

## Spis treści

<b>1 Złożoność obliczeniowa</b>	<b>2</b>
1.1 Dodawanie . . . . .	2
1.2 Mnożenie . . . . .	2
1.3 Potęgowanie . . . . .	2
1.4 Dzielenie . . . . .	2
1.5 Modulo . . . . .	2
1.6 Znajdowanie odwrotności . . . . .	2
<b>2 Struktury algebraiczne</b>	<b>2</b>
2.1 Podgrupa . . . . .	2
2.2 Generatory . . . . .	3
2.3 Warstwy . . . . .	3
2.4 Homomorfizmy . . . . .	3
2.5 Symbol Lagrange'a . . . . .	3
2.6 Ciało p-elementowe . . . . .	3
2.7 Krzywe eliptyczne . . . . .	4
2.7.1 Twierdzenie Hesse'go . . . . .	4
2.7.2 Dodawanie . . . . .	4
2.7.3 Potęgowanie . . . . .	4
2.7.4 Odwracanie . . . . .	4
2.7.5 Element Neutralny . . . . .	4
2.7.6 Generowanie krzywej . . . . .	4
2.7.7 Generowanie punktu . . . . .	5
<b>3 Szyfr Shannona</b>	<b>5</b>
3.1 Szyfr XOR . . . . .	5
3.2 Bezpieczeństwo doskonale . . . . .	5
<b>4 Problemy</b>	<b>5</b>
4.1 Problem logarytmu dyskretnego (DL) . . . . .	5
4.2 Problem DDH . . . . .	5
4.3 CDH . . . . .	6
<b>5 Schematy</b>	<b>6</b>
5.1 Protokół DH . . . . .	6
5.2 Schemat szyfrowania z kluczem publicznym . . . . .	6
<b>6 Ataki</b>	<b>6</b>
6.1 Man-in-the-middle . . . . .	6
6.2 Bezpieczeństwo semantyczne . . . . .	7
6.3 Atak CDA . . . . .	7
<b>7 RSA</b>	<b>7</b>
7.1 Definicja . . . . .	8
7.2 Trudność problemu . . . . .	8
7.3 Przykład . . . . .	8
<b>8 Funkcja Hashująca</b>	<b>8</b>
<b>9 ElGamal</b>	<b>8</b>
<b>10 DES</b>	<b>8</b>
10.1 Enkrypcja . . . . .	9
10.2 Generowanie kluczy . . . . .	9
10.3 Funkcja F (Feistel) . . . . .	9
10.4 Bezpieczeństwo . . . . .	9

# 1 Złożoność obliczeniowa

## 1.1 Dodawanie

Dodanie dwóch liczb binarnych  $a$  i  $b$  o długości  $n$  ma złożoność  $O(n)$ , lub lepiej  $O(\log \max(a, b))$ .

## 1.2 Mnożenie

Mnożenie dwóch liczb binarnych  $a$  i  $b$  o długości  $n$  ma złożoność  $O(n^2)$ , lub lepiej  $O(\log^2 \max(a, b))$ .

## 1.3 Potęgowanie

Potęgowanie liczby  $a$  do potęgi  $b$  ma złożoność  $O(\log^b a)$ .

## 1.4 Dzielenie

Dzielenie liczby  $a$  przez  $b$  ma złożoność  $O(n^2)$ .

## 1.5 Modulo

Modulo liczby  $a$  przez  $b$  ma złożoność  $O(n^2)$ .

## 1.6 Znajdowanie odwrotności

To zależy od grupy, ale dla  $a$  w przypadku  $Z_n$  wymaga obliczenia  $n - a$ , czyli  $O(\log \max(a, n))$ . W przypadku  $Z_n^\times$  wymaga użycia rozszerzonego algorytmu Euklidesa. Ten wykonuje w najgorszym przypadku  $a$  iteracji, więc złożoność wynosi  $O(a \log^2 \max(a, n))$ .

