Wykład 7uzup_2-3.

Temat: Poziomy bezpieczeństwa systemów kryptograficznych. Matematyczne podstawy kryptoanalizy. Techniki i metody kryptoanalityczne.

2.1. Poziomy bezpieczeństwa systemów kryptograficznych.

Wyróżnia się rożne cztery poziomy bezpieczeństwa systemów kryptograficznych oraz informacji (tabela 2.1).

Tabela 2.1.

Umowne poziomy bezpieczeństwa

Poziom	Cecha ogólna
Poziom A1 *Maksymalny (M – Maximum)*	Najwyższy poziom bezpieczeństwa: zweryfikowana realizacja systemu wymaga formalnego dowodu, ze system jest zgodny z postawionymi mu wymaganiami. *Błędy w obszarze IT prowadza do załamania się instytucji oraz groza konsekwencjami dla wybranych dziedzin gospodarki lub funkcjonowania społeczeństwa.*
Poziom B (podpoziomy B1, B2, B3) *Wysoki (H – High)*	Etykietowany poziom zabezpieczeń. Etykiety pozwalają na stopniowanie poziomu poufności obiektów (zasobów) i poziomu zaufania do poszczególnych użytkowników. B1 blokuje możliwość zmiany praw dostępu do obiektu przez właściciela. B2 ma zabezpieczenia strukturalne, w których każdy z obiektów musi mięć własną etykietę (jedną lub kilka). B3 wprowadza sprzętowe rozdzielenie obszarów poufnych, a użytkownik ma gwarancje, ze jego terminal komunikuje się bezpośrednio z systemem operacyjnym (a nie koniem trojańskim). *Zdarzenia szkodliwe prowadza do załamania głównych obszarów funkcjonowania instytucji i powodują znaczne szkody dla niej oraz jej partnerów.*
Poziom C (podpoziomy C1, C2) *Umiarkowany (MR – Moderate)*	Zabezpieczenie uznaniowe, odmienne dla właściciela, grupy użytkowników i klientów. Wymaga logowania i stosowania hasła, przy czym możliwe są rożne stopnie blokowania dostępu. Dla systemów z bezpieczeństwem C1 informacja o nazwach kont i zakodowanie hasła są dostępne wszystkim upoważnionym użytkownikom systemu. W systemach z zabezpieczeniem C2 jedynie informacja o nazwach kont jest dostępna wszystkim użytkownikom, natomiast zakodowanie hasła wyłącznie dostępne dla systemu operacyjnego. Możliwy jest kontrolowany dostęp, śledzenie działań użytkowników, blokowanie niektórych instrukcji. *Wynikiem szkodliwego zdarzenia jest sparaliżowanie instytucji.*
Poziom D *Niski*	Minimalny poziom zabezpieczeń, w praktyce całkowity ich brak. *Wynikiem szkodliwego zdarzenia jest zaklocenie w pracy*

(L – Low)	*instytucji.*

Teoretycznie wszystkie szyfry można złamać, jeśli do dyspozycji będzie dostatecznie dużo czasu i mocy obliczeniowej. Ponieważ złamanie niektórych szyfrów wymaga zbyt dużych nakładów, praktycznie nie można ich złamać i są one uważane za bezpieczne, mianowicie:

a) jeśli koszt złamania szyfru jest większy niż wartość zaszyfrowanej informacji;
b) jeśli czas niezbędny do zlatania szyfru jest dłuższy niż czas, w którym dane musza być utajnione;
c) jeśli ilość danych zaszyfrowanych z zastosowaniem pojedynczego klucza jest mniejsza niż ilość danych niezbędnych do złamania szyfru.

Do określenia szyfrów bezpiecznych stosuje się dwa pojęcia: bezpieczeństwa bezwarunkowego i bezpieczeństwa obliczeniowego:

- szyfr jest bezwarunkowo bezpieczny, jeśli niezależnie od liczby przechwyconych kryptogramów nie ma w nich wystarczających informacji, aby jednoznacznie określić tekst jawny;
- szyfr jest bezpieczny obliczeniowo, gdy nie można go złamać analitycznie za pomocą dostępnych środków.

Ilość czasu i mocy obliczeniowej potrzebnej do złamania szyfru nazywamy nakładem pracy i określamy za pomocą liczby operacji komputerowych. Ponieważ moc obliczeniowa komputerów ciągle rośnie, wiec utrzymanie odpowiedniego poziomu bezpieczeństwa szyfrów wymaga konstruowania coraz lepszych algorytmów kryptograficznych.

