Advanced Topics

Portland State
Computer Science

Block times and sizes

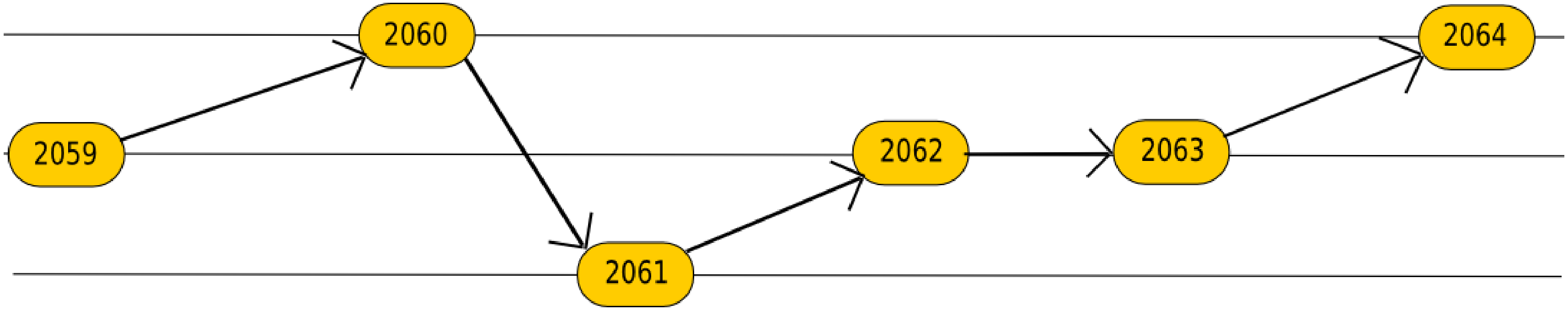
Block size

- Increase block sizes
  - Tension between those who treat BTC as an investment (e.g. like a stock that does not trade frequently) versus a transactional currency (e.g. like cash and credit cards)
    - At 7 transactions/second, it's being treated as the former
- Within Bitcoin: SegWit upgrade (7/21/2017) (2MB)
  - Patch to fix transaction malleability bug that effectively doubles block-size
  - Leads to Bitcoin Cash hard fork (8/1/2017) (8MB)
    - For those who did not believe SegWit did enough
  - Then Bitcoin Cash split again
    - Bitcoin ABC (adjustable Blocksize Cap) 32MB size
    - Bitcoin SV 128MB size

- Larger block sizes
  - Increases amount of hardware needed to handle
  - Decreases transaction time
  - Decreases transaction cost
  - Increases propagation time

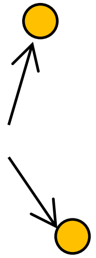
Block time

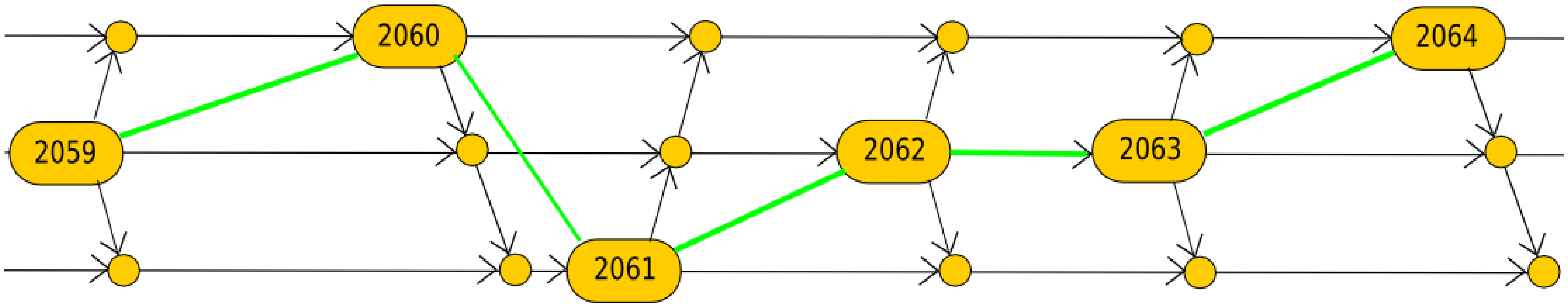
- Decreasing block times improves transaction throughput linearly
- But, impacts consensus
  - Orphan rate of chains increases
  - Amount of wasted work on PoW computation increases
  - Example
    - 3 miners mining and distributing blocks



