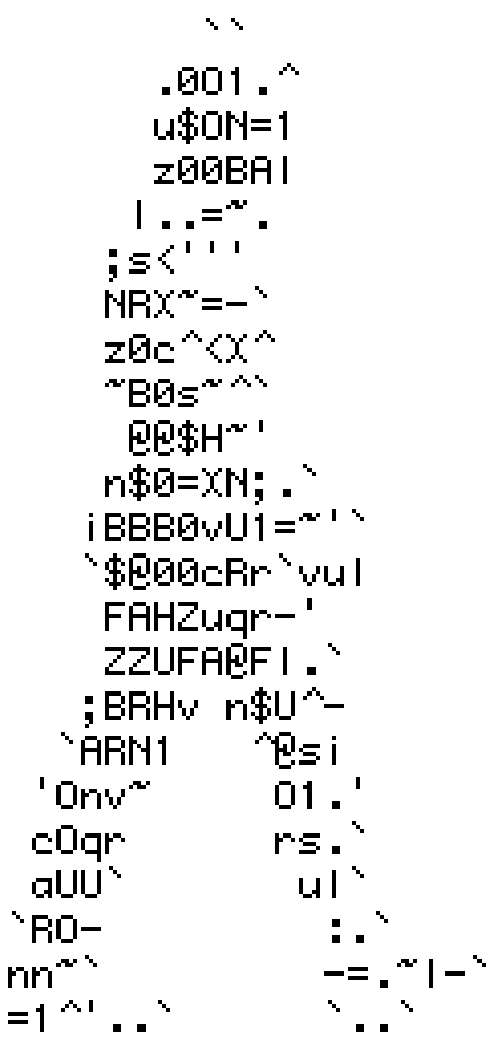


Podstawowe elementy kryptografii

Zagadnienia

część I

1. Terminologia
2. Szyfry symetryczne
3. Szyfry asymetryczne
4. Zarządzanie kluczami (PKI)
5. Funkcje skrótu i podpis cyfrowy
6. Uwierzytelnianie kryptograficzne
7. Narzędzia
8. Aspekty prawne wykorzystania kryptografii

Krótka historia kryptografii

Kamienie milowe postępu kryptograficznego:

ok. 1900 p.n.e. – pierwsze znane przykłady przekształceń kryptograficznych (w inskrypcjach grobowych)

ok. 475 p.n.e. – w Sparcie pierwsze zastosowanie szyfrowania w celach łączności

ok. 60 p.n.e. – szyfr Cezara

1412 – egipski uczony Kalkashandi pisze pierwszy znany traktat o kryptologii

1917 – Edward Hugh Hebern tworzy pierwszy rotor

1971 – Lucifer – system szyfrowania stworzony przez IBM (Horst Faistel, IBM 3514)

1975 – opracowanie standardu szyfrowania danych DES

1976 – Diffie i Hellman – pierwszy algorytm szyfrowania asymetrycznego

1978 – Rivest, Shamir, Adelman – algorytm RSA

1991 – James L.Massey i Xuejia Lai – algorytm IDEA

Podstawowe pojęcia

Kryptologia – wiedza dotycząca bezpiecznej komunikacji, obejmująca ***kryptografię*** i ***kryptoanalizę***.

Kryptografia – dziedzina wiedzy obejmująca zagadnienia związane z utajnieniem danych (przesyłanie wiadomości i zabezpieczenia dostępu do informacji) przed niepożądanymi osobami.

Utajnienie oznacza tu, że wiadomość jest trudna do odczytania (rozszyfrowania) przez osobę nie znającą tzw. klucza rozszyfrowującego – dla niej wiadomość będzie wyłącznie niezrozumiałym ciągiem znaków.

Kryptoanaliza – dziedzina wiedzy zajmująca się łamaniem szyfrów, czyli odczytywaniem zaszyfrowanych wiadomości bez znajomości kluczy rozszyfrowujących.

Podstawowe pojęcia

Kryptogram (***szyfrogram***) – zaszyfrowana postać wiadomości czytelnej.

Klucz szyfrowania – ciąg danych służących do szyfrowania wiadomości czytelnej w kryptogram za pomocą algorytmu szyfrowania. Klucz ten jest odpowiednio ustalany (uzgadniany) przez nadawcę w fazie szyfrowania.

Klucz rozszyfrowujący – ciąg danych służących do rozszyfrowania kryptogramu do postaci wiadomości czytelnej za pomocą algorytmu deszyfrowania. Klucz ten odpowiada kluczowi szyfrowania wykorzystanemu w fazie szyfrowania.

Przemienność kluczy oznacza, że role dwóch kluczy z pary mogą ulec przestawieniu. W takiej parze kluczy informację zaszyfrowaną jednym kluczem można rozszyfrować tylko przy pomocy odpowiadającego mu drugiego klucza z pary, i odwrotnie, informację zaszyfrowaną drugim kluczem można rozszyfrować wyłącznie przy pomocy klucza pierwszego.

Proste szyfry

Szyfrowanie metodą podstawiania

Monogram, przekształcenie szyfrujące $f(x) = x + \Delta$

szyfr Cezara "A" $\Rightarrow$ ("A" + 3) = "D"

kod Captain Midnight "A" $\Rightarrow$ ("A" + Δ); Δ = 1,...,26

Oryginalny znak	Znak podstawiany
A	D
B	E
C	F
⋮	⋮
W	Z
X	A
Y	B
Z	C

Macierz podstawiania

Przykład

Szyfruj		Prześlij		Deszyfruj
P →	S	→ S	→	P
R →	U	→ U	→	R
I →	L	→ L	→	I
V →	Y	→ Y	→	V
A →	D	→ D	→	A
T →	W	→ W	→	T
E →	H	→ H	→	E

Proste szyfry

Szyfrowanie metodą podstawiania

Szyfry monoalfabetyczne

- $f(x) = x + k$ $\quad E[x|k] = x + k$ $\quad E_k[x] = x + k$
- $f(x) = x \cdot k \mod n$ $\quad E[x|k] = x \cdot k \mod n$
- $f(x) = (x \cdot a + b) \mod n$ $\quad E[x|a,b] = (x \cdot a + b) \mod n$
- permutacja alfabetu

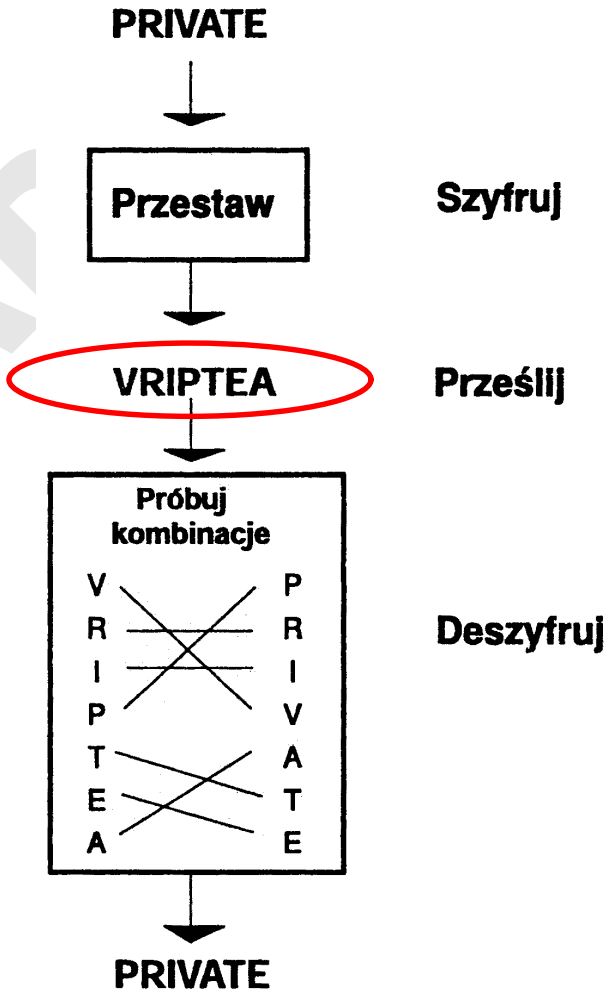
  dla alfabetu 26-znakowego możliwych $26! = 4 \cdot 10^{26}$ permutacji

