***Дмитро Ганжело****, аспірант*

*Чернівецький національний університет ім. Юрія Федьковича*

ORCID: 0000-0002-0836-4568

*e-mail: hanzhelo.dmytro@chnu.edu.ua*

***Денис Трембач****, аспірант*

*Чернівецький національний університет ім. Юрія Федьковича*

ORCID: 0000-0001-8095-4186

*e-mail:* [*trembach.denys@chnu.edu.ua*](mailto:trembach.denys@chnu.edu.ua)

***Олег Макарук****, аспірант*

*Чернівецький національний університет ім. Юрія Федьковича*

*e-mail: makaruk.oleh@chnu.edu.ua*

***Олег Павліченко****, аспірант*

*Чернівецький національний університет ім. Юрія Федьковича*

*e-mail:* [*oleh.pavl@gmail.com*](mailto:oleh.pavl@gmail.com)

**ПРИНЦИПОВА СХЕМА ГЕНЕРАТОРА ПОСЛІДОВНОСТІ ВИПАДКОВИХ ЧИСЕЛ, ЩО ГЕНЕРУЄТЬСЯ ВЕБКАМЕРОЮ**

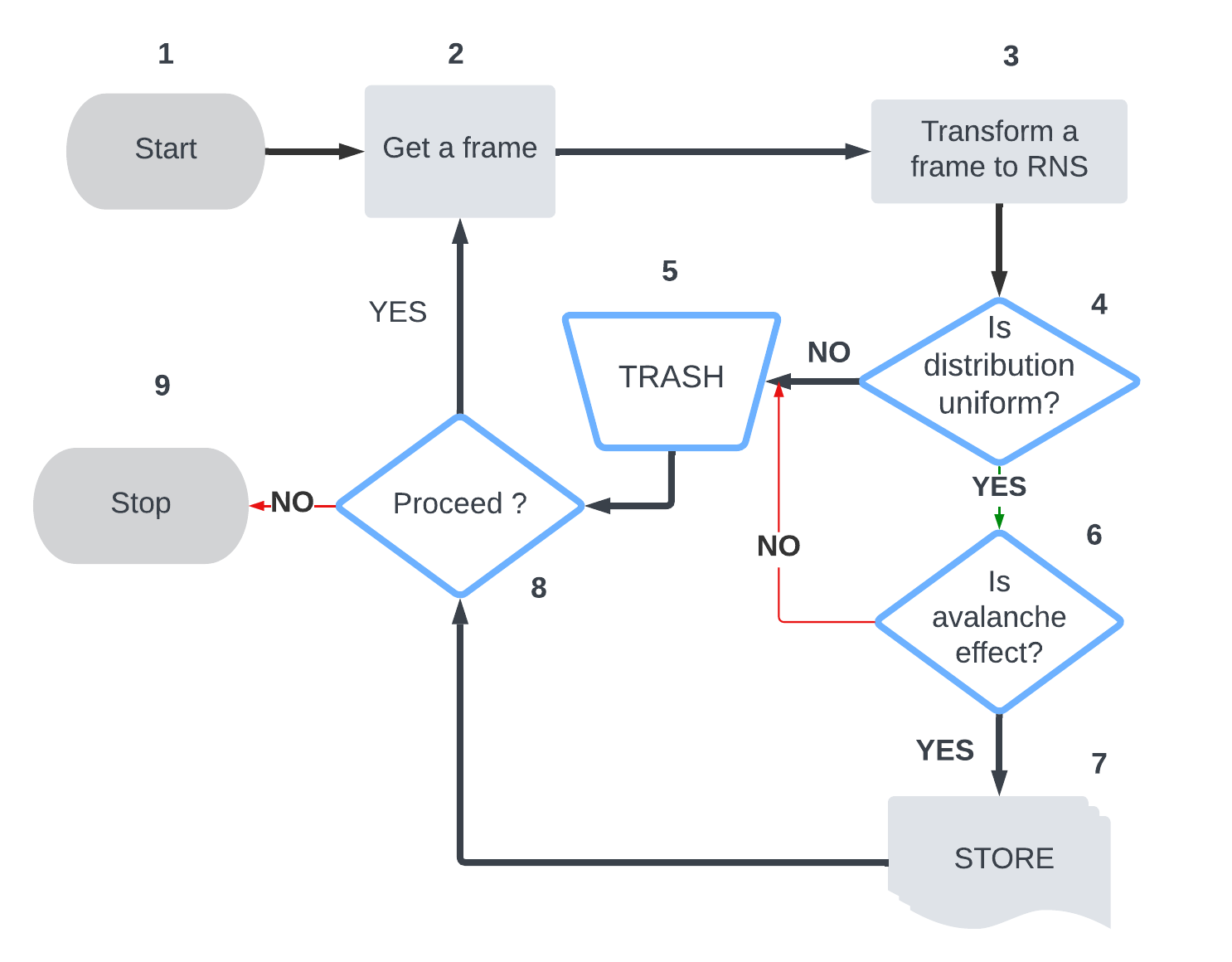
Створення саме апаратного криптостійкого генератора послідовності випадкових чисел (ПВЧ) на основі кадрів веб камери має базуватись спочатку на вирішення задач відповідності згенерованих послідовностей вимогам криптостійкості захисту інформації.

Об’єктом дослідження даної роботи є процес генерації послідовності випадкових чисел (ПВЧ), що одержані з кадрів веб камери.

Проблема, що розглядається, полягала у тому, щоб принципово визначити складові компоненти такого генератора..

Метою даної роботи є попередні дослідження складових компонентів генератора ПВЧ, де джерелом чисел, виступають значення яскравостей пікселів кадра зображення, сгенерованого фотоматрицею веб камери, яка спроектована на основі пристрою із зарядовим зв’язком чи кремній-метал-окисел-напівпровідник матриці веб камери, наприклад, смартфону.

Принципова блок-схема приведена на рисунку 1



Р

Рисунок 1. Принципова блок-схема генератора ПВЧ на основі веб камери.

Старт (1) установлює зв’язок з веб камерою і витягує кадр (2). З цього кадру екстрагується ПВЧ (3). Ця послідовність перевіряється на рівномірність розподілу (4) і на задовільність рівня хаосу (6). У разі невиконання хоча би однієї з цих умов згенерованя послідовність видаляється з пам’яті (5). При виконанні обох цих умов ПВЧ переміщається у сховище (7). У разі підтвердження потреби на продовження роботи система заходить на новий цикл (2). Якщо ні, то закінчує роботу (9).

Перевірка на рівень хаосу (лавинний ефект) і рівномірність розподілу детально розглядалась у роботі [3].

**Висновок:** запопонована блок-схема може бути покладена в основу програмно-апаратного генератора послідовностей випадкових чисел.

**References**

1. Class SecureRandom. All Implemented Interfaces. URL: <https://docs.oracle.com/javase/8/docs/api/java/security/SecureRandom.html>
2. 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
3. Дмитро Ганжело, Георгій Прохоров, 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](file:///C:\Users\fahre\Downloads\ДОСЛІДЖЕННЯ%20ЧИСЛОВОЇ%20ВИПАДКОВОЇ%20ПОСЛІДОВНОСТІ,%20ЩО%20ОДЕРЖАНА%20З%20ВЕБ%20КАМЕРИ.%20(2024).%20Herald%20of%20Khmelnytskyi%20National%20University.%20Technical%20Sciences,%20333(2),%20120-124.%20https:\doi.org\10.31891\2307-5732-2024-333-2-18)