

\\
.
001.
^
u\$ON=1
z00BAI
I...=..
;s<...
NRX~=-
z0c^<X^
~B0s~^
00\$H~
n\$0=XN;..
iBBB0vU1=~..
`\$000cRr`vul
FAHZuqr-
ZZUFA0FI..
;BRHv n\$U^
`ARN1 ^0si
'Onv~ 01..
c0qr rs..
aUU` ul..
`RO- :..
nn~` -=.~|-
=1^'...` :..

Podstawowe elementy kryptografii

Zagadnienia

1. Terminologia

część I

2. Szyfry symetryczne

3. Szyfry asymetryczne

4. Funkcje skrótu i podpis cyfrowy

5. Praktyczne problemy kryptografii

6. Zastosowania kryptografii

7. Prawne aspekty wykorzystania kryptografii

8. Steganografia

Krótką historia kryptografii

κρυπτός (kryptós) – ukryte
γράφειν (gráphein) – pismo

Kamienie milowe postępu kryptograficznego:

- ok. 1900 p.n.e. – pierwsze znane przykłady przekształceń kryptograficznych (w inskrypcjach hieroglificznych)
- ok. 475 p.n.e. – w Sparcie pierwsze zastosowanie szyfrowania w celach łączności
- ok. 60 p.n.e. – szyfr Cezara
- 1412 – egipski uczoney Kalkashandi pisze pierwszy znany traktat o kryptografii
- 1553 – szyfr polialfabetyczny Vigenère'a
- 1883 – zasada Kerckhoffs'a
- 1917 – koncepcja szyfru jednorazowego OTP = One Time Pad (Gilbert Vernam)
- 1971 – komercyjny system szyfrowania Lucifer dla IBM 3514 (Horst Faistel, IBM)
- 1975 – algorytm szyfrowania symetrycznego DES (Horst Faistel, IBM)
- 1976 – koncepcja szyfrowania asymetrycznego (Diffie i Hellman; Stanford Univ.)
- 1978 – algorytm RSA (Rivest, Shamir, Adelman; MIT)

Podstawowe pojęcia

Kryptologia – nauka 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); $\Delta = 1, \dots, 26$

$\Delta = 3$ x $f(x)$

A	D
B	E
C	F
...	
W	Z
X	A
Y	B
Z	C

S	U	S
E	G	E
K	M	K
R	T	R
E	G	E
T	W	T

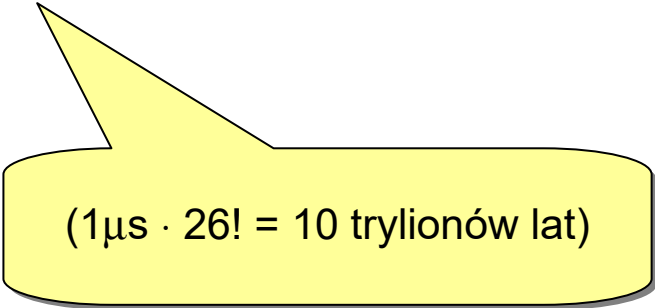
Proste szyfry

Szyfrowanie metodą podstawiania

Szyfry monoalfabetyczne

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

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



$(1\mu s \cdot 26! = 10 \text{ trylionów lat})$

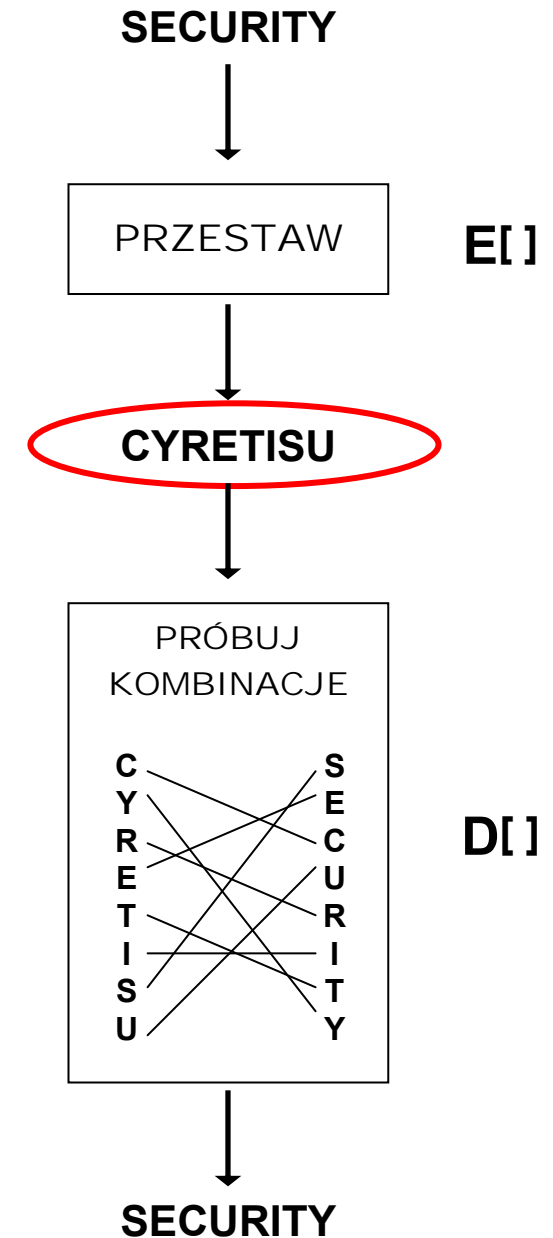
Proste szyfry

Szyfrowanie metodą przestawiania

Przestawianie losowe:

Przestawianie z figurą geometryczną

- figura geometryczna definiuje transpozycję wiadomości czytelnej

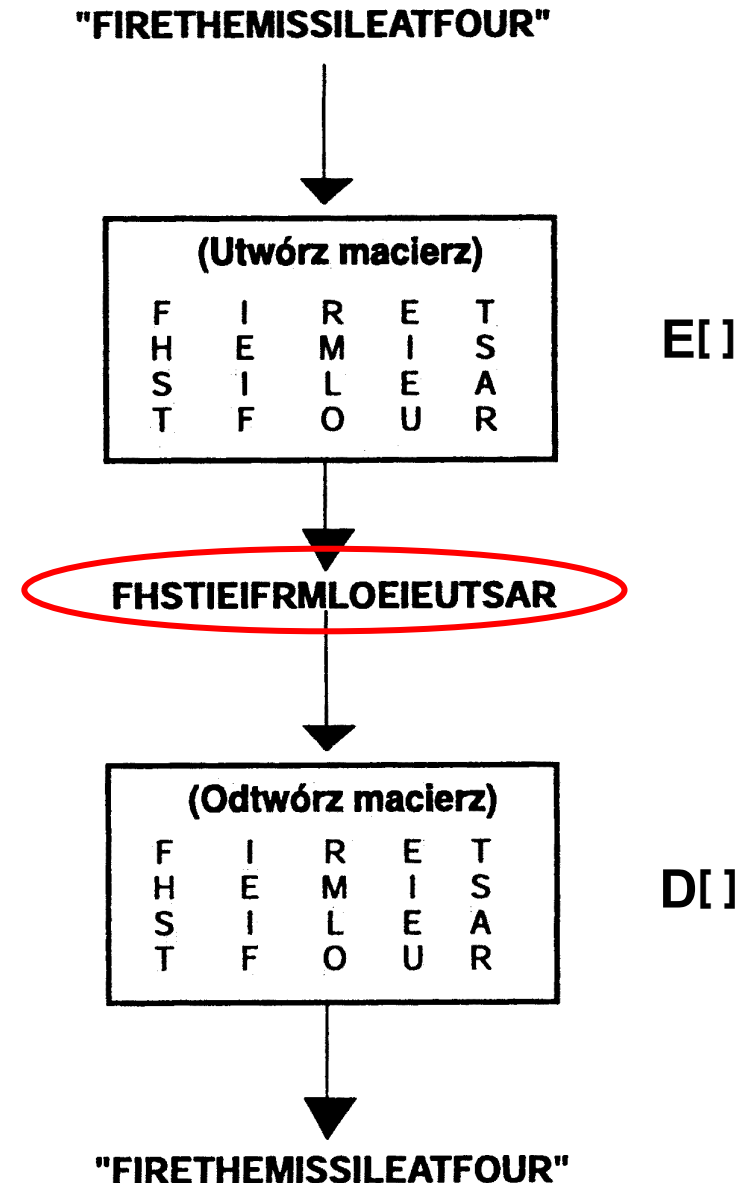


Proste szyfry

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. $k = (5,4;2-5-3-1-4)$
- jaki będzie szyfrogram z poprzedniego przykładu dla takiego klucza?

