K. TRYBICKA-FRANCIK

# KRYPTOANALIZA

Opracowanie wewnętrzne Instytutu Informatyki
Gliwice, 1999

## Kryptoanaliza

Kryptoanaliza jest dziedziną wiedzy i badań zajmującą się metodami przełamywania szyfrów. Szyfr jest przełamywalny, jeśli istnieje możliwość odtworzenia tekstu jawnego bądź klucza na postawie tekstu zaszyfrowanego albo określenie klucza na podstawie tekstu jawnego i zaszyfrowanego. Wyróżnia się trzy podstawowe rodzaje przełamywania szyfru:

1. ***Atak bez tekstu jawnego.*** Kryptoanalityk musi określić klucz znając tekst zaszyfrowany. Mogą być też znane: metoda szyfrowania, język tekstu jawnego, tematyka badanego tekstu oraz słowa charakterystyczne dla tej tematyki z określonym prawdopodobieństwem wystąpienia.
2. ***Atak z tekstem jawnym.*** Kryptoanalityk zna pary tekstów jawnego i zaszyfrowanego.
3. ***Atak z wybranym tekstem jawnym.*** Kryptoanalityk może zdobyć tekst zaszyfrowany odpowiadający wybranemu fragmentowi tekstu zaszyfrowanego.

## 1.1 Kryptoanaliza metodą anagramową

Metoda ta jest szczególnie przydatna w przełamywaniu szyfrów przestawieniowych (permutacyjnych). Kryptoanalitycy mogą łatwo poznać, czy zastosowany szyfr jest szyfrem przestawieniowym, gdyż częstość wystąpień liter tekstu szyfrowanego jest taka sama jak częstość występowania liter w tekście zaszyfrowanym.

Metoda anagramowa polega na odtworzeniu właściwej kolejności przemieszanych znaków z wykorzystaniem tablic częstości występowania digramów (kombinacji 2-literowych) i trigramów (kombinacji 3-literowych).

## 1.2 Analiza częstości występowania liter[1]

Metoda ta jest efektywna, gdy mamy do czynienia z prostymi szyframi podstawieniowymi. Atak polega na porównaniu częstości występowania liter w kryptogramie z częstościami oczekiwanymi. Metoda ta pozwala z dużym prawdopodobieństwem dopasować litery kryptogramu do liter tekstu jawnego. Wielce pomocną w pracy kryptoanalityka jest znajomość częstości występowania digramów i trigramów.

## 1.3 Brutalna metoda przełamywania algorytmów kryptograficznych

Brutalna metoda przełamywania algorytmów kryptograficznych polega na przeszukaniu całej przestrzeni klucza. I tak, gdy mamy do czynienia z szyfrem podstawieniowym typu szyfr Cezara to przestrzeń klucza ogranicza się do 25 kombinacji dla alfabetu angielskiego (kolejno sprawdzamy podstawienia dla alfabetu szyfrującego przesuniętego o 1, 2, 3,…, 25 pozycje w prawo w stosunku do alfabetu tekstu szyfrowanego). Złożoność obliczeniowa rośnie, gdy atak przeprowadzamy na szyfr podstawieniowy z alfabetem szyfrującym będącym dowolną kombinacją odwzorowań znak alfabetu jawnego- znak alfabetu szyfrującego. W tym przypadku mamy do sprawdzenia 26! kombinacji, czyli 403291461126605635584000000 przypadków. Współczesne algorytmy szyfrujące opierają się wprawdzie na alfabecie dwuznakowym (0, 1) ale ilość kombinacji dla klucza 56-bitowego jest równa 256. Zakładając, że superkomputer może sprawdzić milion kluczy na sekundę, znalezienie właściwego klucza zajmie mu ok. 2000 lat.

[1] Por. Załącznik D

## 1.4 Kryptoanaliza różnicowa

Kryptoanaliza różnicowa została opracowana przez E. Bihama i A. Shamira dla algorytmu DES. Później została zaadoptowana do różnych innych szyfrów iteracyjnych. Metoda ta polega na szyfrowaniu par tekstów jawnych różniących się w określony sposób (stąd nazwa metody) i analizie uzyskanych szyfrogramów. W niniejszy rozdziale zostanie opisana merytoryczna strona kryptoanalizy różnicowej algorytmu DES

W algorytmie DES zastosowano skrzynki o sześciu bitach wejściowych i czterech wyjścia. Wynika stąd, iż każda skrzynka ma 64*64 możliwe wartości par wejściowych (zakładając, że istotna jest kolejność w parze). Dla każdej pary określić można różnicę jej wartości poprzez wyliczenie ich bitowej sumy modulo 2 (czyli poprzez wykonanie operacji XOR na odpowiadających sobie bitach; rezultat takiej operacji nazywany jest różnicą wartości, ponieważ na pozycjach, na których bity się różnią, w wyniku otrzymujemy jedynki – rezultat niejako informuje nas na jakich pozycjach są różne wartości). Podobnie wartości XOR'rów można wyliczyć dla par na wyjściu skrzynki.

Ponieważ wejście skrzynki jest sześciobitowe, natomiast wyjście – czterobitowe, mamy 64*16 możliwych par: XOR wejściowy – XOR wyjściowy. Wynika stąd, że na każdą parę XOR wejściowy – XOR wyjściowy przypadają średnio cztery pary wartości wejściowych. Jednak nie wszystkie kombinacje XOR wejściowy – XOR wyjściowy są możliwe, a te możliwe mają nierównomierny rozkład. Tablicę ilustrującą rozkład XOR'ów wejściowych i wyjściowych wszystkich możliwych par wejściowych skrzynki nazywamy *tablicą rozkładu XOR'ów par*. Określony element takiej tablicy jest liczbą mówiącą ile jest par wejściowych o XOR'ze określonym przez numer wiersza tego elementu, które dają na wyjściu XOR wyjściowy określony przez numer kolumny. Przykład fragmentu takiej tablicy – dla skrzynki S1 algorytmu DES – przedstawiony jest na rysunku 1.

**Przykład 1.** Weźmy XOR wejściowy 34 (stosujemy zapis szesnastkowy). Z tablicy z rysunku 1 wynika, że dla takiego XOR'u wejściowego możliwymi wartościami XOR'u na wyjściu są: 1, 2, 3, 4, 7, 8, D i F. XOR wyjściowy równy 1 jest uzyskiwany dla 8 par, 2 – dla 16 par, 3 – dla sześciu, itd..

**Definicja 1.** Niech *dX* oznacza sześciobitowy XOR wejściowy, zaś *dY* czterobitowy XOR wyjściowy skrzynki S. Mówimy, że *dX* może spowodować *dY* dla skrzynki s, jeżeli istnieje co najmniej jedna para wejściowa o XOR'ze równym *dX*, dla którego XOR wyjściowy skrzynki S jest równy *dY*. Fakt, że *dX* może spowodować *dY* oznaczamy: *dX*->*dY*.

**Przykład 2.** Z tablicy z rysunku 1 wynika, że dla skrzynki S1, XOR wejściowy 34 może spowodować XOR wyjściowy 2 z prawdopodobieństwem ¼ - ponieważ sytuacja taka jest dla 16 spośród 64 możliwych par wejściowych o XOR'ze równym 34.

| | **XOR wyjściowy** | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **XOR wejściowy** | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **A** | **B** | **C** | **D** | **E** | **F** |
| **0** | 64 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **1** | 0 | 0 | 0 | 6 | 0 | 2 | 4 | 4 | 0 | 10 | 12 | 4 | 10 | 6 | 2 | 4 |
| **2** | 0 | 0 | 0 | 8 | 0 | 4 | 4 | 4 | 0 | 6 | 8 | 6 | 12 | 6 | 4 | 2 |
| **3** | 14 | 4 | 2 | 2 | 10 | 6 | 4 | 2 | 6 | 4 | 4 | 0 | 2 | 2 | 2 | 0 |
| **4** | 0 | 0 | 0 | 6 | 0 | 10 | 10 | 6 | 0 | 4 | 6 | 4 | 2 | 8 | 6 | 2 |
| **5** | 4 | 8 | 6 | 2 | 2 | 4 | 4 | 2 | 0 | 4 | 4 | 0 | 12 | 2 | 4 | 6 |
| **…** | | | | | | | | | | | | | | | | |
| **30** | 0 | 4 | 6 | 0 | 12 | 6 | 2 | 2 | 8 | 2 | 4 | 4 | 6 | 2 | 2 | 4 |
| **31** | 4 | 8 | 2 | 10 | 2 | 2 | 2 | 2 | 6 | 0 | 0 | 2 | 2 | 4 | 10 | 8 |
| **32** | 4 | 2 | 6 | 4 | 4 | 2 | 2 | 4 | 6 | 6 | 4 | 8 | 2 | 2 | 8 | 0 |
| **33** | 4 | 4 | 6 | 2 | 10 | 8 | 4 | 2 | 4 | 0 | 2 | 2 | 4 | 6 | 2 | 4 |
| **34** | 0 | 8 | 16 | 6 | 2 | 0 | 0 | 12 | 6 | 0 | 0 | 0 | 0 | 8 | 0 | 8 |
| **35** | 2 | 2 | 4 | 0 | 8 | 0 | 0 | 0 | 14 | 4 | 6 | 8 | 0 | 2 | 14 | 0 |
| **…** | | | | | | | | | | | | | | | | |

**Rysunek 1. Tablica rozkładu XOR'rów par dla skrzynki S1 algorytmu DES (fragment).**

Dla skrzynek DES około 20% elementów tablic rozkładu XOR'ów par to zera.

