https://doi.org/10.31891/2219-9365-2024-78-18
УДК 004

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 МІЖКАДРОВОЇ КОРЕЛЯЦІЇ ХАОСУ, ЩО ГЕНЕРУЄТЬСЯ ВЕБКАМЕРОЮ

Об'єктом дослідження даної роботи є послідовності випадкових чисел (ПВЧ), одержаних з послідовних кадрів вебкамери, що може бути практично використано для створення апаратного генератора ПВЧ.

Проблема, що розглядається, полягала у тому, щоб вирахувати рівень схожості ПВЧ, що були одержані з послідовних кадрів вебкамери і оцінити період зациклювання генерації.

Отримані результати підтвердили припущення про високий рівень хаосу та випадковості при генерації ПВЧ вебкамерою. У послідовних кадрах з захоплення (затримка - 40 мілісекунд) фотодіодна матриця реєструє зміни, які людське око не фіксує. В умовах абсолютної темряві зафіксовано зміни значень яскравості майже у 60% пікселів матриці. Що на 10% вище рівня вимоги до криптостійкості по параметру лавинний ефект. Це можливо пояснюється стохастичною природою взаємодії фотонів з матеріалом сенсора і тепловими шумами. Таким чином, можна говорити про чітку позитивну кореляцію рівня хаосу у кадрах з ростом часу затримки між кадрами.

Особливість досліджень полягає у тому, що для чистоти експерименту генерація кадрів здійснювалась при повній темряві (освітленість - 10-4 люкса), і рівномірно освітленої (освітленість 200 люкс) білої поверхні. Випробування камери на граничних умовах дають повну картину детермінування хаосу при генеруванні ПВЧ.

Результат цього дослідження дозволяє спроектувати алгоритм, який ляже в основу розробки апаратного лабораторного генератору ПВЧ без залучення екзотичного обладнання.

Ключові слова: програмна інженерія, теорія хаосу, криптостійкість, генератор послідовності випадкових чисел, лавинний ефект, вебкамера.

HANZHELO Dmytro, PROKHOROV Georgii
Yuriy Fedkovych Chernivtsi National University

INVESTIGATION OF INTER-FRAME CORRELATION OF WEBCAM-GENERATED CHAOS

The essence of research of this work is sequences of random numbers (RNS) obtained from consecutive frames of a webcam, which can be practically used to create a hardware RND generator.

The problem under consideration was to calculate the similarity level of RNS s obtained from consecutive webcam frames and to estimate the looping period of RNS generation.

The obtained results confirmed the assumption of a high level of chaos and randomness in the generation of RNS by an ordinary webcam. In a set of serial capture frames (delay - 40 milliseconds), the photodiode matrix registers changes that the human eye cannot detect. In conditions of absolute darkness, changes in brightness values were fixed in almost 60% of the pixels of the matrix. Which is 10% higher than the level of the requirement for the cryptoresistance in terms of the avalanche effect parameter. This fact can be explained by the stochastic nature of the interaction of photons with the sensor material and thermal noise. Thus, we can talk about a clear positive correlation of the level of chaos in frames with the growth of the delay time between frames.

The specialty of the research is that for the stringency of the experiment, frame generation was carried out in complete darkness (illumination - 10^{-4} lux) and a uniformly illuminated (illumination of 200 lux) white surface. Tests of the ordinary webcamera under extremly boundary conditions give a complete picture of the determination of chaos in the generation of RNS.

The result of this research makes it possible to design an algorithm that will form the basis of the development of a hardware laboratory RNS generator without the involvement of exotic equipment.

Keywords: software engineering, chaos theory, crypto-resistant, random number sequence generator, avalanche effect, webcamera.

ПОСТАНОВКА ПРОБЛЕМИ У ЗАГАЛЬНОМУ ВИГЛЯДІ ТА ЇЇ ЗВ'ЯЗОК ІЗ ВАЖЛИВИМИ НАУКОВИМИ ЧИ ПРАКТИЧНИМИ ЗАВДАННЯМИ

На сучасному етапі розвитку інформаційних технологій питання кібербезпеки набувають визначального значення, що проявляється у ряді вимог, сформованих Міністерством цифрової трансформації України по збереженню даних державних електронних інформаційних ресурсів[1].

