Valutazione di un nuovo utente della rete

Contesto Rete VPN decentralizzata attiva e con connessione multi-hop tra una generica coppia Agent-Client.

Problema 1: identificare gli Agent in posizione endnode che inviano contenuto non originale In una rete VPN decentralizzata gli Agent endnode hanno il compito di recuperare una risorsa web e restituirla al Client attraverso la rete VPN instaurata. L'Agent, che quindi ha accesso intrinseco al dato ottenuto, cifrato o meno, può modificarlo prima di inviarlo al Client. Questa operazione può avere scopi malevoli, tra cui quello del guadagno economico. In possibile scenario, in questo contesto, è quello che vede l'incremento della grandezza del pacchetto da Agent a Client al fine di maggiorare i consumi della banda stabilita nel contratto iniziale di fornitura del servizio.

Problema 2: scoraggiare abusi della rete Per scoraggiare abusi della rete si utilizza una *prova di lavoro* (Proof of Work) per attestare che il nuovo utente della rete ha speso delle risorse al fine di guadagnare reputazione nel contesto distruibuito.
La reputazione iniziale, all'ingresso della rete distribuita, ha valore zero.

Soluzione problema 1

- Dati civetta: è richiesta dal Client all'Agent, con frequenza variabile, una risorsa dal valore noto al Client per la verifica di integrità. Se il test risulta negativo si ipotizza la modifica della risorsa da parte dell'Agent.
- Dati duplicati: il Client si connette a due Agent diversi e chiede ad entrambi la risorsa desiderata, alla ricezione si comparano per la verifica di integrità. Se il test risulta negativo si ipotizza la modifica della risorsa da parte di uno dei due Agent.
  Aspetti negativi: questa soluzione non è applicabile in caso di connessioni con la risorsa che prevedono *stati*.(??)
- Hash Dato: il Client si connette ad un Agent al quale chiede una risorsa e successivamente chiede ad un altro Agent l'hash della risorsa stessa. Il Client effettua quindi un hash (stessa funzione hash) della risorsa ottenuta dall'Agent e la compara con l'hash ottenuto dal secondo Agent. Ulteriore implementazione può essere effettuata richiedendo a più agent l'hash della risorsa.
  Se il test risulta negativo si ipotizza la modifica della risorsa da parte del primo Agent.
  Aspetti negativi: questa soluzione non è applicabile in caso di connessioni con la risorsa che prevedono *stati*.
  Caso d'uso: si reputa questo metodo adatto a verificare un Agent al suo primo ingresso nella rete in quanto non è spesso adatto per l'aspetto negativo sopra descritto.

Soluzione problema 2 L'originalità della soluzione proposta è l'operazione di Proof of Work da far compiere al nuovo utente. Si propone come prova di lavoro che l'utente, prima di entrare attivamente a far parte della rete o durante la sua partecipazione, consumi della banda internet.
La Proof of Work, quindi, consiste nello sfruttare la sua banda internet per ottenere una risorsa internet, sulla quale viene verificata l'integrità, come descritto nella soluzione *Dato Hash* proposta nella soluzione al problema 1.
Più specificatamente, nel caso d'uso si identificano un Agent A, un Client C ed un nuovo utente X che vuole entrare nella rete e deve dimostrare *la prova di lavoro*.
La situazione iniziale prevede una connessione tra A e C. Per verificare che A fornisca dato d^A originale, C richiede a X di ottenere lo stesso dato d^X e di eseguirne la funzione hash così da ottenere $H(d^X) = h_1$. h_1 sarà restituito a C che ha già effettuato la stessa funzione hash su d^A proveniente da A ed ha ottenuto $H(d^A) = h_2$.
Il test di integrità verifica che $h_1 = h_2$.
Se il test ha successo si ha certezza che:

1. il nuovo utente X ha eseguito correttamente la proof of work
2. A ha fornito a C un dato originale, a meno che sia A che X applichino le stesse modifiche al dato prima di effettuarne la funzione hash.
   Questa vincolo è da verificare.

Considerazioni In un contesto di rete distribuita si possono ipotizzare n attori nel ruolo di X e n test di integrità che C effettua per verificare l'integrità del dato ottenuto da A. *riferimento al consenso BFT*
Il risultato dei test condiziona la reputazione di ogni attore nel ruolo di X.
Si può ipotizzare l'applicazione della soluzione m volte, tante quante sono le entità per cui effettuare la proof of work, e considere vincoli come tempo t minimo di esecuzione della proof of work o la quantità minima di dati q che X deve ottenere su cui applicare la funzione hash.

Considerando una costo ipotetico c associato al consumo di una quantità q di dati per un'unità di tempo t si può impostare una soglia $S = c * q * t$ superata la quale la proof of work si può considerare eseguita.
L'ultima affermazione garantisce l'asimmetria della proof of work (??). (che è: hash difficile da calcolare perchè spendo risorsa di tempo e di banda e facile da verificare dal client perchè effettuo un semplice confronto di un hash)(vedere https://it.wikipedia.org/wiki/Hashcash per similitudini sulla variabile del tempo)

La reputazione di ogni entità all'interno della rete, sia Agent che Client, subisce una variazione in quanto ciascun esito delle proof of work viene registrata nella blockchain e questo consente di avere una classifica di affidabilità delle entità quasi in tempo reale.