Tablica rozkładu XOR'ów par daje nam informacje dla ilu par wejściowych o zadanym XOR'ze możliwe jest uzyskanie określonego XOR'u wyjściowego. Kolejną informacją jaka będzie nam potrzebna jest informacja jakie pary dają określoną kombinację XOR wejściowy – XOR wyjściowy.

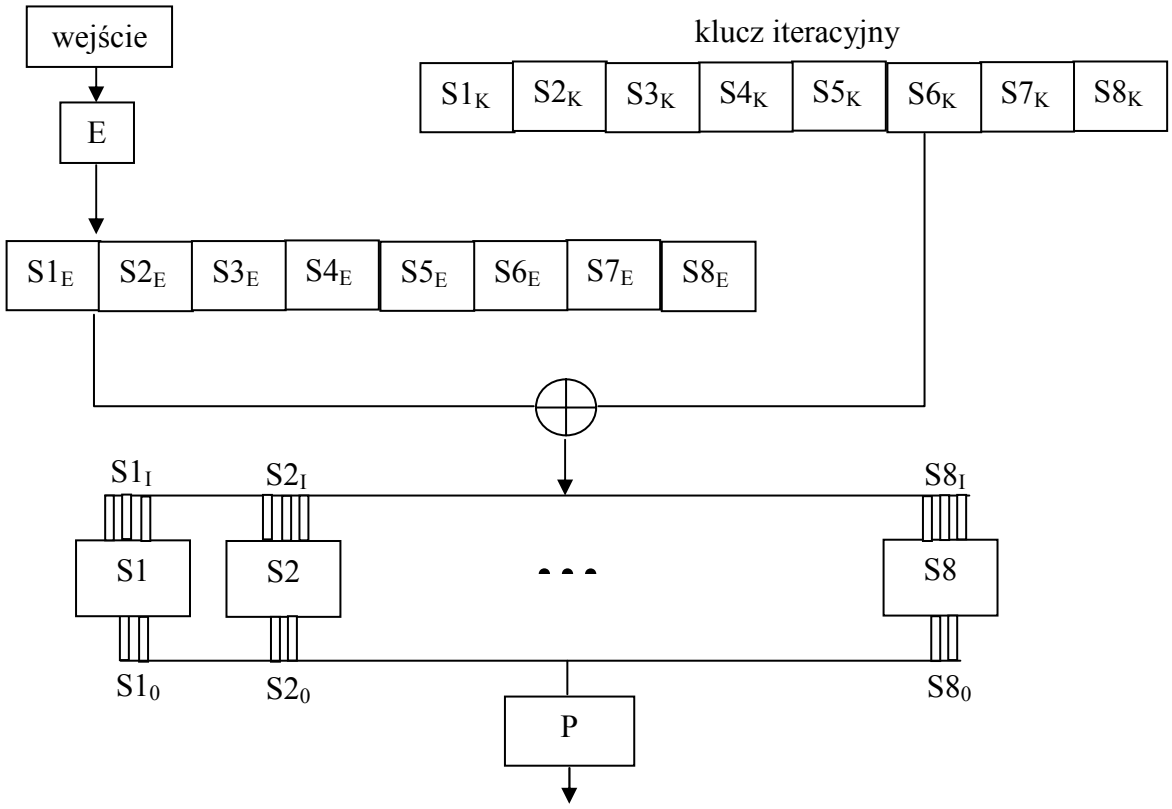
**Przykład 3.** Rozważmy element 34->4 (tzn. element z wiersza 34, kolumny 4) tablicy z rysunku 1. Ponieważ jest on równy 2, to znaczy, że istnieją dwie pary o XOR'ze 34 które na wyjściu skrzynki S1 dają XOR równy 4. Są to pary (13, 27) oraz (27, 13) (jeśli jakaś para jest rozważanym zestawie par to oczywiście zawsze jest w nim też para o elementach w odwróconej kolejności). Tablica wszystkich par wejściowych dla wiersza 34 pogrupowanych wg XOR'ów wyjściowych przedstawiona jest na rysunku 2.

Jak znaleźć bity klucza iteracyjnego (klucza jednej iteracji algorytmu) wykorzystując znajomość par wejściowych i XOR'u wyjściowego funkcji F algorytmu DES. Funkcja F

przedstawiona jest na rysunku 3. W dalszych rozważaniach wykorzystywane będą przedstawione na tym rysunku oznaczenia.

| XOR wyjściowy | XOR wejściowy |
|---|---|
| **1** | 03, 0F, 1E, 1F, 2A, 2B, 37, 3B |
| **2** | 04, 05, 0E, 11, 12, 14, 1A, 1A, 20, 25, 26, 2E, 2F, 30, 31, 3A |
| **3** | 01, 02, 15, 21, 35, 36 |
| **4** | 13, 27 |
| **7** | 00, 08, 0D, 17, 18, 1D, 23, 29, 2C, 34, 39, 3C |
| **8** | 09, 0C, 19, 2D, 38, 3D |
| **D** | 06, 10, 16, 1C, 22, 24, 28, 32 |
| **F** | 07, 0A, 0B, 33, 3E, 3F |

**Rysunek 2. Tablica par wejściowych o XOR'ze równym 34 pogrupowanych wg XOR'ów wyjściowych dla skrzynki S1**




**Rysunek 3. Funkcja F algorytmu DES.**

Poniższy przykład obrazuje jak znaleźć sześć skrajnie lewych bitów klucza iteracyjnego, tzn. 6 bitów znajdujących się na pozycjach odpowiadających skrzynce S1. Zajmować się więc będziemy sześciobitowymi słowami $S1_E$, $S1_K$, $S1_I$, $S1_0$ – patrz rysunek 3.

**Przykład 4.** Załóżmy, że mamy następującą parę wejściową (po operacji rozszerzenia):

$$S1_E = 1,\ S1_E^* = 35$$

XOR tej pary wynosi więc $dS1_E = 1 \oplus 35 = 34$. Załóżmy także, że zastosowano klucz iteracyjny $S1_K = 23$ (oczywiście celem kryptoanalizy jest znalezienie tego klucza; potraktujemy go więc jako wartość nieznaną i będziemy starali się odtworzyć).

Ponieważ wartości na wejściu skrzynki są rezultatem XOR'u wartości na wyjściu operacji E i wartości klucza mamy:

$$S1_I = S1_E \oplus S1_K = 1 \oplus 23 = 22$$

$$S1_I^* = S1_E^* \oplus S1_K = 35 \oplus 23 = 16$$

Wartości tych nie można jednak znaleźć nie znając wartości klucza.

Zauważmy, że XOR na wejściu skrzynki jest ***taki sam*** jak XOR na wyjściu operacji E:

$$S1_I \oplus S1_I^* = (S1_E \oplus S1_K) \oplus (S1_E^* \oplus S1_K) = S1_E \oplus S1_E^* = 34.$$

Wynika stąd, że XOR wartości pary z kluczem nie zmienia XOR'u w parze. Tak więc choć nie znamy wartości $S1_I$ i $S1_I^*$ (założyliśmy, że nie mamy klucza; jego znalezienie jest celem niniejszej analizy) znamy ich XOR! Jest to ważna informacja, która dalej będzie wykorzystywana.