Kategorie łamania szyfrów Lars Knudsen sklasyfikował następująco:

1) całkowite złamanie szyfru. Kryptoanalityk znajduje właściwy klucz $\boldsymbol{K}$ do odszyfrowania wiadomości taki, że $\boldsymbol{D_K(C) = M}$;
2) ogólne wnioskowanie. Kryptoanalityk znajduje alternatywny algorytm, który jest równoważny z algorytmem $\boldsymbol{D_K(C)}$ i nie wymaga znajomości właściwego klucza $\boldsymbol{K}$;
3) częściowe (lokalne) wnioskowanie – odszyfrowanie wiadomości (nie znamy klucza). Kryptoanalityk znajduje tekst jawny dla pewnego przechwyconego kryptogramu.
4) Informacyjne wnioskowanie. Kryptoanalityk znajduje drobną informację o kluczu lub tekście jawnym (nie zna klucza ani całego tekstu jawnego). Taką informacją mogą być kilka bitów klucza, wiadomości o formie tekstu jawnego itd.

2.2. Matematyczne podstawy kryptoanalizy.

Podstawy teoretyczne kryptologii (kryptografia oraz kryptoanaliza) zostały opracowane przez Shannona w roku 1949 na podstawie swej rozprawy dotyczącej teorii informacji. W stworzonej nim teorii ilość informacji zawartej w wiadomości mierzy się za pomocą entropii (equivocation)

$$H(X) = -\sum_{i=1}^{n} p(x_i) \log_2 p(x_i) \quad , \tag{2.1}$$

przy czym $p(x_i)$ - oznacza prawdopodobieństwo wystąpienia wiadomości i.

Entropia jest miara nieokreśloności wiadomości, wiec podaje przeciętną liczbę bitów informacji niezbędną do optymalnego zaszyfrowania wszystkich możliwych wiadomości lub odtworzenia tekstu jawnego z kryptogramu. Gdy sygnały występują z jednakowym prawdopodobieństwem $p(x_i) = \frac{1}{n}$, wtedy entropia osiąga maksymalną wartość, która wynosi

$$H(X) = \log_2 n \quad . \tag{2.2}$$

Zawartością informacyjną języka, inaczej wskaźnikiem języka (rate of the language) r nazywamy przeciętną liczbę bitów informacji przypadających na jeden znak

$$r = \frac{H(X)}{N} \quad , \tag{2.3}$$

gdzie N - długość wiadomości wyrażona w bitach. Dla języka angielskiego r wynosi od 1,0 do 1,5 bita/literę.

Bezwzględna zawartość informacyjna lub wskaźnik bezwzględny języka (absolute rate) R określa maksymalną ilość informacji, która mogłaby być zakodowana w jednym znaku

$$R = \log_2 L \quad , \tag{2.4}$$

gdzie L - ilość znaków alfabetu języka, inaczej długość alfabetu.

Na przykład, dla języka angielskiego $R = \log_2 26 = 4{,}7$ bita/literę.

Nadmiarowość języka lub redundancję określa się różnicą pomiędzy wskaźnikiem bezwzględnym języka a faktycznym wskaźnikiem języka

$$D = R - r \quad . \tag{2.5}$$

Jeżeli przyjmiemy wartość średnią 1,2 bita/literę, oznacza to, ze s ośmiu bitów kodu ASCII tylko 1,2 bita przenosi informacje, a reszta jest nadmiarem.

Im bardziej nadmiarowy jest język, tym łatwiejsza jest kryptoanaliza szyfrów.

Nieokreśloność wiadomości można zmniejszyć dostarczając dodatkową informację. Wtedy mamy do czynienia z entropią warunkową wiadomości $H_Y(X)$, którą przy danym definiujemy jako:

$$H_Y(X) = \sum_{X,Y} p(X, Y) \log_2 \left(\frac{1}{p_Y(X)} \right) \quad . \tag{2.6}$$

Poufność szyfru w kategoriach entropii warunkowej klucza $H_C(K)$ dla danego kryptogramu Shannon określił wzorem

$$H_C(K) = \sum_{X,C} p(K, C) \log_2 \left(\frac{1}{p_C(K)} \right) \quad , \tag{2.7}$$

gdzie $p_C(K)$ - prawdopodobieństwo zastosowania klucza przy danym .

Długość krytyczna jest najmniejsza wartością (długość szyfrogramu), dla której entropia warunkowa klucza $H_C(K)$ jest bliska zera. Stad wypływa, ze szyfr jest bezwarunkowo bezpieczny, jeżeli $H_C(K)$ nigdy nie osiągnie zera, nawet dla bardzo dużej długości szyfrogramu.

