Generowanie prawdziwych liczb losowych przy użyciu źródeł entropii obecnych w komputerach przenośnych

Dokument konferencyjny - lipiec 2021 r.

DOI: 10.1109/CONECCT52877.2021.9622734

6 autorów, w tym:

Hari Om

2 PUBLIKACJE **3** CYTOWANIA

ZOBACZ PROFIL

Abhishek Banerji

Towarzystwo Edukacji Ludowej

1 PUBLIKACJA **3** CYTOWANIA

ZOBACZ PROFIL

Sivaraman Eswaran

Curtin University Sarawak

47 PUBLIKACJI **250** CYTOWAŃ

ZOBACZ PROFIL

Prasad Honnavalli

Towarzystwo Edukacji Ludowej

88 PUBLIKACJI **259** CYTOWAŃ

ZOBACZ PROFIL

Generowanie prawdziwych liczb losowych przy użyciu źródeł entropii obecnych w komputerach przenośnych

Rahul M Koushik
Wydział Informatyki PES University Bangalore, Indie
rahulkaushik1999@gmail.com

Aravind Perichiappan
Wydział Informatyki Uniwersytet PES Bangalore, Indie
aravindperi@gmail.com

Hari Om
Wydział Informatyki PES University Bangalore, Indie
buzz.hari@gmail.com

Abhishek Banerji
Wydział Informatyki PES University Bangalore, Indie
abhishekbanerji1999@gmail.com

Sivaraman Eswaran
Wydział Informatyki PES University Bangalore, Indie
sivaraman.eswaran@gmail.com

Prasad Honnavalli
Wydział Informatyki PES University Bangalore, Indie
prasad.honnavalli@gmail.com

***Streszczenie* - Liczby *losowe* mają szerokie zastosowanie w różnych dziedzinach, takich jak kryptografia, randomizacja wag początkowych w uczeniu maszynowym i symulacji sztucznej inteligencji, obliczenia Monte Carlo, testy przemysłowe, gry komputerowe, hazard. Generowanie liczb losowych jest możliwe tylko ze źródła entropii. Prawdziwy generator liczb losowych (TRNG) wykorzystuje fizyczne źródło entropii do generowania liczb losowych. Losowość TRNG można scharakteryzować naukowo i zmierzyć. Wadą TRNG jest to, że zazwyczaj wymagają one zewnętrznego urządzenia sprzętowego zawierającego fizyczne źródło entropii. Konieczność tę można wyeliminować, próbując wykorzystać źródła, które są już częścią środowiska urządzenia. W niniejszej pracy podjęto próbę zidentyfikowania takich źródeł i przeanalizowania ich poziomów entropii. Zidentyfikowane źródła entropii są następnie wykorzystywane do budowy modelu, który można wykorzystać do generowania prawdziwie losowych liczb.**

***Słowa kluczowe - losowość*, generowanie liczb losowych, źródło entropii, generator liczb prawdziwie losowych**

I. WPROWADZENIE

Generowanie liczb losowych ma kluczowe znaczenie dla wielu procesów obliczeniowych, takich jak aplikacje kryptograficzne i generowanie procedur. Generator liczb losowych to algorytm, który generuje sekwencję liczb wykazujących właściwość losowości. Wynik dobrego generatora liczb losowych powinien być taki, aby nie mógł zostać określony przez zewnętrznego atakującego.

Generatory liczb losowych można podzielić na dwie szerokie kategorie - generatory liczb pseudolosowych (PRNG) i generatory liczb prawdziwie losowych (TRNG). TRNG to generatory liczb losowych, które wykorzystują naturalne źródła, takie jak światło widzialne lub szum atmosferyczny, do zbierania entropii. Z tego zbioru tworzona jest pula entropii. Ta pula entropii zawiera bity entropii, które są wyodrębniane z wartości dostarczanych przez naturalne źródło entropii. Za każdym razem, gdy konieczne jest wygenerowanie liczby losowej, bity są wyczerpywane z puli entropii w celu wygenerowania liczby, a pula jest ponownie uzupełniana poprzez zbieranie danych ze źródła entropii. Proces zbierania entropii z naturalnych źródeł może być powolny, ponieważ tylko kilka bitów entropii jest zbieranych na raz. To sprawia, że masowe generowanie liczb losowych przy użyciu TRNG jest procesem czasochłonnym i może blokować aplikacje, które potrzebują liczb losowych.

Z drugiej strony, PRNG wykorzystują algorytmy do obliczania długich sekwencji liczb, które wydają się być losowe, w oparciu o początkową liczbę zwaną wartością początkową. W związku z tym PRNG nie wykorzystują źródła entropii (innego niż wartość początkowa), a ich wynik jest deterministyczny. Ponieważ nie ma źródła entropii

Ograniczając tempo produkcji liczb losowych, PRNG są szybsze i pozwalają na masowe generowanie liczb losowych. Ponieważ niewiele krytycznych aplikacji wymaga prawdziwych liczb losowych, PRNG nie są odpowiednie w takich przypadkach. Dlatego też TRNG jest niezbędny do osiągnięcia prawdziwej losowości.

