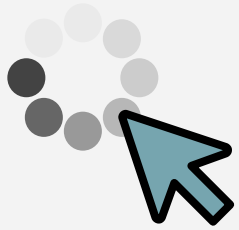
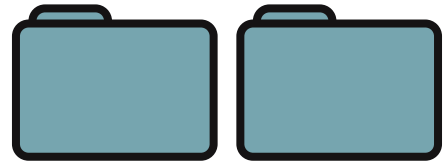
Mining Pool

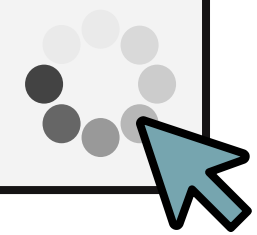
Franco Selleski





Roadmap

- ¿Qué es Bitcoin?
- ¿Qué significa minar Bitcoin? ¿Por qué un pool?
- Protocolo de mining pool



Bitcoin

¿Qué es Bitcoin?

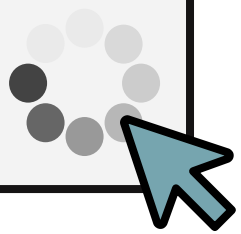
Una libreta de transacciones.

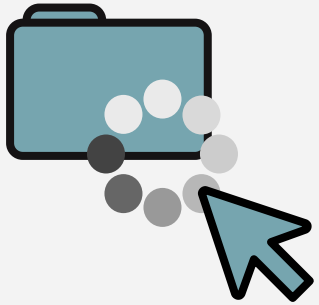
Franco mando 10 BTC a Daniel. Firma: 0445384...

Franco mando 10 BTC a Luca. Firma: 8987574...

Daniel mando 5 BTC a Emilia. Firma: 3521471...

ECDSA





Características principales

Descentralizado

No hay autoridad central

Nodos distribuidos en todo el mundo, con copias de la libreta

Nuevo bloque brodcasteado a todas las libretas

Anónimo

No hay ID, solo direcciones de wallets

Direcciones = clave pública

Acceso = clave privada

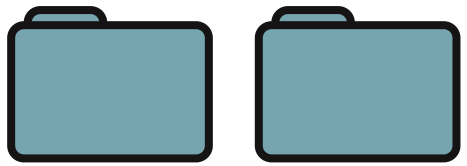
Trustless

No requiere de confianza entre partes

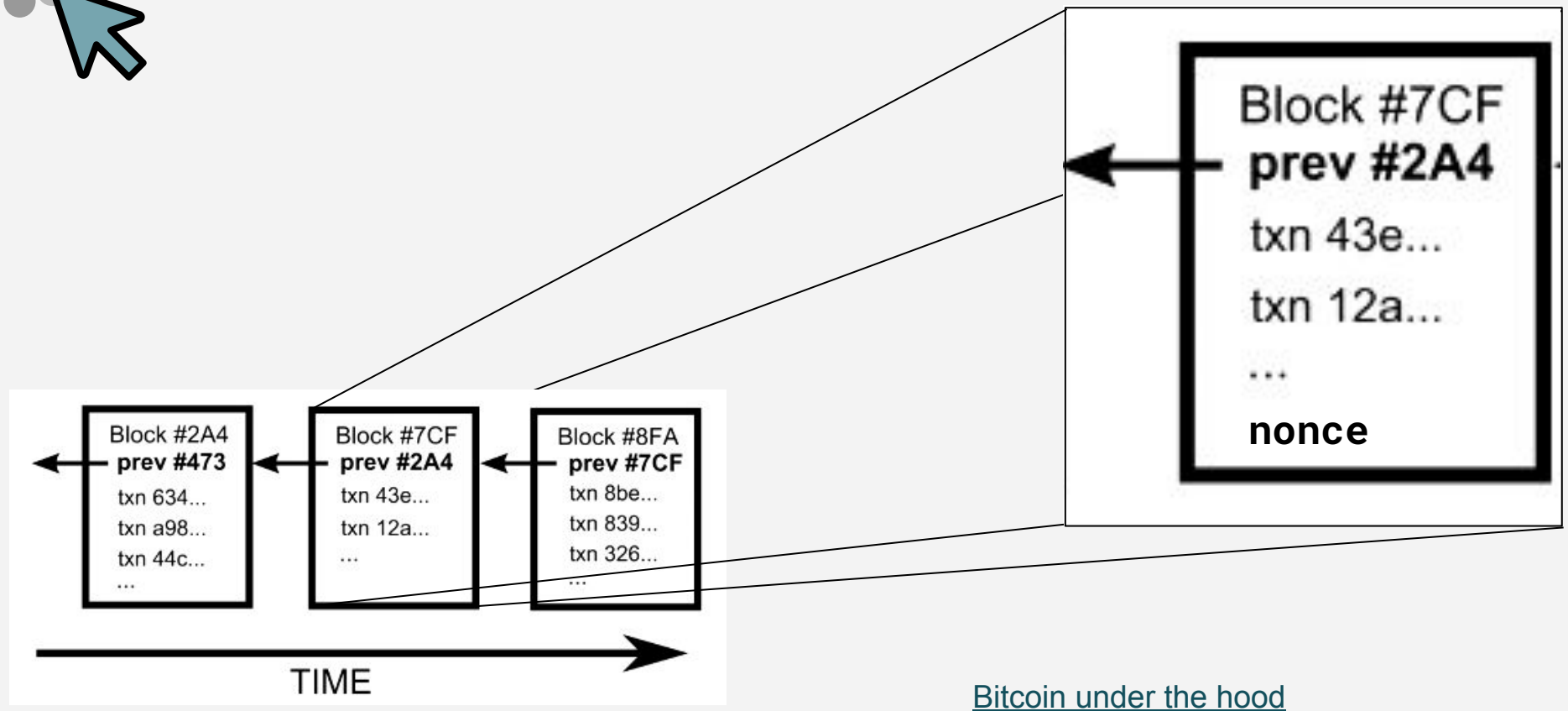
Confianza viene de “proof of work”