(1μs · 26! = 10 trylionów lat)

Proste szyfry

Szyfrowanie metodą przestawiania

Przestawianie losowe:



Przestawianie z figurą geometryczną

- figura geometryczna definiuje transpozycję wiadomości czytelnej

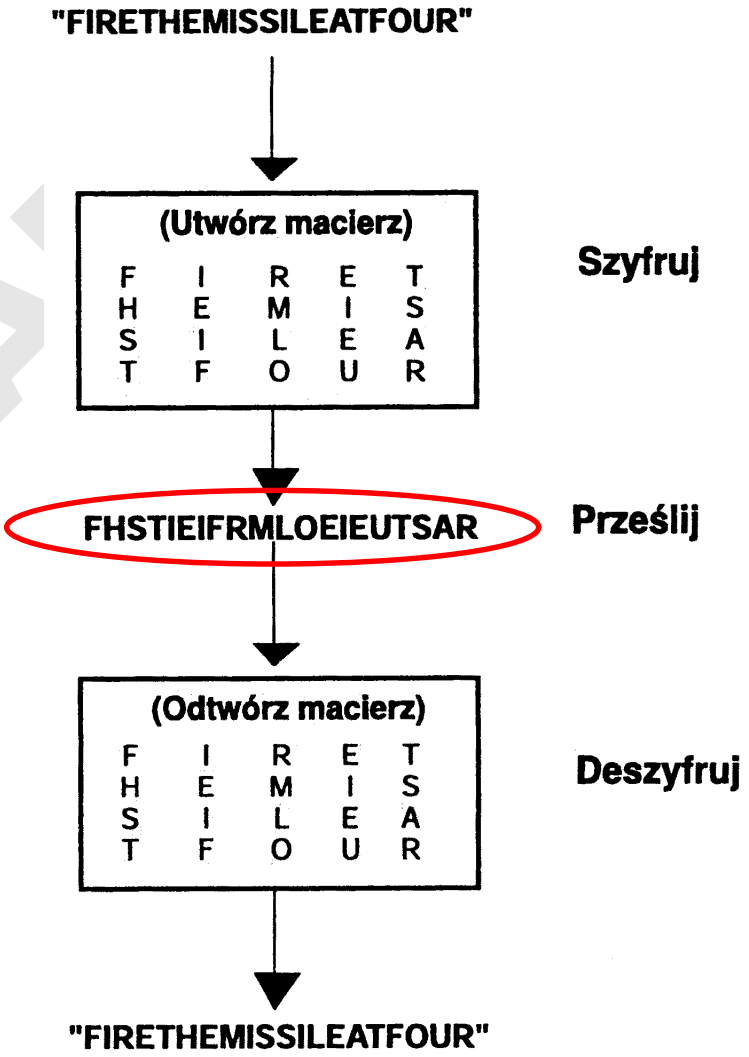
Proste szyfry

Szyfrowanie metodą przestawiania

Przestawianie z macierzą przekształceń:

- rolę figury transpozycji pełni macierz
- kluczem jest rozmiar macierzy,

  np. $k = (5,4)$



Proste szyfry

Szyfrowanie metodą przestawiania

Przestawianie z permutowaną macierzą przekształceń:

- kluczem jest rozmiar macierzy i permutacja kolumn,
  np. $k = (5,4;2\text{-}5\text{-}3\text{-}1\text{-}4)$
- jaki będzie szyfrogram z poprzedniego przykładu dla takiego klucza?

Inne proste metody przestawiania:

- tablice o wierszach zmiennej długości
- przestawienie przekątnokolumnowe
- szyfry siatkowe

Zadania

1. Na czym polega popularnie wykorzystywane w Internecie kodowanie ROT-13?
2. Na czym polega i do czego służy kodowanie UUencoding?

Współczesna kryptografia

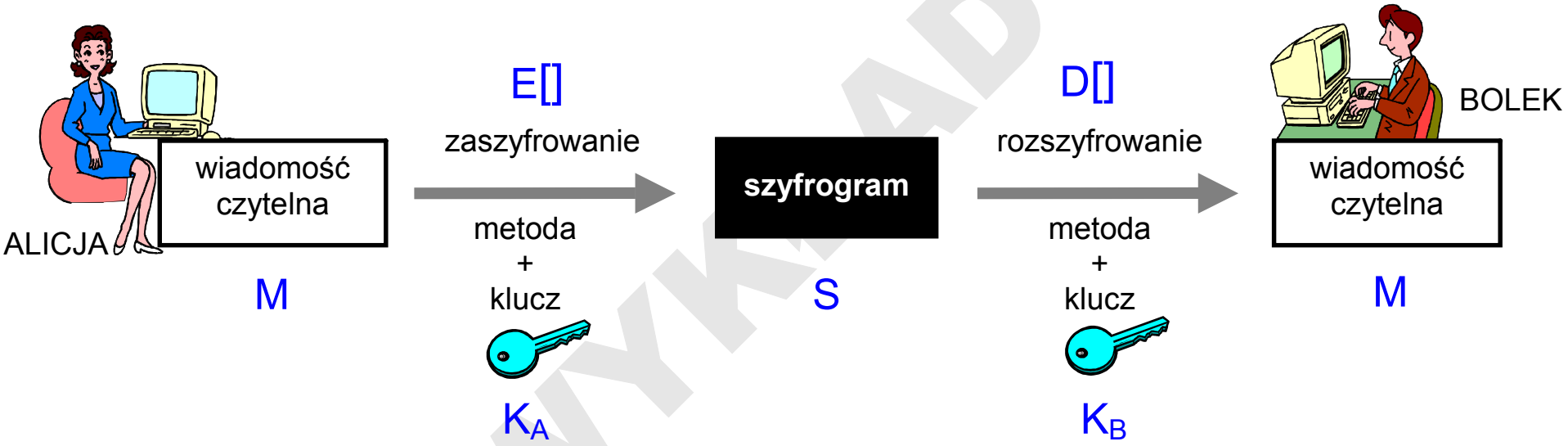
Zasada Kerckhoffsa

Algorytm szyfrowania i deszyfrowania jest jawny

- siła metody kryptograficznej nie polega na tajności algorytmu
- lecz na tajności pewnego zmiennego parametru tego algorytmu (klucza)

