# Eksploatacja i bezpieczeństwo systemów

**dr inż. Mirosław Mazurek**

Zakład Systemów Złożonych
**Bud. F, pok. 305, tel. 17 865 11 04**

# Klasyfikacja metod uwierzytelniania

**Użytkownicy mogą być uwierzytelniani na podstawie jednej lub kilku informacji pochodzących z następujących zbiorów:**

**Tego, co użytkownik wie - tajny tekst np. hasło znane tylko użytkownikowi i systemowi. W procesie rejestracji jest ono wprowadzane przez użytkownika i sprawdzane przez system.**

**Tego, co użytkownik posiada - klucz, plakietka, karta pomagające w weryfikacji użytkownika. W metodzie hasło-odzew (challenge-response) użytkownik dysponuje kartą wyświetlającą identyfikator liczbowy. Można stosować również metodę haseł jednorazowych.**

**Tego, kim użytkownik jest - cechy fizyczne (odciski palców, odciski dłoni, wzorzec siatkówki) lub behawioralne (wzorzec głosu, podpis), które można zapamiętać i porównać. Weryfikacja polega na ponownym zbadaniu użytkownika i porównaniu wyników badań z zapamiętanymi w systemie. Ta metoda umożliwia również ewentualną identyfikację włamywacza. W przypadku metod behawioralnych istnieje możliwość odrzucenia prawidłowego użytkownika (błąd ujemny) i pozytywnej identyfikacji niewłaściwej osoby (błąd dodatni). W chwili obecnej techniki tego typu są zwykle stosowane jako dodatkowe (obok hasła). Dwustopniowa weryfikacja zwiększa bezpieczeństwo.**

# Słowniki haseł

**Hasła mogą zostać ukradzione z bazy haseł lub przechwycone podczas przesyłania poprzez sieć. Do odgadnięcia hasła może być wykorzystywana metoda słownikowa lub łamania brutalnego. Metoda łamania brutalnego polega na pełnym przeglądzie, czyli wypróbowywaniu każdej kombinacji kodowej, która mogłaby być hasłem. Metoda słownikowa polega na sprawdzeniu każdego słowa występującego w pliku nazywanym słownikiem.**

**W latach 90-tych Klein prowadził analizę słabości haseł. Zgromadził duży zbiór haseł. Następnie budował słownik wg poniższego schematu:**
**wykaz nazw użytkowników, ich inicjałów, nazw kont i innej informacji związanej z użytkownikiem,**
**wykaz słów z różnych słowników: imiona i ich permutacje,**
**nazwy miejsc, tytuły filmów i książek i postaci w nich występujących,**
**różne przekształcenia słów z kroku poprzedniego,**
**dowolne zamiany liter małych na duże i odwrotnie,**
**słowa w obcych językach dla użytkowników będących obcokrajowcami.**

**Wyniki eksperymentu pokazały że łatwo jest odgadnąć hasło dysponując informacją o użytkowniku. Wynika z tego, że hasła powinny być trudne do odgadnięcia i ukradzenia. Rozwój komputerów umożliwia coraz szybsze łamanie haseł. Istnieje wiele narzędzi, które umożliwiają łamanie haseł. Jednym z nich jest program LC4, którego okno zamieszczono na rys. 1.**

# Ochrona haseł

**Techniki ochrony haseł można podzielić następująco:**

**Nadzorowanie haseł (wybór, pielęgnacja)**

**Komunikaty systemowe: wyświetlany jest komunikat przed i po rejestracji użytkownika określający dane systemu. Komunikaty takie mogą dostarczyć intruzowi pewnych wskazówek i dlatego powinny zostać wyłączone lub ograniczone.**

**Wprowadzanie hasła: hasła nie powinny być widoczne w momencie wprowadzania.**

**Ograniczanie ilości prób rejestracji: po osiągnięciu limitu nieudanych logowań konto użytkownika powinno zostać zablokowane (uniemożliwienie dalszych prób). Odblokowanie możliwe po weryfikacji przez administratora przy pomocy innej metody niż hasło. Uniemożliwia to stosowanie metody brutalnego łamania haseł. Informacja o nieudanych logowaniach powinna być zachowywana.**

**Starzenie się haseł: hasło powinno mieć określony czas życia, po którym musi zostać zmienione. Możliwa może być również zmiana hasła przed upływem ważności (w pewnych granicach). Wykorzystane hasła powinny być pamiętane (w określonym zakresie). Administrator, w przypadku zagrożenia, powinien mieć możliwość natychmiastowej zmiany hasła.**

# Ochrona haseł

**Systemy z dwoma hasłami: Drugie hasło jest zwykle wykorzystywane podczas próby dostępu do szczególnie chronionych zasobów.**

**Minimalna długość hasła: Krótkie hasła są łatwiejsze do odgadnięcia. Wymaga się aby miały co najmniej 6 lub 8 znaków i występowały w nich określone kombinacje grup znaków.**

**Blokowanie konta użytkownika: blokować należy konta nie używane. Usuwanie blokady po weryfikacji przez administratora.**

**Ochrona hasła administratora: Ze względu na znacznie większe uprawnienia, w porównaniu do innych użytkowników, jest częściej atakowane i powinno być lepiej chronione. Można wymagać aby było ciągiem znaków heksadecymalnych. Nie powinno być przesyłane przez sieć i powinno być często zmieniane.**

**Generowanie hasła przez system: niektóre systemu oferują użytkownikowi kilka haseł do wyboru. Są to zwykle hasła trudne do zapamiętania, co powoduje że użytkownicy je zapisują. Hasła generowane powinny być łatwe do wymówienia.**

# Ochrona haseł

**Zabezpieczanie przed odgadnięciem poprzez odrzucanie zbyt łatwych haseł - sprawdzanie haseł**

**Sprawdzanie bierne: Realizowane jest po wprowadzeniu haseł do użytku za pomocą programu uruchamianego w ustalonych odstępach czasu. Program taki porównuje istniejące hasła z listą haseł łatwych do odgadnięcia. Hasła łatwe są unieważniane a informacja o tym powinna zostać przesłana do użytkownika. Metoda bierna wymaga zużycia znacznych zasobów. Ponadto łatwe hasła funkcjonują w systemie do momentu ich wykrycia stwarzając potencjalne niebezpieczeństwo.**

**Sprawdzanie aktywne. Podczas zmiany hasła przez użytkownika, podane przez niego nowe hasło jest weryfikowane zgodnie z wbudowanym algorytmem. Hasło zbyt łatwe jest odrzucane a użytkownik jest proszony o podanie nowego. W tego typu algorytmie istotne jest zapewnienie równowagi pomiędzy użytecznością i bezpieczeństwem. Zbyt restrykcyjny algorytm będzie powodował niezadowolenie użytkowników. Zbyt liberalny algorytm obniży bezpieczeństwo systemu.**

# Hasła jednorazowe

**Metoda haseł jednorazowych (one-time passwords) polega na jednorazowym wykorzystaniu wygenerowanego hasła. Najczęściej są to liczby wygenerowane na stacji klienckiej i weryfikowane na serwerze.**