- Miners continually mining ⟶
- Miner successfully mines block ⬭

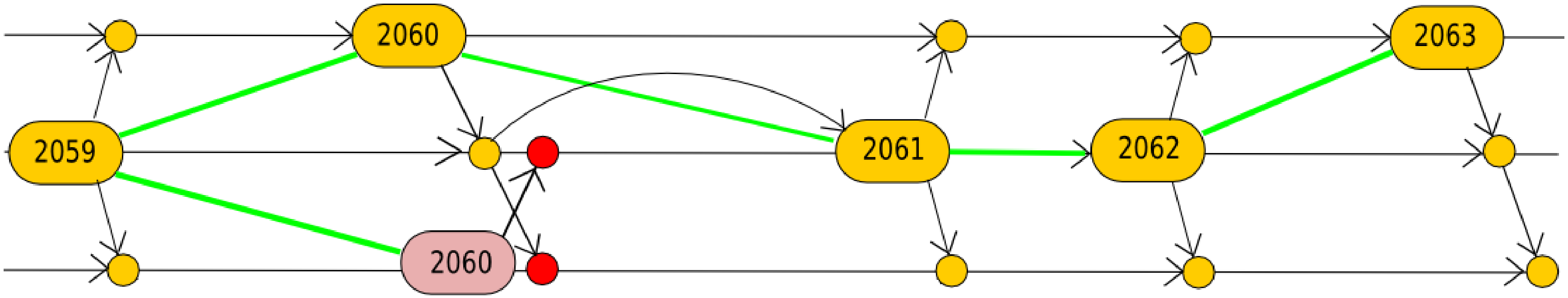
- Block propagated to all other miners so they can move on to mining next block





- During propagation, a miner may successfully mine a different block and propose it (e.g. there may be two valid candidates for block 2060)

- Top and bottom miners successfully mine candidate for 2060 and attempt to propagate *before* receiving each other's proposed block



- Issues
  - Miners working on different versions of 2060 create wasted work with no added stability to blockchain
  - Shorter block times increase wasted work (since propagation time becomes larger as compared to mining time)
  - Mining pools with fast network connections at an advantage
    - Waste less time on performing hashes as successfully mined blocks are being propagated
    - Can immediately go to next block
  - Mining centralization becomes more of a threat
    - With pools and mining devices mostly in China

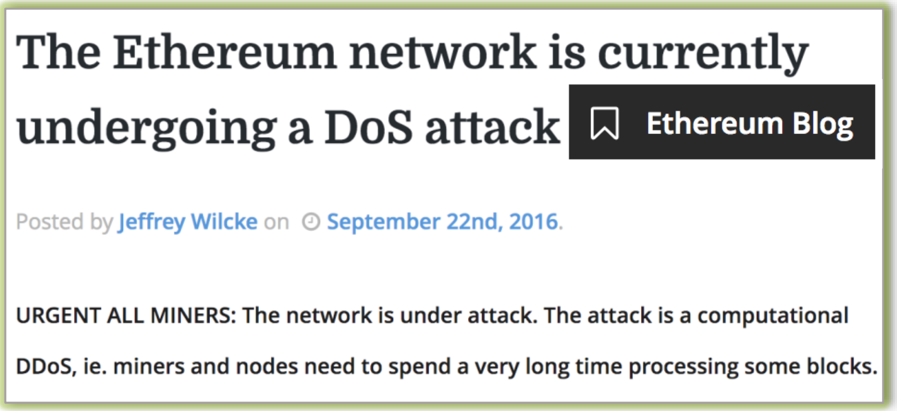
- Ethereum's GHOST (Greedy Heaviest Observed Subtree)
  - Goal: Incentivize miners to coalesce into the main chain, but prevent centralized mining pools from gaining an unfair advantage
  - Address centralization issues with short block-time by incorporating stale blocks
    - Take common sub-tree out of mined blocks being proposed
    - Reward miners who have mined blocks with the sub-tree (even if blocks contain "uncles" that are not ultimately accepted)

Block times in practice

- Bitcoin
  - ~10 minutes
  - But, is 10 minutes way too conservative?
    - Takes 12.6s on average to propagate block to 95% of nodes
    - Perhaps a 1-minute block-time is more appropraiate?
- Ethereum
  - 10-20 seconds due to GHOST

Sharding, side-chains

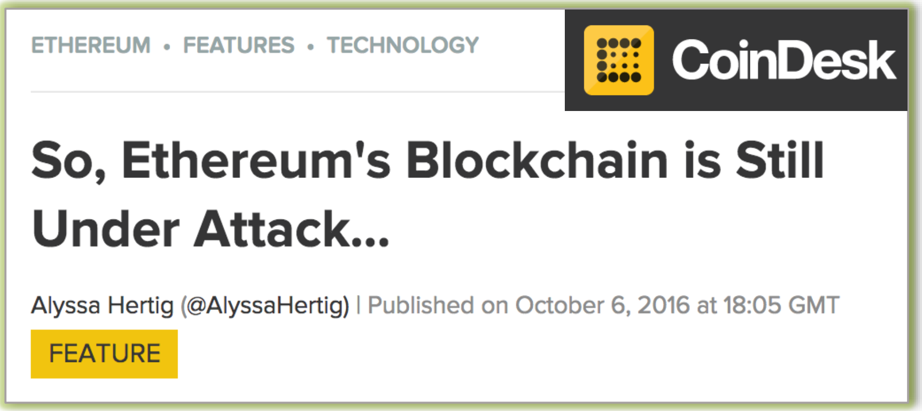
- Issue #1: Resources on blockchain are expensive
  - Full nodes perform the same on-chain computations
  - Full nodes store the same data
  - Gas-limit is relatively small as a result
    - Can’t run an OS on blockchain
    - Can’t increase gas-limit: DoS vector



The Ethereum network is currently undergoing a DoS attack

Posted by Jeffrey Wilcke on September 22nd, 2016.

