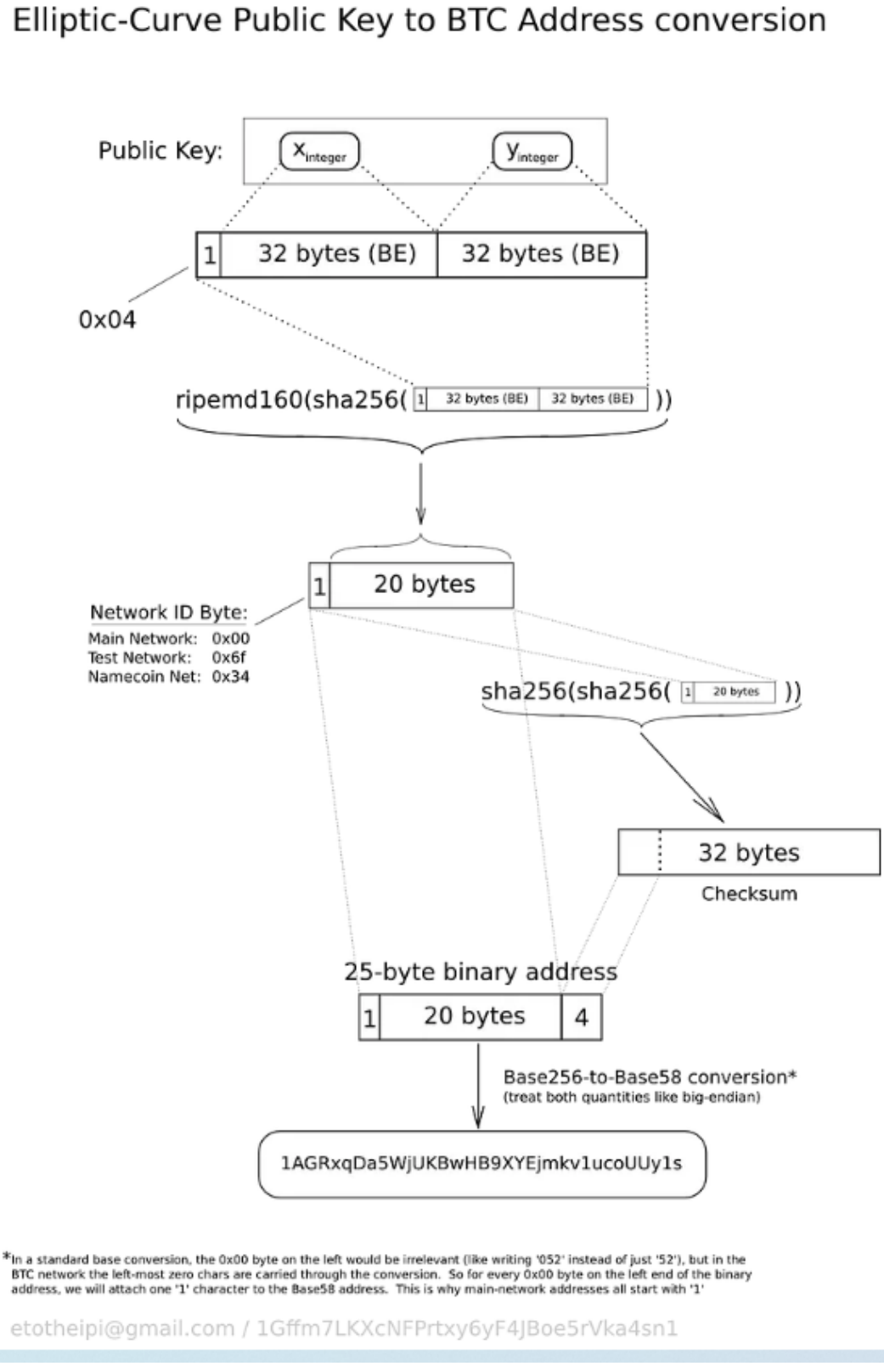
Ejemplo 2 ¿Qué hace Bitcoin?