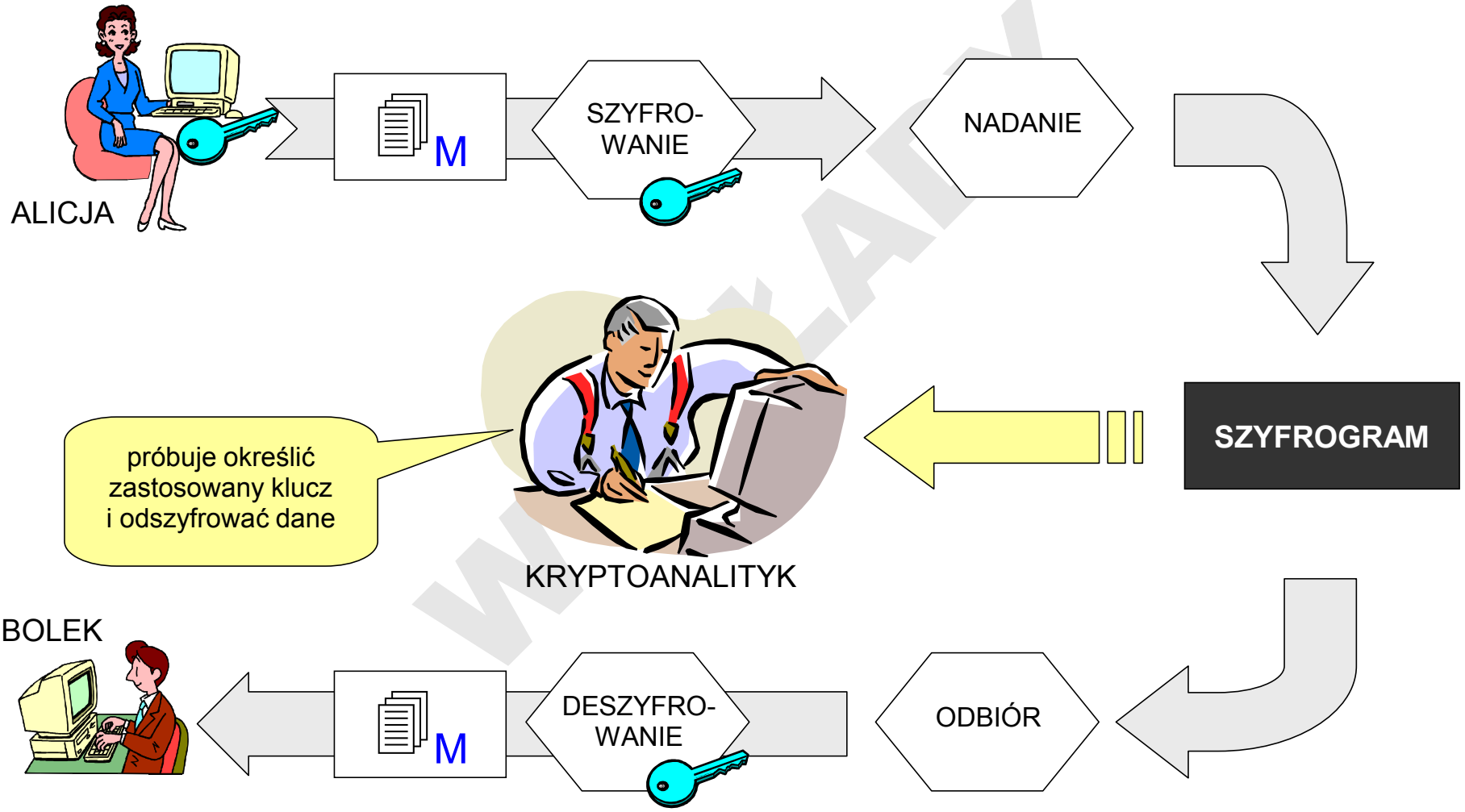
Szyfrowanie z kluczem

Schemat ogólny



$$E_{K_A}[M] = S \rightarrow S \rightarrow D_{K_B}[S] = M$$

Kryptoanaliza



Kryptoanaliza

Wybrane ataki kryptoanalityczne:

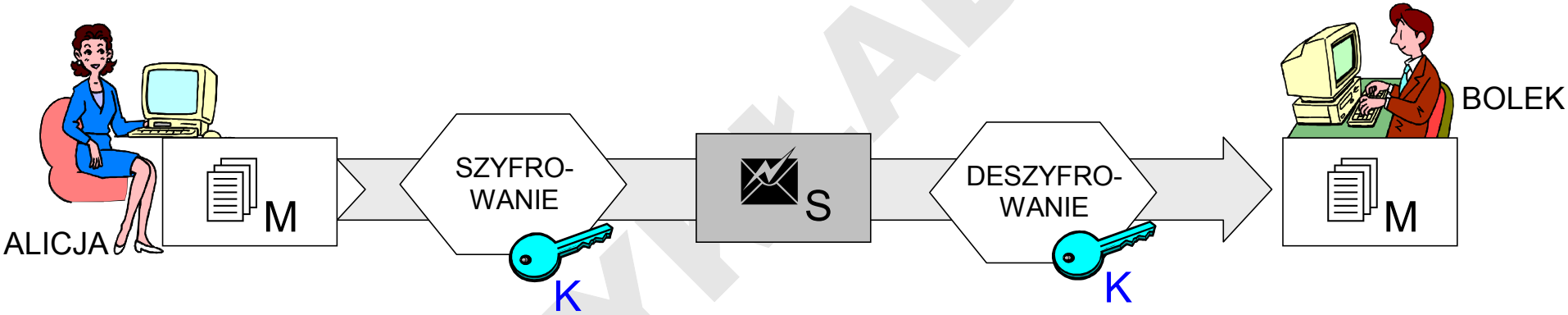
- **przeszukiwanie wyczerpujące** dziedziny kluczy – *brute force*
- **atak znanym tekstem jawnym** – atakujący ma parę (lub wiele par) tekstu zaszyfrowanego i jawnego (tzw. ściąga, ang. *crib*)
- **atak wybranym tekstem jawnym** – atak aktywny: zmuszasz nadawcę do zaszyfrowania twojego tekstu jego kluczem
- **atak wyroczni** – atak aktywny: napastnik może zyskać pewne korzyści dzięki wysyłaniu zapytania do jednej lub do drugiej strony posługując się uczestnikami komunikacji jako wyroczniami
- **atak urodzinowy** – atak na funkcje mieszające w celu znalezienia dwóch komunikatów dających ten sam skrót

Szczególne odmiany kryptoanalizy:

- kryptoanaliza „praktyczna” – zdobycie klucza wszelkimi koniecznymi środkami
- kryptoanaliza gumowego węża – przy użyciu środków fizycznych lub finansowych

Szyfrowanie symetryczne

- wspólny tajny klucz K_{A-B} (dalej oznaczany K)
- $E_K[M] = S \rightarrow S \rightarrow D_K[S] = M$



Cecha:

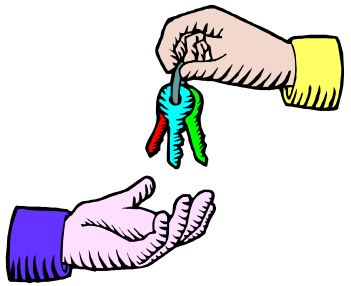
- **$D_K[E_K[M]] = M$**

Szyfrowanie symetryczne

Podstawowe problemy:

problem tajności klucza

- wiadomość jest bezpieczna dopóki osoba trzecia nie pozna tajnego klucza **K**



problem dystrybucji klucza

- jak uzgodnić wspólny klucz bez osób trzecich, będąc oddalonym o setki, a nawet tysiące kilometrów

problem skalowalności

- 2 os. = 1 kl.; 3 os. = 3 kl.; 4 os. = 6 kl.; 10 os. = 45 kl.; 100 os. = 4950 kl.; ...

autentyczność

- czy tajność klucza zapewnia autentyczność?

