

# Wprowadzenie do zagadnień bezpieczeństwa i kryptografii

Patryk Czarnik

Wydział Matematyki, Informatyki i Mechaniki  
Uniwersytet Warszawski

Bezpieczeństwo sieci komputerowych – MSUI 2009/10

# Zagadnienia bezpieczeństwa

- **Identyfikacja i uwierzytelnienie**
- Kontrola dostępu
- Poufność:
  - zabezpieczenie przed ujawnianiem treści
  - zabezpieczenie przed analizą komunikacji
- Integralność danych
- Niezaprzeczalność
- Ochrona praw autorskich
- Poprawność oprogramowania
- Dyspozycyjność usług

# Zagadnienia bezpieczeństwa

- Identyfikacja i uwierzytelnienie
- Kontrola dostępu
- Poufność:
  - zabezpieczenie przed ujawnianiem treści
  - zabezpieczenie przed analizą komunikacji
- Integralność danych
- Niezaprzeczalność
- Ochrona praw autorskich
- Poprawność oprogramowania
- Dyspozycyjność usług

# Zagadnienia bezpieczeństwa

- Identyfikacja i uwierzytelnienie
- Kontrola dostępu
- Poufność:
  - zabezpieczenie przed ujawnianiem treści
  - zabezpieczenie przed analizą komunikacji
- Integralność danych
- Niezaprzeczalność
- Ochrona praw autorskich
- Poprawność oprogramowania
- Dyspozycyjność usług

# Zagadnienia bezpieczeństwa

- Identyfikacja i uwierzytelnienie
- Kontrola dostępu
- Poufność:
  - zabezpieczenie przed ujawnianiem treści
  - zabezpieczenie przed analizą komunikacji
- Integralność danych
- Niezaprzeczalność
- Ochrona praw autorskich
- Poprawność oprogramowania
- Dyspozycyjność usług

# Zagadnienia bezpieczeństwa

- Identyfikacja i uwierzytelnienie
- Kontrola dostępu
- Poufność:
  - zabezpieczenie przed ujawnianiem treści
  - zabezpieczenie przed analizą komunikacji
- Integralność danych
- Niezaprzeczalność
- Ochrona praw autorskich
- Poprawność oprogramowania
- Dyspozycyjność usług

# Zagadnienia bezpieczeństwa

- Identyfikacja i uwierzytelnienie
- Kontrola dostępu
- Poufność:
  - zabezpieczenie przed ujawnianiem treści
  - zabezpieczenie przed analizą komunikacji
- Integralność danych
- Niezaprzeczalność
- Ochrona praw autorskich
- Poprawność oprogramowania
- Dyspozycyjność usług

# Zagadnienia bezpieczeństwa

- Identyfikacja i uwierzytelnienie
- Kontrola dostępu
- Poufność:
  - zabezpieczenie przed ujawnianiem treści
  - zabezpieczenie przed analizą komunikacji
- Integralność danych
- Niezaprzeczalność
- Ochrona praw autorskich
- Poprawność oprogramowania
- Dyspozycyjność usług



# Zagadnienia bezpieczeństwa

- Identyfikacja i uwierzytelnienie
- Kontrola dostępu
- Poufność:
  - zabezpieczenie przed ujawnianiem treści
  - zabezpieczenie przed analizą komunikacji
- Integralność danych
- Niezaprzeczalność
- Ochrona praw autorskich
- Poprawność oprogramowania
- Dyspozycyjność usług

# Zagadnienia bezpieczeństwa

- Identyfikacja i uwierzytelnienie
- Kontrola dostępu
- Poufność:
  - zabezpieczenie przed ujawnianiem treści
  - zabezpieczenie przed analizą komunikacji
- Integralność danych
- Niezaprzeczalność
- Ochrona praw autorskich
- Poprawność oprogramowania
- Dyspozycyjność usług

# Zagadnienia bezpieczeństwa

- Identyfikacja i uwierzytelnienie
- Kontrola dostępu
- Poufność:
  - zabezpieczenie przed ujawnianiem treści
  - zabezpieczenie przed analizą komunikacji
- Integralność danych
- Niezaprzeczalność
- Ochrona praw autorskich
- Poprawność oprogramowania
- Dyspozycyjność usług

# Ochrona systemów komputerowych

- Fizyczna ochrona sprzętu przed niepożądanym dostępem
- Właściwa konfiguracja systemu
  - reguła: udostępniamy tylko to, co potrzebne
- Tworzenie oprogramowania zgodnie z regułami sztuki
  - np. nieużywanie `gets()` w C
- Bezpieczne protokoły sieciowe
- Kryptografia
- Zwykle najslabszy punkt – człowiek
  - wybór i ochrona haseł
  - stosowanie się do ustalonych zasad bezpieczeństwa