# 2 Struktury algebraiczne

$$1. \forall_{a,b \in G} a * (b * c) = (a * b) * c$$

$$2. \forall_{a,b \in G} a * b = b * a$$

$$3. \exists_{e \in G} \forall_{a \in G} a * e = a$$

$$4. \forall_{a \in G} a^{-1} = e$$

- półgrupa: 1

- monoid: 1, 3

- grupa: 1, 3, 4

- grupa abelowa: 1, 2, 3, 4

Zawsze istnieje tylko jeden element neutralny operacji. Rzędem grupy jest moc zbioru  $G$ .

$$\varphi(n) = |\{a \in \mathbb{Z}_n : \gcd(a, n) = 1\}|$$

## 2.1 Podgrupa

Niech  $H$  będzie podgrupą grupy  $G$ . Wtedy:

$$\forall_{a,b \in H} a * b \in H$$

$$\forall_{a \in H} a^{-1} \in H$$

Na przykład, dla  $\mathbb{Z}_{10} = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$ ,  $H = \{0, 2, 4, 6, 8\}$  jest podgrupą grupy  $\mathbb{Z}_{10}$ .

## 2.2 Generatory

$$\langle g \rangle = \{g^k : k \in \mathbb{Z}\}$$

Grupa cykliczna, to grupa, która posiada co najmniej jednoelementowy zbiór generatorów.  $\exists_{g \in G} \langle g \rangle = G$

## 2.3 Warstwy

Dla podgrupy  $H$  grupy  $G$ , warstwą lewostronną  $H$  wyznaczoną przez  $a \in G$  jest zbiór:

$$\begin{cases} a + H = \{a + h : h \in H\} \\ aH = \{ah : h \in H\} \end{cases}$$

Warstwy są identyczne, albo rozłączne. Warstwy  $aH$  i  $bH$  są sobie równe kiedy  $a^{-1}b \in H$ . Suma mnogościowa warstw jest równa grupie  $G$ . Indeksem podgrupy  $H$  w grupie  $G$  ( $G : H$ ) nazywamy moc zbioru warstw względem podgrupy  $H$ .

$$G : H = \frac{|G|}{|H|}$$

Rząd podgrupy  $H$  jest dzielnikiem rzędu grupy  $G$ .

## 2.4 Homomorfizmy

$f : G \rightarrow G'$  nazywamy homomorfizmem grupy  $G$  w grupę  $G'$ , jeśli zachodzi:

$$f(a + b) = f(a) + f(b)$$

$$f(ab) = f(a)f(b)$$

Jeśli:

- $f$  jest iniekcją, to mówimy że  $f$  jest monomorfizmem.
- $f$  jest suriekcją, to mówimy że  $f$  jest epimorfizmem.
- $f$  jest bijekcją, to mówimy że  $f$  jest izomorfizmem.

Z własności homomorfizmu wynika, że  $f(e) = f(ee) = f(e)f(e) = e'$  oraz  $f(a^{-1}) = f(a)^{-1}$  i  $f(a)f(a^{-1}) = f(e)$ .

Zbiór  $Ker(f) = \{a \in G : f(a) = e'\}$  nazywamy jądrem homomorfizmu  $f$ .

Zbiór  $Im(f) = \{f(a) : a \in G\}$  nazywamy obrazem homomorfizmu  $f$ .

## 2.5 Symbol Lagrange'a

Liczba  $a$  w grupie  $G$  jest resztą kwadratową, jeśli istnieje  $b \in G$  takie, że  $a = b^2$ .

