Paweł CZERNIK, Wiesław WINIECKI
POLITECHNIKA WARSZAWSKA, INSTYTUT RADIOELEKTRONIKI,
ul. Nowowiejska 15/19, 00-665 Warszawa

Pomiary losowości danych wytwarzanych przez generator wbudowany w procesory firmy Intel z rodziny Ivy Bridge

Mgr inż. Paweł CZERNIK

Absolwent Wydziału Elektroniki i Technik Informacyjnych PW (2008r., kierunek: Telekomunikacja, specjalność: Radioelektronika i Techniki Multimedialne). Obecnie doktorant na Wydziale Elektroniki i Technik Informacyjnych PW, członek IEEE. Aktualne obszary zainteresowań: algorytmy i systemy kryptograficzne, nowoczesne technologie komunikacyjne i programowe w skupionych i rozproszonych systemach pomiarowo-sterujących, systemy pomiarowe, przyrządy wirtualne.

e-mail: P.Czernik@ire.pw.edu.pl

Prof. dr hab. inż. Wiesław WINIECKI

Prof. nzw. na Wydziale Elektroniki i Technik Informacyjnych PW. Kierownik zespołu Komputerowej Techniki Pomiarowej. Członek Komitetu Metrologii i Aparatury Naukowej PAN, wiceprezes POLSPAR, członek IEEE. Autor lub współautor 4 książek i ponad 170 publikacji naukowych. Obszary zainteresowań: systemy pomiarowe, przyrządy wirtualne, nowoczesne technologie komunikacyjne i programowe w skupionych i rozproszonych systemach pomiarowo-kontrolnych.

e-mail: W.Winiecki@ire.pw.edu.pl

Streszczenie

W artykule przedstawiono pomiary losowości układu generatora kryptograficznie bezpiecznych wartości losowych, wbudowanego w procesor i5-3450. Układ ten wytwarza wartości losowe, przeznaczone do wykorzystania przez oprogramowanie kryptograficzne realizowane na komputerze zarówno w zastosowaniach serwerowych, jak i na stacjach roboczych powszechnego użytku. Prezentowana technika pomiarowa losowości generatora polega na akwizycji danych losowych, a następnie badaniu ich bezpieczeństwa kryptograficznego. Bezpieczeństwo to zostało zmierzone na bazie autorskiego oprogramowania oraz środowiska RDieHarder. Zaprezentowano tu także wyniki pomiarów wydajności generacji danych losowych. Uzyskane najistotniejsze wyniki badań zostały przedstawione oraz przeanalizowane.

Słowa kluczowe: kryptografia, generatory liczb losowych, bezpieczeństwo informacji, Intel „Ivy Bridge".

Measurements of random data generated by the generator built in Ivy Bridge Intel processors

Abstract

Most of the current IT systems or dedicated distributed measurement and control systems use computer networks and various types of wireless communication media, such as mobile GSM/UMTS, Bluetooth, ZeegBee, WiFi. Recently, in the era of globalization, there has been observed a significant increase in threats to various information systems. The answer to these threats must be implementation of cryptographic methods to the existing and proposed systems, to increase security. Security of encryption systems is closely related to generation of some unpredictable values. Random number generators (RNG), and especially true random number generators (TRNG) are one of the most basic elements of cryptography. This paper presents measurements of the randomness of a cryptographically secure random number generator embedded in the i5-3450 processor. This device generates random values which are used by the cryptographic software implemented on both server and general-purpose workstations. The presented measurement technique is based on random data acquisition, and then testing the cryptographic security of this data. The security was determined based on the software developed by the author and the software environment called RDieHarder. There are also presented and analysed the measurement results of the randomness generation performance.

Keywords: cryptography, random number generators, safety information, Intel „Ivy Bridge".

1. Wprowadzenie

Większość projektowanych obecnie systemów teleinformatycznych czy dedykowanych sieci pomiarowo-sterujących wykorzystuje sieci komputerowe oraz różnego rodzaju bezprzewodowe media komunikacyjne, takie jak: telefonia komórkowa GSM/UMTS, Bluetooth, ZeegBee, WiFi. Ostatnimi czasy w dobie globalizacji obserwuje się znaczące zwiększenie zagrożenia różnych systemów informacyjnych. Odpowiedzią na te zagrożenia nieuchronnie musi być wprowadzenie do istniejących oraz projektowanych systemów metod kryptograficznych zwiększających bezpieczeństwo. Z uwagi na coraz większą i częstszą integrację systemów pomiarowych z sieciami komputerowymi i telekomunikacyjnymi bezpieczeństwo rozproszonych systemów pomiarowych uległo drastycznemu obniżeniu z uwagi na powszechny dostęp do stosowanych mediów komunikacyjnych [1].