Pod czas konstruowania algorytmów kryptograficznych, a również łamania ich, niebagatelna role odgrywają:

- teoria liczb (arytmetyka modularna, liczby pierwsze, największy wspólny dzielnik, odwrotności modulo pewna liczba – rozszerzony algorytm Euklidesa, małe twierdzenie Fermata, funkcja Eulera, chińskie twierdzenie o resztach, reszty kwadratowe, obliczenia w ciele Galois),
- faktoryzacja (algorytmy faktoryzacji:
  - ***sito ciała liczbowego*** (Number Field Sieve – NFS). Jest to najszybszy znany algorytm faktoryzacji dla liczb dłuższych niż 110 cyfr dziesiętnych. W ten sposób w 1993 r. była rozłożona dziewiąta liczba Fermata: $2^{512} + 1$ – długość 155 znaków dziesiętnych;
  - ***sito kwadratowe*** (Quadratic Sieve). Jest to najszybszy znany algorytm faktoryzacji dla liczb krótszych niż 110 cyfr dziesiętnych. Wersji tego algorytmu:
    a) szybsza – ***wielokrotne wielomianowe sito kwadratowe*** *(Multiple Polynomial Quadratic Sieve),*
    b) najszybsza – ***odmiana wielokrotnego wielomianowego sita kwadratowego dla podwójnie dużych liczb pierwszych*** *(Double Large Prime Variation of the Multiple Polynomial Quadratic Sieve);*
  - ***metoda krzywych eliptycznych*** (Elliptic Curve Method – ECM). Metoda ta była używana do rozkładu na czynniki pierwsze liczb o długości do 43 cyfr. Dla liczb dłuższych szybsze są inne algorytmy;
  - ***algorytm Monte Carlo Pollarda***;
  - ***algorytm ciągłego podziału*** (Continued Fraction Algorithm). Algorytm ten nie jest stosowany;
  - ***dzielenie próbne*** (Trial Division). Jest to najstarszy algorytm faktoryzacji i polega na testowaniu każdej liczby pierwszej mniejszej od pierwiastka kwadratowego wytypowanej liczby);
- generowanie liczb pierwszych (test Solovaya-Strassena, test Rabina-Millera, mocne liczby pierwsze);
- logarytmy dyskretne w skończonym ciele liczbowym.

2.3. Techniki i metody kryptoanalityczne.

2.3.1. Podstawowe techniki łamania szyfrów przez kryptoanalitykow to są:

1. Technika prób i błędów: polega na sprawdzeniu każdego możliwego klucza i poszukiwaniu sensownego tekstu jawnego. Łamanie klucza przez podstawianie wszystkich możliwych jego kombinacji nazywamy *pełnym przeszukiwaniem* (*exhaustive search*) lub *łamaniem* czy *atakiem brutalnym* (*brute force*):
   - podstawowy atak na wszystkie szyfry (nie tylko blokowe);
   - dla ***k***-bitowego klucza i ***n***-bitowego bloku do jednoznacznego wyznaczenia klucza potrzeba $\lceil k/n \rceil$ par tekst jawny-szyfrogram.

Pełne przeszukiwanie:

- dla ustalonego rozmiaru klucza definiuje górną granicę bezpieczeństwa szyfru;
- wymaga znanego tekstu jawnego lub tekstu jawnego zawierającego nadmiarowość;
- ma bardzo szerokie zastosowanie ze względu na projektowe wymaganie efektywnej realizacji szyfrowania.

Dla ***n***-bitowego szyfru blokowego z ***k***-bitowym kluczem:

- znana jest mała liczba (np. $\lceil (k+4)/n \rceil$) par tekstu jawnego szyfrogramów zaszyfrowanych kluczem ***K***;
- klucz ***K*** odnajdujemy poprzez pełne przeszukiwanie w oczekiwanym czasie 2^{k}-1 operacji.

Algorytm:

- rozszyfrujemy ustalony szyfrogram ***C*** kolejno każdym kluczem;
- odrzucamy te klucze, które nie dają teksty jawnego ***P***;
- poszukiwany klucz znajduje się pośród kluczy nieodrzuconych;
- liczba „fałszywych alarmów” zależy od rozmiarów ***k*** i ***n***;
- dodatkowa para ***P'***, ***C'*** wystarcza aby odrzucić klucze fałszywe;
- oczekujemy, że znajdziemy klucz po przeszukaniu połowy przestrzeni klucza.

2. Analiza statystyczna:

- ataki wykorzystujące strukturę (sposób działania) szyfru:
  - kryptoanaliza różnicowa,
  - kryptoanaliza liniowa,
  - inne specyficzne (dedykowane) ataki;
- sprowadza się do wyznaczania prawdopodobieństwa rozkładu liter (znaków) w kryptogramie i tekście jawnym. Ponieważ rozkład liter jest charakterystyczną cechą każdego języka, informacje takie można wykorzystać do łamania niektórych szyfrów, np. szyfrów podstawieniowych.

3. Analiza pracy urządzenia (programu) szyfrującego:

Paul Kocher badał właśnie takie niekonwencjonalne ataki. Wykorzystują one słabe punkty we wdrażaniu szyfru. W 1995 r. wykazał, że w sprzyjających warunkach można klucz złamać, obserwując czas rozszyfrowania wiadomości tym kluczem. Następnie, w 1998 r. ogłosił, że można zaatakować urządzenia komputerowe, stosując nowy rodzaj ataków, zwany "różnicową analizą mocy". Monitorując zużycie mocy urządzeń kryptograficznych, otrzymuje się informacje, które są skorelowane z kluczem.

