Principles of Cryptocurrency Design

Appunti ed Esercizi

Francesco Pasquale

25 maggio 2026

Il sistema di firma digitale usato originariamente in Bitcoin è ECDSA. Nel 2021, l'upgrade taproot ha introdotto in Bitcoin la possibilità di utilizzare firme di Schnorr [1].

1 Firme di Schnorr

Il sistema di firma digitale di Schnorr sfrutta, come ECDSA, il fatto che il logaritmo discreto è un problema computazionalmente difficile in generale.

Il gruppo che viene usato è sempre quello della curva ellittica SECP256K1, quindi l'algoritmo di generazione della coppia di chiavi $(\texttt{sk}, \texttt{pk})$ semplicemente genera una `sk` u.a.r. in $[n]$ e calcola $\texttt{pk} = \texttt{sk} \cdot G$. L'algoritmo di firma è quello descritto qui di seguito

Algorithm 1 SIGN($\langle msg \rangle$, `sk`)

1: Scegli $k < n$ (possibilmente u.a.r. e con uno pseudorandom generator *sicuro*);
2: Calcola $R = k \cdot G$;
3: Calcola $h = \text{HASH}(\langle msg \rangle \,||\, R)$;
4: Calcola $s = k + h \cdot \texttt{sk}$
5: Restituisci la firma $\sigma = (R, s)$

Si osservi che, come per ECDSA, (1) è cruciale che il valore k scelto alla linea `2` rimanga segreto, altrimenti dal messaggio $\langle msg \rangle$, dalla firma $\sigma = (R, s)$ e da k si può ricavare la chiave segreta `sk` invertendo l'equazione alla linea `4`; (2) è cruciale che non si eseguano mai due firme di due messaggi distinti con lo stesso valore di k.

Esercizio 1. Verificare che, se conosciamo due messaggi distinti, m_1 e m_2, e due firme, σ_1 e σ_2 di m_1 e m_2 rispettivamente, ottenute usando lo stesso k alla linea `2` dell'Algoritmo 1, allora possiamo calcolare la chiave segreta `sk` con cui sono state generate le firme.

L'algoritmo di verifica della firma funziona in questo modo:

Algorithm 2 VERIFY($\langle msg \rangle$, `pk`, $\sigma = (R, s)$)

1: Calcola $h = \text{HASH}(\langle msg \rangle \,||\, R)$;
2: Return $s \cdot G == R + h \cdot \texttt{pk}$

Esercizio 2. Verificare che lo schema di firma digitale descritto negli Algoritmi 1 e 2 è *corretto*: per ogni coppia di chiavi $(\texttt{sk}, \texttt{pk})$ e per ogni messaggio $\langle msg \rangle$, si ha che

$$\textsc{Verify}\big(\langle msg \rangle, \texttt{pk}, \textsc{Sign}(\langle msg \rangle, \texttt{sk})\big) = \textsc{True}$$

Riferimenti bibliografici

[1] Claus-Peter Schnorr. Efficient signature generation by smart cards. *Journal of cryptology*, 4:161–174, 1991. `https://link.springer.com/content/pdf/10.1007/bf00196725.pdf`.