Генератори послідовностей випадкових чисел (ПВЧ) є одними з важливих компонентів кібербезпеки. При використанні ключових даних, сформованих із застосуванням ПВЧ, досягаються декларовані рівні надійності криптографічного захисту інформації (КЗІ). ПВЧ використовуються для

генерації криптографічних ключів, при встановленні захищених з'єднань у різних мережах, при створенні PIN-кодів, для балансування трафіка навантаження, контролю цілісності, і ще для багатьох застосувань [2].

Згенеровані програмно ПВЧ називають псевдовипадковими, оскільки вони детерміністично згенеровані з ентропії. Апаратні генератори формують випадкові числа, засновані на справжніх фізичних стохастичних процесах, наприклад, за рахунок використання шумових процесів резисторів та діодів. Для виробництва спеціальних криптостійких ПВЧ необхідні джерела хаосу, які забезпечують достатню неповторність і непередбачуваність на досить великих об'ємах — від 100 Мбіт і більше.

ПВЧ формують ключову базу, від якості якої залежить надійність та стійкість криптографічних перетворень. Тому одним із важливих напрямів досліджень при розробці генераторів ПВЧ є дослідження методів та засобів оцінки криптографічних характеристик згенерованих випадкових послідовностей.

АНАЛІЗ ДОСЛІДЖЕНЬ ТА ПУБЛІКАЦІЙ

Метод , алгоритм та програмний код програмної генерації псевдовипадкової послідовності є відомим і знаходиться наприклад, для мови Java у відкритому доступі, [3,4], що робить його теоретично вразливим для кібератаки. В кінці 2022 року з'явилась публікація китайських вчених, яка теоретично довела можливість зламу довгих RSA-ключів за допомогою сучасних квантових комп'ютерів. У роботі [5] продемонстровано перший в історії програмної інженерії злом 48-бітного ключа.

Для апаратної генерації випадкових чисел використовують два підходи. Перший це створення та застосування спеціалізованих пристроїв, які використовують будь-які фізичних джерел хаотичного шуму. Так у роботі [5] використовується лічильник бета-випромінювання, що робить відповідні дослідження залежними від додаткового дорогого та екзотичного обладнання.

Однак нерідко виникає завдання отримання ПВЧ на звичайному персональному комп'ютері без застосування додаткового дорогого обладнання.

Враховуючи це, найчастіше більш реальним є другий підхід, заснований на використанні подій від стандартних пристроїв комп'ютера. Найбільш поширеним методом , що використовує цей підхід, є генерація ПВЧ з використанням лічильника внутрішніх тактів процесора. Проте у роботі [6] системні інженери FreeBSD висловлюють недовіру цьому підходу.

У роботі [7] запропонований спосіб генерації на базі оптичного маніпулятора «mouse», що дозволяє генерувати нерівномірно розподілені випадкові числа. Недоліком цього методу є його низька швидкодія, швидкість генерації ПВЧ складає не більше 1 Кбіт/с, що не дозволяє говорити про високошвидкісну систему шифрування.

Недоліки існуючих рішень приводять до висновку про необхідність проведення дослідження кореляції ПВЧ, що згенеровані з послідовних кадрів вебкамери, для подальшої розробки нескладного доступного генератора недетермінованого хаосу на основі вебкамери.

МЕТА ТА ЗАДАЧІ ДОСЛІДЖЕННЯ

Метою даної роботи є дослідження кореляції випадкових послідовностей, де джерелом чисел, виступають значення яскравостей пікселів кадра зображення, сформованого фотоматрицею вебкамери, яка спроектована на основі пристрою із зарядовим зв'язком чи кремній-метал-окисел-напівпровідник матриці вебкамери, підключеної до персонального комп'ютера.

Задачі дослідження ПВЧ полягають у наступному:

- дослідити статистично залежність рівня міжкадрової кореляції від часу;
- оцінити теоретично максимально можливу швидкість генерації..

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

МАТЕРІАЛИ ТА МЕТОДИ ДОСЛІДЖЕНЬ

Для чистоти експерименту як граничний випадок було обрано при денному рівномірному освітленні (200 люкс) зображення білої однорідної стіни (рис. 1). На кадрі можна відзначити, що, незважаючи на однорідність образа, кадр на людське око виглядає неоднорідним. По центру світліший, ніж по периферії, помітні вкраплення інших кольорів, зокрема червоного і фіолетового. Це дає змогу зробити припущення, що у даному масиві пікселів присутні елементи хаосу. Наскільки ця хаотичність задовольняє вимоги КЗІ, наведених у [9,10], буде вияснено у подальшому.