Dla wartości $S1_I = 22$ i $S1_I^* = 16$ na wejściu skrzynki S1 na wyjściu otrzymujemy odpowiednio $S1_0 = 1$ i $S1_O^* = C$. Tak więc XOR na wyjściu skrzynki S1 wynosi:

$$dS1_0 = S1_0 \oplus S1_o^* = 1 \oplus C = D.$$

Podsumowywując: zakładamy, iż znamy następujące wartości:

- $S1_E = 1$ i $S1_E^* = 35$, tzn. wartości pary wejściowej (a więc i ich XOR równy 34)
- $S1_0 = D$, tzn. XOR na wyjściu skrzynki.

Jak powyższe informacje wykorzystać dla znalezienia klucza?

Ponieważ znamy XOR wejściowy skrzynki S1 – równy 34 (jak zostało pokazane, jest on równy XOR'owi na wyjściu operacji E) – i jej XOR wyjściowy – równy D – to z tabeli z rysunku 2 możemy odczytać wszystkie możliwe pary wejściowe. Zgodnie z tabelą z rysunku 1 jest ich 8. Są to pary:

| $S1_I, S1_I^*$ | $S1_I, S1_I$ |
|---|---|
| (6,32) | (16, 22) |
| (32, 6) | (22, 16) |
| (10, 24) | (1C, 28) |
| (24, 10) | (28, 1C) |

Wiemy, że:

$S1_I = S1_I \oplus S1_K$ oraz $S1_I^* \oplus S1_E$

Tak więc klucz możemy obliczyć na jeden z dwóch sposobów:

(a) $S1_K = S1_I \oplus S1_E$ lub

(b) $S1_K = S1_I^* \oplus S1_E^*$

Zastosujemy dowolny z nich, np. sposób (a), do wszystkich możliwych par na wejściu skrzynki. Otrzymamy:

| | |
|---|---|
| $6 \oplus 1 = 7$ | $16 \oplus 1 = 17$ |
| $32 \oplus 1 = 33$ | $22 \oplus 1 = 23$ |
| $10 \oplus 1 = 11$ | $1C \oplus 1 = 1D$ |
| $24 \oplus 1 = 25$ | $28 \oplus 1 = 29.$ |

Każda para daje klucz, który może być poszukiwanym kluczem właśnie. W powyższy sposób wygenerowaliśmy, tzw. zbiór kluczy potencjalnych, w którym na pewno jest klucz właściwy (jest nim klucz 23).

Powstaje pytanie jak ze zbioru kluczy potencjalnych wyłonić klucz właściwy? Aby to zrobić stosujemy inne pary wejściowe (o takim samym lub innym XOR'ze). Każda z takich par wygeneruje nam swój zbiór kluczy potencjalnych. W każdym z takich zbiorów będzie poszukiwany klucz właściwy. Identyfikujemy go znajdując przecięcie zbiorów kluczy właściwych.

**Przykład 5.** Zastosujmy parę $S1_E$ =21, $S1_E^*$ =15 – dla tego samego klucza $S1_K$ = 23. Na wyjściu skrzynki otrzymamy $S1_0$ = 4 i $S1_0^*$ = 7. Tak więc $dS1_E$ =34, $dS1_0$ = 3. Z tabeli z rysunku 2 wynika, że możliwymi parami na wyjściu skrzynki S1 są:

(1, 35), (35, 1), (2, 36), (36, 2), (15, 21), (21, 5).

W sposób analogiczny do opisanego w przykładzie 4 generujemy zbiór kluczy potencjalnych. Zawiera on klucze: 3, 37, 0, 34, 17, 23. Dwa spośród tych kluczy znajdują się również w zbiorze kluczy potencjalnych z przykładu 4. Wyróżnienia klucza właściwego spośród dwóch 17 i 23 można wykonać stosując parę wejściową o XOR'ze innym niż 34.

Definicja 2 jest uogólnieniem definicji 1dla funkcji F algorytmu DES.

**Definicja 2.** Niech dX oznacza 32-bitowy XOR wejściowy funkcji F, zaś dY – jej 32-bitowy XOR wyjściowy. Mówimy, że dX może spowodować dY z prawdopodobieństwem p dla funkcji F, jeżeli dla odsetka p wszystkich par wejściowych, o XOR'ze równym dX zaszyfrowanych przy użyciu wszystkich możliwych wartości klucza iteracyjnego, XOR wyjściowy wynosi dY. Jeśli p>0 to powyższy fakt oznacza: dX->dY.

**Lemat 1.** Prawdopodobieństwo p tego, że dX->dY dla funkcji F jest równy:

$$p = \prod_{i=1}^{8} pi$$

gdzie pi jest prawdopodobieństwem tego, że $dX_i$->$dY_i$ dla skrzynki $S_i$, i = 1, 2,…, 8,

oraz $X_1X_2X_3X_4X_5X_6X_7X_8 = E(X)$, $Y_1Y_2Y_3Y_4Y_5Y_6Y_7Y_8 = P^{-1}(Y)$;

Powyższe rozważania dotyczące znajdowania bitów klucza iteracyjnego odpowiadających pojedynczej skrzynce można rozszerzyć do znajdowania większych fragmentów lub całego klucza iteracyjnego. W kryptoanalizie różnicowej najpierw znajdowane są bity klucza ostatniej iteracji.

Rozważmy ostatnią rundę algorytmu DES – rysunek 4.Szyfrogram oznaczamy przez C, jego lewą i prawą połowę odpowiednio przez l i r, wyjście funkcji F - przez U, zaś wejście funkcji F – przez W.

Znamy wejście ostatniej rundy, ponieważ jest ono prawą połową szyfrogramu. Mamy więc pary wejściowe ostatniej rundy (uzyskane w wyniku zaszyfrowania odpowiednich tekstów jawnych, co będzie opisane dalej). Do przeprowadzenia analizy takiej, jak opisano w przykładzie 4 potrzebna jest nam jeszcze znajomość XOR'u wyjścia funkcji F tzn. wartość dU. Wartość dU można obliczyć wykorzystując znajomość XOR'u lewej połowy szyfrogramu i wartość dW:

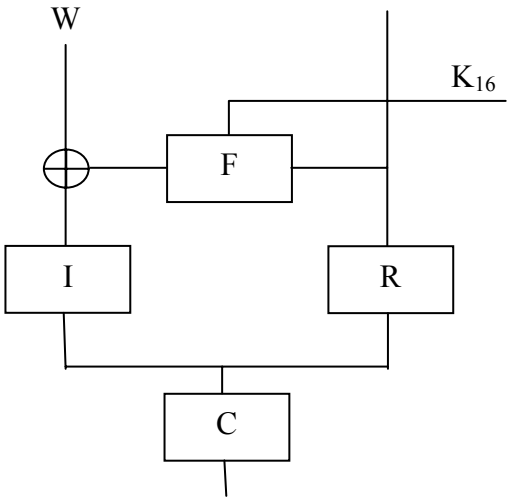
$$DU = dl \oplus dW.$$

Jak jednak znaleźć dW? Wartość tę możemy wyznaczyć z pewnym prawdopodobieństwem wykorzystując tzw. charakterystykę różnicową. Temu zagadnieniu poświęconych będzie kilka następnych zdań.

**Definicja 3** (nieformalna). Z każdą parą szyfrowań związane są:

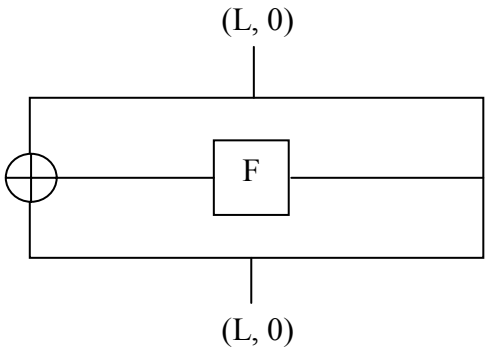
- wartość XOR'u szyfrowych tekstów jawnych,
- wartość XOR'ów otrzymanych szyfrogramów,
- wartość XOR'ów wejść odpowiednich funkcji F w dwóch szyfrowaniach,
- wartość XOR'ów wyjść odpowiednich funkcji F w dwóch szyfrowaniach.

Wymienione wartości tworzą charakterystykę różnicową obejmującą określoną liczbę rund. Charakterystyka różnicowa ma przypisane pewne prawdopodobieństwo. Jest to prawdopodobieństwo tego, że losowo wybrana para tekstów jawnych o XOR'ze zadanym przez charakterystykę (pierwsza z wymienionych powyżej wartości) ma dalsze wartości XOR'ów, tzn. XOR wyjścia funkcji F pierwszej rundy, XOR'y wejść i wyjść funkcji F dalszych rund i XOR szyfrogramu, zgodne z określonymi w charakterystyce.