Jakość TRNG zależy przede wszystkim od źródła entropii. Jednak dokładnej ilości entropii obecnej w źródle nie można dokładnie zmierzyć. Dlatego istnieje potrzeba oszacowania ilości entropii, którą można wyodrębnić z danego źródła. Istnieje wiele estymatorów do tego celu, z których najczęściej używanym jest oszacowanie minimalnej entropii zalecane przez National Institute of Standards and Technology (NIST) [5]. Na podstawie tego oszacowania można zidentyfikować dobre źródła entropii.

Istniejące TRNG wykorzystują wewnętrzny sprzęt [1][4][10], a niektóre wymagają nawet specjalnego sprzętu zewnętrznego [2] w celu pozyskania entropii. Co więcej, fizyczne źródło entropii wymagane dla konkretnego TRNG może nie być obecne w środowisku wszystkich urządzeń. Niniejsza praca jest próbą zaprojektowania TRNG, który wykorzystuje wewnętrzne czujniki lub odczyty, które są już obecne w urządzeniu jako źródło entropii.

II. STUDIUM LITERATURY

Kilka protokołów bezpieczeństwa [12] i algorytmów kryptograficznych wykorzystuje TRN. Carmen Camara i in. zaproponowali, że poprzez pomiar sygnału Galvanic Skin Response można określić wartość przewodnictwa skóry, a następnie wykorzystać ją do generowania liczb losowych [2]. Postępowali zgodnie z zaleceniami NIST opublikowanymi w SP800-90B dotyczącymi testowania i szacowania entropii. Kyungroul Lee i in. zaproponowali TRNG, który generuje liczby losowe z widma widzialnego [3]. Widmo widzialne zawiera informacje o więcej niż 300 fal elektromagnetycznych i zawiera dużo nieprzewidywalnego szumu. Szum ten został wykorzystany do wygenerowania liczb losowych. Dzięki tej pracy udowodniono, że poziom entropii można poprawić poprzez warunkowanie za pomocą operacji XOR na danych ze źródła entropii, w porównaniu do entropii obecnej w oryginalnych nieprzetworzonych danych. Marcin Pawłowski i inni próbowali znaleźć realne źródło generowania liczb losowych o wysokiej entropii w ekosystemie Internetu rzeczy (IoT) w [4]. Zintegrowane czujniki (temperatury, wilgotności i dwa różne czujniki światła) zostały przeanalizowane jako potencjalne źródła entropii. W pracy zaproponowano rozwiązanie oparte na metodzie zbierania entropii z najmniej znaczących bitów. Przedstawione podejście zostało przetestowane w czterech różnych eksperymentach, a następnie do dalszej optymalizacji wykorzystano statystyczną technikę dostrajania.

978-1-6654-2849-1/21/$31.00 ©2021 IEEE

2021 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT) | 978-1-6654-2849-1/21/$31.00 ©2021 IEEE | DOI: 10.1109/CONECCT52877.2021.9622734

III. ZBIERANIE ENTROPII

Tradycyjne źródła entropii wykorzystywane przez TRNG obejmują szum atmosferyczny, widmo widzialne [3] i szum termiczny [1]. Wiarygodne gromadzenie danych z tych źródeł może być trudne, zwłaszcza w przypadku urządzeń mobilnych. To znacznie ogranicza wykorzystanie TRNG do generowania liczb losowych. Aby temu zaradzić, źródła, które są już częścią tych urządzeń, mogą być analizowane jako potencjalne źródła entropii dla TRNG. W ten sposób TRNG będzie w stanie działać w środowisku mobilnym i generować bezpieczne liczby losowe z wystarczającą szybkością.

Celem tej pracy było zbudowanie wydajnego generatora liczb prawdziwie losowych, który wykorzystuje źródło, które było już obecne w środowisku urządzenia jako źródło entropii. Potencjalnymi źródłami entropii, które zostały zidentyfikowane w tym celu, są czasy przybycia pakietów, czasy podróży pakietów w obie strony oraz dane z czujników powszechnie występujących w urządzeniach mobilnych, takich jak akcelerometry, żyroskopy i magnetometry. Źródła entropii musiały być zgodne z zaleceniem SP 800-90B [5], opublikowanym przez NIST. Zalecenie to określa zasady projektowania i wymagania dotyczące źródeł entropii używanych przez generatory bitów losowych oraz testy walidacji źródeł entropii.

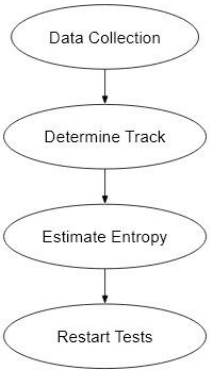
A. Szczegóły projektu

Po pierwsze, potencjalne źródła entropii zostały przeanalizowane poprzez przeprowadzenie walidacji entropii na tych źródłach. W celu przeprowadzenia walidacji zebrano 1 000 000 próbek danych z każdego źródła i utworzono zbiór danych. Po wygenerowaniu zbioru danych, następnym krokiem jest wyodrębnienie bitów entropii i utworzenie pliku binarnego, który jest reprezentacją źródła entropii. Plik ten zostanie wykorzystany do przeprowadzenia walidacji źródła entropii. Proces walidacji pokazano na rys. 1.