# Ochrona systemów komputerowych

- Fizyczna ochrona sprzętu przed niepożądanym dostępem
- Właściwa konfiguracja systemu
  - reguła: udostępniamy tylko to, co potrzebne
- Tworzenie oprogramowania zgodnie z regułami sztuki
  - np. nieużywanie `gets()` w C
- Bezpieczne protokoły sieciowe
- Kryptografia
- Zwykle najsłabszy punkt – człowiek
  - wybór i ochrona haseł
  - stosowanie się do ustalonych zasad bezpieczeństwa

# Ochrona systemów komputerowych

- Fizyczna ochrona sprzętu przed niepożądanym dostępem
- Właściwa konfiguracja systemu
  - reguła: udostępniamy tylko to, co potrzebne
- Tworzenie oprogramowania zgodnie z regułami sztuki
  - np. nieużywanie `gets()` w C
- Bezpieczne protokoły sieciowe
- Kryptografia
- Zwykle najsłabszy punkt – człowiek
  - wybór i ochrona haseł
  - stosowanie się do ustalonych zasad bezpieczeństwa

# Ochrona systemów komputerowych

- Fizyczna ochrona sprzętu przed niepożądanym dostępem
- Właściwa konfiguracja systemu
  - reguła: udostępniamy tylko to, co potrzebne
- Tworzenie oprogramowania zgodnie z regułami sztuki
  - np. nieużywanie `gets()` w C
- Bezpieczne protokoły sieciowe
- Kryptografia
- Zwykle najsłabszy punkt – człowiek
  - wybór i ochrona haseł
  - stosowanie się do ustalonych zasad bezpieczeństwa

# Ochrona systemów komputerowych

- Fizyczna ochrona sprzętu przed niepożądanym dostępem
- Właściwa konfiguracja systemu
  - reguła: udostępniamy tylko to, co potrzebne
- Tworzenie oprogramowania zgodnie z regułami sztuki
  - np. nieużywanie `gets()` w C
- Bezpieczne protokoły sieciowe
- Kryptografia
- Zwykle najsłabszy punkt – człowiek
  - wybór i ochrona haseł
  - stosowanie się do ustalonych zasad bezpieczeństwa



# Ochrona systemów komputerowych

- Fizyczna ochrona sprzętu przed niepożądanym dostępem
- Właściwa konfiguracja systemu
  - reguła: udostępniamy tylko to, co potrzebne
- Tworzenie oprogramowania zgodnie z regułami sztuki
  - np. nieużywanie `gets()` w C
- Bezpieczne protokoły sieciowe
- Kryptografia
- Zwykle najsłabszy punkt – człowiek
  - wybór i ochrona haseł
  - stosowanie się do ustalonych zasad bezpieczeństwa

# Właściwe podejście do ochrony

## Kryptografia i bezpieczne protokoły

- Dają wysoce skuteczne zabezpieczenia:
  - poufność
  - uwierzytelnienie i integralność danych
- Nie dają 100% gwarancji:
  - trudność złamania algorytmów nie dowiedziona
  - zdarzają się odkrycia metod łamania algorytmów kryptograficznych (przykłady: algorytm SHA-1, protokół WEP)
  - ponadto zawsze możliwy (acz kosztowny) atak *brute force*
- Należy stosować aktualne rozwiązania (algorytm, rozmiar klucza)

## Uwaga socjologiczna

Stosowanie bezpiecznych rozwiązań nie może być na tyle niewygodne, aby skłaniało użytkowników do ich obchodzenia.

# Właściwe podejście do ochrony

## Kryptografia i bezpieczne protokoły

- Dają wysoce skuteczne zabezpieczenia:
  - poufność
  - uwierzytelnienie i integralność danych
- Nie dają 100% gwarancji:
  - trudność złamania algorytmów nie dowiedziona
  - zdarzają się odkrycia metod łamania algorytmów kryptograficznych (przykłady: algorytm SHA-1, protokół WEP)
  - ponadto zawsze możliwy (acz kosztowny) atak *brute force*
- Należy stosować aktualne rozwiązania (algorytm, rozmiar klucza)

## Uwaga socjologiczna

Stosowanie bezpiecznych rozwiązań nie może być na tyle niewygodne, aby skłaniało użytkowników do ich obchodzenia.