**System jednorazowego hasła S/Key zdefiniowany przez RFC 1760 oparty jest na funkcji MD4 i MD5. Protokół ten został zaprojektowany do przeciwdziałania atakowi metodą powtórzeń.**

**Procedura przedstawia się następująco:**
**Klient i serwer są wstępnie skonfigurowani tym samym hasłem oraz licznikiem iteracji. Licznik iteracji określa wymaganą ilość powtórzeń funkcji mieszającej. Przy każdym logowaniu licznik iteracji stronie klienta maleje.**

# Hasła jednorazowe

**Klient inicjuje wymianę wysyłając pakiet inicjujący.**

**Serwer odpowiada numerem sekwencji. Wysyła również tzw. ziarno.**

**Po stronie klienta wyliczane jest hasło jednorazowe:**

**operator wprowadza tajne hasło, które jest łączone z ziarnem,**
**kilkakrotnie wykonywana jest funkcja mieszająca generująca dane wyjściowe (wg licznika powtórzeń),**
**dane wyjściowe przekształcane są do postaci czytelnej i prezentowane operatorowi.**

**Klient przesyła jednorazowe hasło do serwera.**

**W serwerze znajduje się plik zawierający dla każdego użytkownika jednorazowe hasło z poprzedniego pomyślnego logowania.**

**Serwer jednokrotnie przepuszcza odebrane hasło przez funkcję mieszającą. Wynik powinien odpowiadać hasłu z poprzedniego logowania.**

# Hasła jednorazowe

**Po pewnym czasie klient musi znowu zainicjować system za pomocą specjalnego polecenia.**

**Inne rozwiązania to uwierzytelnianie hasła za pomocą znacznika. Wymagają użycia tzw. inteligentnej karty (smart card) lub karty znacznika (token card). Chroniony obiekt musi być wyposażony w oprogramowanie agenta. Chroniony obiekt musi być wyposażony w oprogramowanie agenta. Ten mechanizm uwierzytelniania oparty może być na systemie wyzwanie-odpowiedź (challenge-response) lub uwierzytelnienie zsynchronizowane z czasem (time-synchronous authentication).**

# System Kerberos

**W procesie uwierzytelniania może być wykorzystywana zaufana strona trzecia (trusted third-party), która poświadcza tożsamość klienta i serwera. Jest nazywana serwerem bezpieczeństwa (security serwer). Jego zadaniem jest przechowywanie haseł wykorzystywanych podczas weryfikacji użytkowników i serwerów. Jest to jedyne miejsce przechowywania haseł.**

**Wymagania:**

**Zapewnienie dwustronnego uwierzytelnienia dwukierunkowego.**
**Zadaniem trzeciej strony jest przechowywanie i pielęgnacja haseł.**
**Hasła nie powinny być przesyłane poprzez sieć.**
**Hasła nie powinny być przechowywane na stacji klienckiej.**
**Zarejestrowany użytkownik powinien otrzymać tymczasowy klucz tajny. Jest on wykorzystywany przez klienta przy wszystkich dostępach.**
**System powinien pozwalać na bezpieczne przesyłanie kluczy szyfrowania pomiędzy klientami i serwerami.**

# System Kerberos

**System Kerberos powstał w czasie realizacji projektu Athena na uniwersytecie MIT. Projekt miał na celu integrację komputerów uniwersyteckich. System weryfikacji autentyczności jest oparty na znajomości haseł zapisanych w serwerze Kerberosa. W procesie uwierzytelniania wykorzystuje się tajny dzielony klucz (shared secret), który pozwala na identyfikację użytkowników bez eksponowania informacji narażających bezpieczeństwo sieci. W systemie uwierzytelniania wyróżnić można cztery komponenty.**

**Pierwszy komponent to klient czyli użytkownik lub aplikacja reprezentująca użytkownika. Jest to miejsce, z którego użytkownik prowadzi pracę, wprowadza identyfikator i hasło.**

**Drugi komponent to serwer uwierzytelniający (authentication server) służący do przechowywania haseł i sprawdzania tożsamości użytkownika. W czasie wymiany informacji z klientem dostarcza on klientowi bilet uprawniający do korzystania z usługi przyznawania biletów (ticket-granting ticket).**

**Komponent trzeci to serwer przepustek lub serwer przyznawania biletów (ticket-granting server), który dostarcza klientowi bilet uprawniający do skorzystania z serwera aplikacji.**

# System Kerberos

**Komponent czwarty to serwer aplikacji (application server) czyli zasób, który chce się upewnić, że dany klient jest poprawny. Dostarcza klientowi żądanej przez niego usługi.**

**Konto w bazie zawiera dane dotyczące tożsamości oraz klucze główne (np. hasła) wszystkich klientów i serwerów z danego obszaru. Klucz główny serwera uwierzytelniajacego służy do szyfrowania wszystkich kluczy głównych klientów udaremniając nieautoryzowany dostęp do serwera.**

**Poważną wadą systemu jest jego dostęp do zaszyfrowanych haseł użytkowników. Powoduje to, że zawarte są w nim dane krytyczne dla bezpieczeństwa i powinien być on chroniony w sposób szczególny.**

**Serwer Kerberosa jest bezstanowy. Odpowiada po prostu na żądania użytkowników i wydaje przepustki (żetony, bilety). Ułatwia to tworzenie replikowanych serwerów zapasowych. Funkcjonowanie systemu z punktu widzenia użytkownika niczym nie różni się od systemu tradycyjnego.**

# Wybrane algorytmy kryptograficzne:

**DES (ang. Data Encryption Standard)** - standard szyfrowania danych zaakceptowany w roku 1977 przez Narodowe Biuro Standardów (National Bureau of Standards) obecnie Narodowy Instytut Standardów i Technologii (NIST), stosowany do szyfrowania informacji “nie utajnionej, ale ważnej" (ang. Unclassified but Sensitive). Algorytm opracowany przez IBM jest rozwinięciem szyfru LUCIFER.