**Rysunek 4. Ostatnia runda algorytmu DES.**

**Przykład 6.** Rysunek 5 przedstawia prostą, jednorundową charakterystykę o prawdopodobieństwie równym 1.




**Rysunek 5. Jednorundowa charakterystyka o prawdopodobieństwie równym 1**

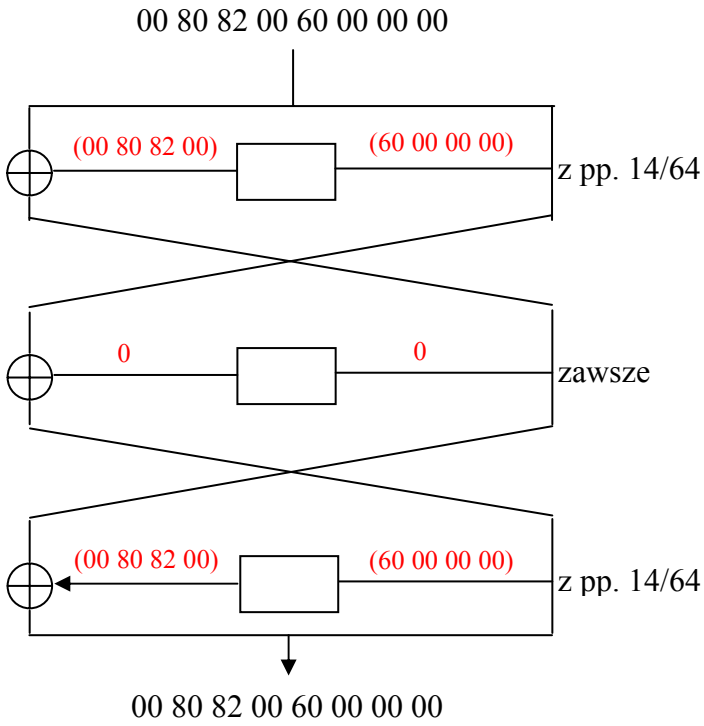
Poniższa definicja wyjaśnia wartość prawdopodobieństwa charakterystyki przedstawionej w przykładzie 6.

**Definicja 4.** Prawdopodobieństwo charakterystyki jednorundowej jest prawdopodobieństwem tego, że XOR wejściowy funkcji F założony w charakterystyce spowoduje XOR wyjściowy funkcji F założony w charakterystyce.

**Definicja 5.** Prawdopodobieństwo charakterystyki wielorundowej jest iloczynem prawdopodobieństw jej składowych charakterystyk jednorundowych.

**Przykład 7.** Na rysunku 6 przedstawiono trzyrundową charakterystykę o prawdopodobieństwie $14/64*14/64 \approx 0{,}05$.

Parą właściwą w odniesieniu do pewnej n-rudowej charakterystyki nazywamy parę tekstów jawnych, która daje w kolejnych n-rundach wartości odpowiednich XOR'ów zgodne z charakterystyką.




**Rysunek 6. Trzyrundowa charakterystyka o prawdopodobieństwie ok. 0,05**

Na drodze eksperymentów Biham i Shamir stwierdzili, że formalnie zdefiniowane definicjami 4 i 5 prawdopodobieństwo charakterystyki różnicowej jest zwykle bardzo dobrą aproksymacją prawdopodobieństwa tego, że podana na wejście algorytmu para tekstów jawnych o XOR'ze równym XOR'owi tekstu jawnego charakterystyki okaże się parą właściwą dla zastosowanego (poszukiwanego) klucza.

Istnieje grupa charakterystyk posiadających tę własność, iż XOR ich szyfrogramów jest po zmianie miejscami połówek równy XOR'owi tekstów jawnych. Charakterystyki wykazujące taką własność nazywamy *charakterystykami iteracyjnymi*. Należą one do najbardziej użytecznych charakterystyk w kryptoanalizie różnicowej. Charakterystyka iteracyjna może być wielokrotnie łączona sama z sobą. Zaletą takiego rozwiązania jest fakt, iż uzyskujemy stały współczynnik redukcji prawdopodobieństwa przy przedłużaniu charakterystyki, podczas gdy dla charakterystyk nieiteracyjnych mamy zwykle lawinowy spadek prawdopodobieństwa. Charakterystyki iteracyjne są więc podstawą konstruowania długich charakterystyk różnicowych o relatywnie dużych prawdopodobieństwach.

Podsumowując; charakterystyka różnicowa daje możliwość określenia XOR'u wejścia funkcji F przedostatniej rundy – czyli wartości dW, zgodnie z rysunkiem 4 – z pewnym prawdopodobieństwem. Mamy więc parę wejściową ostatniej rundy oraz określoną z pewnym prawdopodobieństwem, wartość XOR'u na wyjściu funkcji F ostatniej rundy ($dU = dW \oplus dl$, patrz rysunek 4). Możemy więc dokonać analizę opisaną w przykładzie 4. Należy jednak pamiętać, że jeżeli stosujemy charakterystykę o prawdopodobieństwie p to przeciętnie 1 para na 1/p będzie parą właściwą. Jedynie co do par właściwych mamy pewność, że w generowanym zbiorze kluczy potencjalnych jest klucz właściwy. W większości zbiorów kluczy potencjalnych generowanych przez pary nie będące parami właściwymi nie będzie klucza właściwego (przy dużej ilości danych). Wyłonienie klucza właściwego poprzez znalezienie przecięcia zbiorów kluczy potencjalnych nie będzie więc możliwa – ponieważ, przy dużych ilościach analizowanych par, prawie na pewno takie przecięcie nie będzie istniało. Identyfikowanie poszukiwanego klucza właściwego polega na zliczeniu pojawień poszczególnych kluczy. Klucz właściwy będzie należał do kluczy występujących najczęściej. Ponieważ prawdopodobieństwo charakterystyk różnicowych jest są relatywnie małe a do wyłonienia klucza właściwego potrzeba w praktyce kilka – kilkanaście jego wystąpień kryptoanaliza różnicowa algorytmu DES, na obecnym poziomie rozwoju techniki obliczeniowej, wciąż należy do zadań kosztownych.

Często jest tak, że efektywna kryptoanaliza różnicowa jaką dysponujemy dla danego algorytmu pozwala na znalezienie 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 (np. w przypadku DES) oznacza to często znalezienie pewnej, dość istotnej liczby bitów klucza głównego. Pozostałe bity znaleźć można stosując metody przeszukiwania wyczerpującego (ataki brutalne).

Jednak w przypadku algorytmów, których klucze iteracyjne są niezależne, w przypadku znalezienia części klucza ostatniej iteracji, niemożliwe jest zastosowanie powyższego rozwiązania. W takich przypadkach stosuje się inne charakterystyki różnicowe w celu znalezienia brakującej części klucza ostatniej iteracji. Po uzyskaniu całego klucza wykonywane jest deszyfrowanie ostatniej iteracji (ponieważ znany jest jej klucz) i wykonywany jest analogiczny atak różnicowy tylko, że na algorytm skrócony o jedną iterację.

### 1.4.1 Rezultaty kryptoanalizy różnicowej algorytmu DES

Na kryptoanalizę algorytmu DES składają się dwa etapy:

- etap gromadzenia danych,
- etap analizy zebranych danych.

Pierwszy etap polega na szyfrowaniu dużej liczby par tekstów jawnych przy użyciu nieznanego klucza. Złożoność całego procesu kryptoanalizy określona jest przez złożoność etapu gromadzenia danych (etap szyfrowań), ponieważ jest on zadaniem znacznie bardziej pracochłonnym niż etap drugi. W tabeli na rysunku 7 przedstawione są oszacowania złożoności obliczeniowej kryptoanalizy dla algorytmu DES o różnej liczbie iteracji.

| Liczba rund | Klucze zależne | Klucze niezależne | Przybliżony czas (dla kluczy zależnych) |
|---|---|---|---|
| 4 | $2^3$ | $2^4$ | 0,8 μs |
| 6 | $2^8$ | $2^8$ | 25 μs |
| 8 | $2^{14}$ | $2^{16}$ | 1,6 ms |
| 9 | $2^{24}$ | $2^{26}$ | 1,6 s |
| 10 | $2^{24}$ | $2^{35}$ | 1,6 s |
| 11 | $2^{31}$ | $2^{36}$ | 3,5 min |
| 12 | $2^{31}$ | $2^{36}$ | 3,5 min |
| 13 | $2^{39}$ | $2^{44}$ | 15 h |
| 14 | $2^{39}$ | $2^{51}$ | 15 h |
| 15 | $2^{47}$ | $2^{52}$ | 163 dni |
| 16 | $2^{47}$ | $2^{61}$ | 163 dni |

**Rysunek 7. Złożoność obliczeniowa kryptoanalizy algorytmu DES.**

Przy oszacowaniu czasu założono wykorzystanie procesora taktowanego 100 MHz, wykonującego jedno szyfrowanie w 10 taktach zegara. Innymi słowy założono szybkość szyfrowania równą 10 milionów szyfrowań na sekundę.

