XIII Konferencja PLOUG
Kościelisko
Październik 2007

Przekształcenia szyfrujące – realizacje neuronowe, ocena jakości algorytmu szyfrującego

Piotr Kotlarz
Kazimierz Wielki University, Bydgoszcz

piotrk@ab.edu.pl

Zbigniew Kotulski
Institute of Fundamental Technological Research, PAS and Institute of Telecommunications, WUTe

piotrk@ukw.edu.pl

Abstrakt. Artykuł ten to kontynuacja rozważań prowadzonych na temat budowy neuronowego układu szyfrującego dającego możliwości realizacji różnych symetrycznych algorytmów szyfrujących. Tematyka ta była już prezentowana na konferencji PLOUG'06, wtedy prezentowane było dopiero pierwsze koncepcje i pomysły. Na konferencji PLOUG'07 chcemy zaprezentować wyniki badań eksperymentalnych oraz dyskusję bezpieczeństwa proponowanego rozwiązania. Jak zwykle postaramy się osadzić treść pracy, w tematyce bezpieczeństwa baz danych ze względu na charakter konferencji. Poniższy artykuł ma nieci bardziej niż zwykle na PLOUG charakter teoretyczny. Poczynione w nim, zostały pewne wprowadzenia pojęć podstawowych z teorii informacji, jednak tylko te, których rozumienie niezbędne będzie do przyswojenia prezentowanego zagadnienia. Zamieszczenie wiele szczegółów teoretycznych w pracy pozwoli podczas wystąpienia na konferencji przedstawić temat bardziej ogólnie i ciekawie.

Informacja o autorze. Piotr Kotlarz stopień magistra uzyskał w 2002 roku na Akademii Bydgoskiej, specjalność Technika Komputerowa. Od 2003 roku Asystent w Instytucie Mechaniki Środowiska i Informatyki Stosowanej UKW, gdzie kierował projektami z zakresu pogłębiania specjalizacji jednostki, bierze udział w badaniach naukowych oraz w pracy dydaktycznej. Wyróżniony nagrodą zespołową Rektora UKW II stopnia za badania naukowe. Od 2005 roku doktorant IPPT Polskiej Akademii Nauk.
Zainteresowania naukowe: kryptografia, bezpieczeństwo informatyczne, sztuczna inteligencja. Autor i współautor wielu prac naukowych publikowanych na konferencjach krajowych i międzynarodowych. Autor kilku referatów popularno-naukowych wygłoszonych na konferencjach branżowych. Autor oraz współredaktor dwóch pozycji książkowych.
W kręgach zainteresowań leży: nowoczesna technologia, biznes i ekonomia, historia nauki. Hobby: turystyka górska (czynna), gotowanie (kuchnia orientalna i polska), literatura popularno-naukowa.
Autor posiada również bogate doświadczenie poza akademickie w zakresie administrowania, projektowania i budowania sieci informatycznych.

1. Wstęp

W pracy tej prezentowane są kolejne etapy badań zastosowania sieci neuronowych jako programowalnych układów szyfrujących. Na konferencjach PLOUG'05 oraz PLOUG'06 zaprezentowane zostały ogólne założenia badań [15,16]. W naszych pracach wcześniejszych [12][13][14], proponowaliśmy przykłady realizacji przekształceń szyfrujących za pomocą sieci neuronowej. Ponadto w pracy prezentowanej między innymi na zeszłorocznej konferencji PLOUG'06 [16] przedstawiona została dyskusja konstrukcji ewentualnego protokołu kryptograficznego, który miałby umożliwić praktyczne wykorzystanie neuronowego układu szyfrującego. Kolejnym etapem prac, jak i głównym tematem tego referatu jest problematyka oceny jakości projektowanych algorytmów szyfrujących pod względem wymogów bezpieczeństwa. Ponieważ nasze badania dotyczą algorytmów z grupy symetrycznych to skupimy się na ocenie parametrów funkcji boolowskich jako kryterium oceny jakości przekształceń kryptograficznych. W naszych pracach zmierzamy do stworzenia układy neuronowego, który będzie mógł realizować różne algorytmy szyfrujące. Zmiana realizowanej funkcji szyfrującej ma się odbywać poprzez uczenie sieci, ważnym elementem takiego procesu jest, więc ocena tego procesu pod względem kryptograficznych kryteriów projektowych, właśnie o tych kryteriach traktuje ta praca.

