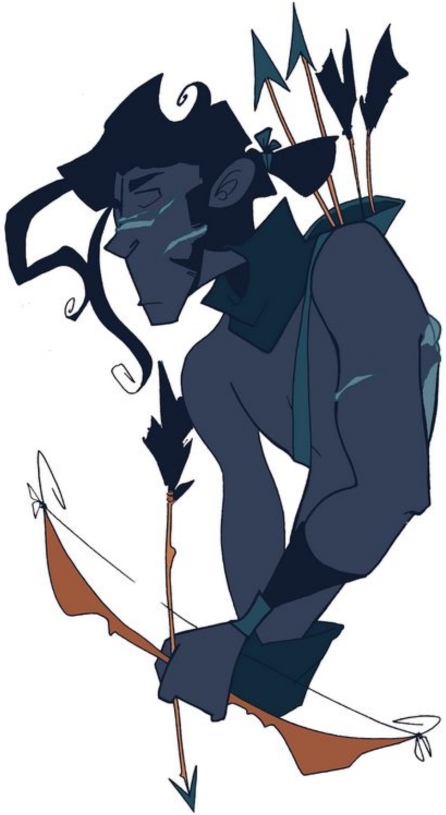
Lelantus-MW

The ~~hybrid~~ symbiosis



Brief overview of MW

- UTXO as Pedersen commitment
  - $C = \alpha \cdot G + v \cdot H$
- Transactions:
  - No scripts, no transactions in the “classical” sense
  - Balance-to-zero principle
  - Merged non-interactively!
- Cut-through
  - Block is one big transaction
  - The whole blockchain history is one huge transaction
  - Spent outputs are removed

So far so good

- Great anonymity out-of-the-box
  - **All** transactions are confidential
  - Values are blinded (concealed)
  - No addresses, accounts, user tokens or etc.
  - Transaction graph is obfuscated
- Great scalability
  - Spent outputs are completely erased
  - Only kernels remain (~100 bytes per tx)

What could be wrong with MW?

So far so good

- Great anonymity out-of-the-box
  - **<u>All</u>** transactions are confidential
  - Values are blinded (concealed)
  - No addresses, accounts, user tokens or etc.
  - Transaction graph is obfuscated not really…
- Great scalability
  - Spent outputs are completely erased
  - Only kernels remain (~100 bytes per tx)

What could be wrong with MW?

Linkability

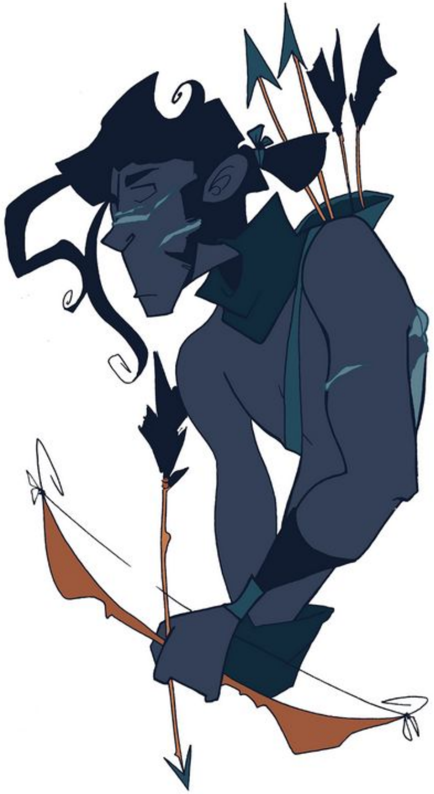
- The Achilles heel of MW!
- Cut-through doesn’t improve anonymity!
- Optimistically – up to ~1000 transactions in a block are mixed
  - But not all blocks are big!
- Transaction broadcast is non-trivial
- Not good enough against “active” attacker

Possible “laundry” solutions

- Current solution:
  - Modified Dandelion with transaction join during stem phase
  - Decoy inputs/outputs (UTXOs with zero value)
- Other poor man’s solutions:
  - Coinjoin
  - Trusted payment hubs
- Drastic solutions:
  - zk-SNARKs, zk-STARKs
  - Bulletproofs (for arbitrary circuit)