# Właściwe podejście do ochrony

## Kryptografia i bezpieczne protokoły

- Dają wysoce skuteczne zabezpieczenia:
  - poufność
  - uwierzytelnienie i integralność danych
- Nie dają 100% gwarancji:
  - trudność złamania algorytmów nie dowiedziona
  - zdarzają się odkrycia metod łamania algorytmów kryptograficznych (przykłady: algorytm SHA-1, protokół WEP)
  - ponadto zawsze możliwy (acz kosztowny) atak *brute force*
- Należy stosować aktualne rozwiązania (algorytm, rozmiar klucza)

## Uwaga socjologiczna

Stosowanie bezpiecznych rozwiązań nie może być na tyle niewygodne, aby skłaniało użytkowników do ich obchodzenia.

# Właściwe podejście do ochrony

## Kryptografia i bezpieczne protokoły

- Dają wysoce skuteczne zabezpieczenia:
  - poufność
  - uwierzytelnienie i integralność danych
- Nie dają 100% gwarancji:
  - trudność złamania algorytmów nie dowiedziona
  - zdarzają się odkrycia metod łamania algorytmów kryptograficznych (przykłady: algorytm SHA-1, protokół WEP)
  - ponadto zawsze możliwy (acz kosztowny) atak *brute force*
- Należy stosować aktualne rozwiązania (algorytm, rozmiar klucza)

## Uwaga socjologiczna

Stosowanie bezpiecznych rozwiązań nie może być na tyle niewygodne, aby skłaniało użytkowników do ich obchodzenia.

# Kryptologia

Nauka o przekazywaniu informacji w sposób zabezpieczony przed niepowołanym dostępem

## Podział

kryptografia tworzenie systemów zabezpieczeń

kryptoanaliza łamanie systemów zabezpieczeń

## Niektóre założenia kryptografii

- **teoria złożoności** – pewne problemy są trudno rozwiązywalne
- **teoria prawdopodobieństwa** – istnieje losowość we wszechświecie
- **mechanika kwantowa** – nie możemy dokładnie określić stanu cząstek elementarnych

# Kryptologia

Nauka o przekazywaniu informacji w sposób zabezpieczony przed niepowołanym dostępem

## Podział

**kryptografia** tworzenie systemów zabezpieczeń

**kryptoanaliza** łamanie systemów zabezpieczeń

## Niektóre założenia kryptografii

- **teoria złożoności** – pewne problemy są trudno rozwiązywalne
- **teoria prawdopodobieństwa** – istnieje losowość we wszechświecie
- **mechanika kwantowa** – nie możemy dokładnie określić stanu cząstek elementarnych

# Kryptologia

Nauka o przekazywaniu informacji w sposób zabezpieczony przed niepowołanym dostępem

## Podział

**kryptografia** tworzenie systemów zabezpieczeń

**kryptoanaliza** łamanie systemów zabezpieczeń

## Niektóre założenia kryptografii

- **teoria złożoności** – pewne problemy są trudno rozwiązywalne
- **teoria prawdopodobieństwa** – istnieje losowość we wszechświecie
- **mechanika kwantowa** – nie możemy dokładnie określić stanu cząstek elementarnych



# Poufność

Do zapewnienia poufności informacji korzystamy z szyfrów.

## Podział szyfrów

- Szyfry symetryczne
  - blokowe
  - strumieniowe
- Szyfry asymetryczne (kryptografia klucza publicznego)

# Poufność

Do zapewnienia poufności informacji korzystamy z szyfrów.

## Podział szyfrów

- Szyfry symetryczne
  - blokowe
  - strumieniowe
- Szyfry asymetryczne (kryptografia klucza publicznego)

# Szyfr

**Szyfr symetryczny** to para funkcji:

- szyfrującej

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

- deszyfrującej

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

taka, że dla każdego klucza  $k$  i każdej wiadomości  $m$  musi zachodzić

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

# Oczekiwane własności szyfrów

Aby szyfr zapewniał bezpieczeństwo, musimy na funkcję szyfrującą nałożyć dodatkowe ograniczenia. Można je sformalizować na kilka sposobów.

Najmocniejsza własność to **bezpieczeństwo bezwarunkowe**.

Znajomość szyfrogramu nie ujawnia żadnej informacji o odpowiadającym mu tekście jawnym.

W języku bardziej formalnym:

Entropia tekstu jawnego jest taka sama przed jak i po poznaniu szyfrogramu.