DES

Algorytm DES (*Data Encryption Standard*)

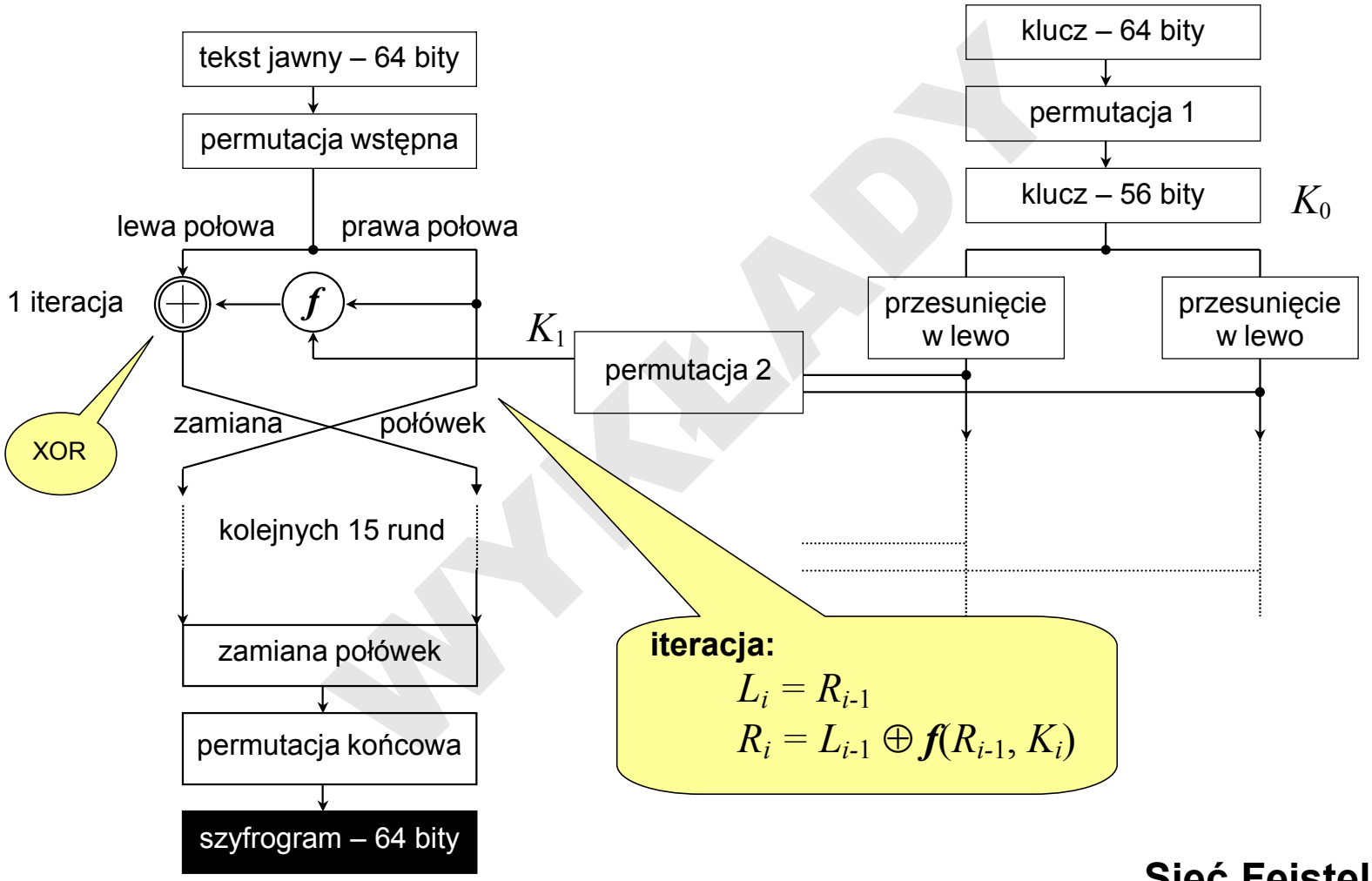
- opracowany w latach ’70. przez IBM na zamówienie NSA (*National Security Agency*)
- zmodyfikowany przez NSA i przyjęty w 1976 przez NBS (*National Bureau of Standards*, obecnie NIST = *National Institute of Standards and Technology*) – ochrona patentowa już wygasła
- opublikowany w 1977 przez nieporozumienie między NSA a NBS
- pracuje na 64-bitowych blokach tekstu jawnego, odpowiada to 8 literom 8b ASCII
- klucz składa się z 64 bitów (przy czym 8 bitów jest bitami parzystości) – w istocie, w trakcie wyboru klucza można określić jedynie 56 bitów

DES

Fazy działania (sieć Feistela)

- wstępna permutacja wejściowego bloku danych (na podstawie tabeli transpozycji)
- podział bloku na lewą i prawą połowę o długości 32 bitów każda
- 16 jednakowych cykli operacji podstawiania i przestawiania – funkcje f, w czasie których dane zostają połączone z kluczem.
- połączenie lewej i prawej połowy bloku
- permutacja końcowa (odwrotność permutacji wstępnej)

DES



Sieć Feistela

DES

Fazy

- każda kolejna runda, dokonuje tych samych obliczeń, ale na wynikach obliczeń z poprzedniej rundy i specjalnym podkluczu K_i generowanym z 56b klucza K_0 (64-bitowego po usunięciu 8 bitów parzystości)

Początek iteracji:

- 56 bitów klucza dzielone jest na dwie połowy po 28 bitów
- w każdej iteracji bity obu połówek są cyklicznie przesuwane w lewo o jeden lub dwa bity, w zależności od numeru iteracji
- ostatecznie wykonywana jest permutacja kompresująca, dzięki której z 56b klucza, otrzymujemy 48b podklucz używany w funkcji $f(L_i, K_i)$
- klucz ten (K_i) podawany jest do następnej iteracji $i+1$

DES

Fazy

Funkcja f:

- prawa połowa R_{i-1} rozszerzana jest z 32 bitów do 48 bitów za pomocą permutacji rozszerzonej (e-blok) i sumowana mod 2 z 48 bitami podklucza K_i danego cyklu
- otrzymany wynik poddawany jest operacji podstawienia poprzez wykorzystanie bloków podstawień (S-bloki):
  - ciąg 48 bitów dzielony jest na 8 bloków po 6 bitów
  - każdy ciąg 6 bitów jest redukowany do 4 bitów funkcją podstawienia
  - z 48 bitów otrzymujemy 32b ciąg, który poddawany jest permutacji zwykłej
  - następnie sumowany mod 2 z lewą połową L_{i-1} bloku wejściowego

Koniec iteracji:

- zamiana lewej i prawej połowy miejscami

DES

Bloki postawień (S-bloki)

- tabela 4 wiersze na 16 kolumn
- element tabeli jest 4b liczbą (podstawiana wartość)
- wybór wiersza i kolumny za pomocą 6b wejścia:
  - pierwszy i ostatni bit 6b ciągu tworzy 2b liczbę
  - liczba ta wybiera wiersz
  - pozostałe środkowe 4 bity wybierają kolumnę
- wskazany element tabeli (4b) jest podawany na wyjście
- każdy z 8 S-bloków jest inny (→ [Schneier], [Stallings])

DES

Deszyfrowanie

- podobnie jak szyfrowanie
- klucze stosowane są w kolejności odwrotnej od K_{16} do K_1

DES

Tryby pracy

ECB (*Electronic CodeBook*) – podstawowy tryb szyfrowania blokowego

- cały tekst jawny jest dzielony na bloki 64b (ostatni – *padding*)
- każdy 64b blok jest szyfrowany niezależnie
- dla danego bloku i danego klucza wynik szyfrowania będzie zawsze ten sam
- jeśli blok wystąpi w wiadomości częściej niż raz – za każdym razem otrzyma taki sam blok szyfrogramu ECB
- przy pewnym standardowym formacie wiadomości (np. rozpoczynających się od tych samych stałych pól) – ułatwienie dla kryptoanalityka

DES

Tryby pracy

CBC (*Cipher Block Chaining*)

- na pierwszym 64b bloku jest wykonywana operacja XOR z pewnym wektorem początkowym (IV = *Initialization Vector*) znanym nadawcy i odbiorcy
  $S_i = E_K [M_i \oplus IV]$
- wynikowy ciąg jest podawany na wejście algorytmu DES
- na każdym kolejnym 64b bloku jest wykonywany XOR z zaszyfrowanym poprzednim blokiem przed podaniem na wejście algorytmu DES
  $S_i = E_K [M_i \oplus S_{i-1}]$
- powtórzone takie same bloki 64b dadzą bloki zaszyfrowane różnej postaci
- deszyfrowanie:
  $M_i = D_K [S_i] \oplus S_{i-1}$

