STUDIA INFORMATICA 2003
Volume 24 Number 4 (56)

Katarzyna TRYBICKA-FRANCIK
Politechnika Śląska, Instytut Informatyki

WIZUALNA METODA OCENY JAKOŚCI SZYFRÓW[1]

Streszczenie. Artykuł jest podsumowaniem końcowego etapu prac nad wykorzystaniem wizualizacji algorytmów w kryptoanalizie [2-5]. Zaproponowana w nim metoda jest mechanizmem uzupełniającym dla klasycznych technik oceny jakości algorytmów kryptograficznych. Na potrzeby tej publikacji wykorzystano algorytm DES (Data Encryption Standard) [1] z uwagi na jego popularność i bogaty materiał kryptoanalityczny.

Słowa kluczowe: kryptoanaliza, wizualizacja

A VISUAL METHOD OF ESTIMATING THE QUALITY OF CIPHERS

Summary. This article is a summary of the final stage of the research on the application of software visualization in cryptanalysis [2-5]. The method proposed in the paper is supplemental to the classical techniques of cipher quality estimating. For the needs of this publication the DES algorithm (Data Encryption Standard) [1] has been used because of its popularity and rich cryptanalytical material.

Keywords: cryptanalysis, visualization

1. Wstęp

Problem oceny jakości algorytmów kryptograficznych jest zagadnieniem złożonym. Z dotychczasowych prac wynika, że w celu przeprowadzenia wiarygodnej oceny jakości szyfru konieczne, ale nie wystarczające, jest zbadanie następujących właściwości:

- właściwość komplementarności,
- efekt lawiny,

[1]Praca finansowana z funduszu Badań Statutowych Instytutu Informatyki w roku 2003.

- klucze słabe,
- klucze półsłabe,
- punkty stałe,
- kolizja kluczy,
- podatność na kryptoanalizę różnicową,
- podatność na kryptoanalizę liniową,
- podatność na inne znane typy ataków.

Aby tego dokonać konieczne jest zrozumienie, w jaki sposób działa algorytm. Formalny opis, choć precyzyjny, może być dla człowieka trudny do zrozumienia. Omówienie działania szyfru na wyższym poziomie abstrakcji w języku naturalnym jest bardziej zrozumiałe dla obserwatora, ale niesie ze sobą niebezpieczeństwo pominięcia istotnych informacji, które mogą mieć niebagatelne znaczenie w procesie oceny jakości zaproponowanego rozwiązania.

Dlatego też 2 stycznia 1997 r. amerykański instytut standaryzacji NIST (ang. National Institute of Standards and Technology) ogłosił otwarty konkurs na algorytm szyfrowania danych, przeznaczony do szyfrowania nieutajnionych danych rządowych i zastosowań komercyjnych AES (ang. Advanced Encryption Standard). Algorytmy mogły zgłaszać dowolne instytucje jak i osoby prywatne, ale w dokumentacji oprócz opisu szyfru musiała znaleźć się jego działająca implementacja. Bardzo często, we wzmiankowanym konkursie, opisy wzbogacone były rysunkami w myśl zasady, że „obraz jest wart tysiąca słów".

Statyczny obraz nie pozwala jednak zaobserwować przepływu danych i zmian ich wartości, a są to informacje bardzo istotne dla prawidłowego zrozumienia i zinterpretowania działania algorytmu. Dynamiczne wizualizacje uwidaczniają cechy algorytmu w oderwaniu od jego konkretnej realizacji. Mogą to być wizualizacje opisu algorytmu dokonanego na wysokim, abstrakcyjnym poziomie.

Artykuł przedstawia wizualną metodę oceny jakości szyfrów, która może być traktowana jako istotny mechanizm uzupełniający dla klasycznych technik kryptoanalitycznych.

2. Analiza

By zrozumieć działanie algorytmu, konieczne jest wychwycenie związków przyczynowo - skutkowych, zachodzących pomiędzy kolejnymi chwilowymi stanami algorytmu. Wizualizacja nie powinna więc ograniczać się tylko do przedstawienia stanów, które program osiąga podczas wykonywania, ale także ilustrować przejścia pomiędzy nimi. Konieczne jest też podjęcie decyzji, które z obserwowanych elementów algorytmu są istotne, a z których należy zrezygnować,

by nadmiar wyświetlanych informacji nie komplikował niepotrzebnie projekcji, utrudniając jej zrozumienie i wychwycenie informacji istotnych. Ważne są więc abstrakcje.