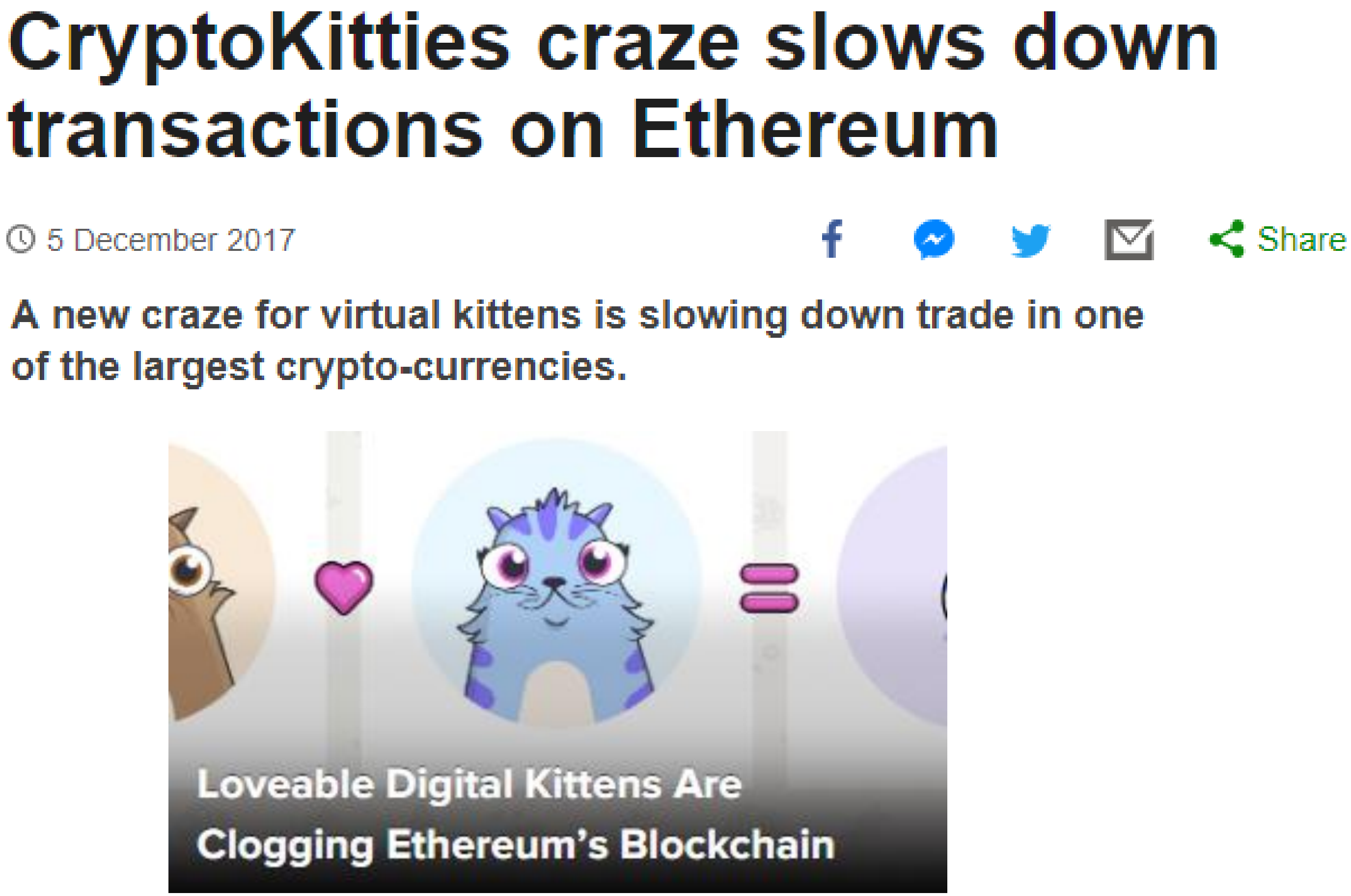
URGENT ALL MINERS: The network is under attack. The attack is a computational DDoS, ie. miners and nodes need to spend a very long time processing some blocks.



So, Ethereum's Blockchain is Still Under Attack...

Alyssa Hertig (@AlyssaHertig) | Published on October 6, 2016 at 18:05 GMT

- Issue #2: Single blockchain for all DApps to share
  - Implements a total order on events within a DApp and events across all DApps
  - For independent DApps, why is this necessary?