Wyróżnia się trzy główne ataki wykorzystujące analizę pracy urządzeń szyfrujących:

- różnicowa analiza mocy (*Differential Power Analysis*): ataki na implementację wykorzystujące fakt, że pobór energii elektrycznej przez urządzenie elektroniczne (w szczególności kryptograficzne) zależy od wykonywanych przez nie operacji i argumentów tychże operacji. Ataki tego typu przeprowadzono, między innymi, na urządzenia implementujące algorytmy DES i RSA. Analogicznie różnicowa analiza promieniowania elektromagnetycznego (*differential electromagnetic analysis*).
- analiza czasu pracy (*Timing Analysis*): atak "stoperowy" - atak na implementację, wykorzystujący to, że czas działania urządzenia kryptograficznego może zależeć od wykonywanych przez układ (względnie algorytm) operacji i ich argumentów. Atak "stoperowy" przeprowadzono, m.in. na urządzenia implementujące algorytmy IDEA i RSA.
- analiza różnicowa błędów (*Fault Analysis*): atak na implementację. Atakujący próbuje zakłócić pracę urządzenia tak, aby wyprodukowało ono błędny wynik, a następnie wnioskuje na podstawie znajomości wyniku prawidłowego (bez zakłóceń) i błędnego. Ataki tego typu przeprowadzono, m.in. na urządzenia implementujące algorytmy DES i RSA

Ataki na urządzenia wskazują, że nawet jeżeli algorytm szyfrowania jest w zasadzie niełamalny, to środowisko, w którym go użyto, może mieć słabe miejsca. I rzeczywiście, wiele systemów szyfrowania handlowych produktów programowych jest łatwych do złamania, nie dlatego, że złe są ich podstawy matematyczne lub za krótkie klucze, ale z powodu innych słabości systemu lub sposobu ich eksploatacji.

4. Pełna analiza danych (ataki wykorzystujące małą długość bloku):

– atak słownikowy (dla ustalonego klucza):

- 2^n par – pełny słownik,
- $2^{n/2}$ par i $2^{n/2}$ szyfrogramów (paradoks urodzin);

– dopasowywanie szyfrogramów: wyszukiwanie w szyfrogramie (złożonym z odpowiednio dużej liczby bloków rzędu $2^{n/2}$) jednakowych bloków i próba wysnucia na tej podstawie wniosków odnośnie tekstu jawnego;

– wyszukiwanie prawdopodobnych słów: każdy program lub dokument zawiera pewne słowa a zwroty pojawiające się w określonych miejscach. W programach takimi słowami są np. słowa kluczowe języka programowania, a w dokumentach zwroty grzecznościowe i nazwy miejscowości.

5. Analiza matematyczna: polega na napisaniu układu równań na podstawie znanych algorytmów, rozwiązanie których da wartości zmiennych, reprezentujących fragmenty klucza lub wiadomości. Wtedy można także uzyskać wyrażenia generujące klucze kryptograficzne.

W przypadku kluczy generowanych za pomocą algorytmów komputerowych dąży się do znalezienia algorytmu i jego parametrów. Podczas łamania klucza wykorzystuje się jego okresowość i redundancję.

Problem łamania klucza kryptograficznego, generowanego komputerowo, sprowadza się do znalezienia wielomianu charakterystycznego sekwencji klucza. Rozważmy klucz kryptograficzny generowany przez generator liniowy. Aby znaleźć wielomian charakterystyczny sekwencji okresowej, wystarczy znać co najmniej $2m$ bitów tej sekwencji, przy czym m jest stopniem wielomianu generującego sekwencję okresową, a okres binarnej sekwencji pseudolosowej wynosi $2^m - 1$. Mając fragment sekwencji okresowej $s_0, s_1, s_2, \ldots, s_{2m-1}$, można napisać układ m równań zawierający m niewiadomych:

$$\left.\begin{aligned} a_0 s_0 + a_1 s_1 + \ldots + a_m s_m &= 0 \\ a_0 s_1 + a_1 s_2 + \ldots + a_m s_{m+1} &= 0 \\ &\ldots \\ a_0 s_{m-1} + a_1 s_m + \ldots + a_m s_{2m-1} &= 0 \end{aligned}\right\} . \qquad (2.8)$$

Rozwiązanie powyższego układu równań pozwoli znaleźć wielomian generujący sekwencję okresową stopnia m, lecz wymaga ono zastosowania m^3 operacji i m^2 komórek pamięci. Metodę tę ilustruje przykład 2.1.

P r z y k ł a d 2.1.

Wyznaczanie wielomianu charakterystycznego binarnej sekwencji okresowej. Zakładamy, że mamy fragment sekwencji okresowej $1000100110\ldots$. Wyznaczmy współczynniki wielomianu stopnia piątego generującego tą sekwencję

$$a_5x^5 + a_4x^4 + a_3x^3 + a_2x^2 + a_1x + a_0 = 0 \quad . \tag{2.9}$$

Układ równań (2.8) napiszemy, podstawiając elementy sekwencji okresowej. Stosujemy w tym celu poniższą tabelę.