Symbol Lagrange'a jest zdefiniowany następująco:

$$\frac{a}{p} = \begin{cases} 1 & \text{jeśli } a \text{ jest resztą kwadratową} \\ 0 & \text{jeśli } p|a \\ -1 & \text{jeśli } a \text{ nie jest resztą kwadratową} \end{cases}$$

## 2.6 Ciało p-elementowe

Dla liczby pierwszej  $p$ :

$$\mathbb{F}_p = \{0, 1, \dots, p-1\}$$

Struktura  $(\mathbb{F}_p, +_p)$  to grupa abelowa o  $p$  elementach. Równocześnie  $(\mathbb{F}_p \setminus \{0\}, \cdot_p)$  jest grupą abelową. Ciałem  $p$ -elementowym jest  $(\mathbb{F}_p, +_p, \cdot_p)$ , gdzie:

$$\forall_{a,b,c \in \mathbb{F}_p} a(b + c) = ab + ac \wedge (b + c)a = ba + ca$$

## 2.7 Krzywe eliptyczne

Dla  $p > 3$ , krzywą eliptyczną  $E$  nad ciałem  $\mathbb{F}_p$  jest dana przez:

$$E : y^2 = x^3 + ax + b \pmod{p}$$

Krzywa eliptyczna musi mieć trzy pierwiastki, stąd:

$$\Delta_E = 4a^3 + 27b^2 \pmod{p} \neq 0$$

Punkt  $P = (x_1, y_1)$  leży na krzywej  $E/\mathbb{F}_p$  (nad  $\mathbb{F}_p$ ), jeśli spełnia równanie:

$$y_1^2 = x_1^3 + ax_1 + b \pmod{p}$$

Zatem, zbiorem wartości krzywej jest:

$$E(\mathbb{F}_p) = \{(x, y) \in \mathbb{F}_p \times \mathbb{F}_p : y^2 = x^3 + ax + b \pmod{p}\} \cup \{\mathcal{O}\}$$

### 2.7.1 Twierdzenie Hesse'go

$$\#E(\mathbb{F}_p) = p + 1 - t$$

gdzie  $t < 2\sqrt{p}$ , oraz zależy od  $E$ . Należy wspomnieć, że  $\#E$  to jej rzad oraz ilość punktów na krzywej.

### 2.7.2 Dodawanie

Dla dwóch punktów  $P = (x_1, y_1), Q = (x_2, y_2); x_1 \neq x_2$ , oraz  $R = P \oplus Q = (x_3, y_3)$ .

$$\lambda = (y_2 - y_1)(x_2 - x_1)^{-1} \pmod{p}$$

następnie:

$$x_3 = \lambda^2 - x_1 - x_2 \pmod{p}$$

$$y_3 = \lambda(x_1 - x_3) - y_1 \pmod{p}$$

### 2.7.3 Potęgowanie

Dla punktu  $P = (x_1, y_1), R = P \oplus P = (x_3, y_3)$ .

$$\lambda = (3x_1^2 + a)(2y_1)^{-1} \pmod{p}$$

następnie jak dla dodawania.

### 2.7.4 Odwracanie

$$P = (x_1, y_1)$$

$$P^{-1} = (x_1, -y_1)$$

### 2.7.5 Element Neutralny

$\mathcal{O}$  nazywamy elementem neutralnym dla grupy  $(E(\mathbb{F}_p), \oplus)$ .

$$P \oplus Q = \mathcal{O} \leftrightarrow x_1 = x_2 \wedge y_1 = -y_2$$

$$P \oplus P^{-1} = \mathcal{O}$$

### 2.7.6 Generowanie krzywej

1. Generuj  $k$ -bitową liczbę pierwszą  $p$
2. Losuj  $a, b \in \mathbb{F}_p$
3. Oblicz  $\Delta_E = 4a^3 + 27b^2$
4. Sprawdź, czy  $\Delta_E \neq 0 \pmod{p}$ , w przeciwnym razie goto 2
5. `return a, b, p`

### 2.7.7 Generowanie punktu

1. Losuj  $x \in \mathbb{F}_p$
2. Oblicz  $y^2 = x^3 + ax + b \pmod p$
3. Jeśli  $\frac{y^2}{p} = -1$  goto 1
4. **return** x, y