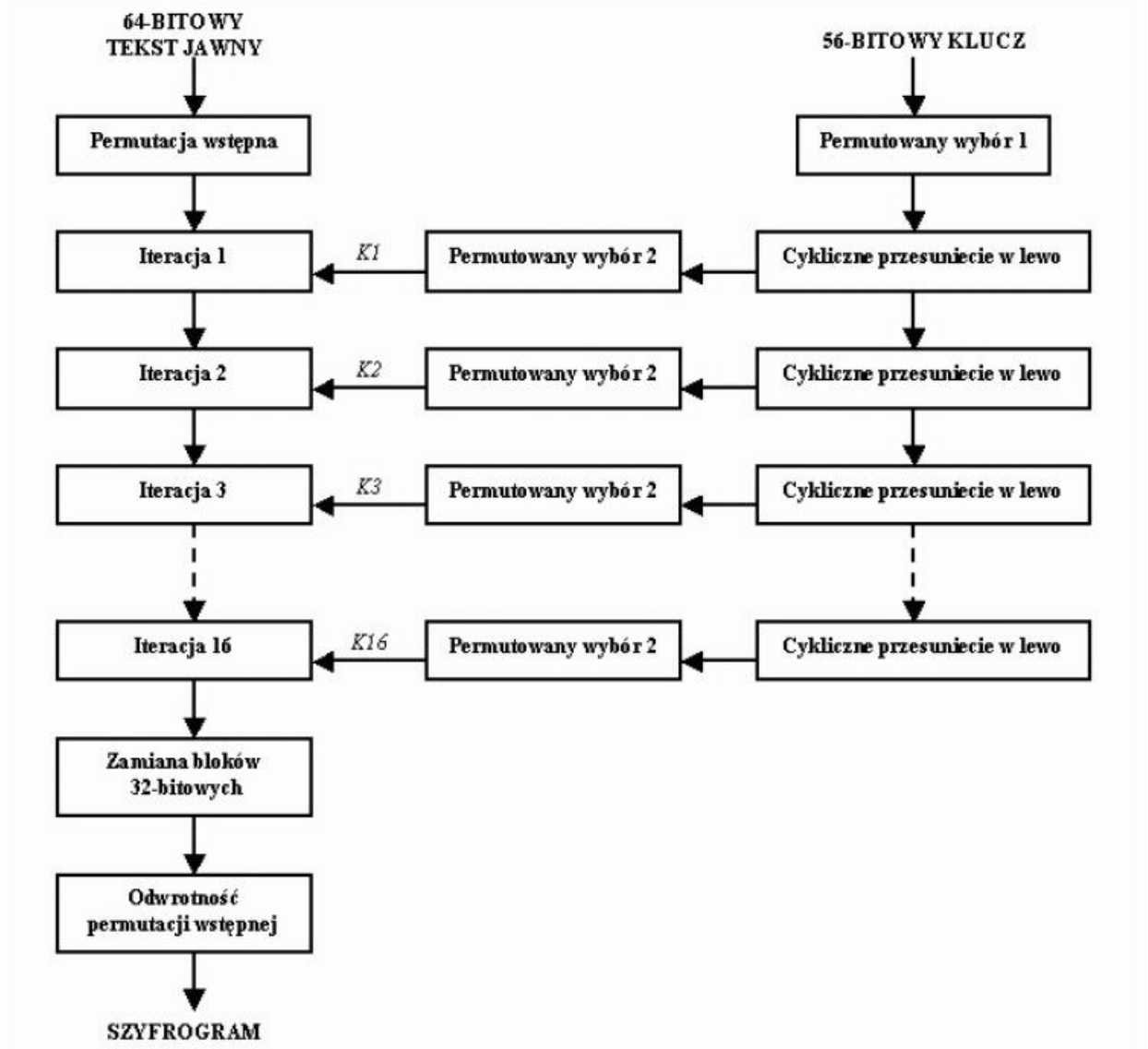
**RSA** (Rivest, Shamir i Adelman) - najpopularniejszy algorytm z kluczem publicznym. Stosowany zarówno do szyfrowania, jak i do podpisów cyfrowych.

**DSA** (ang. Digital Signaturre Algorithm) jest również algorytmem z kluczem publicznym, ale może być stosowany tylko do podpisów cyfrowych.

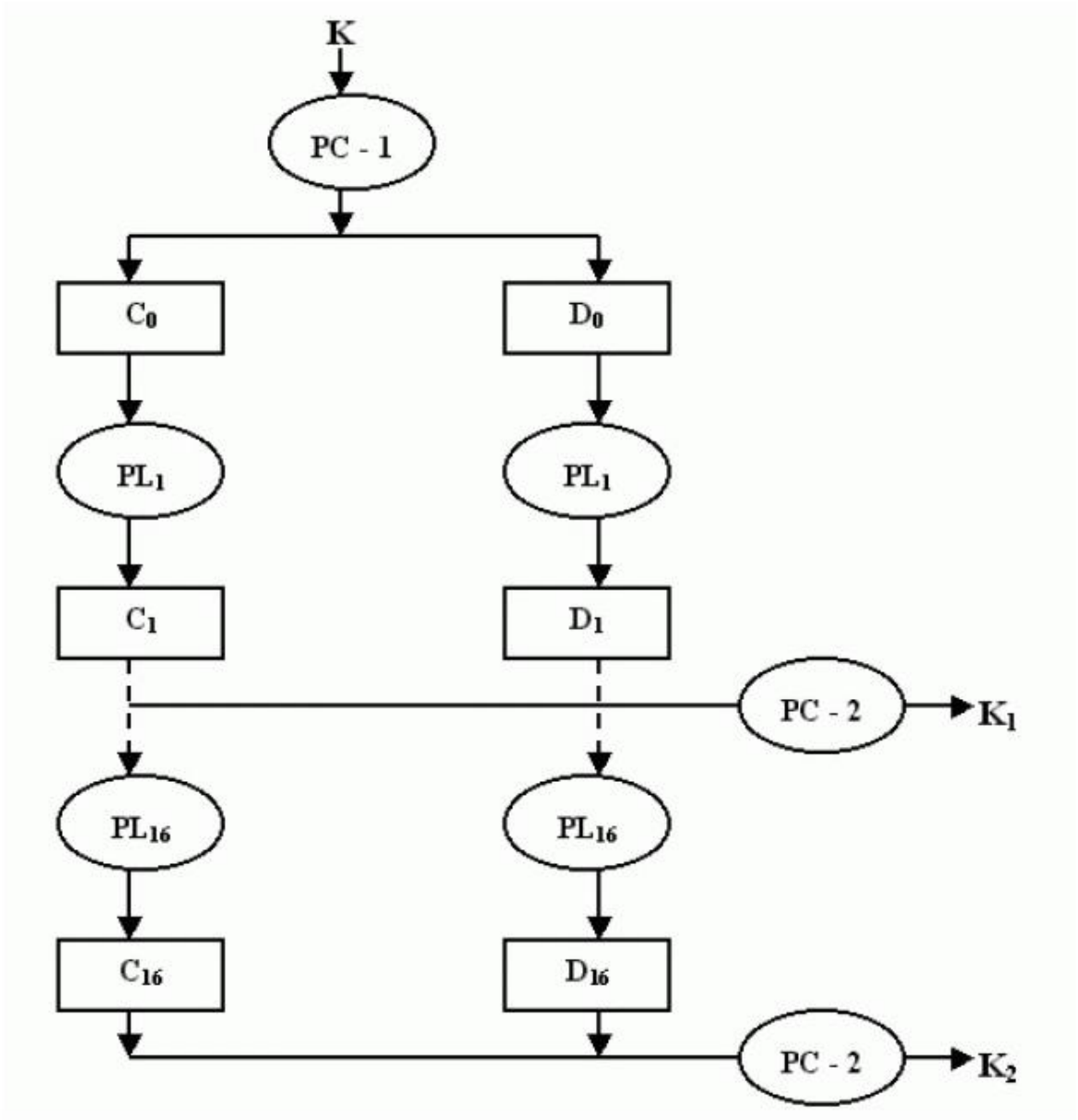
# Jak działa algorytm DES?