Do szacowania entropii NIST SP 800-90B zalecił dwie ścieżki, w zależności od tego, czy źródło entropii jest niezależne i identycznie rozłożone (IID), czy nie-niezależne i identycznie rozłożone (IID). W przypadku IID przeprowadzono testy Chi-kwadrat i testy permutacyjne, a w przypadku nie-IID zastosowano 10 estymatorów zalecanych przez NIST SP 800-90B do oszacowania entropii źródła nie-IID.

B. Gromadzenie danych

Gromadzenie danych ze źródła entropii odgrywa kluczową rolę w generowaniu liczb losowych. Dane są gromadzone na temat zmiennych będących przedmiotem zainteresowania w systematyczny sposób. Dane są następnie czyszczone i przetwarzane w celu utworzenia zbioru danych, który zostanie wykorzystany do oszacowania entropii dostępnej w źródle.



Rys. 1. Walidacja źródła entropii

1) Czas nadejścia pakietu: Sniffer pakietów został zbudowany w języku C przy użyciu libpcap (biblioteka do przechwytywania ruchu sieciowego). Sniffer działa przy użyciu programu obsługi przechwytywania pakietów na urządzeniu sieciowym, które ma być monitorowane. Jeśli żadne urządzenie nie zostanie przekazane jako parametr, wówczas sniffer przechwytuje pakiety na "dowolnym" urządzeniu. Dla każdego przechwyconego pakietu wywoływana jest funkcja zwrotna, która zawiera dane takie jak sam pakiet, długość pakietu itp. oraz najważniejszą wartość danych w tym przypadku znacznik czasu (w mikrosekundach), kiedy pakiet został przechwycony.

Program sniffera pakietów może działać w tle podczas normalnej pracy urządzenia i zbiera dane o pakietach docierających do urządzenia. Dane te są przechowywane w pliku tekstowym. Z tego pliku generowany jest zestaw danych o czasie przybycia pakietów, wyodrębniając tylko istotne informacje dotyczące czasu przybycia pakietów i odrzucając wszystkie inne informacje o pakietach, które zostały podsłuchane.

2) Czas podróży pakietu w obie strony: Aby uzyskać czas podróży pakietu, pakiet HTTP zostanie wysłany do kilku serwerów. Zostanie utworzone gniazdo i utrzymane połączenie z serwerami. Następnie wysyłane jest podstawowe żądanie HTTP GET i otrzymywana jest również odpowiedź. Daje to czas potrzebny pakietowi na dotarcie do serwera i późniejszą odpowiedź na dotarcie do hosta w nanosekundach. Wartości te są przechowywane w pliku tekstowym. Z tego pliku można wygenerować zestaw danych czasu podróży pakietu.

3) Dane z czujników: Aplikacja na Androida została stworzona w celu zbierania danych z czujników znajdujących się w telefonach komórkowych. Aplikacja wykorzystuje platformę czujników systemu Android do wykrywania czujników obecnych w urządzeniu. Na tej podstawie rejestruje się z dostępnymi czujnikami. Za każdym razem, gdy czujnik zgłasza nowe dane, aplikacja przechowuje zgłoszone odczyty w dzienniku. W ten sposób aplikacja jest w stanie zbierać wszystkie dane z czujników, gdy jest uruchomiona i przechowywać je w dzienniku. Na podstawie odczytów można zdecydować, które czujniki są realnym źródłem entropii. Aplikacja zapewnia również opcję eksportu odczytów czujników w postaci pliku CSV, w którym dane każdego czujnika są raportowane osobno. Odbywa się to poprzez przeniesienie zawartości dziennika do pliku w formacie CSV. Plik CSV można wyczyścić i przetworzyć w celu wygenerowania zestawu danych zawierającego dane czujnika. Dane zostały zebrane z akcelerometru, żyroskopu, magnetometru, czujnika światła, czujnika zbliżeniowego, czujników temperatury otoczenia, ciśnienia atmosferycznego i wilgotności względnej podczas ciągłego korzystania z telefonu w celu wygenerowania zestawu danych.

IV. CZYSZCZENIE DANYCH

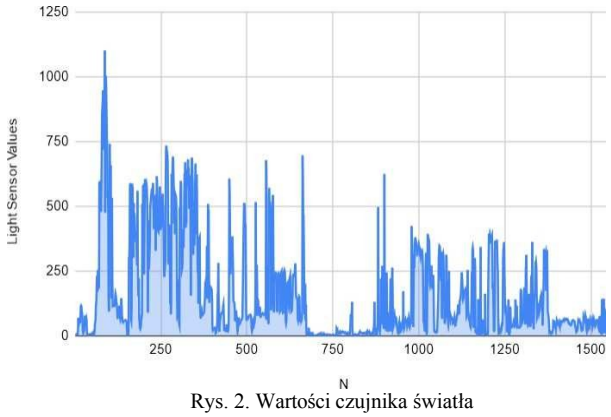
Zanim dane wyodrębnione ze źródła entropii będą mogły zostać ocenione, należy je przygotować do dalszej analizy poprzez modyfikację lub usunięcie danych, które są niekompletne, nieistotne, zbędne, nieprawidłowe lub niewłaściwie sformatowane. W tym przypadku tylko dane z czujników musiały zostać oczyszczone.