CryptoKitties craze slows down transactions on Ethereum

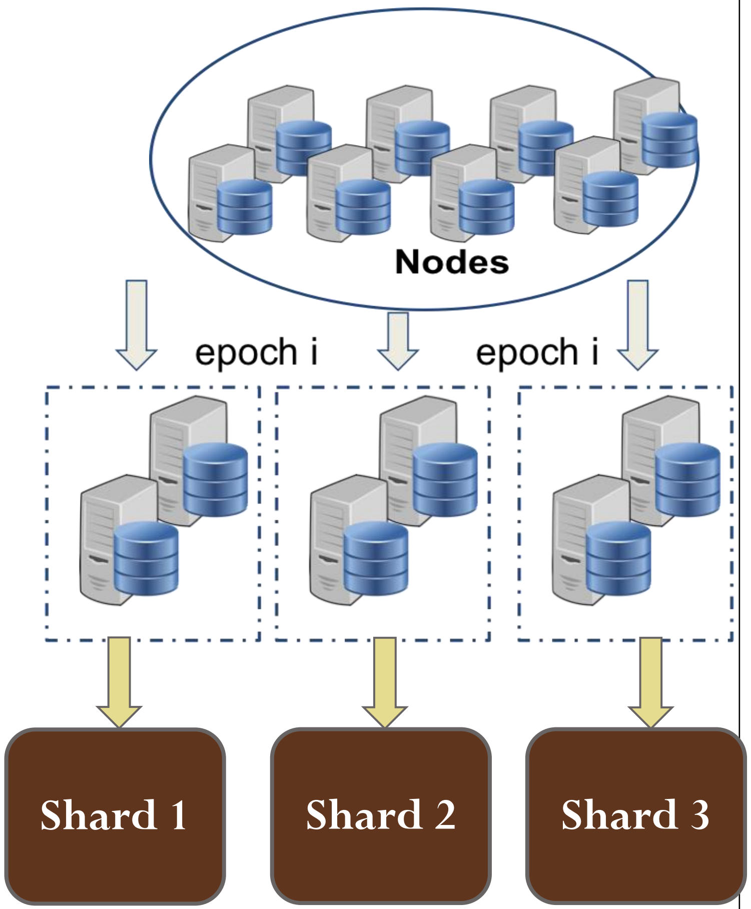
5 December 2017

A new craze for virtual kittens is slowing down trade in one of the largest crypto-currencies.

Loveable Digital Kittens Are Clogging Ethereum's Blockchain

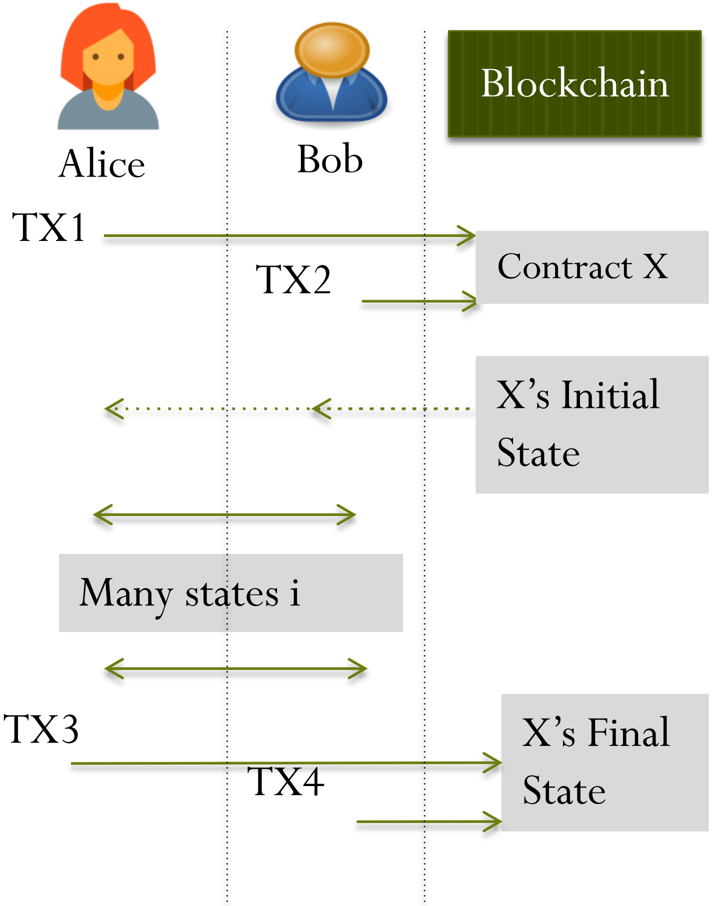
Solution 1: Sharding

- Divide the network into sub-networks
  - Each stores and manages a fraction of the blockchain (a shard)
  - Allow scaling up as the network grows
- Hierarchical block-chains



Solution 2: State Channel, Layer-2 solutions

- Similar to payment channel (e.g. lightning network) but for states
  - Scaling by using off-chain transactions
  - Can update the state multiple times off-chain
  - Only settlement transactions are on-chain



Formal verification

Tools to prove correctness

- Formal methods to ensure correctness of EVM itself via Isabelle
- Formal methods to verify smart contracts
  - Why3 programming language (4/2019)
  - Language for writing formal and verified smart contracts via deductive verification
- Integrate contract testing into IDE
  - Truffle development environment

Decoupling state machine and consensus

Tendermint

- Ethereum VM and Solidity conjoin both the state in a contract with the replication of it across nodes
- Why can't the state machine be managed by any programming language and then use the blockchain only as a replication service?
  - e.g. write DApp in Java and then have blockchain replicate JVM underneath
- Tendermint approach
  - Separate state management (e.g. PL and its VM) from the replication and consensus of it