## 3 Szyfr Shannona

Szyfr według Shannon'a jest zdefiniowany jako:

$$\pi = (E, D) : (C, M, K)$$

gdzie schemat szyfrujący  $E$  i schemat deszyfrowania  $D$  są funkcjami:

$$E : M \times K \rightarrow C$$

$$D : C \times K \rightarrow M$$

$$D(k, E(k, m)) = m$$

### 3.1 Szyfr XOR

$$K = M = C = \{0, 1\}^L$$

$$E(m, k) = m \oplus k$$

$$D(c, k) = c \oplus k$$

### 3.2 Bezpieczeństwo doskonałe

Niech  $\pi$  będzie szyfrem Shannona. Rozważmy eksperyment losowy, w którym zmienna losowa  $K$  ma rozkład jednostajny nad  $K$ . Jeśli zachodzi:

$$\forall_{m_0, m_1 \in M} \forall_{c \in C} P(E(k, m_0) = c) = P(E(k, m_1) = c)$$

to mówimy, że szyfr  $\pi$  jest szyfrem doskonałym.

Jeśli  $\pi$  jest szyfrem doskonałym, to  $|K| \geq |M|$ .

## 4 Problemy

### 4.1 Problem logarytmu dyskretnego (DL)

Niech  $G = \langle g \rangle$ . Problemem jest znalezienie  $x$  takiego, że  $g^x = a$ . W zależności od grupy oraz jej rozmiaru, ten problem może być niezwykle trudny.

### 4.2 Problem DDH

Mamy daną grupę cykliczną  $G = \langle g \rangle$ , rzędu  $q$ , gdzie  $q$  jest liczbą pierwszą. Losujemy  $\alpha, \beta, \gamma \in \mathbb{Z}_q$ . Następnie obliczamy:

$$u = g^\alpha, v = g^\beta, w_0 = g^{\alpha\beta}, w_1 = g^\gamma$$

Celem problemu, jest odgadnięcie  $b$ , dla danego  $u, v, w_b$ .

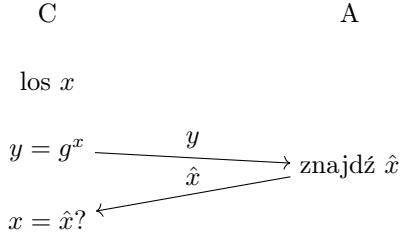


Diagram 1: Formalizm gry dla problemu logarytmu dyskretnego

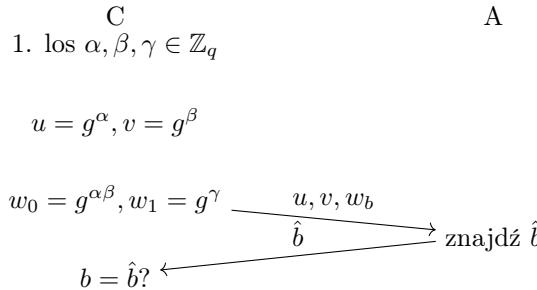


Diagram 2: Formalizm gry dla problemu DDH

### 4.3 CDH

Mamy daną grupę cykliczną  $G = \langle g \rangle$ , rzędu  $q$ , gdzie  $q$  jest liczbą pierwszą. Losujemy  $\alpha, \beta, \gamma \in \mathbb{Z}_q$ . Następnie obliczamy:

$$u = g^\alpha, v = g^\beta, w = g^{\alpha\beta}$$

Celem problemu, jest odgadnięcie  $w$ , dla danego  $u, v$ .

## 5 Schematy

### 5.1 Protokół DH

Mamy daną grupę cykliczną  $G = \langle g \rangle$ , rzędu  $q$ , gdzie  $q$  jest liczbą pierwszą. Protokół Diffie-Hellman (DH), polega na losowym wybraniu sekretów przez dwóch użytkowników (A, B)  $\alpha, \beta \in \mathbb{Z}_q$ . Obliczeniu szyfrogramów  $u = g^\alpha, v = g^\beta$ , a następnym wysłaniu  $u$  i  $v$ . Sekret wspólny  $s = g^{\alpha\beta}$ .

