Grzegorz GANCARCZYK [2], Agnieszka DĄBROWSKA – BORUCH [1, 2], Kazimierz WIATR [1, 2]
[1] AKADEMIA GÓRNICZO – HUTNICZA IM. STANISŁAWA STASZICA W KRAKOWIE, Al. Mickiewicza 30, 30-059 Kraków
[2] AKADEMICKIE CENTRUM KOMPUTEROWE CYFRONET AGH, ul. Nawojki 11, 30-950 Kraków

Sprzętowy detektor szyfrowanej informacji przesyłanej w sieciach TCP/IP

Mgr inż. Grzegorz GANCARCZYK

Absolwent kierunku Elektronika i Telekomunikacja (2009), student I roku studiów doktoranckich Wydziału EAIiE Akademii Górniczo – Hutniczej w Krakowie. Obecnie pracownik ACK CYFRONET AGH. Jego zainteresowania związane są z probabilistyką, procesami stochastycznymi, zjawiskiem szumu, cyfrowym przetwarzaniem sygnałów oraz akceleracją obliczeń numerycznych z wykorzystaniem logiki reprogramowalnej.

e-mail: g.gancarczyk@cyfronet.pl

Dr inż. Agnieszka DĄBROWSKA - BORUCH

Absolwentka kierunku Elektronika i Telekomunikacja Wydziału EAIiE AGH (2002), dr nauk technicznych (2007). Obecnie jest adiunktem w Katedrze Elektroniki AGH oraz członkiem Zespołu Akceleracji Obliczeń ACK CYFRONET AGH. Jej zainteresowania naukowe to kompresja obrazu, systemy czasu rzeczywistego, układy programowalne oraz rekonfigurowalne.

e-mail: adabrow@agh.edu.pl

Prof. dr hab. inż. Kazimierz WIATR

Studia AGH Kraków (1980), doktor nauk technicznych (1987), doktor habilitowany (1999) i profesor (2002). Profesor zwyczajny w Akademii Górniczo - Hutniczej oraz dyrektor ACK CYFRONET AGH. Prowadzone prace badawcze dotyczą komputerowego sterowania procesami, systemów wizyjnych, systemów wieloprocesorowych, układów programowalnych, rekonfigurowalnych systemów obliczeniowych i sprzętowych metod akceleracji obliczeń.

e-mail: wiatr@agh.edu.pl

Streszczenie

Artykuł prezentuje sposób realizacji, cechy charakterystyczne i zasadę działania urządzenia wykrywającego pakiety zawierające dane zaszyfrowane przesyłane w sieciach opartych o stos protokołów TCP/IP. Detektor zrealizowano w oparciu o system *SPARTAN 3E Development Kit* firmy Digilent [1]. Kluczowym elementem jest układ FPGA xc3s1600e firmy Xilinx [2]. W artykule przedstawiono schemat blokowy detektora, informacje o sprawności detekcji rozwiązania programowego oraz sprzętowego, zasobach logicznych zajętych przez układ.

Słowa kluczowe: analiza ruchu sieciowego, FPGA, logika reprogramowalna, sniffing, szyfrowanie.

Hardware detector of encrypted information transmitted in the TCP/IP networks

Abstract

The paper describes how to realize a device which can detect encrypted data transfer in computer networks based on the TCP/IP protocols stack. Its features and principles of operation are given. The device is based on the Digilent's *SPARTAN 3E Development Kit* [1] whose key element is the Xilinx's xc3s1600e [2]. The available publications about distinguishing ciphertext from plaintext tell only that methods typical for randomness check of encrypting algorithms can be used [6]. Many alternative (in field of data distinguishing), interesting publications about steganography [7], computer worms and viruses detection can be easily found [3, 4]. Exemplary implementations of those in FPGA are not difficult to find, either [8]. Lack of publications in the field of encrypted message detection was partial motivation for this paper (Section 1). The presented algorithm of encrypted data detection is based on theorems from [9, 10]. It has advantages and disadvantages, which are discussed (Section 2). The detector (of so called 2nd order) chosen for implementation has good theoretical efficiency (Tab. 1). Its block diagram is shown in Fig. 1 (Section 3). The results of synthesis and implementation are given in Tab. 2, and its efficiency in Tab. 3. The functionality of all blocks of Fig. 1 is discussed (Sections 4 and 5). The efficiency of the implemented device is almost as good as the theoretical one. There are two main limitations – lower (100 B) and upper (1460 B) length of the Ethernet frame data field, and maximum frequency of device clock, which makes it unable (as for xc3s1600) to operate in Gigabit Ethernet networks (Section 6). The presented device can be used as a network data analyzer, a ciphertext detector and a network anomaly detector.