Z przedstawiony zgrubnych oszacowań czasu kryptoanalizy wynika, że złamanie szyfru o małej liczbie iteracji może być wykonany bardzo szybko. Kryptoanaliza pełnego 16-rundowego DES wymaga wykonania analizy $2^{36}$ szyfrogramów uzyskanych z większej puli $2^{43}$ wybranych tekstów jawnych. Faza analizy danych (druga z wyżej wymienionych) wymaga $2^{37}$ czasu (ok. 1000 razy mniej niż faza przygotowani szyfrogramów) i niewielkiej ilości pamięci.

W tabeli na rysunku 7 przedstawiono także złożoność kryptoanalizy różnicowej DES w przypadku zastosowania kluczy niezależnych. Jak widać złożoność tej analizy jest mniejsza niż złożoność ataku wyczerpującego dla algorytmu ograniczonego do 15 i mniej rund.

## 1.5 Kryptoanaliza liniowa

Liniowa kryptoanaliza jest dziś najbardziej efektywną metodą kryptoanalizy DES. Wymaga średnio $2^{43}$ par tekst jawny-kryptogram dla znalezienia klucza. Metoda ta została odkryta i opublikowana w latach dziewięćdziesiątych przez Mitsuru Matsui'ego. Inaczej jak w przypadku kryptoanalizy różnicowej, nie była brana pod uwagę przy projektowaniu algorytmu DES. Kryptoanaliza liniowa może być użyta jako:

1) atak ze znanym tekstem jawnym,
2) atak z tekstem zaszyfrowanym.

Jest to atak, który wykorzystuje zależności miedzy bitami danych wejściowych rundy, bitami klucza oraz wynikami rundy. Oczywiście, związki te zachodzą tylko statystycznie dla stosunkowo dużej lub stosunkowo małej liczby danych wejściowych. Kluczową rolę odgrywają tu formuły aproksymujące linowo pojedyncze S-boksy.

### 1.5.1 Liniowa aproksymacja

Funkcja boolowska

$$l: \Sigma^n \rightarrow \Sigma$$

określona dla $n$ zmiennych $s_1, \ldots, s_n$ jest liniowa jeżeli może być przedstawiona w formie:

$$l = a_1 s_1 \oplus a_2 s_2 \oplus \ldots \oplus a_n s_n$$

gdzie $a_i$ są stałymi boolowskimi.

Zbiór wszystkich funkcji boolowskich określonych dla $n$ zmiennych jest zdefiniowany jako:

$$L_n = \{l: \Sigma^n \rightarrow \Sigma \mid l = a_1 s_1 \oplus a_2 s_2 \oplus \ldots \oplus a_n s_n\}$$

Zbiór afiniczny jest to

$$A_n = L_n \cup \{l \oplus 1 | l \in L_n = L_n \cup \overline{L_n}\}$$

Tablica prawdy funkcji boolowskiej $f: \Sigma^n \rightarrow \Sigma$ może być jednoznacznie przedstawiona w postaci wektora

$$F = (f(0), f(1), \ldots, f(2^n - 1))$$

Odległość Hamminga $d(f, g)$ pomiędzy dwiema funkcjami $f, g$: $\Sigma^n \rightarrow \Sigma$ wyraża liczbę jedynek w wektorze

$$(f(0) \oplus g(0), f(1) \oplus g(1), \ldots f(2^n - 1) \oplus g(2^n - 1))$$

Nieliniowość $N(f)$ funkcji boolowskiej $f: \Sigma^n \rightarrow \Sigma$ jest zdefiniowana jako

$$N(f) = \min_{l \in An} d(l, f)$$

$(n \times m)$ *S-box* $S: \Sigma^n \rightarrow \Sigma^m$ jest to kolekcja funkcji boolowskich

$$f_i : \Sigma^n \rightarrow \Sigma$$

dla $i = 1, \ldots, m$.

Nieliniowość *(n* x *m) S-box'u*

$$N(S) = min_{w \in \Sigma n,\ v \in \Sigma} N(w_1 f_1 \oplus \ldots \oplus w_m f_m \oplus v)$$

Rozważmy $f: \Sigma^2 \rightarrow \Sigma$ w postaci $f(s) = s_1 s_2$. Tablica prawdy i lista funkcji liniowych ze zbioru $L_2 = \{0, s_1, s_2, s_1 \oplus s_2\}$:

| $s_1 s_2$ | $f$ | 0 | $s_1$ | $s_2$ | $S_1 \oplus s_2$ | $f \oplus s_1$ | $f \oplus s_2$ | $f \oplus s_1 \oplus s_2$ |
|---|---|---|---|---|---|---|---|---|
| 00 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 01 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 |
| 10 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 |
| 11 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 |

Tak wiec $d(f, 0) = d(f, s_1) = d(f, s_2) = 1$ oraz $d(f, s_1 \oplus s_2) = 3$.

**UWAGA:**

**Znając odległość *d(f, l)* łatwo jest znaleźć odległość $d(f, l \oplus 1) = 2^n - d(f, l)$**

### 1.5.2 Profil liniowy

Algorytm DES wykorzystuje 8 *S-box'ów* $(S_1, \ldots, S_8)$. Każda skrzynka to

$$S_i: \Sigma^6 \rightarrow \Sigma^4$$

Profil liniowy skrzynki $S_i$ jest to tablica z $2^6$ wierszami i $2^4$ kolumnami. Każdy wiersz i kolumna są indeksowane liczbami szesnastkowymi (kod heksadecymalny). Wartość na przecięciu wiersza $w$ z kolumną $k$ jest to odległość Hamminga

$$D(l_w, f_k)$$

Jeśli

$$w = (w_6, w_5, w_4, w_3, w_2, w_1) \text{ oraz } k = (k_4, k_3, k_2, k_1)$$

to

$$d(l_w, f_k) = d(w_6 s_6 \oplus \ldots \oplus w_1 s_1, k_4 f_4 \oplus \ldots \oplus k_1 f_1).$$