Protokół jest odporny na atak pasywny (tylko czytanie). Z kolei, jest podatny na atak jeśli atakujący ma wpływ na kanał komunikacji, chociażby poprzez atak Man-in-the-middle.

### 5.2 Schemat szyfrowania z kluczem publicznym

$$\varepsilon = (G, E, D) \text{ nad } (M, C, K)$$

gdzie  $G : \mathbb{N} \rightarrow K$ ,  $E : K \times M \rightarrow C$ ,  $D : K \times C \rightarrow M$ .

$$\begin{aligned} (pk, sk) &= G(\lambda) \\ C &= E(pk, m) \\ M &= D(sk, C) \end{aligned}$$

## 6 Ataki

### 6.1 Man-in-the-middle

Jeśli atakujący ma wpływ na kanał komunikacji, to może przechwycić komunikaty podczas przekazywania kluczy. W takim momencie, może się podszyć pod drugą stronę, aby uzyskać dostęp do klucza prywatnego. Równocześnie

$$\begin{array}{ccc} \text{C} & & \text{A} \\ 1. \text{ los } \alpha, \beta, \gamma \in \mathbb{Z}_q & & \end{array}$$

$$u = g^\alpha, v = g^\beta$$

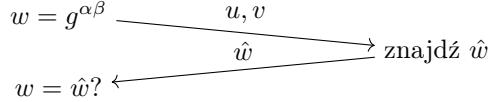


Diagram 3: Formalizm gry dla problemu CDH

Alice	Bob
-------	-----

$\alpha \in \mathbb{Z}_q$	$\beta \in \mathbb{Z}_q$
---------------------------	--------------------------

$$\begin{array}{ccc} u = g^\alpha & & v = g^\beta \\ & \diagup u, v \diagdown & \\ s = v^\alpha & \longleftrightarrow & s = u^\beta \end{array}$$

Diagram 4: Formalizm gry dla protokołu DH

może przekazywać dalej komunikację, aby ukryć swoją obecność. W ten sposób zna obydwa sekrety i tylko siedzi po środku.

## 6.2 Bezpieczeństwo semantyczne

Dla pewnego  $\varepsilon = (G, E, D)$ , atakujący ma dostęp do klucza publicznego  $pk$ . Wybiera on dwie wiadomości  $m_0, m_1 \in M$ . Przeciwnik wybiera jedną wiadomość  $b \in \{0, 1\}$ , szyfruje ją  $c = E(pk, m_b)$  i zwraca atakującemu. Atakujący musi zgadnąć  $b$ .

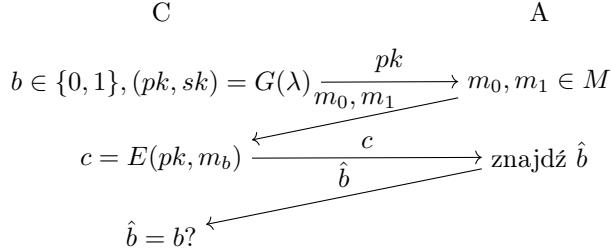


Diagram 5: Formalizm gry dla bezpieczeństwa semantycznego

## 6.3 Atak CDA

Jest to powielona wersja bezpieczeństwa semantycznego. Wielokrotnie atakujący może tworzyć wiadomości i dostawać losowy kryptogram na podstawie ich. To czyni ten atak o wiele trudniejszym niż bezpieczeństwo semantyczne.

Jeśli schemat szyfrowania kluczem publicznym jest semantycznie bezpieczny, to jest też odporny na ataki CDA.