Keywords: encryption, FPGA, network data analysis, reconfigurable computing, sniffing.

1. Wstęp

Bezpieczeństwo lokalnych sieci komputerowych wymaga ich nieustannego monitorowania w celu wykrycia anomalii pojawiających się w trakcie ich pracy. Odstępstwo od modelu zachowania stworzonego dla monitorowanej sieci może świadczyć o infekcji takiej sieci (np. zawirusowanie, zarobaczenie), bądź nieautoryzowanym dostępie do niej przez osoby trzecie [3, 4]. Analiza ruchu sieciowego i przesyłanych danych może mieć więc na celu nie tylko poprawę przepustowości sieci i podniesienie jakości świadczonych usług, lecz również poprawę ich bezpieczeństwa. W [4] autorzy zaproponowali analizę pól nagłówka i danych ramki sieciowej w celu wykrycia zainfekowania przez robaki sieciowe. Tymczasem w [3] autor proponuje dodatkowo stworzenie modelu sieci *„zdrowej"*, a następnie monitoring ilości i sposobu (wykorzystywane w komunikacji porty TCP) przesyłania w niej danych.

Nietrudno wyobrazić sobie sieci komputerowe, w których wszelka transmisja powinna być obligatoryjnie utajniona (bankowość, agencje rządowe, bezpieczeństwo publiczne, bazy danych medycznych i biometrycznych). Typowe sieci domowe nie powinny generować znacznych ilości ruchu rozproszonego i szyfrowanego (działanie klienta sieci *bitTorrent* przez okres 12 godzin powoduje rozesłanie 18,45 GB danych na 72 288 różnych adresów IP w oparciu o protokół SSH – pomiar własny). Przesył danych jawnych/tajnych przy wykorzystaniu protokołów i/lub portów nieprzypisanych do tego celu przez organizację IANA (*Internet Assigned Number Authority*) [5] stanowi kolejny typ anomalii, na który należy zwrócić uwagę. Pojawienie się w systemie informacji o charakterze dokładnie przeciwnym do obowiązującego oznacza odstępstwo od narzuconego modelu, czyli niebezpieczeństwo.

W literaturze krajowej próżno szukać publikacji traktujących o detekcji danych zaszyfrowanych na potrzeby jakiejkolwiek dziedziny wiedzy. Sytuacja przedstawia się lepiej w przypadku literatury zagranicznej, gdzie wspomnieć należy dokumenty NIST (*National Institute of Standards and Technology*). Opisują one sposoby testowania przypadkowości algorytmu szyfrującego [6].

Celem prac było stworzenie analizatora ruchu sieciowego, którego zadaniem byłoby informowanie, że z dużym prawdopodobieństwem doszło w monitorowanej sieci komputerowej do zachowania nietypowego (przesył informacji zaszyfrowanej/jawnej).

Nauką podobną do kryptografii jest *steganografia*, zajmująca się ukrywaniem informacji istotnej w postaci jawnej lub niejawnej w obrębie innych danych (np. ukrycie tekstu w obrębie obrazu). Metody wykorzystywane w *steganoanalizie* przedstawiono pokrótce w [7]. Przykładowe realizacje detektorów steganografii znaleźć można w [8]. Znacznie większa liczba publikacji odnośnie teorii i sposobu realizacji detektorów steganografii dodatkowo motywowała w realizacji detektora informacji zaszyfrowanej.

2. Metody wykrycia danych niejawnych

W poprzedniej pracy [9] zaproponowano, by wyekstrahować dane zaszyfrowane z ogółu przechwyconych, wykorzystując do tego sformułowaną przez C. E. Shannona definicję informacji idealnie tajnej [9, 10]. Wyniki testów potwierdziły, że opierając algorytm detekcji o jedną z czternastu zaproponowanych wielkości statystycznych, uzyskuje się sprawność z zakresu 80,61% do 92,20% (w zależności od metody).

