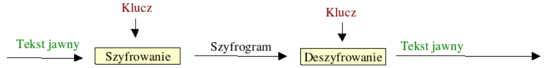
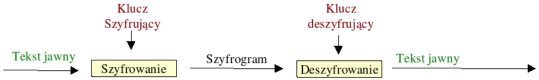
# Outline

Kryptografia Asymetryczna
- Szyfrowanie z kluczem jawnym
- Algorytm plecakowy
- Algorytm RSA
- Wymiana Kluczy Diffiego-Hellmana

# Podstawy




$E_K(M) = C, D_K(C) = M$ , ALGORYTMY SYMETRYCZNE




$E_{K1}(M) = C, D_{K2}(C) = M$ , ALGORYTMY ASYMETRYCZNE

# Zasady systemów szyfrowania z kluczem jawnym

- Problemy z szyfrowaniem konwencjonalnym - wymiana kluczy (dzielenie go z centrum dystybucji kluczy)
- Potrzebny odpowiednik podpisów stosowanych w dokumentach papierowych

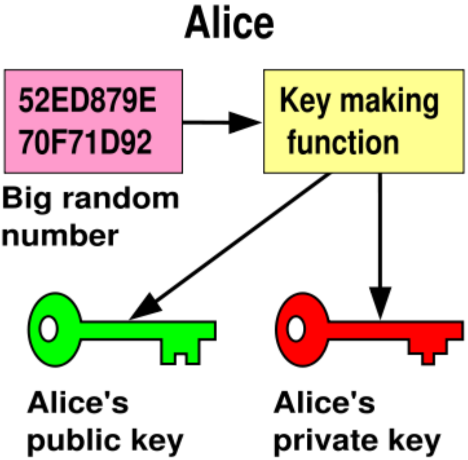
# Zastosowania systemów szyfrowania z kluczem jawnym

- Szyfrowanie/Deszyfrowanie - nadawca szyfruje kominukat za pomocą jawnego klucza
- Sygnatura cyfrowa - nadawca sygnuje komunikat swoim kluczem prywatnym
- Wymiana kluczy - obie strony współpracują przy wymianie klucza sesji

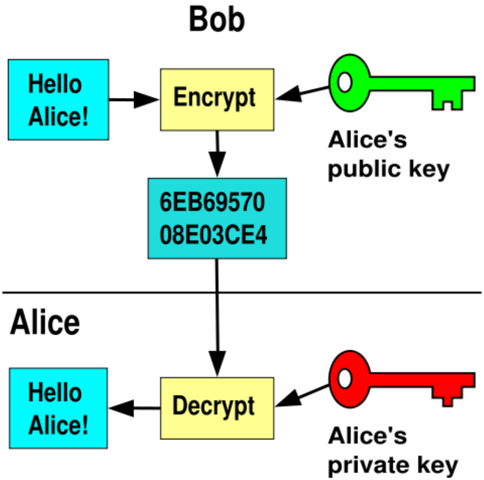
# Uproszczony model szyfrowania z kluczem jawnym

- Każdy system końcowy w sieci generuje parę kluczy do szyfrowania i deszyfrowania komunikatów
- Każdy system publikuje swój klucz szyfrujący przez umieszczenie go w publicznym rejestrze lub pliku. Ten klucz jest kluczem jawnym. Drugi klucz pozostaje prywatny
- Jeżeli A chce wysłać komunikat do B, szyfruje go za pomocą klucza jawnego B.
- Gdy B otrzymuje komunikat, deszyfuje go za pomocą klucza prywatnego B. Żaden inny odbiorca nie może odszyfrować komunikatu, ponieważ tylko B zna swój prywatny klucz

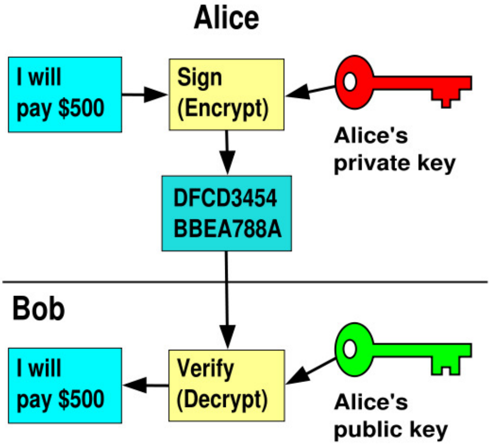
# Generowanie pary kluczy - publiczny(PK) - prywatny(SK)



# Szyfrowanie/Deszyfrowanie



# Sygnatura cyfrowa (Podpis Cyfrowy)



# Bezpieczeństwo systemu

- Dopóki system ma kontrolę nad swoim kluczem prywatnym, dopóty nadchodzące do niego komunikaty są bezpieczne
- W każdej chwili system może zmienić swój klucz prywatny i ogłosić odpowiadający mu nowy klucz jawny, który zastąpi stary.