DES jest szyfrem blokowym z blokami o długości 64 bitów. Do szyfrowania i deszyfrowania danych wykorzystywanych jest 56 bitów klucza, który zapisany jest w postaci 64 bitowego ciągu, w którym co 8 bit jest bitem kontrolnym i może służyć do kontroli parzystości.

1. Na początku tekst jawny, który ma zostać zaszyfrowany, dzielony jest na bloki 64-bitowe.



# Procedura przygotowania kluczy



# Algorytm DES

2. Następnie dla każdego bloku wykonywane są następujące operacje:

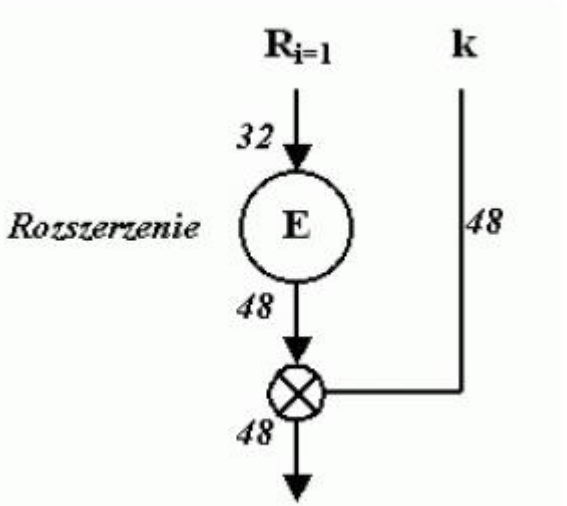
dokonywana jest permutacja początkowa bloku przestawiająca bity w pewien określony sposób – nie zwiększa ona bezpieczeństwa algorytmu, a jej początkowym celem było ułatwienie wprowadzania danych do maszyn szyfrujących używanych w czasach powstania szyfru

blok wejściowy rozdzielany jest na dwie 32-bitowe części: lewą oraz prawą wykonywanych jest 16 cykli tych samych operacji, zwanych funkcjami **Feistela**, podczas których dane łączone są z kluczem.

# Algorytm DES

Operacje te wyglądają następująco:

- bity klucza są przesuwane, a następnie wybieranych jest 48 z 56 bitów klucza,
- prawa część danych rozszerzana jest do 48-bitów za pomocą permutacji rozszerzonej
- rozszerzona prawa połowa jest sumowana modulo 2 z wybranymi wcześniej (i przesuniętymi) 48 bitami klucza
- zsumowane dane dzielone są na osiem 6-bitowych bloków i każdy blok podawany jest na wejście jednego z S-bloków (pierwszy 6-bitowy blok na wejście pierwszego S-bloku, drugi 6-bitowy blok na wejście drugiego S-bloku, itd.).

# Algorytm DES

Operacje te wyglądają następująco:

- Pierwszy i ostatni bit danych określa wiersz, a pozostałe bity kolumnę S-BOXa.

- Po wyznaczeniu miejsca w tabeli, odczytuje się wartość i zamienia na zapis dwójkowy.

- Wynikiem działania każdego S-bloku są 4 bity wyjściowe – tworzą one 32-bitowe wyjście S-bloków. Każdy S-Blok ma inną strukturę wyjście S-bloków poddawane jest permutacji w P-blokach

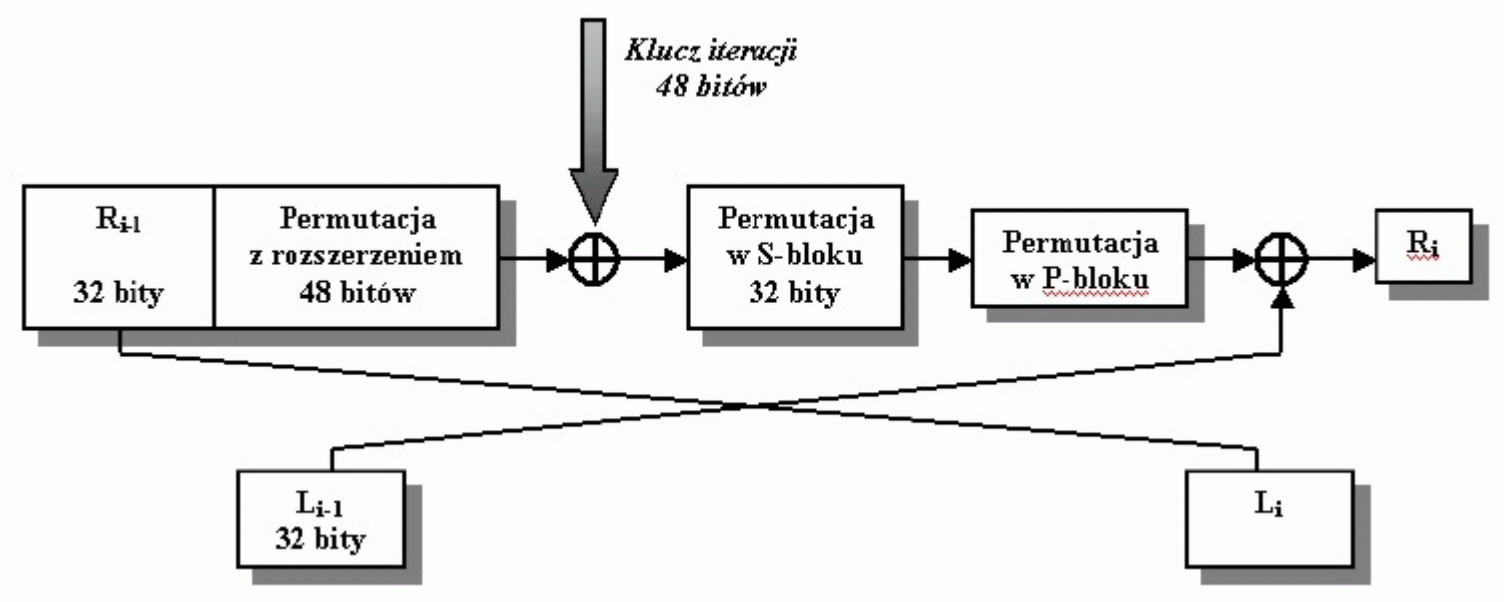
# Algorytm DES

Operacje te wyglądają następująco:

- bity tak przekształconego bloku sumowane są z bitami lewej połowy danych - tak zmieniony blok staje się nową prawą połową, poprzednia prawa połowa staje się natomiast lewą połową - cykl dobiega końca

- po wykonaniu 16 cykli operacji lewa i prawa połowa danych jest łączona za pomocą operacji XOR