1	0	0	0	1	0	0	1	1	0	Równania
a_0	a_1	a_2	a_3	a_4	a_5					$a_0 + a_4 = 0$
	a_0	a_1	a_2	a_3	a_4	a_5				$a_3 = 0$
		a_0	a_1	a_2	a_3	a_4	a_5			$a_2 + a_5 = 0$
			a_0	a_1	a_2	a_3	a_4	a_5		$a_1 + a_4 + a_5 = 0$
				a_0	a_1	a_2	a_3	a_4	a_5	$a_0 + a_3 + a_4 = 0$

W pierwszym wierszu tabeli podano elementy sekwencji okresowej $s_0\ s_1\ s_2 \ldots s_9$, a w ostatniej kolumnie - równania (2.8). Jeśli założymy, że $a_5 = 1$ i $a_0 = 1$, to po rozwiązaniu układu równań otrzymamy: $a_4 = 1$, $a_3 = 0$, $a_2 = 1$ i $a_1 = 0$. Wielomian generujący sekwencję ma postać

$$x^5 + x^4 + x^2 + 1 = (x^4 + x + 1)(x + 1) \quad . \tag{2.10}$$

Otrzymany wielomian piątego stopnia można rozłożyć na dwa wielomiany: stopnia czwartego i pierwszego. Łatwo sprawdzić, że wielomiany stopnia piątego i czwartego generują taką samą sekwencję okresową. Wielomian stopnia czwartego można również otrzymać bezpośrednio, zakładając $a_5 = 0$.

Kryptoanaliryk poszukujący równania charakterystycznego klucza zwykle nie zna jego stopnia. Może on założyć stopień równania, a następnie, jeśli uzyska rozwiązanie, zredukować jego stopień, rozkładając je na wielomiany nierozkładalne.

Znany jest również algorytm Berlekampa-Masseya, który umożliwia znalezienie bezpośrednio równania charakterystycznego klucza najniższego stopnia.

2.3.2. Metody kryptoanalizy.

Anagramowa: jest efektywna, gdy mamy do czynienia z szyframi permutacyjnymi (przestawieniowymi). Kryptoanalityk może łatwo określić, czy używany szyfr jest szyfrem permutacyjnym, gdyż częstość wystąpień liter tekstu jawnego jest taka sama jak częstość występowania liter w kryptogramie.

Atak polega na odtworzeniu właściwej kolejności przemieszanych znaków (liter, spacji, liczb oraz kropki i przecinka) z zastosowaniem tablic częstości występowania wybranych znaków w tekstach oraz programach, np. digramów (kombinacji 2-literowych) i trigramów (kombinacji 3-literowych).

Tabela 2.2.

Częstość występowania wybranych znaków w tekstach

Znak	Polski	Ang.	Pascal	Znak	Polski	Ang.	Pascal	Znak	Polski	Ang.	Pascal
A	0,080	0,067	0,037	N	0,047	0,053	0,050	**Spacja**	0,172	0,197	0,192
B	0,013	0,013	0,013	O	0,071	0,063	0,046	0	-	-	0,003
C	0,038	0,019	0,032	P	0,024	0,012	0,022	1	-	-	0,004
D	0,030	0,031	0,028	Q	-	0,001	-	2	-	-	0,002
E	0.069	0,089	0,081	R	0,035	0,042	0,057	3	-	-	0,001
F	0,001	0,021	0,014	S	0,038	0,043	0,034	4	-	-	0,001
G	0,010	0,017	0,017	T	0,024	0,070	0,060	5	-	-	0,002
H	0,010	0,043	0,015	U	0,018	0,021	0,019	6	-	-	0,001
I	0,070	0,054	0,050	V	-	0,006	0,008	7	-	-	0,001
J	0,019	0,002	0,002	W	0,036	0,018	0,007	8	-	-	0,001
K	0,027	0,009	0,003	X	-	0,001	0,008	9	-	-	0,002
L	0,031	0,033	0,031	Y	0,032	0,023	0,008		0,009	0,008	0,012
M	0,024	0,022	0,014	Z	0,056	0,001	0,001		0,009	0,002	0,010

Częstości występowania liter: jest szczególnie przydatna do łamania prostych szyfrów podstawieniowych.

