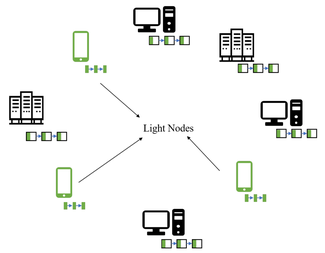
Concentrated Stopping Set Design for Coded Merkle Tree: Improving Security Against Data Availability Attacks in Blockchain Systems

Debarnab Mitra, Lev Tauz, Lara Dolecek

debarnabucla@ucla.edu, levtauz@ucla.edu, dolecek@ee.ucla.edu

University of California, Los Angeles

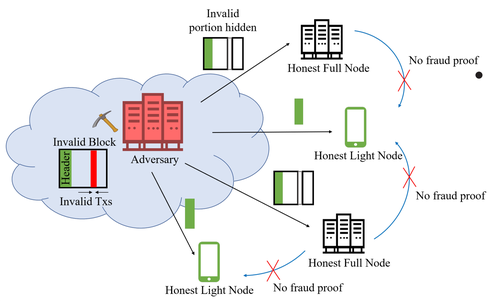
Blockchain systems



- Blockchain ledger maintained by a network of nodes
- Full nodes: maintain a local copy of the entire ledger → prohibitive storage costs
  Bitcoin ledger size ∼ 350GB, Ethereum ledger size ∼ 600GB[a]
- Light nodes: Only store block headers (total size ∼ 1GB for Ethereum)
  → Rely on honest Full nodes for fraud notification via verifiable fraud proofs

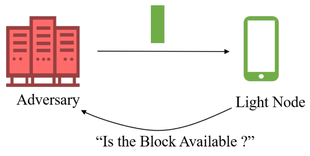
[a] As of 3/12/2021, https://bitinfocharts.com/

Data Availability (DA) Attacks in Blockchains



- Systems with light nodes and a dishonest majority of full nodes are vulnerable to DA attacks [Al-Bassam '18], [Yu '19]
  - Adversary: Provides block to Full node but hides invalid portion
    Provides header to Light node
  - Honest Nodes: Cannot verify missing transactions → No fraud proof
  - Light Nodes: No fraud proof → accept the header.

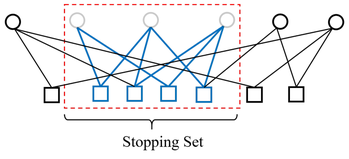
Ensuring Data Availability: Light node sampling

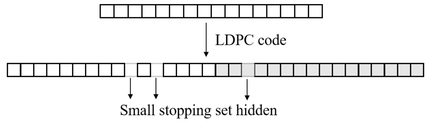


- Anonymously request/sample few random chunks of the block
- Adversary can hide a small portion
- To improve storage efficiency, LDPC codes are used [Yu '19]
  - Characterized by a sparse parity check matrix
  - Tanner Graph representation

Probability of failure affected by small stopping sets

- If hidden, prevents a peeling decoder from decoding the block → No fraud proof from full nodes





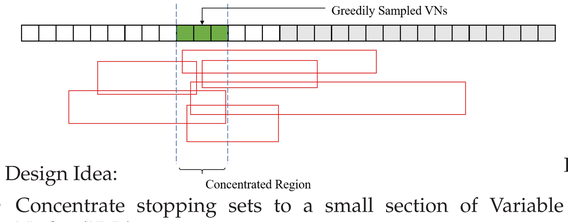
Probability of failure (2 samples):

$$\left(1-\frac{3}{32}\right)\left(1-\frac{3}{31}\right)=0.81$$

Our work: Design specialized LDPC codes with a coupled sampling strategy to achieve a significantly lower probability of failure

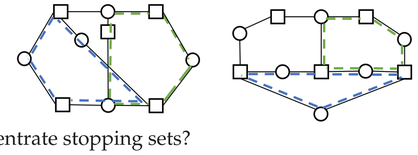
Concentrated Stopping Set Design

In this work, we consider an adversary which randomly hides a stopping set of a particular size.



Code Design Idea:

- Concentrate stopping sets to a small section of Variable Nodes (VNs)
- Greedily sample this small section of VNs

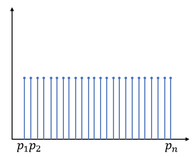


How to concentrate stopping sets?