- dokonywana jest permutacja końcowa



# Tryby pracy - DES

elektroniczna książka kodowa – w tym trybie pracy każdy blok tekstu jawnego szyfrowany jest w blok szyfrogramu, dzięki czemu możliwe jest stworzenie książki kodowej tekstu jawnego oraz odpowiadającego mu szyfrogramu

wiązanie bloków zaszyfrowanych – tryb ten wykorzystuje mechanizm sprzężenia zwrotnego: w operacji szyfrowania bieżącego bloku tekstu jawnego wykorzystywany jest poprzedni blok szyfrogramu, w związku z czym każdy blok szyfrogramu zależny jest zarówno od bloku tekstu jawnego, jak i od poprzedniego bloku szyfrogramu

sprzężenie zwrotne szyfrogramu – tryb ten umożliwia szyfrowanie danych strumieniowych; do procesu szyfrowania informacji wykorzystywany jest rejestr, najczęściej o pojemności odpowiadającej wielkości bloku, a rozpoczęcie szyfrowania możliwe jest dopiero po odebraniu pełnego bloku danych. W jednym przebiegu n zaszyfrowanych bitów z rejestru sumowanych jest modulo 2 z n bitami tekstu jawnego – tak powstaje pierwszych n bitów szyfrogramu, bity te są następnie dodawane na koniec kolejki

sprzężenie zwrotne wyjściowe – tryb ten jest podobny do trybu sprzężenia zwrotnego szyfrogramu – z tą różnicą, że na koniec kolejki dodawane jest n bitów poprzedniego bloku wyjściowego, a nie szyfrogramu

# Klucze słabe i półsłabe

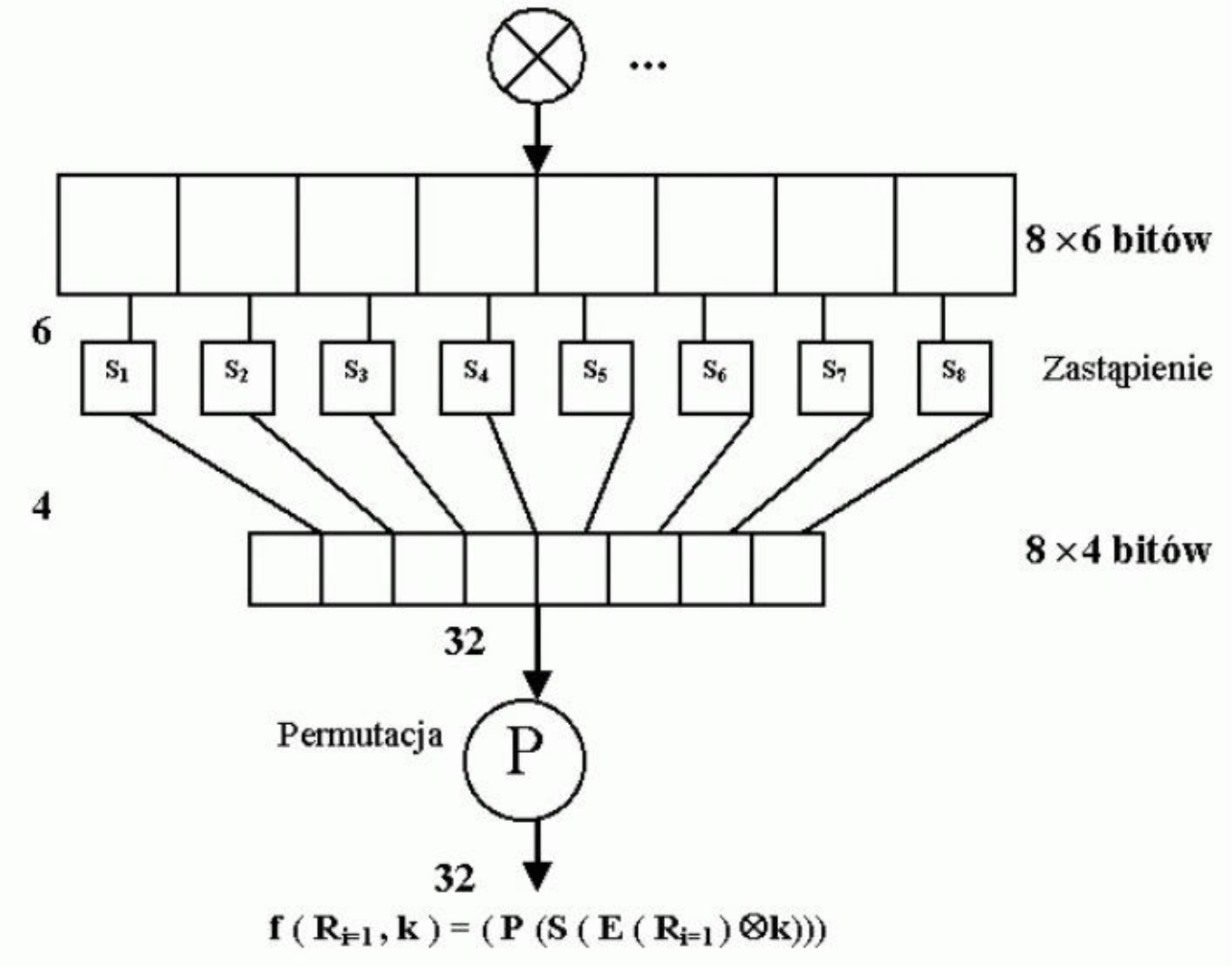
Przy wykorzystaniu takiego klucza podklucz wykorzystywany do szyfrowania będzie taki sam w każdym cyklu.

Do kluczy słabych w DES należą następujące klucze (zapis w systemie szesnastkowym):