O abstrakcji w konkretnej realizacji algorytmów świadczą następujące cechy:

- **Abstrahowanie** od tych elementów programu lub algorytmu, które nie są istotne dla zrozumienia algorytmu. Mogą nimi być pomocnicze struktury danych, wprowadzone w konkretnej implementacji.
- **Zmiana realizacji** algorytmu. Bywa, że faktycznie zastosowana implementacja nie sprzyja zrozumieniu algorytmu. Przykładem niech będą implementacje szyfrów ukierunkowane na jak najlepszą efektywność czasową, wykorzystujące właściwości sprzętu, na którym są uruchamiane.
- Dodanie **zawartości intencyjnej**, to jest elementów nie występujących w programie poddawanym wizualizacji. Najłatwiejszym sposobem jest dobór odpowiedniej reprezentacji graficznej, stanowiącej trafną metaforę przedstawianych danych.

Pracę z algorytmem należy rozpocząć od zapoznania się z jego opisem, co w przypadku udostępnionej implementacji pozwoli zorientować się w strukturze programu. Ta wiedza w znacznym stopniu ułatwi wprowadzenie niezbędnych elementów sterujących do programów wizualizacyjnych. Założono, że algorytm kryptograficzny dostępny jest w postaci programu. Jeżeli tak nie jest, to na potrzeby dalszej analizy konieczne będzie zaimplementowanie szyfru. W trakcie kodowania istotne jest zachowanie dla potrzeb wizualizacji sensu algorytmu. Oznacza to, że abstrahujemy od niuansów technologicznych, które są bardzo istotne, gdy mamy do czynienia z gotowym produktem. Zła implementacja nawet najbezpieczniejszego algorytmu może sprawić, że powstały na tej bazie system kryptograficzny jest bezużyteczny.

W pierwszym kroku należy zapoznać się z ogólną strukturą szyfru, na wysokim poziomie abstrakcji w oderwaniu od realizacji poszczególnych funkcji i struktur danych. Na tym etapie możemy posiłkować się statycznymi obrazami, bardzo często dostarczanymi wraz z opisem algorytmu.

Za przykład niech nam posłuży schemat blokowy algorytmu DES [1]. Jak nietrudno zauważyć, algorytm zbudowany jest z dwu wątków. Pierwszy to ciąg operacji wykonywanych na tekście jawnym, w rezultacie których uzyskujemy szyfrogram. Nazwijmy go wątkiem szyfrującym. Składa się on z 16 cykli, w których wykonywana jest taka sama funkcja szyfrująca F. Funkcja szyfrująca F wymaga użycia klucza.

Klucze, dalej nazywane podkluczami, dla kolejnych cykli generowane są w osobnym wątku. Podobnie jak w poprzednim przypadku, przyjmijmy dla uproszczenia, że jest to wątek klucza. Struktura podkluczy w żaden sposób nie zależy od rezultatu operacji wykonywanych w wątku szyfrującym.

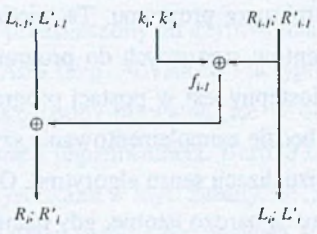
Wynika z tego, że część obserwacji będzie można wykonać osobno dla wątków szyfrowania lub klucza.

3. Właściwość komplementarności

Pierwszą właściwością, jaka znalazła się na liście cech, które należy zbadać, jest komplementarność. Ciąg komplementarny do danego to taki, w którym zastąpiono wszystkie zera jedynkami, a jedynki zerami.

Definicja: Jeżeli K' jest kluczem komplementarnym do K i odpowiednio M' jest tekstem jawnym komplementarnym do M i zachodzi zależność $C = E_K(M)$ oraz $C' = E_{K'}(M')$, gdzie C' to szyfrogram komplementarny do C, to znaczy, że szyfr spełnia **warunek komplementarności.**

Najłatwiej sprawdzić czy szyfr jest komplementarny wykonując serię testów. Jeżeli dane testowe zostaną przygotowane prawidłowo i wykonamy odpowiednio dużo prób, to istnieje bardzo małe prawdopodobieństwo, że tak uzyskana odpowiedź będzie fałszywa. Takie podejście do problemu nie pozwoli nam jednak ustalić, dlaczego algorytm wykazuje, lub nie, badaną właściwość.