A. Dane z czujników

Aplikacja na Androida została zbudowana przy użyciu Android Studio i wykorzystuje bibliotekę Androida do

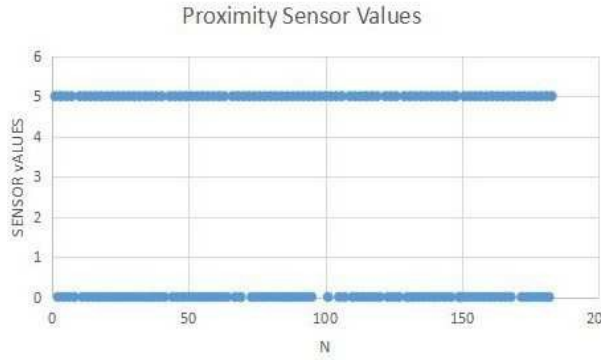
wydobywania informacji o urządzeniu użytkownika. Ta aplikacja najpierw wyodrębnia informacje o wszystkich czujnikach i stale rejestruje wartości z czujników. W tym przypadku niektóre czujniki mogą być nieobecne na kilku urządzeniach. Wartości tych czujników zostały odrzucone. Niektórym czujnikom brakowało zmienności pod względem odczytów. W przypadku czujnika światła maksymalny zakres wartości wynosił 30 000 liczb i

było w większości stopniowe nachylenie, dzięki czemu wartość entropii była bardzo niska, jak zaobserwowano na rys. 2.



Rys. 2. Wartości czujnika światła

Zdarzały się jednak rzadkie przypadki, w których osiągał on wartość limitu. W niektórych przypadkach, nawet bez zmiany wartości, odczyty czujnika były rejestrowane. Wszystkie te wartości zostały również usunięte z wygenerowanego zestawu danych. Niektóre wartości były nieprawidłowo rejestrowane i nie były precyzyjne. Na przykład - 0.0002593994 zostałoby zarejestrowane jako - 2.59E-.
04. Zostało to ponownie przedstawione i dokładnie wyczyszczone. Jak widać na rys. 3, odczyty czujnika zbliżeniowego nie reprezentowały rzeczywistych wartości, ale raczej wyświetlały wartości należące do tej klasy wartości, 0 lub 5,0035, reprezentujące bliskość i odległość.



Rys. 3. Wartości czujnika zbliżeniowego

Można również zaobserwować, że wartości dla czujników światła po wykreśleniu reprezentowały wzgórza o stopniowym nachyleniu. Tempo rejestrowania wartości dla czujników światła i zbliżeniowych było znacznie wolniejsze, ponieważ odnotowywało wartości tylko wtedy, gdy nastąpiła zauważalna zmiana w środowisku w odniesieniu do tych wartości. W związku z tym te dwa czujniki zostały pominięte jako źródła entropii. Ponadto stwierdzono, że czujniki temperatury otoczenia, wilgotności względnej i ciśnienia atmosferycznego są nieobecne na większości urządzeń z systemem Android, w tym na urządzeniach używanych do gromadzenia danych. Na podstawie tej analizy akcelerometr, żyroskop i magnetometr zostały zidentyfikowane jako czujniki, z których zostanie wygenerowany zestaw danych.

V. ANALIZA ENTROPII

Po uzyskaniu zestawów danych z czujników, czasów przybycia pakietów i czasów podróży w obie strony, dane zostały dokładnie wyczyszczone. Każdy zestaw danych zawierał co najmniej 1 000 000 próbek.

A. Zbieranie entropii

Na podstawie przykładowego zbioru danych zauważono, że większość wahań w odczytach dotyczyła najmniej znaczących bitów. Wyodrębniono więc ostatnie 8 bitów każdej wartości próbki w celu utworzenia puli entropii próbki w postaci pliku binarnego. Analiza entropii zostanie następnie przeprowadzona na tym pliku binarnym.

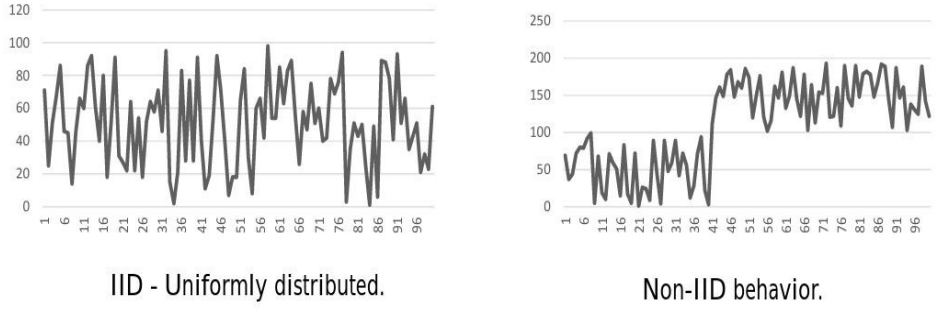
B. Walidacja źródła entropii

Każde z trzech źródeł entropii zostało zweryfikowane zgodnie z publikacją specjalną NIST 800-90b poprzez wykonanie testów minimalnej entropii na przykładowych zestawach danych. Zgodnie z zaleceniami, dla każdego ze źródeł należy wygenerować co najmniej 1 000 000 próbek danych. Pierwszym krokiem było ustalenie, czy próbka reprezentuje niezależne i identycznie rozproszone źródło losowe (IID), czy też nie-niezależne i identycznie rozproszone źródło losowe (nie-IID). Każda próbka w źródle IID ma taki sam rozkład prawdopodobieństwa jak każda inna próbka z tego źródła, a wszystkie próbki są wzajemnie niezależne. Próbki danych IID i danych Non-IID pokazano na rys. 4.