Inne proste metody przestawiania:

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

Współczesna kryptografia

Zasada Kerckhoffsza

Algorytm szyfrowania i deszyfrowania jest jawny

- siła systemu kryptograficznego nie powinna polegać na tajności algorytmu
- lecz wyłącznie na tajności pewnego zmiennego parametru tego algorytmu (klucza)

= ~~“security by obscurity”~~

Współczesna kryptografia

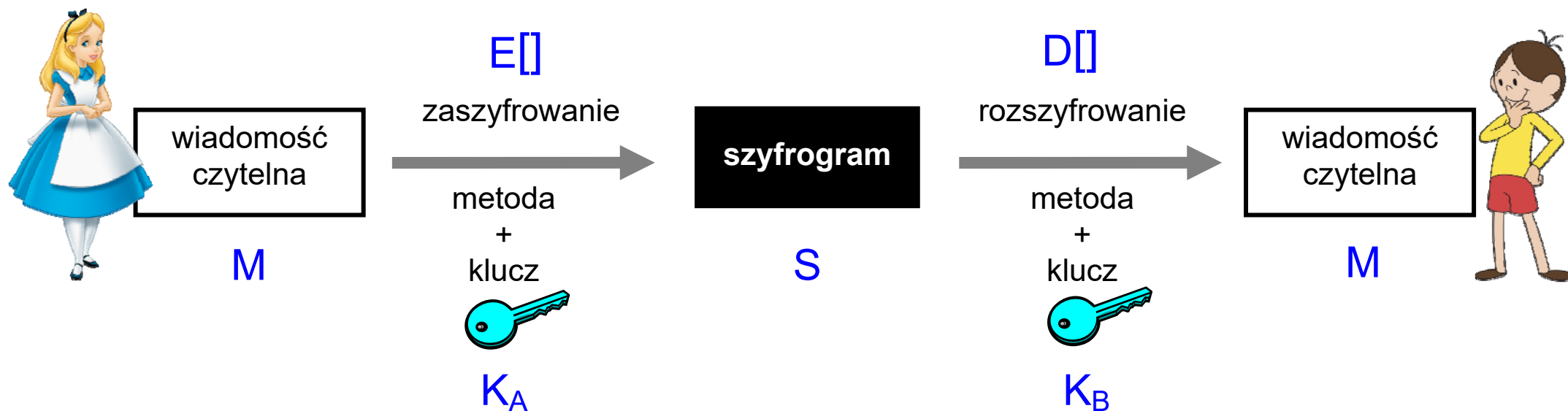
"Anyone who creates his or her own cryptographic primitives is either a genius or a fool. Given the genius/fool ratio for our species, the odds aren't very good."

Bruce Schneier



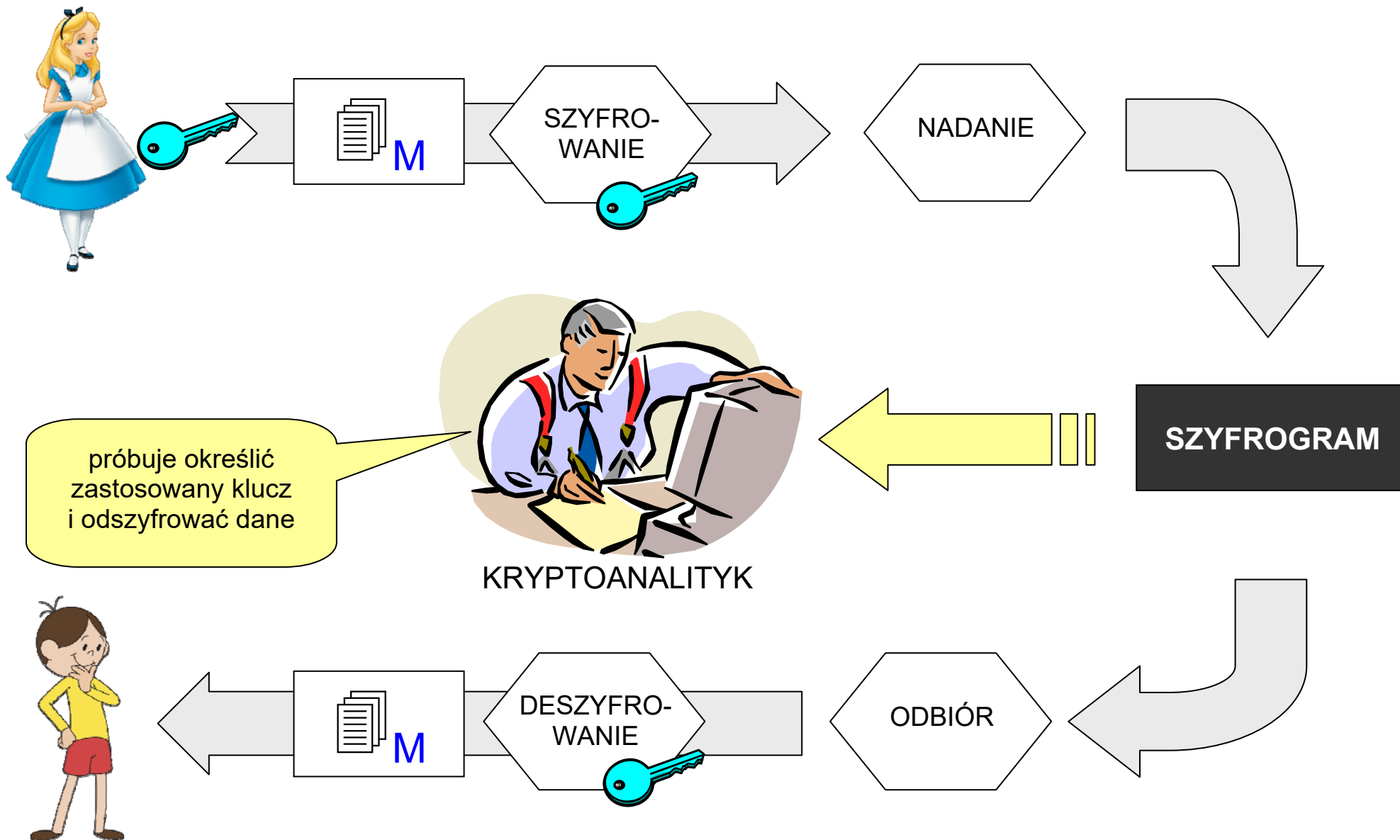
Szyfrowanie z kluczem

Schemat ogólny



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

Kryptoanaliza



Kryptoanaliza

Przykładowe ataki kryptoanalityczne:

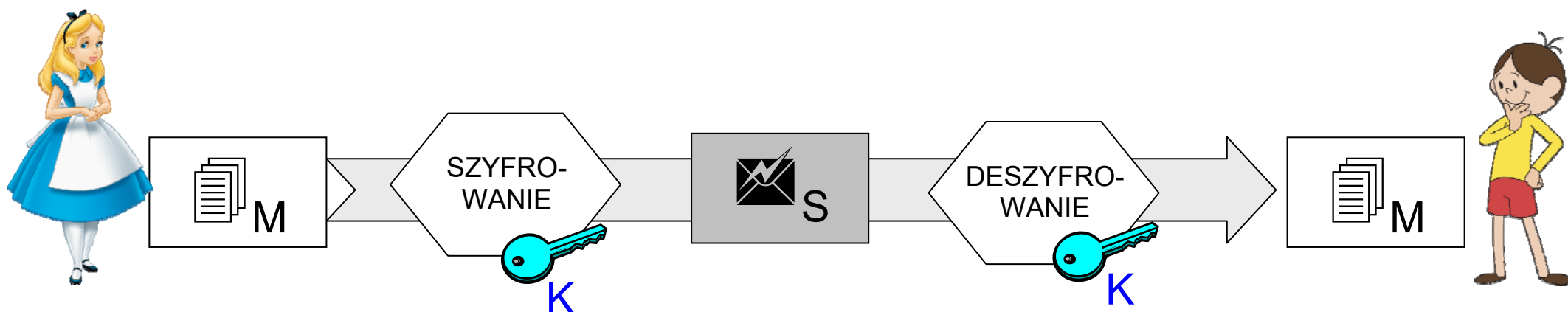
- **przeszukiwanie wyczerpujące** dziedziny kluczy – *brute force*
- **atak znanym tekstem jawnym** – kryptoanalityk ma parę (lub wiele par) tekstu zaszyfrowanego i jawnego (tzw. ściągą, ang. *crib*)
- **atak wybranym tekstem jawnym** – atak aktywny: kryptoanalityk zmusza kryptosystem (nadawcę) do zaszyfrowania własnego tekstu
- **black-box / white-box analysis** → white-box cryptography (WBC)
- ataki **SCA (Side-Channel Attacks)** na sprzętowe układy kryptograficzne