# Wymagania dotyczące systemów kryptograficznych z kluczem jawnym

1. Dowolna strona B może łatwo na drodze obliczeń wygenerować parę kluczy; klucz jawny -PK, klucz prywatny -SK
2. Nadawca A, znając klucz jawny strony B i wiadomość do zaszyfrowania, może łatwo na drodze obliczeń wygenerować odpowiedni tekst zaszyfrowany - $C = E_{PK_B}(M)$
3. Odbiorca B może łatwo na drodze obliczeń odszyfrować otrzymany tekst zaszyfrowany za pomocą swojego klucza prywatnego i uzyskać tekst jawny - $M = D_{SK_B}(C)$

# Wymagania cz.2

- Dla przeciwnika znającego klucz jawny strony B - $PK_B$ określenie klucza prywatnego $SK_B$ na drodze obliczeń jest niewykonalne
- Dla przeciwnika znającego klucz jawny $PK_B$ i tekst zaszyfrowany C , uzuskanie na drodze obliczeń wiadomości M jest niewykonalne
- Dodatkowy warunek - Funkcje szyfrowania i deszyfrowania mogą być stosowane w dowolnym porządku

# Funkcja jednokierunkowa z bocznym wejściem

- Jest to funkcja której obliczenie w jednym kierunku jest łatwe a niewykonalne w drugim kierunku (funkcja skrótu)
- Istnieje możliwość obliczenia funkcji w drugim kierunku przy znajmości pewnych dodatkowych informacji.
- Funkcja jednokierunkowa z bocznym wejściem to rodzina funkcji odwracalnych
- System szyfrowania z kluczem jawnym polega na opracowaniu odpowiedniej funkcji jednokierunkowej z bocznym wejściem

# Podstawy teorii liczb

Liczby pierwsze:

Liczba całkowita $p < 1$ jest liczbą pierwszą, jeżeli jej jedynymi dzielnikami są $\pm$ 1 i $\pm$ p

Liczby względnie pierwsze:

Liczby a i b są względnie pierwsze, jeżeli nie mają żadnych wspólnych czynników pierwszych, czyli jeżeli ich wspólny dzielnik to tylko 1. Zapis: nwd(a,b) =1. Przykład: 8 i 15, nwd(8,15)=1.

# Operator modulo

(a mod b - reszta z dzielenia a przez b )
Dwie liczby przystają modulo jeżeli:
(a mod n) = (b mon n)
wtedy zapisujemy:
a $\equiv$ (b mod n)

Operator modulo ma następujące własności:

1. a $\equiv$ b mod n , jeżeli n | (a-b)
2. a $\equiv$ b mod n wynika, że b $\equiv$ a mod n

# Odwrotność względem mnożenia

Jeżeli n jest liczbą pierwszą to wszystkie elementy $Z_n$ (Jest to zbiór liczb całkowitych na który odwzorowane są wszystkie liczby całkowite (0,1,..,n-1) będą względnie pierwsze względem n.

Wówczas można wprowadzić odwrotność względem mnożenia czyli:

Dla każdego w $\in Z_n$ istnieje takie $w^{-1}$ że : $w * w^{-1} \equiv 1 mod n$

Warunkiem istnienia odwrotności jest to to, żeby nwd(w,n) = 1. Istnieje tylko jedna odwrotność (mod n) spełniająca zależność: $w * w^{-1} \equiv 1 mod n$

# Algorytm plecakowy

- Problem plecakowy polega na określeniu, które przedmioty ze zbioru wszystkich możliwyych przedmiotów znajdują się w pojemniku
- Każdy przedmiot posiada swoją szczególną wagę
- Trudność polega na określeniu, które przedmioty znajdują się w plecaku przy danej wadze wypełnienia plecaka