| | Kombinacje wyjść | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $1_x$ | $2_x$ | $3_x$ | $4_x$ | $5_x$ | $6_x$ | $7_x$ | $8_x$ | $9_x$ | $A_x$ | $B_x$ | $C_x$ | $D_x$ | $E_x$ | $F_x$ |
| $0_x$ | 32 | 32 | 32 | 32 | 32 | 32 | 32 | 32 | 32 | 32 | 32 | 32 | 32 | 32 | 32 |
| $1_x$ | 32 | 32 | 32 | 32 | 32 | 32 | 32 | 32 | 32 | 32 | 32 | 32 | 32 | 32 | 32 |
| $2_x$ | 36 | 30 | 34 | 30 | 34 | 28 | 32 | 36 | 32 | 34 | 30 | 34 | 30 | 32 | 28 |
| $3_x$ | 32 | 34 | 26 | 34 | 34 | 28 | 36 | 32 | 32 | 34 | 26 | 34 | 34 | 28 | 36 |
| $4_x$ | 34 | 30 | 32 | 32 | 34 | 30 | 32 | 32 | 34 | 34 | 36 | 28 | 30 | 30 | 32 |
| $5_x$ | 30 | 30 | 36 | 32 | 22 | 38 | 36 | 32 | 30 | 42 | 32 | 28 | 34 | 30 | 28 |
| $6_x$ | 34 | 36 | 38 | 34 | 36 | 30 | 32 | 32 | 34 | 32 | 34 | 38 | 40 | 30 | 32 |
| $7_x$ | 34 | 32 | 34 | 30 | 40 | 38 | 32 | 28 | 38 | 32 | 26 | 30 | 32 | 26 | 28 |
| $8_x$ | 32 | 34 | 38 | 32 | 32 | 30 | 26 | 30 | 34 | 36 | 20 | 34 | 38 | 28 | 36 |
| $9_x$ | 36 | 26 | 34 | 32 | 36 | 38 | 38 | 26 | 34 | 32 | 36 | 30 | 38 | 40 | 36 |
| $A_x$ | 28 | 32 | 32 | 34 | 38 | 30 | 30 | 30 | 30 | 34 | 30 | 28 | 36 | 36 | 32 |
| $B_x$ | 36 | 36 | 36 | 38 | 34 | 30 | 30 | 30 | 30 | 30 | 34 | 32 | 24 | 28 | 32 |
| $C_x$ | 30 | 32 | 34 | 32 | 30 | 28 | 22 | 34 | 28 | 34 | 40 | 34 | 28 | 38 | 36 |
| $D_x$ | 38 | 32 | 34 | 32 | 30 | 36 | 22 | 30 | 32 | 30 | 36 | 30 | 40 | 26 | 32 |
| $E_x$ | 30 | 30 | 32 | 30 | 36 | 32 | 34 | 30 | 32 | 36 | 34 | 28 | 38 | 30 | 28 |
| $F_x$ | 34 | 34 | 24 | 26 | 28 | 32 | 30 | 30 | 28 | 24 | 34 | 24 | 38 | 30 | 32 |
| $10_x$ | 34 | 30 | 32 | 32 | 30 | 26 | 24 | 32 | 30 | 30 | 28 | 32 | 34 | 42 | 12 |
| $11_x$ | 30 | 34 | 32 | 28 | 30 | 34 | 36 | 28 | 30 | 30 | 32 | 40 | 38 | 30 | 28 |
| $12_x$ | 34 | 32 | 34 | 30 | 36 | 34 | 40 | 28 | 26 | 28 | 26 | 34 | 28 | 38 | 32 |
| $13_x$ | 26 | 32 | 34 | 30 | 36 | 34 | 32 | 36 | 26 | 36 | 34 | 26 | 36 | 30 | 32 |
| $14_x$ | 28 | 36 | 32 | 32 | 32 | 32 | 32 | 36 | 36 | 28 | 28 | 32 | 28 | 36 | 32 |
| $15_x$ | 36 | 32 | 28 | 28 | 36 | 24 | 24 | 32 | 32 | 28 | 36 | 40 | 36 | 32 | 36 |
| $16_x$ | 32 | 38 | 38 | 34 | 30 | 36 | 32 | 36 | 32 | 38 | 34 | 34 | 34 | 32 | 32 |
| $17_x$ | 28 | 38 | 34 | 26 | 34 | 36 | 28 | 28 | 36 | 38 | 30 | 34 | 30 | 32 | 28 |
| $18_x$ | 26 | 32 | 30 | 28 | 42 | 36 | 30 | 30 | 32 | 34 | 32 | 30 | 28 | 34 | 36 |
| $19_x$ | 34 | 36 | 26 | 32 | 30 | 36 | 30 | 38 | 40 | 38 | 36 | 42 | 32 | 34 | 28 |
| $1A_x$ | 34 | 34 | 24 | 30 | 36 | 32 | 34 | 30 | 32 | 36 | 34 | 32 | 30 | 30 | 32 |
| $1B_x$ | 30 | 26 | 36 | 38 | 32 | 32 | 30 | 26 | 24 | 32 | 34 | 36 | 38 | 34 | 32 |
| $1C_x$ | 32 | 30 | 34 | 36 | 32 | 26 | 34 | 30 | 38 | 28 | 32 | 34 | 30 | 32 | 32 |
| $1D_x$ | 28 | 34 | 26 | 40 | 32 | 34 | 30 | 22 | 34 | 40 | 40 | 30 | 30 | 32 | 28 |
| $1E_x$ | 36 | 40 | 32 | 34 | 34 | 34 | 30 | 34 | 30 | 34 | 26 | 28 | 28 | 28 | 32 |
| $1F_x$ | 28 | 40 | 24 | 34 | 26 | 26 | 30 | 30 | 34 | 30 | 30 | 24 | 32 | 32 | 28 |
| $20_x$ | 32 | 32 | 32 | 32 | 32 | 32 | 32 | 32 | 32 | 32 | 32 | 32 | 32 | 32 | 32 |
| $21_x$ | 32 | 32 | 32 | 32 | 32 | 32 | 32 | 32 | 32 | 32 | 32 | 32 | 32 | 32 | 32 |
| $22_x$ | 28 | 30 | 34 | 30 | 34 | 28 | 40 | 28 | 32 | 26 | 38 | 34 | 30 | 16 | 20 |
| $23_x$ | 32 | 34 | 34 | 26 | 34 | 36 | 28 | 32 | 32 | 34 | 34 | 34 | 26 | 28 | 36 |
| $24_x$ | 30 | 38 | 36 | 32 | 38 | 30 | 36 | 36 | 26 | 30 | 36 | 32 | 46 | 34 | 32 |
| $25_x$ | 26 | 30 | 32 | 32 | 26 | 30 | 32 | 36 | 38 | 30 | 40 | 32 | 34 | 26 | 36 |
| $26_x$ | 30 | 28 | 34 | 34 | 32 | 30 | 36 | 28 | 34 | 36 | 34 | 26 | 32 | 34 | 32 |
| $27_x$ | 22 | 32 | 30 | 38 | 36 | 38 | 28 | 32 | 38 | 20 | 34 | 34 | 32 | 38 | 28 |
| $28_x$ | 36 | 30 | 30 | 32 | 36 | 26 | 34 | 34 | 26 | 36 | 32 | 38 | 30 | 28 | 32 |
| $29_x$ | 32 | 30 | 26 | 32 | 32 | 26 | 30 | 30 | 34 | 40 | 32 | 34 | 38 | 32 | 32 |
| $2A_x$ | 32 | 36 | 40 | 26 | 26 | 26 | 38 | 26 | 30 | 34 | 34 | 40 | 28 | 36 | 28 |
| $2B_x$ | 40 | 32 | 36 | 38 | 30 | 26 | 38 | 34 | 38 | 30 | 38 | 28 | 32 | 36 | 36 |
| $2C_x$ | 30 | 28 | 38 | 32 | 38 | 32 | 26 | 34 | 36 | 30 | 36 | 34 | 28 | 26 | 32 |
| $2D_x$ | 30 | 28 | 30 | 32 | 30 | 24 | 34 | 30 | 32 | 26 | 24 | 30 | 32 | 30 | 36 |
| $2E_x$ | 38 | 34 | 28 | 38 | 36 | 36 | 30 | 22 | 24 | 32 | 30 | 36 | 30 | 34 | 32 |
| $2F_x$ | 26 | 38 | 36 | 26 | 36 | 28 | 34 | 30 | 28 | 28 | 38 | 32 | 30 | 34 | 36 |
| $30_x$ | 34 | 30 | 32 | 28 | 26 | 30 | 28 | 36 | 34 | 34 | 32 | 32 | 34 | 34 | 36 |
| $31_x$ | 30 | 34 | 32 | 32 | 34 | 30 | 32 | 32 | 34 | 34 | 36 | 32 | 30 | 30 | 28 |
| $32_x$ | 26 | 32 | 34 | 34 | 24 | 30 | 28 | 32 | 22 | 32 | 30 | 34 | 28 | 30 | 32 |
| $33_x$ | 26 | 32 | 42 | 34 | 32 | 30 | 28 | 32 | 38 | 32 | 22 | 34 | 36 | 30 | 32 |
| $34_x$ | 32 | 44 | 28 | 36 | 32 | 28 | 40 | 36 | 32 | 36 | 32 | 36 | 36 | 32 | 32 |
| $35_x$ | 24 | 32 | 32 | 40 | 28 | 36 | 32 | 32 | 28 | 28 | 32 | 36 | 36 | 28 | 36 |
| $36_x$ | 36 | 30 | 26 | 30 | 30 | 40 | 32 | 36 | 28 | 30 | 30 | 38 | 34 | 28 | 32 |
| $37_x$ | 40 | 38 | 38 | 38 | 26 | 32 | 28 | 20 | 32 | 30 | 34 | 30 | 30 | 28 | 36 |
| $38_x$ | 30 | 28 | 38 | 32 | 34 | 28 | 34 | 38 | 28 | 38 | 32 | 26 | 28 | 34 | 32 |
| $39_x$ | 30 | 40 | 34 | 28 | 38 | 28 | 26 | 30 | 28 | 34 | 36 | 30 | 32 | 34 | 32 |
| $3A_x$ | 38 | 22 | 32 | 34 | 36 | 32 | 30 | 38 | 28 | 32 | 34 | 36 | 30 | 30 | 28 |
| $3B_x$ | 34 | 38 | 36 | 42 | 32 | 40 | 34 | 42 | 28 | 28 | 34 | 32 | 30 | 34 | 28 |
| $3C_x$ | 24 | 26 | 30 | 32 | 28 | 34 | 34 | 26 | 34 | 36 | 32 | 42 | 30 | 36 | 36 |
| $3D_x$ | 28 | 30 | 30 | 28 | 28 | 34 | 30 | 34 | 22 | 32 | 32 | 30 | 30 | 28 | 32 |
| $3E_x$ | 36 | 28 | 36 | 30 | 30 | 34 | 30 | 30 | 34 | 34 | 34 | 28 | 36 | 32 | 28 |
| $3F_x$ | 28 | 28 | 28 | 46 | 38 | 26 | 30 | 34 | 30 | 38 | 30 | 32 | 32 | 28 | 32 |

