

**LABORATORIUM BEZPIECZEŃSTWA SYSTEMÓW TELEINFORMATYCZNYCH (CZ.
1)**

Wykonali:

Mateusz Irski, Michał Kierzkowski

Data oddania

14.05.2025 r.

Podstawa opracowania:

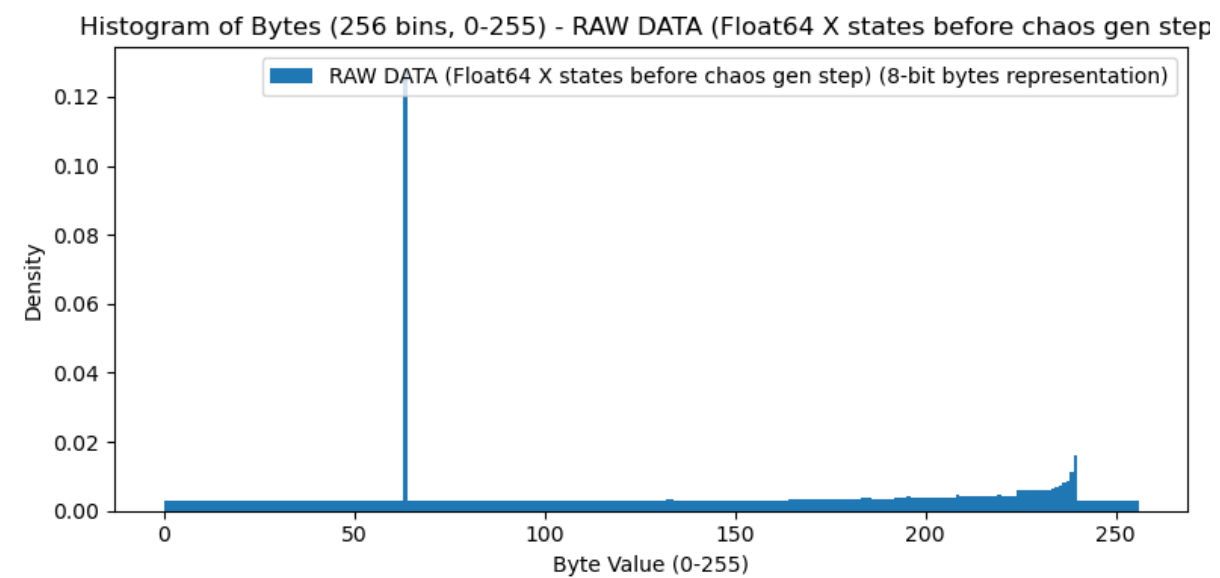
Teh, J.S., Samsudin, A., Al-Mazrooie, M. *et al.* GPUs and chaos: a new true random number generator. *Nonlinear Dyn* **82**, 1913–1922 (2015). <https://doi.org/10.1007/s11071-015-2287-7>

Systematyczny przegląd literatury:

1. baza danych IEEE Xplore,
2. słowa kluczowe: TRNG, graphics,
3. okres publikacji: 2015-2019,
4. relatywnie prosta implementacja
5. zdefiniowany pseudokod algorytmu;
6. spełniane testy NIST.

Analiza źródła entropii:

Algorytm wykorzystuje nieprzewidywalne zachowanie jednostek przetwarzania graficznego (GPU) jako główne źródło entropii. Konkretnie, entropia pochodzi z warunków wyścigu (race conditions), które powstają, gdy wiele równoległych wątków GPU próbuje jednocześnie uzyskać dostęp i modyfikować te same współdzielone lokalizacje pamięci (tablice X i R przechowywane w pamięci globalnej GPU, aby zwiększyć częstotliwość występowania wyścigów). Te niewielkie, nieprzewidywalne zmiany wynikające z warunków wyścigu, wpływają na parametry wejściowe (np. wartość początkową X lub parametr kontrolny r) mapy chaotycznej (konkretnie, mapy logistycznej: $F(x_{n+1}) = rx_n(1 - x_n)$). Ze względu na inherentną właściwość chaosu, tj. wysoką wrażliwość na warunki początkowe i parametry ("efekt motyla"), te minimalne fluktuacje są znacząco wzmacniane podczas iteracji mapy chaotycznej. Połączenie tych dwóch mechanizmów – losowości sprzętowej z GPU i deterministycznego chaosu wzmacniającego tę losowość – skutkuje generowaniem sekwencji o wysokim stopniu nieprzewidywalności.