DES

Tryby pracy

CFB (*Cipher FeedBack*) i OFB (*Output FeedBack*) – tryby sprzężenia zwrotnego

- tryby szyfrowania strumieniowego – szyfruje każdorazowo po jednym znaku 8b
- szyfrogram jest tej samej długości co tekst jawny
- przydatne do tekstów zmiennej długości (danych asynchronicznie wprowadzanych – jak znaki z klawiatury)
- w **CFB** na wejście funkcji szyfrującej podawana jest zawartość 64b rejestru przesuwnego – początkowo zawiera on IV, który jest szyfrowany: $R_1 = E_K [IV]$
- na ośmiu najstarszych bitach rejestru jest wykonywany XOR ze znakiem szyfrowanym M_i: $S_i = R_i \oplus M_i$
- zawartość rejestru jest przesuwana w lewo 8b, a jako 8 najmłodszych jest wpisywany szyfrogram S_i wprowadzonego znaku (R_{i+1})
- uszkodzenie 1 bitu propaguje się na 9 znaków szyfrogramu

DES

Tryby pracy

CFB (*Cipher FeedBack*) i OFB (*Output FeedBack*) – tryb sprzężenia zwrotnego

- w **OFB** w miejsce 8 najmłodszych bitów wpisywany jest tylko szyfrogram 8 najstarszych (bez XOR z porcją tekstu szyfrowanego)
- w trybie OFB błędy się nie propagują – uszkodzenie 1 bitu wpłynie tylko na rozszyfrowanie 1 znaku (zawierającego ten bit) – kryptoanalityk kontrolujący szyfrowany strumień może kontrolować zmiany w tekście jawnym (XOR!) → kryptoanaliza różnicowa
- IV powinien być każdorazowo unikalny, ale nie musi być tajny! (dlaczego?)

DES

Tryby pracy

Wnioski

- ECB jest trywialny
  - nie powinien być stosowany do szyfrowania sesji danych
  - może być wykorzystany do przesłania kluczy oraz IV
- problem implementacyjny z IV w trybie CBC
  - jego celem jest upodobnienie bloków szyfrogramu do postaci losowych danych
  - typowe IV wartości nijak nie przypominają losowych (często zawierają powtarzające się najstarsze bity lub mają inną łatwą do przewidzenia strukturę)
  - nawet jeśli IV byłby prawdziwie losowy, to trzeba go przekazać odbiorcy (skoro jest losowy)
  - np. wysyłając w pierwszym bloku szyfrogramu (wydłuża szyfrogram – problem z małymi porcjami szyfrowanych danych)

DES

Algorytm CDMF (*Commercial Data Masking Facility*)

- wersja uproszczona DES opracowana przez IBM
- klucz 40b (export)
- dokładniej: DES uzupełniony o wstępną metodę skracania klucza do 40b

DES

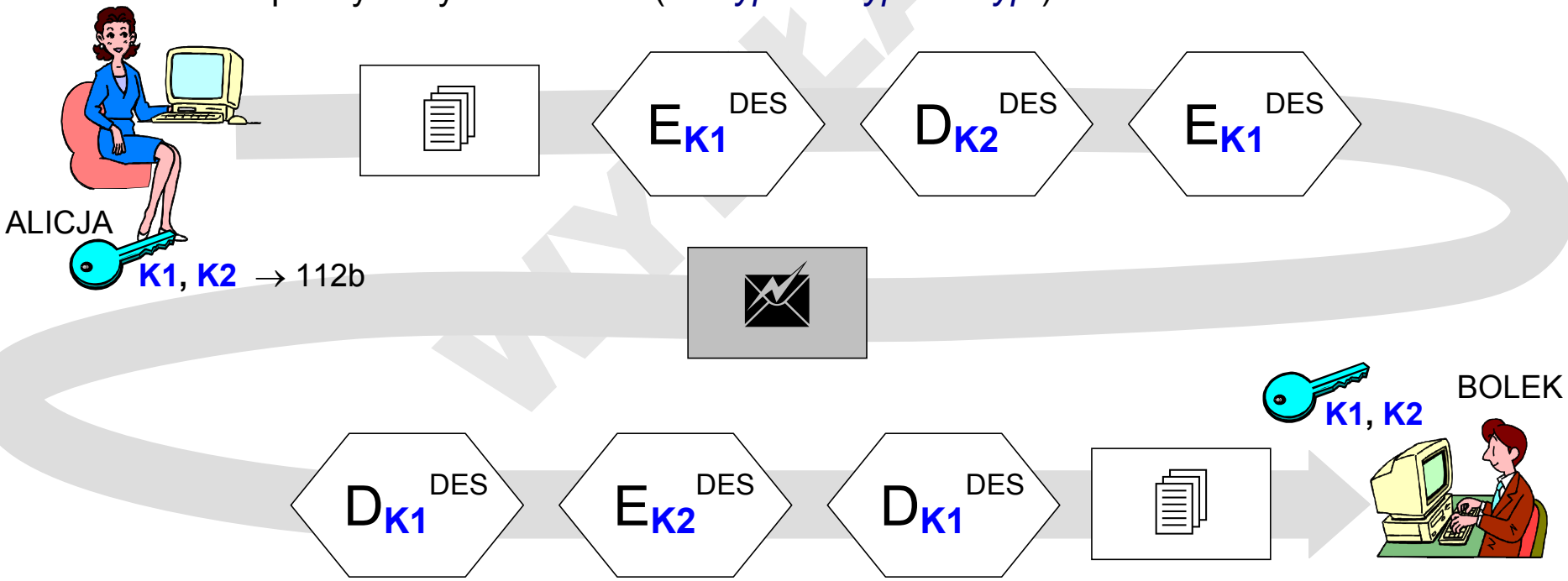
Odporność

- w 1998 r. DES z kluczem 56b został złamany w 56 godzin kryptoanalizy metodą przeszukiwania wyczerpującego
- koszt sprzętu (wówczas) 250 tys. USD (EFF DES Cracker)
- rok później zajęło to już 22 godziny
- dziś to kwestia minut

3DES

Algorytm 3DES (*Triple DES*)

- trzy iteracje szyfrowania i deszyfrowania tekstu jawnego
- każda iteracja może używać innego klucza 56b → klucz 168b
- w praktyce tryb DES-EDE (*encrypt-decrypt-encrypt*) z dwoma kluczami:



RC2 / RC4 / RC5 / RC6

- prawnie zastrzeżone algorytmy opracowane przez Rona Rivesta (RSA Data Security) – chociaż kod źródłowy szeroko dostępny w Internecie
- bardzo wydajne algorytmy symetryczne (ok. 10 razy szybsze od DES) o zmiennej długości klucza (do 2048b)
- RC2, RC5, RC6 – blokowe
- RC4 – strumieniowy
- specjalny status eksportowy USA dla kluczy 40b lub 56b (dla instytucji powiązanych z interesami USA)
- powszechnie wykorzystywane w Lotus Notes, Apple OCE (*Open Collaboration Enviromnent*), Oracle, protokołach SSL i S-HTTP, sieciach bezprzewodowych i komórkowych

CAST

Algorytmy CAST

- określenie CAST opisuje schemat zastosowany w rodzinie zbliżonych do DES algorytmów kryptograficznych o zmiennej długości kluczy i bloków
- najpowszechniej znany reprezentant – CAST-128
- opublikowany w 1997 [RFC 2144]

IDEA

Algorytm IDEA (*International Data Encryption Algorithm*)

