# Protocolli di Sicurezza nella Blockchain

# Introduzione

La sicurezza di una blockchain è il risultato di una **moltitudine di protocolli**

Infatti, l'esperienza reale ci mostra che i fallimenti di sicurezza si verificano a vari livelli:

- **bug nello strato di smart contract**,
- alle vulnerabilità nella **rete P2P**
- strategie di **governance poco chiare**

# Introduzione

Esamineremo

- **Sicurezza di rete** e protezione dalle minacce.
- **Sicurezza degli smart contract** e protocolli di auditing, testing e governance del codice.
- **Procedure di protezione dei dati**
- **Strumenti e protocolli di recovery e di emergenza**.
- **Protocolli di privacy** e advanced cryptography.

# Il Contesto della Sicurezza: Minacce e Ambiti d’Azione

1. **Livello di rete**: manipolazioni del routing, saturazione di banda (DDoS), attacchi Sybil/Eclipse, infiltrazioni di nodi malevoli.
2. **Livello di dati**: corruzione di blocchi, malfunzionamento di meccanismi di hash, eventuale rottura di schemi crittografici.
3. **Livello di smart contract**: bug di programmazione, exploit logici, reentrancy, overflow, funzioni di emergenza non protette.
4. **Livello di interazione umana**: phishing, ingegneria sociale, gestione scorretta delle chiavi da parte degli utenti o degli amministratori.
5. **Livello di governance**: processi di aggiornamento e di emergenza non trasparenti, rischi di fork indesiderati o collusioni tra attori potenti.
6. **Livello di interoperabilità**: attacchi su bridge cross-chain, oracoli, sidechain che possono minare la sicurezza globale.

# Sicurezza del Livello di Rete e Protezione P2P

# Possibili attacchi: Sybil

Un **Sybil Attack** si verifica quando un singolo attore crea **molteplici identità false** all'interno di una rete **peer-to-peer**, con lo scopo di:

1. Avere un'influenza sproporzionata sulle operazioni.
2. Manipolare i meccanismi di reputazione, voto o routing.
3. Distorcere metriche di affidabilità, saturare gli slot di connessione di alcuni nodi target o mascherare attività malevole dietro molte identità.

# Probabilità di infiltrazione

## Scenario:

- Abbiamo una **rete** con un totale di N nodi.
- Tra questi, A sono nodi controllati dall'attaccante (*Sybil*).
- Un nodo onesto decide di connettersi a **m** peer, scelti in modo casuale dall'insieme dei N nodi disponibili.

# Distribuzione Binomiale: Selezione con Rimpiazzo

La **distribuzione binomiale** fornisce la probabilità di ottenere esattamente **k** successi in **m** prove, quando ogni prova ha probabilità **p** di successo, in modo indipendente. Qui:

- **Successo** = la connessione va a un nodo Sybil.
- **p=A/N** (frazione di nodi malevoli rispetto al totale).

La formula per la probabilità di ottenere **k** successi in **m** prove è

$$P(X = k) = \binom{m}{k} p^k (1-p)^{m-k}$$

Nel nostro contesto

$$P(\text{k nodi malevoli}) = \binom{m}{k} \left(\frac{A}{N}\right)^k \left(\frac{N-A}{N}\right)^{m-k}.$$

# Distribuzione Ipergeometrica: Selezione senza Rimpiazzo

In questo caso, la probabilità di pescare **k** nodi Sybil è data dalla **distribuzione ipergeometrica**, perché si estraggono **m** elementi da un insieme di **N** senza rimpiazzo, e si vuole sapere con quale probabilità **k** di questi appartengono al sottoinsieme di **A** elementi “malevoli”.

$$P(\text{k nodi malevoli}) = \frac{\binom{A}{k}\binom{N-A}{m-k}}{\binom{N}{m}}.$$

$\binom{A}{k}$ → Numero di modi per scegliere $k$ nodi Sybil dal totale di $A$.

$\binom{N-A}{m-k}$ → Numero di modi per scegliere i rimanenti $m-k$ nodi onesti.

$\binom{N}{m}$ → Numero di modi totali di scegliere $m$ nodi da $N$.

# Conclusione

La **probabilità di infiltrazione** in un Sybil Attack si modella con una **distribuzione binomiale** (in caso di scelte con rimpiazzo e indipendenti) o **ipergeometrica** (senza rimpiazzo).

Più è grande la frazione **A/N** di nodi Sybil, più facilmente un nodo onesto verrà “circondato” da identità malevoli.

- **Se l’attaccante** riesce a controllare molte connessioni del nodo, può manipolare informazioni, preludendo a un **Eclipse Attack**.
- **Se la rete** non ha meccanismi di costo/identità, un aggressore motivato può creare decine di IP Sybil, alzando **A/N**.
- **Se la blockchain** usa PoW/PoS, creare nodi “vuoti” non incide sul consenso, ma può incidere sulla topologia e sugli attacchi di isolamento.

# Possibili attacchi: Eclipse

Un **Eclipse Attack** si ha quando un nodo (vittima) viene **completamente isolato** dal resto della rete e vede solo i messaggi provenienti da nodi controllati dall'attaccante. Il nodo è inconsapevole:

- **Riceve blocchi "falsi"** o transazioni manipolate.
- **Double-spending** locale: l'attaccante fa vedere alla vittima che una transazione è stata confermata, quando in realtà non compare nella catena reale.
- **Rimane all'oscuro della catena autentica**, con la possibilità di generare un fork isolato.

Questo attacco è particolarmente subdolo perché può colpire un nodo strategico (ad esempio, un exchange o un nodo di un validatore secondario), inducendolo a prendere decisioni finanziarie errate.