Thwarting miner centralization

Issue

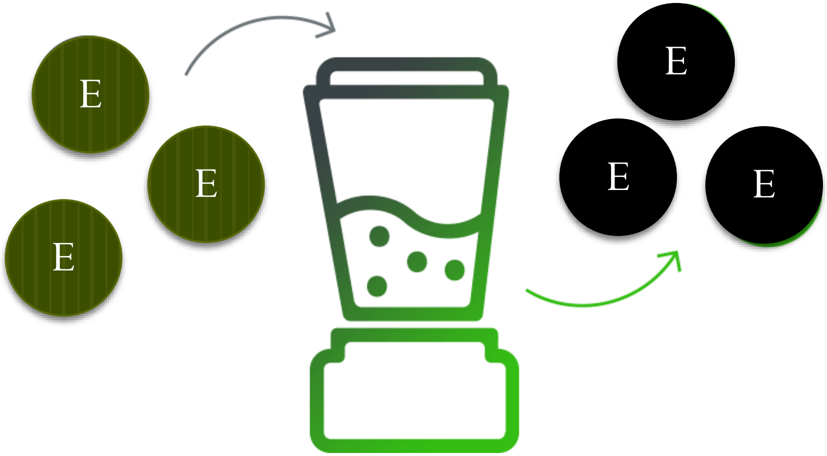
- 80-90% of all mining hardware in Bitcoin from a single factory in Shenzhen China (Bitmain)
  - Highly parallelizable hashing algorithm eventually done by ASICs
- Alternatives
  - Memory bound puzzles (Ethhash)
    - Use a scheme in which miner must store data in high-speed memory that is randomly accessed to compute puzzle solution
    - Use a size that fits in L3 cache (too big for ASICs and some GPUs)
  - Puzzle algorithms that continually change
    - Update algorithm for mining to invalidate ASICs and force a redevelopment of hardware
    - ProgPoW in Ethereum
  - Both techniques used in CryptoNote/Monero

Privacy

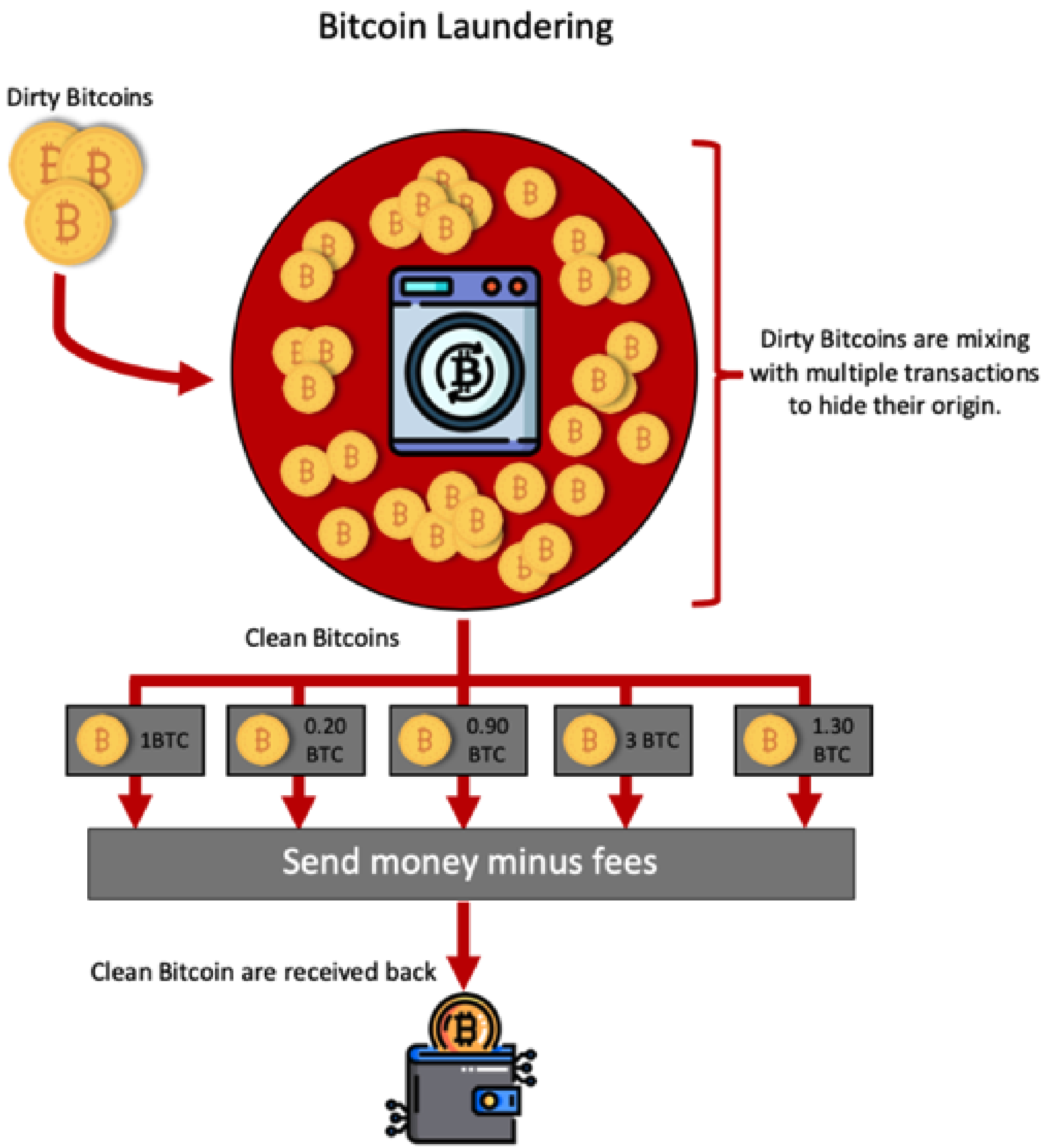
- Blockchain supports consensus, correctness, authenticity, and availability, but not privacy for smart contracts or transactions
- All Bitcoin transactions public (transactions of wallets public)
  - Tracing Bitcoin transactions per wallet simple (and effective)
    - Analysing transaction graph [IMC'13]
  - Good for law enforcement
- All Ethereum smart contract executions (data & code) public
  - Cannot execute on private data
  - e.g. Can not have a death will that remains secret until the owner dies