Istota algorytmu jest prosta. Dla badanej porcji danych obliczana jest wartość danego parametru statystycznego, a następnie porównywana z wartością wzorcową. Jeśli wartość obliczona mieści się w zakresie tolerancji, dane uznaje się za zaszyfrowane. W przeciwnym wypadku zostają one oznaczone jako jawne.

Głównymi zaletami metody, predysponującymi do implementacji w układzie FPGA, są:
- prostota algorytmu wykorzystującego jedynie nieskomplikowane operacje i przekształcenia arytmetyczno-logiczne na danych,
- interpretacja danych wejściowych, jako ośmiobitowe liczby całkowite bez znaku (*uint8*),
- zero-jedynkowy charakter odpowiedzi układu odnośnie typu analizowanych danych.

O użyciu układu FPGA do budowy *sniffera* (analizatora danych sieciowych nieingerującego w ich strukturę) zadecydowały:
- możliwość realizacji całego toru cyfrowego przetwarzania danych w pojedynczym układzie scalonym (detektor, kontroler MAC, kontroler wyświetlacza/zapisu do karty pamięci),
- niewielkie rozmiary i pobór mocy w porównaniu z rozwiązaniem programowym uruchomionym na komputerze stacjonarnym lub przenośnym.

3. Detektor II rzędu

Detektor danych zaszyfrowanych działający w oparciu o pojedynczą wielkość statystyczną został nazwany klasyfikatorem I rzędu. Jak już wspomniano wcześniej, sprawność takiego rozwiązania nie przekroczyła podczas testów 93%.

Stworzony klasyfikator II rzędu łączy w sobie trzy – działające niezależnie – klasyfikatory I rzędu. Każdy z klasyfikatorów składowych określa, czy przechwycone dane są zaszyfrowane, czy też nie. Odzwierciedlone jest to poprzez ustawienie flagi. Końcową decyzję odnośnie natury danych podejmuje układ arbitra. Decyzja zapada w oparciu o ilość podniesionych flag.

Na etapie teoretycznym stworzono dwa klasyfikatory II rzędu. Pierwszy miał za zadanie maksymalizację efektywności poprawnego rozróżnienia danych. Drugi miał zapewnić minimalizację użytych zasobów logicznych. Tab. 1 zawiera podsumowanie wyników uzyskanych dla obu podejść. Efektywność algorytmu określono w oparciu o ten sam zbiór danych testowych co w [9].

Tab. 1. Efektywność klasyfikatora II rzędu w detekcji informacji szyfrowanej
Tab. 1. Effectiveness of the 2nd order classifier in encrypted data detection

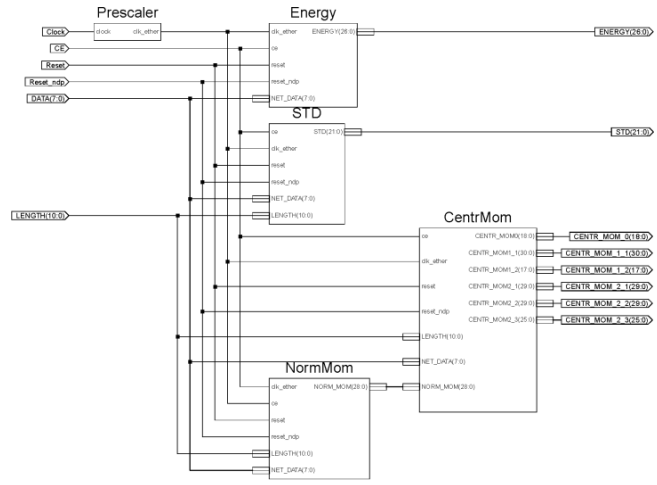
Wielkości składowe	Efektywność [%]
Energia, Odchylenie standardowe zmodyfikowane, Momenty centralne od 0. do 3.	94,78
Zmodyfikowana wartość średnia, Zmodyfikowany histogram, Momenty centralne od 0. do 3.	93,42

Różnica pomiędzy rozwiązaniami nie jest duża i wynosi mniej niż 1,5 punktu procentowego. W stosunku do najlepszego rozwiązania klasyfikatora I rzędu różnica efektywności wynosi 2,6%.