## Oczekiwane własności szyfrów

Aby szyfr zapewniał bezpieczeństwo, musimy na funkcję szyfrującą nałożyć dodatkowe ograniczenia. Można je sformalizować na kilka sposobów.

Najmocniejsza własność to **bezpieczeństwo bezwarunkowe**.

Znajomość szyfrogramu nie ujawnia żadnej informacji o odpowiadającym mu tekście jawnym.

W języku bardziej formalnym:

Entropia tekstu jawnego jest taka sama przed jak i po poznaniu szyfrogramu.

## Oczekiwane własności szyfrów

Aby szyfr zapewniał bezpieczeństwo, musimy na funkcję szyfrującą nałożyć dodatkowe ograniczenia. Można je sformalizować na kilka sposobów.

Najmocniejsza własność to **bezpieczeństwo bezwarunkowe**.

Znajomość szyfrogramu nie ujawnia żadnej informacji o odpowiadającym mu tekście jawnym.

W języku bardziej formalnym:

Entropia tekstu jawnego jest taka sama przed jak i po poznaniu szyfrogramu.

# Własności semantycznego bezpieczeństwa

Słabszymi własnościami są **własności semantycznego bezpieczeństwa**, zwykle definiowane jako **nierozróżnialność ze względu na** atak określonego rodzaju.

## Atak kryptologiczny

- działanie mające na celu złamanie zabezpieczeń kryptograficznych,
- w skrajnych przypadkach umożliwia poznanie tajnego klucza, odczytanie zaszyfrowanej informacji lub podszycie się
- ale niepożądane jest jakiegokolwiek osłabienie bezpieczeństwa bezwarunkowego

# Własności semantycznego bezpieczeństwa

Słabszymi własnościami są **własności semantycznego bezpieczeństwa**, zwykle definiowane jako **nierozróżnialność ze względu na** atak określonego rodzaju.

## Atak kryptologiczny

- działanie mające na celu złamanie zabezpieczeń kryptograficznych,
- w skrajnych przypadkach umożliwia poznanie tajnego klucza, odczytanie zaszyfrowanej informacji lub podszyć się
- ale niepożądane jest jakiegokolwiek osłabienie bezpieczeństwa bezwarunkowego



# Często rozważane rodzaje ataków

- atak ze znanym szyfrogramem (oczywisty)
  - atak siłowy (*brute force*),
  - analiza statystyczna,
- atak ze znanym tekstem jawnym (na ogół łatwy do przeprowadzenia),
- atak z wybranym tekstem jawnym (łatwy w przypadku klucza publicznego)
- atak z wybranym szyfrogramem (wymaga dostępu do maszyny deszyfrującej)
- atak na hasło (słownikowy, „socjalny”),
- atak „człowiek w środku” (dotyczy protokołów komunikacyjnych, a nie samych algorytmów kryptograficznych)

## Atak z wybranym tekstem jawnym

- Atakujący (Bartek) ma dostęp do maszyny szyfrującej, tj. może wielokrotnie wybrać tekst jawny i poznać jego szyfrogram.
- Korzystając z tego stara się uzyskać informacje, które pozwolą mu obniżyć bezpieczeństwo mechanizmu (np. tajny klucz).
- Własność krytyczna dla mechanizmów opartych o klucz publiczny.

## Nierozróżnialność zwn. atak z wyb. tekstem jawnym

Po wykonaniu powyższego ataku tak można sprawdzić własność nierozróżnialności:

- Alicja posiada maszynę szyfrującą, Bartek nie posiada (już) do niej dostępu.
- Bartek wysyła do Alicji dwie różne wiadomości  $m_1$  i  $m_2$ .
- Alicja rzuca monetą. W zależności od wyniku szyfruje wiadomość  $m_1$  lub  $m_2$  i wysyła szyfrogram do Bartka.
- Bartek próbuje odgadnąć szyfrogram której wiadomości otrzymał.

Jeśli Bartek nie jest w stanie odpowiedzieć na to pytanie z prawdopodobieństwem znacząco większym od  $\frac{1}{2}$ , własność **nierozróżnialności** zachodzi.

# Nierozróżnialność zwn. atak z wyb. tekstem jawnym

Po wykonaniu powyższego ataku tak można sprawdzić własność nierozróżnialności:

- Alicja posiada maszynę szyfrującą, Bartek nie posiada (już) do niej dostępu.
- Bartek wysyła do Alicji dwie różne wiadomości  $m_1$  i  $m_2$ .
- Alicja rzuca monetą. W zależności od wyniku szyfruje wiadomość  $m_1$  lub  $m_2$  i wysyła szyfrogram do Bartka.
- Bartek próbuje odgadnąć szyfrogram której wiadomości otrzymał.

