропонований метод Java обробки згенерованої послідовності дає миттєву статистичну характеристику розподілу значень на відміну від [7], де для цього застосовується громіздке програмне забезпечення. Що дає змогу використовувати звичайний смартфон як апаратну і програмну основу генератора.

Висновки

1. Розподілення елементів по величині носить випадковий характер, проте індивідуальний для кожного окремо взятого прилада (вебкамери), що дозволяє його ідентифікувати.

2. Обробка ПВЧ клітинними автоматами хаотичного типу, підвищують рівномірність розподілу. Самий швидкий метод — це метод Правила 30. Час обробки послідовності приблизно рівний часу генерації.

3. Інтенсивність обробки (кількість ітерацій) залежить від визначеного рівня якості (рівномірності розподілу). Для практично ідеального розподілу необхідно не менше 20 ітерацій.

Загальний висновок: обробка послідовностей випадкових чисел, що згенеровані за допомогою вебкамери можуть слугувати основою для надійного гібридного апаратно-програмного генератора ПВЧ з продуктивністю у перспективі до 2 Гбіт/сек.

Література

1. Ashok Jammi, Raju Y., Munishankaraiah S., Srinivas K. Steganography: an overview. International Journal of Engineering Science and Technology, Vol. 2(10), 2010, 5985-5992.

2. 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

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

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

5. 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

6. Taewan Kim, Seyoon Lee, Seunghwan Yun, Jongbum Kim, Okyeon Yi, Analysis of Radioactive Decay Based Entropy Generator in the IoT Environments, Information Security Applications, 10.1007/978-3-031-25659-2_20, (277-288), (2023). https://doi.org/10.4218/etrij.2020-0119

7. 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

8. R. Diachuk, Y. Dobrovolsky, D. Hanzhelo, H. Prokhorov, and D. Trembach, “Research the Level of Chaotic and Reliability in Webcam-generated Random Number Sequences”, SISIOT, vol. 2, no. 1, p. 01004, Aug. 2024, doi: 10.31861/sisiot2024.1.01004.

9. Hanzhelo D., Prokhorov H. (2024). Investigation Of Statistical Characteristics Of Numerical Random Sequence Obtained From A Web Camera Frame. Herald of Khmelnytskyi National University. Technical Sciences, 337(3(2), 46-51. https://doi.org/10.31891/2307-5732-2024-337-3-6

10. Toffoli T., Margolis N. (1987). Cellular Automata Machines. Cambridge: MIT Press. doi: http://doi.org/10.7551/mitpress/1763.001.0001

11. Dong Y. (2023). Descriptive Statistics and Its Applications. Highlights in Science, Engineering and Technology, 47, 16-23. https://doi.org/10.54097/hset.v47i.8159