Spis treści

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 $O(\log b \log^2 a)$.

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. Złożoność wynosi $O(\log^2 \max(a, n))$.

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.

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

$$a \in \mathbb{Z}_p^\times \to a^{-1} = a^{p-2} \mod p$$

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

$f : G \to G'$ 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:

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

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:

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

2.7 Krzywe eliptyczne

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

$$E : y^2 = x^3 + ax + b \mod p$$

Krzywa eliptyczna musi mieć trzy pierwiastki, stąd:

$$\Delta_E = 4a^3 + 27b^2 \mod p \neq 0$$

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

$$y_1^2 = x_1^3 + ax_1 + b \mod p$$

Zatem, zbiorem wartości krzywej jest:

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

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 rząd 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 $R = P \oplus Q = (x_3, y_3)$.

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

następnie:

$$x_3 = \lambda^2 - x_1 - x_2 \mod p$$
$$y_3 = \lambda(x_1 - x_3) - y_1 \mod p$$

2.7.3 Potęgowanie

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

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

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 $(E(\mathbb{F}_p), \oplus)$.

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

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

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 $\Delta_E \neq 0 \mod p$, w przeciwnym razie `goto` 2
5. `return` a, b, p

2.7.7 Generowanie punktu

1. Losuj $x \in \mathbb{F}_p$
2. Oblicz $y^2 = x^3 + ax + b \mod p$
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:

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

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

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

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 π 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 π jest szyfrem doskonałym.

Jeśli π 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.

$$\begin{array}{lcl}
\text{C} & & \text{A} \\
\text{los } x & & \\
y=g^x & \xrightarrow{\quad y \quad} & \text{znajdź } \hat{x} \\
x=\hat{x}? & \xleftarrow{\quad \hat{x} \quad} &
\end{array}$$

Rysunek 1: Formalizm gry dla problemu logarytmu dyskretnego

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.

$$\begin{array}{lcl}
\text{C} & & \text{A} \\
\text{1. los } \alpha,\beta,\gamma\in\mathbb{Z}_q & & \\
u=g^\alpha, v=g^\beta & & \\
w_0=g^{\alpha\beta}, w_1=g^\gamma & \xrightarrow{\quad u,v,w_b \quad} & \text{znajdź } \hat{b} \\
b=\hat{b}? & \xleftarrow{\quad \hat{b} \quad} &
\end{array}$$

Rysunek 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.

$$
\begin{array}{lcl}
\text{C} & & \text{A} \\
1.\ \text{los } \alpha, \beta, \gamma \in \mathbb{Z}_q & & \\
u = g^{\alpha}, v = g^{\beta} & & \\
w = g^{\alpha\beta} & \xrightarrow{u, v} & \text{znajdź } \hat{w} \\
w = \hat{w}? & \xleftarrow{\hat{w}} &
\end{array}
$$

Rysunek 3: Formalizm gry dla problemu CDH

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 $s = g^{\alpha\beta}$.

$$
\begin{array}{ccc}
\text{Alice} & & \text{Bob} \\
\alpha \in \mathbb{Z}_q & & \beta \in \mathbb{Z}_q \\
u = g^{\alpha} & \xrightarrow{u} & v = g^{\beta} \\
s = v^{\alpha} & \xleftarrow{v} & s = u^{\beta}
\end{array}
$$

Rysunek 4: Formalizm gry dla protokołu DH

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

$$(pk, sk) = G(\lambda)$$
$$C = E(pk, m)$$
$$M = D(sk, C)$$

6 Ataki

6.1 Man-in-the-middle

Jeśli atakujący ma wplyw 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 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.

$$
\begin{tikzcd}
\text{C} & \text{A} \\
b \in \{0,1\}, (pk, sk) = G(\lambda) \arrow[r, "pk"] & m_0, m_1 \in M \arrow[dl, "m_0, m_1"'] \\
c = E(pk, m_b) \arrow[r, "c"] & \text{znajdź } \hat{b} \arrow[dl, "\hat{b}"'] \\
\hat{b} = b? &
\end{tikzcd}
$$

Rysunek 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.

$$n = pq$$
$$\varphi(n) = (p-1)(q-1)$$

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

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

8 Funkcja Hashująca

$$H : X \to Y$$

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 k oraz algorytmu (E, D). Zazwyczaj stosuje się dodatkowo funkcje haszującą H : $G \times G \to K$ do generowania k:

$$
\begin{gathered}
s = g^{\alpha\beta} \pmod{q} \\
k = H(u, s) \\
c = E(m, k) \\
m = D(c, k)
\end{gathered}
$$

Jeśli H jest "dobry", czyli deterministyczny, lecz zbliżony do losowej funkcji, oraz problem CDH jest trudny, to ElGamal jest bezpieczny.