Przedmioty: P1 - 10g, P2 - 3g, P3 - 5g, P4 - 11g, P5 - 14g
Waga Plecaka: 17g - które przedmioty są w plecaku ?
Waga $= \text{P5} + \text{P2} = 14\text{g} + 3\text{g} = 17\text{g}$

# Podstawowe pojęcia

Wektor ładunku: $a = (a_1, a_2, a_3, .., a_n)$
Tekst jawny: $x = (x_1, x_2, x_3, \ldots, x_n)$
Tekst zaszyfrowany:

$$S = a * x = \sum_{i=1}^{n} (a_i * x_i)$$

Wektor ładunku - lista elementów które mogą być włożone do plecaka, każdy element jest równy swojej wadze
x - wybór elementów wektora ładunku
a - klucz jawny
S - szyfrogram

Odbiorca żeby odszyfrować wiadomość musi uzyskać x znając a i S.

# Warunki dla algorytmu plecakowego

1. Dla każdej wartości S jest tylko jedna odwrotność, np. $a = (1, 3, 2, 5)\,; S = 3$
2. Odszyfrowanie jest ogólnie trudne lecz staje się łatwe przy dostępie do specjalnych informacji

Tworzymy wektor szybko rosnący, każdy element a ma być większy niż suma poprzedzających go elementów:

$$a_i > \sum_{j=1}^{i-1} a_j \; gdzie \; 1 < i \leq n$$

# Przykład - łatwy problem plecakowy

$$a' = (171, 197, 459, 1191, 2410)$$

Np.: S'=a' * x'  S' = 3798
1. S' = 3798 > 2410 czyli $x_5 = 1$

Ponieważ bez $x_5$ inne elementy nie wystarczyłyby do 3798
2. 3798-2410 = 1388 > 1191 $\longmapsto x_4 = 1$
3. 1388-1191 = 197 < 459 $\longmapsto x_3 = 0$
4. 197 = 197 $\longmapsto x_2 = 1$
5. 197 -197 = 0  171 $\longmapsto x_1 = 0$

Problem: Jak połączyć łatwy problem plecakowy z trudnym problemem plecakowym?

# Tworzenie trudnego problemu plecakowego

- Losowo wybieramy wektor szybko rosnący $a'$ o $n$ elementach.
- Wybieramy dwie liczby całkowite m i w takie, że m jest większe od sumy elementów a' oraz $nwd(m, w) = 1$ oraz $m > w$
- Budujemy trudny problem plecakowy – mnożymy łatwy problem plecakowy przez: $w \ modm$ czyli $a = a' * w \ modm$
- Powstały wektor nie będzie łatwo rosnący.

Istnieje możliwość konwersji trudnego problemu plecakowego do, łatwego problemu plecakowego.
Dlatego, że nwd (m,w)=1 to istnieje tylko jedna odwrotność do w czyli $w^{-1}$: $w * w^{-1} = 1 \ modm$

# System plecakowy

a' - wektor szybko rosnący; klucz prywatny, wybrany

m - liczba całkowita większa od sumy elementów a' ; klucz prywatny, wybrany

w - liczba całkowita względnie pierwsza z m ; klucz prywatny, wybrany

$w^{-1}$ - odwrotność w mod m ; klucz prywatny, obliczony

a - trudny problem plecakowy, równy a' * w mod m ; klucz jawny, obliczony

# Przykład 1

a' = 1, 3, 7, 13, 26, 65, 119, 267
w = 467 , m = 523 ,

w*$w^{-1}$ ≡ 1 mod m ; w*$w^{-1}$ = k * m +1 ; k - liczba całkowita

467 * 28 = 25*523 +1, czyli dla k=25 jest spełniona równość, czyli $w^{-1}$ = 28

Obliczamy trudny problem plecakowy:
$a_1$ = 1 * 467 mod 523 = 467;
$a_2$ = 3 * 467 mod 523 = 355;
$a_3$ = 7 * 467 mod 523 = 131;
.....
a = 467, 355, 131, 318, 113, 21, 135, 215