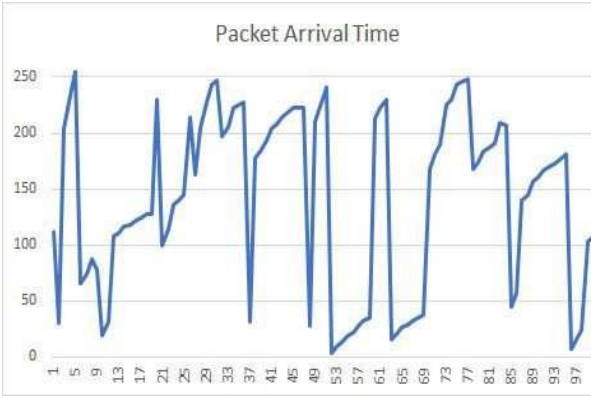
Aby przetestować zachowanie IID, przeprowadzono testy permutacji i testy chi-kwadrat, jak wspomniano wcześniej. Jeśli źródło nie przejdzie któregokolwiek z tych testów, zostanie uznane za źródło inne niż IID.

TABELA I. WYNIKI TESTÓW IID

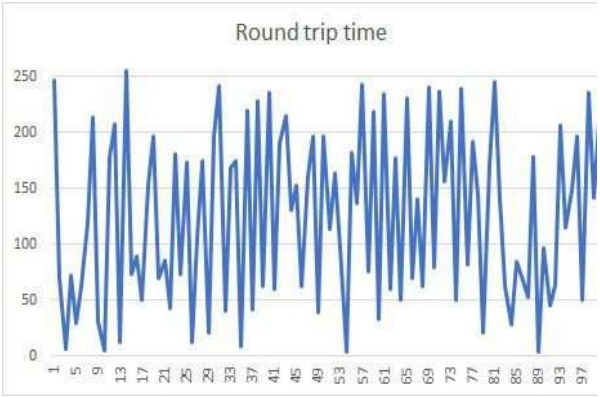
	Testy Chi-kwadrat Testowanie	**Permutacja**
Czas przybycia pakietu	Nie powiodło się	Nie powiodło się
Czas podróży w obie strony	Przyjęto	Przyjęto
Dane czujnika	Nie powiodło się	Nie powiodło się



Rys. 4. Zachowanie IID i bez IID



Rys. 5. Zachowanie czasu przybycia pakietów bez IID



Rys. 6. Zachowanie IID czasu okrążenia pakietu

W oparciu o ścieżkę IID lub ścieżkę bez IID przeprowadzono różne testy w celu określenia minimalnej entropii.

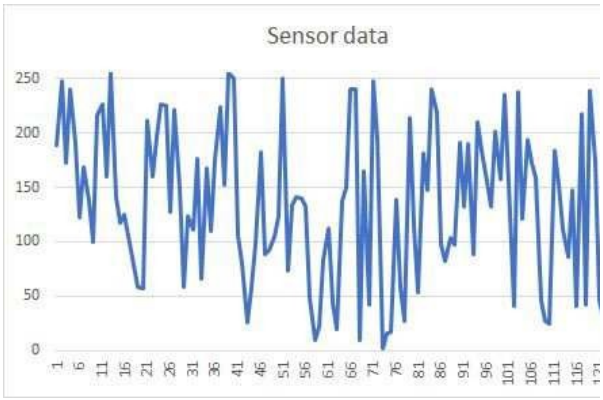
1) Czas przybycia pakietu: 1 000 000 próbek 256 różnych symboli o szerokości 8 bitów, tj. 8 000 000 próbek binarnych zostało załadowanych do pliku binarnego w celu przeprowadzenia analizy entropii. Stwierdzono, że ta próbka ma surową średnią 127,4182258 i medianę 127,000000. Na podstawie oszacowania najczęstszej wartości (MCV) stwierdzono, że entropia ciągu bitów wynosi 0,998120, a entropia przykładowego zbioru danych wynosi 7,869929. Minimalna entropia to min (entropia przykładowego zbioru danych, 8 × entropia ciągu bitów) = 7,869929. Ponieważ jednak nie powiodło się zarówno testowanie permutacji, jak i test chi-kwadrat, pokazuje to, że dane ze źródła entropii nie są ścieżką IID, dlatego oszacowanie MCV nie może być tutaj użyte i należy przeprowadzić dalsze testy nie-IID, aby uzyskać oszacowanie minimalnej entropii źródła. Zachowanie tego źródła inne niż IID pokazano na rys. 5.