Jeśli Bartek nie jest w stanie odpowiedzieć na to pytanie z prawdopodobieństwem znacząco większym od  $\frac{1}{2}$ , własność **nierozróżnialności** zachodzi.

# Atak z wybranym szyfrogramem

- Atakujący (Bartek) ma dostęp do maszyny deszyfrującej, tj. może wielokrotnie wybrać szyfrogram i poznać odpowiadający mu tekst jawny.
- Korzystając z tego stara się uzyskać informacje, które pozwolą mu obniżyć bezpieczeństwo mechanizmu (np. tajny klucz).
- Atakiem tego rodzaju jest także taki, gdy atakujący może odszyfrować tylko niektóre szyfrogramy.

# Nierozróżnialność zwn. atak z wyb. szyfrogramem

Po wykonaniu ataku tak można sprawdzić własność nierozróżnialności:

- Alicja posiada maszynę deszyfrującą, Bartek nie posiada (już) do niej dostępu.
- Bartek wysyła do Alicji dwa różne (poprawne) szyfrogramy  $s_1$  i  $s_2$ .
- Alicja rzuca monetą. W zależności od wyniku deszyfruje szyfrogram  $s_1$  lub  $s_2$  i wysyła odszyfrowaną wiadomość do Bartka.
- Bartek próbuje odgadnąć wiadomość odpowiadającą któremu szyfrogramowi otrzymał.

Jeśli Bartek nie potrafi odpowiedzieć na to pytanie z prawdopodobieństwem znacząco większym od  $\frac{1}{2}$ , własność **nierozróżnialności** zachodzi.

# Nierozróżnialność zwn. atak z wyb. szyfrogramem

Po wykonaniu ataku tak można sprawdzić własność nierozróżnialności:

- Alicja posiada maszynę deszyfrującą, Bartek nie posiada (już) do niej dostępu.
- Bartek wysyła do Alicji dwa różne (poprawne) szyfrogramy  $s_1$  i  $s_2$ .
- Alicja rzuca monetą. W zależności od wyniku deszyfruje szyfrogram  $s_1$  lub  $s_2$  i wysyła odszyfrowaną wiadomość do Bartka.
- Bartek próbuje odgadnąć wiadomość odpowiadającą któremu szyfrogramowi otrzymał.

Jeśli Bartek nie potrafi odpowiedzieć na to pytanie z prawdopodobieństwem znacząco większym od  $\frac{1}{2}$ , własność **nierozróżnialności** zachodzi.

# Klasyfikacja ataków ze względu na adaptowność

Ataki powyższych rodzajów dodatkowo dzieli się ze względu na adaptowność (wynikającą często z czasu dostępu do maszyny).

- **Atak nieadaptowny** (*lunchtime attack*) – atakujący ma uprzednio przygotowany zestaw wiadomości / szyfrogramów.
- **Atak adaptowny** (*midnight attack*) – atakujący może na bieżąco preparować nowe wiadomości / szyfrogramy.



# Szyfry symetryczne

- Ten sam klucz używany do szyfrowania jak i deszyfrowania (ewentualnie istnieje łatwy sposób na otrzymanie klucza deszyfrującego z klucza szyfrującego).
- Stosowane od bardzo dawna. Jako przykład jednego z pierwszych szyfrów podaje się szyfr Cezara.
- Działają na zasadzie jak największego zagmatwania.
- Są efektywne:
  - szyfrowanie i deszyfrowanie przebiega bardzo szybko,
  - używane klucze są małe – aktualnie 128–256 bitów.

# Szyfry blokowe i strumieniowe

Szyfry symetryczne można podzielić na dwie kategorie:

- **szyfry blokowe** – szyfrowane są bloki danych,
- **szyfry strumieniowe** – szyfrowanie następuje bit po bicie.

# Zasady konstrukcji szyfrów blokowych

## Schemat Feistel

Najbardziej powszechny schemat budowy szyfru w oparciu o funkcję jednej „rundy”. Idea:

- Na bloku do zaszyfrowania wykonywanych jest sukcesywnie wiele rund tego samego przekształcenia, ale z różnymi kluczami.
- Runda składa się ze złożenia prostych operacji, które same w sobie nie są wystarczająco bezpieczne (np. przekształcenia liniowe, translacje, podstawienia).
- Klucze dla poszczególnych rund są generowane przez ekspansję klucza głównego.

