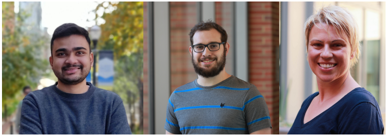
Concentrated Stopping Set Design for Coded Merkle Tree: Improving Security Against Data Availability Attacks in Blockchain Systems

Debarnab Mitra, Lev Tauz, and Lara Dolecek

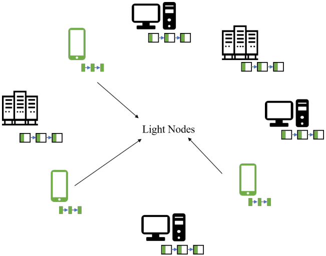
Electrical and Computer Engineering
University of California, Los Angeles

ITW 2020



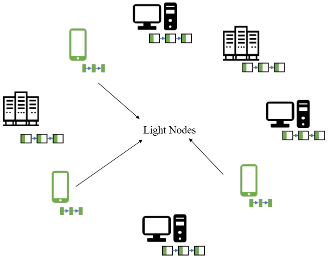
UCLA Samueli School of Engineering

Data Availability (DA) Attacks in Blockchains



Light Nodes:

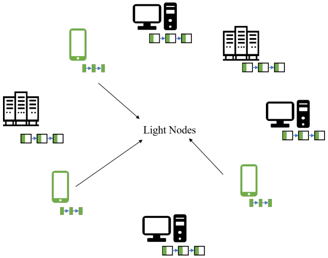
Data Availability (DA) Attacks in Blockchains



Light Nodes:

- Only store block headers

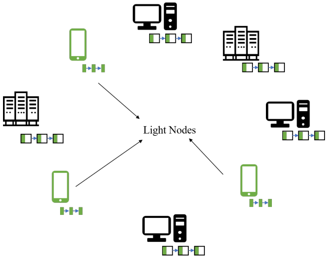
Data Availability (DA) Attacks in Blockchains



Light Nodes:

- Only store block headers
- Rely on honest full nodes for fraud notification

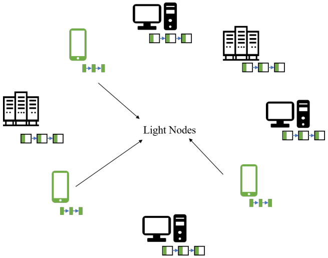
Data Availability (DA) Attacks in Blockchains



Light Nodes:

- Only store block headers
- Rely on honest full nodes for fraud notification
- Systems with dishonest majority of full nodes are vulnerable to DA attacks [Al-Bassam '18], [Yu '19]

Data Availability (DA) Attacks in Blockchains

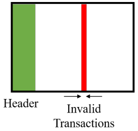


Light Nodes:

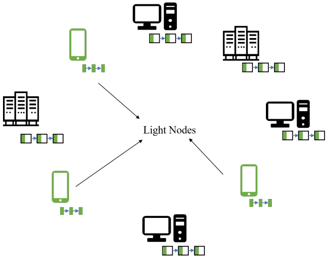
- Only store block headers
- Rely on honest full nodes for fraud notification
- Systems with dishonest majority of full nodes are vulnerable to DA attacks [Al-Bassam '18], [Yu '19]

DA attack:

Adversary creates an invalid block



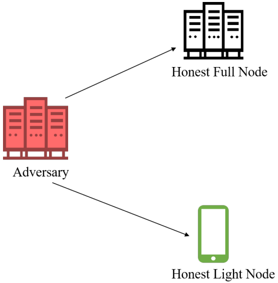
Data Availability (DA) Attacks in Blockchains



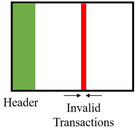
Light Nodes:

- Only store block headers
- Rely on honest full nodes for fraud notification
- Systems with dishonest majority of full nodes are vulnerable to DA attacks [Al-Bassam '18], [Yu '19]

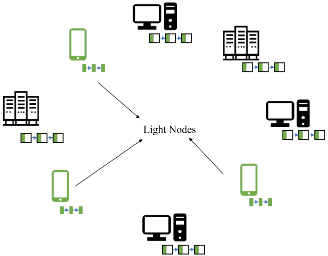
DA attack:



Adversary creates an invalid block



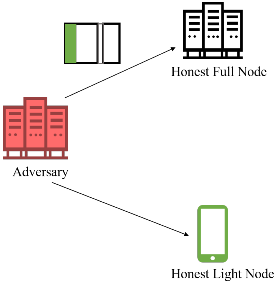
Data Availability (DA) Attacks in Blockchains