TABELA II. WYNIKI ESTYMACJI ENTROPII DLA ZBIORU DANYCH CZASU PRZYBYCIA PAKIETÓW PRZY UŻYCIU TESTÓW INNYCH NIŻ IID

Metoda szacowania	Szacunkowy ciąg bitów
Najczęstsza wartość	0,998120 / 1 bit
Kolizja	1,000000 / 1 bit
Markov	0,999339 / 1 bit
Kompresja	1,000000 / 1 bit
t-Krotka	0,729735 / 1 bit
Przewidywanie MultiMCW	0,987814 / 1 bit
Przewidywanie opóźnienia	0,888284 / 1 bit
Przewidywanie MultiMMC	0,760044 / 1 bit
LZ78Y	0,998408 / 1 bit

Z tabeli II wynika, że minimalna entropia dla tej próbki danych wynosi 0,760044 na bit dla każdego ciągu bitów. Uzupełniający test LRS, który jest częścią tego pakietu, nie został uruchomiony na tym zbiorze danych, ponieważ jest on alternatywą dla testu t-Tuple i jest konieczny do uruchomienia tylko wtedy, gdy test t-Tuple nie działa wydajnie.

2) Czas podróży pakietu: 1 007 203 próbek 256 różnych symboli o szerokości 8 bitów, tj. 8 057 624 próbek binarnych zostało załadowanych do pliku binarnego w celu przeprowadzenia analizy entropii. Stwierdzono, że ta próbka ma surową średnią 127,440730 i medianę 127,000000. Na podstawie najczęstszej wartości oszacowano, że entropia ciągu bitów wynosi 0,998588, a entropia przykładowego zbioru danych wynosi 7,880283. Min-entropia to min(entropia przykładowego zbioru danych, 8 × entropia ciągu bitów) = 7,880283. W przypadku testu niezależności Chi-kwadrat i testu dobroci dopasowania Chi-kwadrat zaobserwowane wartości p wyniosły odpowiednio 0,885775 i 0,179445. Przeszedł również wszystkie testy w permutacji



Rys. 7. Zachowanie danych z czujników inne niż ID

zestaw testów. Testy te obejmują między innymi liczbę przebiegów kierunkowych, testy średniej i maksymalnej kolizji, testy okresowości i kowariancji w odstępach w potęgach 2, testy kompresji. Zachowanie IID czasu podróży w obie strony pokazano na Rys. 6, gdzie przedstawiono zestaw próbek z tego źródła danych.

3) Dane z czujnika: 1 014 288 próbek 256 różnych symboli o szerokości 8 bitów, tj. 8 114 304 próbek binarnych zostało załadowanych do pliku binarnego w celu przeprowadzenia analizy entropii. Stwierdzono, że ta próbka ma surową średnią 127,467826 i medianę 127,000000. Na podstawie oszacowania najczęstszej wartości stwierdzono, że entropia ciągu bitów wynosi 0,986571, a entropia przykładowego zbioru danych wynosi 7,658332. Minimalna entropia to min (entropia przykładowego zbioru danych, 8 × entropia ciągu bitów) = 7,658332. Jednak test chi kwadrat nie powiódł się, była to próbka nie-IID, dlatego ta minimalna entropia nie może być uważana za minimalną entropię przykładowych danych i należy przeprowadzić dalsze testy nie-IID. Podzbiór przykładowych danych został wykreślony na wykresie, jak pokazano na Rys. 7, aby pokazać zachowanie źródła inne niż IID. Wyniki testu Non-IID dla oszacowania entropii tego źródła podano w tabeli III.

TABELA III. WYNIKI ESTYMACJI ENTROPII DLA ZBIORU DANYCH CZUJNIKÓW PRZY UŻYCIU TESTÓW INNYCH NIŻ IID

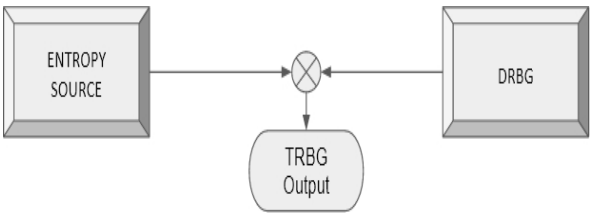
Metoda szacowania	Szacunkowy ciąg bitów
Najczęstsza wartość	0,986571 / 1 bit
Kolizja	0,946944 / 1 bit
Markov	0,988442 / 1 bit
Kompresja	0.843168 / 1 bit
t-Krotka	0,928582 / 1 bit
Przewidywanie MultiMCW	0,992958 / 1 bit
Przewidywanie opóźnienia	0,810587 / 1 bit
Przewidywanie MultiMMC	0,983829 / 1 bit
LZ78Y	0,986633 / 1 bit

Po przeanalizowaniu poziomów entropii trzech źródeł, jako źródła entropii dla TRNG wybrano dane z czujników i dane dotyczące czasu podróży pakietów. Jednak czas przybycia pakietu został odrzucony, ponieważ minimalna entropia tego źródła była nieoptymalna do dalszego przetwarzania.