# Szyfrowanie

Szyfrowanie:
a = 467, 355, 131, 318, 113, 21, 135, 215
x = 01001011

Tekst zaszyfrowany = 0*467 + 1*355 + 0*131 + 0*318 + 1*113 + 0*21 + 1*135 ++ 1*215 = 818 ; S = 818

# Deszyfrowanie

S' $= S * w^{-1}$ mod m ; 818 * 28 mod 523 = 415

a' = 1, 3, 7, 13, 26, 65, 119, 267

- S' $> a'_8$ ; 415 $>$ 267 czyli $x_8 = 1$
- 415 – 267 = 148 $>$ 119 czyli $x_7 = 1$
- 148 – 119 = 29 $<$ 65 czyli x6 = 0
- 29 $>$ 26 czyli $x_5$ =1
- 29-26 = 3 $<$ 13 czyli $x_4 = 0$
- 3 $<$ 7 czyli $x_3 = 0$
- 3 = 3 czyli $x_2$ =1 ; x1=0

Teks jawny = 01001011

# RSA - wstęp

- System stworzony przez Rivesta Shamira i Adelemana
- Tekst jawny jest szyfrowany blokami, z których każdy ma wartość binarną mniejszą od pewnej liczby n

Szyfrowanie:
$C = M^e \; mod n$
Deszyfrowanie:
$M = C^d \; mod \; n = (M^e)^d \; mod \; n = \left(M^d\right)^e \; mod \; n$

Klucz jawny - PK = n i e
Klucz prywatny - SK = n i d

# RSA - zasada działania

1. Wybieramy dwie liczby pierwsze – p i q
2. Obliczamy n = p*q
3. Wybieramy liczbę e taką, że nwd $(\phi(n), e) = 1$ i $1 < e < \phi(n)$
   $\phi(n) = (p-1)(q-1)$ – funkcja Eulera
4. Obliczamy odwrotność wybranej liczby e – czyli d
   $d * e \equiv 1 mod \phi(n)$ ; $k * \phi(n) + 1 = d * e$ gdzie: k – l.całkowita

Klucz publiczny: n i e
Klucz prywatny: n i d
Szyfrowanie: $C = M^e (mod\ n)$ M – wiadomość; M < n
Odszyfrowanie: $M = C^d (mod\ n)$

# RSA - Przykład

1. Wybieramy p $=$ 7 i q $=$ 17
2. Obliczamy n$=$p*q $=$ 7 * 17 $=$ 119 ; $\phi(n)$ $=$ 6*16 $=$ 96
3. Wybieramy e $=$ 5 ; nwd $(\phi(n), e) = 1$
4. Obliczamy odwrotność e czyli d ; e*d $=$ k *$\phi(n) + 1$
   5*d $=$ k * 96 $+$ 1 ; 5 * 77 $=$ 385 $=$ 4 * 96 $+$ 1 czyli d $=$ 77

Szyfrowanie wiadomości:
m$=$1410 czyli m1$=$14 , m2$=$10
c1$=14^5 \pmod{119} = 63$
c2$=10^5 \pmod{119} = 40$
C $=$ 6340
Odszyfrowywanie wiadomości:
m1$=63^{77} \pmod{119} = 14$
m1$=40^{77} \pmod{119} = 10$

# Kryptoanaliza algorytmu RSA

- Metoda brutalna - wypróbować wszystkie klucze prywatne
- Rozłożyć n na dwa czynniki pierwsze, czyli liczbę n na iloczyn dwóch liczb. To umożliwia obliczenie $\phi(n)$=(p-1)(q-1) a to umożliwia obliczenie d z e*d = k *$\phi(n) + 1$
- Określić $\phi(n)$ bezpośrednio
- Określić d bezpośrednio

# Nieporozumienie między szyframi sym. i asym.

1. Szyfrowanie z kluczem jawnym jest bardziej zabezpieczone przed złamaniem niż szyfry konwencjonalne
2. Szyfrowanie z kluczem jawnym jest techniką ogólnego zastosowania, która uczyniła szyfrowanie konwencjonalne przestarzałym
3. Dystrybucja klucza w przypadku szyfrowania z kluczem jawnym jest trywialna, zwłaszcza w porównaniu z szyframi konwencjonalnymi

# Problem logarytmu dyskretnego

Problem dotyczy potęgi liczby całkowitej modulo n

Rozważmy potęgi liczby 7 modulo 19:
$7^1 = 7 mod 19$
$7^2 = 49 mod 19 = 11 mod 19$
$7^3 = 343 mod 19 = 1 mod 19$
$7^4 = 2401 mod 19 = 7 mod 19$
$7^5 = 16807 mod 19 = 11 mod 19$

Ciąg ten się powtarza.