Entropia wyliczona zgodnie ze wzorem: $e = -\sum_i p_i \log_2(p_i)$, dla powyższego rozkładu wynosi **7,4612** bita.

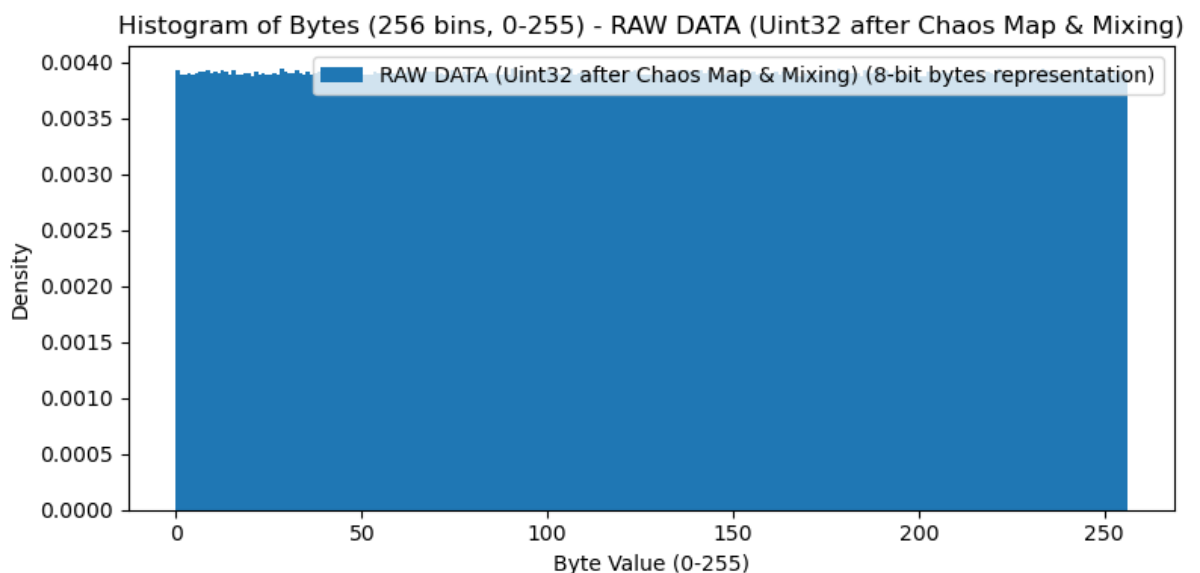
Wyniki testu statystycznego NIST 800-22:

C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	P-VALUE	PROPORTION	STATISTICAL TEST
100	0	0	0	0	0	0	0	0	0	0.000000	0/100	ApproximateEntropy
100	0	0	0	0	0	0	0	0	0	0.000000	0/100	BlockFrequency
100	0	0	0	0	0	0	0	0	0	0.000000	0/100	CumulativeSums
100	0	0	0	0	0	0	0	0	0	0.000000	0/100	FFT
100	0	0	0	0	0	0	0	0	0	0.000000	0/100	Frequency
13	9	15	3	9	10	6	8	19	8	0.025193	100/100	LinearComplexity
100	0	0	0	0	0	0	0	0	0	0.000000	0/100	LongestRun
19	10	15	10	10	10	5	6	9	6	0.144127	44.52/100	NonOverlappingTemplate
100	0	0	0	0	0	0	0	0	0	0.000000	0/100	OverlappingTemplate
0	0	0	0	0	0	0	0	0	0	----	-----	RandomExcursions
0	0	0	0	0	0	0	0	0	0	----	-----	RandomExcursionsVariant
7	16	10	7	9	13	10	8	13	7	0.474986	100/100	Rank
100	0	0	0	0	0	0	0	0	0	0.000000	0/100	Runs
100	0	0	0	0	0	0	0	0	0	0.000000	0/100	Serial
100	0	0	0	0	0	0	0	0	0	0.000000	0/100	Universal