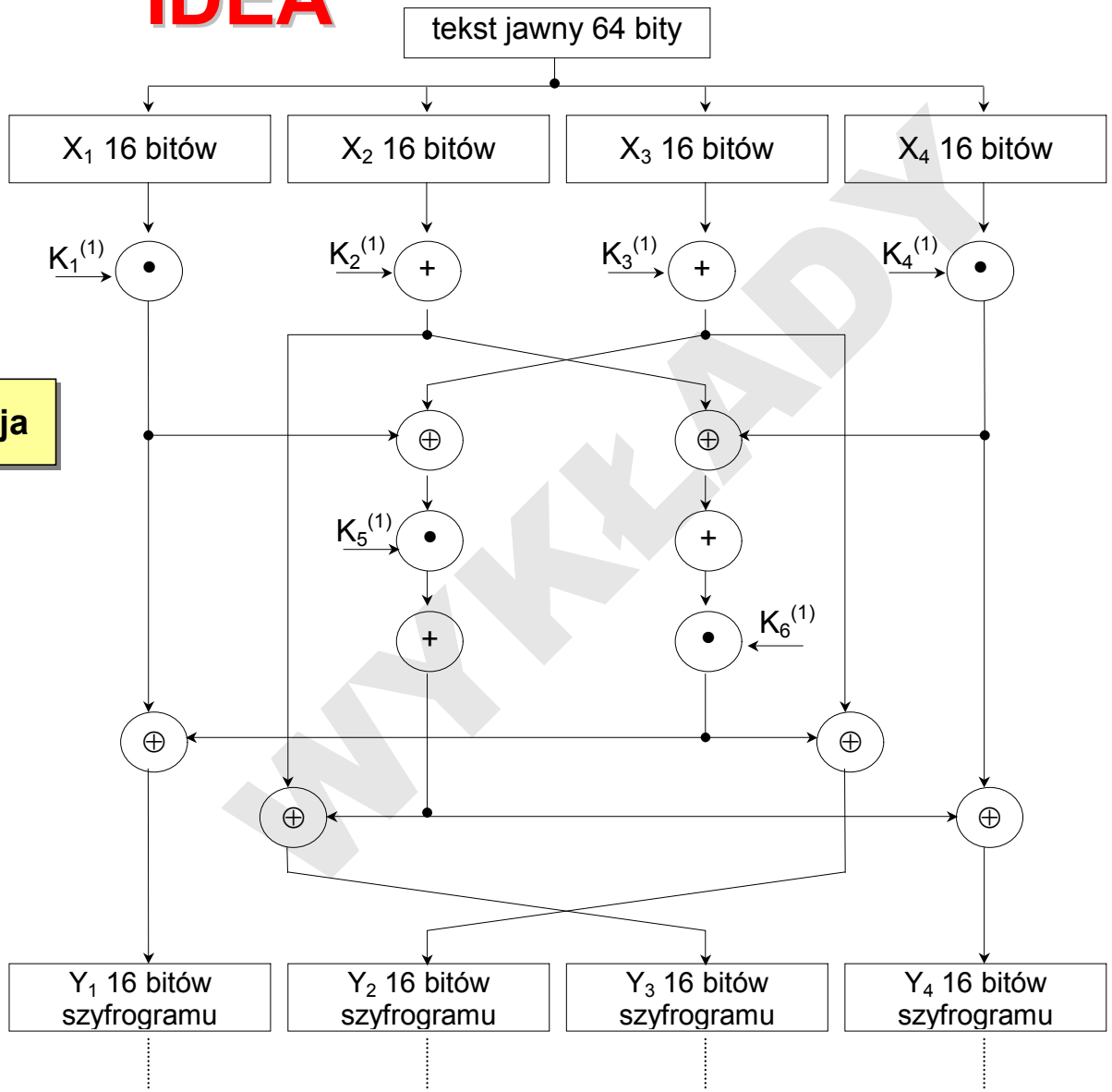
- opracowany w 1991r. przez Swiss Federal Institute of Technology (James L. Massey i Xuejia Lai)
- 64b bloki danych (jak DES)
- klucz 128b
- 64b blok dzielony na 16b podbloki
- a 128b klucz na 16b podklucze
- 8 iteracji (w DES jest 16, ale w IDEA 1 iteracja odpowiada 2 w DES)

IDEA

Fazy działania

- w każdym kroku cztery 16b podbloki danych poddawane są operacji dodawania modulo 2, dodawania modulo 2^{16} i mnożenia modulo 2^{16} z innymi blokami i z sześcioma 16b podkluczami.
- pomiędzy każdym krokiem następuje zamiana drugiego i trzeciego podbloku

IDEA



pierwsza iteracja

- + dodawanie
- • mnożenie
- ⊕ XOR
- $K_i^{(1)}$ podklucze

IDEA

Podklucze

- 128-bitowy klucz jest dzielony na osiem 16-bitowych podkluczy
- pierwszych 6 podkluczy jest używanych w pierwszej iteracji,
  2 pozostałe podklucze – w kolejnej
- następnie cały 128b klucz rotuje o 25 pozycji w lewo
- tak otrzymany klucz jest ponownie dzielony na osiem 16b podkluczy, z których pierwsze 4 uzupełniają podklucze w drugiej iteracji, a kolejne 4 są przydzielane do trzeciej iteracji
- w kluczu jest powtarzana rotacja o 25 pozycji w lewo – klucz jest ponownie dzielony na 8 podkluczy używanych w kolejnych krokach

IDEA

Podklucze

- czynności powtarzane są do momentu przydzielenia kluczy do wszystkich kroków, przy czym w fazie zakończenia zamiast sześciu podkluczy, stosuje się tylko cztery podklucze.
- łącznie w algorytmie wykorzystuje się 52 podklucze, które generowane są z klucza szyfrującego

Deszyfracja

- podklucze używane do deszyfrowania odpowiadają podkluczom szyfrowania wymienionym w odwrotnej kolejności
- operacje arytmetyczne przy użyciu czterech podkluczy są wykonywane nie na początku, ale na końcu deszyfrowania

SAFER

Algorytm SAFER

- opracowany przez Jamesa L. Masseya
- szyfr blokowy
- wersja z kluczem 64b (SAFER-K64): 6 rund
- wersja z kluczem 128b (SAFER-K128): do 12 rund (rekomendowane 10)

Blowfish

(łac. *Arothron hispidus*)

Algorytm Blowfish

- opracowany w 1994 przez Bruce'a Schneiera
- blok danych 64b, klucz podstawowy o długości do 448b
- 16 iteracji wykorzystujących wyznaczane każdorazowo (przed szyfrowaniem i deszyfrowaniem) 18 kluczy pomocniczych i 4 S-bloki 256-elementowe o wartościach zależnych od: klucza podstawowego, danych oraz liczby π
- deszyfrowanie jest operacją identyczną z szyfrowaniem – jedynie odwrotna kolejność kluczy pomocniczych