10 Protokół OT

Protokół OT, pozwala na przekazanie informacji bezpiecznie, bez serwera wiedzącego o zawartości wiadomości zwrotnej.

$$
\begin{tikzcd}
\text{P} & \text{Bob} \\
\beta \in \mathbb{Z}_q, v = g^\beta \arrow[r, "v"] & \alpha \in \mathbb{Z}_q, u = g^\alpha v^{-i} \arrow[dl, "u"'] \\
u_j = uv^j, w_j = u_j^\beta & w_i = v^\alpha \\
k_j = H(v, w_j) & k = H(v, w_i) \\
c_j = E(k_j, m_j), C = (c_0, c_1, \dots) \arrow[r, "C"] & m_i = D(c_i, k)
\end{tikzcd}
$$

Rysunek 6: Schemat protokołu

$$w_i = u_i^\beta = (uv^i)^\beta = (g^\alpha v^{-i} v^i)^\beta = g^{\alpha\beta}$$

11 Szyfry blokowe

Szyfrem blokowym, nazywamy szyfr, który szyfruje blok bitów wejściowych w blok równej długości, uznawany za szyfrogram.

$$E : M \to M, D : M \to M$$

11.1 Bezpieczeństwo

Klasę permutacji, na zbiorze X, będziemy nazywać $Perms(X)$

C A

$b \in \{0, 1\}$

$f = E(k, \dots) : b = 0 \vee f \in Perms(X) : b = 1$

$x_1, x_2, \dots, x_n \in X$

$y_i = f(x_i) \xleftarrow{x_i}$

$y_i = f(x_i) \xrightarrow{y_i}$ znajduje $\hat{b} \sim b$

$b = \hat{b}? \xleftarrow{\hat{b}}$

Rysunek 7: Gra określająca bezpieczeństwo szyfru blokowego

Szyfr blokowy jest bezpieczny, jeśli $P(b = \hat{b}) \approx 1/2$

11.2 Nieprzewidywalność

C A

$k \in K$ $x_1, x_2, \dots, x_n \in X$

$y_i = E(k, x_i) \xleftarrow{x_1, x_2, \dots, x_n}$

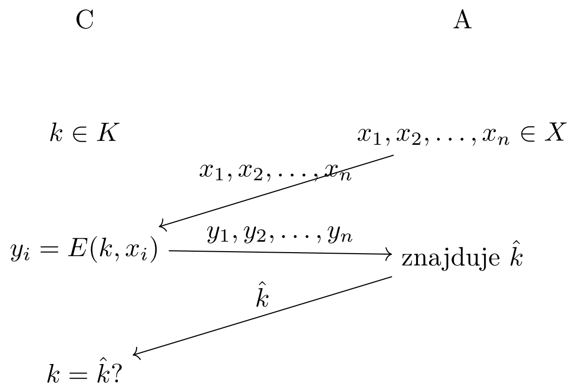
$y_i = E(k, x_i) \xrightarrow{y_1, y_2, \dots, y_n}$ znajduje $\hat{y}_{n+1}$ dla x_{n+1}

$E(k, x_{n+1}) = \hat{y}_{n+1}? \xleftarrow{\hat{y}_{n+1}}$

Rysunek 8: Gra określająca nieprzewidywalność szyfru blokowego

Jeśli szyfr blokowy jest bezpieczny, to nieprzewidywalność jest spełniona. Jeśli szyfr jest przewidywalny, to nie może być bezpieczny, ponieważ mogąc przewidzieć $\hat{y}$, możemy łatwo porównać czy $\hat{y} = E(k, x_{n+1})$.

11.3 Odzyskanie klucza



Rysunek 9: Gra określająca odzyskanie klucza szyfru blokowego

Jeśli E jest nieprzewidywalny, to jest odporny na odzyskiwanie klucza.

