

Principles of Cryptocurrency Design

Appunti ed Esercizi

Francesco Pasquale

3 marzo 2025

1 Byzantine Broadcast nel modello sincrono, permissioned, con PKI setup

Si consideri il seguente sistema distribuito:

- C'è un insieme di n nodi $[n] = \{1, 2, \dots, n\}$;
- Ogni nodo può eseguire computazioni locali arbitrarie e inviare messaggi di lunghezza arbitraria a ogni altro nodo.

E il problema *Byzantine Broadcast (BB)*: Degli n nodi, ce ne sono f *corrotti*, con $0 \leq f \leq n$, gli altri $n - f$ nodi sono *onesti*. Al nodo 1 (la *sorgente*) viene affidato in input un messaggio b . Vogliamo progettare un *protocollo* al termine del quale ogni nodo onesto i dia in output un valore y_i tale che

- **Consistency**: Se i e j sono due nodi onesti, allora $y_i = y_j$;
- **Validity**: Se la sorgente è un nodo onesto, allora $y_i = b$ per ogni nodo onesto i .

Gli $n - f$ nodi onesti applicheranno il nostro protocollo. Gli f nodi corrotti non sappiamo come si comporteranno: la nostra assunzione *worst-case* è che tutti i nodi corrotti si coalizzeranno per far fallire il nostro protocollo. Non sappiamo chi sono i nodi corrotti.

Assumiamo di essere in un sistema *sincrono*:

- C'è un *global clock* noto a tutti i nodi che scandisce il tempo in *round* discreti: $0, 1, 2, \dots$;
- Ogni messaggio inviato in un round t arriva a destinazione prima dell'inizio del round $t + 1$.

Assumiamo che ci sia un *Public Key Infrastructure (PKI)* setup:

- Ogni nodo i ha una coppia di chiavi (sk_i, pk_i) ;
- L'insieme delle chiavi pubbliche $\{pk_i : i \in [n]\}$ è noto a tutti a priori;
- Dato un messaggio m , indichiamo con $\langle m \rangle_i$ il messaggio m con l'aggiunta di una firma valida eseguita con la chiave segreta sk_i .

Esercizio 1. Mostrare un protocollo banale che soddisfa la condizione di *validity* ma non soddisfa la condizione di *consistency*¹ e un protocollo banale che, viceversa, soddisfa *consistency* ma non *validity*.

¹ *Consistency*: Tutti i nodi onesti danno in output lo stesso valore; *Validity*: Se la sorgente è onesta e riceve in input b , tutti i nodi onesti danno in output b .

Esercizio 2. Mostrare un protocollo banale nel caso in cui l'insieme dei nodi corrotti è noto.

Esercizio 3. Spiegare perché nessuno dei due protocolli seguenti funziona:

Algorithm 1 Tentativo 1

ROUND 0. La sorgente (il nodo 1) riceve in input b .
ROUND 1. La sorgente invia $\langle b \rangle_1$ a tutti i nodi.
ROUND 2. Ogni nodo i :
 se nel ROUND 1 ha ricevuto $\langle \hat{b} \rangle_1$, invia $\langle \hat{b} \rangle_i$ a tutti i nodi;
 altrimenti invia $\langle 0 \rangle_i$ a tutti i nodi.
ROUND 3. Ogni nodo i :
 se c'è un unico valore $\hat{b}$ per cui nel ROUND 2 ha ricevuto
 più di $n/2$ messaggi $\langle \hat{b} \rangle_j$, restituisce in output $\hat{b}$;
 altrimenti restituisce in output 0.

Algorithm 2 Tentativo 2

ROUND 0. La sorgente (il nodo 1) riceve in input b ;
ROUND 1. La sorgente invia $\langle b \rangle_1$ a tutti i nodi;
ROUND 2. Ogni nodo i
 se nel ROUND 1 ha ricevuto $\langle \hat{b} \rangle_1$, invia $\langle \hat{b} \rangle_i$ a tutti i nodi;
 altrimenti invia $\langle 0 \rangle_i$ a tutti i nodi.
ROUND 3. Ogni nodo i
 se nel ROUND 2 ha ricevuto $\langle \hat{b} \rangle_j$ da ogni altro nodo j ,
 restituisce in output $\hat{b}$;
 altrimenti restituisce in output 0.

Nella prossima lezione vedremo il protocollo di Dolev-Strong [1]. Cercate di risolvere il prossimo esercizio in autonomia prima di vedere la soluzione trovata da Dolev e Strong.

Esercizio 4. Provare a progettare un protocollo per BB nel modello sincrono con PKI: dimostrarne la correttezza in funzione del numero f di nodi corrotti, oppure trovare un attacco che gli f nodi corrotti possono mettere in atto per far fallire il protocollo.

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>.