## Tryby szyfrowania

- Dodatkowo szyfrowania kolejnych bloków dokonuje się w **trybach** ECB, CBC, CFB, OFB itp.

# Zasady konstrukcji szyfrów blokowych

## Schemat Feistel

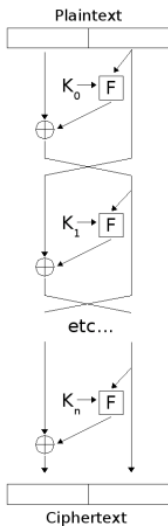
Najbardziej powszechny schemat budowy szyfru w oparciu o funkcję jednej „rundy”. Idea:

- Na bloku do zaszyfrowania wykonywanych jest sukcesywnie wiele rund tego samego przekształcenia, ale z różnymi kluczami.
- Runda składa się ze złożenia prostych operacji, które same w sobie nie są wystarczająco bezpieczne (np. przekształcenia liniowe, translacje, podstawienia).
- Klucze dla poszczególnych rund są generowane przez ekspansję klucza głównego.

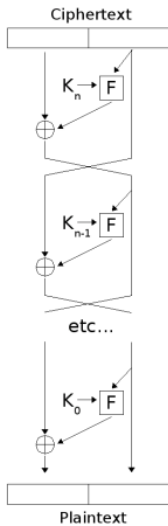
## Tryby szyfrowania

- Dodatkowo szyfrowania kolejnych bloków dokonuje się w **trybach** ECB, CBC, CFB, OFB itp.

Encryption:



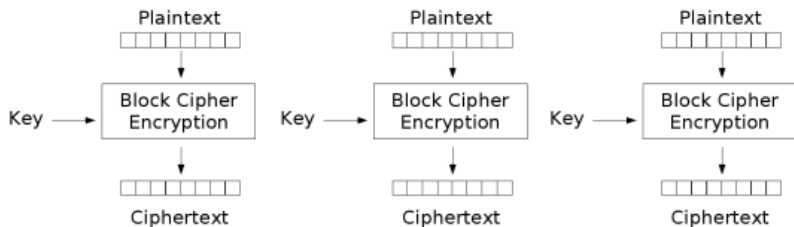
Decryption:



Feistel Cipher

## Tryb ECB (*Electronic Code Book*)

Jeden blok tekstu jawnego jest przekształcany za pomocą przekształcenia szyfrującego na jeden blok tekstu zaszyfrowanego.

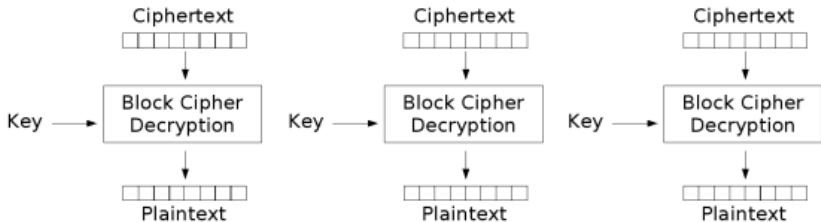


Electronic Codebook (ECB) mode encryption

Źródło obrazków: Wikipedia

## Tryb ECB (*Electronic Code Book*)

Jeden blok tekstu jawnego jest przekształcany za pomocą przekształcenia szyfrującego na jeden blok tekstu zaszyfrowanego.



Electronic Codebook (ECB) mode decryption

Źródło obrazków: Wikipedia

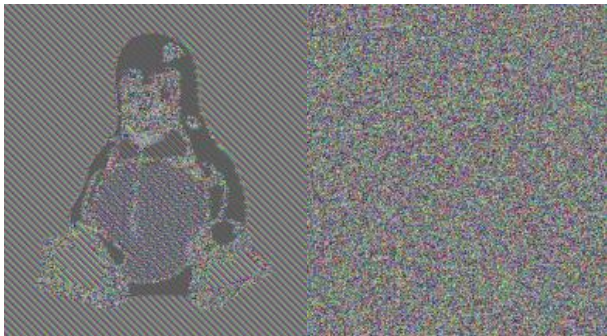
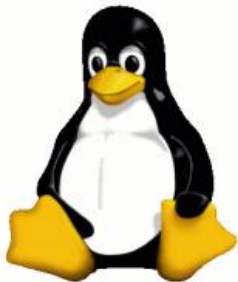
## Tryb ECB – wady

- jednakowe bloki tekstu jawnego dają jednakowy szyfrogram,
- możliwa analiza statystyczna szyfrogramu.