Zadecydowano, że sprzętowej implementacji poddany zostanie klasyfikator wydajniejszy pod względem poprawności detekcji. Jego schemat blokowy przedstawiono na rys. 1.

W wyborze algorytmów składowych kierowano się następującymi przesłankami:
- maksymalizacją prawidłowego rozróżnienia danych jawnych (*zmodyfikowane odchylenie standardowe*),
- maksymalizacją prawidłowego rozróżnienia danych (*momenty centralne*),
- poprawną klasyfikacją danych błędnie zaklasyfikowanych przez dwa powyższe (*energia*).

Takie rozwiązanie pozwoliło uzyskać sprawność lepszą niż dla każdej z metod osobno, a dodatkowo uzyskać zdolność prawidłowej klasyfikacji danych zaszyfrowanych na poziomie 99% (algorytm częściej klasyfikuje dane jawne, jako tajne niż na odwrót).



Rys. 1. Schemat blokowy zaimplementowanego klasyfikatora II rzędu
Fig. 1. Block diagram of the implemented encrypted data 2nd order classifier

Podobnie jak w [9] należy podkreślić istnienie granicy stosowalności klasyfikatora. Wynika ona ze spadku sprawności rozróżniania danych poniżej 50% wraz ze spadkiem długości ramki danych (wzrastają różnice charakteru analizowanej informacji od danych losowych o rozkładzie płaskim). Dolną granicę stosowalności stwierdza się na poziomie 80 – 100 bajtów (w zależności od metod składowych użytych w klasyfikatorze II rzędu).

4. Wyniki implementacji

Klasyfikator II rzędu zaproponowany w poprzednim rozdziale (tab. 1) okazał się zbyt duży (pod względem potrzebnych zasobów logicznych), by możliwą była jego implementacja w układzie xc3s1600e firmy Xilinx. Zaimplementowany system pozbawiony został najbardziej zajmującej zasoby części. Służyła ona do obliczenia momentu centralnego trzeciego rzędu.

Dla układu wykonano operację *syntezy* i *implementacji* przy użyciu narzędzia ISE 12.4 (M.81d) firmy Xilinx. Obiektem implementacji był układ xc3s1600 w obudowie fg320. *Speed grade* układu dysponowanego wynosił -4. Syntezę i implementację przeprowadzono w oparciu o strategię *„Timing Performance”* z wybraną opcją *„Performance with Physical Synthesis”*. Na schemacie blokowy z rys. 1. wyróżnić można bloki:
- dzielnika zewnętrznego sygnału zegarowego 50 MHz (przez 2, do wymaganych 25 MHz),
- klasyfikatora I rzędu opartego o parametr *energia*,
- klasyfikatora I rzędu opartego o parametr *zmodyfikowanego odchylenia standardowego*,
- kalkulacji *momentu normalnego pierwszego rzędu* (na potrzeby momentów centralnych),
- klasyfikatora I rzędu opartego o parametr *momenty centralne od zerowego do drugiego*.

Na rys. 1 nie zostały wyszczególnione następujące bloki, wchodzące w skład zaimplementowane detektora:
- *pipeline* dla odchylenia standardowego zmodyfikowanego, energii i danych sieciowych (konieczny, by zapewnić pojawienie się na wejściu arbitra wartości tych wielkości w tej samej chwili czasowej, co wartości obliczonych momentów centralnych) o długości 1460 (maksymalna długość ramki ethernetowej),
- arbiter.

Bloków nie uwzględniono, by zwiększyć czytelność rysunku. Mają one wkład w zestawienie dotyczące zajętości zasobów lo-

gicznych. W trakcie implementacji użyto dodatkowo bloku obsługi wyświetlacza LCD, co pozwoliło na weryfikację wyników pracy urządzenia (bloku tego nie uwzględniono na schemacie blokowym oraz w wynikach dotyczących implementacji).

Tab. 2 w sposób zbiorczy przedstawia zajętość dostępnych zasobów sprzętowych przez klasyfikator II rzędu.