Rijndeal

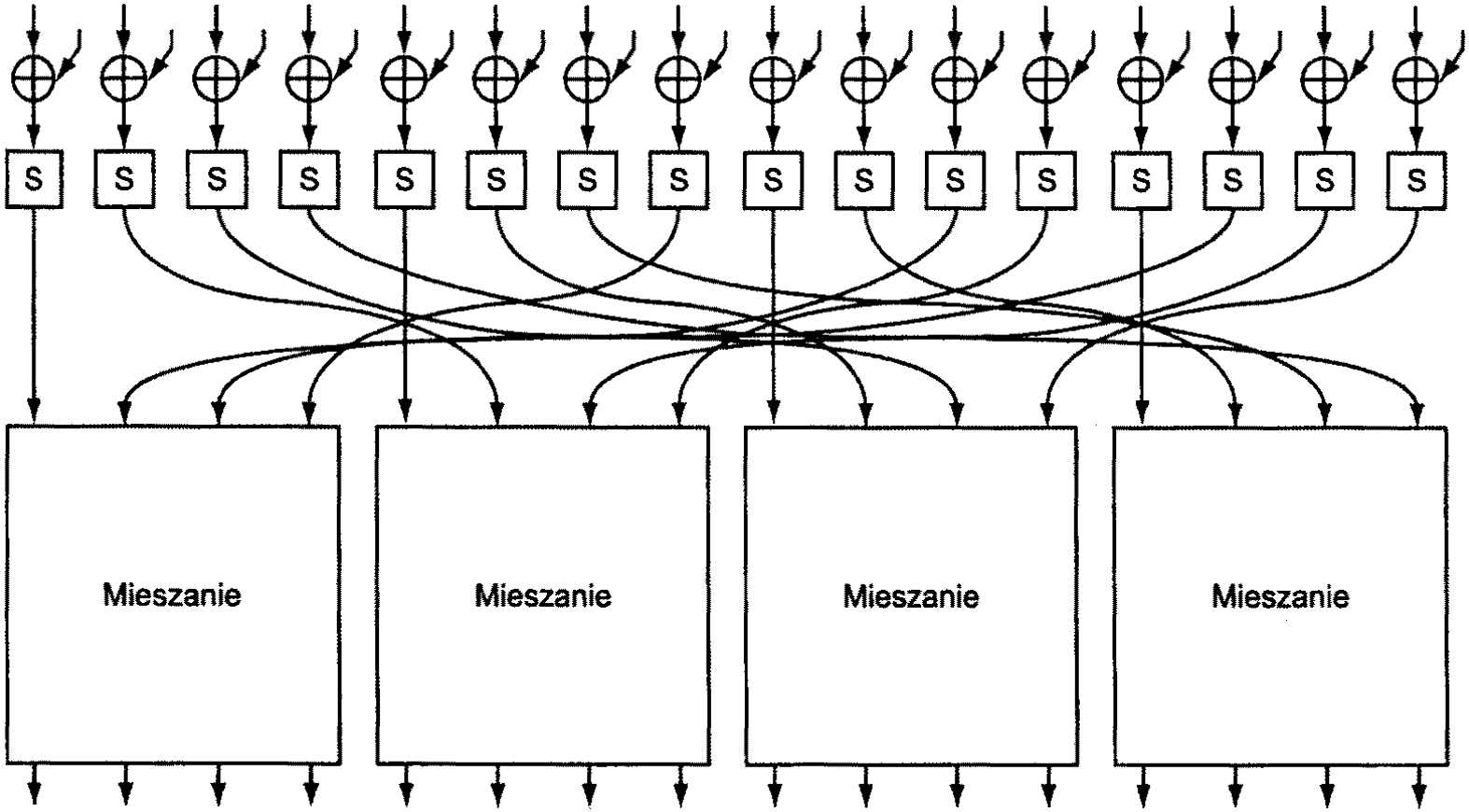
Algorytm Rijndeal

- opracowany w 1999 przez Belgów: Joana Daemena i Vincenta Rijmena
- bloki po 128b, 196b lub 256b
- klucze również 128b, 196b lub 256b
- 10 (128b), 12 (196b) lub 14 (256b) iteracji
- każda iteracja to:
  1. podstawienie (S-bloki)
  2. przesuwanie (rzędów i kolumn bloku-macierzy)
  3. XOR

  na poszczególnych bajtach danych i klucza

Rijndeal

Schemat pojedynczej iteracji



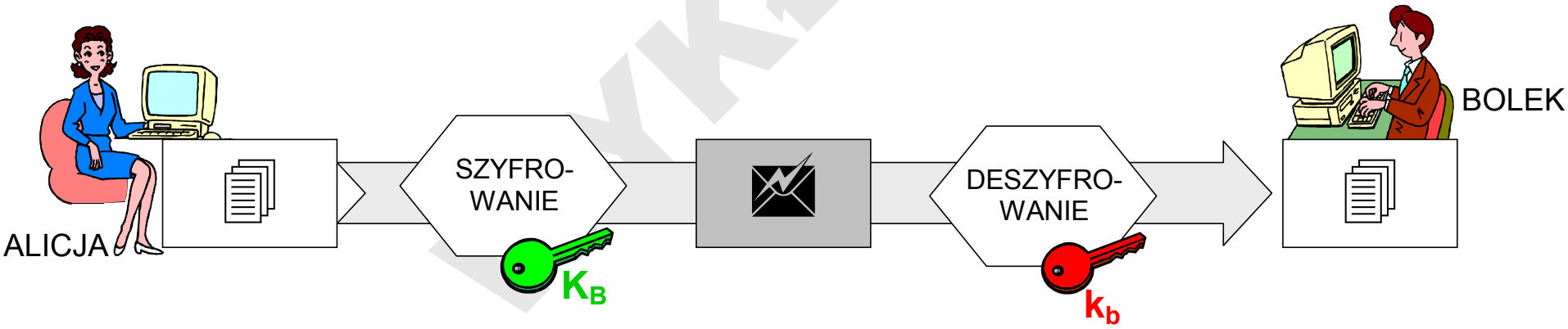
AES

AES (*Advanced Encryption Standard*)

- następca DES-a od 2001 r.
- wykorzystuje algorytm Rijndeal (wygrał rywalizację z alg. Serpent, Twofish, RC6, MARS)
- tryb blokowy (blok 128b) i strumieniowy
- klucze 128b, 192b, 256b
- dopuszczalne również inne kombinacje
- nowy strumieniowy tryb licznikowy (CTR = *Counter Mode*) rejestr jest inkrementowany wraz z kolejnymi operacjami szyfrowania porcji danych – tryb ten oferuje możliwość zrównoleglenia operacji na różnych porcjach danych

Szyfrowanie asymetryczne

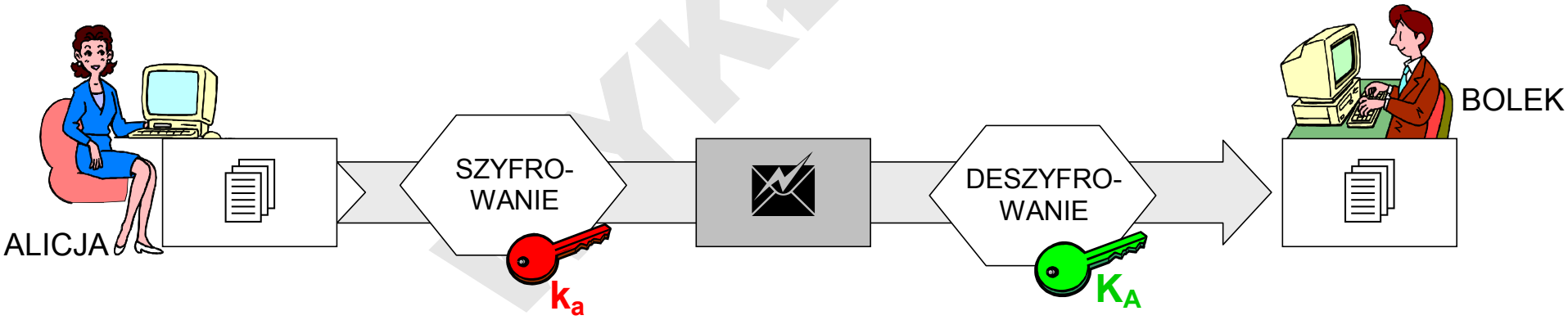
- odbiorca **B** posiada parę kluczy: prywatny klucz k_b oraz publiczny klucz K_B
- $E_{K_B}[M] = S \rightarrow S \rightarrow D_{k_b}[S] = M$
- znajomość klucza publicznego K_B nie wystarcza do naruszenia poufności szyfrogramu uzyskanego przy zastosowaniu tego klucza