2. Wprowadzenie

W rozdziale tym przedstawione są, w bardzo skrótowej i elementarnej formie, ogólne koncepcje zastosowania sieci neuronowych do realizacji funkcji kryptograficznych.

2.1. Inspiracje

Wiele popularnych protokołów kryptograficznych takich jak SSL i SSH, oraz oprogramowanie do realizacji usług kryptograficznych posiadają zaimplementowanych kilka algorytmów kryptograficznych. Przyczyn tego rodzaju rozwiązania jest kilka, jednak istotną z punktu widzenia, ciągle rozwijających się metod ataków jest możliwość wyboru z pośród kilku algorytmów szyfrujących. Dobrym przykładem jest tutaj algorytm DES, którego słabości zostały udowodnione. Szyfr ten jest wykorzystywany w realizacji wielu protokołów kryptograficznych, obecnie stosowanych, jednak większość z nich ma zaimplementowanych kilka innych szyfrów, dzięki czemu użytkownik może zrezygnować z wykorzystywania DES-a. Na przestrzeni kilku ostatnich lat pojawiły się publikacje na temat wykorzystania układów programowalnych w kryptografii do realizacji algorytmów kryptograficznych. Jedną z pierwszych prac była [17] gdzie zaproponowano realizację w oparciu o technologię układów programowalnych modułu akceleratora kryptograficznego realizującego symetryczny szyfr blokowy IDEA. Inną również wczesną pracą jest [18] gdzie autorzy wykorzystują procesor RipeRench, do realizacji algorytmów kryptograficznych. Wykorzystując rekonfigurowany procesor RipeRench zrealizowano między innymi takie algorytmy jak CRYPTON [19], czy RC6 [20]. Realizację szyfru CRYPTON w technologii układów rekonfigurowanych opisano również w pracy [21]. Szerzej na temat tego podejścia do realizacji szyfrów, oraz ogólnie na temat wykorzystania układów programowalnych w przetwarzaniu sygnałów napisano w [22].

Wyżej opisany stan rzeczy stał się inspiracją do poszukiwań możliwości wykorzystania sieci neuronowych (SN) jako układu realizującego algorytm szyfrujący, z możliwością zmiany go na inny, jeśli zaistnieje taka potrzeba. Z praktycznego punktu widzenia ważne jest to aby zmiana taka przeprowadzana była jak najmniejszym kosztem. Celem naszym jest skonstruowanie sieci neuronowej, która była by w stanie realizować różne algorytmy szyfrujące. Zmiana realizowanego algorytmu ma następować po przeprowadzonym w określonych warunkach procesie uczenia. Poniższy rysunek prezentuje w dużym uproszczeniu to do czego zmierzamy.

Rys. 1. Neuronowy układ szyfrujący – ogólna koncepcja

Koncentrując się na algorytmach szyfrujących z rodziny symetrycznych skupiamy się na problemie realizacji poprzez sieć neuronowa dwóch podstawowych przekształceń kryptograficznych: permutacji i podstawienia.

2.2. Permutacje za pomocą SN

Obecnie udało się nam opracować metody realizacji oraz uczenia sieci neuronowych, mających na celu realizację przekształcenia kryptograficznego, jakim jest permutacja. Ze względy na ograniczoną objętość tej pracy poniżej przedstawiona jest ogólna koncepcja realizacji permutacji dla prostego przykładu permutacji trzech bitów.

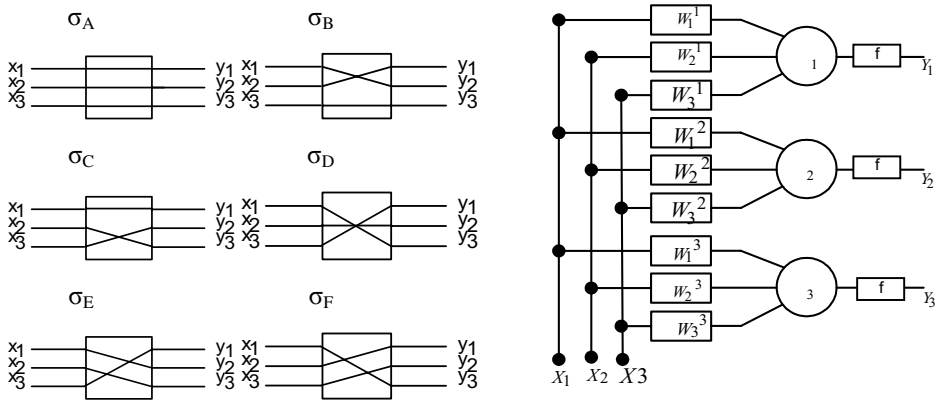
Założenia: przygotować sieć neuronową (czarną skrzynkę), którą będzie realizowała permutację, trzech bitów:

