Informe Final: "Monero, privacidad en la blockchain"

Pasquet Felipe Luc - LU 1084/22

Facultad de Ciencias Exactas y Naturales
2do cuatrimestre 2025

1. Introducción

El objetivo de este informe es analizar y explicar el paper "Monero, privacy in the blockchain". En este trabajo se estudian y explican las mejoras criptográficas que permiten ofuscar al remitente, al destinatario y los montos de las transacciones, manteniendo la verificabilidad pública de la red. A diferencia de Bitcoin, donde el grafo de transacciones es público, Monero utiliza primitivas avanzadas para proteger la privacidad del usuario por defecto.

2. Curvas Elipticas

Si bien las curvas elipticas se usan en Bitcoin y en la mayoría de blockchains, en el paper se dice que la criptografía de monero se basa en las curvas elipticas de *Twisted Edwards*, que son el conjunto de puntos (x, y) que satisfacen la ecuación:

$$ax^2 + y^2 = 1 + dx^2y^2 \tag{1}$$

Dados dos puntos de la curva $P_1 = (x_1, y_1)$ y $P_2 = (x_2, y_2)$, se define el producto $P_1 + P_2 = P_3$ como $P_3 = (x_3, y_3)$:

$$x_3 = \frac{x_1y_2 + y_1x_2}{1 + dx_1x_2y_1y_2} \ (mod\ p) \tag{2}$$

$$y_3 = \frac{y_1y_2 + ax_1x_2}{1 - dx_1x_2y_1y_2} \ (mod\ p) \tag{3}$$

Si P_1, P_2 pertenecen a la curva, entonces P_3 también pertenece a la curva.
De esta manera también, si multiplicamos un punto P de la curva por cualquier número $n \in \mathbb{N}$, vamos a conseguir otro punto perteneciente a la curva.
En general utilizamos un punto de la curva conocido G, llamado **generador**, el cual satisface que para todo punto de la curva P, existe un n tal que $nG = P$.

Calcular el producto escalar nP es relativamente sencillo. Sin embargo si nos dan dos puntos P_1, P_2 y nos piden encontrar n tal que $nP_1 = P_2$ es *computacionalmente difícil*. En otras palabras, la multiplicación escalar puede ser utilizada como una **One-Way Function**.

En este contexto de curvas elípticas, un par de **clave pública y clave privada** (K, k), consiste de un punto de la curva K y un número entero k, tal que $K = kG$.
Si conocemos la clave pública K es computacionalmente imposible calcular la clave privada k.

3. Ocultando al Remitente: Firmas de Anillo (LSAG)

Para evitar que se conozca quién inicia una transacción, Monero utiliza Firmas de Anillo (Ring Signatures). Una firma de anillo permite a un miembro de un grupo firmar un mensaje en nombre del grupo, sin revelar cuál de los miembros produjo la firma.

El protocolo construye un "anillo" compuesto por:

- La salida real que se está gastando (UTXO del remitente).
- Múltiples salidas ajenas tomadas de la blockchain (señuelos o *decoys*).

Para un observador externo, todas las entradas del anillo son equiprobables. La firma verifica que "alguien" del grupo autorizó el gasto, pero es computacionalmente imposible determinar quién.

Monero utiliza una variante de las firmas de anillo llamada LSAG (*Linkable Spontaneous Anonymous Group*). Supongamos un grupo de n claves públicas $P_0, P_1, \ldots, P_{n-1}$, donde el firmante posee la clave privada x_s correspondiente a P_s (siendo s un índice secreto).

La firma se construye para asegurar dos propiedades: el anonimato del firmante dentro del grupo y la no-reutilización de los fondos (linkability).

3.1. Generación de la Imagen de Clave (Key Image)

Antes de iniciar el anillo, el firmante calcula la imagen de clave $\tilde{K}$. Esta imagen es determinista respecto a la clave privada x_s y la clave pública P_s, pero computacionalmente es imposible revertirla para hallar x_s.

$$\tilde{K} = x_s \cdot \mathcal{H}_p(P_s) \tag{4}$$

Donde $\mathcal{H}_p$ es una función de hash que mapea un punto a la curva elíptica. Dado que $\tilde{K}$ es necesario para verificar la firma, si un usuario intenta gastar el mismo output P_s nuevamente, generará idéntico $\tilde{K}$, permitiendo a la red detectar y rechazar el doble gasto sin conocer la identidad del firmante (s).

3.2. Generación del Anillo y Desafíos

La firma consiste en un desafío inicial c_0 y una serie de respuestas r_i. Para validar la firma, los verificadores deben reconstruir una cadena de valores L_i y R_i para cada miembro del anillo $i \in \{0, \ldots, n-1\}$.

Las ecuaciones de verificación para cada paso son:

$$L_i = r_i G + c_i P_i \tag{5}$$

$$R_i = r_i \mathcal{H}_p(P_i) + c_i \tilde{K} \tag{6}$$

Los desafíos c_i se calculan de manera recursiva (en anillo). El desafío para el siguiente participante $i+1$ depende del resultado del participante anterior:

$$c_{i+1} = \mathcal{H}_s(m, L_i, R_i) \tag{7}$$

Donde m es el mensaje (hash de la transacción) y $\mathcal{H}_s$ es una función de hash escalar.