Bezpieczeństwo dzisiejszych systemów kryptograficznych jest ściśle związane z generacją pewnych nieprzewidywalnych wartości. Generatory liczb losowych (ang. random number generator - RNG), a szczególnie generatory liczb prawdziwie losowych (ang. True Random Number Generator - TRNG), są jednym z najbardziej podstawowych elementów kryptograficznych. Projektowanie generatorów liczb losowych do celów kryptograficznych jest dzisiaj dynamicznie rozwijającą się dziedziną badań, nad którą pracują rzesze naukowców z całego świata, świadczą o tym coraz to wydajniejsze i bezpieczniejsze rozwiązania, za jakimi stoją takie marki jak Intel, czy Microsoft.

Generator liczb losowych (RNG) [2] jest to program komputerowy lub układ elektroniczny generujący liczby losowe. Ze względu na sposób generowania liczb losowych można wyróżnić dwa rodzaje stosowanych generatorów: sprzętowe (TRNG) działające na zasadzie ciągłego pomiaru procesu stochastycznego i programowe (PRNG). Olbrzymią zaletą generatora sprzętowego, szczególnie ważną w kryptografii, jest jego nieprzewidywalność, wynikająca z nieprzewidywalności samego procesu fizycznego [3]. Dlatego też często w literaturze są one określane jako „prawdziwe" generatory liczb losowych. Najczęściej wykorzystywane procesy losowe to szum termiczny oraz rozpad pierwiastka promieniotwórczego, choć możliwe są również inne rozwiązania. W [4] można znaleźć opis różnych metod generowania ciągów prawdziwie losowych opartych m. in. na: pomiarze czasu spoczynku klawiatury, szumu termicznego diody półprzewodnikowej, rozpadu radioaktywnego, niestałości częstotliwości własnej oscylatora oraz ładunku gromadzonego w kondensatorze polowym w ustalonym czasie. Oczywiście ciągi uzyskane z tych źródeł nie są dokładnie losowe: zazwyczaj obarczone są korelacją. Istnieją specjalne metody na usunięcie takich wad z ciągu. Przy ich wykorzystywaniu należy pamiętać o wpływie środowiska zewnętrznego na ich jakość.

Odpowiednie urządzenia są produkowane przez różne firmy np. IBM, OMNISEC, SOPHOS.

Istnieje również programowa możliwość realizacji takiego typu generatora, która została zaimplementowana w jądrze systemu operacyjnego Linux (urządzenie /dev/random). Algorytm pracy tego „urządzenia" przedstawia się następująco. Generator zbiera, dane losowe z otoczenia i gromadzi w „pojemniku losowości" (ang. random pool) o rozmiarze 512 bajtów. Danymi zbieranymi przez generator są np. kod naciśniętego klawisza lub odstęp czasu między kolejnymi przerwaniami. Generator szacuje w czasie

dodawania danych, na ile dane te były losowe i na tej podstawie ustala ilość losowych danych w pojemniku (dalej zwana entropią). Kodu naciśniętego klawisza nie można uznać za losowy (więc jego dodanie do pojemnika nie zwiększa entropii). Także przerwania występujące w regularnych odstępach czasu nie dają losowych danych. W tym przypadku generator oblicza różnicę poprzedniego odstępu i obecnego i na tej podstawie określa ile losowych bitów zostało dodane (tak naprawdę, to tylko odstępy między przerwaniami mogą zwiększyć entropię). Przy dodawaniu do pojemnika 32-bitowego słowa, jest ono przesuwane cyklicznie o określoną ilość bitów, a następnie jest realizowana operacja XOR ze słowami na pozycjach i, i + 7, i + 9, i + 31, i + 99 (liczby to współczynniki pewnego wielomianu CRC; stosuje się też inne współczynniki), gdzie i to pozycja w pojemniku ostatnio wstawionego słowa. Otrzymana wartość jest znowu obracana cyklicznie (tym razem o jeden bit) i wstawiana na pozycję i - 1 (i jest liczone modulo rozmiar pojemnika). Przy dodawaniu słowa do pojemnika nie można wykorzystać kryptograficznych funkcji haszujących ze względu na wymaganą wydajność (słowa są dodawane przy prawie każdym przerwaniu). Przy pobieraniu bajtu z pojemnika (o ile entropia jest większa od zera), zwracana jest zawartość pojemnika przekształcona funkcją jednokierunkową (SHA lub MD5). Następnie cała zawartość pojemnika jest przekształcana jeszcze raz za pomocą SHA lub MD5. Oczywiście zmniejszana jest także entropia. Niestety wadą tego generatora jest jego mała wydajność, przez co jest stosowany jako jedynie zarodek (ang. seed), dla kryptograficznie bezpiecznych generatorów liczb pseudolosowych np. „urządzenia" */dev/urandom*.

