

# Digital Signature Algorithm

Mattia Biral

2024-03-09

# 1 Introduzione

Il **Digital Signature Algorithm** è un sistema crittografico a chiave pubblica e uno *standard federale per l'elaborazione delle informazioni* per le **firme digitali**.

La chiave privata è utilizzata per generare la firma, mentre la chiave pubblica per verificarla.

E' basato sul problema matematico del logaritmo discreto.

## 1.1 Firma digitale

La firma digitale fornisce:

- Autenticazione: so chi ha inviato il messaggio
- Integrità: so che il documento non è stato modificato dopo la firma
- Non-ripudio: l'autore non può dire di non essere stato lui a firmare (side-effect dell'autenticazione)

## 1.2 Operazioni

DSA si svolge in quattro operazioni:

- Generazione delle chiavi
- Distribuzione delle chiavi
- Firma
- Verifica della firma

## 2 Algoritmo

### 2.1 Generazione delle chiavi

#### 2.1.1 Parametri

I parametri dell'algoritmo sono  $(p, q, g)$

- $H$  funzione crittografica di hash di lunghezza  $|H|$  bit (se  $|H|$  è maggiore della lunghezza del modulo  $N$  solo gli  $N$  bit più significativi dell'output saranno utilizzati)
- $L$  lunghezza della chiave
- $N$  lunghezza del modulo tale che  $N < L \wedge N \leq |H|$
- $q$  primo di  $N$  bit
- $p$  primo di  $L$  bit tale che  $q \mid p - 1$
- $h$  casuale in  $\{2, \dots, p - 2\}$
- $g := h^{p-1/q} \mod p$  (se  $g = 1$  è necessario generare un nuovo  $h$ )

#### 2.1.2 Chiavi per-user

- $x$  casuale in  $\{1, \dots, q - 1\}$ , **chiave privata**
- $y := g^x \mod p$ , **chiave pubblica**

### 2.2 Distribuzione delle chiavi

Il firmatario pubblica la chiave *pubblica*  $y$  e mantiene segreta  $x$

## 2.3 Firma

Un messaggio  $m$  è firmato come segue:

- $k$  casuale in  $\{1, \dots, q-1\}$
- $r := (g^k \bmod p) \bmod q$  (se  $r = 0$  è necessario generare un nuovo  $k$ )
- $s := (k^{-1}(H(m) + xr)) \bmod q$  (se  $s = 0$  è necessario scegliere un'altro  $k$ )

La firma è  $(r, s)$

Nota: dato che ogni volta che si firma il messaggio viene scelto un  $k$  casuale, è molto probabile che più firme di uno stesso documento siano **diverse ma ugualmente valide**.

Nota: la computazione più costosa riguarda  $r$ , ma può essere fatta prima di conoscere  $m$  poiché non dipende da esso.

Nota: la seconda computazione più costosa riguarda  $k^{-1}$ , ma anch'essa può essere effettuata prima che  $m$  sia noto.

## 2.4 Verifica della firma

Si verifica che una firma è autentica come segue:

- $0 < r < q \wedge 0 < s < q$  (in quanto risultati di  $\dots \bmod q$ )
- $w := s^{-1} \bmod q$
- $u_1 := H(m) \cdot w \bmod q$
- $u_2 := r \cdot w \bmod q$
- $v := (g^{u_1} y^{u_2} \bmod p) \bmod q$
- $v = r$

### 3 Correttezza dell'algoritmo

Lo schema di firma è corretto nel senso che il verificatore accetterà sempre firme autentiche.

Dato che

$$g = h^{(p-1)/q} \pmod{p} \implies g^q \equiv h^{p-1} \equiv 1 \pmod{p}$$

poiché  $g > 0$  e  $g$  è primo,  $g$  ha ordine  $q$ .

Il firmatario calcola

$$s = k^{-1}(H(m) + xr) \pmod{p}$$

Quindi

$$\begin{aligned} k &\equiv H(m)s^{-1} + xrs^{-1} \\ &\equiv H(m)w + xrw \pmod{q} \end{aligned}$$

Dal momento che  $g$  ha ordine  $q$  abbiamo

$$\begin{aligned} g^k &\equiv g^{H(m)w} g^{xrw} \\ &\equiv g^{H(m)w} y^{rw} \\ &\equiv g^{u_1} y^{u_2} \pmod{p} \end{aligned}$$

Infine

$$\begin{aligned} r &= (g^k \pmod{p}) \pmod{q} \\ &= (g^{u_1} y^{u_2} \pmod{p}) \pmod{q} \\ &= v \end{aligned}$$