Kryptoanaliza „praktyczna”:

- zdobycie klucza wszelkimi koniecznymi środkami, np. „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:

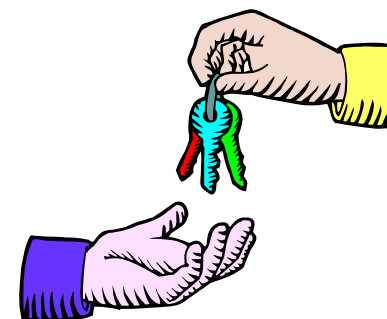
problem tajności klucza

- 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 + niezaprzeczalność

- czy tajność klucza zapewnia niezaprzeczalność?

Szyfrowanie symetryczne



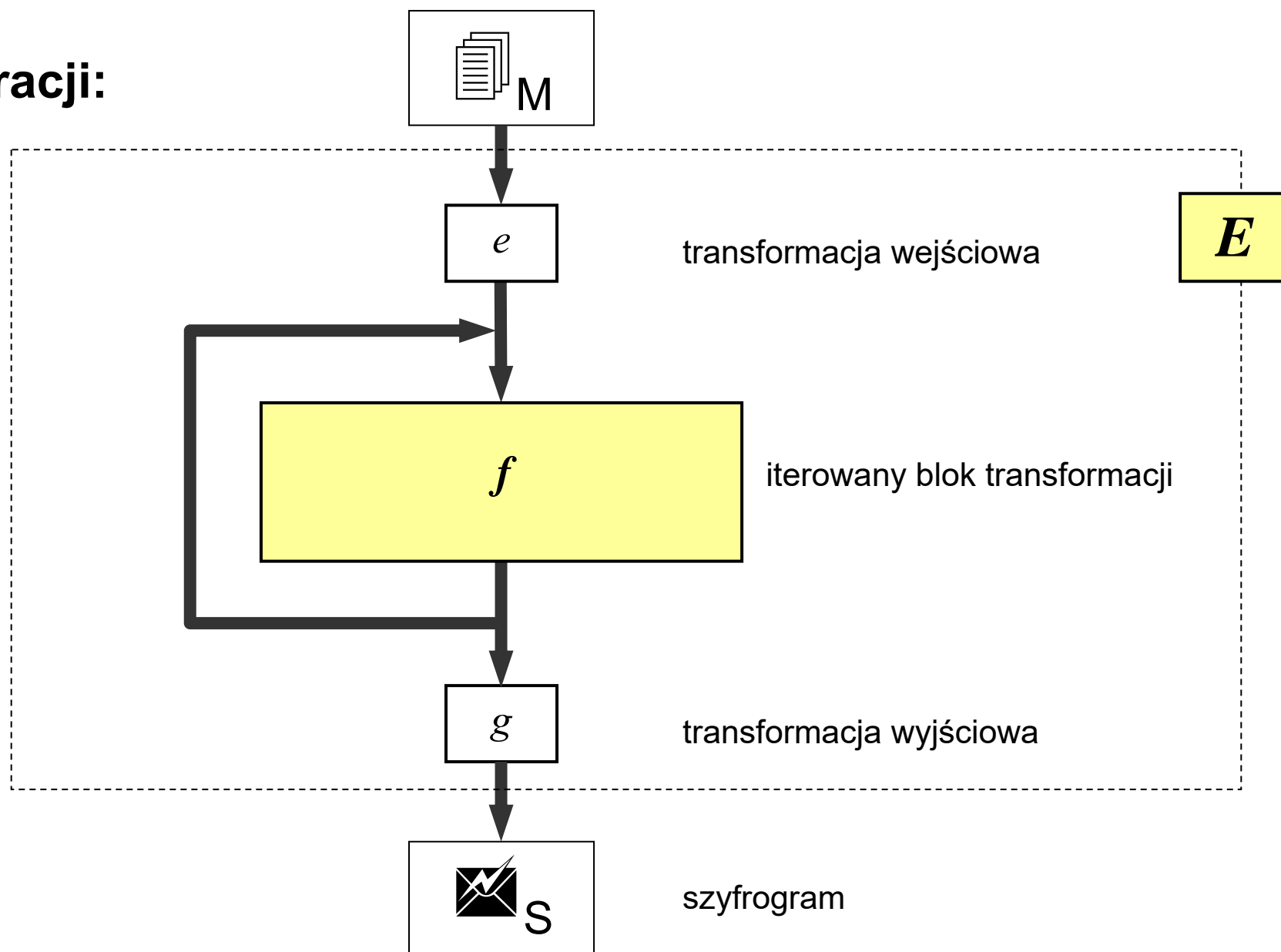
- 2 os. = 1 kl.; 3 os. = 3 kl.; 4 os. = 6 kl.; 10 os. = 45 kl.; 100 os. = 4950 kl.; ...

autentyczność + niezaprzeczalność

- czy tajność klucza zapewnia niezaprzeczalność?

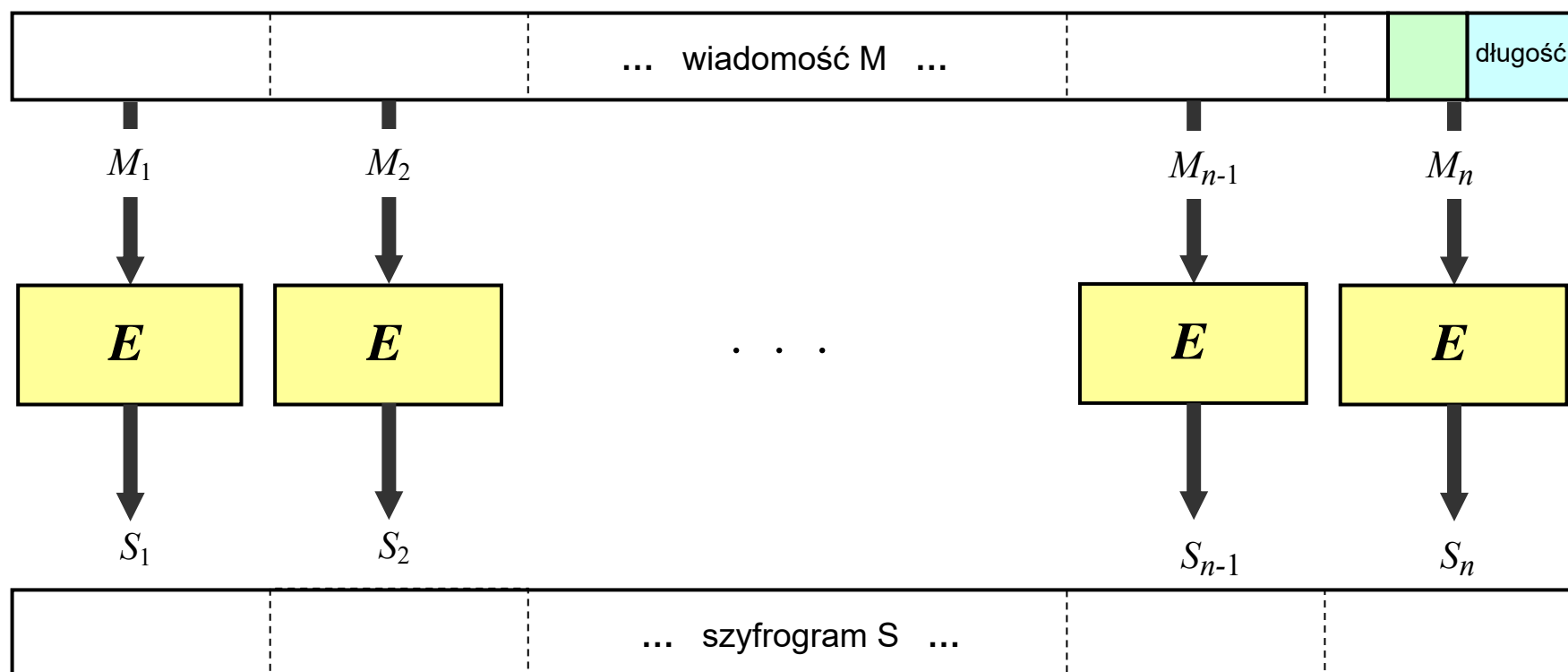
Algorytmy

Siatka operacji:



Algorytmy

Szyfry blokowe:



DES

Algorytm DES (*Data Encryption Standard*)

- opracowany w 1975 przez IBM na zamówienie NSA (*National Security Agency*)
- przyjęty w 1976 przez NBS (*National Bureau of Standards*, obecnie NIST = *National Institute of Standards and Technology*)
- 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

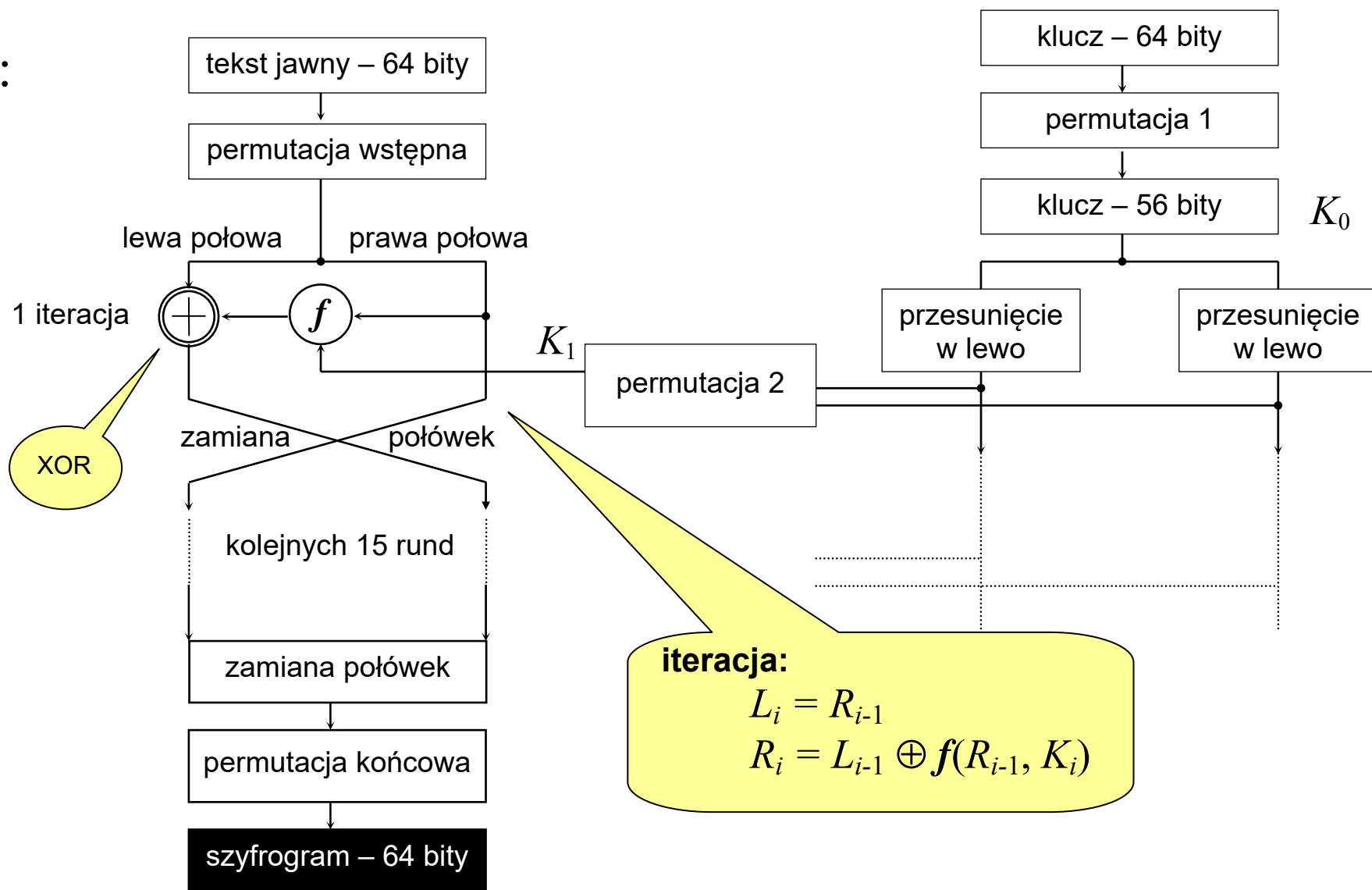
DES

Fazy działania (siatka 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

E:



Siatka Feistela

DES

Fazy

- każda kolejna runda, dokonuje obliczeń na wynikach poprzedniej rundy i podkluczu K_i generowanym z klucza K_0

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
- każdy z 8 S-bloków jest inny (→ [Schneier], [Stallings])

DES

Deszyfrowanie

- ta sama sieć operacji
- klucze stosowane są w kolejności odwrotnej od K_{16} do K_1
- kryptoanaliza: złożoność obliczeniowa procesu dopasowania kolejnych możliwych wartości klucza (w latach '80 setki/tysiące lat) uczyniła DES odpornym na atak metodą przeszukiwania wyczerpującego

DES

Tryby pracy

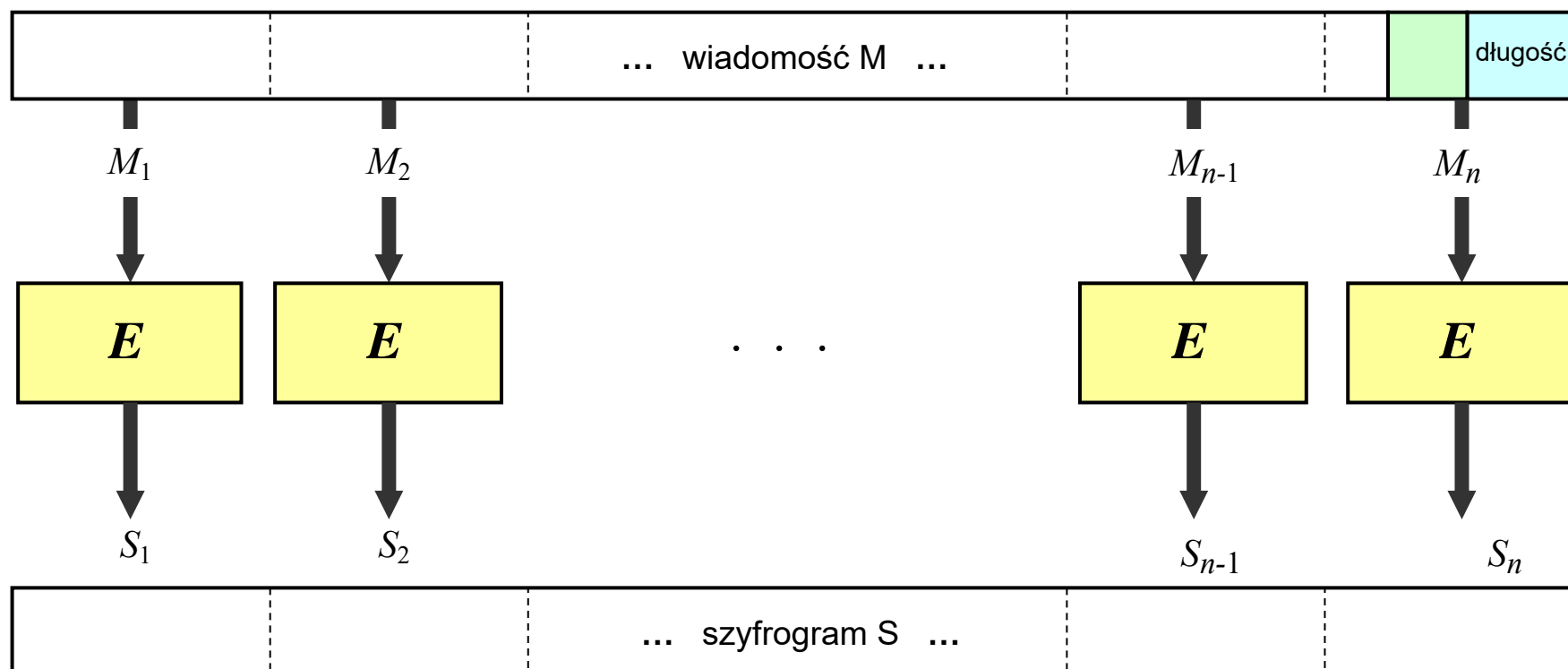
ECB (*Electronic Code Book*) – podstawowy tryb szyfrowania blokowego

- 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

ECB (*Electronic Code Book*) – podstawowy tryb szyfrowania blokowego



DES

Tryby pracy

ECB (*Electronic Code Book*) – podstawowy tryb szyfrowania blokowego



Patterns within plaintext encrypted using an ECB cipher may be visible within the ciphertext.