Tab. 2. Wyniki syntezy i implementacji detektora w układzie xc3s1600fg320-4
Tab. 2. Synthesis and implementation results for the xc3s1600fg320-4

Zasób	Liczba wykorzystanych	Procent ogółu dostępnych
Slices Flip Flops	1 322	4
4 input LUTs	19 719	66
Zajęte Slices	10 673	72
Mnożarki 18x18	27	75

Maksymalna częstotliwość pracy układu nie pozwala na jego użycie w sieciach Gigabit Ethernet, jednak dopuszcza współpracę z sieciami o przepustowościach na poziomie 10 i 100 Mbit/s.

5. Weryfikacja poprawności pracy układu

Wykorzystując darmowy, programowy analizator danych sieciowych Wireshark [11] oraz 113 ramek sieciowych wybranych losowo ze zbioru ok. 10 000 przechwyconych, dokonano weryfikacji poprawności pracy zaprojektowanego klasyfikatora II rzędu.

Wireshark umożliwia w bardzo łatwy sposób identyfikację protokołu przechwyconej ramki oraz podgląd jej zawartości. Wykorzystując tę informację jako wzorcową, dokonano jej porównania z wynikami wygenerowanymi przez omawiany klasyfikator. Zbiór testowy wygenerowany został w ramach normalnego ruchu sieciowego (działanie komunikatora Skype, otworzenie połączenia ssh, przesłanie pliku jpeg, połączenie ze zdalnym serwerem, itp.) Ze zbioru tego wybrano jedynie około setki ramek ethernetowych, a następnie z ich pomocą określono efektywność zaprojektowanego klasyfikatora sprzętowego. Spośród 113 pakietów, 56 przenosiło dane jawne. Wyszczególniono następujące protokoły – HTTP, IP, ICMP, IRC, UDP, DNS, ESP (szyfrowanie IPsec ver. 4 wg [12]), KERBEROS (szyfrowanie RC4), SSL (wersje 2 i 3).

Wynik zebrano w tab. 3.

Tab. 3. Efektywność sprzętowego detektora informacji szyfrowanej
Tab. 3. Effectiveness of the hardware encrypted data detector

Wielkości składowe	Efektywność [%]
Energia, Odchylenie standardowe zmodyfikowane, Momenty centralne od 0. do 2.	94,69

6. Wnioski

Rozwój Internetu, aplikacji sieciowych i sprzętu komputerowego na przestrzeni ostatnich kilkunastu lat sprawił, że niemożliwym stał się osobisty nadzór nad administrowaną siecią bez pomocy wyspecjalizowanych narzędzi. Nie można jednak z drugiej strony zaprzestać zarządzania lokalnymi i globalnymi sieciami komputerowymi. Nie tylko z powodów technicznych (np. zapewnienie odpowiedniej jakości połączenia), ale również bezpieczeństwa (np. zabezpieczenie przed piractwem komputerowym, kradzieżą danych, działaniami terrorystycznymi). Zaprezentowany klasyfikator danych sieciowych może zostać użyty do zapewnienia tego drugiego. Jego rolą jest zasygnalizowanie administratorowi o pojawiających się w sieci nieprawidłowościach i odstępstwach od modelowego zachowania.

Głównymi zaletami rozwiązania są jego niewielkie rozmiary i zapotrzebowanie na energię elektryczną w porównaniu z komputerem klasy PC z zainstalowanym odpowiednim oprogramowaniem gwarantującym równoważną funkcjonalność. Efektywność zaimplementowanego klasyfikatora jest o 0,09% mniejsza niż dla klasyfikatora II rzędu uruchomionego na komputerze klasy PC (kod programu napisany w programie MATLAB). Różnica w efektywności jest więc pomijalnie mała. Należy zwrócić uwagę na bardzo ważny fakt – komputer PC wykonuje obliczenia na danych zmiennoprzecinkowych typu *double*, podczas gdy zaprojektowany układ operuje wyłącznie na danych typu *integer*.

