UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

ciie
Centro de Investigación
e Innovación Educativa

Direcciones y billeteras Bitcoin

4.1 Direcciones Bitcoin

Direcciones Bitcoin

Una dirección de Bitcoin, es un identificador de 26-35 caracteres alfanuméricos, comenzando con el número 1, 3 o ‘bc1’, que representa un posible destino para un pago de Bitcoin. Actualmente hay tres formatos de direcciones en uso:

1. P2PKH que comienzan con el número 1, por ejemplo: 1GofkAGY2G56xNQs5KFg7zqDwoaCiupdEd
2. El tipo P2SH comienza con el número 3, por ejemplo: 3Bs5dRUmz2eQiM3mRCszco3MyeeQt1JMAw
3. El tipo Bech32 comienza con bc1, por ejemplo: bc1qdj34499ap7l5hh4yxgtdy349lcmxh4xvvsx36r

Nota: No mandar BTC a estas direcciones.

P2PKH | Pay-to-pubkeyHash

Una dirección Bitcoin es solo un hash, por lo que el remitente no puede proporcionar una clave pública completa en scriptPubKey. Al utilizar monedas que se han enviado a una dirección Bitcoin, el destinatario proporciona tanto la firma como la clave pública. La secuencia de comandos verifica que la clave pública provista hace un hash al hash en scriptPubKey y, a continuación, también verifica la firma con la clave pública.

```
scriptPubKey: OP_DUP OP_HASH160 <pubKeyHash> OP_EQUALVERIFY OP_CHECKSIG
scriptSig: <sig> <pubKey>
```

P2SH | Pay-to-ScriptHash

Las direcciones P2SH se crearon con la motivación de trasladar "la responsabilidad" de proporcionar las condiciones para canjear una transacción del remitente de los fondos al redentor. Permiten que el remitente financie una transacción arbitraria, sin importar qué tan complicada sea, utilizando un hash de 20 bytes. Pay-to-script-hash proporciona un medio para transacciones complicadas. La especificación no establece limitaciones en el script, y por lo tanto, absolutamente cualquier contrato puede ser realizado usando estas direcciones.

```
scriptPubKey: OP_HASH160 <scriptHash> OP_EQUAL
scriptSig: ..signatures... <serialized script>
```

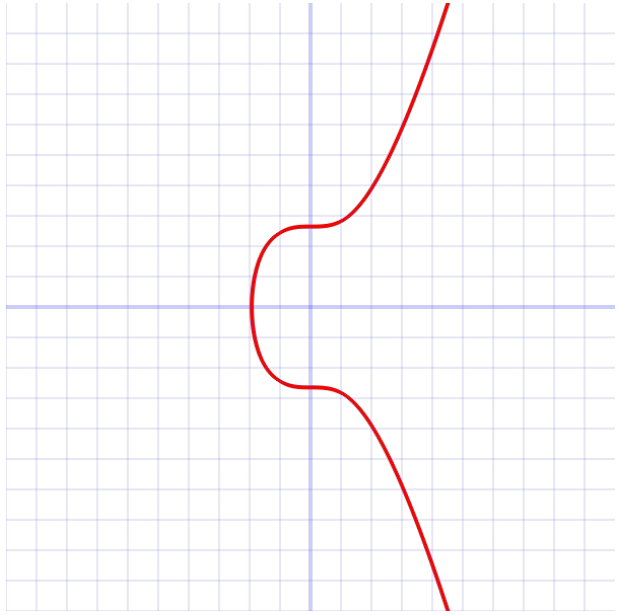
```
m-of-n multi-signature transaction:
scriptSig: 0 <sig1> ... <script>
script: OP_m <pubKey1> ... OP_n OP_CHECKMULTISIG
```

Bech32 | Segwit

Bech32 es un formato de dirección segwit especificado por BIP 0173. Este formato de dirección también se conoce como "direcciones bc1". Si bien este formato de dirección se ha incluido en algunas implementaciones, desde diciembre de 2017, el formato de dirección no se recomienda hasta que haya más software compatible con el formato.

Criptografía de curva elíptica

- Curva elíptica $y^2 = x^3 + 7$
- *Usa el estándar criptográfico eficiente secp256k1*
- $P = 2^{256} - 2^{32} - 2^9 - 2^8 - 2^7 - 2^6 - 2^4 - 1$
- *La multiplicación escalar era kG=P*