$$\sigma_A = \begin{pmatrix} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \end{pmatrix} \quad \sigma_B = \begin{pmatrix} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 1 & 3 \end{pmatrix}$$

$$\sigma_C = \begin{pmatrix} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 \\ 1 & 3 & 2 \end{pmatrix} \quad \sigma_D = \begin{pmatrix} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 \\ 3 & 2 & 1 \end{pmatrix}$$

$$\sigma_E = \begin{pmatrix} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \end{pmatrix} \quad \sigma_F = \begin{pmatrix} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 \\ 3 & 1 & 2 \end{pmatrix}$$

Sieć realizując powyższe permutacje przedstawiona jest poniżej.



Rys. 2. Sieć neuronowa do realizacji permutacji trzech bitów

Powyżej przedstawiona sieć składa się z neuronów, których działanie można opisać następująco:

$$\varphi_n = \sum w_i^n x_k$$

N – nr neuronu

i – nr wejścia neuronu

$$f = \begin{cases} 1, & \varphi > p \\ 0, & \varphi < p \end{cases}$$

k – nr wejścia sieci

l – nr wyjścia sieci

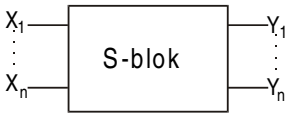
$$y_l = f(\varphi_n)$$

φ_n – potencjał membranowy

Przełączanie układu, pomiędzy realizacją poszczególnych permutacji odbywa się poprzez przeprowadzenie uczenia sieci z wykorzystaniem odpowiedniego zbioru uczącego.

2.3. Podstawienie za pomocą SN

Podstawienie jest drugim, po permutacji przekształceniem kryptograficznych wykorzystywanym poprzez algorytmy szyfrujące. Ich zadaniem jest realizowanie określonego podstawienia (operacji nie liniowej), czyli zamiany ciągu n bitów wejściowych na ciąg m bitów wyjściowych.

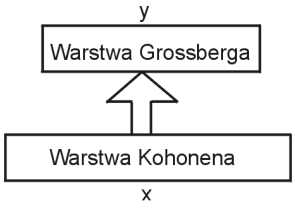


Rys. 3. S-blok

Rozpatrując s-blok jako funkcję boolowską realizuje ona odwzorowanie :

$S : \{0,1\}^n \rightarrow \{0,1\}^m$, gdzie m,n – wymiary s-bloku.

Tak więc s-boxy można traktować jako tablice dwu wymiarowe zawierające wartości binarne. Traktując s-blok jako macierz S o wymiarach m x n, to działanie s-bloku można porównać do realizacji zadania wybierania z tablicy. Punktem wyjścia do zastosowania sieci neuronowych do realizacji s-bloku jest podjęcie próby konstrukcji sieci neuronowej, która realizowałaby właśnie wybieranie z tablicy. Wygodnym sposobem na realizację s-bloku za pomocą SN jest skorzystanie ze znanej już koncepcji sieci CP [23]. Na potrzeby wykorzystania jej w neuronowym układzie szyfrującym konieczna jest pewna modyfikacja opis jej jednak wykracza poza zakres tej pracy.