Główną wadą układu jest znaczna zajętość dostępnych zasobów logicznych największego układu FPGA z rodziny Spartan 3E. Tak duża zajętość sprawia, że na etapie routowania połączeń wewnętrznych dochodzi do ograniczenia maksymalnej częstotliwości pracy układu. Brak również możliwości współpracy detektora z sieciami Gigabit Ethernet. Dodatkowo układ jest ograniczony do współpracy tylko z typowymi ramkami ethernetowymi o długości pola danych nieprzekraczającej 1460 oktetów (brak obsługi ramek typu *jumbo*). Wadą urządzenia na poziomie koncepcji jest brak możliwości rozróżnienia danych jawnych o charakterze zbliżonym do losowego (np. obrazy binarne lub w odcieniach szarości) oraz czysto pseudolosowym (np. przesyłana w ramce sekwencja synchronizująca urządzenia sieciowe wytworzona w rejestrze typu LSFR [13]) od danych utajnionych.

Wartym rozważenia jest wykorzystanie w przyszłości dodatkowych informacji w celu jeszcze lepszej klasyfikacji danych. Proponuje się wykorzystanie nie tylko wiedzy na temat wartości poszczególnych parametrów statystycznych dla informacji niesionej w polu danych przechwyconej ramki, ale również numerów portów TCP nadawcy i odbiorcy (dany protokół ze stosu TCP/IP korzysta *zazwyczaj* z pewnego ściśle przyporządkowanego mu numeru portu). Również integracja w ramach pojedynczego układu FPGA kontrolera MAC pozwoli zmniejszyć rozmiary końcowego urządzenia. Żadne z powyższych nie zostanie jednak spełnione bez wcześniejszej optymalizacji kodu VHDL detektora, zmiany układu FPGA na wyposażony w większą liczbę ekwiwalentnych bramek logicznych lub wykorzystania innego klasyfikatora drugiego rzędu (np. rozwiązanie drugie z tab. 2).

Praca finansowana ze środków Narodowego Centrum Badań i Rozwoju w ramach projektu SYNAT.

7. Literatura

[1] Xilinx: (2008, 20 czerwca) Spartan–3E FPGA Starter Kit Board User Guide. [Online]. http://www.xilinx.com/support/documentation/boards_and_kits/ug230.pdf
[2] Xilinx: (2009, 26 sierpnia) Spartan–3E FPGA Family: Data Sheet. [Online]. http://www.xilinx.com/support/documentation/data_sheets/ds312.pdf
[3] Mika S.: Koncepcja hybrydowego systemu detekcji robaków sieciowych wykorzystującego metody eksploracji danych, Metody Informatyki Stosowanej, vol. 23, nr 2/2010, str. 105-115, 2010.
[4] Cheema F.M., Akram A. i Iqbal Z.: Comparative Evaluation of Header vs. Payload based Network Anomaly Detectors," Proceedings of the World congress on Engineering, vol. 1 –WCE 2009, 2009.
[5] IANA: (2010, 24 grudnia) Port Numbers. [Online]. http://www.iana.org/assigments/port-numbers
[6] Rukhin A. i in.: A Statistical test suite for random and pseudorandom number generators for cryptographic applications, NIST Special Publication 800-22, 2001.
[7] Składnikiewicz M.: Entropia – pomiar i zastosowanie, nr 3, Software Wydawnictwo, 2008. [Online]. http://www.hakin9.org
[8] Damiani E. i in.: Signal Processing for Image Enhancement and Multimedia Processing, str. 268–278, ed. 1, Springer, 2007.
[9] Gancarczyk G., Dąbrowska-Boruch A., Wiatr K.: Efektywność parametrów statystycznych w detekcji informacji szyfrowanej, Pomiary Automatyka Kontrola, vol. 56, nr 10/2010, str. 1137–1143, 2010.
[10] Shannon C.E.: Communication Theory of Secrecy Systems, Bell System Technical Journal, vol. 28, no. 4, pp. 656-715, 1949.
[11] Wireshark Foundation. (2011, luty) Wireshark. [Online]. http://www.wireshark.org
[12] Inter-Corporate Computer & Network Services, Inc. (2011, luty) RFC 4305. [Online]. http://rfc4305.openrfc.org
[13] Golomb S.W.: Shift Register Sequences, Aegean Park Press, 1982.

otrzymano / received: 13.05.2011
przyjęto do druku / accepted: 04.07.2011 *artykuł recenzowany*