Rys. 1. Pojedyncza runda DES
Fig. 1. Single DES round

Algorytm DES jest komplementarny. Przyjrzyjmy się pojedynczej rundzie DES [1]. Prawa część ciągu wejściowego R_{i-1} przekazywana jest do funkcji F. W funkcji F podciąg R_{i-1} poddawany jest permutacji rozszerzającej E i następnie sumowany modulo 2 z podkluczem k_i, generowanym na podstawie klucza głównego K. W rezultacie otrzymujemy ciąg wchodzący do *s-bloków*. Zauważmy, że permutacja E nie wpływa w żaden sposób na komplementarność dwu ciągów R_{i-1} i R'_{i-1} oraz że sposób generacji podkluczy też nie zmienia relacji komplementarności między k_i i k'_i. Możemy więc uprościć schemat pojedynczego cyklu DES rezygnując z tych operacji. Z właściwości operatora $\oplus$ wynika, że $R_{i-1} \oplus k_i = R'_{i-1} \oplus k'_i$. Oznacza to, że kombinacja bitów pojawiających się na wejściu do *s-bloków* jest taka sama w obu przypadkach, zatem i wyjścia z *s-bloków* generują takie same rezultaty. Tak więc i to przekształcenie możemy pominąć wraz z permutacją P, która zmienia jedynie uszeregowanie bitów w ciągu. Rezultat działania funkcji F jest ponownie sumowany modulo 2, ale tym razem z lewym podciągiem tekstu wejściowego. Z właściwości operatora $\oplus$ mamy $R_i = L_i \oplus f_{i-1}$ oraz, w przypadku komplementarnym, $R'_i = L'_i \oplus f_{i-1}$, gdzie f_{i-1} to rezultat pracy funkcji F. Rysunek 1 przedstawia uproszczony, ze względu na komplementarność, schemat pojedynczej rundy DES. Fakt, że algorytm DES jest komplementarny, ogranicza przy ataku brutalnym liczbę prób, które trzeba wykonać z 2^{56} do 2^{55}, czyli o połowę.

Dla tej właściwości przeprowadzenie analizy jest dość proste, nawet w przypadku gdy nie zastosujemy do tego celu wizualizacji, choć nie da się zaprzeczyć faktowi, że jej zastosowanie znacznie upraszcza sprawę. W przypadku charakterystyk lawinowych nie jest już to takie proste.

4. Charakterystyki lawinowe

Nadmiarowość języka naturalnego sprawia, że mamy do czynienia z przesyłem dodatkowej informacji, która może być użyta do skutecznej analizy statystycznej tekstu. By udaremnić taką analizę stosuje się dwie techniki: konfuzję (poplątanie) i dyfuzję (uśrednienie rozproszenia).

Celem konfuzji jest uczynienie związku pomiędzy kluczem i szyfrogramem tak złożonym, jak tylko to możliwe. Innymi słowy kryptoanalityk nie powinien móc uzyskać użytecznej informacji o kluczu na podstawie badań statystycznych szyfrogramu.

Druga z nich ma na celu zatarcie statystycznych cech tekstu jawnego w kryptogramie. Służą do tego dwie techniki:

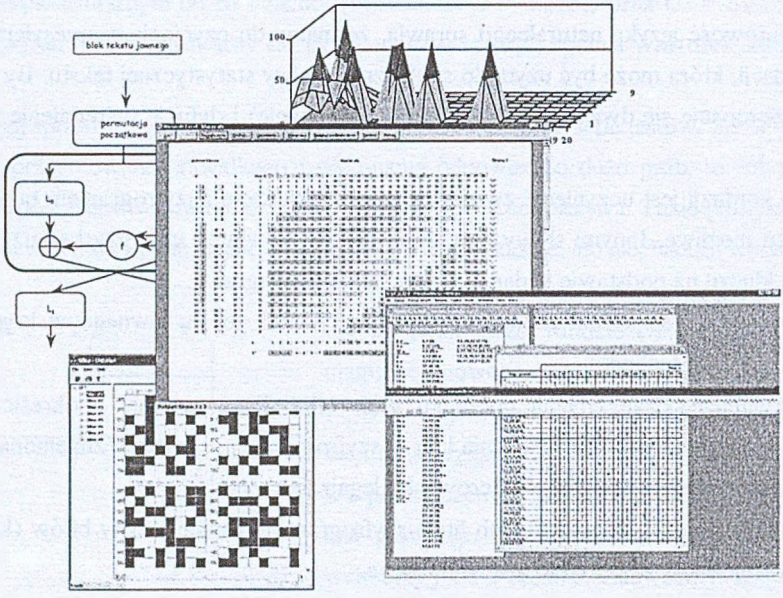
- Permutacje, czyli zmiana porządku bitów (lub liter) zgodnie z określoną regułą. Wprawdzie częstość wystąpienia liter w szyfrogramie nie zostanie zmieniona, ale statystyka wystąpień par i trójek literowych ulega zaburzeniu.
- Uzależnienie każdego bitu (lub liter) szyfrogramu od takiej liczby bitów (liter) tekstu jawnego i klucza jak tylko jest to możliwe.

Reasumując, istotną cechą algorytmu jest to, aby na podstawie szyfrogramu nie dało się wnioskować o postaci tekstu jawnego ani o kluczu. Jednym z parametrów badanych w tym celu jest wpływ zmiany jednego bitu w danych wejściowych (tekście jawnym, kluczu) na stan wyjścia. Charakterystyki lawinowe opisują te relacje. Przyjmuje się, że dobry rozkład to taki, w którym zmiana jednego bitu na wejściu pociąga za sobą zmianę połowy bitów na wyjściu.

Dla oceny jakości szyfru DES została stworzona aplikacja **Wizualizacja-DES**, która pozwala obserwować pewne wybrane cechy algorytmu. Jedną z nich jest efekt lawinowy. Przyglądając się pojedynczemu cyklowi DES możemy prześledzić rozprzestrzenianie się zmian w wyniku przesunięć i przestawień, jakie mają miejsce w algorytmie. W pojedynczym cyklu wykorzystywane są aż trzy przekształcenia typu permutacja (permutacje E i P w wątku szyfrującym oraz w wątku klucza *PS2*) i podstawienia zdefiniowane w *s-blokach*. Dzięki graficznej reprezentacji można łatwo określić wagi tych przekształceń i stwierdzić, które z nich są najistotniejsze dla funkcjonowania szyfru. DES jest tak zbudowany, by osiągnąć w jak najkrótszym czasie stan, w którym każdy bit szyfrogramu zależy od każdego bitu tekstu jawnego i

każdego bitu klucza. W pesymistycznym przypadku stan taki osiągniemy po siódmym cyklu algorytmu.

Niewątpliwie strukturą wzbudzającą najwięcej emocji są *S-bloki*. Najefektywniejszy atak skierowany przeciwko DES wykorzystuje pewne specyficzne właściwości statystyczne tej struktury [6]. W tym miejscu skupimy się jednak tylko nad tym, jak zmiana bitu na wejściu *s-bloku* wpływa na stan jego wyjścia.



Rys. 2. „Zrzuty" okien wizualizacji DES
Fig. 2. Screenshots of DES visualization

W przypadku *s-bloków* stosowana jest notacja tablicowa (wprowadzona przez IBM), niestety nie jest ona zbyt użyteczna, gdy chcemy badać relacje między danymi wejściowymi i wyjściowymi. Jeżeli wprowadzimy drobną modyfikację i zastosujemy notację binarną i takie uszeregowanie kolumn i wierszy, by pozycje kodowe różniły się od siebie tylko jednym bitem, to okaże się, że:

- jeśli dwie sekwencje wejściowe s-bloku różnią się dokładnie o 1 bit, to sekwencja wyjściowa różni się co najmniej o dwa bity,
- jeśli dwie sekwencje wejściowe s-bloku różnią się o 2 początkowe bity i są identyczne na przynajmniej dwóch bitach, to sekwencje wyjściowe są różne.