**Tabela 1. Profil liniowy S-box'ów**

Na przykład:

Funkcja $f_{8x} \sim (f_4, f_3, f_2, f_1)_{1,\ 0,\ 0,\ 0} \sim f_4$ posiada swoją najlepszą liniową aproksymację przez funkcję:

$$L_{37x} \sim (s_6, s_5, s_4, s_3, s_2, s_1)_{110111} = s_6 \oplus s_5 \oplus s_3 \oplus s_2 \oplus s_1$$

Najlepszą (globalną) aproksymację można określić dla funkcji $f_{Fx}$ ponieważ

$$d(f_{Fx}, l_{10x}) = 12$$

gdzie $f_{Fx} = f_4 \oplus f_3 \oplus f_2 \oplus f_1$ oraz

$l_{10x} \sim (s_6, s_5, s_4, s_3, s_2, s_1)_{010000} = s_5$

Probabilistyczny punkt widzenia:

Dana funkcja boolowska $f: \Sigma^n \rightarrow \Sigma$ i jej liniowa aproksymacja $l$, wtedy prawdopodobieństwo, że sygnał na wyjściu będzie się różnił wynosi:

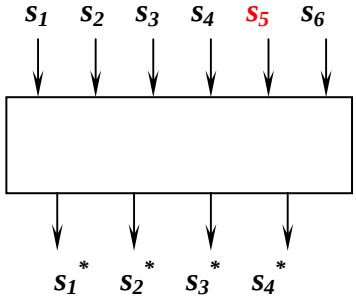
$$\frac{2^n - d(l,f)}{2^n}$$

Najgorsza aproksymacja zachodzi gdy $d(l, f) \approx 2^n/2$.

### 1.5.3 Analiza uproszczonego DES

- Atak wykorzystuje najlepszą aproksymację skrzynki $S_5$, czyli

$$s_1^* \oplus s_2^* \oplus s_3^* \oplus s_4^* = s_5$$




**Rysunek 8. Budowa S-box'ów**

- Rozważając powyższe równanie kontekście pojedynczej iteracji, mamy

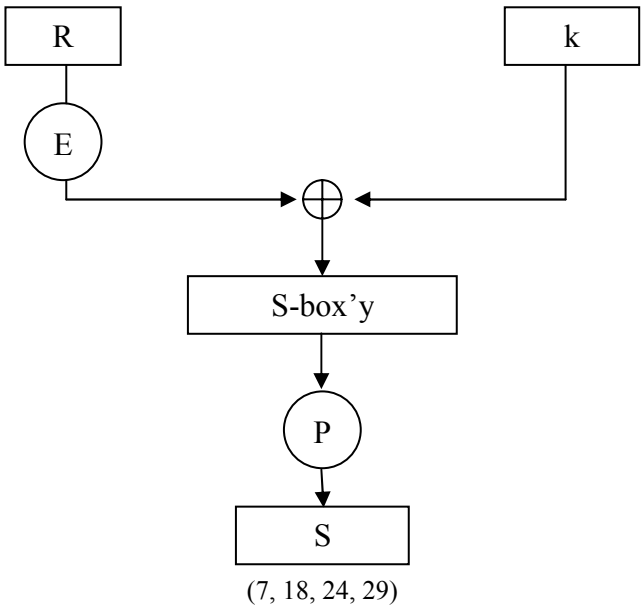
$$R_{(15)} \oplus k_{(22)} = S_{(7)} \oplus S_{(18)} \oplus S_{(24)} \oplus S_{(29)} \overset{def}{=} S_{(7,\ 18,\ 24,29)}$$

Trzy aproksymacje DES można aproksymować następującymi równaniami

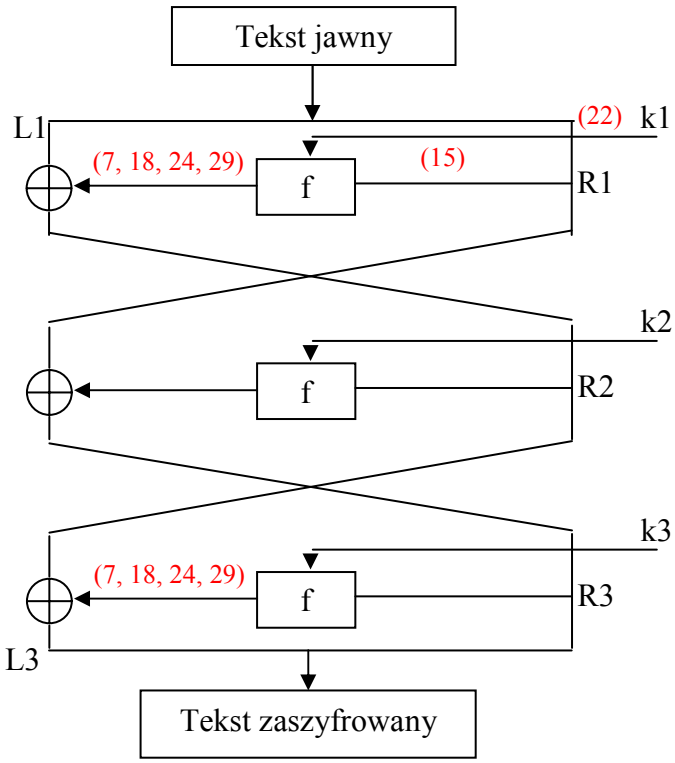
$$R2_{(7,\ 18,\ 24,\ 29)} \oplus L1_{(7,\ 18,\ 24,\ 29)} = k1_{(22)} \oplus R1_{(15)}$$

$$R2_{(7,\ 18,\ 24,\ 29)} \oplus L3_{(7,\ 18,\ 24,\ 29)} = k3_{(22)} \oplus R3_{(15)3}$$

Po połączeniu dwóch równań, otrzymujemy




**Rysunek 9. Liniowa aproksymacja pojedynczej iteracji DES**




**Rysunek 10. Liniowa aproksymacja trzech iteracji DES.**

$$L1_{(7,\,18,\,24,\,29)} \oplus L3_{(7,\,18,\,24,\,29)} \oplus R1_{(15)} \oplus R3_{(15)} = k1_{(22)} \oplus k3_{(22)}$$

Jakie jest prawdopodobieństwo, że ostatnie równanie jest prawdziwe?

Równanie jest prawdziwe w dwu przypadkach:

1. jeśli obie aproksymacje są prawdziwe,
2. jeśli obie aproksymacje są fałszywe.

Każda aproksymacja jest prawdziwa prawdopodobieństwem 52/64, czyli łączna aproksymacja jest prawdziwa z prawdopodobieństwem

$$(52/64)^2 + (12/64)^2 \approx 0,7$$

Weźmy pod uwagę równanie

$$L1_{(7,\ 18,\ 24,\ 29)} \oplus L3_{(7,\ 18,\ 24,\ 29)} \oplus R1_{(15)} \oplus R3_{(15)} = k1_{(22)} \oplus k3_{(22)}$$

Możemy teraz liczyć ile razy lewa strona równania przyjmuje wartość zero a ile wartość jeden. Po wystarczająco dużej liczbie obserwacji możemy określić wartość $k1_{(22)} \oplus k3_{(22)}$

### 1.5.4 Liniowe charakterystyki

Rozważmy DES z pięcioma iteracjami jak na rysunku 11.