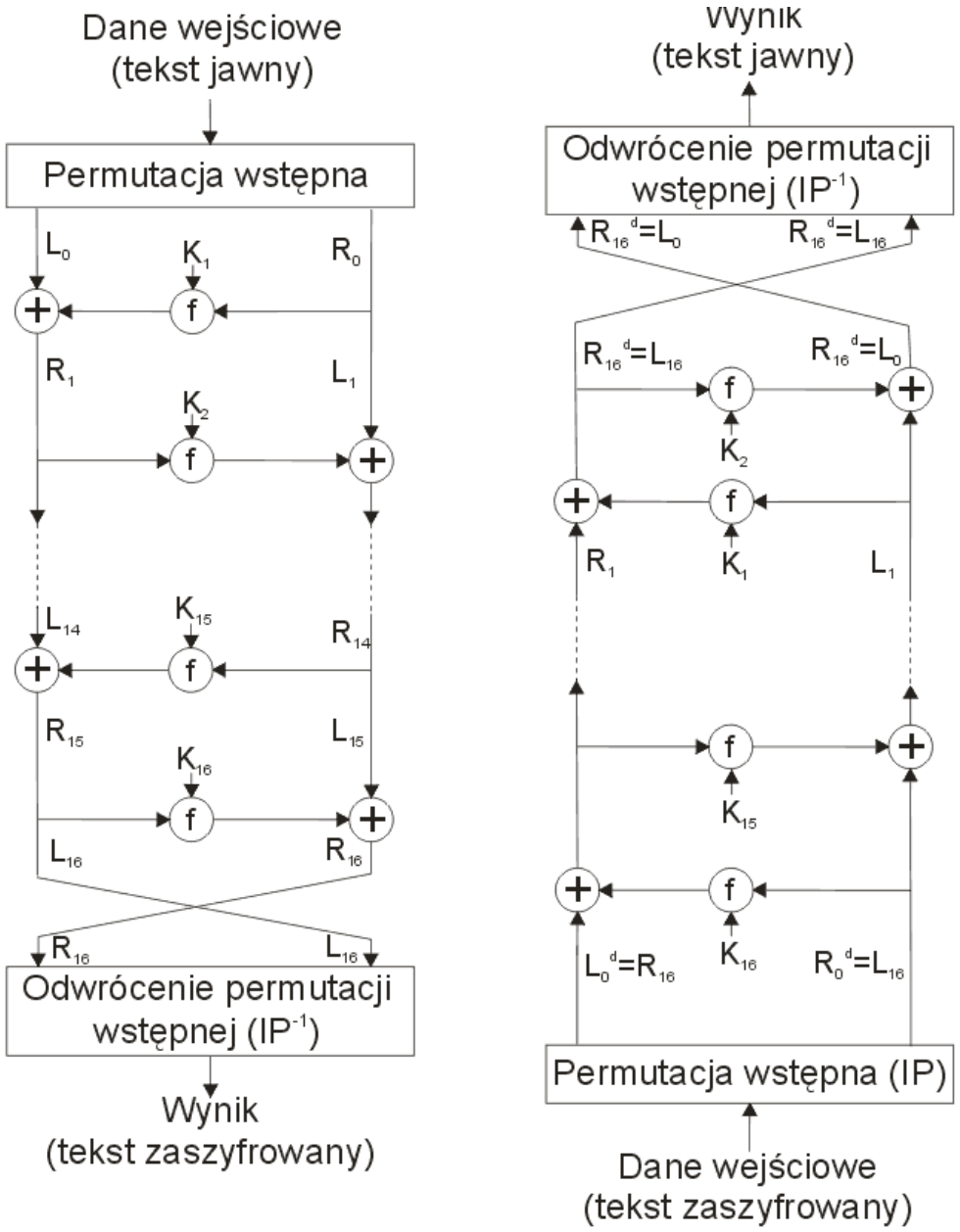
| | |
|---|---|
| klucz składający się z samych zer: | 00 00 00 00 00 00 00 |
| klucz składający się z samych jedynek: | FF FF FF FF FF FF |
| klucze, w których połowy składają się z samych zer lub jedynek: | FF FF FF F0 00 00 00<br>00 00 00 0F FF FF FF |

*Klucz półsłaby* - istnieje sześć par takich kluczy, które dany tekst jawny szyfrują do takiego samego szyfrogramu. Oznacza to także, że tekst zaszyfrowany jednym kluczem półsłabym może zostać odszyfrowany za pomocą drugiego klucza z pary.

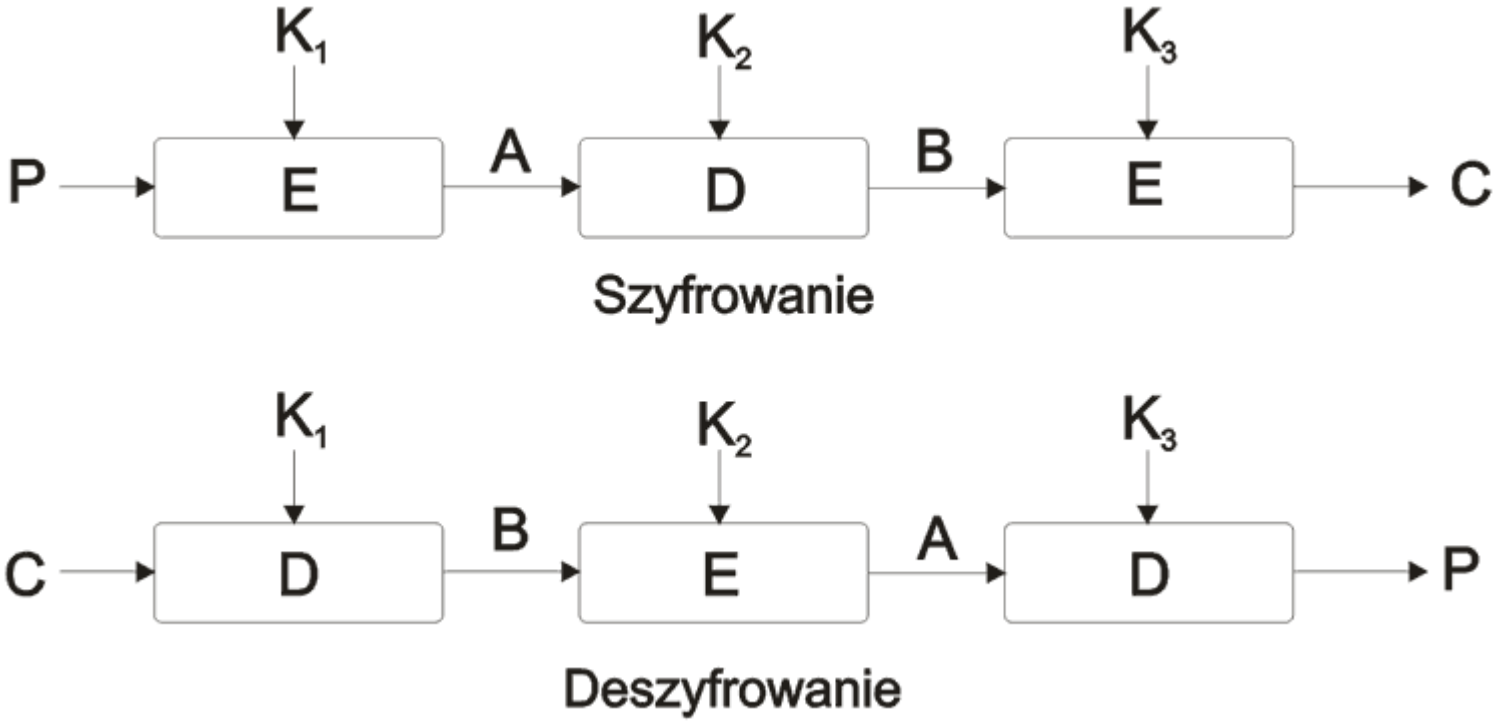
# Odzyskanie 32-bitowej struktury połowy zaszyfrowanego bloku danych po operacji sumy symetrycznej