VI. TRNG CONSTRUCTION

Po zidentyfikowaniu realnych źródeł entropii wykorzystano je do skonstruowania dwóch TRNG zgodnie z zaleceniami określonymi w SP 800-90C [6]. Do skonstruowania TRNG wybrano metodę XOR, jak pokazano na rys. 8.

W ramach tego podejścia konieczne było zidentyfikowanie i zintegrowanie zatwierdzonego przez SP 800-90A PRNG i wykorzystanie jego wyjścia XOR-ed z wyjściem TRNG do wygenerowania liczby losowej. Zaletą takiego podejścia jest to, że w przypadku awarii źródła entropii, TRNG nadal będzie generować liczby losowe na poziomie losowości PRNG. W ten sposób PRNG działa jako rozwiązanie awaryjne, a także zwiększa losowość wyprowadzaną przez TRNG. W przypadku TRNG czasu podróży w obie strony, do tego celu wybrano generator liczb pseudolosowych obecny w bibliotece UNIX gcrypt. W przypadku TRNG opartego na czujnikach użyto zatwierdzonego przez SP 800-90A generatora liczb losowych znajdującego się w bibliotece kryptograficznej Bouncy Castle.



Rys. 8. Konstrukcja XOR-TRNG

Ciągłe testy kondycji zostały zintegrowane z TRNG w celu monitorowania kondycji źródeł entropii podczas ich działania, aby zapewnić wykrycie wszystkich awarii źródeł. Testy kondycji obejmowały test liczby powtórzeń i adaptacyjny test proporcji. Krótki ogólny pseudokod dla obu TRNG podano poniżej:

1) Żądanie otrzymania losowej liczby o określonej długości bitu.
2) Żądanie niezbędnej ilości danych ze źródła (źródeł) entropii.
3) Analizować entropia źródło zdrowie poprzez przeprowadzanie ciągłych testów kondycji na próbkach danych.
4) Wyodrębnienie bitów entropii z pobranych danych.

5) Połączenie bitów entropii wyodrębnionych z wielu próbek danych w jeden ciąg bitów.
6) Zażądaj zatwierdzonego przez NIST SP 800-90A DRBG do wygenerowania losowych bitów o wymaganej długości.
7) Wykonanie operacji XOR na losowych bitach wygenerowanych przez DRBG z ciągiem bitów wygenerowanym ze źródła entropii (s).
8) Konwertuje wynik operacji XOR na dodatnią liczbę całkowitą.
9) Zwraca tę liczbę całkowitą jako wygenerowaną liczbę losową, a następnie czeka na następne żądanie.

VII. WALIDACJA TRNG

Do walidacji losowości skonstruowanego TRNG wykorzystano testy Diehard, baterię testów statystycznych opracowanych przez George'a Marsaglię. Testy te obejmują między innymi test rang macierzy, test odstępów urodzinowych, test strumienia bitów, test parkingu i test losowych kulek. Testy diehard zostały zaimplementowane w aplikacji dieharder opracowanej przez Roberta G. Browna [7]. Aplikacja ta została użyta na zbiorach danych do obliczenia siły losowości TRNG.

1) Czas okrążenia pakietu: W przypadku czasu transferu w obie strony TRNG wygenerowano losową sekwencję ponad 36 milionów 32-bitowych liczb w celu utworzenia zbioru danych wymaganego do przeprowadzenia trudnych testów. Czas przejazdu w obie strony TRNG przeszedł wszystkie testy w pakiecie diehard. Wyniki testów przedstawiono w tabeli IV.

TABELA IV. WYNIKI TESTÓW DIEHARD DLA RTT TRNG

Test	p-value	Ocena
Test odstępów urodzinowych	0.24781361	Przyjęto
Nakładające się permutacje	0.55055752	Przyjęto
Szeregi macierzy 32x32	0.99257668	Przyjęto
Ranga macierzy 6x8	0.08266680	Przyjęto
Test strumienia bitów	0.58486077	Przyjęty
Nakładające się pary o rzadkim obłożeniu test	0.00510408	Przyjęty
Nakładające się czworokąty rzadkie test zajętości	0.01441708	Przyjęty
Test DNA	0.50687371	Przyjęty
Test zliczania 1s (strumień)	0.38285950	Przyjęty
Test zliczania 1s (bajty)	0.34599065	Przyjęto
Test parkingu	0.14465251	Przyjęto
Test sfer 2d	0.41704303	Przyjęty
Test kuli 3d	0.80421574	Przyjęty
Test ściskania	0.37776156	Przyjęty
Przeprowadza test	0.79189668	Przyjęty
Test kości	0.72493546	Przyjęty

2) Dane czujników TRNG: W przypadku danych czujników TRNG wygenerowano losową sekwencję ponad 12 milionów 32-bitowych liczb w celu utworzenia zbioru danych wymaganego do przeprowadzenia trudnych testów. Sensor TRNG przeszedł wszystkie testy w pakiecie diehard. Chociaż przeszedł test 2d sfer, został oceniony jako słaby. Nawet w przypadku bardzo silnego RNG jest bardzo prawdopodobne, że od 0 do 2 testów da słabą ocenę. Zwiększając przykładowy zestaw danych, wynik ten można jeszcze poprawić. Wyniki testów przedstawiono w tabeli V.