Rys. 4. Sieć CP

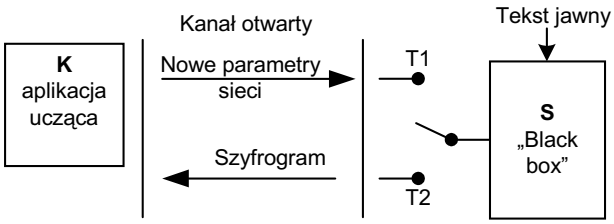
Warstwa Kohonena działa na zasadzie konkurencyjności. Wartość jeden pojawia się jedynie na wyjściu neuronu, którego wzbudzenie jest największe.

$$y_i = \begin{cases} 1 & dla \max(\varphi_i) \\ 0 & w \quad inny \quad wypadku \end{cases} \quad , \; gdzie \; \varphi_i = x * w^T$$

Warstwa ta wobec tego może realizować wybór wskazanej wartości z S-bloku poprzez współrzędne podane w wektorze wejściowym. Druga warstwa sieci jak się nietrudno domyśleć realizuje już podanie konkretnej wartości. Na wejście odpowiednio przygotowanej sieci trafia wektor wartości wejściowych, który zawiera współrzędne komórki s-bloku. Nauczona warstwa Kohonena zareaguje poprzez wyprowadzenie wartość „1” tylko na jednym wyjściu. W ten sposób neurony warstwy drugiej (Grossberga), które realizują podanie konkretnej wartości zostaną wybrane właśnie poprzez pojawienie się wartości „1” na wyjściu sieci Kohonena.

2.4. Aspekt praktyczny

Pewnym problemem do rozstrzygnięcia jest sposób wprowadzania zmian do realizowanego algorytmu przez sieć. Zmiana algorytmu – szyfru realizowanego przez sieć dokonuje się poprzez prowadzony w określonych warunkach proces uczenia. Przyjmując założenie, że skonstruowaną odpowiednio sieć traktujemy jako czarną skrzynkę umieszczoną na przykład gdzieś na odległym serwerze, w jakiś sposób konieczna jest możliwość zdalnego wpływania na realizowane przez nią przekształcenie. Trzymając się, więc ustalenia, że rozważany układ pracuje w układzie klient serwer, rolę klienta będzie pełnił „właściciel" czarnej skrzynki, który będzie chciał decydować jakie przekształcenie szyfrujące ma realizować owa czarna skrzynka. Z przeprowadzonych do tej pory eksperymentów [15,16] wynika, że najkorzystniejsze jest przeprowadzenie procesu uczenia po stronie klienta (K). Do serwerów natomiast rozsyłane są nowe wartości wag sieci ustalone w procesie uczenia. Koncepcję tą pokazuje poniższy rysunek:



Rys. 5. Tryby pracy neuronowego układu szyfrującego

Jak zostało to pokazane po przeprowadzeniu procesu uczenia po stronie klienta (K), do serwera (S) pracującego w trybie zmiana algorytmu (T1) przesyłane są nowe wartości wag sieci. Po zakończeniu tej czynności S przełączany jest w tryb T2 i od tego momentu realizuje inny algorytm szyfrujący. Serwer informacje na ten temat można znaleźć w poprzednich a naszych pracach [12-16].

3. Ocena jakości symetrycznych szyfrów blokowych

Na obecnym etapie naszych badań, gdy potrafimy już realizować za pomocą sieci neuronowych pojedyncze przekształcenia szyfrujące a nawet całe algorytmy szyfrujące. Stanęliśmy przed problemem oceny pod względem bezpieczeństwa kryptograficznego wprowadzanych zamian w procesie uczenia neuronowego układu szyfrującego. W kolejnych podrozdziałach przedstawione zostały wybrane własności funkcji boolowskich. Zamierzamy jest wykorzystywać do oceny kolejnych etapów procesu wprowadzania zmian w realizowanych przez nasz neuronowy układ szyfrujący algorytmów kryptograficznych.

3.1. Trochę klasyki

3.1.1. zasady Kerkhoff'a

Ogólne wytyczne, co do bezpieczeństwa i funkcjonalności algorytmów szyfrujących znaleźć można w prace z 1883 rok [6]. Autor sformułował tam „Zasady budowy dobrej maszyny szyfrującej" - zasady Kerkhoff'a.

1. System praktycznie nie do złamania
2. Utajniony jest klucz, a nie sama budowa maszyny
3. Klucz powinien być łatwy do zapamiętania
4. Kryptogram powinien być łatwy do przekazania
5. Maszyna szyfrująca powinna być łatwa do przenoszenia

6. System powinien być łatwy w użyciu

Szczególnie ważną zasadą z punktu widzenia dzisiejszych rozwiązań w zakresie bezpieczeństwa informacji jest zasad 2. Wskazuje ona na to, że funkcje szyfrujące *E* i deszyfrująca *F* powinny być ogólnie znane i dobrze udokumentowane. Wykonanie przekształcenia szyfrogramu *C* do postaci tekstu jawnego *M* powinno być proste pod warunkiem znajomości klucza *K*. Brak znajomości klucza *K*, a samo posiadanie szyfrogramu oraz funkcji deszyfrującej nie powinno dawać możliwość wyznaczenia tekstu jawnego *M*. Spełnienie tych postulatów bezpieczeństwa realizowane jest poprzez zastosowanie przekształceń szyfrujących jakimi są: przekształcenia liniowe (permutacje) oraz nieliniowe (podstawienia).