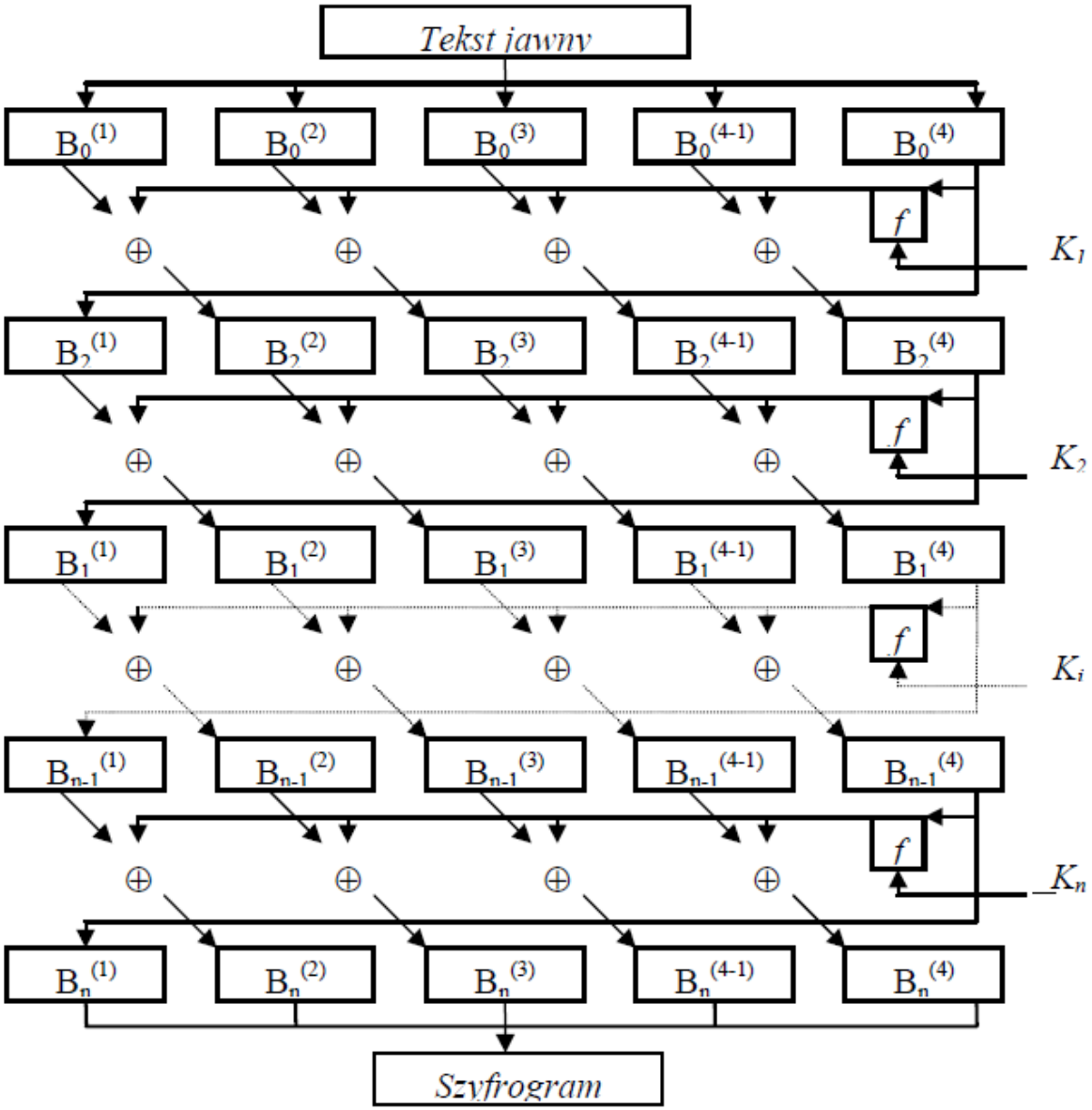
# Porównanie procesu szyfrowania i deszyfrowania



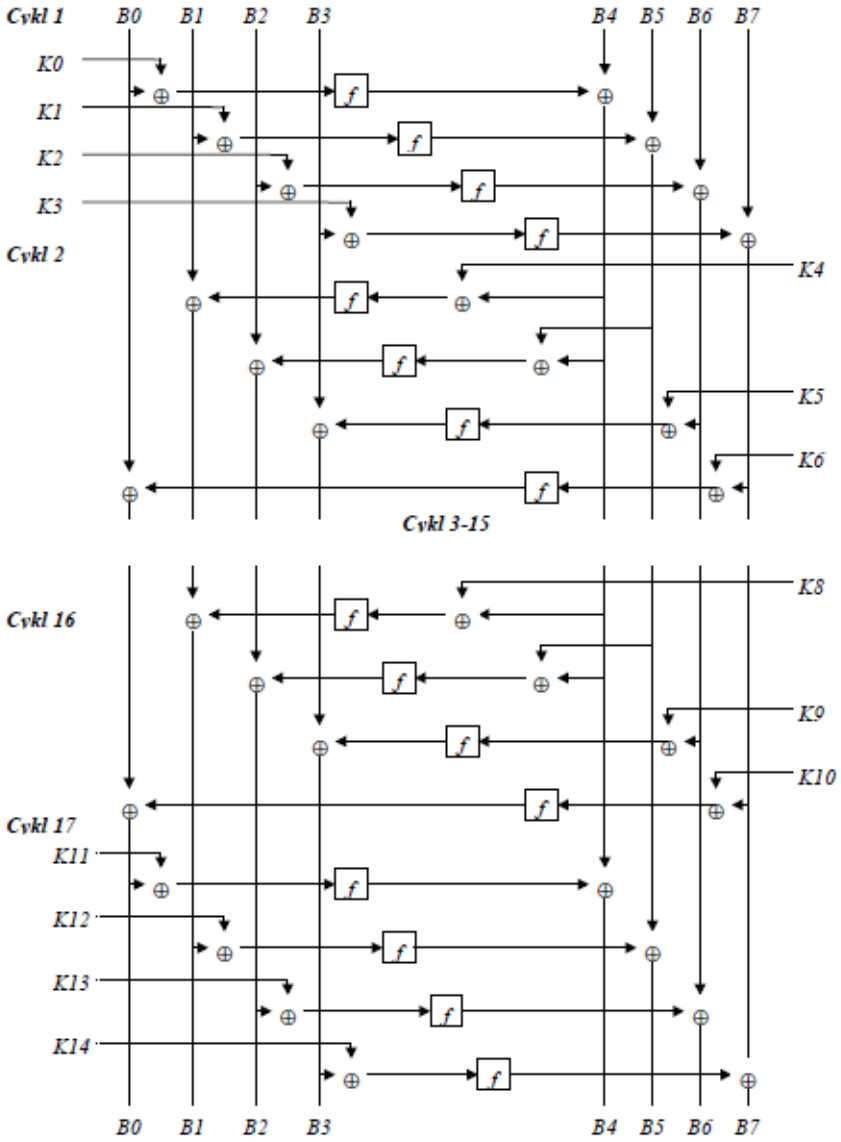
# Potrójny DES



# G-DES


Tekst jawny
$B_0^{(1)}$ $B_0^{(2)}$ $B_0^{(3)}$ $B_0^{(4-1)}$ $B_0^{(4)}$
f
$K_1$
$B_2^{(1)}$ $B_2^{(2)}$ $B_2^{(3)}$ $B_2^{(4-1)}$ $B_2^{(4)}$
f
$K_2$
$B_1^{(1)}$ $B_1^{(2)}$ $B_1^{(3)}$ $B_1^{(4-1)}$ $B_1^{(4)}$
f
$K_i$
$B_{n-1}^{(1)}$ $B_{n-1}^{(2)}$ $B_{n-1}^{(3)}$ $B_{n-1}^{(4-1)}$ $B_{n-1}^{(4)}$
f
$K_n$
$B_n^{(1)}$ $B_n^{(2)}$ $B_n^{(3)}$ $B_n^{(4-1)}$ $B_n^{(4)}$
Szyfrogram

