ALGORITMOS Y ESTRUCTURA DE DATOS

TP Integrador Bitcoin Fase III.
Hashing.

Requerimiento

El objetivo de esta fase es interiorizarse con los conceptos de hashing, encriptación y firmas digitales aplicados al trabajo de Bitcoins. Se deberá implementar sobre las fases I y II .

Funcionamiento del programa

Basándonos en la funcionalidad de las fases anteriores, ampliaremos las opciones de forma que ahora el usuario en lugar de crear solamente nodos Full y SPV, puede crear también nodos mineros.

Los nodos Mineros:

1. Realizan todas las acciones de los nodos full (se consideran un nodo full).
2. Pueden minar según se explica en el apartado **Proceso de minado**. A partir de esta fase, la única forma de crear nodos en la blockchain es a partir de minarlos o inyección de nodos falsos como se explica más adelante.

Modificaremos los nodos full de forma que ahora:

1. Al recibir una transacción deberán validarla. Si no fuera válida la descartarán. Se deberá validar según se explica en el apartado **Firmas digitales.**
2. No podrán crear más bloques; solamente transmitirlos.
3. Una vez que reciben un bloque deberán validarlo y solamente lo agregarán en la blockchain luego de haberlo validado.

Modificaremos todos los IDs de las transacciones y bloques de forma que ahora los IDs constituyen el SHA256 de lo que se está caracterizando según lo que se especifica en el apartado **Hashing**.

Además de agregarle la opción de poder crear nodos mineros le agregaremos las siguientes opciones a la GUI:

1. Podremos enviar transferencias falsas (usando UTXOs ya usados) para ver como otros nodos la rechazan.

2. Podremos enviar bloques falsos para ver como los demás nodos lo rechazan 3. Cuando un nodo rechaza una transacción o un bloque deberá indicárselo de alguna forma en la GUI al usuario para que este entienda lo que está sucediendo.

Proceso de Minado

Cada nodo minero intentará minar un bloque con las transacciones que haya recibido y validado previamente. Para minar un bloque se debe realizar un hash del header del mismo (considerando la estructura del header del bloque como el "contenido" a hashear) y verificar que el resultado del hash cumpla con la regla de tener N bits en cero al principio (la cantidad de bits en cero se ajusta dinámicamente de forma de que en promedio se logre hashear un bloque cada un minuto). Este concepto se conoce como "challenge", del inglés "desafío".

Si no se altera el bloque de ninguna forma entonces el hash resultará siempre en el mismo valor. Para evitar esto, cada bloque incluye un campo de 16 bits conocido como nonce en el cual el miniero prueba distintos valores hasta lograr la condición mencionada arriba. Se conoce como Proof of Work (PoW) al nonce que logra resolver dicha regla.

La forma más eficiente de obtener el PoW es probar con valores aleatorios en el nonce hasta lograrlo (teniendo en cuenta repeticiones y dejándolas de lado).

Cuando un minero logra minar un bloque (resolver el "challenge"), se notificará al usuario.

Cuando un bloque minado en la posición n de la blockchain llega a un nodo minero (distinto del que lo minó claro está) y se verifica que fue minado correctamente, este dejará de intentar minar un bloque en esa posición y borrará de su registro de transacciones aquellas que estén en el bloque recibido. Entonces comenzará a minar un nuevo bloque n+1 con las transacciones que continúen en el registro. Cabe resaltar que durante el tiempo en que cada minero intentó minar el bloque n, estos continuaron recibiendo transacciones las cuales agregaron a su registro y que intentarán agregar luego al n+1.

Se le deberá poder ordenar a un nodo que comunique una transacción falsa, este la hará y los demás nodos la ignorarán y no será agregada a la Blockchain por ningún minero.

Firmas digitales

Una transacción contendrá entradas y salidas. Las entradas deben ser una referencia a las UTXO del usuario que paga y la firma para validar el pago. Las salidas deberán tener la cantidad de EDAcoins a pagar, y la public key del usuario que recibe el pago. Se utilizará la función SHA256, al igual que en la red de Bitcoins, para realizar el hashing de cualquier bloque de bytes que sea necesario y el sistema de firma digital ECDSA(Eliptic Curve Digital Signature Algorithm) para las firmas a través de public y private keys. Las implementaciones de los algoritmos se harán con la librería Cryptopp (https://www.cryptopp.com).

La firma privada será implementada como un objeto ECDSA<ECP,SHA256>::PrivateKey de Cryptopp. La clave pública se obtendrá a partir de la clave privada utilizando la función miembro MakePublicKey() de la clase PrivateKey.

Recuerden que cada Output deberá contar con la clave pública que identifica al nodo “dueño” de los EDACoins que el mismo posee.

Finalmente, para firmar una entrada (que indica que se está “gastando” un UTXO), se deberá crear un objeto de tipo ECDSA<ECP, SHA256>::Signer con la private key y utilizar la función miembro SignMessage() pasándole entre sus parámetros el mensaje a firma (la estructura que represente un input íntegra) de esta forma cada input tendrá una firma distinta, pero todas verificarán contra la publicKey de su dueño. Para verificar una firma, cada nodo al recibir una transacción utilizará la función miembro VerifyMessage() de la clase ECDSA<ECP, SHA256>::Verifier.

Hashing

Los IDs de nodos, transacciones y bloques que hasta ahora se calculaban con la función generateID() deberán ser reemplazados por sus correspondientes Hashes. El Hash es un valor de 32 bytes que se calcula con la función de la librería Cryptopp. Además todos los nodos al recibir una transacción deberán realizar las validaciones criptográficas antes de guardarla en su lista de transacciones. Una transacción es válida cuando:

A. El HashID debe verificar (el HashID de la Tx debe ser igual al calculado por el nodo al recibirla).
B. La UTXO referenciada en el Input Transaction de la Tx debe pertenecer al arreglo de UTXOs o a las transacciones pendientes (esta verificación solo se hace en el caso de los nodos Full. Los SPV siguen verificando en su lugar el Merkle Path).
C. La suma de los montos de EDACoin de los UTXOs referenciados en los Input Transactions tiene que coincidir con la suma de los montos de EDACoin referenciados en los Output Transactions.
D. Los unlocking scripts referidos en cada Input Transaction deben efectivamente desbloquear los UTXO referidos en cada una de ellas (es decir la firma debe poder ser validada con la publicKey).

Un bloque (para los nodos Full) o un block header (para los nodos SPV) es válido cuando:

A. Se verifica que cumple con el challenge.
B. El previous block hash coincide con el block hash del bloque anterior.
C. Todas las transacciones son válidas (sólo verificable por los Full y no por los SPV).