3.1.2. Wymogi bezpieczeństwa szyfrów według Shannona.

Dwie pierwsze własności bezpiecznego szyfru mówią o mieszaniu i rozproszeniu. Z tymi pojęciami wiążą się kilka fundamentalnych pojęć dla kryptografii. W swoich pracach [1], [7] Claude Shannon zdefiniował elementarne dla całej kryptografii pojęcia:

Entropia – ilość informacji zawartej w wiadomości M.

Entropia wiadomości, która może przyjmować n równo prawdopodobnych znaczeń jest równa:

$$H(M) = \log_2 n$$

Zawartość informacyjna języka - Shannon, w swoich pracach pisze, że entropia tekstu zależy od jego długości (N). Entropię testów 16-litrowych określić można jako 1,3 bit/znak.

$$r = H(M) / N$$

Bezwzględna zawartość informacyjna języka, co definiuje się jako założoną liczbę bitów, równo prawdopodobnych, które mogą być przyporządkowane każdemu znakowi. Dla języka, który używa alfabetu L znakowego jego bezwzględna wartość informacyjna wynosi

$$R = \log_2 L$$

Nadmiarowość języka

$$D = R - r$$

Dla języka angielskiego nadmiarowość można, więc wyznaczyć w następujący sposób:

$$R = \log_2 26 = 4{,}7$$
$$r = 1{,}3$$
$$D = R - r = 3{,}4$$

Z wyliczeń powyższych wynika, więc że każdy znak języka angielskiego przenosi 3,4 bitu nadmiarowej informacji. Z powyższego wynika prawidłowość, że im bardziej nadmiarowy język w jakim zapisano tekst jawny tym łatwiejsza kryptoanaliza szyfrogramu.

Bierze się to z faktu, że naturalna nadmiarowość języka wykorzystana może zostać w procesie kryptoanalizy do zawężenia liczby możliwych tekstów jawnych. Konsekwencją tego jest częste wykorzystywanie przez programy szyfrujące algorytmów kompresji przed zastosowanie szyfrowania tekstu jawnego. W samym procesie szyfrowania w celu zmniejszenia nadmiarowości tekstu jawnego wykorzystuje się dwie podstawowe operacje:

Mieszanie (podstawienie) – którego, zadaniem jest dokonanie zmniejszenia pomiędzy tekstem jawnym a szyfrogramem.

Rozproszenie (permutacja) – którego, zadaniem jest rozproszenie nadmiarowości po szyfrogramie.

3.2 Własności funkcji boolowskich

Rozważając współczesne funkcje szyfrujące rozważania często sprowadza się do badania funkcji boolowskich. Funkcja boolowska n zmiennych to funkcja :

$$f : B^n \rightarrow B \text{ , gdzie } B = \{0,1\} \text{ (*)}$$

W definiowaniu właściwości funkcji binarnych często posługujemy się pojęciem tabeli prawdy, jako sposobu zapisu funkcji. Tak więc, jeśli rozpatrzymy funkcję (*), która danym wektorom:

$$\alpha_0 = (00...00),$$
$$\alpha_1 = (00...01),$$
$$...$$
$$\alpha_{2^n-1} = (11...11).$$

przypisuje wartości binarne, takie że $f(\alpha_n) \in B$ to powstający w ten sposób ciąg binarny :

$$\left(f(\alpha_0), f(\alpha_1), ..., f(\alpha_{2^n-1})\right)$$

nazywamy tabelą prawdy.

3.3. Zrównoważenie

W sytuacji analizy algorytmów szyfrujących pod kontem ich bezpieczeństwa często sięga się po pojęcia zrównoważanie funkcji boolowskiej oraz wartości wagi Hemminga. Wagą Hemminga wektora v nazywamy liczbę jedynek w nim występujących, hwt(v) w odniesieniu do funkcji boolowskiej waga Heminga to liczba jedynek w jej tabeli prawdy.