Generatory używane w kryptografii poza tym, że powinny być dobre w sensie statystycznym, muszą być także nieprzewidywalne. Ma to na celu zapewnienie, aby przeciwnik znając poprzednie bity produkowane przez generator nie był w stanie przewidzieć kolejnych. Ta ważna cecha jest nazywana bezpieczeństwem kryptograficznym generatora (ang. cryptographicaly secure) [6].

Głównym celem badań układu generatora kryptograficznie bezpiecznych wartości losowych, wbudowanego w procesor i5-3450 są pomiary jego rzeczywistego bezpieczeństwa kryptograficznego w różnych warunkach klimatycznych. Badania te weryfikują możliwość praktycznego zastosowania tego urządzenia dla systemów pomiarowych pracujących w przemysłowych i wojskowych zakresach temperatur.

2. Losowość generatorów

Obecnie mamy do czynienia z szeroką gamą generatorów liczb losowych i pseudolosowych, ale niestety niewiele z nich nadaje się do zastosowań w systemach kryptograficznych, a jeszcze mniej w kryptograficznie bezpiecznych sieciach pomiarowo-sterujących małej mocy. W jaki sposób klasyfikować i „mierzyć" przydatności tych generatorów? Odpowiedź na to pytanie daje dziedzina matematyki, jaką jest probabilistyka. Umożliwia ona dogłębną analizę przydatności do określonych zastosowań rozmaitych typów obecnych, jak i nowo projektowanych generatorów liczb losowych i pseudolosowych.

Oceniając generatory zazwyczaj używa się pojęcia losowości, mniej lub bardziej intuicyjnie. Ogólnie uważa się iż losowość generatora, czyli jego jakość, oceniamy po tym jak produkowana przez niego sekwencja jest różna od prawdziwie losowej sekwencji. Jak w ogóle zmierzyć jakość losowość? Właśnie do tego używa się testów statystycznych. Nie udzielają one dokładnej, deterministycznej odpowiedzi, ale pozwalają z pewnym prawdopodobieństwem stwierdzić, czy dana sekwencja „wygląda" na losową. Pokazują, czy sekwencja wyprodukowana przez dany generator posiada pewne własności statystyczne, charakterystyczne dla sekwencji prawdziwie losowej lub czy nie występują w niej pewne defekty. Należy podkreślić iż nie ma jednoznacznej metody pozwalającej określić, czy generator jest „dobry", czy „zły". Zazwyczaj raczej stosuje się założenie, że badany układ lub algorytm „wygląda na losowy" dopóki nie znajdziemy testu, którego produkowana sekwencja nie przejdzie. Do tego momentu taką sekwencję uznajemy za losową, a generator za „dobry".

Pierwszym, historycznym podejściem do sformułowania metod oceny PRNG były postulaty Golomba [7]:

- w cyklu s^N (sekwencja okresowa) liczba jedynek różni się od liczby zer o co najwyżej jeden;
- w cyklu s^N przynajmniej połowa podsekwencji złożona z tych samych bitów ma długość 1, ¼ długość 2, 1/8 długość 3, itd. aż przekroczymy długość sekwencji. Wśród tych podsekwecji powinno występować prawie tyle samo złożonych z zer i z jedynek;
- funkcja autokorelacji przyjmuje tylko dwie wartości:

$$N * C(t) = \sum_{i=0}^{N-1} (2s_i - 1) * (2s_{i+t} - 1) = \begin{cases} N \\ K \end{cases} \qquad (1)$$

gdzie równanie to przyjmuje wartość równą N jeśli $t=0$ oraz wartość K gdy $1 \le t \le N-1$, gdzie K jest pewną liczbą całkowitą.