Proposed solutions

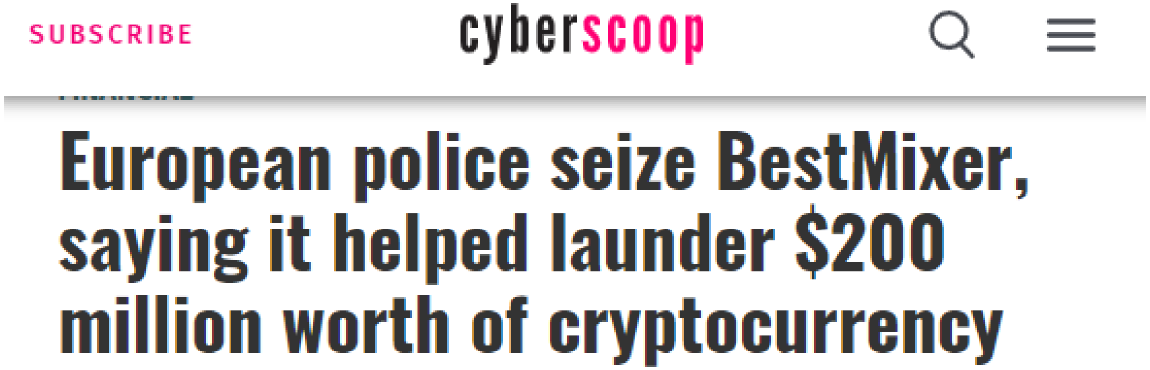
- Crowds
  - Clearinghouse account for mixing coin transactions to support "k-anonymity"



- Should this be legal?


Bitcoin Laundering
Dirty Bitcoins
Dirty Bitcoins are mixing with multiple transactions to hide their origin.
Clean Bitcoins
1BTC
0.20 BTC
0.90 BTC
3 BTC
1.30 BTC
Send money minus fees
Clean Bitcoin are received back

- Depends on how you market your service
  - Bestmixer.io laundering pool taken down