# Tryb ECB – wady

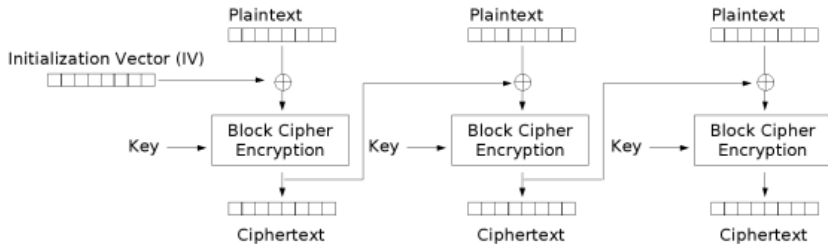
- jednakowe bloki tekstu jawnego dają jednakowy szyfrogram,
- możliwa analiza statystyczna szyfrogramu.



Źródło: Wikipedia

## Tryb CBC (*Cipher Block Chaining*)

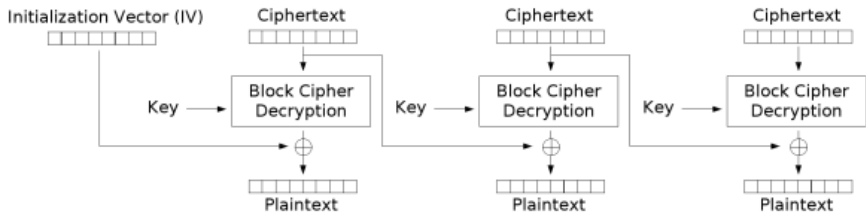
Na każdym kolejnym bloku tekstu jawnego jest wykonywana operacja XOR z poprzednio uzyskanym blokiem zaszyfrowanym i tak uzyskany wynik jest poddawany szyfrowaniu. Wymaga wektora inicjującego stan.



Cipher Block Chaining (CBC) mode encryption

## Tryb CBC (*Cipher Block Chaining*)

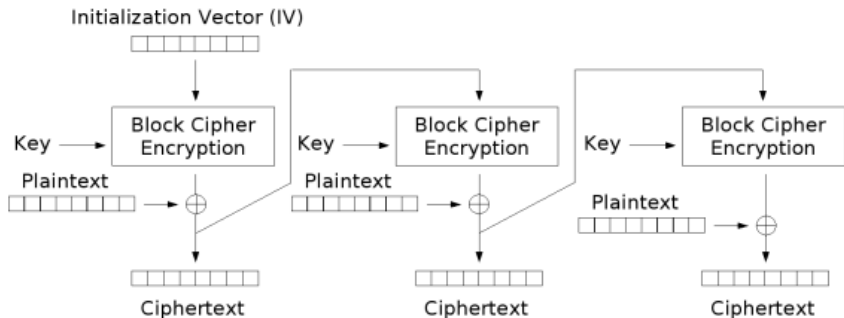
Na każdym kolejnym bloku tekstu jawnego jest wykonywana operacja XOR z poprzednio uzyskanym blokiem zaszyfrowanym i tak uzyskany wynik jest poddawany szyfrowaniu. Wymaga wektora inicjującego stan.



Cipher Block Chaining (CBC) mode decryption

## Tryb CFB (*Cipher Feedback*)

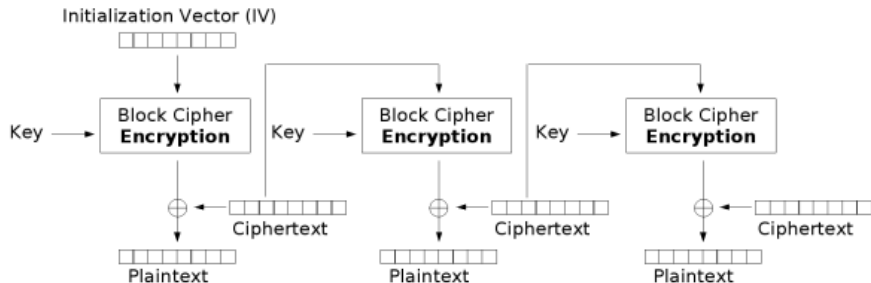
Szyfrowany jest szyfrogram z poprzedniego bloku, jawny tekst jest XOR-owany z wynikiem tego szyfrowania.