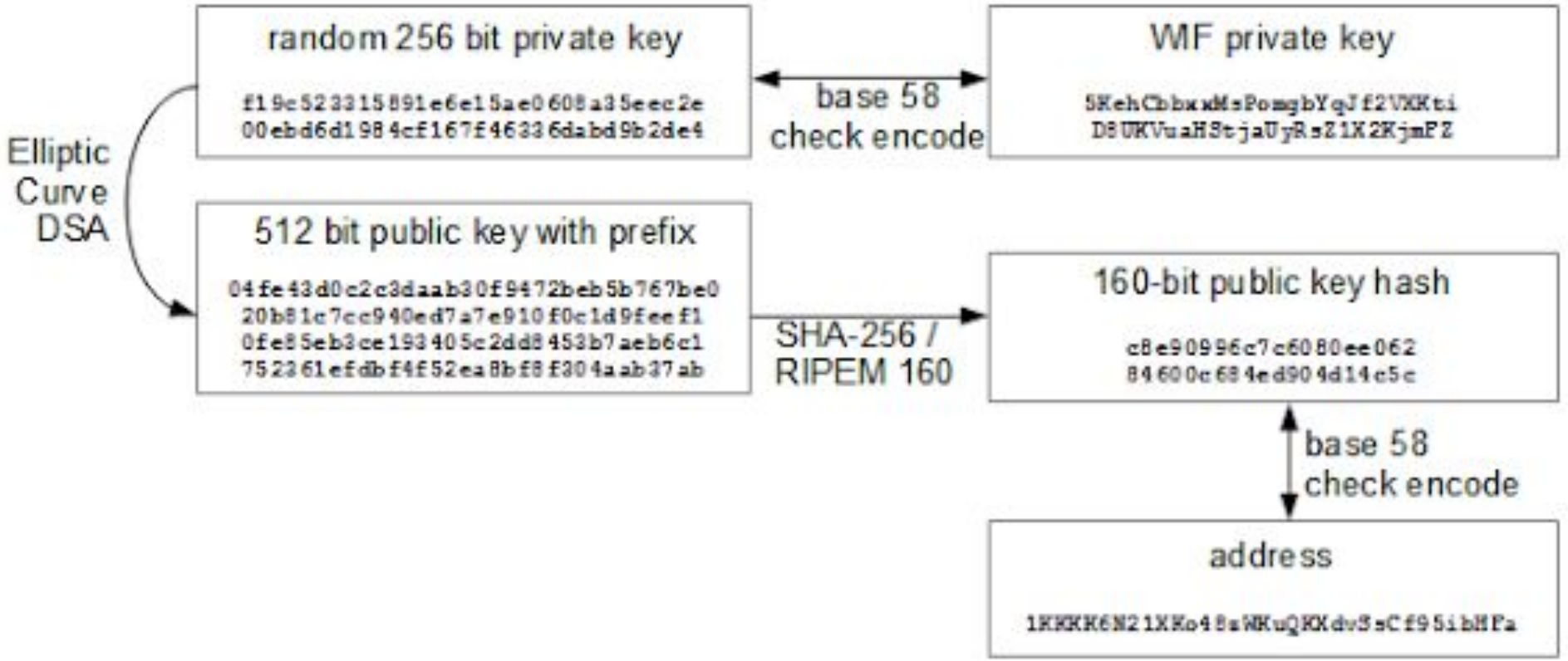
Criptografía de curva elíptica

Al aplicar la ECDSA a la clave privada, obtenemos un entero de 64 bytes. Consiste en dos enteros de 32 bytes que representan las coordenadas X e Y del punto en la curva elíptica concatenados juntos.