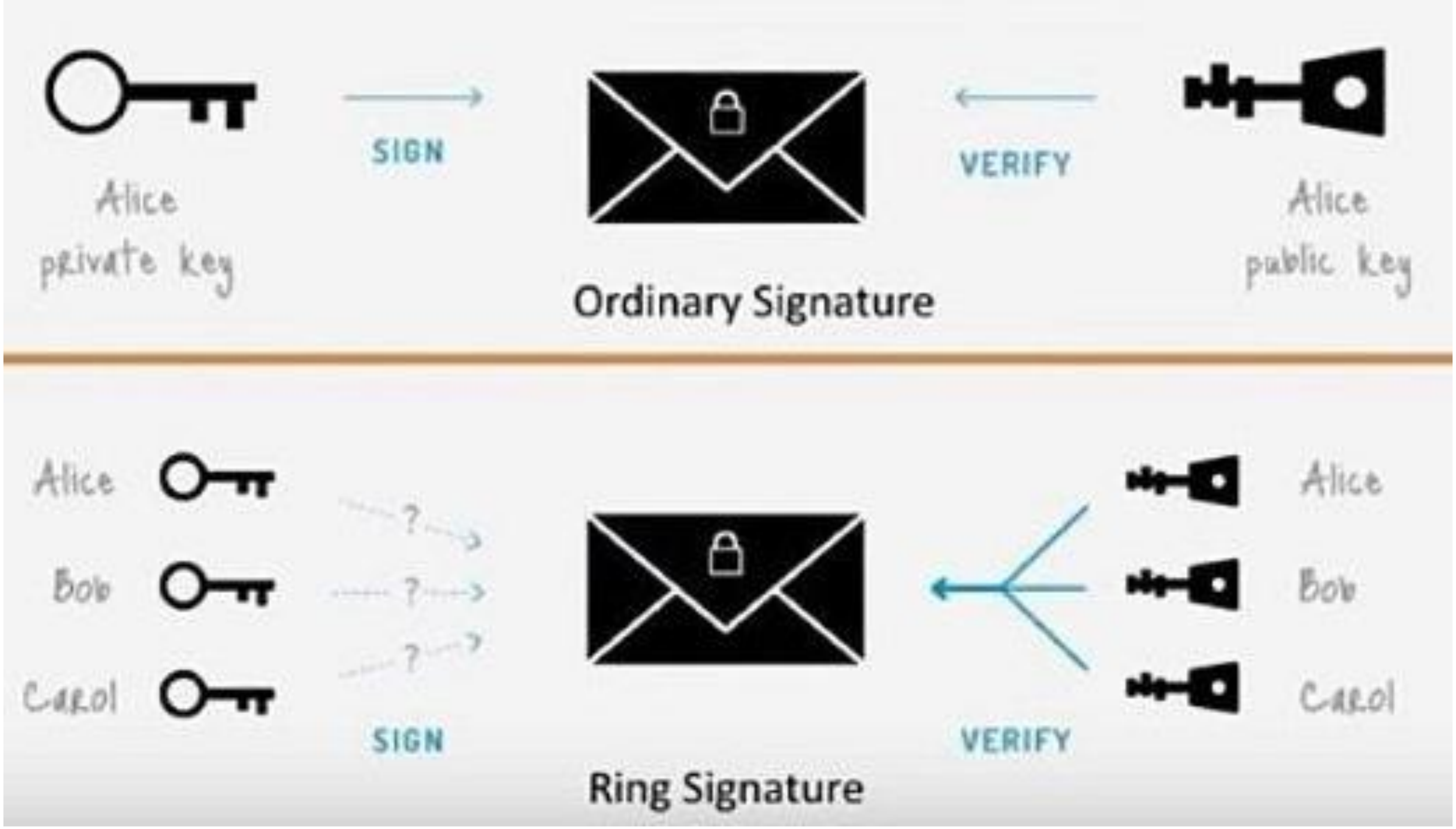


European police seize BestMixer, saying it helped launder $200 million worth of cryptocurrency

  - “Mixing bitcoins that are obtained legally is not a crime but, other than the mathematical exercise, there is no real benefit to it”
  - “The legality changes when a mixing service advertises itself as a success method to avoid various anti-money laundering policies via anonymity.”

Ring signatures (a.k.a. group signatures)

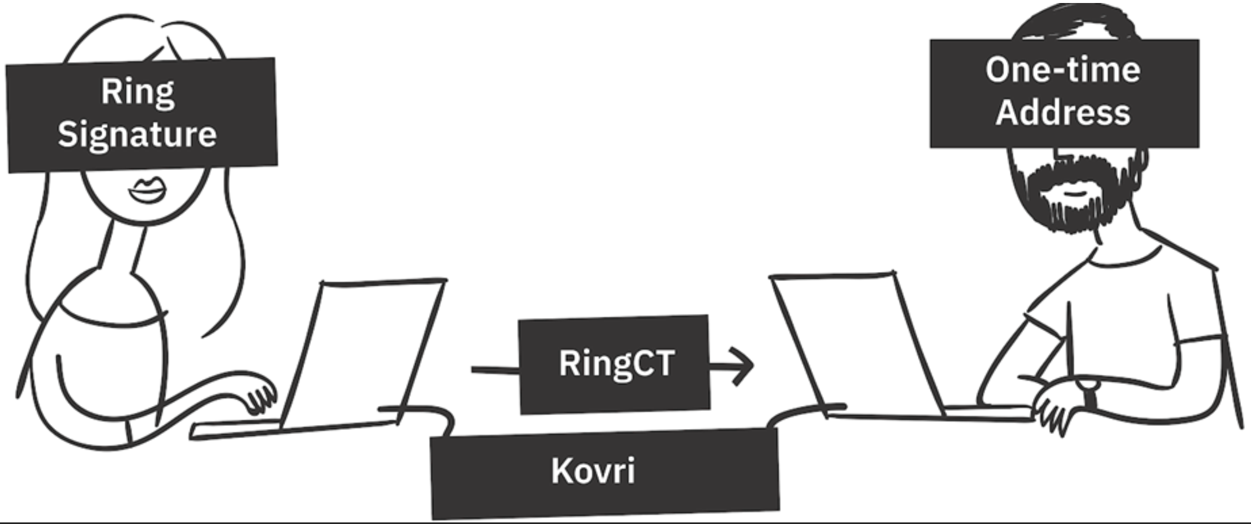
- Implementation of a mixer
- Example
  - Five users send their public keys in alongside a deposit of 0.1 ETH
  - Withdraw 0.1 ETH specifying the address with a linkable ring signature
  - Simultaneously guaranteeing that
    - Everyone who deposited 0.1 ETH will be able to withdraw 0.1 ETH exactly once
    - It's impossible to tell which withdrawal corresponds to which deposit.
  - On Ethereum (description | mixing contract)



