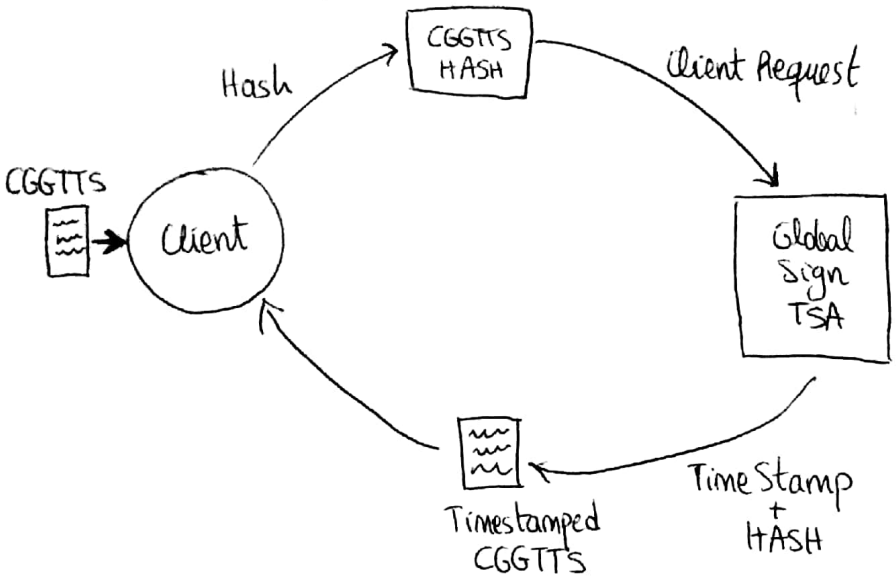
SECURITY TIMESTAMP

- Time Stamping Authority (TSA)