Для ще більшої чистоти і граничності експерименту, щоб уникнути стороннього впливу. вебкамера була вміщена і загерметизована у темній коробці,. Ми спеціально не публікуємо темний знімок, бо на ньому людське око на помітить неоднорідності розподілення точок різної кольорової гами.

Таким чином, дослідження проводились при протилежних граничних умовах: повна темрява (освітленість приблизно 10^{-4} люкса), та рівномірне біле освітлення у 200 люкс. Температура у приміщенні 20^{o} С.

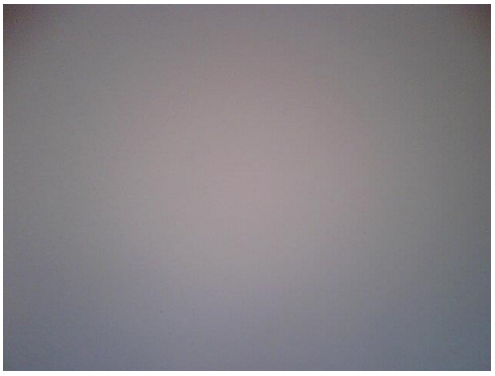

Рис. 1. Зображення білої однорідної стіни, що отримане за допомогою вебкамери

Апаратно-програмна частина

Експеримент проводився на апаратній частині із наступними характеристиками: m/b ASUS Z97K; CPU Intel® Core™ i3-4I70 CPU @ 3.70GHz × 4; 24 Gb RAM; SDD 240 Gb; Web Digital Camera FULL HD 1080P, TrueColor.

Програмне забезпечення, що було використане у експерименті: ОС Ubuntu 22-LTS, 64 bit; Java Amazon.Corretto 17.0.5; IntelliJ IDEA 2023.3.4(Ultimate); пакет com.github.sarxos.webcam версія 0.3.12 – захоплення кадру; пакет javax.imageio – обробка кадра зображення; пакет org.apache.commons.math3.stat.descriptive.DescriptiveStatistics; – статистичні методи.

Обрана вебкамера підтримує наступні формати роздільної здатності: QQVGA (176 × 144); QVGA (320 × 240); VGA (640 × 480); SVGA (800 × 600). У обраній вебкамері верхній рівень роздільної здатності згідно документації становить 800 x 600 (VGA), а по замовчуванню – режим 176 × 144 , Quarter-QVGA resolution. При бажані цей розмір можна розширити до WebcamResolution.HXGA (4096 × 3072) - все залежить від специфікації обраної камери [8].

Дослідження рівня хаосу у згенерованих числових послідовностях

Обчислювальні методи, швидкодія та гнучкість класу BufferedImage та Webcam мови програмування Java дозволили оптимально вилучити з об'єкта кадра вебкамери одразу ж саму послідовність чисел, що відповідають величинам яскравості пікселів матриці, без складних матричних перетворень. Сама послідовність була розміщена у просту структуру даних типу Array і готова для подальшого дослідження.

Пілотні дослідження проводились для роздільної здатності QQVGA (176 × 144), таким чином довжина послідовності становила 76032 числа типу byte [-128 ... +127].

Для 25 захоплених послідовних кадрів (40 мілісекунд затримка) був запрограмований алгоритм порівняння кожного кадра з наступним — всього 24 пари Для кожної такої пари було вирахувано кількість пікселів, які змінили своє значення за 40 мілісекунд.

Камера знаходилась в умовах відсутності освітлення (10^{-4} люкс) без усякого стороннього впливу, то ж можна припустити, зміна значень яскравості пікселів матриці являються результатом внутрішнього неконтрольованого хаосу. Процент пікселів, які змінили свою величину упродовж часу 40 мілісекунд, і буде детермінувати міру хаосу та рівень лавинного ефекту. А процент пікселів, які не змінили свого значення - рівень міжкадрової кореляції з часом.

На рис. 2 зображена міжкадрова кореляція пікселів для режиму захоплення кадрів QVGA(320 × 240) TrueColor.