Resolución de puzzle criptográfico

¿Cómo se organiza esto?

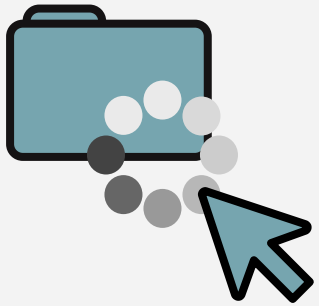


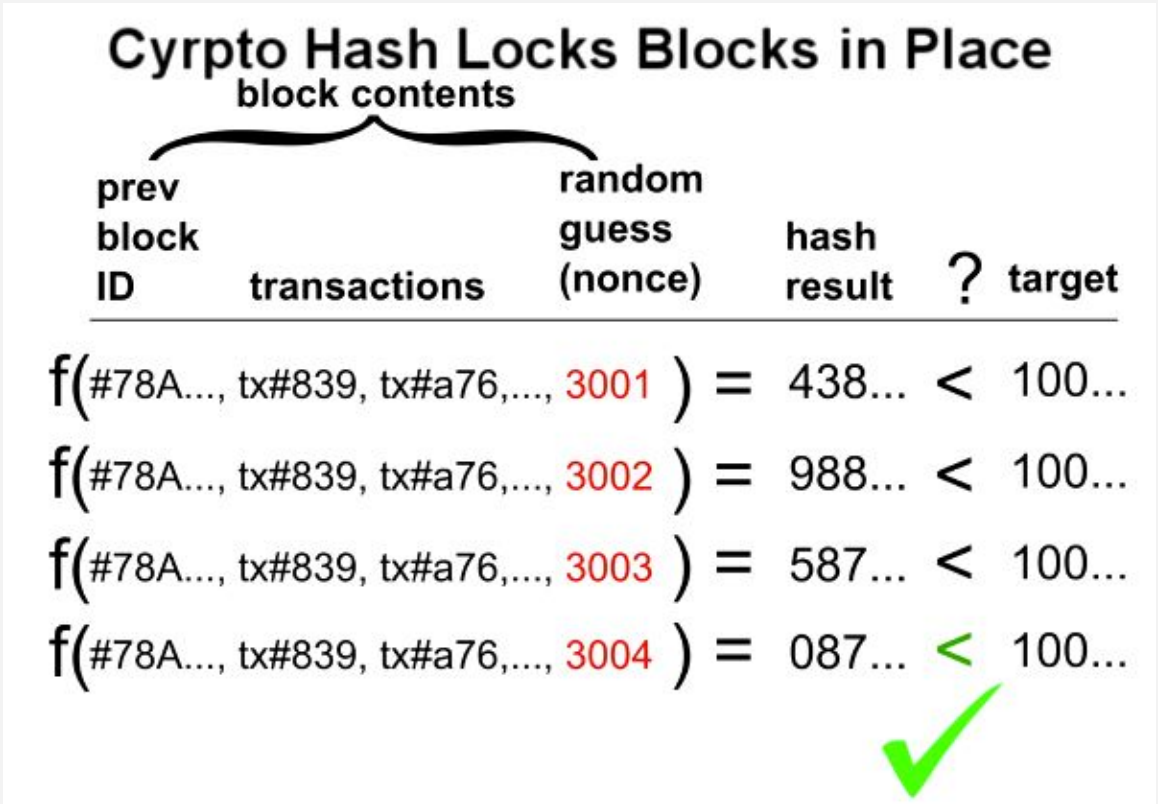
Blockchain



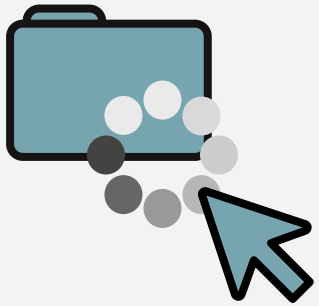
Bitcoin under the hood

¿Qué es minar?





- Encontrar un nonce = Ganar Bitcoins
- Dificultad adaptable, promedio 10 mins
- Recompensas de bloque decrecientes Max.Supply = 21M BTC



¿Por qué mining pools?

Probabilidad muy baja de acertarle al nonce

Idea: juntarse y probar entre varios

Problema 1: ¿Cómo hacerlo?

Problema 2: ¿Cómo repartir la ganancia?

Solución a 1:

Dividir el nonce en secciones

Un pool operator arma el bloque y reparte rangos de nonce

Cada minero prueba en el rango asignado

Solución a 2: Shares

Cuando el hash < pool target, enviar el nonce

Pago en función del # shares enviados

Mining Protocol

```
Miner_t{
   uint64_t MinerID,
   Range range,
   int Working,
   int Sockfd,
   Server* srv
}
Block_t{
uint64_t prev_hash,
uint64_t tx,
uint64_t nonce
}
```

```
Server_t{
   Miner_t miners[MAX_MINERS],
   Block_t block,
   uint8_t target,
   uint8_t diff,
   int Sockfd,