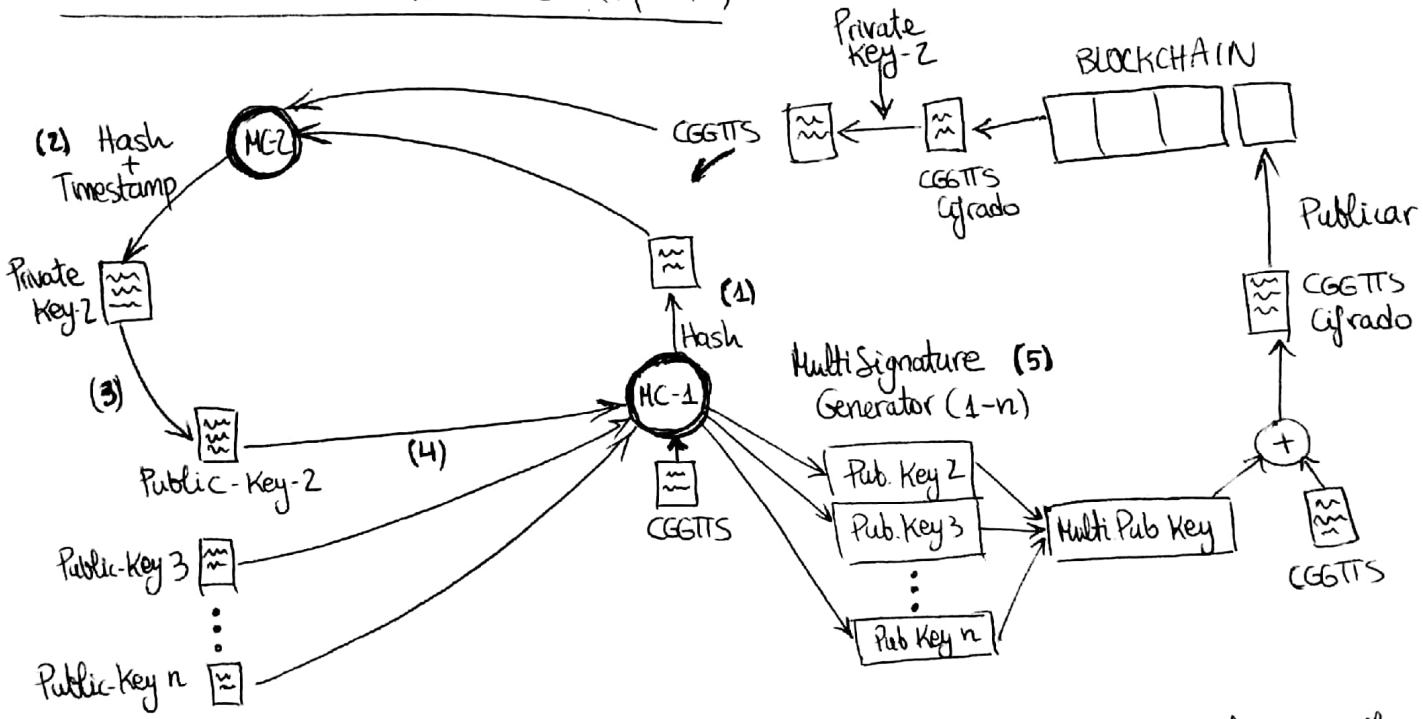


(*) Así es como se podría aplicar el sellado del tiempo regular.

Sin embargo, esto implicaría confiar en una TSA.

Centralizado ¿Costes?

TIMESTAMP NO-CENTRALIZADO (Aprox. 1)



(1) Master-Clock 1 realiza hash al CGGTTS que quiere publicar y lo envía a MC-2

(2) MC-2 Incluirá su Timestamp y hace HASH. Señalar que cuanto más sincronizados estén MC1 y MC2 y UTC más preciso será el Timestamp.

(3) El Hash resultante de (2) será su clave privada con la que genera la Pública. se relaciona el Timestamp con las claves para el NO-Repudio de MC-2 a ese Timestamp en caso de ser necesario

Quizá es necesario incluir otro valor que sólo conozca MC-2 en (2) Para que no puedan obtener su clave privada por ataque fuerza-bruta

(4) Tanto MC-2 como otros MCs envían claves publicas

(5) Se genera una clave publica 1-n, que podrá ser descifrada por cualquiera de los poseedores de las claves privadas relacionadas con las claves públicas con las que se genera (2...n)

Aclaraciones

a) El Master-Clock 2 tendrá disponible en Blockchain el fichero CGGTTS que podrá descifrar con su clave privada. Una vez descifrado podrá comprobar con el HASH la integridad del fichero desde el paso (1) y a su vez sabrá que esa integridad tiene como hora su TimeStamp.

b) El mismo fichero que obtenemos de Blockchain es el que descifran otros MCs, por lo tanto, se gana en fiabilidad.

c) En caso de necesitar más fiabilidad, se podría solucionar con Smart Contracts en la Blockchain. Un HASH y un TimeStamp (obtenido de la media de los TimeStamps de todos los MCs) se añadiría a la Blockchain. Cada MC podría modificar una variable que dice si considera esa info válida o no. En caso de que más del 75% no lo considerase válido el fichero no tendría validez.

d) Idéntico proceso se realizaría de manera simétrica para que el MC-2 publicase su fichero CGGTTS.

e) En caso de algún conflicto cada MC tendrá su fichero validado y con TimeStamp del resto de MCs disponible.
En caso de realizar un cruce de ficheros de trazabilidad para calibrar el reloj y mejorar la sincronización, todos los ficheros están disponibles y validados en la Blockchain.

Surge con esto mi mayor duda:

¿El objetivo principal de la Trazabilidad es tener el fichero como respaldo en caso de conflicto o el de la Calibración para afinar la sincronización entre dos MCs.?

TIMESTAMP NO-CENTRALIZADO (Aprox 2)

1) El MC que quiere publicar un CGGTIS comparte un HASH con el resto de MCs.
2) Publica el fichero en BLOCKCHAIN
3) Cada MC (una vez se ha añadido a la Blockchain) tiene la oportunidad de (en un SmartContract) votar sobre cuál es el TimeStamp que considera correcto para ese fichero.
4) El TimeStamp se obtiene de la media de todos los MCs que votan
5) No se tienen en cuenta TimeStamps que se alejen del cuartil 75 y 25 para evitar acciones malintencionadas.

TIMESTAMP NO-PRECISO

La librería de Solidity contiene funciones para obtener el block.timestamp. En caso de que no se necesite un TimeStamp preciso sería suficiente el proporcionado por Blockchain

Hay que tener en cuenta el tiempo que puede tardar en añadirse la info a la Blockchain y que una vez añadida, no podemos asegurar que el reloj del minero que añade el bloque sea preciso o coincida con el UTC.

Un minero no intentará falsear el timestamp del bloque que intenta minar, ya que entonces el resto de mineros no seguirían su cadena (o también tendrían que falsear su timestamp), ya que el timestamp de un bloque tiene que ser mayor al anterior obligatoriamente.

Se habla de que el margen que un minero puede "falsear" es de 900 segundos (15 minutos).