Sekwencję spełniającą postulaty Golomba nazywamy sekwencją pn (ang. pseudo-noise sequence). Postulaty Gołomba są przykładem warunków jaki musi spełnić sekwencja, aby została uznana za losową.

Większość gotowych pakietów testów nie zmusza użytkownika do samodzielnego znajdowania odpowiednich wartości odpowiednich rozkładów. Występuje tam za to pojęcie *p - wartości*, czyli wyniku testu, będącego liczbą z przedziału (0,1). *p - wartość* dla danej wartości statystyki testowej i testu T, wyraża się wzorem:

$$p = P[T > t/H_0] = 1 - F(t) \qquad (2)$$

gdzie H_0 oznacza hipotezę podstawową. Jeśli dystrybuanta zmiennej losowej F jest ciągła, to p ma rozkład równomierny w przedziale (0,1).

Jak analizowanych warunkach zinterpretować wynik testu postaci *p - wartości*? W tym przypadku uznaje się, że jeśli uzyskana *p - wartość* jest zbyt bliska 0 lub 1, to sekwencja nie przeszła testu. Oczywiście pojęcie „zbyt bliska" jest umowne i zależy od uznania użytkownika, który może zdecydować się na mniej lub bardziej ostre kryterium oceny testu. Można zastosować dokładniejszą metodę oceny wyników takiego testu. W tym przypadku generuje się pewną ilość sekwencji i odpowiadających im *p - wartości*, a następnie stosuje się np. test Kolmogorov'a-Smirnov'a wskazujący, czy dana sekwencja ma rozkład równomierny.

3. System laboratoryjny do badania układu generatora wartości losowych

W celu wykonania badań układu generatora wartości rzeczywiście losowych firmy Intel został zakupiony procesor z rodziny „Ivy Bridge" i5-3450 z zintegrowanym procesorem grafiki, płyta główna firmy Asus z wyprowadzeniem dla grafiki zintegrowanej, 4GB pamięci RAM typu DDR3 firmy Patriot, dysk 500GB na interfejsie SATA III firmy Seagate oraz zasilacz ATX 500W. Na tym zestawie badawczym został osadzony system operacyjny Linux Ubuntu 12.04 w wersji 64 bitowej. Następnie został zainstalowany kompilator „gcc", środowisko „R", oprogramowanie „Dieharder" i „RDieharder" oraz autorskie oprogramowanie do badania losowości generatorów [6]. Z witryny internetowej firmy Intel została pobrana biblioteka dedykowana do obsługi badanego układu generatora. Biblioteka ta stawowi środowisko programistyczne dedykowane na platformę operacyjną Linux, zaimplementowane w języku C i asemblerze, umożliwiające bezpośredni dostęp do rejestrów procesora. Rejestry te przechowują wyprodukowane przez ten generator wartości losowe. Na bazie tej biblioteki zostało zaimplementowane dedykowane oprogramowanie w języku C. Oprogramowanie to umożliwia przebadanie generatora, na bazie autorskiego oprogramowania przy wykorzystaniu RDieHarder. Zaimplementowany algorytm pomiarowy zapewnia

funkcjonalność rejestracji 100MB danych losowych w postaci pliku na dysku oraz w pamięci RAM. Rejestracja danych w pamięci RAM ma na celu umożliwienie przeprowadzenia badania szybkości produkcji wartości losowych przez badany generator. Taki sposób postępowania jest zawiązany z bardzo długim czasem zapisu danych na dysku w porównaniu z czasem zapisu na pamięci RAM. Rejestracja na dysku umożliwia późniejszą analizę wyprodukowanych danych za pomocą autorskiego oprogramowania przy wykorzystaniu środowiska RDieHarder, na bazie metodyki badania *p-wartości*. Metodyka badawcza pomiaru losowości została szerzej omówiona w [6].

4. Badania układu generatora wartości losowych

Wykonane zostały badania bezpieczeństwa kryptograficznego układu generatora wartości losowych na bazie autorskiego oprogramowania [6]. Badania te zostały przeprowadzone w temperaturze otoczenia wynoszącej 23^o Celsjusza. Zarówno dla przypadku pracy komputera przez ok. 30 min, kiedy nie pobierano danych z generatora; dla pracy komputera przez ok. 30 min z załączonym nieprzerwanym poborem wartości losowych oraz w chwili startu systemu operacyjnego. Wykonane zostały także badania tego układu w temperaturze otoczenia wynoszącej 8^o Celsjusza, w chwili startu systemu operacyjnego oraz zostały one powtórzone po 5ciu minutach pracy komputera.