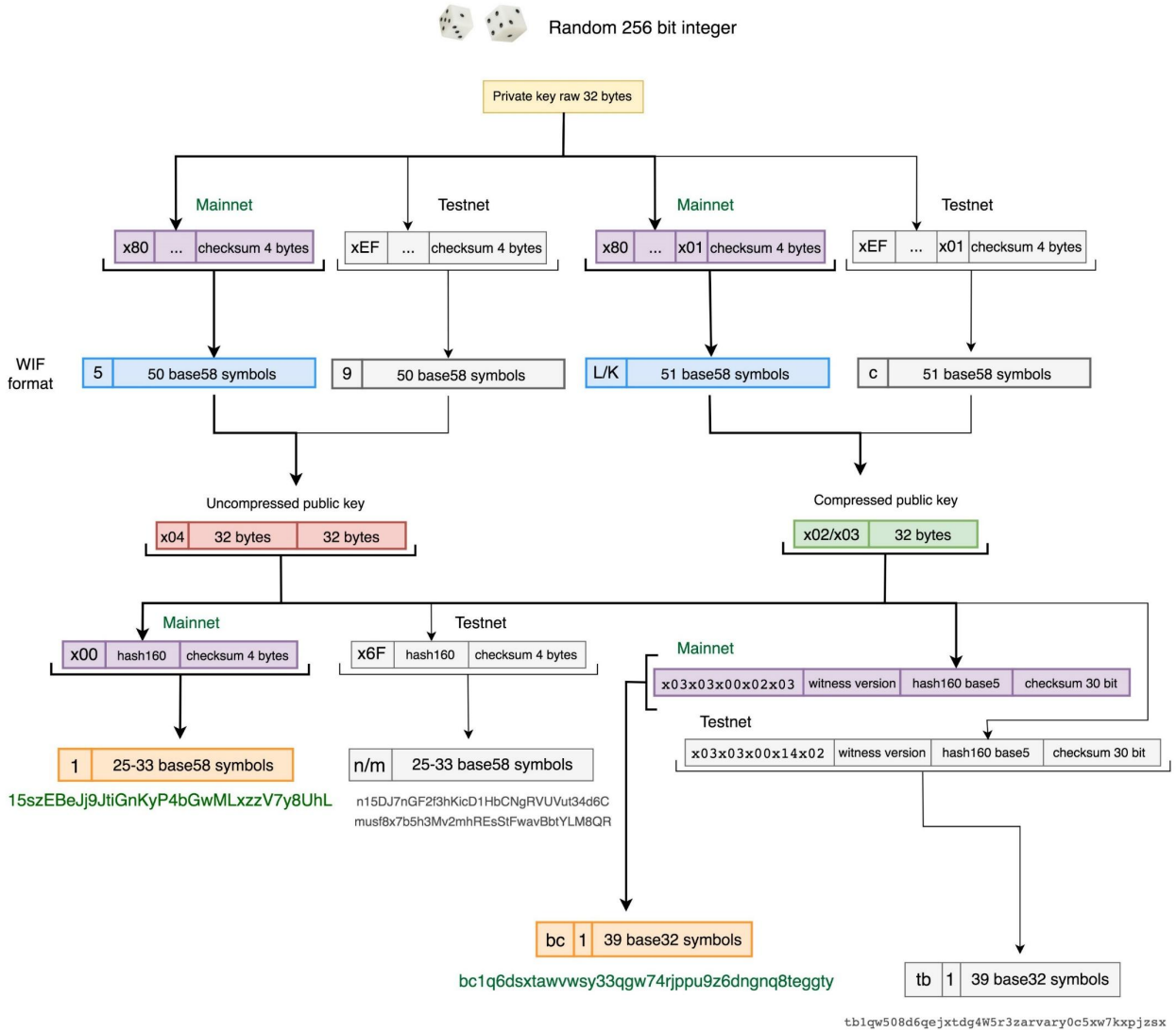
Por ejemplo:

“1e7bcc70c72770dbb72fea022e8a6d07f814d2ebe4de9ae3f7af75bf706902a7b73ff919898c836396a6b0c96812c3213b99372050853bd1678da0ead14487d7”

Claves Bitcoin



Mapa de direcciones Bitcoin



Clave pública sin comprimir

Una vez que usamos ECDSA, tenemos agregar los bytes 0x04 al inicio de nuestra clave pública. El resultado es una clave pública completa de Bitcoin.
Se ve así:

“041e7bcc70c72770dbb72fea022e8a6d07f814d2ebe4de9ae3f7af75bf706902a7b73ff919898c836396a6b0c96812c3213b99372050853bd1678da0ead14487d7”

Clave pública comprimida

La clave pública es un punto (X, Y) en la curva. Conocemos la curva, y para cada X solo hay dos Y que definen el punto que se encuentra en esa curva. Entonces, mantenemos X y el signo de Y. Los detalles son los siguientes: tomamos X de la clave pública de ECDSA. Ahora, agregamos el 0x02 si el último byte de Y es par, y el byte 0x03 si el último byte es impar:

“031e7bcc70c72770dbb72fea022e8a6d07f814d2ebe4de9ae3f7af75bf706902a7”

Encriptamos la clave pública

Lo que debemos hacer aquí es aplicar SHA-256 a la clave pública y luego aplicar RIPEMD-160 al resultado. El orden es importante:

“453233600a96384bb8d73d400984117ac84d7e8b”

Agregamos el byte de la red

Bitcoin tiene dos redes, la red principal y la de prueba. La red principal es la red en la que todas las personas transferimos Bitcoin. La red de prueba se creó, para probar nuevas funciones y software (tiene otras reglas de consenso, muchos la llaman testnet). Para usar la red Bitcoin se le agregan 0x00 bytes y el resultado es:

“00453233600a96384bb8d73d400984117ac84d7e8b”

Checksum

Calculamos la suma de comprobación de nuestra clave mainnet. La idea de la suma de comprobación es asegurarse de que los datos (en nuestro caso, la clave) no se hayan alterado durante la transmisión. Para calcular la suma de comprobación de la clave, debemos aplicar SHA-256 dos veces y luego tomar los primeros 4 bytes del resultado. Para nuestro ejemplo, el doble SHA-256 es: *“512f43c48517a75e58a7ec4c554ecd1a8f9603c891b46325006abf39c5c6b995”*

Y entonces la comprobación es *“512f43c4”* (4 bytes es 8 en hexadecimal)

Dirección Bitcoin

Finalmente, para obtener una dirección, simplemente concatenamos la clave mainnet y la suma de comprobación. Esto nos devuelve:
00453233600a96384bb8d73d400984117ac84d7e8b512f43c

Las direcciones Bitcoin están codificadas en Base 58 y por fin obtenemos la dirección de Bitcoin:

17JsmEygbbEUEpvt4PFtYaTeSqfb9ki1F1