(źródło: "The Web Application Hacker's Handbook")

DES

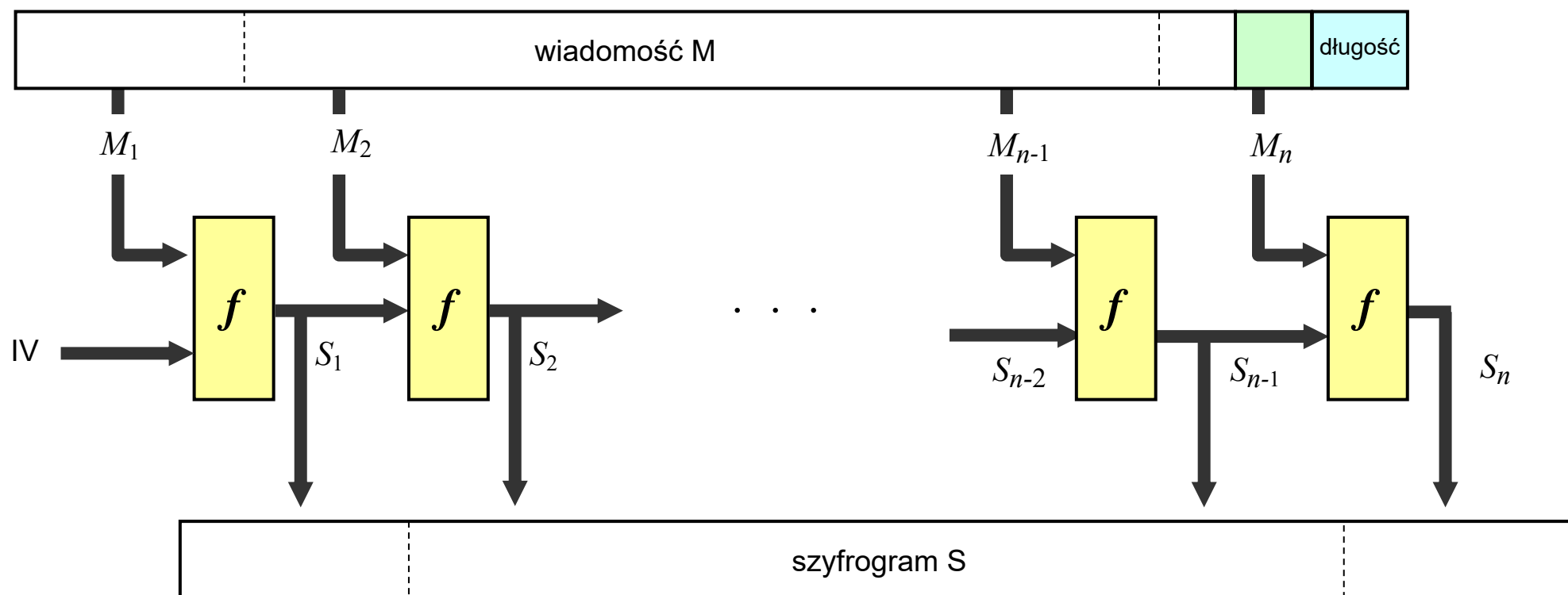
Tryby pracy

CBC (*Cipher Block Chaining*) – tryby blokowy sprzężenia zwrotnego

- na pierwszym 64b bloku jest wykonywana operacja XOR z pewnym wektorem początkowym (IV = *Initialization Vector*) znanym nadawcy i odbiorcy
 $M_1 \oplus IV$
- wynikowy ciąg jest podawany na wejście algorytmu DES
 $S_1 = E_K [M_1 \oplus IV]$
- 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

Sprzężenie zwrotne:



DES

Tryby pracy

CFB (*Cipher FeedBack*) i **OFB** (*Output FeedBack*) – tryby strumieniowe

- tryby szyfrowania strumieniowego – szyfruje każdorazowo po jednym znaku 8b
- szyfrogram jest tej samej długości co tekst jawny
- przydatne wobec danych wprowadzanych asynchronicznie– jak znaki z klawiatury
- w **CFB** na wejście funkcji szyfrującej podawana jest zawartość 64b rejestru przesuwanego – 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*) – tryby strumieniowe

- 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

DES

Tryby pracy

Wnioski

- ECB jest trywialny
 - nie powinien być stosowany do szyfrowania danych
 - ale mógłby być wykorzystany do przesłania kluczy oraz IV

w istocie jednak, żaden z przedstawionych powyżej trybów nie powinien być współcześnie wykorzystywany w kryptosystemie ogólnego przeznaczenia – żaden bowiem nie jest szyfrowaniem uwierzytelnionym (→ następny wykład)

DES

Odporność

- w 1998 r. DES z kluczem 56b został złamany w 56 godzin kryptoanalizy metodą przeszukiwania wyczerpującego
- EFF DES Cracker (Electric Frontier Foundation) – prawie 10^{11} kluczy/sek.
– koszt sprzętu (wówczas) 250 tys. USD
- rok później zajęło to już 22 godziny
(Deep Crack – system rozproszony: 10 tys. stanowisk w Internecie)
- dziś to kwestia minut

Długość klucza

$$T = \frac{2^K}{P}, \text{ czyli } K = \log_2(P \cdot T)$$

mając do dyspozycji taką moc (P)

można w takim czasie (T)

złamać taki klucz (K)

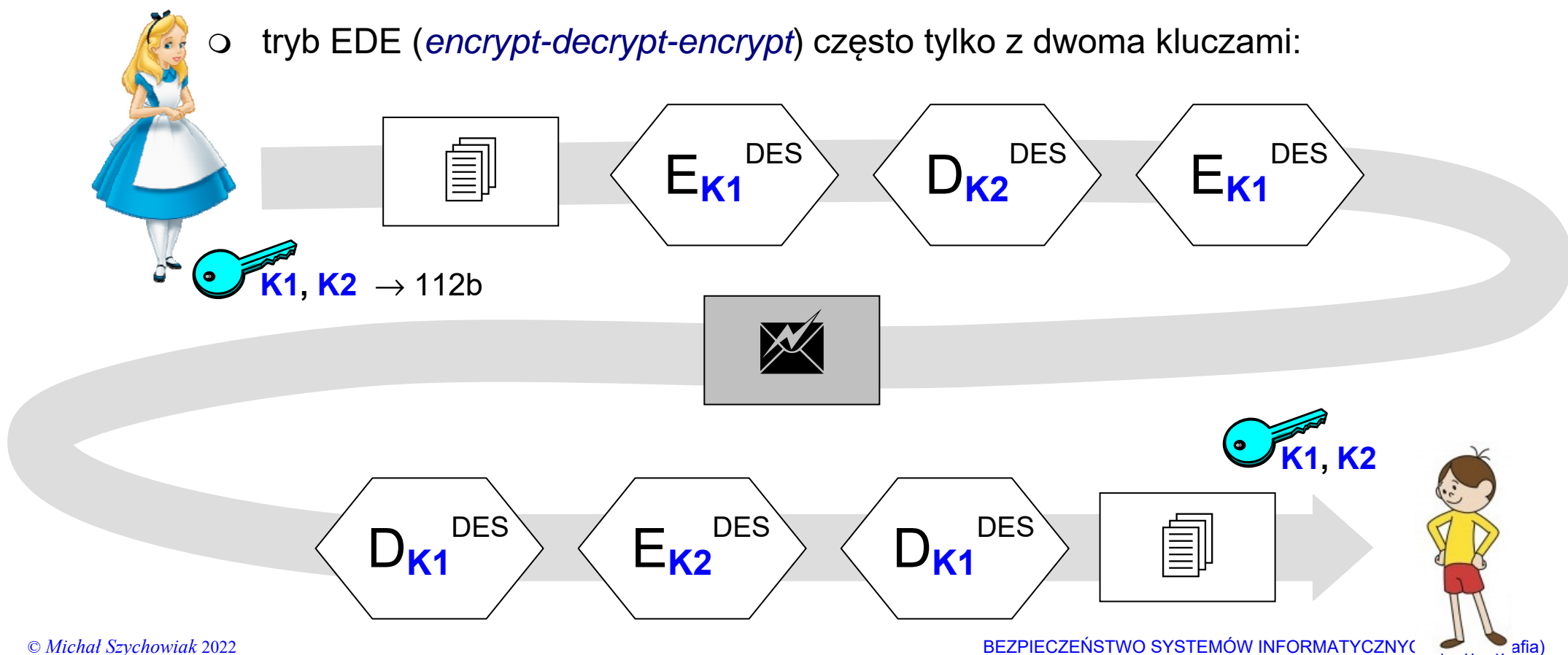
	1 sek.	1 godz.	1 dzień	1 tydzień	1 miesiąc	1 rok
10^{11}	37 b	48 b	53 b	56 b	58 b	61 b
10^{12}	40 b	52 b	56 b	59 b	61 b	65 b
10^{13}	43 b	55 b	60 b	62 b	64 b	68 b
...						
10^{20}	66 b	78 b	83 b	86 b	88 b	91 b

