Clase 2 - Ethereum

jueves, 3 de octubre de 2024 11:36

Usa el algoritmo de hash Keccak-256
Usa criptografía asimétricq--> tiene una clave pública

Generación de claves
Es muy parecida a la de bitcoin:

1. Genera frase semilla
2. Transforma la semilla en una semilla binaria
3. Usando la semilla binaria, se genera con BIP32 una clae privada y su cadena de derivación. Bitcoin también usa bip32 pero Ethereum no usa la jerarquía compleja de claves del mismo (ni puta idea de que es esto, me da fiaca revisar la otra diapo). Directamente deriva claves individuales desde la privada maestra.
4. Se usa secp256k1 para conseguir clave pública idem bitcoin
5. La dirección ethereum se deriv de la clave pública, se aplica un el hash se usan los últimos 20 bytes del resultado. Esto se muestra en hexa y se usa para recibir fondos e identificar cuentas.

Antes usaba Pow como bitcoin pero cambió a PoS en 2022. **POS--> prueba de participación**

- Pos usa validadores en lugar de mineros. Los validadores bloquean (gambling, gambling GAMBLING) ETH como garantía.
- Se elige aleatoriamente según varios factores qué validador es el ue se pickea para proponer un bloque
- Una vez que el bloque se valida x veces se finalia y se queda en la cadena
- Se recompena a los validadores que proponen y verifican bien bloques, se los penaliza si no están cuando se los neccesita o actuan maliciosamente.

Smart contract es un autoejecutable que se depliega y ejecuta en la bchain. Ejecutan lógica de negocio (no sé concretamente qué sería esto) cuando se dan x condiciones.

- Inmutable: cuando se despliega no se puede modigicar
- Transparente: el código es público
- Seguro: se ejecutan en sandbox, no pueden afectar directamente a otras apps o partes del sistema fuera de su contexto (no entendí)
- Descentralizado: x lo que es la blockchain en sí
- Interconectado: pueden interactuar smart contracts entre sí
- Gas y tarifas: ejecutar un smart contract requiere gas, una medida de trabajo computacional. Usuaros pagan tarigas en ether para compensar a los validadores por ejecutar y validar las transacciones y contratos
- Se codean en solidity o vyper que se compilan para ejecutqarse en la evm (máquina virtual e ethereum)

Definiciones de cositas:

- Nonce: cant de transacciones mandadas desde una cueta. Se usa para ordenarlas y asegurarse que cada transacción solo se haga una vez y en orden.
- Gas: cant de trabajo computacional necesario para hacer cositas
- Gas limit max cant que va a usar una transacción. Si usa más que el máximo se detiene pero el gas usado no se reembolsa. Cada bloque tiene un gas limit
- Wei: unidad más chikita de ether.
- Gas price: cant de eth que un remitenete estpa dispuesto a pagar por cda unidad de gas al

hacer una transacción. La unidad de medida es el gwei

Tipos de cuenta en EVM

- Cuentas de propietario externas(EOA)
  - Controladas por claves privadas, sin código asociado a ellas
  - Wallet de usuarios. Desde las mismas mamndan transacciones de ether, interactuan con contratos o crean nuevos.
  - Tienen saldo en ether
- Cuentas de contrato(Contract account
  - Controladas por código de contrato. Se activan por una eoa o por otros contratos mediante mensajes
  - Cada cuenta tiene un código asoicado que se ejecuta cuando se recibe un mensaje en la cuenta
  - También tienen saldo en ether
  - Almacenamiento persistente: se almacenan y recuperan datos entre llamadas o transacciones
  - Creación de cuenta de contrat se da cuando un contrato se despliega

Address de contrato:
Se genera deterministicamnte a partir de la dirección de la cuenta que crea el contrato y el nonce de esa cuenta.

1. Se crea una transacciónn: transacción especial donde el destino está vacio y los datos tienen el código del contrato
2. Se genera la dirección dedl contrato mediante el hash de la direción del creador del contrato eoa y su nonce. Se toman los últimos 20 bytes del hash y eso es la direc del contrato.
3. Se despliega el contrato. Cuando se confirma la transacción y se agrega a un bloque, se despliega y la dirección se puede usar para interactuar con él.

Como el proceso es determinístico si sé la direc del creador y su nonce puedo predecir la dirección del contrato

Gracias chatGPT por tanto, perdón por tan poco
Ejemplo sencillo
Imagina que tienes un contrato inteligente que dice:

Condición: Si Alice transfiere 1 Ether a Bob, entonces Bob debe enviar un producto a Alice.
Cuando Alice envía el Ether a la dirección del contrato inteligente, este verifica la transacción y, al confirmarla, automáticamente le ordena a Bob que envíe el producto a Alice. Todo esto ocurre sin que ninguna de las partes tenga que intervenir manualmente, y ambas pueden estar seguras de que el contrato se ejecutará según las condiciones establecidas.