- Size of ring based on user's desired ambiguity degree
- Senders verify each other using group of public keys in ring

Unlinkable payments via one-time keys

- Add a level of indirection similar to Tor
- Private key of sender creates
  - SendKey private/public key pair
  - ViewKey private/public key pair
  - Address
- Sender uses private SendKey to initiate payment and gives recipient ViewKey
  - Passes through ring signature to hide sender address
  - Transaction sent to a one-time Stealth wallet address
- Receiver uses private ViewKey to check wallet address for available funds
  - Done over an anonymizing network (Kovri)

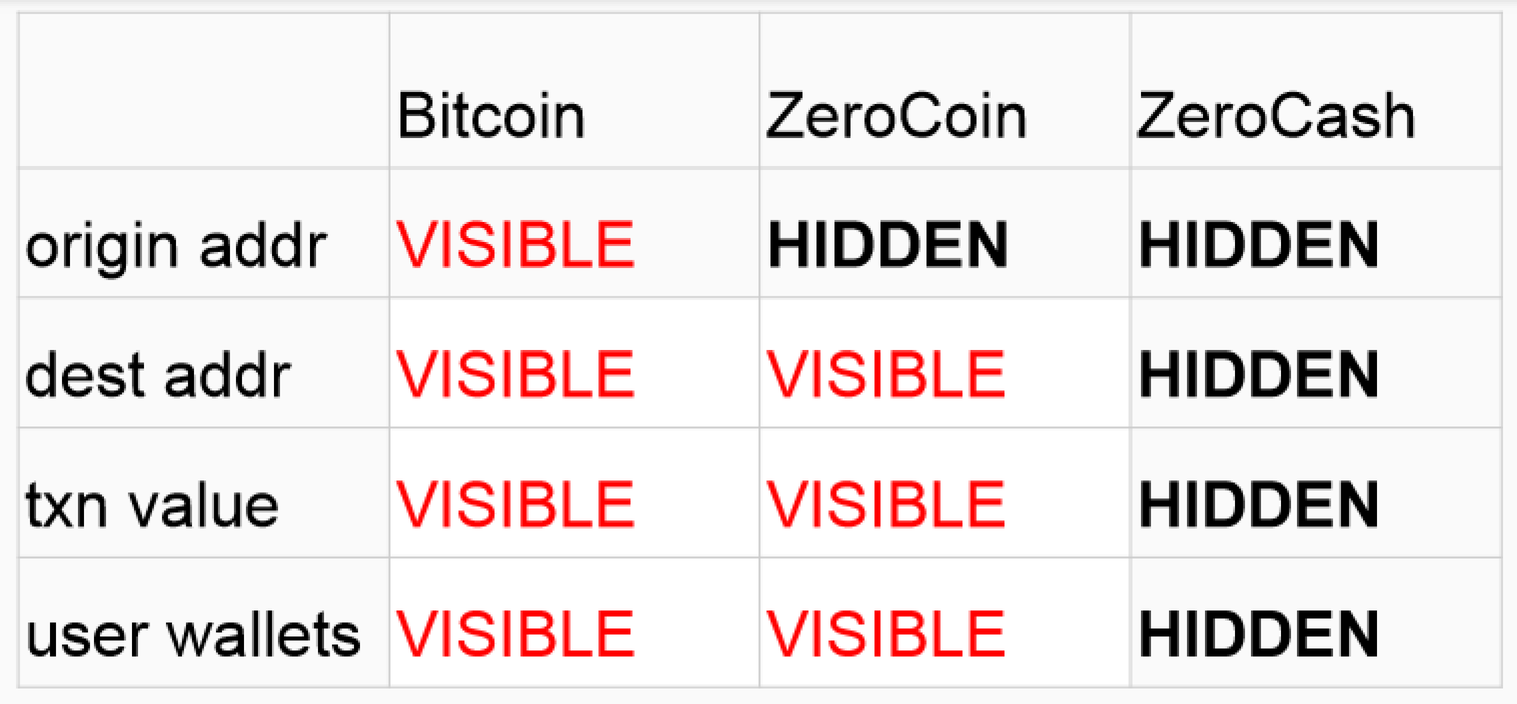


Example: ZeroCoin

- Proposed extension to Bitcoin
  - Unlink transactions to their origins
  - Payment destination and amounts still linked and traceable
  - Done via a de-centralized mixer where coins can be periodically washed of their transaction history
  - Fixed denomination coins initially
  - Extra steps required to perform transaction
  - Not quite anonymous

Example: Zcash

- Fully anonymous and decentralized protocol
- Done via zero-knowledge proofs (ZKPs)
  - See extra slides
- ZeroCash over Ethereum



	Bitcoin	ZeroCoin	ZeroCash
origin addr	VISIBLE	**HIDDEN**	**HIDDEN**
dest addr	VISIBLE	VISIBLE	**HIDDEN**
txn value	VISIBLE	VISIBLE	**HIDDEN**
user wallets	VISIBLE	VISIBLE	**HIDDEN**