Lelantus

- Work of Aram Jivanyan, Zcoin's cryptographer
  - Disclaimer: Our design and implementation are based on the publicly-available Lelantus scientific paper. All our code was developed from scratch based on this paper alone.
- Natural ally:
  - Designed as an add-on (laundry) to any protocol
  - Same cryptographic assumptions (DLP, no trusted setup)
  - Similar constructs: Pedersen commitments, rangeproofs, vector commitments
  - Based on the One-out-of-many Sigma-protocol by Jens Groth



Brief overview of Lelantus

- Lelantus UTXO
  - $C = \alpha \cdot G + v \cdot H + \boldsymbol{s} \cdot \boldsymbol{J}$
  - $\boldsymbol{s}$ – serial number, derived from pubkey Pk.
- Spend transaction
  - Pk is revealed, and the whole transaction is signed by appropriate secret key
  - $\boldsymbol{s} \cdot \boldsymbol{J}$ is subtracted (methodically) from the commitments in the pool
  - Modified Sigma-protocol in terms of G, H generators.
- The net value extracted from the shielded pool is revealed
  - Separate proof proves its correctness
  - For this original Sigma-protocol is significantly modified

Lelantus-MW

- Why not just use Lelantus as indented for Zcoin?
  - Values should not be revealed
  - Keep cut-through for the MW part
- Our (Beam) modified version
  - Reveal Pedersen commitments instead of values
  - Reveal commitment for each individual spent UTXO
    - Would be a bad idea if values were revealed
    - Separate spend proof can be omitted!
  - Keep balance-to-zero principle
  - Keep MW-style transactions!
    - MW/Lelantus inputs/outputs can come in any combination

Lelantus-MW primitives

- Input
  - Pedersen commitment
  - MW: must be in the current UTXO set
  - Lelantus: Spend proof is attached
- Output
  - Pedersen commitment
  - MW: Bulletproof (rangeproof)
  - Lelantus: double-blinded bulletproof
- Kernel
  - Pedersen commitment
  - MW: Schnorr’s signature
  - Lelantus: generalized Schnorr’s signature (in terms of G, J generators)

Spend proof

- Pedersen commitment
  - $C = \alpha' \cdot G + v \cdot H$
    - Value v is the same as of the spent UTXO
    - Blinding factor α' <u>different</u>
  - Generalized Schnorr's signature to prove the above
- $\boldsymbol{s}$ – serial number, derived from the revealed pubkey Pk
- $(C + s \cdot G)$ is subtracted (methodically) from the commitments in the shielded pool
- Original Sigma-protocol proves the knowledge of an element in the pool, in terms of G generator only.
  - The witness data is the blinding factor difference $\alpha - \alpha'$
- Separate balance proof is not needed!

Lelantus-MW implications

- Pros:
  - Linkability break!
  - One-side payments
- Cons
  - Obviously no cut-through for shielded pool
  - Verification time is dramatically higher
    - Nearly linear in anonymity set size
    - 1 sec for anonymity set of 65536 elements
    - But only 15 msec for each additional proof for the same anonymity set
    - Easily parallelized
    - Precomputations are effective, but dramatically inflate the storage size
- Most of transactions should remain in MW
- Lelantus should be a "premium feature"
- Consensus rules must restrict the overall anonymity set referenced by a block and limit the number of spend proofs.
  - This should create a fee market

Conclusions

- So, problem solved? Not completely!
    - Dust attack is a threat
    - Proper strategy must separate “clean” UTXOs from others
- Compared to Zcash
    - Great technology, but NOT immune either!
    - Unlimited anonymity set is a big advantage, but:
        - Probability distribution is not uniform!
        - Recent outputs are more likely to be spent
        - Only hundreds of shielded outputs per day
    - Metadata leakage (correlated values, number of JoinSplits, etc.)
- Breaking linkability is HARD!
- ANY induced (stereotypic) behavior in an attack target!
    - Theoretically with enough experiments the attacker can reach arbitrary precision
    - The goal is to make such attacks infeasible in practice

Thank you!