Metoda poprawy właściwości statystycznych:

Aby przetworzyć surowe dane uzyskane z iteracji mapy chaotycznej (która sama w sobie wzmacnia nieprzewidywalność wynikającą z operacji GPU), algorytm stosuje dedykowaną, prostą funkcję post-processingu. Proces ten ma na celu uzyskanie nieprzewidywalnego wyjścia i dalsze zwiększenie dyfuzji:

1. Każda 64-bitowa liczba zmiennoprzecinkowa (double) $X[t_i]$, będąca wynikiem działania mapy logistycznej, jest najpierw **rzutowana (reinterpretowana)** na 64-bitową liczbę całkowitą bez znaku (unsigned long long), oznaczmy ją jako C1.
2. Następnie pobierana jest druga, podobnie przetworzona 64-bitowa wartość C2, pochodząca z innej, odseparowanej lokalizacji ($X[t_i + \text{offset}]$, gdzie $\text{offset} = (\alpha \times \beta) / 2 \bmod \beta$) w tablicy wyników generowanych przez mapę chaotyczną. Ten krok wprowadza mieszanie danych z różnych strumieni obliczeniowych.
3. Obie te wartości, C1 i C2, są dzielone na dwie 32-bitowe części każda: mniej znaczące bity (LSB) i bardziej znaczące bity (MSB). Otrzymujemy w ten sposób cztery 32-bitowe liczby:
 - M1 (LSB z C1)
 - M2 (MSB z C1)
 - M3 (LSB z C2)
 - M4 (MSB z C2)
4. Finalna 32-bitowa liczba losowa jest generowana poprzez kombinację tych części za pomocą **operacji dodawania modularnego i operacji XOR**: $v[t_i] \leftarrow ((M1 + M4) \oplus M3) + M2$.



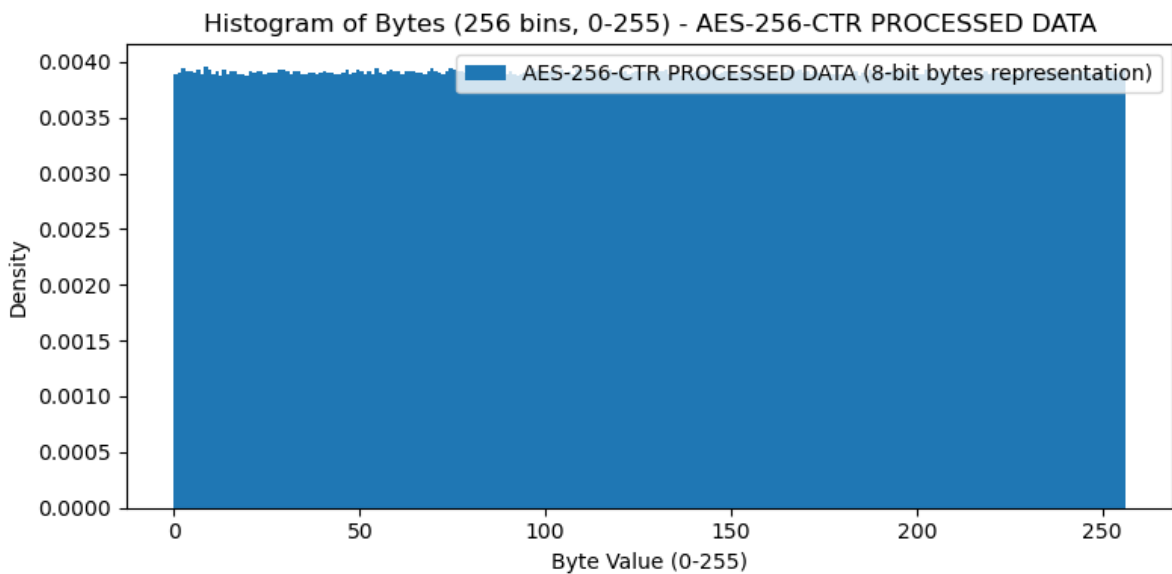
Entropia wyliczona zgodnie ze wzorem: $e = -\sum_i p_i \log_2(p_i)$, dla powyższego rozkładu wynosi **7,9999** bita.