Cryptographic Key Length Recommendation → <https://www.keylength.com>

3DES

Algorytm 3DES (*Triple DES*)

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



3DES

Algorytm 3DES (*Triple DES*)

- EDE: dlaczego w środku jest D, nie E?

$$S = E_{K1}[D_{K2}[E_{K1}[M]]]$$

$$M = D_{K1}[E_{K2}[D_{K1}[S]]]$$

3DES

Algorytm 3DES (*Triple DES*)

- EDE: dlaczego w środku jest D, nie E?

$$S = E_{K1}[D_{K2}[E_{K1}[M]]]$$

$$M = D_{K1}[E_{K2}[D_{K1}[S]]]$$

- dzięki temu 3DES może odszyfrować S uzyskany „pojedynczym” DES-em:

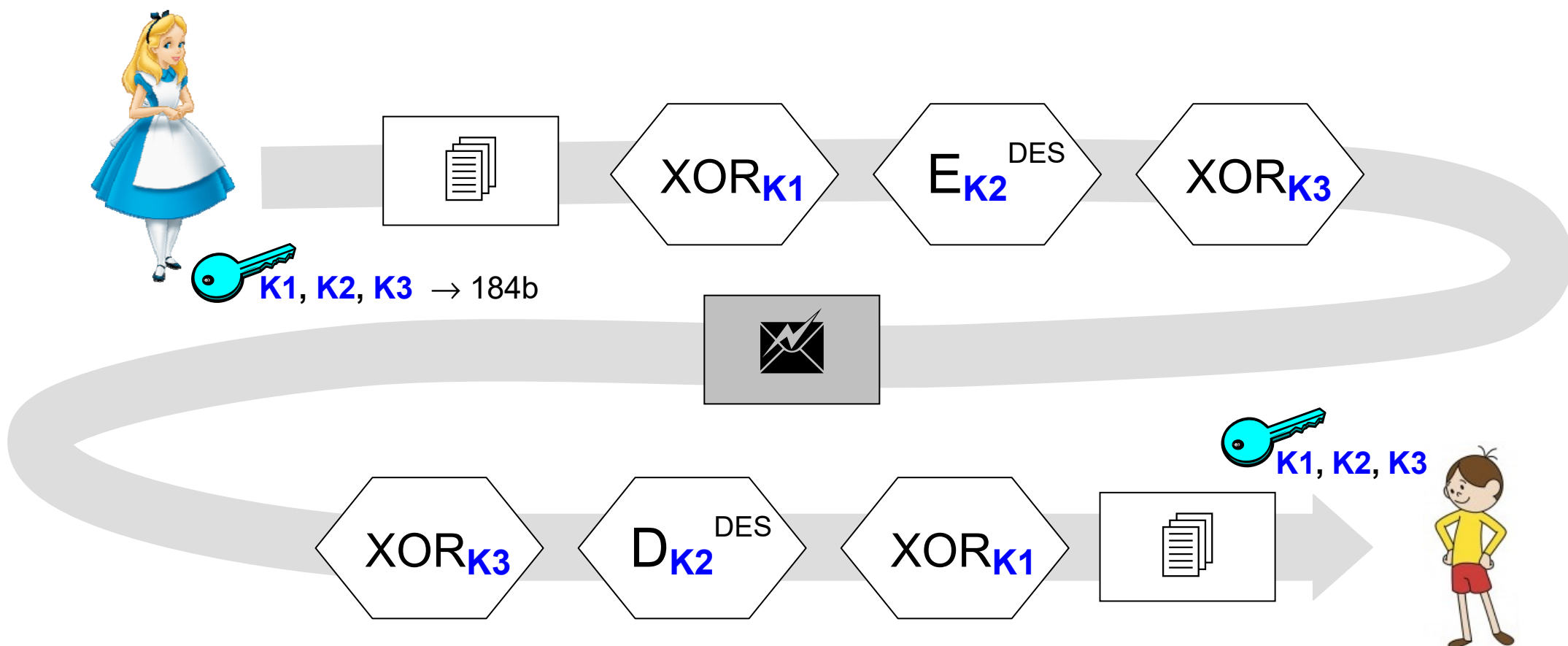
$$S = E_{K1}[M]$$

$$M = D_{K1}[E_{K1}[D_{K1}[S]]], \quad K1=K2$$

DESX

Algorytm DESX (*DES XOR-ed*)

- tryb XEX: 3 klucze (64b + 56b + 64b) → klucz 184b



DES-768

Algorytm DES-768

- zmiana sposobu generowania 16 podkluczy w iteracjach
- $16 \text{ podkluczy} \cdot 48b \rightarrow \text{klucz } 768b$

RC2 / RC4 / RC5 / RC6

- prawnie zastrzeżone algorytmy opracowane przez Rona Rivesta (RSA Data Security)
 - chociaż od 1994 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 m.in. w SSL, MPPE, WEP, WPA, BitTorrent
- problemy z generowaniem kluczy (słabą losowością) odkryte w 2001 r. wykluczyły RC4 z dalszego użycia (w lutym 2015 RFC 7465 zakazał stosowania w TLS)

Blowfish

(łac. *Arothron hispidus*)



- opracowany w 1993 przez Bruce'a Schneiera
 - blok danych 64b, klucz podstawowy o długości od 32b do 448b
 - 16 iteracji wykorzystujących 18 kluczy pomocniczych (wyznaczanych każdorazowo przed szyfrowaniem i deszyfrowaniem) 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
- Twofish (1998), Treefish (2008)

IDEA

IDEA (*International Data Encryption Algorithm*)

- szybki szyfr opracowany w 1991r. przez Swiss Federal Institute of Technology (James L. Massey i Xuejia Lai)
- do 2011 r. chroniony patentem, ale dostępny bezpłatnie dla celów niekomercyjnych
- 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 1 w IDEA iteracja odpowiada 2 w DES)

Rijndael

Algorytm Rijndael

- opracowany w 1999 przez Belgów: Vincenta Rijmena i Joana Daemena
- bloki po 128b, 196b lub 256b
- klucze również 128b, 196b lub 256b
- 10 (128b), 12 (196b) lub 14 (256b) iteracji