Cipher Feedback (CFB) mode encryption

## Tryb CFB (*Cipher Feedback*)

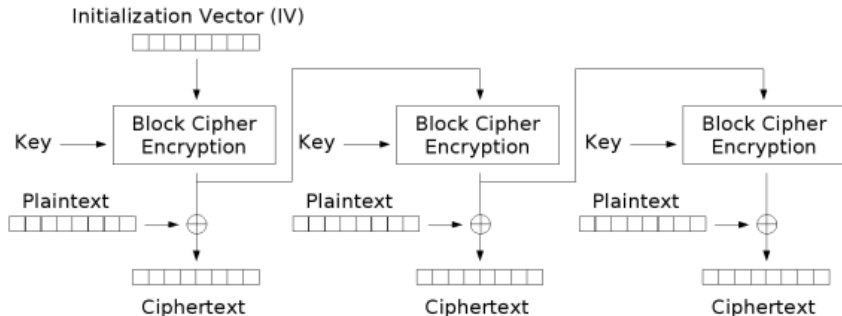
Szyfrowany jest szyfrogram z poprzedniego bloku, jawny tekst jest XOR-owany z wynikiem tego szyfrowania.



Cipher Feedback (CFB) mode decryption

# OFB (*Output Feedback*)

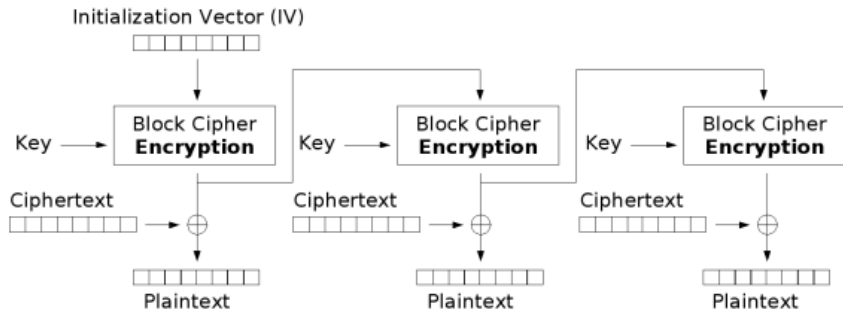
Podobne do CFB, ale dane wejściowe nie są brane pod uwagę przy generowaniu następnego bloku do szyfrowania.



Output Feedback (OFB) mode encryption

## OFB (*Output Feedback*)

Podobne do CFB, ale dane wejściowe nie są brane pod uwagę przy generowaniu następnego bloku do szyfrowania.



Output Feedback (OFB) mode decryption

# Klasyfikacja szyfrów strumieniowych

- **Z jednorazowym strumieniem kluczy** – *one time pad*,
- **Synchroniczne** – strumień kluczy jest generowany niezależnie od tekstu jawnego i szyfrogramu.
- **Samosynchronizujące** – strumień kluczy jest generowany jako funkcja klucza głównego oraz pewnego fragmentu ostatnio zaszyfrowanych/odszyfrowanych danych.



# Własności strumieniowych szyfrów synchronicznych

- Nadawca i odbiorca muszą być zsynchronizowani.
- Brak propagacji błędów – zmodyfikowanie fragmentu szyfrogramu nie ma wpływu na wynik deszyfrowania pozostałych danych.
- Możliwe są ataki aktywne (wstawienie, usuwanie lub powtarzanie danych) korzystające z powyższych własności.

# Własności strumieniowych szyfrów samosynchronizujących

- Samosynchronizacja.
- Ograniczona propagacja błędów.
- Trudniejsze ataki aktywne – modyfikacja jednego bitu szyfrogramu ma wpływ na pewną liczbę bitów tekstu po odszyfrowaniu.
- Rozpraszanie tekstu jawnego – każdy bit tekstu jawnego ma wpływ na to, jak zostaną zaszyfrowane bity następujące po nim.

# Najbardziej znane szyfry symetryczne blokowe

- DES (*Data Encryption Standard*) – Feistel  
(M=64, K=56, C=64)
- FEAL (*Fast Data Encryption Algorithm*) – Feistel  
(M=64, K=64, C=64)
- IDEA (*International Data Encryption Standard*) – Feistel  
(M=64, K=128, C=64)
- SAFER (*Secure And Fast Encryption Routine*)  
(M=64, K=64, C=64)
- RC5  
(zmienne)
- AES/Rijndael (*Advanced Encryption Standard*)  
(M=128, K=128–192–256, C=128)

# Najbardziej znane szyfry symetryczne strumieniowe

- Szyfry bazujące na LFSR (*Linear Feedback Shift Registers*).
- SEAL (1993) – addytywny szyfr binarny bazujący na funkcji pseudolosowej zaprojektowany do efektywnej implementacji softwarowej.