Principles of Cryptocurrency Design

Appunti ed Esercizi

Francesco Pasquale

9 marzo 2026

Nella lezione precedente abbiamo visto che, nel modello sincrono con *PKI setup*, il protocollo di Dolev-Strong [1] risolve il problema Byzantine Broadcast (BB) qualunque sia il numero f di nodi corrotti, in $f+1$ *round* di comunicazione. Qui vedremo che se rimuoviamo l'ipotesi che ci sia un *PKI setup*, allora nello stesso modello nessun protocollo può garantire *validity* e *consistency* contemporaneamente, se ci sono $n/3$ o più nodi corotti, dove n è il numero totale di nodi.

1 Lower bound per Byzantine Broadcast senza PKI

Il teorema seguente è stato dimostrato per la prima volta in [5, 4]e successivamente, con un'altra tecnica, in [2, 3].

Teorema 1 ([5, 2])**.** In assenza di PKI setup e con $f \geqslant n/3$ nodi corrotti, nessun protocollo per il problema Byzantine Broadcast può garantire *validity* e *consistency*.

Dimostrazione. Consideriamo il sistema $\mathcal{G}$ in Figura 1 con 3 nodi, dove il nodo A è la sorgente, e supponiamo per assurdo che esista un protocollo Π che garantisce *validity* e *consistency* nel sistema $\mathcal{G}$ anche in presenza di un nodo corrotto.

Consideriamo poi l'esperimento concettuale $\mathcal{H}$ in Figura 1 dove gli archi del grafo indicano i canali di comunicazione (quindi ogni nodo può comunicare soltanto con i suoi due vicini) e

- x e x' sono copie esatte di A (quindi, in particolare, entrambi i nodi "credono" di essere la sorgente A del problema BB);
- y e y' sono copie esatte di B;
- z e z' sono copie esatte di C.

Figura 1: Il sistema reale $\mathcal{G}$ e l'esperimento concettuale $\mathcal{H}$

Osserviamo che un nodo non può distinguere se si trova nel sistema reale $\mathcal{G}$ o nell'esperimento concettuale $\mathcal{H}$: per esempio, il nodo z' in $\mathcal{H}$ è una copia del nodo C e comunica con due nodi, x e

y', il primo è una copia del nodo A e il secondo è una copia del nodo B. L'esecuzione del protocollo Π può essere quindi "simulata" nell'esperimento concettuale $\mathcal{H}$ (i nodi x e x' eseguono le istruzioni previste per A, y e y' quelle previste per B e z e z' quelle previste per C). Ci chiediamo cosa danno in output i sei nodi in $\mathcal{H}$ se simuliamo l'esecuzione del protocollo Π dove al nodo x (una delle due copie della sorgente A) viene dato in input il bit 1 e al nodo x' (l'altra copia della sorgente A) viene dato in input 0.

Claim 1. I nodi x e y devono dare in output 1.

Dimostrazione. Se eseguiamo il protocollo Π in $\mathcal{G}$ dando ad A input 1 e se A e B sono nodi onesti mentre C è corrotto, allora A e B devono dare in output 1 qualunque cosa faccia C, perché stiamo assumendo per ipotesi che Π soddisfi *validity* anche in presenza di un nodo corrotto. Ora osserviamo che il comportamento del nodo corrotto C potrebbe essere il seguente: per decidere quali messaggi inviare ad A e B il nodo C potrebbe

- *Simulare* l'esecuzione del protocollo Π in $\mathcal{H}$;
- Inviare a B tutti i messaggi che nella simulazione il nodo z (che è una copia di C) invia ad y (che è una copia di B);
- Inviare ad A tutti i messaggi che nella simulazione il nodo z' (che è una copia di C) invia ad x (che è una copia di A).

I nodi x e y (che sono copie di A e B) allora devono dare in output esattamente quello che danno in output i due nodi onesti A e B in $\mathcal{G}$, ossia 1 perché ricevono in $\mathcal{H}$ gli stessi messaggi che ricevono A e B in $\mathcal{G}$. □

Claim 2. I nodi x' e z devono dare in output 0.

Claim 3. I nodi y e z devono dare in output lo stesso bit.

Esercizio 1. Dimostrare i Claim 2 e 3.

Siccome i Claim 1 2 e 3 non possono essere tutti e tre veri abbiamo trovato un assurdo. Quindi la nostra ipotesi di partenza, ossia che il protocollo Π garantisce *validity* e *consistency* nel sistema $\mathcal{G}$ in presenza di un nodo corrotto, non può essere vera. □

È importante riflettere sul perchè il risultato di impossibilità nel Teorema 1 non si applica nel caso ci sia un *PKI setup*. In particolare, nel prossimo esercizio vediamo perché il teorema non si applica al protocollo di Dolev-Strong.

Esercizio 2. Considerate il sistema $\mathcal{G}$ e l'esperimento concettuale $\mathcal{H}$ in Figura 1.

1. Eseguire il protocollo di Dolev-Strong con $f = 1$ nel sistema $\mathcal{G}$, quando tutti i nodi sono onesti, A è il nodo sorgente e riceve in input 1.
2. Simulare il protocollo di Dolev-Strong con $f = 1$ nell'esperimento concettuale $\mathcal{H}$ dove x e x' sono copie esatte[1] di A e ricevono in input 1, y e y' sono copie esatte di B e z e z' sono copie esatte di C (tutti i nodi sono onesti e seguono il protocollo).
3. Osservare che le esecuzioni dei due punti precedenti sono equivalenti (A, B e C non possono distinguere se sono nel sistema $\mathcal{G}$ o nell'esperimento concettuale $\mathcal{H}$).

[1] Ossia entrambe hanno la stessa chiave privata

4. Simulare il protocollo di Dolev-Strong con $f = 1$ nell'esperimento concettuale $\mathcal{H}$ in cui x e x' sono copie esatte di A, ma alla copia di A in x viene dato in input 1 mentre alla copia di A in x' viene dato in input 0; y e y' sono copie esatte di B e z e z' sono copie esatte di C (tutti i nodi sono onesti e seguono il protocollo). Quali sono gli output dei sei nodi? In che punto la dimostrazione del Claim 1 non funziona nel caso del protocollo di Dolev-Strong? Perché?

Riferimenti bibliografici

[1] Danny Dolev and H. Raymond Strong. Authenticated algorithms for Byzantine agreement. *SIAM Journal on Computing*, 12(4):656–666, 1983. `https://www.cs.huji.ac.il/~dolev/pubs/authenticated.pdf`.

[2] Michael J Fischer, Nancy A Lynch, and Michael Merritt. Easy impossibility proofs for distributed consensus problems. In *Proceedings of the fourth annual ACM symposium on Principles of distributed computing*, pages 59–70, 1985. `https://dl.acm.org/doi/pdf/10.1145/323596.323602`.

[3] Michael J. Fischer, Nancy A. Lynch, and Michael Merritt. Easy impossibility proofs for distributed consensus problems. *Distributed Computing*, 1:26–39, 1986. `https://apps.dtic.mil/sti/pdfs/ADA157411.pdf`.

[4] Leslie Lamport, Robert Shostak, and Marshall Pease. The Byzantine Generals Problem. *ACM Transactions on Programming Languages and Systems*, pages 382–401, July 1982. `https://dl.acm.org/doi/pdf/10.1145/357172.357176`.

[5] Marshall Pease, Robert Shostak, and Leslie Lamport. Reaching agreement in the presence of faults. *Journal of the ACM (JACM)*, 27(2):228–234, 1980. `https://dl.acm.org/doi/pdf/10.1145/322186.322188`.