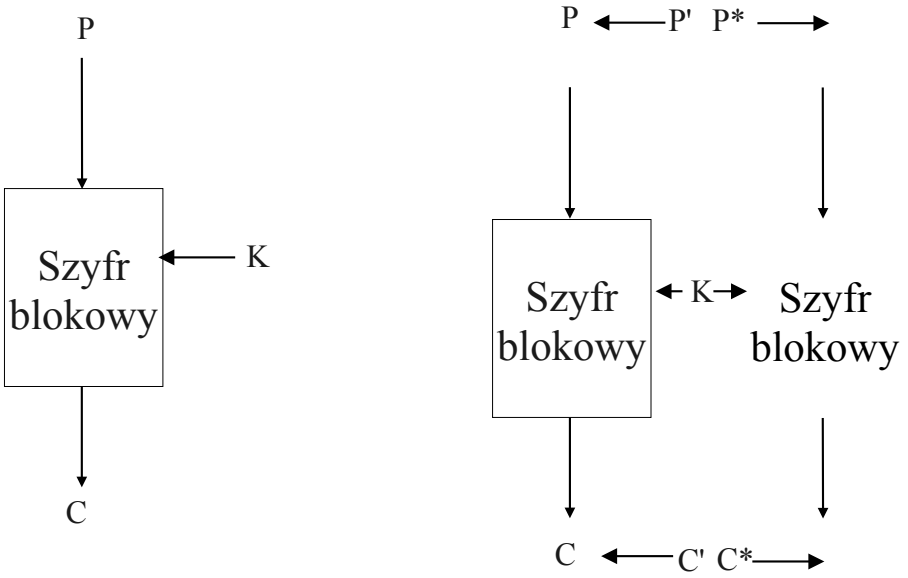
Ta metoda:

- polega na porównaniu częstości występowania znaków w szyfrogramie z częstościami oczekiwanymi;
- pozwala z dużym prawdopodobieństwem dopasować znaki szyfrogramu do znaków wiadomości jawnej.

Wielce pomocna dla kryptoanalityka jest znajomość względnej częstości występowania znaków, mianowicie digramów i trigramów.

Różnicowa (oopracowana przez Eli Biham'a i Adi Shamir'a w "Differential Cryptanalysis of the Data Encryption Standard" w 1993 r.): polega na porównaniu dwóch szyfrogramów, które powstały w wyniku zaszyfrowania pary tekstów jawnych różniących się w określony sposób. Oba teksty są szyfrowane tym samym kluczem i są analizowane uzyskane zmiany miedzy kryptogramami w miarę wykonywania kolejnych cykli algorytmu.

Idea kryptoanalizy różnicowej:



$(P, C) \Rightarrow K = ?$ $(P, C, P', C') \Rightarrow K \in \{ ... \}$

K = xxx00110xxxxx

$P' = P \oplus P^*$

$C' = C \oplus C^*$ $\oplus$ - suma modulo 2

P = 0101010001001111

P* = 0101010111101111

P' = 0000000110100000

Stosując te metodę Biham i Shamir znaleźli metodę ataku na algorytm DES z wybranym tekstem jawnym, która była bardziej efektywna od ataku brutalnego. Później metoda różnicowa została zaadaptowana do rożnych innych szyfrów iteracyjnych. Wybieramy parę tekstów jawnych (atak z wybranym tekstem jawnym - CPA) z ustalonymi różnicami. One mogą być wybrane losowo, lecz różnice miedzy nimi powinni spełniać pewne warunki. Kryptoanalitycy nie musza znać wartości tych różnic. (W przypadku DES określenie *różnica* jest zdefiniowane za pomocą operacji XOR. Sposób wyznaczania różnicy można rożnie zdefiniować dla różnych algorytmów kryptograficznych). Następnie stosując różnicę miedzy kryptogramami, przypisujemy różne prawdopodobieństwa do różnych kluczy. Pewne różnice w tekście jawnym charakteryzują się dużym prawdopodobieństwem otrzymania określonych różnic miedzy wynikowymi szyfrogramami. Są one nazywane *charakterystykami*. *Charakterystyki różnicowe* rozwijają się wraz z liczbą wykonanych cykli (rund) i wyznaczają ścieżkę między cyklami. Różnicy wejściowej, różnicy w każdym cyklu i różnicy wyjściowej odpowiadają specyficzne prawdopodobieństwa.

n-rundowa *charakterystyka*: ***n+1*** elementowy wektor $\boldsymbol{\Omega_n = (X'_1, X'_2, \ldots, X'_{n+1})}$
złożony z ***n*** różnic wejściowych do ***n*** rund i różnicy wyjściowej po ***n***-tej rundzie.

Para tekstów jawnych z właściwościami charakterystyki jest nazywana *parą właściwą*. Para tekstów jawnych bez właściwości charakterystyki jest nazywana *parą niewłaściwą*. Para właściwa sugeruje wartość poprawnego klucza ostatniego cyklu (dla ostatniego cyklu charakterystyki). Para niewłaściwa sugeruje przypadkowy klucz cyklu.

Prawdopodobieństwo charakterystyki

p(Ω) – iloczyn prawdopodobieństw przejść dla poszczególnych rund;
Prawdopodobieństwo przejścia dla pojedynczej rundy = iloczyn prawdopodobieństw spowodowania $X' \rightarrow Y'$ dla wszystkich aktywnych skrzynek podstawieniowych.