TABELA V. WYNIKI TESTÓW DIEHARD DLA CZUJNIKÓW TRNG

Test	p-value	Ocena
Test odstępów urodzinowych	0.59240373	Przyjęto
Nakładające się permutacje	0.03219095	Przyjęto
Szeregi macierzy 32x32	0.84276204	Przyjęto
Ranga macierzy 6x8	0.82542551	Przyjęty
Test strumienia bitów	0.33635406	Przyjęty
Nakładające się pary nieliczne test zajętości	0.06644921	Przyjęty
Nakładające się czworokąty rzadkie test zajętości	0.09174928	Przyjęty
Test DNA	0.65076972	Przyjęto
Test zliczania 1s (strumień)	0.41364038	Przyjęto
Test zliczania 1s (bajty)	0.21762042	Przyjęty
Test parkingu	0.58615511	Przyjęty
Test sfer 2d	0.99871810	Słaby
Test kuli 3d	0.90351670	Przyjęty
Test ściskania	0.06892150	Przyjęty
Przeprowadza test	0.39494760	Przyjęty
Test kości	0.40835817	Przyjęty

VIII. WNIOSKI

Na podstawie analizy i wyników można stwierdzić, że źródła takie jak czujniki urządzeń i czas obiegu pakietów mogą być wykorzystywane jako źródła entropii do konstruowania prawdziwych generatorów liczb losowych, które dobrze sprawdzają się w praktycznych zastosowaniach. Dzięki tym badaniom ustalono losowość tych źródeł entropii i skonstruowanych przy ich użyciu TRNG. Jako rozszerzenie tych badań, przyszłe prace mogą zostać wykonane w celu analizy bezpieczeństwa tych TRNG poprzez testowanie ich podatności na ataki. Można również przeprowadzić dalsze testy, na przykład wykorzystując dane wyjściowe TRNG w aplikacjach do szyfrowania obrazów, a następnie oceniając siłę szyfrowania za pomocą testów losowości NPCR i UACI [13]. Można również pracować nad TRNG, aby zbudować dla nich interfejsy opakowujące zgodne z GSL, tak aby mogły być używane przez inne aplikacje.

REFERENCJE

[1] B. Jun i P. Kocher, "The Intel random number generator", Cryptography Research Inc., biała księga, kwiecień 1999, s. 1-8.

[2] C. Camara, H. Mart´ın, P. Peris-Lopez i M. Aldalaien, "Design and analysis of a true random number generator based on GSR signals for body sensor networks", Sensors, 19 (9): 2033, 2019.

[3] K. Lee, S. Lee, C. Seo i K. Yim, "TRNG (True Random Number Generator) method using visible spectrum for secure communication on 5G network", IEEE Access, vol. 6, styczeń 2018,
pp. 12838-12847.

[4] M. P. Pawłowski, A. Jara i M. Ogorzałek, "Harvesting entropy for random number generation for internet of things constrained devices using on-board sensors" Sensors, 15(10): 26838-26865, 2015.

[5] M. S. Turan, E. Barker, J. Kelsey, K. McKay, M. Baish i M. Boyle, "Recommendation for the entropy sources used for random bit generation", NIST Special Publication 800-90B, styczeń 2018.

[6] E. Barker i J. Kelsey, "Recommendation for random bit generator (RBG) constructions", NIST Special Publication 800-90C, kwiecień 2016.

[7] R. G. Brown, "Dieharder: A random number test suite", https://webhome.phy.duke.edu/~rgb/General/dieharder.php, [dostęp online].

[8] L. E. Bassham, A. L. Rukhin, J. Soto, J. R. Nechvatal, M. E. Smid, S. D. Leigh, M. Levenson, M. Vangel, N. A. Heckert i D. L. Banks, "A statistical test suite for random and pseudorandom number generators for cryptographic applications", NIST Special Publication 800-22 Rev 1a, Sept. 2010.

[9] E. B. Barker i J. Kelsey, "Recommendation for random number generation using deterministic random bit generators", NIST Special Publication 800-90A Rev 1, czerwiec 2015.

[10] E. Schreck i W. Ertel, "Disk drive generates high speed real random numbers", Microsystem Technologies, vol. 11, 2005, str. 616-622.

[11] C. Hennebert, H. Hossayni i C. Lauradoux, "Entropy harvesting from physical sensors", Proceedings of the sixth ACM Conference on Security and Privacy in Wireless and Mobile Networks (WISEC'13), Apr. 2013, pp. 149-154.

[12] S. Eswaran, A. Srinivasan i P. Honnavalli, "A threshold-based, real-time analysis in early detection of endpoint anomalies using SIEM expertise", Network Security, 2021 (4), Apr. 2021, pp. 7.16. https://doi.org/10.1016/S1353-4858(21)00039-8.

[13] Y. Wu, J. Noonan i S. Agaian, "NPCR and UACI randomness tests for image encryption", Cyber Journals: Multidisciplinary Journals in Science and Technology, Journal of Selected Areas in Telecommunications, Apr. 2011, pp. 31-38.