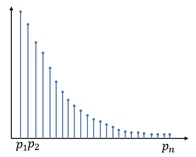
- Concentrating cycles ⟹ Concentrating stopping sets
- We concentrate cycles by modifying the Progressive Edge Growth (PEG) algorithm

Entropy to Concentrate Cycles: EC-PEG Algorithm

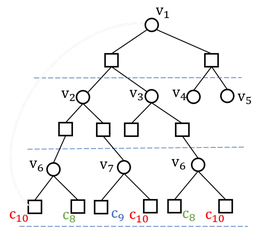
- Uniform distributions have high entropy
- Concentrated distributions have low entropy



High Entropy



Low Entropy



EC (Entropy Constrained)-PEG Algorithm
For each VN v_j
  Expand Tanner Graph in a BFS fashion
  If ∃ CNs not connected to v_j
    • select a CN with min degree not connected to v_j
  Else *New cycles created*
    • Find CNs most distant to v_j
    • Select CN that results in minimum entropy of resultant cycle distribution
    • Update cycle distribution

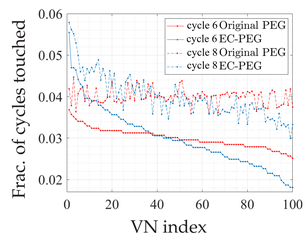
We want the cycle distributions to be concentrated
→ Select Check Nodes (CNs) such that the entropy of the cycle distribution is minimized
Sampling Strategy:

- Greedy Sampling: greedily sample VNs that are part of a large number of cycles
- Random Sampling (with replacement): sample each variable node with equal probability

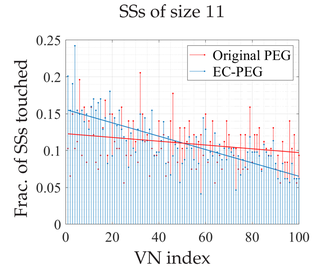
Simulation Results: Evidence of Concentration

Code parameters: Code length = 100, VN degree = 4, Rate = $\frac{1}{2}$, girth = 6.

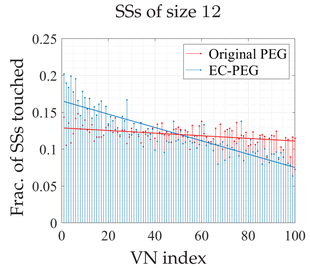
- VN indices arranged in decreasing order of cycle 6 fractions
- Cycles and SSs are concentrated towards the same set of VNs



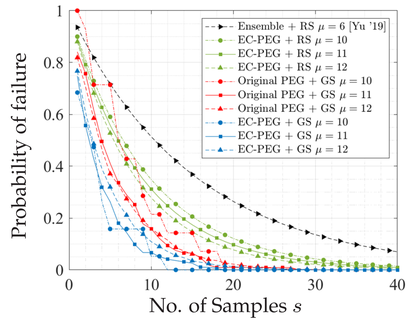
SSs of size 11



SSs of size 12

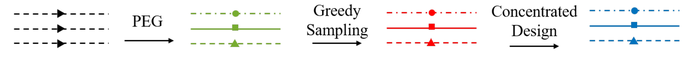


Simulation Results: Probability of Failure



- Probability of failure for a Stopping set of size μ
- Code parameters: Code length = 100, VN degree = 4, Rate = $\frac{1}{2}$
- RS: Random Sampling, GS: Greedy Sampling

Three Levels of improvement:



Concentrated LDPC codes coupled with a Greedy sampling strategy improve the probability of failure

References

- (Al-Bassam '18) M. Al-Bassam, et al., *"Fraud and Data Availability Proofs: Maximising Light Client Security and Scaling Blockchains with Dishonest Majorities,"* arXiv preprint arXiv:1809.09044, 2018.
- (Yu '19) M. Yu, et al., *"Coded Merkle Tree: Solving Data Availability Attacks in Blockchains,"* International Conference on Financial Cryptography and Data Security, Springer, Cham, 2020.
- ⋆ Full paper of this work: D. Mitra, L. Tauz, and L. Dolecek, *"Concentrated Stopping Set Design for Coded Merkle Tree: Improving Security Against Data Availability Attacks in Blockchain Systems,"* in Proc. of IEEE Information Theory Workshop (ITW), Apr. 2020. (available at https://arxiv.org/abs/2010.07363)