Odległość Heminga_między dwiema funkcjami jest parametrem, który określa liczbę wejść dwóch, funkcji, na których te funkcje się różnią. Zdefiniować ją można jako:

$$d(f,g) = hwt(f \oplus g) = \sum_x f(x) \oplus g(x)$$

Funkcja binarna jest afiniczna jeśli możliwe jest zapisanie jej jako :

$$f(x_1,...,x_n) = a_0 \oplus a_1 x_1 \oplus ... \oplus a_n x_n \text{ , gdzie } a_i \in B, B = \{0,1\}, i = 0,...,n$$

Funkcja afiniczna jest określana jako liniowa jeśli a0=0.

Funkcja boolowska jest zrównoważona, jeśli jej tabela prawdy zawiera 2n-1 jedynek lub zer, inaczej mówiąc tyle samo zer co jedynek. Właściwość ta ma szczególne znaczenie dla s-bloków. Nie zrównoważenie powoduje podatność na wszelkiego rodzaju metody kryptoanalizy, które bazują na analizie prawdopodobieństw ciągu wyjściowego. Niebezpieczeństwo to wzrasta wraz ze wzrostem liczby rund, w których wykorzystywany jest niezrównoważony s-blok, ponieważ powstaje pewne odchylenie, przy którym niektóre ciągi wyjściowe są bardziej prawdopodobne niż inne. Stanowi to tym samym punkt zaczepienia dla kryptoanalizy.

3.4. Zupełność

Kam i David w pracy [8] w podniesieniu do sieci S-P zdefiniowali pojęcie zupełności funkcji szyfrującej. Funkcja $E : \{0,1\}^k \times \{0,1\}^l \rightarrow \{0,1\}^l$ jest zupełna jeśli dla każdej pary pozycji bitów

$i, j \varepsilon \{0,1,...,t-1\}$ istnieją przynajmniej dwa bloki wejściowe X_1, X_2 takie że $X_1 \oplus X_2 = i$, dla których dane wyjściowe Y_1, Y_2 takie że $Y_1 \oplus Y_2 = j$. Inaczej mówiąc, dla kryptograficznej funkcji zupełnej każdy bit szyfrogramu zależy od wszystkich bitów tekstu jawnego.

3.5. Efekt lawinowy

Definicja właściwości jaką jest efekt lawinowy, wiąże się z wyżej wspomnianą właściwością zupełności. W pracach [5],[9] Feistel zdefiniował kryterium lawinowości:

Funkcja $f : B^n \to B$ spełnia kryterium lawinowości, jeśli zmiana jednego bitu w ciągu wejściowym powoduje zmianę przynajmniej połowy bitów na wyjściu.

Rozwinięciem tej definicji jest ścisłe kryterium lawinowości . Właściwość tę określa się często skrótem SAC (Strict Avalanche Criterion) zdefiniowali ją A.F. Webster and S.E. Tavaresw 1986.

Funkcja $f : B^n \to B$, spełnia warunek SAC jeśli $f(x) \oplus f(x \oplus \alpha)$ jest zrównoważona dla wszystkich wektorów α takich że ich waga Heminga równa jest 1. Funkcjonuje również pojęcie warunku SAC k-tego rzędu. Warunek ten to niejako uogólnienie zwykłego warunku SAC, do przypadku gdy zmianie w wektorze wejściowym ulega więcej niż jeden bit. W takim przypadku funkcja f spełnia warunek SAC rzędu K jeśli $f(x) \oplus f(x \oplus \alpha)$ jest zrównoważone dla każdego $\alpha \in B,$ oraz $1 \le W(\alpha) \le k$ (W-waga Hemminga).

3.6. Nieliniowość

Nieliniowości funkcji boolowskiej f, którą można zdefiniować jako odległość tej funkcji od zbioru wszystkich jej funkcji afinicznych oznaczanego jako An, zdefiniowanych nad Bn.