Dla ustalonego klucza pary tekstu jawnego można podzielić na:

- ✓ pary prawidłowe (*właściwe)* – zgodne z *charakterystyką*,
- ✓ pary fałszywe *(niewłaściwe)* – niezgodne z *charakterystyką*.

Po filtracji (odrzuceniu par fałszywych) zostają wszystkie *pary właściwe* (prawidłowe) i *pary niewłaściwe* (fałszywe) „podobne" do prawidłowych. Identyfikowanie poszukiwanego klucza właściwego polega na zliczeniu pojawień poszczególnych kluczy.

Aby znaleźć właściwy klucz cyklu, wystarczy przeanalizować coraz więcej par szyfrogramów, tzn. wykonać dostateczną liczbę prób z parami właściwymi, w wyniku których jeden z kluczy będzie powtarzał się częściej niż pozostałe. W efekcie właściwy klucz wyłoni się ze zbioru losowych możliwości. W efekcie właściwy klucz wyłoni się ze zbiorów losowych możliwości i okaże się najbardziej prawdopodobny. To będzie właśnie klucz zastosowany do szyfrowania.

Często się zdarza tak, ze skuteczna kryptoanaliza różnicowa, jaką dysponujemy dla pewnego algorytmu szyfrującego, pozwala na identyfikację nie całego, a tylko części klucza ostatniej iteracji. W przypadku algorytmu, którego klucze iteracyjne tworzone są z relatywnie krótkiego klucza głównego w algorytmie generacji kluczy iteracyjnych, oznacza to często znalezienie pewnej, dostatecznie istotnej liczby bitów klucza głównego. Brakujące bity znaleźć można stosując metody wypróbowywania każdego możliwego klucza (łamanie brutalne).

Jednak w stosunku algorytmów, których klucze iteracyjne są niezależne, w przypadku znalezienia części klucza ostatniej iteracji, niemożliwe jest zastosowanie powyższego rozwiązania. Wtedy się stosuje inne charakterystyki różnicowe do znalezienia pozostałej części klucza ostatniej iteracji. Po uzyskaniu całego klucza wykonywane są:

- odszyfrowanie ostatniej iteracji (ponieważ znany jest jej klucz) i
- analogiczna metoda różnicowa, tylko ze na algorytm krótszy o jedną iterację.

Powiązanych kluczy: jest podobna do metody różnicowej, ale sprawdza się różnicę miedzy kluczami; wtedy kryptoanalitycy wybierają relację między dwoma kluczy, lecz nie wybierają samych kluczy.

Wiadomość jest szyfrowana za pomocą oby kluczy. W wersji łamania szyfru ze znanym tekstem jawnym kryptoanalitycy znają tekst jawny a szyfrogram, otrzymany z zaszyfrowania przypadkowych danych wyznaczonym kluczem. W przypadku łamania szyfru z wybranym tekstem jawnym kryptoanalitycy wybierają tekst jawny zaszyfrowany parą kluczy.

Liniowa (jest nowszą i efektywniejszą metodą ataku niż kryptoanaliza różnicowa; opracował ją w latach dziewięćdziesiątych Mitsuru Matsui): polega na znalezieniu prostego przybliżenia złożonej funkcji w każdej rundzie szyfru, np. takiego jak DES; stosuje opis działania bloku szyfrującego w postaci aproksymacji liniowej i wykorzystuje zależności między bitami danych wejściowych cyklu, bitami klucza i wynikami cyklu.

Aproksymacja liniowa to równanie zachodzące dla szyfru blokowego z prawdopodobieństwem p≠½ postaci:

$$P^{[i_1,\ldots,i_a]} \oplus C^{[j_1,\ldots,j_b]} = K^{[k_1,\ldots,k_c]} \qquad (2.11)$$

dla $0 \le a, b \le n$, $0 < c \le k$.

Wartość |p - ½| odpowiada prawdopodobieństwu spełnienia lub niespełnienia równania (2.11).

Oznacza to, ze jeśli zsumujemy modulo 2 niektóre bity tekstu jawnego, zsumujemy modulo 2 niektóre bity szyfrogramu i zsumujemy modulo 2 te wyniki, to powinniśmy otrzymać 1 bit, który jest sumą modulo 2 niektórych bitów klucza. Jest to aproksymacja liniowa i prawdopodobieństwo jej poprawności jest równe *p*. Jeśli $p \neq 1/2$, to warto te nierówność wykorzystać. Wystarczy zastosować zebrane teksty jawne oraz związane z nimi szyfrogramy i zgadywać wartość kluza. Im więcej danych posiadamy, tym bardziej skuteczne jest zgadywanie. Im lepsza aproksymacja, tym większe szanse na sukces z ta samą ilością danych.

Algorytm:

Z pośród ***N*** analizowanych tekstów jawnych dla ***T (< N)*** lewa strona równania (2.11) jest równa 0.

