Proof of Reserve

Digital Assets - Week 7 (Pre-record)

Rhys Bidder
rhys.m.bidder@kcl.ac.uk

KBS, QCGBF

Autumn 2024

Disclaimer 1: Any opinions expressed, errors or omissions should be regarded as those of the author and not necessarily those of KCL, KBS, QCGBF, QCB, CBI, BoE, or ChainLink Labs.

Disclaimer 2: These notes on cryptography are heavily simplified and leave out many important details. For any real-world security applications these notes should not be relied upon. Instead you should consult appropriate security resources and reliable security professionals.

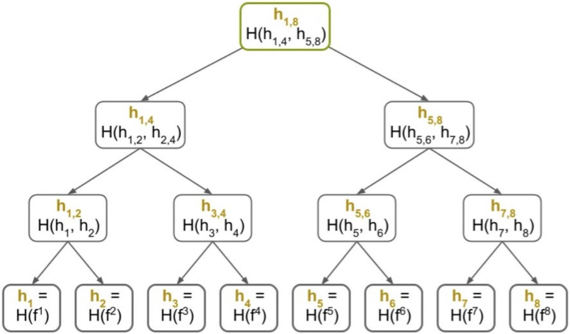
Disclaimer 3: Cryptoasset transactions are illegal in some jurisdictions. Do not violate these or any other laws. I am not promoting the use of crypto in countries where it is illegal in any form and these slides are not a promotion of crypto or an invitation to participate in crypto-related activities in such countries. They are purely for educational purposes.

Disclaimer 4: I currently am an advisor to Chainlink Labs.

Merkle trees

- In our introduction to cryptography and in our early lectures we discussed Merkle trees
- We emphasized that they were a useful structure for checking and reconciling data
- Summarizes all the transactions in a block in a particular way
- Minimizes work required (and even amount of data required) to verify if, say, a transaction has already occurred in the blockchain, or to check if there has been tampering/changes

Merkle trees



- Hashes of individual transactions (f^i) are 'leaves'
- Each intermediate parent node hashes a pair of lower nodes
- The parent hash at the top is the 'root'
- Any change in any transaction (or addition of a transaction) will change the root **in a way that is simple to detect**

Merkle trees - useful for blockchains

> *Consider the consequences if we were to replace Merkle roots with a single hash of all the transactions in a block. If you ever wanted to confirm the integrity of a transaction that occurred in a block (e.g., that it actually occurred in X block at position Y),* **with a single hash approach you would need to know and examine every transaction ID in the associated block** *(since the single layer of data underlying the hash includes all transaction IDs, which you would have to examine to confirm). This authentication process requires a lot of memory to be stored and transmitted across the network*
>
> *With Merkle trees, the verification process is more efficient and* **does not require downloading the data of all transactions that occurred in a block***. By just knowing the transaction ID (or leaf) in question, the Merkle root, and the 'branch' consisting of all of the hashes up the path from the leaf to the root, data can be verified. These Merkle 'proofs'* **allow for the efficient authentication of a small amount of data (like a single transaction in a block) within large databases** *of potentially unbounded size.*

- Griffith, G. (2022) - my emphasis added

Merkle trees and proof of reserves

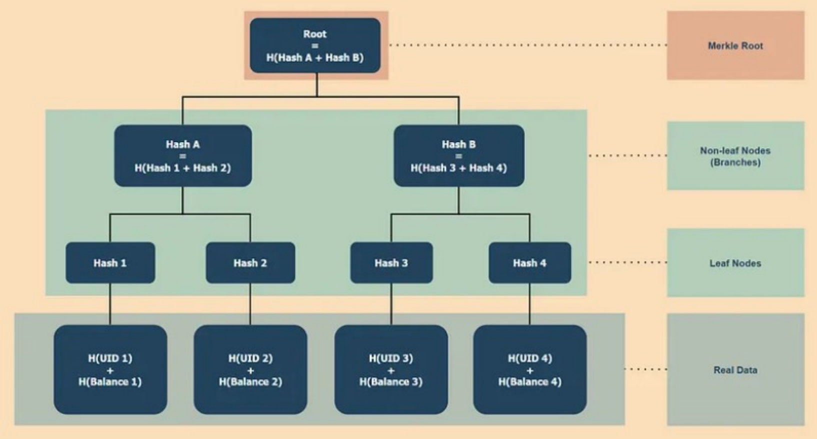
- Merkle trees typically underpin the technique known as ‘proof of reserves’ (PoR)
  - Proof of reserves is intended to show that claimed assets are in fact present
  - Thus can be redeemed/honored at any time
- Some obvious use cases arise:
  - CEX (especially after FTX debacle!)
  - Stablecoins
  - Tokenized assets
- PoR is not purely about Merkle proofs
  - Most effective if combined with old fashioned audit, blockchain monitoring
  - Not least because off-chain assets and liabilities may complicate matters