Wyniki testu statystycznego NIST 800-22

C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	P-VALUE	PROPORTION	STATISTICAL TEST
2	11	7	15	19	15	5	4	15	7	0.000439	100/100	ApproximateEntropy
12	11	4	11	13	8	11	12	6	12	0.534146	100/100	BlockFrequency
10	15	9	9	12	12	14	6	7	6	0.316165	100/100	CumulativeSums
15	12	11	6	10	15	9	7	8	7	0.401199	99/100	FFT
11	11	11	13	14	4	9	8	12	7	0.514124	98/100	Frequency
8	6	8	9	7	14	8	11	10	19	0.137282	98/100	LinearComplexity
11	5	10	11	15	6	8	9	9	16	0.275709	100/100	LongestRun
10	7	13	12	7	10	14	9	9	9	0.499393	98.95/100	NonOverlappingTemplate
15	7	18	7	12	11	10	3	6	11	0.037566	96/100	OverlappingTemplate
4	6	3	6	9	4	4	5	6	7	0.442965	53.50/54	RandomExcursions
4	3	7	3	10	8	4	3	8	4	0.350156	53.17/54	RandomExcursionsVariant
14	11	11	6	15	5	8	9	12	9	0.401199	99/100	Rank
11	4	15	11	13	7	7	10	13	9	0.350485	99/100	Runs
12	10	7	12	10	13	9	8	10	9	0.563851	99.50/100	Serial
10	10	6	13	13	13	8	7	11	9	0.759756	98/100	Universal

Metoda poprawy właściwości statystycznych przez zastosowanie AES-256 CTR

Aby przetworzyć pobrany szum audio, każde pobrane 128/256 bitów jest poddawane działaniu AES-256 CTR/SHA3-256



Entropia wyliczona zgodnie ze wzorem: $e = -\sum_i p_i \log_2(p_i)$, dla powyższego rozkładu wynosi 8 bitów.

Wyniki testu statystycznego NIST 800-22

C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	P-VALUE	PROPORTION	STATISTICAL TEST
12	6	9	12	8	15	11	7	11	9	0.678686	99/100	ApproximateEntropy
14	12	6	6	3	14	10	11	9	15	0.108791	99/100	BlockFrequency
9	10	14	14	10	7	8	5	13	10	0.343955	99/100	CumulativeSums
15	8	9	14	7	13	11	14	3	6	0.102526	99/100	FFT
12	12	6	7	15	8	14	12	3	11	0.153763	99/100	Frequency
13	9	7	14	7	12	9	13	4	12	0.366918	99/100	LinearComplexity
12	7	6	10	14	16	9	8	8	10	0.437274	99/100	LongestRun
5	15	7	13	17	16	6	7	9	5	0.476749	98.64/100	NonOverlappingTemplate
11	10	13	10	16	9	12	5	8	6	0.383827	98/100	OverlappingTemplate
4	2	9	6	9	4	5	8	6	9	0.508456	61.50/62	RandomExcursions
7	7	5	6	3	5	4	9	5	11	0.560982	61.44/62	RandomExcursionsVariant
10	10	11	11	10	11	12	11	6	8	0.971699	100/100	Rank
15	6	13	7	9	14	8	10	8	10	0.494392	98/100	Runs
11	9	16	13	6	12	5	9	12	7	0.429273	99/100	Serial
10	6	13	12	13	12	15	4	6	9	0.213309	99/100	Universal

Uwagi:

Praca demonstruje praktyczne i bezpieczne podejście do generowania prawdziwej losowości przy użyciu powszechnie dostępnej technologii.