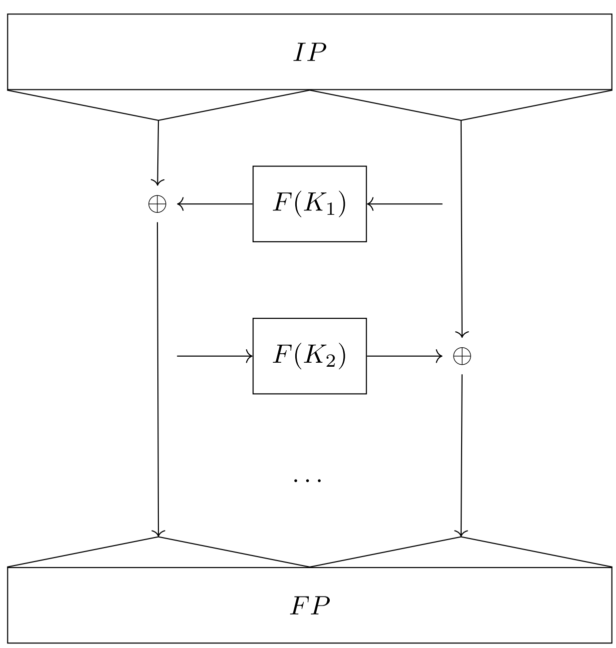
11.4 DES

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.

11.4.1 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.



Rysunek 10: Schemat szyfrowania DES

11.4.2 Generowanie kluczy

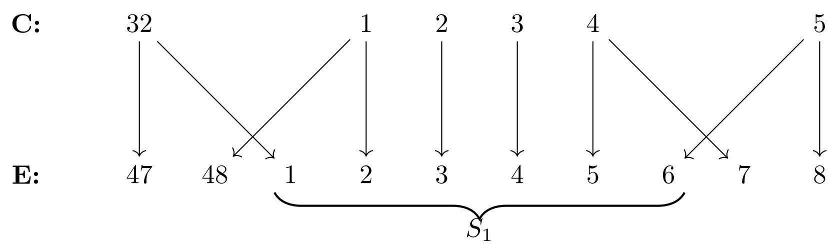
Generowanie kluczy S_n polega na kolejnych transformacji połówek klucza początkowego K. Najpierw klucz K jest transformowany przez permutację $PC1$ do 56 bitów i dzielony na dwie 28-bitowe połowy L_0 i R_0. Następnie L_0 i R_0 są przesuwane o $p(n)$ bitów w lewo, gdzie $p(n)$ jest zależne od rundy n.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

11.4.3 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 $(I_1, I_2, \ldots, I_8)$, które są przekazywane do odpowiednich S-boxów. S-boxy działają jak funkcje nieliniowe i stanowią główne źródło bezpieczeństwa DES. Wyniki działania są transformowane przez P, dzięki czemu bity są równo dystrybuowane po całym 32-bitowym wyjściu.

$$I = E(C) \oplus K_n$$
$$F(K_n) = P(S_1(I_1)|S_2(I_2)|\ldots|S_8(I_8))$$

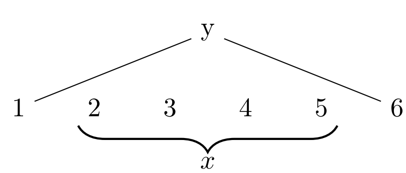
11.4.4 Permutacja E



Rysunek 11: Permutacja E

11.4.5 S-Boxy

S-Boxy, to funkcje nieliniowe, mające zapewnić większy poziom bezpieczeństwa DES. Są one definiowane przez tablicę 16×4. Indeksowana jest przez x i y, gdzie $x \in \{0, 1, \ldots, 15\}$ i $y \in \{0, 1, 2, 3\}$, wyciągane z sześciu bitów wejścia.



Rysunek 12: Ilustracja interpretacji wejścia do S-Boxa jako koordynaty (x, y)

Założeniem S-boxa, jest zapewnienie tego, że zmiana jednego bitu wejścia zapewnia **co najmniej** zmianę dwóch bitów wyjścia. To oznacza, że ilość bitów zmienionych przez S-box jest **co najmniej** równa 2 i potrzeba maksymalnie 6 rund aby zmiana jednego bitu wpłynęła na całe wyjście.

11.4.6 Bezpieczeństwo

DES nie jest bezpieczny na dwóch frontach: jego wielkość klucza i jego własności kryptoanalityzcne.

64 bitowy klucz to po prostu za mało, zwłascza, że tak naprawdę klucz ma 56 bitów. Klucz DES mieści się w jednym rejestrze w architekturach 64-bitowych i przeszukanie wszystkich kluczy wiąże się z jedną pętlą `for` na type `uint64_t`. *Notka od autora:* założyłem się z wykładowcą, czy taki atak jest praktyczny. Iteracja po 40 bitach klucza

przy pomocy instrukcji `avx512` zajmuje około 8.6s. To oznacza, że iteracja po 56 bitach powinna zająć około 6.5 dnia.

