zkLedger: Privacy-Preserving Auditing for Distributed Ledgers

- Reference: https://www.usenix.org/conference/nsdi18/presentation/narula

1. Contributions

- the first distributed ledger system to support strong transaction privacy, public verifiability, and practical, userful auditing
    - Privacy: hides transacting banks and amounts
    - Public Verification: everyone can verify transactions
    - Auditing: compute linear functions over transactions
- do not require trusted setup and only rely on widely-used cryptographic assumptions

2. Overview

1. System Model
    - Banks
        - issue thransactions to transfer digital assets
        - should not be able to hide assets from the Auditor
    - Auditor
        - verifies certain operational aspects of transactions performed by the participants
        - should not be able to see individual bank transactions
        - should be able to detect an incorrect answer
    - Depositor
        - use and withdraw assets from the system
        - transactions are public
2. Building Blocks
    - Pedersen Commitment
    - Public-Key Encryption
        - a secret key sk_i
        - public key $pk_i = h^{sk_i}$
    - Non-Interactive Zero-Knowledge Proofs
3. Security Model
    - does not assume that banks will behave honestly
    - assume banks can arbitrarily collude
    - Banks or the auditor might try to learn transaction contents

3. Design

- Commitment

- each transaction has an entry for each Bank
  - each entry includes a commitment to a value the amount of the asset that is being debited or credited to the bank
  - the sum of every entry should be zero
  - use Pedersen commitments to commit to the value
- Audit Token
  - $T_i := (pk_i)^{r_i}$
  - to answer audits without knowing the randomness used in the commitment
  - To sum of values in a bank's column, $\prod cm_i = g^{\sum v_i} h^{\sum r_i}$
  - a bank does not necessarily know all the commitment randomnesses r_k
- Zero-Knowledge Proofs
  - the spender can create to prove the invariants are maintained
  - Proof of Consent
    - consent to transfer
    - signature
  - Proof of Asset
    - a new commitment cm'_i is a re-commitment of cm_i or $\prod cm_i$
    - cm'_i is in range $[0, N)$ where N is the size of message space (in zkLedger, $N = 2^{40}$)
    - range proofs
  - Proof of Balance
    - the committed values satisfy $\sum v_i = 0$
  - Proof of Consistency
    - cm_i and T_i are generated by the same random r_i for each i
    - $\prod cm_i$ and $\prod T_i$ are generated by the same random $\sum r_i$