Uzyskane wyniki zostały obliczone przez „silnik" oprogramowania „DieHarder", niezależnie dla 25ciu testów, po zebraniu odpowiednio dużego, reprezentatywnego zbioru badanych danych, jaki stanowi 100MB wartości losowych.

5. Wyniki badań

Na poniższych rysunkach zostały zaprezentowane wykresy *p-wartości*, na których zamieszczono na osi poziomej konkretne zrealizowane testy (oznaczona numerami, jednoznacznie je identyfikujące na bazie tabeli nr. 1), a na osi pionowej odpowiadające im *p-wartości*, odpowiednio dla: testu Kuiper-Kolmogorov-Smirnov'a, testu Kolmogorov-Smirnov'a oraz testu Wilcoxon'a.

P-wartość (*ang. P-value* - czyli prawdopodobieństwo testowe) jest to w statystycznym testowaniu hipotez najmniejszy poziom istotności testu, dla którego hipoteza zerowa jest odrzucona. Na każdym z poniższych rysunków pozioma niebieska linia oznacza dolną granicę poziomów istotności α równą 0.01, natomiast czerwona - górną granice $1 - \alpha$ równą 0.99.

Rysunek 1 prezentuje otrzymane wyniki dla układu generatora wartości losowych w temperaturze wynoszącej 23^o Celsjusza, dla przypadku pracy komputera przez ok. 30 min, kiedy nie pobierano danych z generatora. Rysunek 2 przedstawia wyniki dla układu pomiarowego w tej samej temperaturze, uzyskane w chwili startu systemu operacyjnego. Rysunek 3 prezentuje wyniki dla układu pomiarowego w temperaturze otoczenia wynoszącej 8^o Celsjusza, w chwili startu systemu operacyjnego. Na rysunku 4 zostały przedstawione wyniki uzyskane w temperaturze otoczenia wynoszącej 8^o Celsjusza, po 5-ciu minutach pracy komputera.

Najlepsze wyniki zwróciły testy układu generatora wartości rzeczywiście losowych wbudowanego w procesor Intel i5-3450 w warunkach pracy w temperaturze wynoszącej 23^o Celsjusza oraz przy przyjęciu wartości tzw. poziomu istotności α równego 0.01. W analizowanym przypadku układ ten nie przechodzi poprawnie jednego badania pakietu, tj.: *diehard_operm5* dla pracy komputera przez ok. 30 min oraz testu *diehard_squeeze* dla przypadku pracy komputera przez ok. 30 min z załączoną nieprzerwaną generacją wartości losowych.

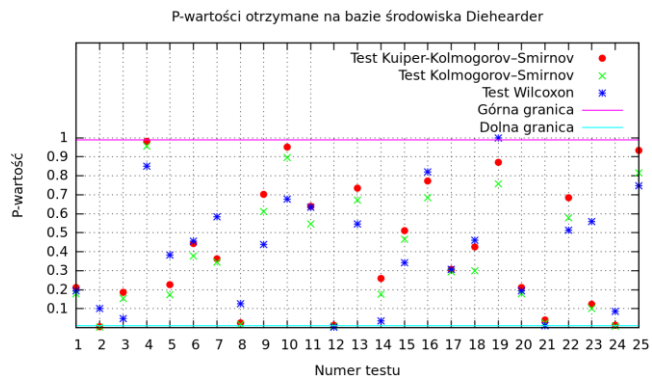
W temperaturze 23^o dla przypadku realizacji procesu archiwizacji danych losowych przy starcie systemu operacyjnego, zgromadzone dane nie przechodzą dwóch testów, tj.: *diehard_count_1s_byte* i *diehard_craps*.

Ten sam układ pomiarowy, gdy temperaturę otoczenia obniżymy do 8^o Celsjusza w warunkach startu systemu operacyjnego, nie przechodzi czterech badań, tj.: *diehard_operm5*, *diehard_rank_32x32*, *diehard_squeeze* i *rgb_lagged_sum*.