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

11.4.7 Whitening

Poziom bezpieczeństwa DES można poprawić poprzez dodanie dodatkowego klucza do każdego bloku danych. Ten dodatkowy klucz jest używany do $\oplus$ z każdym blokiem danych przed szyfrowaniem.

12 Funkcje pseudolosowe

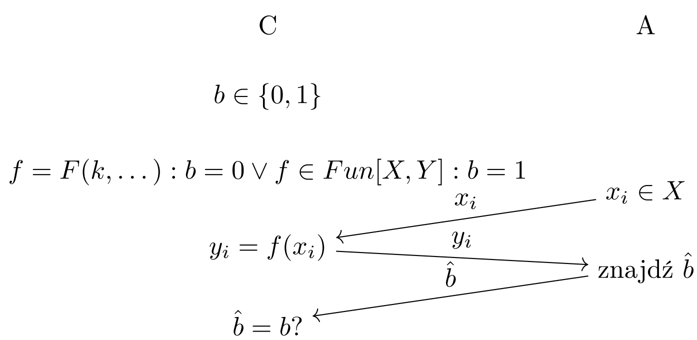
Funkcje pseudolosowe (PRF) to funkcje deterministyczne, które mają generować wartości podobne do losowych (pseudolosowe).

$$y = F(k, x)$$

$$f = F(k, \dots), f : X \to Y$$

12.1 Bezpieczeństwo semantyczne

Niech $Fun[X, Y]$ oznacza zbiór wszystkich funkcji $X \to Y$; $\#Fun[X, Y] = \#Y^{\#X}$.

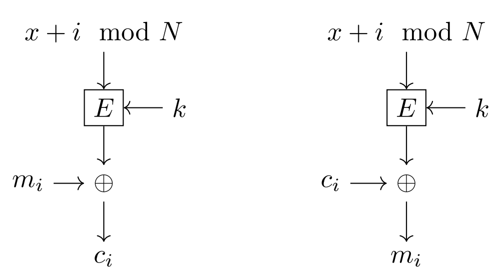


Rysunek 13: Formalizm gry dla bezpieczeństwa semantycznego funkcji pseudolosowych

Funkcja PRF jest bezpieczna semantycznie, jeśli $|P(b = \hat{b}) - P(b \neq \hat{b})| < \epsilon$. Mówiąc krótko, funkcja PRF jest semantycznie bezpieczna, jeśli nie da się rozróżnić jej od losowej funkcji.

12.2 Counter

Przykładem funkcji PRF jest counter, który dla danego szyfru blokowego $E(k, x)$ generuje kolejne szyfrogramy c_i. Publicznym jest x oraz m_i.



Rysunek 14: Counter

Jeśli F jest funkcją PRF, oraz N jest wystarczająco duża, to dla wielomianowej liczby bloków konstrukcja counter jest CPA bezpieczna.

13 MAC

Schemat MAC (Message Authentication Code) zakłada przechowywanie wiadomości m wraz z kodem t, nazywanym tagiem, jako (m, t). Ma to zapobiec atakom aktywnym, zapewniając weryfikację poprawności wiadomości.

$$t = S(k, m)$$

$$V(k, m, t) = \begin{cases} \texttt{accept} & \text{jeśli } t = S(k, m) \\ \texttt{reject} & \text{w przeciwnym wypadku} \end{cases}$$

$$V(k, m, S(k, m)) = \texttt{accept}$$

13.1 Przykład

$$\begin{array}{ccc} \text{Alice} & \xrightarrow{\quad (m,t) \quad} & \text{Bob} \\ & \boxed{\text{Mallet}} & \\ t = S(k,m) & & V(k,m,t) \end{array}$$

Rysunek 15: Ilustracja wykorzystania MAC

13.2 Bezpieczeństwo

$$\begin{tikzcd}
\text{C} & & \text{A} \\
k \in K & & m_1, m_2, \ldots m_{n-1} \arrow[dll, "m_1, m_2, \ldots m_{n-1}"'] \\
t_i = S(k, m_i) \arrow[rr, "t_1, t_2, \ldots t_{n-1}"] & & \text{znajdź } \hat{t}_n \arrow[dll, "\hat{t}_n, m_n"'] \\
\hat{t}_n = S(k, m_n) & &
\end{tikzcd}$$

Rysunek 16: Formalizm gry dla bezpieczeństwa MAC

Schemat MAC jest bezpieczny jeśli dla wszystkich efektywnych atakujących, prawdopodobieństwo $P(\hat{t} = S(k, m_n))$ jest pomijalnie małe.

'Wyszło jak wyszło …'