W pierwszej i ostatniej iteracji użyto następującej aproksymacji:

$$S_{(15)} = k_{(27)} \oplus k_{(28)} \oplus k_{(30)} \oplus k_{(31)} \oplus R_{(27)} \oplus R_{(28)} \oplus R_{(30)} \oplus R_{(31)} \overset{def}{=} k_{(27,\ 28,\ 30,\ 31)} \oplus R_{(27,\ 28,\ 30,\ 31)}$$

Bit $S_{(15)}$ jest wzięty ze skrzynki $S_1$ i posiada nieliniowość 22. Iteracje te można opisać:

$$R2_{(15)} = L1_{(15)} \oplus S1_{(15)} = L1_{(15)} \oplus k1_{(27,\ 28,\ 30,\ 31)} \oplus R1_{(27,\ 28,\ 30,\ 31)}$$

$$R4_{(15)} = L5_{(15)} \oplus S5_{(15)} = L5_{(15)} \oplus k5_{(27,\ 28,\ 30,\ 31)} \oplus R5_{(27,\ 28,\ 30,\ 31)}$$

W drugiej i czwartej iteracji używamy optymalnej aproksymacji dla $S_5$. Czyli mamy:

$$R3_{(7,\ 18,\ 24,\ 29)} = R1_{(7,\ 18,\ 24,\ 29)} \oplus S2_{(7,\ 18,\ 24,\ 29)} = R1_{(7,\ 18,\ 24,\ 29)} \oplus k2_{(22)} \oplus R2_{(15)}$$

$$R3_{(7,\ 18,\ 24,\ 29)} = R5_{(7,\ 18,\ 24,\ 29)} \oplus S4_{(7,\ 18,\ 24,\ 29)} = R5_{(7,\ 18,\ 24,\ 29)} \oplus k4_{(22)} \oplus R4_{(15)}$$

Otrzymujemy zatem:

$$R1_{(7,\ 18,\ 24,\ 29)} \oplus R5_{(7,\ 18,\ 24,\ 29)} = k2_{(22)} \oplus R2_{(22)} \oplus k2_{(22)} \oplus R4_{(22)}$$

Końcowa postać charakterystyki ma postać:

$$L1_{(15)} \oplus L5_{(15)} \oplus R1_{(7,\ 18,\ 24,\ 27,\ 28,\ 29,\ 30,\ 31)} \oplus R5_{(7,\ 18,\ 24,\ 27,\ 28,\ 29,\ 30,\ 31)} =$$

$$= k1_{(27,\ 28,\ 30,\ 31)} \oplus k2_{(22)} \oplus k4_{(22)} \oplus k5_{(27,\ 28,\ 30,\ 31)}.$$

Otrzymana charakterystyka używa czterech liniowych aproksymacji. Jakie jest prawdopodobieństwo, że charakterystyka jest słuszna?

**Twierdzenie (Matsui).** Dane $n$ niezależnych zmiennych losowych $X_1, \ldots, X_n$ z prawdopodobieństwami kreślonymi jako $P(X_i = 1) = 1\text{-}p_i$ dla $i = 1, \ldots, n$. Wtedy prawdopodobieństwo, że $X_1 \oplus \ldots \oplus X_n = 0$ wynosi:

$$\frac{1}{2} + 2^{n-1} \prod_{i=1}^{n} (p_i - 0{,}5)$$

Liniowa charakterystyka dla DES z pięcioma iteracjami jest oparta na czterech liniowych aproksymacjach z następującymi prawdopodobieństwami: 42/64, 52/64, 52/64, 42/64. A więc prawdopodobieństwo, że jest ona słuszna wynosi ≈ 0,519.

Oznacza to, że po ≈ 2800 obserwacji par (tekst jawny/kryptogram), można określić wartość $k1_{(27, 28, 30, 31)} \oplus k2_{(22)} \oplus k4_{(22)} \oplus k5_{(27, 28, 30, 31)}$.

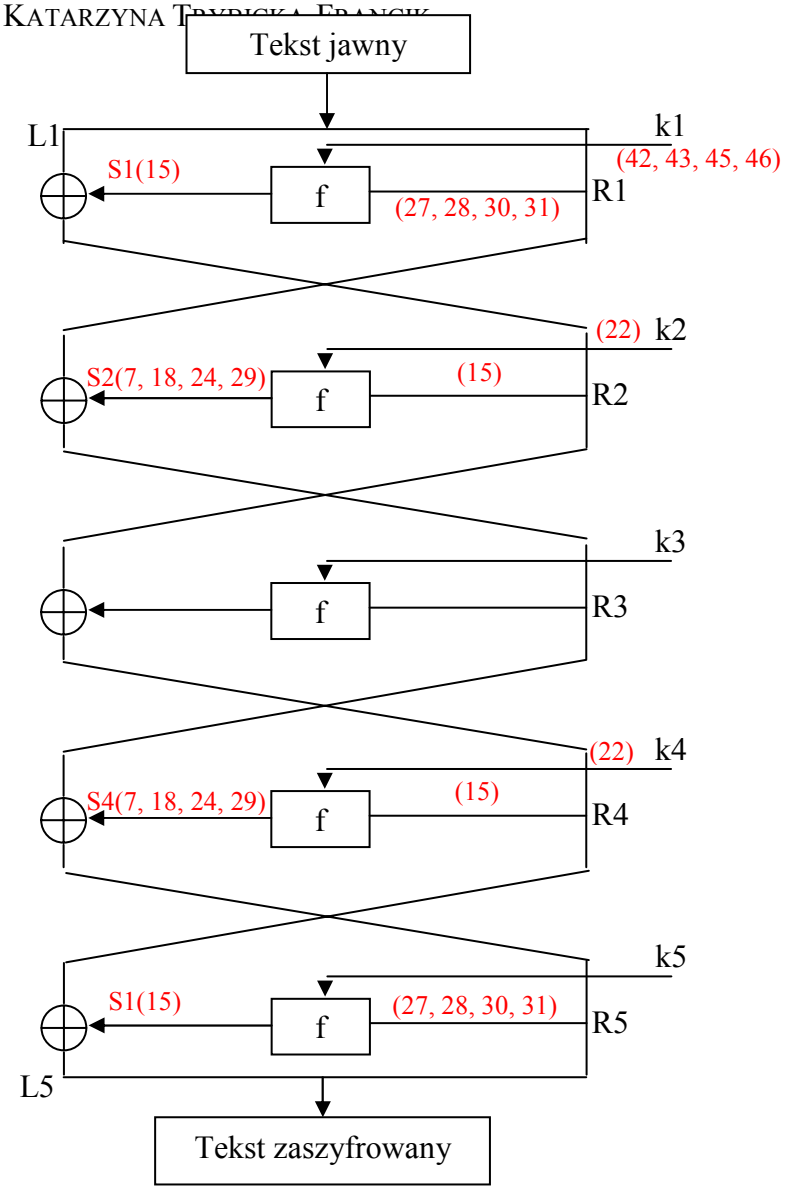
### 1.5.5 Szybszy atak liniowy

- Matsui pokazał, że atak można znacznie przyspieszyć poprzez zgadywanie bitów klucza zastosowanego w pierwszej i ostatniej iteracji.
- Jeżeli charakterystyka zależy od *m* bitów klucza (zastosowanego w pierwszej i ostatniej iteracji), to można oczekiwać zauważalnego odchylenia prawdopodobieństwa, gdy bity są poprawnie odgadnięte (wyliczanie można prowadzić równolegle na tych samych obserwacjach).

### 1.5.6 Atak na DES

- Do analizy wystarczają dwie linowe charakterystyki. Druga charakterystyka jest uzyskana z pierwszej poprzez zastąpienie bitów wejściowych z bitami wyjściowymi.
- Charakterystyki aproksymują wszystkie iteracje oprócz pierwszej i ostatniej gdzie staramy się zgadnąć 12 bitów klucza (13 bit otrzymujemy z charakterystyki).
- Ponieważ atak używa dwóch charakterystyk, więc można otrzymać 26 bitów klucza. Resztę 30 bitów można wyznaczyć stosując atak wyczerpujący.

Aby ustrzec się przed atakiem liniowym, należy stosować S-boxy z maksymalną nieliniowością oraz dużą liczbę iteracji.




**Rysunek 11. Liniowe charakterystyki – DES z pięcioma iteracjami.**