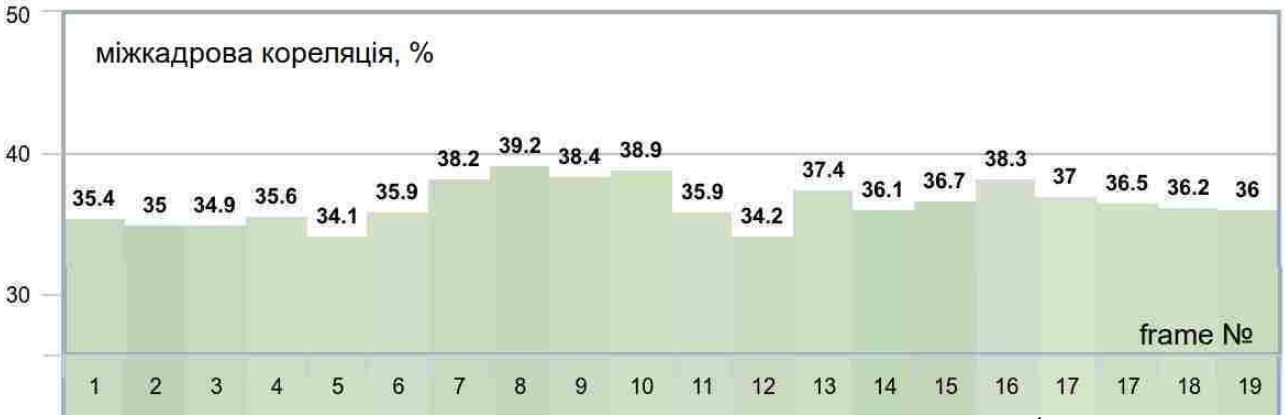

Рис. 2 Рівень міжкадрової кореляції пікселів вебкамери у режимі QVGA при освітленні 10^{-4} люкса

З рисунку видно, що, наприклад, рівень кореляції між першим і другим кадром становить 35.4% - це значить, що 35.4% пікселів не змінили свого стану при переході від першого кадру до другого. Між другим і третім кадром рівень кореляції становив рівно 35% .

Рівень хаосу при переході від першого кадру до другого становив 64.6%, для кадрів 2-3 — 65.0% відповідно. В середньому рівень кореляції потрохи зростає від кадра до кадра і становить приблизно 36%, а рівень хаосу відповідно 64%. Ніякої закономірності не спостерігається, що ще раз підкреслює хаотичність та непередбачуваність значень яскравості пікселів від кадру до кадру.

Подібний експеримент був проведений для різних умов освітленості та різних режимів захоплення кадру. Результати приведені у таблиці 1.

Таблиця 1

Залежність статистичних характеристик рівня кореляції пікселів сусідніх кадрів від режиму захоплення та рівня освітленості

Режим освітленість	**QQVGA(176 * 144)**	**QVGA(320*240)**	**VGA (640*480)**
10^{-4} люкс	min: 0.34 max:0.39 mean: 0.36 std dev: 0.02 cov: 0.81	min: 0.30 max:0.37 mean: 0.33 std dev: 0.02 cov: 0.85	min: 0.28 max:0.33 mean: 0.30 std dev: 0.02 cov: 0.88
200 люкс, однорідна біла поверхня	min: 0.18 max: 0.38 mean: 0.30 std dev: 0.06 cov: 1.0	min: 0.04 max: 0.20 mean: 0.15 std dev: 0.10 cov: 1.0	min: 0.01 max: 0.28 mean: 0.11 std dev: 0.08 cov: 1.0
150 люкс, офіс	min: 0.14 max: 0.20 mean: 0.17 std dev: 0.02 cov: 1.0	min: 0.04 max: 0.15 mean: 0.12 std dev: 0.06 cov: 1.0	min: 0.01 max: 0.12 mean: 0.07 std dev: 0.05 cov: 1.0

Пояснити результати приведені у таблиці 1 можна спираючись на рис.1. Справді, для повної темряви (10^{-4} люкса) та режиму QQVGA(176 × 144) згідно рис.1 мінімальне значення кореляції пікселів становить 34% , що відповідає значенню min: 0.34 у таблиці. Максимальне значення кореляції згідно рис.1 — 39.2 %, що відповідає значенню max: 0.39 у таблиці. Значення mean: 0.36 означає середнє арифметичне, а std dev: 0.02 — середньоквадратичне відхилення. У таблиці присутнє ще одне значення: cov: 0.81 (coverage - покриття), воно означає, що за 25 кадрів 81% всіх пікселів хоча би раз змінив своє значення.

