Simulació d'un Sistema Blockchain Segur amb API i Sockets

Imagina que estàs desenvolupant un sistema de blockchain simulat per gestionar transaccions de manera segura. La teva tasca consisteix en crear una API RESTful que permeti als usuaris realitzar transaccions entre ells i afegir-les a un llibre major (ledger). A més, el sistema utilitzarà Sockets per comunicar-se en temps real i actualitzar els usuaris sobre l'estat de la xarxa (nous blocs, transaccions, etc.).

Per garantir la seguretat de les transaccions i blocs, utilitzaràs tècniques de xifratge simètric, asimètric i híbrid. Cada transacció serà autenticada, xifrada i registrada de manera segura.

---

1. Creació de l'Usuari i Autenticació amb Hashing

La teva API ha de permetre la creació d'usuaris i l'autenticació. Com que estem parlant d'un sistema blockchain, la seguretat dels usuaris és fonamental.

- Implementa un punt final POST /users/register/:id per crear un nou usuari amb nom, correu electrònic i contrasenya.

- La contrasenya ha de ser xifrada amb hashing (utilitzant bcrypt o un altre algorisme segur) abans de ser emmagatzemada.

- Implementa un punt final POST /users/login per autenticar als usuaris amb el seu nom d'usuari i contrasenya (comparant amb el hash de la contrasenya).

- Utilitza un token JWT per autenticar els usuaris un cop inicïin sessió i permetre el seu accés a les transaccions i la creació de blocs.

Explica per què és important utilitzar hashing per a les contrasenyes i la seva seguretat.

---

2. Gestió de Transaccions amb Xifratge Simètric

Els usuaris podran realitzar transaccions entre ells a través de la plataforma. Aquestes transaccions s'han de xifrar abans de ser emmagatzemades en el llibre major (ledger).

- Implementa un punt final POST /transactions/:id que permeti als usuaris enviar transaccions (quantitat de diners, destinatari, etc.).

- Cada transacció ha de ser xifrada utilitzant AES abans d'emmagatzemar-la en el llibre major. La clau secreta per al xifratge ha de ser generada dinàmicament i única per a cada transacció.

- El sistema ha de verificar que les transaccions són vàlides abans de procedir a afegir-les al blockchain.

Explica com utilitzaràs AES per a garantir la seguretat de les transaccions i com protegiràs la clau secreta utilitzada per al xifratge.

---

3. Creació i Validació de Blocs amb Xifratge Asimètric

Cada transacció es registra en un bloc. Els blocs han de ser xifrats i verificats abans de ser afegits al blockchain per garantir la integritat de les dades.

- Quan un bloc és creat, utilitza un sistema de xifratge asimètric (per exemple, RSA) per signar el bloc amb la clau privada del servidor, permetent que qualsevol usuari validi la integritat del bloc utilitzant la clau pública.

- Implementa un punt final POST /blocks/:id per crear un nou bloc amb les transaccions vàlides.

- Els blocs afegits al blockchain han de ser enviats a tots els usuaris a través de Sockets, per garantir que tots els nodes de la xarxa estiguin sincronitzats amb la informació més recent.

Explica com funciona el xifratge asimètric i com es verifica la integritat del bloc utilitzant la signatura digital del bloc.

---

4. Xifratge Híbrid per a la Comunicació Segura de Transaccions

Per millorar la seguretat i l'eficiència del sistema, implementa un sistema de xifratge híbrid que utilitza RSA per intercanviar la clau simètrica AES per a la xifratge de transaccions.

- Cada vegada que es creï una nova transacció, el sistema utilitzarà RSA per xifrar una clau simètrica AES i després utilitzarà aquesta clau per xifrar la transacció abans

d'enviar-la.

- Un cop rebut, el sistema utilitzarà la clau privada RSA per desxifrar la clau AES i després utilitzarà aquesta clau simètrica per desxifrar la transacció.

Explica quins avantatges ofereix el xifratge híbrid i com millora l'eficiència i la seguretat en el sistema blockchain.

---

5. Sincronització de la Xarxa mitjançant Sockets

Per mantenir tots els usuaris informats sobre l'estat actual del sistema (nous blocs, transaccions, etc.), utilitza Sockets per comunicar les actualitzacions en temps real.

- Quan es creï un bloc o es faci una nova transacció, tots els usuaris connectats hauran de rebre una notificació en temps real mitjançant Sockets.

- Implementa una comunicació bidireccional que permeti enviar i rebre informació entre el servidor i els usuaris (com notificacions de nous blocs o transaccions).

Explica com es gestionaran les connexions de Sockets i com garantiràs que la comunicació entre els nodes de la xarxa sigui segura.

6. Auditoria de la Xarxa Blockchain i Protecció de l'API

Cada acció del sistema ha de ser registrada de manera segura per permetre una auditoria detallada.

- Registra totes les transaccions, creació de blocs i altres accions crítiques en un log de seguretat.

- Protegeix l'API contra atacs comuns com SQL Injection, XSS, i CSRF.

- Fes servir pràctiques de seguretat com consultes preparades i tokens anti-CSRF per protegir l'API.

Explica com dissenyaries el sistema d'auditoria i quines tècniques utilitzaràs per garantir la seguretat de la plataforma.