AES

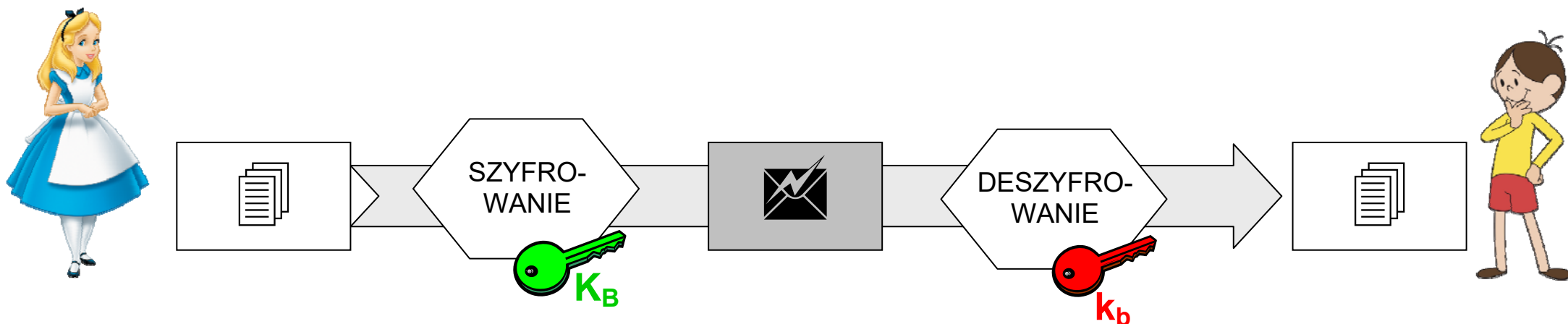
AES (*Advanced Encryption Standard*)

- następca DES-a od 2001 r.
- wykorzystuje algorytm Rijndael (wygrał rywalizację z alg. Serpent, Twofish, RC6,...)
- klucze 128b, 192b, 256b
- tryb blokowy (16B) i strumieniowy
- tryby szyfrowania znacznie utrudniające kryptoanalizę (np. XEX, XTS)
- strumieniowy tryb licznikowy (CTR/SIC/CM = *Counter Mode*) rejestr jest inkrementowany wraz z kolejnymi operacjami szyfrowania bloków danych – tryb ten oferuje możliwość zrównoleglenia operacji na różnych blokach danych, ale jego bezpieczeństwo jest dyskusyjne

V · T · EBlock ciphers (security summary)		
Common algorithms	AES · Blowfish · DES (internal mechanics, Triple DES) · Serpent · Twofish	
Less common algorithms	Camellia · CAST-128 · GOST · IDEA · RC2 · RC5 · RC6 · SEED · ARIA · Skipjack · TEA · XTEA	
Other algorithms	3-Way · Akelarre · Anubis · BaseKing · BassOmatic · BATON · BEAR and LION · CAST-256 · Chiasmus · CICS-1 · CIPHERUNICORN-A · CIPHERUNICORN-E · CLEFIA · CMEA · Cobra · COCONUT98 · Crab · Cryptomeria/C2 · CRYPTON · CS-Cipher · DEAL · DES-X · DFC · E2 · FEAL · FEA-M · FROG · G-DES · Grand Cru · Hasty Pudding cipher · Hierocrypt · ICE · IDEA NXT · Intel Cascade Cipher · Iraqi · Kalyna · KASUMI · KeeLoq · KHAZAD · Khufu and Khafre · KN-Cipher · Kuznyechik · Ladder-DES · Libelle · LOKI (97, 89/91) · Lucifer · M6 · M8 · MacGuffin · Madryga · MAGENTA · MARS · Mercy · MESH · MISTY1 · MMB · MULTI2 · MultiSwap · New Data Seal · NewDES · Nimbus · NOEKEON · NUSH · PRESENT · Prince · Q · RC6 · REDOC · Red Pike · S-1 · SAFER · SAVILLE · SC2000 · SHACAL · SHARK · Simon · SM4 · Speck · Spectr-H64 · Square · SXAL/MBAL · Threefish · Treyfer · UES · xmx · XXTEA · Zodiac	
Design	Feistel network · Key schedule · Lai–Massey scheme · Product cipher · S-box · P-box · SPN · Confusion and diffusion · Avalanche effect · Block size · Key size · Key whitening (Whitening transformation)	
Attack (cryptanalysis)	Brute-force (EFF DES cracker) · MITM (Biclique attack · 3-subset MITM attack) · Linear (Piling-up lemma) · Differential (Impossible · Truncated · Higher-order) · Differential-linear · Distinguishing (Known-key) · Integral/Square · Boomerang · Mod <i>n</i> · Related-key · Slide · Rotational · Side-channel (Timing · Power-monitoring · Electromagnetic · Acoustic · Differential-fault) · XSL · Interpolation · Partitioning · Rubber-hose · Black-bag · Davies · Rebound · Weak key · Tau · Chi-square · Time/memory/data tradeoff	
Standardization	AES process · CRYPTREC · NESSIE	
Utilization	Initialization vector · Mode of operation · Padding	
V · T · EStream ciphers		
Widely used ciphers	RC4 · block ciphers in stream mode · ChaCha	
eSTREAM Portfolio	Software	HC-256 · Rabbit · Salsa20 · SOSEMANUK
	Hardware	Grain · MICKEY · Trivium
Other ciphers	A5/1 · A5/2 · Achterbahn · E0 · F-FCSR · FISH · ISAAC · MUGI · Panama · Phelix · Pike · Py · QUAD · Scream · SEAL · SNOW · SOBER · SOBER-128 · VEST · VMPC · WAKE	
Theory	shift register · LFSR · NLFSR · shrinking generator · T-function · IV	
Attacks	correlation attack · correlation immunity · stream cipher attacks	

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



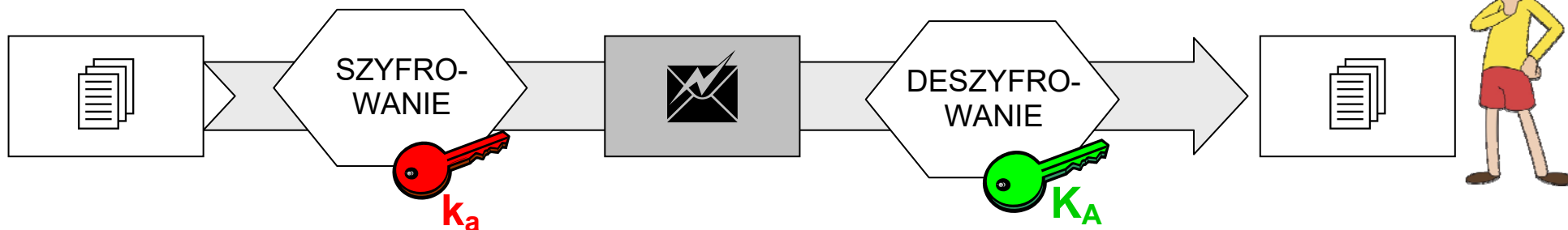
Szyfrowanie asymetryczne

Cechy:

- o przemienność kluczy:

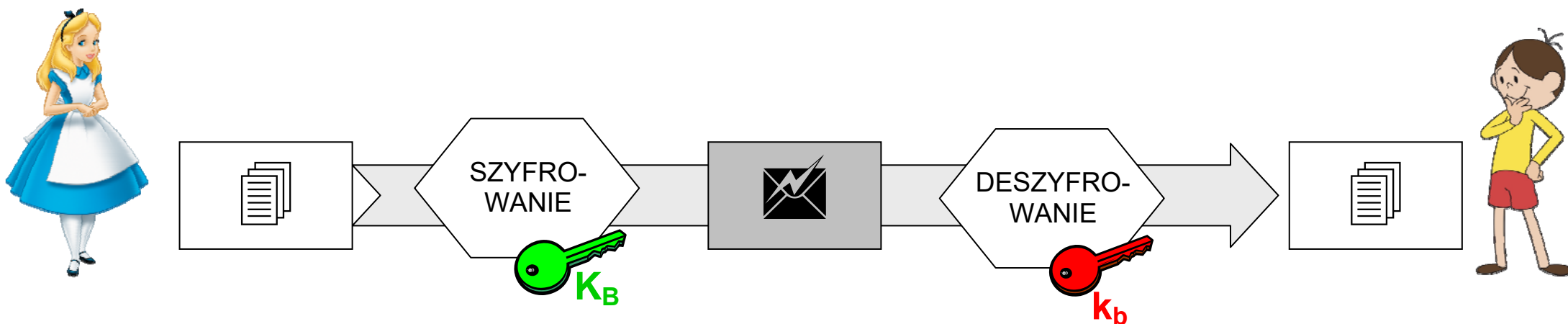
- $D_k[E_K[M]] = M$

- $D_K[E_k[M]] = M$

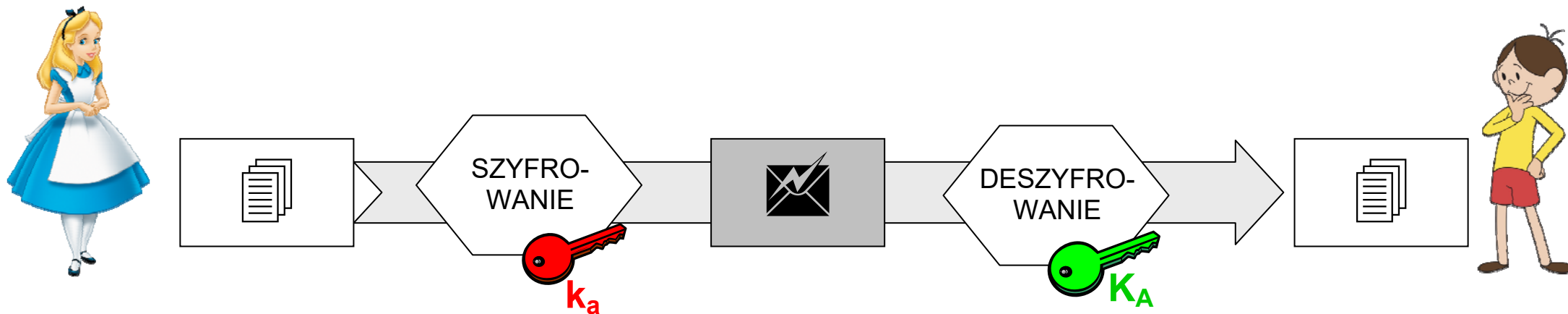


Szyfrowanie asymetryczne

poufność:



autentyczność + niezaprzeczalność:



RSA

Algorytm RSA (Rivest–Shamir–Adleman)

- opublikowany w 1978 roku przez Ronalda Rivesta, Adi Shamira, Leonarda Adlemana – w 2000 r. 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 algorytmów symetrycznych – nawet do 1000 razy)

RSA

Dobór kluczy

p, q – losowo wybrane duże liczby pierwsze

$n = p \cdot q$ – moduł

wartość funkcji Eulera dla n : $\varphi(n) = (p-1)(q-1)$

$\varphi(p) = p-1$ dla liczby pierwszej p

$\varphi(p \cdot q) = \varphi(p) \cdot \varphi(q)$

e – liczba względnie pierwsza z $\varphi(n)$

→ $\text{NajwiększyWspólnyDzielnik}(e, \varphi(n)) = 1$

d – liczba wyznaczona tak, że zachodzi $(e \cdot d) \bmod \varphi(n) = 1$

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

$k_a \Rightarrow n, d$

$K_A \Rightarrow n, e$

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

$$D_{k_a}[S] = S^d \bmod n = M^{ed} \bmod n = (M \cdot 1) \bmod n = M$$

z twierdzenia Eulera:

$$x^{\varphi(n)} \bmod n = 1$$

warunek: $x < n$ czyli: $M < n$

RSA

Przykład

$$k_a \Rightarrow 187, 107$$

$$K_A \Rightarrow 187, 3$$

$$E_{K_A}[5] = 5^3 \bmod 187 = 125$$

$$D_{k_a}[125] = 125^{107} \bmod 187 = ?$$

$$\text{Euler: } 125^4 \bmod 187 = 125^{2 \cdot 2} \bmod 187 = (125^2 \bmod 187)^2 \bmod 187$$

$$D_{k_a}[125] = 125^{64+32+8+2+1} \bmod 187 = [(125^{64} \bmod 187)(125^{32} \bmod 187)$$

$$(125^8 \bmod 187)(125^2 \bmod 187)(125)] \bmod 187 = 5$$

RSA

Łamanie

RSA Factoring Challenge

- w ramach infrastruktury projektu Grid5000 (INRIA) dokonano w 2010 r. udanej faktoryzacji dla klucza 768 b

123018668453011775513049495838496272077285356959533479219732245215172640050726
365751874520219978646938995647494277406384592519255732630345373154826850791702
6122142913461670429214311602221240479274737794080665351419597459856902143413
=
334780716989568987860441698482126908177047949837137685689124313889828837938780
02287614711652531743087737814467999489
.
367460436667995904282446337996279526322791581643430876426760322838157396665112
79233373417143396810270092798736308917

- w sumie zajęło to 2 i pół roku (512 b to dziś kwestia dni!)
- RSA-240 challenge, 2019: 795 b
- RSA-250, 2020: 829 b (<https://phys.org/news/2020-03-cryptographic.html>)

ElGamal (ELG)

Algorytm ElGamala

- opublikowany w 1985 roku
- szyfrowanie wymaga każdorazowo wybranej losowo pewnej wartości 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 ($\text{GF}(p)$ lub $\mathbb{Z}/p\mathbb{Z}$)
- wybieramy liczbę g , która jest elementem pierwotnym (generatorem) grupy \mathbb{Z}_p^*
- generator – generuje ciąg $1, g, g^2, g^3, \dots$
- z którego tylko skończenie wiele należy do \mathbb{Z}_p^* (inne będą się powtarzać modulo p)
- w ogólności mamy q elementów: $1, g, g^2, \dots, g^{q-1}$ ($g^q \bmod p = 1$)
- istnieje przynajmniej jedno g generujące całą grupę! (tzn. $q = p-1$)
- czyli zamiast $1, \dots, p-1$ możemy grupę traktować jako $1, g, g^2, \dots, g^{p-2}$

ElGamal (ELG)

Generowanie kluczy

- wybieramy sobie (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 ($\bmod p$)
- kluczem prywatnym jest x

ElGamal (ELG)

Szyfrowanie

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

Deszyfrowanie

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

ElGamal (ELG)

Przykład

- $p = 37, g = 7$
- wybieramy $x = 6$ i $k = 7$
- dla $M = 26$
- obliczamy $a = g^x \bmod p = 28$
- obliczamy $b = y^k \cdot M \bmod p = 13$
- szyfrogram to para $(28, 13)$

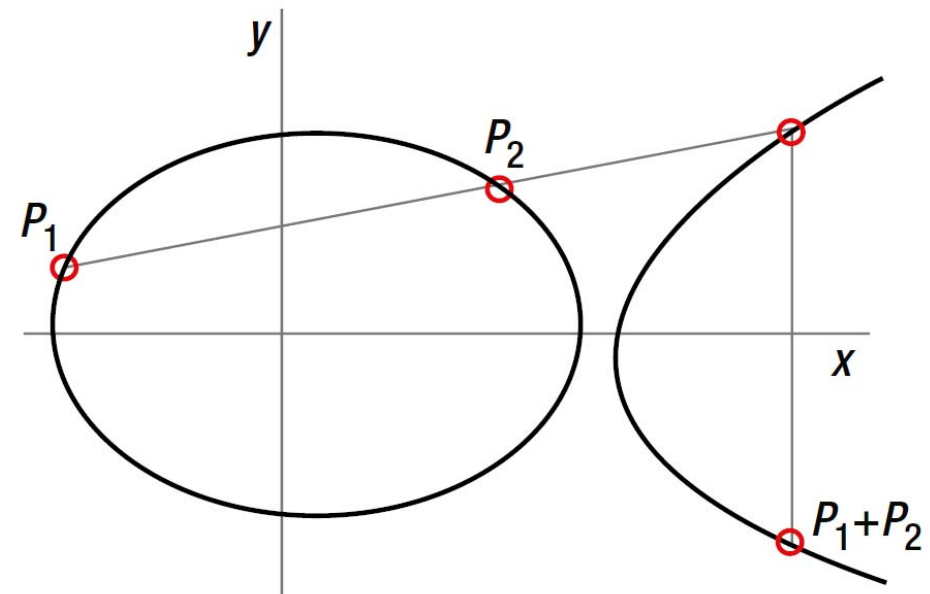
Deszyfrowanie: $M = b/a^x \bmod p$

- $a^x \bmod p = 1/2$
- $M = 13/1/2 = 26$

Długość klucza

Szyfry ECC (*Elliptic Curve Cryptography*)

długość bitowa	
algorytmy klasyczne	ECC
1024	138
1149	147
1369	160
...	...
3072	256



- Curve25519
- Curve448
- Curve41417

<https://arstechnica.com/information-technology/2013/10/a-relatively-easy-to-understand-primer-on-elliptic-curve-cryptography/>