Diffi Helman

- Alicja i Bob w dowolny sposób wybierają dwie liczby względnie pierwsze: dużą liczbę p oraz liczbę g, będącą generatorem grupy multiplikatywnej Z_p^*.
- Alicja losuje liczbę a; Bob losuje liczbę b
- Alicja wysyła Bobowi $\mathbf{g^a \bmod p}$; Bob wysyła Alicji $\mathbf{g^b \bmod p}$
  $\mathbf{k = (g^b)^a \bmod p}$

RSA

Generowanie kluczy

- Wybieramy losowo dwie duże liczby pierwsze p i q
- Obliczamy wartość n = p*q
- Obliczamy: φ(n) = (p-1)*(q-1)
- Wybieramy liczbę e (1 < e < φ(n)) względnie pierwszą z φ(n)
- Znajdujemy liczbę d odwrotną do e mod φ(n): $d = e^{-1} \bmod \varphi(n)$

Klucz publiczny jest definiowany jako para liczb (n, e),
natomiast **kluczem prywatnym** jest para (n, d).

Szyfrowanie i deszyfrowanie

Zanim zaszyfrujemy wiadomość, dzielimy ją na bloki m_i o wartości liczbowej nie większej niż n, a następnie każdy z bloków szyfrujemy według wzoru:
$\mathbf{c_i = m_i^e \bmod n}$

Zaszyfrowana wiadomość będzie się składać z kolejnych bloków c_i. Tak stworzony szyfrogram przekształcamy na tekst jawny, odszyfrowując kolejne blok ci według wzoru:
$\mathbf{m_i = c_i^d \bmod n}$

Podpis RSA

Generacja

- H(m) - skrót wiadomości
- h = H(M)
- $s=h^d \bmod n$

Weryfikacja

- h_1 = H(m')
- $h_2 = S^e \bmod n$
- Jeżeli $h_1 = h_2$ to OK

ElGamal

Generowanie klucza

wybieramy dowolną liczbę pierwszą p, dowolny generator α podgrupy multiplikatywnej, tzn. taki element, którego rząd jest równy $p - 1$, oraz dowolne k takie, że: $1 < t < p$. Liczymy β:

$\beta = \alpha^t \bmod p$

Następnie publikujemy $k_1 = (p,\alpha,\beta)$ jako **klucz publiczny** i zachowujemy $k_2 = (p, t)$ jako **klucz prywatny.**

Szyfrowanie i deszyfrowanie

Szyfrowanie: mając do zaszyfrowania wiadomośćx, przedstawiamy ją jako element grupy $[1 < x < p - 1]$ wybieramy losowo liczbę r i liczymy (modulo p)

$$\mathbf{C=(Y_1, Y_2) = (\alpha^r \bmod p, x * \beta^r \bmod p)}$$

Deszyfrowanie:

$$\mathbf{D_{k_2}(Y_1, Y_2) = Y_2 * (Y_1^t)^{-1} \bmod p = y_2 * Y_1^{p-1-t} \bmod p}$$

Podpis cyfrowy

Klucz jest generowany w ten sam sposób.

- h = H(m) - skrot wiadomości
- losujemy liczbę r $1 < r < p-1$ oraz $\gcd(r, p-1) = 1$
- $u = \alpha^r \bmod p$
- obliczamy $r^{-1} \bmod (p-1)$
- $s = r^{-1} * (h - t* u) \bmod (p-1)$
- podpisem jest para m = (u, s)

Żeby zweryfikować podpis:

- otrzymujemy wiadomosc oraz u' i s'
- wyznaczć skrot h' wiadomosci
- $f = \alpha^{h'} \bmod p$
- $g = \beta^{u'} * (u')^{s'} \bmod p$
- jesli f = g to OK

DSA

Generacja kluczy

- wybrać liczbę pierwszą p o dlugości L bitow (512 <= L <=1024) i jest

wielokrotnością 64:
$2^{511 + 64t} < p < 2^{512 + 64t}$, $0 <= t <= 8$

- Wybrać liczbę oierwszą q o dlugości 160 bitow ktora dzieli liczbę p-1
  $2^{159} < q < 2^{160}$, $q | (p-1)$
  - Wybrać liczbę $\alpha \in Z_p^*$ i obliczyć $g = \alpha^{(p-1)/q}$
  - jesli g = 1 to, szukać innej α
- wybrać losowo liczbę x taką, że $1 <= x <= q-1$
- obliczyć $y = g^x \bmod p$

klucze: publiczny $k_1 = (p, q, g, y)$ prywatny $k_2 = (p, q, g, x)$

Generacja podpisu

- wybrać losową, tajną liczbę calkowitą: $0 < k < q$
- Obliczyć $r = (g^k \bmod p) \bmod q$
- Obliczyć $s = [k^{-1}(h(m) + x * r)] \bmod q$
- **Podpisem cyfrowym jest para liczb (r,s)**

Weryfikacja podpisu

- obliczyć $w = s^{-1} \bmod q$
- obliczyć $u_1 = w * h(m) \bmod q$
- obliczyć $u_2 = r * w \bmod q$
- $v = [(g^{u1} * y^{u2}) \bmod p] \bmod q$
- Jezlei v = r to OK