$N_f = \min_{g \in A_n} d(f, g)$, gdzie N – stopień nieliniowości, g – funkcja afiniczna

W pracy [11] wprowadzone zostało pojęcie bent funkcji, które określane są funkcjami doskonale nieliniowymi. Funkcja binarna jest doskonale nieliniowa jeśli dla $f(x) \oplus f(x \oplus \alpha)$ jest zrównoważona dla każdego wektora α takiego że jego waga Hemminga zawiera się w przedziale od 1 do n. Bent funkcje, cechują się dużymi stopniami nieliniowości oraz spełniają warunek SAC ale nie są funkcjami zrównoważonymi. Mimo to są często wykorzystywane w procesie konstrukcji s-bloków poprzez wykorzystanie odpowiednich metod numerycznych.

3.7. Inne własności funkcji szyfrujących

Rozmiar s-bloku: generalnie istnieje zasada im większy s-box tym lepiej, ponieważ trudniejsze jest wyznaczenia odpowiednich charakterystyk na potrzeby kryptoanalizy liniowej czy różnicowej. Znaczenie ma też stosunek liczby bitów wejściowych do wyjściowych:

Jeżeli: $n \ge 2^m - m, \Rightarrow$ s-blok realizuje f liniową

Jeżeli: $n \ge 2^m \Rightarrow$ to między bitami n zachodzi zależność liniowa

n – bity wejściowe, m – bity wejściowe

Profile XOR. Są podstawowym narzędziem kryptoanlizy różnicowej. Z punktu widzenia projektowania s-bloku należy dążyć do tego aby tablica XOR-profilu dla danego s-bloku nie zawierała dużych liczb. Co znacznie ułatwiło by wyznaczenie klucza rundy. Problem dużych liczb występujących w XOR-profilu można rozwiązać częściowo poprzez zwiększanie liczby rund, jednak dzieje się to kosztem wzrostu złożoności czasowej.

Kompletność: dotyczy szyfru jako całości. Przy odpowiednio przygotowanych s-blokach oraz p-blokach należy zadbać o właściwe ich „otoczenie". Między innymi należy dobrze określić licz-

bę rund, sposoby generowania kluczy dla poszczególnych cykli. Tak, aby osiągnięty został określony poziom odporności na znane metody kryptoanalityczne.

4. Po co nam to wszystko

W poprzednich rozdziałach przedstawione zostały wybrane właściwości funkcji logicznych. Wybrane pod kontem oceny jakości algorytmów szyfrujących. Głównym celem naszych badań jest pokazanie, alternatywy dla obecnie powszechnego podejścia do projektowania i implementacji algorytmów szyfrujących. Tą alternatywą mają być układu uczące się. Zmierzamy do tego, aby udowodnić, że pisanie programu komputerowego, lub programowanie układu programowalnego to nie jedyny bezpieczny sposób realizacji szyfrów. Zamiast programowania zamierzamy wykorzystać proces uczenia.

Na obecnym etapie prac, potrafmy już skonstruować sieci neuronowe, które realizują dowolne przekształcenia p i s bloków. Ponieważ docelowo nasze starania zmierzają do skonstruowania uniwersalnego, neuronowego układu szyfrującego, którego właściciel będzie w stanie realizować za jego pomocą różne funkcje szyfrujące. Koniecznym jest przygotowanie pewnych narzędzi, które pozwoliłyby zabezpieczyć właściciela takiego układu przed korzystaniem z przekształceń szyfrujących nie zapewniających odpowiedniego poziomu bezpieczeństwa. Powyższe rozdziały to zdefiniowanie pewnych cech jakie powinny być ocenianie z punkt widzenia bezpieczeństwa realizowanego algorytmu szyfrującego.

Podczas PLOUG'07 zaprezentowane zostaną wyniki oceny wyżej omówionych właściwości dla rzeczywistych realizacji przekształceń szyfrujących realizowanych przez nasze „neuronowe układy szyfrujące". Do oceny jakości naszych rozwiązań wykorzystane zostaną testy statystyczne służące ocenie jakości algorytmów szyfrujących.

Bibliografia

[1] C. E. Shannon. Communication theory of secrecy systems. Bell Sys. Tech. J., 28:657{715, 1949.

[2] Lecture Notes on Cryptography, Shafi Goldwasser1 Mihir Bellare2, August 2001,

[3] J. L Smith, The Design of Lucifer, A Cryptographic Device for Data Communication, RC 3326, White Plains: IBM Research.