Uzyskiwane rozkłady zer i jedynek na wyjściu DES należy zaliczyć do dobrych. Ten rodzaj testu nie jest jednak wystarczający. Trzeba sprawdzić czy taki dobry rozkład utrzymuje się dla całej dziedziny. Oczywiście sprawdzenie całej przestrzeni klucza i wiadomości jest niemożliwe, ale są podprzestrzenie w dziedzinie, które są szczególnie interesujące, tzn. kluczy

słabych, półsłabych i potencjalnie słabych [7-8]. Wzmiankowana aplikacja Wizualizacja-DES pozwala obserwować nie tylko przepływ danych, ale i analizować wyniki liczbowe testów, takie jak rozkłady częstotliwości funkcji będących liniowymi aproksymacjami *s-bloków*.

Na rys. 2 umieszczono „zrzuty" kilku ekranów z aplikacji Wizualizacja-DES.

5. Wizualizacyjna metoda oceny jakości szyfrów

Przedstawione w artykule dwa przykłady pokazują jedynie bardzo pobieżnie możliwości, jakie niesie ze sobą wizualizacja algorytmów jako narzędzie przy ocenie szyfrów. Wprawdzie zastosowanie wizualizacji wymaga niebanalnych przygotowań wstępnych, jednak korzyści płynące z faktu, że wizualizacje mogą operować różnymi poziomami abstrakcji oraz stosować specyficzne reprezentacje graficzne nawiązujące do obserwowanych realiów, są nie do przecenienia. Dobór środków prezentacji jest zależny od charakteru, ilości i organizacji danych opisujących problem. Wizualizacje są bardzo wygodnym narzędziem pozwalającym zrozumieć sposób działania algorytmu i pozwalają zaobserwować zdarzenia trudno uchwytne metodami analitycznymi oraz wskazują na sytuacje, które mogą być kluczowe dla metod analitycznych.

Przeprowadzając analizę kryptograficzną szyfru należy odpowiedzieć sobie na kilka pytań:

- Jak działa badany przez nas szyfr? Można w tym celu wykorzystać aplikacje wizualizacyjne. Rezultatem tych obserwacji powinno być: wyodrębnienie spójnych jednostek funkcjonalnych i określenie relacji, jakie zachodzą pomiędzy nimi (np. w przypadku DES wyodrębnienie dwu wątków szyfrowania i klucza).
- Do jakich celów algorytm będzie wykorzystywany? Pewne cechy szyfru mogą być nieistotne w sytuacji, gdy jest on wykorzystywany do utajniania informacji, ale ich znaczenie może wzrosnąć, gdy chcemy na bazie badanego algorytmu zbudować generator ciągów pseudolosowych lub jednokierunkową funkcję skrótu (np. fakt że DES posiada 256 tzw. kluczy potencjalnie słabych, w żaden sposób nie wpływa na oferowane przez szyfr bezpieczeństwo do czasu, gdy nie spróbujemy na jego bazie zbudować jednokierunkowej funkcji skrótu [7]).
- Które z wyznaczonych elementów można badać niezależnie od pozostałych i jakie ich cechy są ważne dla procesu krytpoanalizy? Innymi słowy abstrahujemy od tych elementów algorytmu, które nie są istotne dla badanej właściwości (np. wyznaczając klucze kolidujące dla DES nie musimy analizować wątku szyfru).
- Jakie są relacje pomiędzy tymi elementami i jakie przepływy danych występują pomiędzy nimi? Scalanie ma na celu spojrzenie na szyfr jako na całość. Podobnie jak w poprzednim przypadku należy abstrahować od przekształceń nieistotnych dla badanej ce-

chy (np. badając charakterystyki lawinowe mogliśmy pominąć permutacje początkowe w wątkach szyfru i klucza oraz końcową w wątku szyfru. Nieistotne też były odwzorowania w samych *s-blokach*, wystarczająca była informacja o tym, że różniące się jednym bitem wejścia *s-bloku* na wyjściu różnią się przynajmniej dwoma bitami).