## 7 RSA

Asymetryczny algorytm szyfrujący, w którym każda strona ma parę kluczy: publiczny i prywatny. Enkrypcja odbywa się przy pomocy klucza publicznego drugiej strony, a dekrypcja przy pomocy klucza prywatnego.

## 7.1 Definicja

Dla danych liczb pierwszych  $p$  i  $q$ .

$$\begin{aligned} n &= pq \\ \varphi(n) &= (p-1)(q-1) \end{aligned}$$

Następnie wybieramy liczbę  $e$  względnie pierwszą z  $\varphi(n)$ . Klucz prywatny  $d$  musi spełniać warunek  $ed \equiv 1 \pmod{\varphi(n)}$ , zatem

$$d = e^{-1} \pmod{\varphi(n)}$$

$(n, e)$  tworzy klucz publiczny, a  $(n, d)$  klucz prywatny.

Szyfrowanie wiadomości  $M$  odbywa się za pomocą wzoru:

$$C = M^e \pmod{n}$$

Odkrycie wiadomości  $M$  odbywa się za pomocą wzoru:

$$M = C^d \pmod{n}$$

## 7.2 Trudność problemu

Trudność wynika ze znalezienia  $\varphi(n)$ , a ponieważ weryfikacja czy znalezione  $\varphi(n)$  jest poprawne wymaga zastosowania rozszerzonego algorytmu Euklidesa; odszyfrowanie wiadomości  $C$  wymaga znalezienia  $d$ .

## 7.3 Przykład

$$\begin{aligned} p = 7, q = 11 \Rightarrow n &= 77, \varphi(n) = 60 \\ e = 13 \Rightarrow d = 37 \Rightarrow \begin{cases} (n, e) = (77, 13) \\ (n, d) = (77, 37) \end{cases} \\ M = 15 \Rightarrow C &= 15^{13} \pmod{77} = 64 \\ C = 64 \Rightarrow M &= 64^{37} \pmod{77} = 15 \end{aligned}$$

## 8 Funkcja Hashująca

$$H : X \rightarrow Y$$

gdzie  $|X| > |Y|$ . Można interpretować jako funkcję jednokierunkową, bez zapadki. Mówimy, że  $H$  jest złamana, jeśli łatwo można znaleźć przykłady  $m_0 \neq m_1$ , takie że zachodzi  $H(m_0) = H(m_1)$ .

## 9 ElGamal

Dla grupy cyklicznej  $G$  rzędu  $q$ , wykonujemy najpierw protokoł DH, a następnie szyfrowanie wiadomości  $m$  przy pomocy ustalonego sekretu  $s = g^{\alpha\beta}$  w następujący sposób:

$$\begin{aligned} c &= m \cdot s \\ m &= c \cdot s^{-1} \end{aligned}$$

Aby zapewnić bezpieczeństwo, należy wymieniać  $s$ , czyli również  $\alpha, \beta$  za każdą wiadomością.

## 10 DES

DES (Data Encryption Standard) to archetypiczny szyfr blokowy. Transformuje on, albowiem pewien blok bitów (64) w blok równej długości, uznawany za szyfrogram. DES używa klucza długości 64 bitów, ale jedynie 56 bitów jest używanych do szyfrowania.

Deszyfrowanie składa się z tej samej serii operacji, jedynie z kluczami używanymi w odwrotnej kolejności. To oznacza, że wystarczy jedna implementacja sprzętowa szyfrowania.

## 10.1 Enkrypcja

DES składa się z 16 rund, poprzedzonych permutacją wejściową ( $IP$ ), oraz zakończoną permutacją wyjściową ( $FP = IP^{-1}$ ). Blok wejściowy jest dzielony na pół, i operacje są wykonywane na obu częściach na przemian.  $F(S)$  oznacza funkcję szyfrowania, dla klucza 48 bitowego  $S$ .