Дослідження швидкодії генерації числових послідовностей

Весь процес генерації можна умовно розділити на три основні етапи:

- одержання самого об’єкту кадра з вебкамери через USB-порт;
- екстракція з цього об’єкту числової послідовності;
- збереження послідовності у якесь сховище (структуру).

В подальшому ці числові послідовності можна зчіпляти (конкатенувати) у більш довгі послідовності або трансформувати у довготривалий потік випадкових чисел.

Код на Java, який захоплює саме кадр з вебкамери виглядає наступним чином:

```
Webcam webcam = Webcam.getDefault();
Dimension dimension = WebcamResolution.VGA.getSize();
webcam.setViewSize(dimension);
webcam.open();
BufferedImage image = webcam.getImage();
```

Код, який відповідає за екстракцію числової послідовності з об’єкта *image:*

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

На виході маємо числову послідовність, розмір якої залежить від режиму захоплення камери (роздільної здатності кадра). Для самого простого режиму QQVGA(176*144) довжина послідовності становить 76082 числа типу byte [-128 ... +127].

Експеримент полягав у наступному:

1. Одержати 1000 кадрів без обробки і заміряти час захоплення.
2. З одного і того ж кадру 1000 раз зробити екстракцію послідовності і заміряти час.
3. Визначити час у перерахунку на 1 кадр і на послідовність довжиною 100 Кбайт.

В результаті експерименту вияснено, що захоплення одного кадру займає 46 мілісекунд в незалежності від кількості кадрів і режиму роздільної здатності.

Збереження (пакування) послідовності в структуру типу Array займає 1 мілісекунду. Дослідження процесу екстракції числової послідовності приведено в табл. 2

Таблиця 2

Залежність часу обробки одного кадру від режиму захоплення

Режим час , мілісекунд	**QQVGA (176*144)**	**QVGA (320*200)**	**VGA (640*480)**	**SVGA (800*600)**
один кадр, захоплення	42 мс	43 мс	45 мс	45 мс
один кадр, обробка.	2.3 мс	5.1 мс	16.9 мс	26.2 мс
100 Кбайт, обробка	3.1 мс	2.2 мс	1.9 мс	1.5 мс
швидкість, Мбайт/сек	1.73	3.99	14.86	20.28

Пояснимо результати експерименту наведені у таблиці знову ж таки для роздільної здатності кадра QQVGA(176*144).

Час захоплення кадру з вебкамери — 42 мілісекунди. За цей час один кадр з вебкамери переганяється через порт USB до в оперативну пам'ять у вигляді об'єкту класу *BufferedImage*. Цей час приблизно однаковий для всіх режимів роздільної здатності і пояснюється тим, що стандарт 25 кадрів/сек обумовлює самий такий час — 1000мс/25 = 40.

За допомогою функціоналу ByteArrayOutputStream з цього об'єкта екстрагується послідовність байтів, що відповідають за стан пікселів матриці фотоприймача. Цей процес займає 2.3 мілісекунди.

У цьому режимі один кадр генерує послідовність у 76 082 байт, то ж у перерахунку на 100 000 байт одержуємо 3.1 мілісекунд. Саме стільки часу потрібно, щоб згенерувати у цьому режимі послідовність у 100 Кбайт.

Захоплення кадру — 42 мілісекунд плюс обробка — 2.3 мілісекунди, згенеровано з кадру 76 082 байти, то ж швидкість генерації становить 76 082 / 44.3 мс = 1.73 Мбайт/сек

Обговорення результатів дослідження метода генерації послідовностей випадкових чисел

Особливістю запропонованого методу була ізоляція вебкамери від зовнішнього впливу, а саме забезпечення повної темряви – 10-4 люкса як граничної умови. Протилежна сторона граничної умови – біла однорідна поверхня, рівномірно освітлена 200 люкс. Ці умови було виконати досить просто порівняно, наприклад, з роботою [6], де генератором хаосу було джерело бета-випромінювання.

Обговорення результатів дослідження міжкадрової кореляції

Опрацювання великих ПВЧ (до 1.5 млн. пікселів) було спрощено використанням класу Stream API та DescriptiveStatistics мови Java. Це швидко і точно обчислило статистичні характеристики.