Jeśli $T > \frac{N}{2}$, to przyjmujemy: $K^{[k_1,\ldots,k_c]} = 0$ (gdy p > ½) lub 1 (p < ½),

w przeciwnym przypadku: $K^{[k_1,\ldots,k_c]} = 1$ (gdy p > ½) lub 0 (p < ½).

Prawdopodobieństwo sukcesu tego algorytmu zależy od ***N*** i p i rośnie wraz ze wzrostem ***N*** i |p - ½|. Znajdziemy w ten sposób parzystość niektórych bitów klucza.

Metoda liniowa może być użyta ze znanym tekstem jawnym oraz ze znanym kryptogramem.

Kierunki dalszych badań. Idee metody różnicowej rozbudowano o różnicę wyższych rzędów.

Inną nową metodą kryptoanalizy jest metoda różnicowo-liniowa. Czas wykonania tego typu ataku jest porównywalny z poprzednimi czasami ataku; wymaga jednak znacznie mniej tekstów jawnych. Nie wydaje się jednak, ze metodę te można będzie łatwo tak rozszerzyć, by miała zastosowanie do większej liczby cykli. Metoda ta jest nowa (zwłaszcza w połączeniu z metodą opartą na różnicach wyższych rzędów) i trwają prace nad jej usprawnieniem.

2.4. PRZYKŁAD: Atak różnicowy na szyfr blokowy Q

Parametry szyfru:

Długość bloku n = 128 bitów;

Długość klucza k = 128, 256 bitów;

Struktura SPN, oparta na Rijndaelu i Serpencie;

Liczba rund 8 lub 9;

Wyniki:

- ✓ Atak brutalny wymaga 2127 szyfrowań;
- ✓ Atak różnicowy wymaga 2105 tekstów jawnych 277 szyfrowań.

Wnioski:

- ✓ szyfr Q nie jest odporny na kryptoanalizę różnicową;
- ✓ sposób ataku: użycie charakterystyk „prawie iteracyjnych" (kombinacji wielu charakterystyk);
- ✓ użycie elementów z dwóch bezpiecznych szyfrów nie daje szyfru bezpiecznego.

2.5. Ataki algebraiczne

Idea:

Zapisać szyfr blokowy jako układ równań kwadratowych wiążących tekst jawny z szyfrogramem i kluczem.

Rozwiązać ten układ równań np. za pomocą algorytmu XL wprowadzając równania dodatkowe (ang. *overdefined equation*).

Przykład

Dany jest układ równań kwadratowych:

$$x_1^2 + ax_1x_2 = \alpha$$
$$x_2^2 + bx_1x_2 = \beta$$

z trzema jednomianami kwadratowymi: x_1^2, x_2^2, x_1x_2 .

Mnożąc oba równania przez wszystkie jednomiany otrzymujemy układ:

$$
\begin{aligned}
x_1^4 + ax_1^3 x_2 &= \alpha x_1^2 \\
x_1^2 x_2^2 + bx_1^3 x_2 &= \beta x_1^2 \\
x_1^2 x_2^2 + ax_1 x_2^3 &= \alpha x_2^2 \\
x_2^4 + bx_1 x_2^3 &= \beta x_2^2 \\
x_1^3 x_2 + ax_1^2 x_2^2 &= \alpha x_1 x_2 \\
x_1 x_2^3 + bx_1^2 x_2^2 &= \beta x_1 x_2
\end{aligned}
$$

Mamy teraz 6 równań z 8 jednomianami.

2.6. Złożoność algorytmu XL

Parametry:

n – liczba zmiennych (niewiadomych);

m – liczba równań kwadratowych;

D – stopień jednomianów (w końcowym układzie);

$R = \binom{n}{D-2}$ nowych równań i $T = \binom{n}{D}$ nowych wyrażeń. XL działa gdy: R ≥ T i wtedy $D \approx \frac{n}{\sqrt{m}}$.

Złożoność algorytmy jest równa: $\binom{n}{D}^{\omega}$, gdzie ω jest wykładnikiem redukcji Gaussa.

W Rijndaelu skrzynka podstawieniowa wyraża się równaniem y = x-1 lub xy = 1. Dlatego możemy utworzyć zestaw równań kwadratowych:

$$
\begin{aligned}
x^2 y &= x \\
x^4 y &= x^3 \\
&\vdots
\end{aligned}
$$

Ostatecznie mamy:

- ✓ dla Rijndael'a: 4800 równań z 1600 zmiennymi (złożoność XL 2419);
- ✓ dla Serpent'a: 43680 równań z 8192 zmiennymi.

Możliwe usprawnienia – algorytm XSL:

- ✓ dla Rijndael'a z 128 bitowym kluczem: złożoność XSL 2230;
- ✓ dla Rijndael'a z 256 bitowym kluczem: złożoność XSL 2255;
- ✓ dla Serpent'a z 128/192/256 bitowym kluczem: złożoność XSL 2143.