- Jak zaprezentować wyniki? Na podstawie obserwacji przepływu danych podejmujemy decyzję, które z obszarów dziedziny poddać dokładniejszej analizie. Przygotowanie właściwych danych testowych, a potem analiza uzyskanych rezultatów są kluczowe dla kryptoanalizy. Duża liczba danych wynikowych może być dla naszej percepcji nic nie znaczącym zbiorem znaków. Ich graficzna reprezentacja zmienia ten stan rzeczy (już inne uszeregowanie danych może zmienić obraz całości, czego przykładem niech będzie reprezentacja *s-bloków*).
- Jak wykorzystać wyniki obserwacji w klasycznej (opartej na wywodzie matematycznym) kryptoanalizie? Wizualizacje mają być elementem wspierającym dla analizy kryptograficznej, dlatego obserwacje powinny być poparte rzetelnym dowodem analitycznym. Niestety, bardzo często jest tak, że choć domyślamy się, że pewna właściwość ma miejsce, to jednak przeprowadzenie dowodu nie jest takie oczywiste (np. bezpieczeństwo potrójnego DES wynika z faktu, że DES nie tworzy grup, co najpierw stwierdzono empirycznie, a potem udowodniono analitycznie).

LITERATURA

1. Data Encryption Standard. Federal Information Processing Standard (FIPS), Publication 46, National Bureau of Standards, U.S. Departament of Commerce, Washington D.C., January 1977.
2. Tybicka-Francik K.: Wykorzystanie wizualizacji algorytmów w kryptoanalizie. IX Konferencja Sieci Komputerowych, Studia Informatica, Vol. 23, No. 2B (49), Zakopane 2002, pp. 279-286.
3. Tybicka-Francik K.: Wykorzystanie wizualizacji algorytmów w procesie oceny jakości szyfrów. VIII Konferencja Sieci Komputerowych, Studia Informatica, Vol. 22, No. 2 (44), Krynica 2001.
4. Tybicka-Francik K.: Metody badania efektywności szyfrowania informacji. Konferencja Ciechocinek 2000, Zakład Poligraficzny Urzędu Statystycznego w Bydgoszczy.
5. Francik J., Tybicka-Francik K.: Zastosowanie wizualizacji algorytmów w kryptologii. VII Konferencja Sieci Komputerowych, Studia Informatica, Vol. 21, No. 1 (39), Zakopane 2000.

6. Matsui R. M..: Linear cryptanalysis method for DES cipher. Helleseth T. (ed.), Advances in Cryptology - EUROCRYPT'93. LNCS 765.
7. Moore J. H., Simmons G. J.: Cycle Structure of the DES with Weak and Semi-Weak Keys. Advance of Cryptology - Crypto'86 Proceedings, Berlin: Springer - Verlag, 1987, pp. 3-32.
8. Davies D. W.: Some Regular Properties of the DES. Advance of Cryptology: Proceedings of Crypto'82, Plenum Press, 1983, pp. 89-96.

Recenzent: Dr inż. Andrzej Białas

Wpłynęło do Redakcji 13 maja 2003 r.

Abstract

Two examples presented in the paper (section 4 & 5) show – very generally – the potential of software visualization as a tool for cipher estimation. Applying visualization requires non-trivial prerequisites and preparation. However the profit gained from the fact that the technique is capable to operate on various levels of abstraction and to apply specific graphical representation for each observed phenomenon – is difficult to overestimate. The choice of presentation means depends on type, quantity and organization of data being analyzed.

There are several questions to answer during the cryptanalysis:

- How the cipher operates? It is possible to apply visualization applets. As a result, coherent functional units should be defined as well as relations between them.
- What is the purpose of the algorithm? Some features may be of no importance if the algorithm is used to make the information confidential, but they may appear much more important if it is used to build a pseudo-random generator or a hash function.
- Which of the functional units mentioned above should be independently analyzed and which of their features are important for the cryptanalysis? The goal is to analyze them apart from other units or features which are not important is the context of the analysis.
- What are the relations between the units and what data flows occur between them? This is a kind of synthesis.
- How to present the results? Preparation of test data and the analysis of the results are very important, but the graphical representation applied could be crucial if we want to avoid accumulation of hardly readable information.

- How to apply the result of classical cryptanalysis, i.e. based on a mathematical argumentation? Visualization may accelerate the cryptanalysis, but should be supported with reliable mathematical background. Unfortunately, in many cases it is extremely difficult to obtain.

Adres

Katarzyna TRYBICKA-FRANCIK: Politechnika Śląska, Instytut Informatyki,
ul. Akademicka 16, 44-101 Gliwice, Polska, kasiat@zeus.polsl.gliwice.pl .