Szyfrowanie asymetryczne

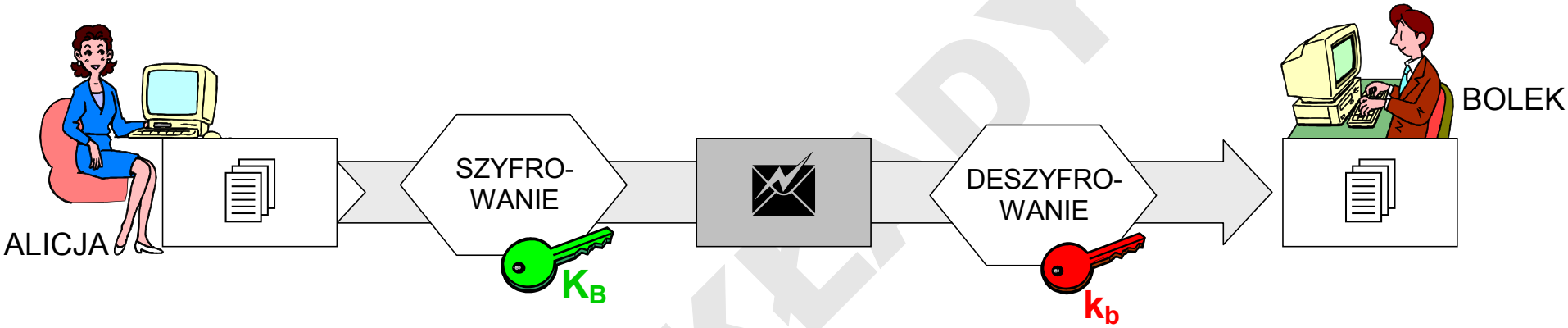
Cechy:

- **przemienność kluczy:**
  - $D_k[E_K[M]] = M$
  - $D_K[E_k[M]] = M$

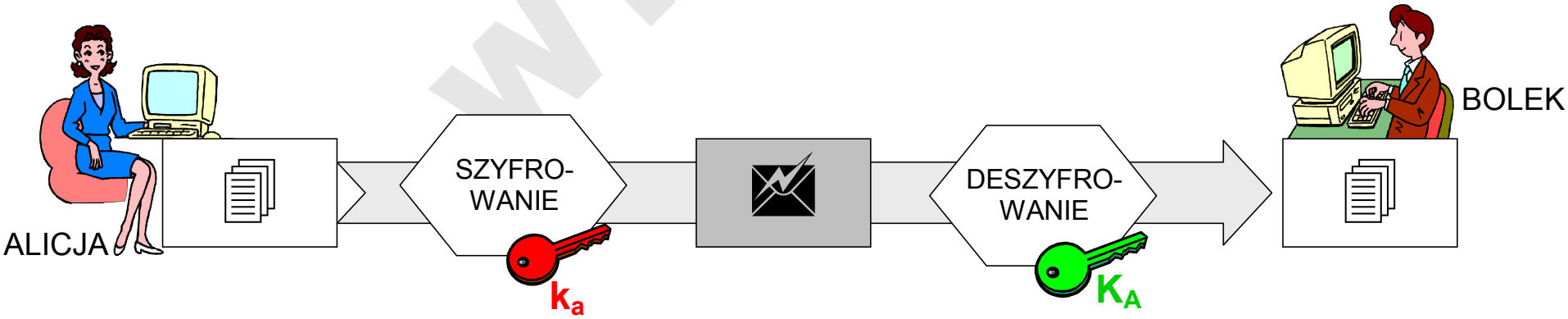


Szyfrowanie asymetryczne

zapewnienie poufności:



zapewnienie autentyczności:



RSA

Algorytm RSA (Rivest–Shamir–Adleman)

- opublikowany w 1978 roku przez Ronalda Rivesta, Adi Shamira, Leonarda Adlemana – niedawno wygasła ochrona patentowa
- pozwala dowolnie ustalić długość klucza
- wymaga użycia 2 dużych liczb pierwszych (przez duże rozumiemy liczby co najmniej stucyfrowe w systemie dziesiętnym)
- do szyfrowania i deszyfrowania wykorzystuje operacje potęgowania dyskretnego
- wymaga dużej liczby działań arytmetycznych
  (jest zdecydowanie wolniejszy od DES – nawet do 1000 razy)

RSA

Dobór kluczy

p, q – losowo wybrane duże liczby pierwsze

n = p·q – moduł

e – liczba względnie pierwsza z (p-1)(q-1)

d – liczba wyznaczona tak, że zachodzi $(e \cdot d) \; mod \; (p-1)(q-1) = 1$

$$d = e^{-1} \; mod \; (p-1)(q-1)$$

$k_a \Rightarrow n, d$

$K_A \Rightarrow n, e$

$$E_{K_A}[M] = M^e \; mod \; n = S$$

$$D_{k_a}[S] = S^d \; mod \; n = M^{ed} \; mod \; n = M^1$$

ElGamal (ELG)

Algorytm ElGamala

- opublikowany w 1985 roku – ale nie jest chroniony patentem
- wykorzystuje koncepcję (i patent) Diffiego-Helmana
- brak ograniczeń eksportowych USA – patent Diffiego-Helmana wygasł w 1997 r.
- szyfrowanie wymaga każdorazowo losowo wybranej wartość *k*
- dlatego też ten sam tekst jawny każdorazowo daje inny szyfrogram
- niestety szyfrogram jest dwukrotnie dłuższy od tekstu jawnego

ElGamal (ELG)

Generowanie kluczy

- wybieramy losowo liczbę pierwszą p
- wykorzystujemy multiplikatywną grupę modulo p – $\mathbb{Z}_p^*$
- gdzie p jest liczbą pierwszą, a $\mathbb{Z}_p$ jego ciałem skończonym (GF(p) lub $\mathbb{Z}/p\mathbb{Z}$)
- wybieramy liczbę g, która jest elementem pierwotnym (generatorem) grupy $\mathbb{Z}_p^*$
- generator – generuje ciąg 1, g, g^2, g^3, ...
- z którego tylko skończenie wiele należy do $\mathbb{Z}_p^*$ (potem zaczną się powtarzać modulo p)
- w ogólności mamy q elementów: 1, g, g^2, ... , g^{q-1} (g^q mod p = 1)
- istnieje przynajmniej jedno g generujące całą grupę! (tzn. $q = p$-1)
- czyli zamiast 1, ..., p-1 możemy grupę traktować jako 1, g, g^2, ... , g^{p-2}

ElGamal (ELG)

Generowanie kluczy

- wybieramy losowo liczbę $x < p$
- obliczamy $y = g^x \bmod p$
- klucz publiczny stanowią y, g i p – zarówno g, jak i p mogą być wspólnie wykorzystywane przez grupę użytkowników (*mod* p)
- kluczem prywatnym jest x

ElGamal (ELG)

Szyfrowanie

- wybieramy losowo liczbę k względnie pierwszą z p
- obliczamy $a = g^k \bmod p$
- obliczamy $b = y^k \cdot \mathrm{M} \bmod p$
- szyfrogram to para (a,b)

Deszyfrowanie

- $\mathrm{M} = b/a^x \bmod p$
- ponieważ $a^x \equiv g^{kx} \bmod p$
- $b/a^x \equiv y^k \cdot \mathrm{M}/a^x \equiv g^{xk} \cdot \mathrm{M}/g^{kx} \equiv \mathrm{M} \bmod p$

Wybrane inne tryby szyfrowania

Stare, rzadko stosowane

- CBCC, OFBNLF, PFB, CBCPD, ...

Nowe, mało znane (i mało analizowane)

- OCB
- XCBC
- IACBC

chronione świeżymi patentami **!**

- CCM

Wybrane inne szyfry

Jawne

- Serpent – bardzo wysokie bezpieczeństwo, bardzo duży koszt obliczeniowy
- Twofish – bezpieczeństwo wyższe niż AES, złożone obliczeniowo operowanie kluczem
- Arcfour, szyfr Rabina,...

Standardy

- GOST – standard rosyjski (GOST 28147-89)
- CAST – standard Northern Telecom (Kanada)

Niejawne

- Skipjack (NSA 1990, sprzętowe układy Clipper, Capstone zabezpieczone przed demontażem)