PoR for CEX

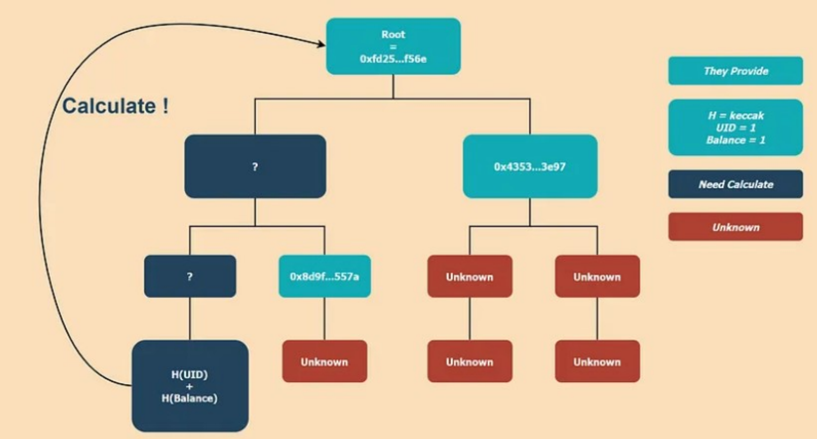
- FTX revealed some gaping holes in governance in the newly emerging CEX industry
  - Recall, CEXs take custody of client's tokens (not your keys, not your tokens)
  - Absent some other mechanism, you don't know and can't control what they are doing with your tokens
- They are not supposed to invest your money without explicit permission
  - They are not banks
  - They *should* be custodians and only enable explicitly authorized transactions (FTX didn't respect this)

PoR for CEX

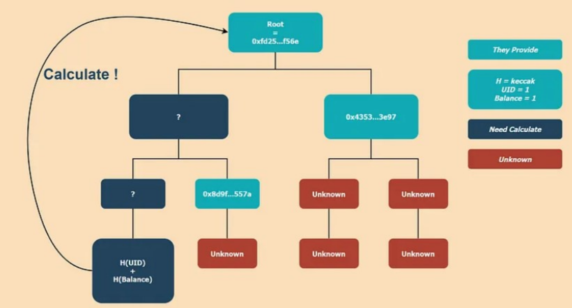
- PoR is supposed to show that the exchange has all the assets it claims to hold, on behalf of clients **and** that they are correctly associated with each particular client's account
  - A prominent example is Binance's PoR page (though you would need an account to see the report)
  - Emerging best practice is to pair self-certification (by clients) with a third-party audit (see also Kraken)
- However, not universally used - see Coinbase (an apparently respectable CEX)



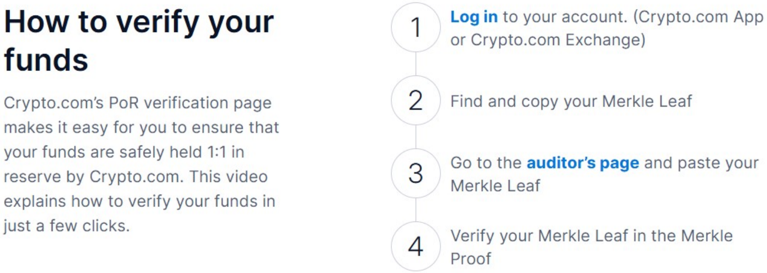
Merkle tree for Proof of Reserve. *Source:* Orbiter finance



Proof of Reserve - user calculation. *Source:* Orbiter finance



- User will be told:
  - Her own details (or knows them already): ID and balance
  - Root hash
  - A minimal number of other hashes (which do not reveal other users' info - recall properties of hashes. . .)
- More thorough details (emphasizing auditor) here and here



How to verify your funds

Crypto.com's PoR verification page makes it easy for you to ensure that your funds are safely held 1:1 in reserve by Crypto.com. This video explains how to verify your funds in just a few clicks.

1. Log in to your account. (Crypto.com App or Crypto.com Exchange)
2. Find and copy your Merkle Leaf
3. Go to the auditor's page and paste your Merkle Leaf
4. Verify your Merkle Leaf in the Merkle Proof

Crypto.com user interface for Proof of Reserve. *Source:* Crypto.com

Issues with PoR

- All else equal, better to have than not but far from sufficient
- Improved if a third party auditor (centralization?) can analyze all balances, dig into custody and standard accounting checks
- PoR need to be done very frequently (ideally close to continuous) as balances could be temporarily manipulated (e.g. through borrowing/lending)
- Incomplete without proper accounting of liabilities (could be off chain, could be senior. . .)
- Regulatory frameworks and even abstract legal concepts of ownership/duty of care) are not yet settled
- Requires some level of user sophistication
- Adds complexity - possibility of errors (e.g. how to rapidly check reserves controlled by cold wallets?)

Additional PoR use cases

- Stablecoins:
  - Stablecoins should only allow minting if the assets are available to back coins
  - Value and nature of assets should be checked at high frequency
  - PoR would be accompanied by PoL (liabilities)
  - Coin holdings ‘match up’ to the (pooled) backing assets
- Tokenized RWAs:
  - Challenging problem since assets (almost by definition) are off-chain
  - Likely need close interaction with auditor or bank/custodian
  - Argument in favor of on-chain sovereign debt and/or CBDC?
- Cross chain protocols/activity:
  - Reserves may be scattered across different blockchains
  - Tricky to coordinate simultaneous checks (and complex)
- Oracles (e.g. Chainlink) often provide PoR services
  - Interesting avenue for research - especially as compliance (which varies across jurisdiction) complicates matters