   int Running,
   Pthread_mutex_t Lock,
   uint64_t Top_Range,
   uint64_t Prev_hash
}
```

Mining Protocol

```
Header{
      uint32_t Payload_Size8,
      uint32_t Msg_type
}
MSG{
    Header HDR,
    union{
        Login login,
        Job_Req job_req,
        Sub_share sub_share,
        Job_Resp job_resp,
        Job_End job_end,
    } Payload
}
```

```
 Sub_share{
    uint64_t id;
    uint64_t nonce;
}
 Job_Resp{
    Block_t block;
    uint8_t target;
    Range range;
}
```

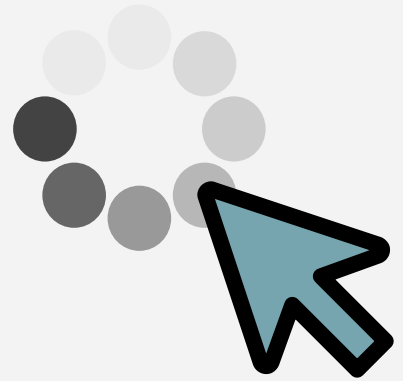
- Funciones de set y get
- Funciones de sendMsg y recvMsg

Mining Protocol

```
RunServer(Server *srv){
 ...
   Acepta la conexión
   Guarda el minero
   pthread_create(RunMiner, (void*)pm)

}
RunMiner{
while(1){
        recvd = recvMsg(miner->sockfd,
&msg);
switch (msg.HDR.msg_type){
   ...
      }
}
```

Demo



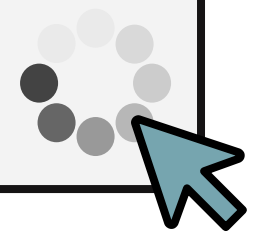
Conclusiones

Implementación protocolo mining pool

Simulación de generación y minado de bloques

Servidor y cliente conectados mediante sockets TCP

Txs realistas, broadcasting del bloque, Pay-Per-Share



Muchas gracias
&Preguntas?