3.3. Cierre del Anillo (The Trapdoor)

Para los miembros que no son el firmante ($i \neq s$), los valores r_i y c_i se eligen aleatoriamente para simular la participación. Sin embargo, al llegar al índice del firmante real s (se empieza a calcular los desafíos desde el siguiente participante al firmante real, de manera que nosotros calculemos último el desafío del firmante real y podamos cerrar el anillo), este debe calcular un r_s específico que "cierre" el círculo, haciendo que el último desafío c_n sea igual al desafío inicial c_0 ($c_n \equiv c_0 \pmod{l}$).

El firmante resuelve la siguiente ecuación utilizando su clave privada x_s (la "trapdoor"):

$$r_s = \alpha - c_s \cdot x_s \pmod{l} \tag{8}$$

Donde α es un valor aleatorio elegido por el firmante al inicio. Esta capacidad de conectar el final del anillo con el principio es lo que prueba matemáticamente que uno de los miembros conoce la clave privada correspondiente, sin revelar cuál de ellos fue.

4. Protección de destinatario: *Dirección de un solo uso*

El mecanismo que se utiliza para poder ocultar al destinatario es creando un nuevo par de claves para el destinatario (una dirección de un solo uso), a través del protocolo de intercambio *Diffie-Hellman*, de manera que nadie sepa el destinatario real.

En Monero, cada usuario tiene dos pares de claves públicas y privadas. Un par se utiliza para saber si una dirección de un solo uso nos pertenece, y el otro par lo usamos para poder saber la clave privada de esa dirección de un solo uso y poder utilizarla.

El paper lo explica con el siguiente sencillo ejemplo:

Supongamos que Alice quiere hacer una simple transacción a Bob, con exactamente un input y un output. Bob tiene las claves privadas/publicas (k_{B_1}, k_{B_2}) and (K_{B_1}, K_{B_2}). Para crear las claves de un solo uso Alice haría lo siguiente:

1. Alice genera un número aleatorio r y calcula la clave pública de un solo uso $K_o = H_n(rK_{B_1})G + K_{B_2}$
2. Alice crea una transacción usando K_o como destinatario y le agrega a la datos de la transacción el valor rG, luego manda la transacción a la red. El valor rG será usado por Bob para saber si él es el destinatario real de la transacción.
3. Bob recibe los datos de la transacción y ve el valor rG. Con ese valor puede calcular $k_{B_1}rG = rk_{B_1}G = rK_{B_1}$. Luego puede calcular $K_o = H_n(rK_{B_1})G + K_{B_2}$ y saber si la transacción estaba dirigida a él.
4. Finalmente las *direcciones de un solo uso* del output de la transacción son:

$$K_o = H_n(rK_{B_1})G + K_{B_2} = H_n(rK_{B_1})G + k_{B_2}G = (H_n(rK_{B_1}) + k_{B_2})G \tag{9}$$

$$k_o = H_n(rK_{B_1}) + k_{B_2} \tag{10}$$

Podemos observar que de esta manera solo Alice y Bob saben que el destinatario de la transacción es Bob, pues es necesario conocer r (solo lo conoce Alice) o k_{B_1} (que solo lo conoce Bob); y solo Bob puede saber la clave privada del output k_o, pues es necesario conocer la clave privada de Bob k_{B_2}.

5. Ocultando el Monto: RingCT

Para ocultar las cantidades transferidas, Monero utiliza *Ring Confidential Transactions* (RingCT). En lugar de publicar los montos en texto plano, se utilizan **Compromisos de Pedersen**. Un compromiso C para un monto a se define como:

$$C = xG + aH \tag{11}$$

Donde x es un factor de ocultamiento aleatorio (máscara) y H es un punto base (Generador) diferente a G. Gracias a la propiedad homomórfica de estos compromisos, los mineros pueden verificar que no se ha creado dinero de la nada comprobando que la suma de los compromisos de entrada menos los de salida es igual a un compromiso de cero:

$$\sum C_{in} - \sum C_{out} = zG \tag{12}$$

Adicionalmente, se incluyen *Range Proofs* (Pruebas de Rango) para certificar criptográficamente que los montos ocultos a son números positivos dentro de un intervalo válido $[0, 2^{64}]$, evitando así la creación de dinero mediante desbordamiento negativo.

6. Conclusión

El protocolo de Monero demuestra que es posible construir una blockchain pública y descentralizada sin sacrificar la privacidad financiera. Mediante la combinación de Direcciones Furtivas (para el receptor), Firmas de Anillo con Imágenes de Clave (para el emisor) y RingCT (para los montos), se logra ofuscar todo el grafo de transacciones. Aunque esto conlleva un mayor tamaño de transacción en comparación con Bitcoin, ofrece fungibilidad real y resistencia a la censura.