# Meccanismo di Base

1. **Fase di Pre-Attacco**: Sybil
2. **Connessione ai Peer**: Il nodo vittima, all'avvio o durante la rotazione peer, cerca nodi noti. L'attaccante risponde presentando i suoi Sybil come validi.
3. **Saturazione**: Se la vittima ha slot di connessione massimi, l'attaccante si accaparra tutti o quasi, in modo che la vittima non riceva blocchi o transazioni dal resto della rete.
4. **Inganno Continuo**: La vittima vede solo i blocchi e le transazioni fornite dagli IP dell'attaccante. L'attaccante può manipolare il ledger presentato alla vittima.

# DDoS (Distributed Denial of Service)

Un **DDoS** è un attacco informatico in cui **molti dispositivi compromessi**, coordinati da un attaccante, **inondano di traffico** un server, con l'obiettivo di **sovraccaricarne le risorse** e **rendere illegittimo l'accesso agli utenti legittimi**.

1. **Creazione di una botnet**
   - L'attaccante infetta molti computer o dispositivi IoT con malware.
   - Questi device infetti diventano "bot" sotto il suo controllo.
2. **Coordinazione**
   - L'attaccante invia comandi a tutti i bot della botnet.
   - I bot iniziano a inviare **richieste massicce** e simultanee verso il bersaglio (server, nodo blockchain, sito web).
3. **Saturazione**

   Il bersaglio riceve un volume di richieste superiore alla capacità di elaborazione o di banda di rete.
   - Gli utenti legittimi non riescono più a collegarsi o subiscono ritardi enormi, e il servizio diventa **inaccessibile**.

## Protezione Anti-Sybil e Anti-Eclipse

I protocolli di sicurezza a livello di rete includono:

- **Meccanismi di “peer discovery” controllata**: La blockchain limita il numero di connessioni che un nodo può instaurare contemporaneamente, e verifica la reputazione o l’IP. Alcune implementazioni implementano tabelle di routing, con soglie di connessioni per IP.
- **Uso di PoW/PoS per ridurre la convenienza del Sybil**: Nelle reti permissionless, l’economia del consenso fa sì che gestire numerosi nodi sia costoso (in termini di potenza di calcolo o stake).
- **Connessioni multiple e randomizzate**: Un nodo dovrebbe connettersi a peer selezionati casualmente e cambiare periodicamente i peer, rendendo più difficile a un attaccante controllare tutte le connessioni di un target.

# Difesa dai DDoS

I **DDoS (Distributed Denial of Service)** possono saturare la rete di nodi o colpire i nodi “chiave”. Contromisure:

- **Protezione a livello di IP**: Rate limiting, filtri anti-spoofing, blacklisting di IP noti.
- **Architetture decentrate**: Se la rete è abbastanza ampia e diffusa geograficamente, un DDoS su alcuni nodi non abbatte l’intero sistema.
- **Sistemi di onion routing** (come TOR o I2P): Alcune blockchain o progetti sperimentano canali onion per rendere più difficile l’individuazione e l’attacco ai nodi critici.

# Protezione On-chain dei Dati: Strutture e Pratiche

## Immutabilità e Strutture di Verifica

Il primo livello di protezione on-chain è l'**immutabilità** garantita dal linking dei blocchi tramite hash:

- **Funzione di hash resistente**: Se l'hash scelto (es. SHA-256) diventasse insicuro, si potrebbero generare blocchi "validi" con contenuti diversi, minando la catena. La sicurezza futura passa da indagini su SHA-3, BLAKE2/3 e altre funzioni post-quantum.
- **Alberi di Merkle**: Consentono di verificare singole transazioni senza scaricare l'intera blockchain (utenti "light client"). Tuttavia, la sicurezza di questa tecnica dipende ancora dalla resistenza dell'hash.
- **Merkle Proof** e **SPV** (Simplified Payment Verification): Protocollo usato, ad esempio, in Bitcoin per i light client. Uno dei potenziali attacchi è l'**SPV fraud proof**: se un nodo fornisce un albero di Merkle manipolato, il light client potrebbe accettare transazioni false se non ha meccanismi di cross-check con la rete.

## Protezione della Privacy

Sebbene le blockchain pubbliche siano trasparenti, alcuni protocolli cercano di proteggere la privacy degli utenti:

1. **Mixing e CoinJoin**: In Bitcoin, esistono servizi di mixing che aggregano input di più utenti, rendendo complesso tracciare i flussi.
2. **Confidential Transactions** e **Pedersen Commitment**: Alcune chain usano commitments per nascondere gli importi, pur mantenendo la validità delle transazioni.
3. **ZK-SNARK/zk-STARK**: Protocolli di zero-knowledge che permettono di validare transazioni e bilanci senza rivelare i dettagli

# Sicurezza degli Smart Contract e Procedure di Auditing

## Ciclo di Vita dello Smart Contract: Sviluppo e Deployment

1. **Design**: Utilizzo di pattern sicuri (Checks-Effects-Interactions, Access Control, Rate Limit).
2. **Implementazione**: Linguaggi come Solidity o Vyper su Ethereum e Chaincode in Go/Node.js per Fabric.
3. **Testing**:
   - **Test unitari** e di integrazione.
   - **Fuzz testing** (Mythril, Echidna) che iniettano input casuali per scovare bug.
   - **Simulatori** di attacco: tentano reentrancy, overflow, logiche di bypass.
4. **Audit**: Squadre di sicurezza esterne (Trail of Bits, OpenZeppelin, CertiK, Quantstamp) eseguono revisioni approfondite del codice.
5. **Deployment**: In reti permissionless, una volta rilasciato il contratto su mainnet, risulta immutabile