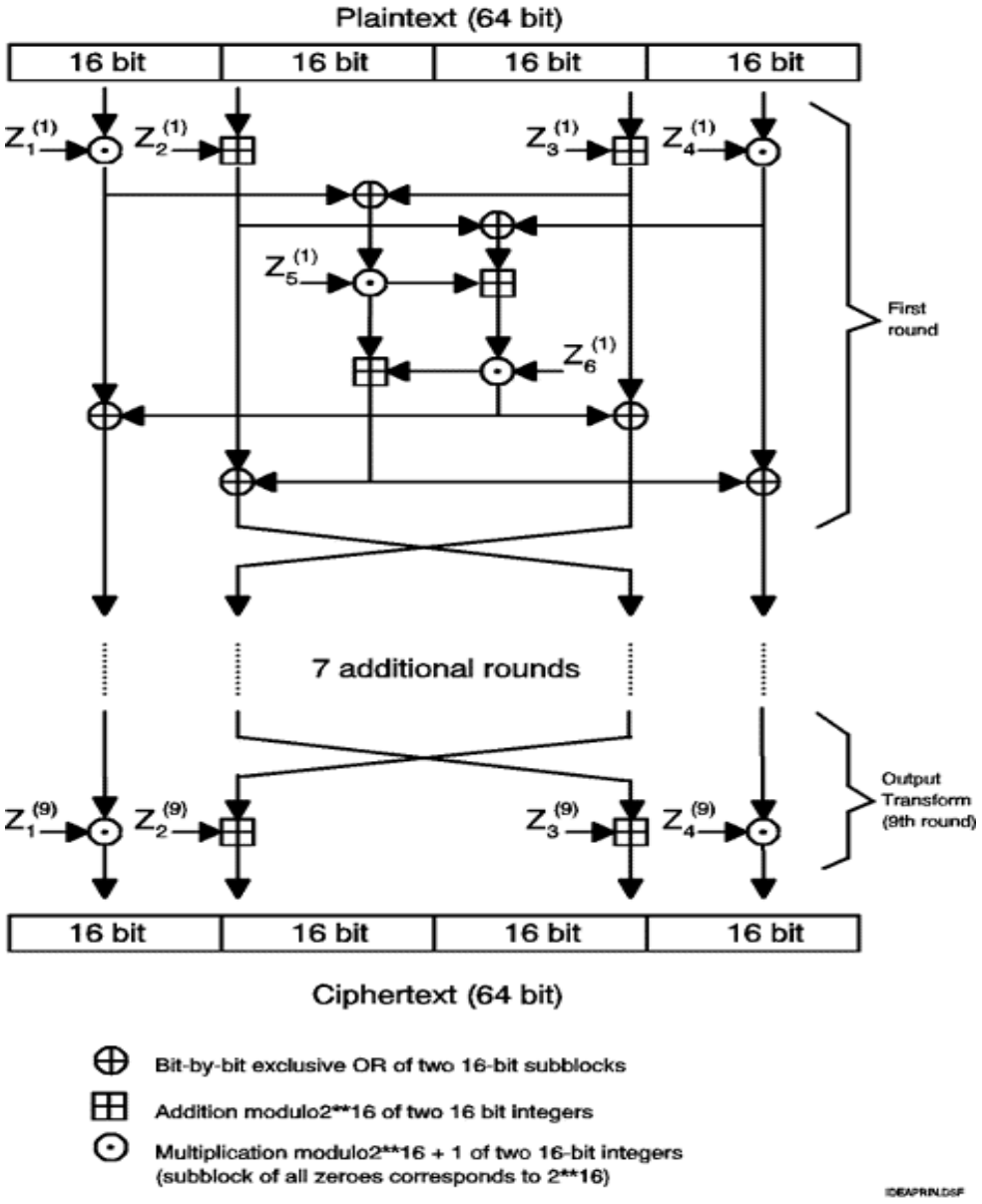
# New-DES


Cykl 1
B0
B1
B2
B3
B4
B5
B6
B7
K0
K1
K2
K3
Cykl 2
K4
K5
K6
f
Cykl 3-15
Cykl 16
K8
K9
K10
Cykl 17
K11
K12
K13
K14

# IDEA – INTERNATIONAL DATA ENCRYPTION ALGORITHM

- IDEA używa 128 bitowego klucza i jest generalnie uważany za bardzo bezpieczny.
- Jest obecnie jednym z najbardziej znanych publicznie algorytmów.
- Jest stosunkowo nowym algorytmem.
- Nie udało się dotychczas przeprowadzić na niego udanego ataku.
- IDEA jest opatentowany w USA i w większości krajów europejskich.
- Nie komercyjne użycie IDEA jest darmowe.
- Poleca się używanie tego algorytmu.

# Algorytm IDEA



# Algorytm IDEA

- **MD5** jest również jedną z najpopularniejszych obecnie tzw. funkcji skrótu kryptograficznego (obok np. starszego **MD4**, czy nowszego **SHA1**, czyli *Secure Hash Algorithm 1*).

- **MD5** jest skrótem od angielskiej nazwy **Message-Digest algorithm 5** (co oznacza **Skrót Wiadomości wersja 5)**.

- Jest to szeroko stosowany algorytm haszujący, który z dowolnego ciągu danych generuje 128-bitowy skrót.

- Funkcje skrótu kryptograficznego są to funkcje, które na podstawie pewnego tekstu generują relatywnie krótki tekst (czy też liczbę, zwaną sumą kontrolną) związaną w pewien sposób z tym tekstem (np. może to być suma kodów ASCII wszystkich znaków tekstu).

# Funkcja skrótu

Aby funkcja była dobrą funkcją skrótu kryptograficznego musi spełniać dodatkowe założenie:

- funkcja musi być jednokierunkowa, tzn. nie może być znany łatwy sposób generowania tekstów, dla których funkcja skrótu zwróci zadaną wartość (np. taką samą jak dla pewnego danego tekstu).

- funkcja może być wykorzystywana do kontroli integralności danych: jeżeli obliczona dwukrotnie wartość funkcji dla pewnego tekstu uległa zmianie, to zmianie uległ sam tekst;

- Jeżeli wartość funkcji pozostaje ta sama, to wejściowy tekst zapewne nie został zmodyfikowany.