[4] Arthur Sorkin, "Lucifer, A Cryptographic Algorithm," Cryptologia (Jan 1984)

[5] H. Feistel, Cryptography and computer privacy. Scientific American, 1973

[6] 1883r. August Kerkhoff 'La Cryptographie Militare'

[7] C. E. Shannon, „A mathematical theory of communication" Bell System Technical Journal, vol. 27, pp. 379-423 and 623-656, July and October, 1948.

[8] Kam J. B. Davida G. I, Structured Design of Substitution-Permutation Encryption Networks,Transactions on Computers, 1979

[9] Feistel H.; Notz W. A.; Smith J. L., Some cryptographic techniques for machine-to-machine data communications, Proceedings of the IEEE, 1975

[10] A. F. Webster and S. E. Tavares. On the design of S-boxes. Advances in Cryptology (CRYOTO'85). Lecture Notes in Computer, pages 523{534 Springer, Berlin Heidelberg New York, 1986.

[11] O. S. Rothaus. On bent functions. Journal of Combinatorial Theory, Series A, 1976.

[12] P. Kotlarz, Z. Kotulski, "Application of neural networks for implementation of cryptographic functions", w: "Multimedia w biznesie i edukacji", Tom 1, str. 213, Białystok 2005, ISBN 83-9182218-7-0

[13] P. Kotlarz, Z. Kotulski, On application of neural networks for S-boxes design, in: P. S. Szczepaniak, J. Kacprzyk, A. Niewiadomski, Advances in Web Intelligence: Third International Atlantic Web Intelligence Conference, AWIC 2005, Łódź, Poland, June 6-9, 2005. Lecture Notes in Artificial Intelligence, LNCS 3528, pp. 243-248, Springer, Berlin 2005.ISBN: 43-540-26219-9.

[14] Autor: mgr Piotr Kotlarz, dr hab. Zbigniew Kotulski, "Neural network as a programmable block cipher – experimental results", Advances in Information Processing and Protection, Springer-Verlag Berlin Heidelberg 2007, ISBN: 978-0-387-73136-0

[15] XI Konferencja użytkowników i deweloperów ORACLE, ISSN 1641-2117, str.179-189, Kościelisko 18-21 październik 2005 Tytuł : "Metody sztucznej inteligencji we współczesnej kryptografii" Autor : mgr Piotr Kotlarz, dr hab. Zbigniew Kotulski

[16] XII Konferencja użytkowników i deweloperów ORACLE, ISSN 1641-2117, str. 227-235 Kościelisko 17-20 październik 2006 Tytuł: "Neuronowy układ szyfrujący – analiza bezpieczeństwa", Autor: mgr Piotr Kotlarz, dr hab. Zbigniew Kotulski

[17] Emeka Mosanya, Christof Teuscher, Héctor Fabio Restrepo, Patrick Galley, Eduardo Sanchez, : CryptoBooster: A Reconfigurable and Modular Cryptographic Coprocessor, Lecture Notes in Computer Science , Volume 1717 / 1999 Title: Cryptographic Hardware and Embedded Systems: First International Workshop, CHES'99, Worcester, MA, USA, August 1999. Proceedings

[18] Taylor R., Goldstein S.: A High-Perfomance Flexible Architecture for Cryptography, Proceedings of the Workshop on Cryptographic Hardware and Embedded Systems, Worcester August 1999

[19] Lim C. H.: CRYPTON: A New 128-bit Block Cipher, Proceedings of the First Advanced Encryption Standard Candidate Conference, Ventura, California, National Institute of Standards and Technology (NIST), August 1998

[20] Lars R. Knudsen Department of Informatics, University of Bergen, N-5020 Bergen : Correlations in RC6, July 29, 1999

[21] Wojciech Laskowski, Wojsko Polskie, JW 4468, Wrocław, Układy programowalne jako narzędzia wspomagające kryptograficzną ochronę danych, Przegląd telekomunikacyjny, Rocznik LXXIV , nr 3/2001

[22] Tadeusz Łuba, Krzysztof Jasiński, Bogdan Zwierzchowski, Instytut Telekomunikacji Politechniki Warszawskiej, Programowalne układy przetwarzania sygnałów i informacji – technika cyfrowa w multimediach i kryptografii, Przegląd Telekomunikacyjny Rocznik LXXVI nr 8–9/2003

[23] R. Tadeusiewicz, Neural networks, Akademicka Oficyna Wydawnicza RM, 1993