## logaritmo discreto

- sia $G$ un gruppo ciclico di ordine $n$, sia $g$ un generatore di $G$
- vuol dire che $G = \{g, g^2, g^3, \ldots, g^{n-1}, g^n = 1\}$
- dato $y \neq 1 \in G$
- bisogna determinare l'unico intero $x$ con $1 \leq x \leq n-1$ tale che

$$g^x = y$$

- l'intero $x$ si chiama il logaritmo discreto di $y$ in base $g$, e si denota con $\log_g y$
- Esempio: il gruppo $U(\mathbb{Z}_7) = \{1, 2, 3, 4, 5, 6\}$ è ciclico, 3 è un generatore
- $\log_3 6 = 3$ perché $3^3 = 6$ modulo 7

## logaritmo discreto come funzione unidirezionale

- in generale, lavoreremo con il gruppo $U(\mathbb{Z}_p) = \mathbb{Z}_p^*$
- dati $g$ generatore di $\mathbb{Z}_p^*$ e $x$ tale che $1 \leq x \leq p-1$, calcolare $y = g^x$ è computazionalmente facile
- ($y \equiv g^x \pmod{p}$ – si usa l'algoritmo square-and-multiply )
- si ritiene che, dati $g$ generatore di $\mathbb{Z}_p^*$ e $y \in \mathbb{Z}_p^*$, determinare $x = \log_g y$ sia difficile (sotto opportune ipotesi su $p$)
- in particolare, $p$ dev'essere grande (1024 bit)
- quindi $p \approx 2^{1023}$
- dati $g$ e $y$, posso trovare $x$ tale che $g^x = y$ per tentativi – calcolando $g^x$ per tutti gli $x$, $1 \leq x \leq p-1$
- ma il numero di tentativi è enorme

## crittosistema Elgamal (ca 1985)

- sia $p$ un primo, $g$ un elemento primitivo mod $p$
- $\mathcal{P} = U(\mathbb{Z}_p)$
- $\mathcal{C} = U(\mathbb{Z}_p) \times U(\mathbb{Z}_p)$
- Lo spazio delle chiavi è

$$\mathcal{K} = \{(p, g, a, \beta) \mid \beta \equiv g^a \pmod{p}\}.$$

- $p$, $g$ e $\beta$ sono la chiave pubblica $a$ è la chiave privata

## crittosistema Elgamal

- prima di cifrare il messaggio $x \in \mathcal{P}$, Alice sceglie un numero casuale (segreto) $h \in \{2, \ldots, p-2\}$
- $e_k(x, h) = (y_1, y_2)$
- con $y_1 = g^h$, $y_2 = x\beta^h \pmod{p}$
- Bob riceve $(y_1, y_2) \in U(\mathbb{Z}_p) \times U(\mathbb{Z}_p)$ – non conosce $h$ ma conosce $a$
- calcola $y_1^a = (g^h)^a = (g^a)^h = \beta^h \pmod{p}$
- calcola $(\beta^h)^{-1} \pmod{p}$ e ottiene $x = y_2(\beta^h)^{-1}$
- $d_k((y_1, y_2)) = y_2(y_1^a)^{-1} \pmod{p}$
- notare la somiglianza con DH – non si inverte la funzione $x \to g^x \pmod{p}$
- Alice sceglie un nuovo $h$ a ogni trasmissione

## esempio

- Bob sceglie $p = 83$, $g = 2$ e la chiave privata $a = 30$
- $\beta = 2^{30} \equiv 40 \bmod 83$
- la chiave pubblica di Bob è $(83, 2, 40)$
- il messaggio di Alice è $x = 54$ – il numero scelto per la cifratura è $h = 13$
- Alice invia $(g^h, x\beta^h) = (2^{13}, 54 \cdot 40^{13}) \equiv (58, 71) \bmod 83$
- per decifrare, Bob calcola $(g^a)^h = 58^{30} = 9$
- l'inverso di 9 mod 83 è 37 – il messaggio è quindi $37 \cdot 71 = 54 \bmod 83$

- anche per Elgamal, si usa $p$ di almeno 1024 bit, $p - 1$ con un fattore primo grande e a fattorizzazione nota
- anche per violare Elgamal, basta che Eve sappia risolvere il DH problem
- se dati $g^h$ e $\beta = g^a$ sa trovare $g^{ah} = \beta^h$, può leggere il messaggio
- se Eve sa risolvere il logaritmo discreto può ricavare l'esponente $h$ e quindi ricavare direttamente $x$
- oppure ricavare $a$ e decifrare come Bob (conviene – $h$ cambia in ogni trasmissione)