За рівень кореляції між кадрами була взята процентна частка пікселів , які від кадру до кадру не змінила свого значення. Виявилось :

- рівень кореляції дуже слабкий — максимум 40%;
- рівень зберігається навіть при повній темряві (10^{-4} люкса);
- при денному світлі падає до 11%;
- при зростанні роздільної здатності кадра рівень кореляції падає;
- зміни охоплюють весь об'єм числової послідовності, за 24 кадри кожен піксель щонайменше один раз змінив своє значення;

Обговорення результатів дослідження швидкодії генерації

Швидкість генерації ПВЧ за допомогою вебкамери має принципове обмеження. Це кадрова швидкість. Більше 25 кадрів за секунду принципово не досягнути. Таким чином роздільна здатність — це єдиний параметр який можна варіювати для підвищення швидкодії. Швидкість обробки захопленого кадру у порівнянні з кадровою швидкістю — порівняно мала і можна знехтувати. Максимальну швидкість генерації, що вдалось одержати , 20Мбайт/сек або 160 Мбіт/сек - для роздільної здатності VGA(800 × 600). Використання вебкамери високої роздільної здатності , наприклад 1920 × 1080, теоретично дозволить наблизитись до 1 Гбіт/сек

ВИСНОВКИ З ДАНОГО ДОСЛІДЖЕННЯ І ПЕРСПЕКТИВИ ПОДАЛЬШИХ РОЗВІДОК У ДАНОМУ НАПРЯМІ

1. Рівень міжкадрового хаосу вебкамери навіть при самих несприятливих умовах становить 60%, що дає можливість генерувати довгі ПВЧ з послідовних кадрів вебкамери, що задовільняє вимогу по максимізації періоду зациклювання і рівня лавинного ефекту згідно вимог КЗІ.

2. Швидкість генерації ПВЧ за допомогою послідовних кадрів вебкамери була досягнута 20 Мбайт/сек або 160 Мбіт/сек. Використання камери більш високої роздільної здатності дозволить підняти швидкість на порядок. А використання програмних методів паралелізації обчислень теоретично дозволить необмежену швидкість.

Подальші дослідження мають бути сконцентровані на дослідженні відповідності згенерованих ПВЧ великої довжини (100 Мбіт) тестам NIST [9,10]. Це дозволить оптимізувати межі допустимих значень вхідних параметрів генератора ПВЧ.

Література

1.Про основні засади забезпечення кібербезпеки України: Закон України від 21.06.2018 № 2469-VIII. URL: https://uteka.ua/ua/publication/news-14-novosti-zakonodatelstva-1-osnovnye-principy-obespecheniya-kiberbezopasnosti-ukrainy-prinyat-zakon.

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

4. 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

5. 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

6. 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

7. 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.

8. Webcam-capture Resolution URL: https://github.com/sarxos/webcam-capture/blob/ master/webcam-capture/src/main/java/com/github/sarxos/webcam/WebcamResolution.java

9. **Randomness test. URL :** https://en.wikipedia.org/wiki/Randomness_test

10. Lothar Afflerbach. Criteria for the assessment of random number generators, Journal of Computational and Applied Mathematics, Volume 31, Issue 1, 1990, Pages 3-10, ISSN 0377-0427, https://doi.org/10.1016/0377-0427(90)90330-3.

References

1. Pro osnovni zasady zabezpechennia kiberbezpeky Ukrainy: Zakon Ukrainy vid 21.06.2018 № 2469-VIII. URL: https://uteka.ua/ua/publication/news-14-novosti-zakonodatelstva-1-osnovnye-principy-obespecheniya-kiberbezopasnosti-ukrainy-prinyat-zakon.

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

4. 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

5. 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

6. 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

7. 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.

8. Webcam-capture Resolution URL: https://github.com/sarxos/webcam-capture/blob/ master/webcam-capture/src/main/java/com/github/sarxos/webcam/WebcamResolution.java

9. Randomness test. URL : https://en.wikipedia.org/wiki/Randomness_test

10. Lothar Afflerbach. Criteria for the assessment of random number generators, Journal of Computational and Applied Mathematics, Volume 31, Issue 1, 1990, Pages 3-10, ISSN 0377-0427, https://doi.org/10.1016/0377-0427(90)90330-3.