Niektóre ciągi mają długość 18, czyli nie pojawia się okres.

Jeżeli a jest pierwiastkiem pierwotnym liczby n to jej potęgi: $a^1, a^2, a^3, ..., a^{\phi(n)}$ są różne mod n i wszystkie są względnie pierwsze z n.

# Indeksy

Rozważmy teraz pierwiastek pierwotny a dla pewnej liczby pierwszej p.

Wiemy, że potęgi a od 1 do (p -1) dają w wyniku liczby całkowite od 1 do (p - 1), przy czym każda z nich występuje dokładnie raz.

Dla każdej liczby całkowitej b i pierwiastka pierwotnego a z liczby pierwszej p można znaleźć jeden wykładnik i taki, że:
$b = a^i modp$ gdzie $0 \leq i < (p - 1)$

Wykładnik i nazywamy indeksem liczby b przy podstawie a mod p:
$ind_{a,p}(b)$

# logarytm dyskretny

- Dla zwykłych dodatnich liczb rzeczywistych funkcja logarytmiczna jest odwrotnością potęgowania.
- W arytmetyce modulo istnieje analogiczna funkcja.
- Istnieje analogia między właściwościami klasycznych logarytmów a indeksami.
- Dlatego te ostatnie nazywane są logarytmami dyskretnymi.

# Obliczanie logarytmów dyskretnych

$$y = g^x \; mod \; p$$

- Przy danych g,x,p obliczenie y jest sprawą prostą. W najgorszym wypadku trzeba będzie wykonać x mnożeń g i dokonać operacji mod p.
- Jednak, przy danych y,g,p bardzo trudno obliczyć x (obliczyć logarytm dyskretny)
- Trudność jest podobnego rzędu co w przypadku rozkładania na czynniki pierwsze potrzebne w algorytmie RSA

# Wymiana kluczy Diffiego-Hellmana

- Pierwszy opublikowany algorytm szyfrowania z kluczem jawnym
- Powszechnie nazywany - Wymianą kluczy Diffiego-Hellmana
- Celem algorytmu jest umożliwienia użytkownikom A i B, bezpiecznej wymianie kluczy
- Efektywność algorytmu D-H zależy od stopnia trudności obliczania logarytmu dyskretnego

# Wymiana kluczy D-H

Globalne elementy jawne:

q - liczba pierwsza

$\alpha$ - jest pierwiaskiem pierwotnym q; $\alpha < q$

Generowanie klucza użytkownika A:

Wybór klucza prywatnego - $SK_A$ ; $SK_A < q$

Oblicznie Jawnego - $PK_A = \alpha^{SK_A} \, mod \, q$

Generowanie klucza użytkownika B:

Wybór klucza prywatnego - $SK_B$ ; $SK_B < q$

Oblicznie Jawnego - $PK_B = \alpha^{SK_B} \, mod \, q$

Generowanie klucza do wymiany informacji przez A:

$K = (PK_B)^{SK_A} \, mod \, q$

Generowanie klucza do wymiany informacji przez B:

$K = (PK_A)^{SK_B} \, mod \, q$

# Wymiana kluczy D-H

Jawne=a, q

| Alice | | Bob |
|---|---|---|
| 1. Wybór $SK_A$<br>$PK_A = a^{SK_A} \bmod q$ | 2. $PK_A$ → | |
| | | 3. Wybór $SK_B$<br>$PK_B = a^{SK_B} \bmod q$ |
| | ← 4. $PK_B$ | |
| 5. $K = (PK_B)^{SK_A} \bmod q$ | | |
| | ← $E_K(M)$ | 6. $K = (PK_A)^{SK_B} \bmod q$ |
| | $E_K(M)$ → | |

# Bezpieczeństwo - Wymiany D-H

Bezpieczeńśtwo wymiany kluczy D-H wynika z tego, że o ile stosunkowo łatwo potęguje się modulo o tyle obliczyć logarytm dyskretny jest bardzo trudno