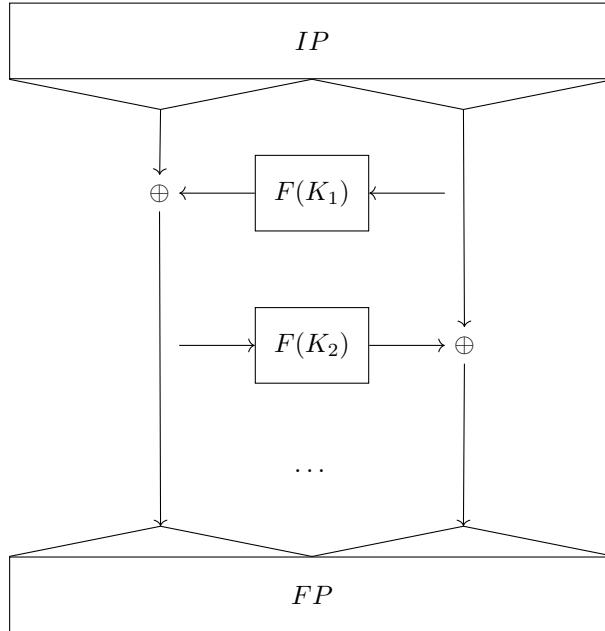


Diagram 6: Schemat szyfrowania DES

## 10.2 Generowanie kluczy

Generowanie kluczy  $S_n$  polega na kolejnych transformacjach połówek klucza początkowego  $K$ . Najpierw klucz  $K$  jest transformowany przez permutację  $PC1$  do 56 bitów i dzielony na dwie 28-bitowe połówki  $L_0$  i  $R_0$ . Następnie  $L_0$  i  $R_0$  są przesuwane o  $p(n)$  bitów w lewo, gdzie  $p(n)$  jest zależne od rundy  $n$ . Po przesunięciu,  $L_n$  i  $R_n$  są połączone i permutowane przez  $PC2$  do 48 bitów, aby uzyskać klucz  $K_n$ . Generacja  $K_n = PC2(L_{n-1} \ll p(n) | R_{n-1} \ll p(n))$ . Do dekrypcji, używamy kluczy  $K_n$  w odwrotnej kolejności, zatem najpierw  $K_{16}$ , a następnie  $K_{15}$ , itd.

## 10.3 Funkcja F (Feistel)

Najpierw nasze 32 bity wejściowe szyfrogramu  $C$  są poddawane transformacji  $E$  do 48 bitów ( $4 \cdot 6$ ). Następnie szyfrogram jest mieszany z kluczem, zatem  $I = E(C) \oplus K_n$ . Po tym,  $I$  jest dzielone na 8 bloków po 6 bitów ( $I_1, I_2, \dots, I_8$ ), które są przekazywane do odpowiednich S-boxów. S-boxy działają jak funkcje nielinowe i stanowią główne źródło bezpieczeństwa DES. Wyniki działania są transformowane przez  $P$ , dzięki czemu bity są równo dystrybuowane po całym 32-bitowym wyjściu.

$$\begin{aligned} I &= E(C) \oplus K_n \\ F(K_n) &= P(S_1(I_1) | S_2(I_2) | \dots | S_8(I_8)) \end{aligned}$$

## 10.4 Bezpieczeństwo

DES nie jest bezpieczny na dwóch frontach: jego wielkość klucza i jego własności kryptoanalytyczne. 64 bitowy klucz to po prostu za mało, zwłaszcza, że tak naprawdę klucz ma 56 bitów.  $2^{56} \sim 10^{16}$ , co więcej, klucz DES mieści się w jednym rejestrze w architekturach 64-bitowych i przeszukanie wszystkich kluczy wiąże się z jedną pętlą `for` na typie `uint64_t`.

Bardziej zaawansowane ataki, opierające się na kryptoanalizie potrafią rozwiązać problem CDA w  $2^{50}$  rund z prawdopodobieństwem 50%.