*In a standard base conversion, the 0x00 byte on the left would be irrelevant (like writing '052' instead of just '52'), but in the BTC network the left-most zero chars are carried through the conversion. So for every 0x00 byte on the left end of the binary address, we will attach one '1' character to the Base58 address. This is why main-network addresses all start with '1'

Aplicación a un caso concreto

Un poco de contexto: La curva elíptica de bitcoin $\mathbf{y^2 = x^3 + 7}$ el punto **generador G** es el siguiente:

X=55066263022277343669578718895168534326250603453777594175500187360389116729240

Y=32670510020758816978083085130507043184471273380659243275938904335757337482424

Con este punto podemos Generar cualquier clave pública P_k multiplicando por la clave privada S_k

El punto obtenido al multiplicar el punto generador **G** por la clave privada es la clave pública la cual se suele representar en dos formatos diferentes, comprimida y descomprimida. En el ejercicio, cogeré la clave Privada S_k = 1, y esta la multiplicaré por **G**, obteniendo la clave Pública P_k

Es decir $P_k = G * S_k$ siendo: **P_k la clave Pública y S_k la clave Privada**

La clave pública descomprimida inicia con “04” seguido de la coordenada X en hexadecimal y luego la coordenada Y en formato hexadecimal, mientras que la clave pública comprimida inicia con “02” o “03” (si la coordenada Y es par inicia con “02”, en caso contrario inicia con “03”) seguido de la coordenada X en hexadecimal.

ESTO LO VEREMOS BIEN CUANDO ESTUDIEMOS LAS CURVAS ELÍPTICAS. AHORA QUEREMOS RESOLVER UN EJERCICIO DE HASHES TAL COMO LO HACE BITCOIN.

Inicio del ejercicio

Tenemos el punto o la coordenada (x,y) mod (P) de la curva con el módulo que utiliza Bitcoin, Estas coordenadas son el punto de partida **G** y lo multiplicamos por la clave Privada $\mathbf{S_k}$ **=1** Nos sale:

X=55066263022277343669578718895168534326250603453777594175500187360389116729240
Y=32670510020758816978083085130507043184471273380659243275938904335757337482424

Paso 1 OK

Convertimos X en hexadecimal *y añadimos el byte 02 al principio porque la Y es par (ya que acaba en 24). Al coger solo la coordenada X, estamos haciendo la clave pública reducida.*
El valor de la coordenada X en hexadecimal *es:*

0279BE667EF9DCBBAC55A06295CE870B07029BFCDB2DCE28D959F2815B16F81798

Hacemos ripemd160(sha256(del código en hexadecimal anterior). Primero sha256.

Hacemos el hash SHA256 de este valor(0279….1798). Nos queda:

0f715baf5d4c2ed329785cef29e562f73488c8a2bb9dbc5700b361d54b9b0554

Paso 2 OK

Se aplica la función de hash RIPEMD 160,

751e76e8199196d454941c45d1b3a323f1433bd6

Paso 3 OK

A la salida anterior se añade **al principio el byte 00** porque se trata de la red **Mainnet** o la red principal

00*751e76e8199196d454941c45d1b3a323f1433bd6*

Paso 4 OK

Para calcular el **checksum** (mecanismo de verificación para asegurarse que la dirección bitcoin está bien escrita) se aplica el algoritmo SHA-256 dos veces al resultado obtenido en el paso 3, se seleccionan los primeros 4 bytes del último hash SHA-256 los cuales representan el checksum de la dirección bitcoin.

Double SHA256 de (***00****751e76e8199196d454941c45d1b3a323f1433bd6)*

Double hash sale

510d1634d943109b69da527ef5948106f22b655fb5193b4e9ef7e4dcd342d245

El Checksum es igual a los primeros 4 bytes = 510d1634

Paso 5 OK-Se **concatena** la dirección obtenida en el paso 3 con el checksum obtenido en el paso 4.

00*751e76e8199196d454941c45d1b3a323f1433bd6510d1634*

Paso 6 OK-Se convierte el resultado obtenido utilizando la codificación Base58Check agregando un 1 a la dirección bitcoin.

1BgGZ9tcN4rm9KBzDn7KprQz87SZ26SAMH