Principles of Cryptocurrency Design

Appunti ed Esercizi

Francesco Pasquale

19 marzo 2024

Sia $H : \mathbb{N} \cup \{0\} \to [n]$ una funzione tale che $H(0) = 1$ e per ogni $r \in \mathbb{N}$ si comporta come un *random oracle* [1]. A lezione abbiamo dimostrato che il protocollo seguente per Byzantine Broadcast soddisfa *Validity* con probabilità 1 e soddisfa *Consistency* con probabilità almeno $1 - (2/3)^{k-1}$.

Algorithm 1 Randomized Byzantine Broadcast

1: ROUND 0. La sorgente (il nodo 1) riceve in input b e inizializza $\text{sb}_i = b$.
2: Ogni nodo i inizializza $\text{sb}_i = \perp$.
3: PER OGNI ITERAZIONE $r = 0, \ldots, k-1$:
4: ROUND $3r$. Il *leader* $\ell_r = H(r)$ dell'iterazione:
5: Se $\text{sb}_{\ell_r} \neq \perp$ allora invia a tutti sb_{ℓ_r};
6: Altrimenti sceglie un bit $\{0, 1\}$ u.a.r. e lo invia a tutti.
7: ROUND $3r + 1$. Ogni nodo i:
8: Se $\text{sb}_i \neq \perp$ allora invia a tutti sb_i;
9: Altrimenti invia a tutti il bit ricevuto nel round $3r$ dal leader ℓ_r (se non ha ricevuto nulla da ℓ_r o ha ricevuto entrambi i bit, invia 0 o 1 arbitrariamente)
10: ROUND $3r + 2$. Ogni nodo i:
11: Se c'è un bit $\hat{b}$ che ha ricevuto nel round $3r + 1$ da almeno $2n/3$ nodi distinti allora imposta $\text{sb}_i = \hat{b}$;
12: Altrimenti imposta $\text{sb}_i = \perp$.
13: ROUND $3k$. Ogni nodo i:
14: OUTPUT sb_i

Esercizio 1. Si consideri il protocollo in Algorithm 1. Dato $\delta > 0$, quanto deve essere grande il numero di iterazioni $k = k(\delta)$ affinché la probabilità che l'algoritmo soddifi la condizione di *consistency* sia almeno $1 - \delta$? Quanto deve essere grande k se scegliamo $\delta = 1/n$? Confrontare il numero di round di questo protocollo con quello del protocollo di Dolev-Strong.

Esercizio 2. Alla linea 11 il protocollo stabilisce che ogni nodo i deve decidere come impostare il bit sb_i a seconda che abbia ricevuto o no almeno $2n/3$ "voti" per uno specifico bit $\hat{b}$ nel round precedente.

1. Mostrare un attacco che i nodi corrotti potrebbero mettere in atto se la decisione fosse presa in funzione di un numero di voti $h < 2n/3$;
2. Mostrare un attacco che i nodi corrotti potrebbero mettere in atto se la decisione fosse presa in funzione di un numero di voti $h > 2n/3$.

Esercizio 3. Cosa succederebbe se non imponessimo che $H(0) = 1$ (ossia che il leader ℓ_0 dell'iterazione $r = 0$ è il nodo sorgente)?

Riferimenti bibliografici

[1] Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *Proceedings of the 1st ACM Conference on Computer and Communications Security*, pages 62–73, 1993.