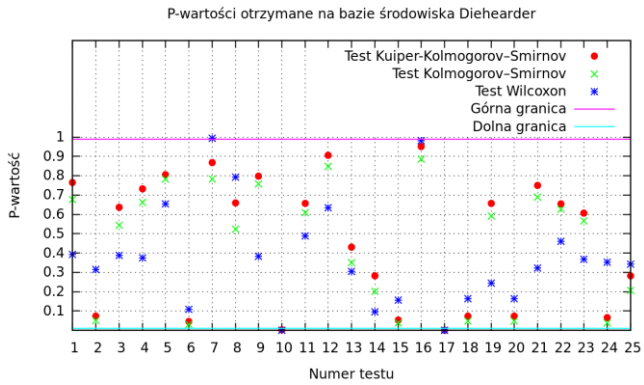
W temperaturze 8^o Celsjusza dla przypadku przypadku pracy komputera przez 5 minut generator ten nie przechodzi trzech badań, tj.: *diehard_count_1s_byte*, *diehard_squeeze* i *rgb_lagged_sum*.

Tab. 1. Numery testów przypisane konkretnym testom pakietu DieHarder
Tab. 1. The numbers assigned to the suitable tests of the DieHarder test environment

Nazwa testu	Numer testu
diehard_birthdays	1
diehard_operm5	2
diehard_rank_32x32	3
diehard_rank_6x8	4
diehard_bitstream	5
diehard_opso	6
diehard_oqso	7
diehard_dna	8
diehard_count_1s_stream	9
diehard_count_1s_byte	10
diehard_parking_lot	11
diehard_2dsphere	12
diehard_3dsphere	13
diehard_squeeze	14
diehard_sums	15
diehard_runs	16
diehard_craps	17
sts_monobit	18
sts_runs	19
sts_serial	20
rgb_bitdist	21
rgb_minimum_distance	22
rgb_permutations	23
rgb_lagged_sum	24
rgb_kstest_test	25

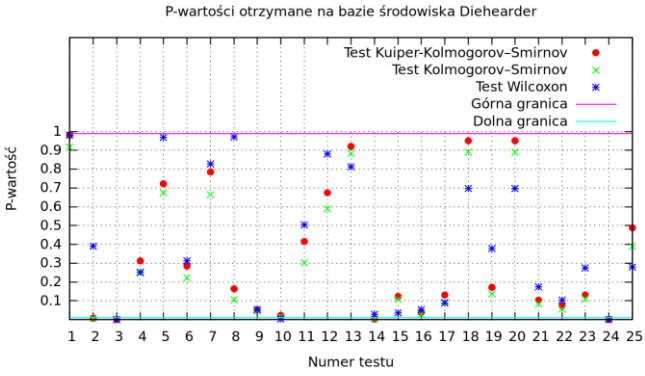


Rys. 1. Wykres *p-wartości*, prezentujący wyniki dla układu generatora wartości losowych w temperaturze 23^o Celsjusza, dla przypadku pracy komputera przez ok. 30 min, gdy nie pobierano danych z generatora
Fig. 1. P-value chart presenting the results for the random generator at the ambient temperature equal to 23 ° C, the computer working for about 30 minutes, when random data was not taken from the generator



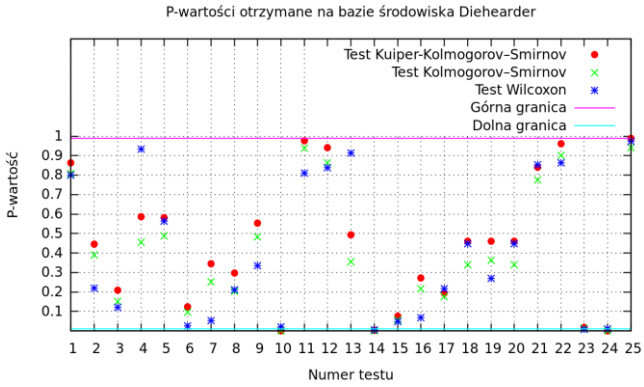
Rys. 2. Wykres *p-wartości*, prezentujący wyniki dla układu pomiarowego w temperaturze otoczenia wynoszącej 23° Celsjusza, w chwili startu systemu operacyjnego

Fig. 2. P-value chart presenting the results for the random generator at the ambient temperature equal to 23 ° C and for starting the computer



Rys. 3. Wykres *p-wartości*, prezentujący wyniki dla układu pomiarowego w temperaturze otoczenia wynoszącej 8° Celsjusza, w chwili startu systemu operacyjnego

Fig. 3. P-value chart presenting the results for the random generator at the ambient temperature equal to 8 ° C, for starting the computer



Rys. 4. Wykres *p-wartości*, prezentujący wyniki dla układu pomiarowego w temperaturze otoczenia wynoszącej 8° Celsjusza, po 5ciu minutach pracy komputera

Fig. 4. P-value chart presenting results for the random generator at the ambient temperature equal to 8 ° C, after 5 minutes of the computer work

Podniesienie poziomu istotności α do wartości 0.1 powoduje, że układ ten nie przechodzi 6 badań w temperaturze 23° dla pracy komputera przez ok. 30 min oraz 6 testów dla pracy komputera przez ok. 30 min z załączoną nieprzerwaną generacją wartości losowych. W temperaturze 23° dla przypadku realizacji procesu archiwizacji danych losowych przy starcie systemu operacyjnego, zgromadzone dane nie przechodzą 8 testów. Dla temperatury 8° Celsjusza, w warunkach startu systemu operacyjnego, generator ten nie przechodzi 10ciu badań oraz 9ciu badań dla przypadku pracy komputera przez czas równy 5 minut.

6. Wnioski

Uzyskane wyniki wykazują, że układ generatora wartości rzeczywiście losowych wbudowany w procesor Intel i5-3450 przechodzi większość badań zestawu DieHarder. Udowadnia to, iż układ ten stanowi bezpieczny generator wartości losowych dedykowany dla systemów komputerowych ogólnego przeznaczenia oraz rozwiązań serwerowych. Uzyskane wyniki potwierdzają przypuszczenia odnośnie losowości występującej w badanym układzie pomiarowym. Bardzo istotną zaletą tego generatora jest jego bardzo wysoka wydajność generacji losowych danych, tj. generacja 100MB danych wymaga zaledwie ok. 450ms (zapis danych do pamięci RAM). Dostępny do tej pory generator wartości losowych w systemie operacyjnym Linux o nazwie */dev/urandom*, wymaga do produkcji takiego samego zbioru danych czasu generacji wynoszącego ok. 5s i 320ms. Niestety wadą generatora wbudowanego w procesor Intel i5-3450 jest istotna wrażliwość na warunki klimatyczne (temperaturę pracy). Wada ta jest bardzo niekorzystna w przypadku konieczności zastosowania tego generatora w rozwiązaniach przemysłowych czy wojskowych. Zrealizowany przez autorów generator wartości rzeczywiście losowych dedykowany dla systemów pomiarowo – sterujących o asymetrycznych zasobach, opisany w [4] i [6], został przebadany w granicach temperatur -6° do 30°. Badania wykazały, że układ jest niewrażliwy na warunki klimatyczne (temperaturę pracy) w tych granicach.

7. Literatura

[1] Winiecki W., Adamski T., Bobiński P., Łukaszewski R.: Bezpieczeństwo Rozproszonych Systemów Pomiarowo Sterujących (RSPS), Przegląd Elektrotechniczny, (2008), nr 5, 220-227.

[2] Zieliński R..: Generatory liczb losowych, WNT, Warszawa 1979.

[3] Czernik P.: Sprzętowy generator kluczy kryptograficznych dedykowany dla systemów pomiarowo-sterujących o asymetrycznych zasobach. Elektronika – Konstrukcje Technologie Zastosowania, 11 (2012), 5-19.

[4] Schneider B.: Kryptografta dla praktyków, WNT, 1995.

[5] Knuth D., Sztuka programowania, Tom II – Algorytmy seminumeryczne, WNT, 2002.

[6] Czernik P.: Cryptographically secure hardware random number generator dedicated for distributed measurement and control systems: Proceedings of the SPIE, nr 84542D, (2012).

[7] Menezes A., Oorschot P., Yanstone S.: Handbook of Applied Cryptography, WNP, 2005.

[8] Czernik P.: Metodyka testowania bezpieczeństwa generatorów liczb pseudolosowych w systemach pomiarowo-sterujących. Prace Instytutu Lotnictwa, 201, nr 6 (2010), 20-34.

[9] Czernik P., Winiecki W.: Analiza sygnałów chaotycznych do realizacji kryptograficznie bezpiecznego bezprzewodowego kanału komunikacyjnego. Pomiary Automatyka Kontrola, nr 9 (2012), s. 785-788.

otrzymano / received: 08.02.2013
przyjęto do druku / accepted: 01.04.2013 *artykuł recenzowany / revised paper*