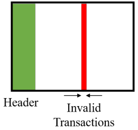
Light Nodes:

- Only store block headers
- Rely on honest full nodes for fraud notification
- Systems with dishonest majority of full nodes are vulnerable to DA attacks [Al-Bassam '18], [Yu '19]

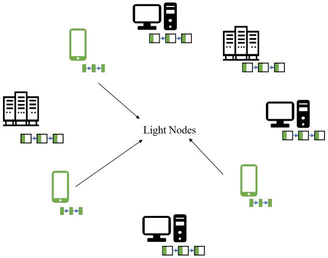
DA attack:



Adversary creates an invalid block



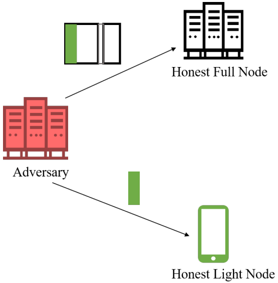
Data Availability (DA) Attacks in Blockchains



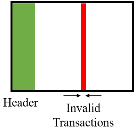
Light Nodes:

- Only store block headers
- Rely on honest full nodes for fraud notification
- Systems with dishonest majority of full nodes are vulnerable to DA attacks [Al-Bassam '18], [Yu '19]

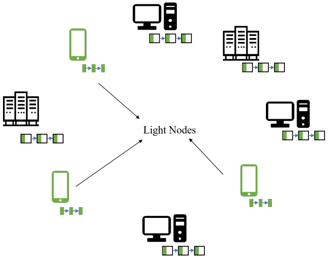
DA attack:



Adversary creates an invalid block



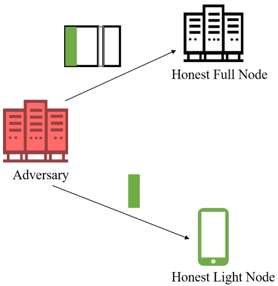
Data Availability (DA) Attacks in Blockchains



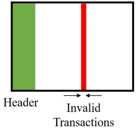
Light Nodes:

- Only store block headers
- Rely on honest full nodes for fraud notification
- Systems with dishonest majority of full nodes are vulnerable to DA attacks [Al-Bassam '18], [Yu '19]

DA attack:

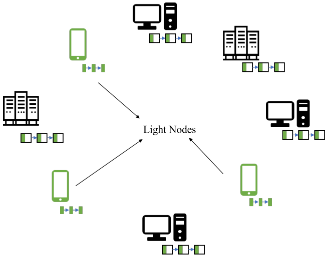


Adversary creates an invalid block



- Full nodes: cannot send fraud proof

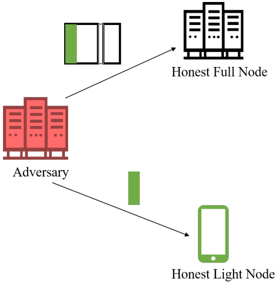
Data Availability (DA) Attacks in Blockchains



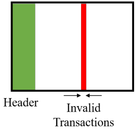
Light Nodes:

- Only store block headers
- Rely on honest full nodes for fraud notification
- Systems with dishonest majority of full nodes are vulnerable to DA attacks [Al-Bassam '18], [Yu '19]

DA attack:

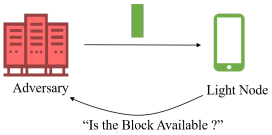


Adversary creates an invalid block



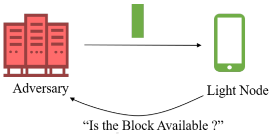
- Full nodes: cannot send fraud proof
- Light nodes: accept the invalid header

Ensure Data Availability: Light Node Sampling



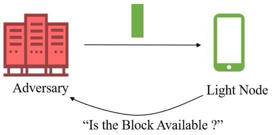
- Anonymously sample few random chunks of the block

Ensure Data Availability: Light Node Sampling



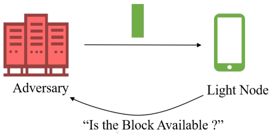
- Anonymously sample few random chunks of the block
- Erasure coding used to improve probability of detection

Ensure Data Availability: Light Node Sampling



- Anonymously sample few random chunks of the block
- Erasure coding used to improve probability of detection
- To improve storage efficiency, LDPC codes are used [Yu '19]

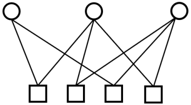
Ensure Data Availability: Light Node Sampling



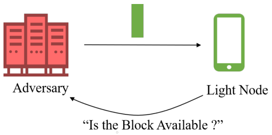
- Anonymously sample few random chunks of the block
- Erasure coding used to improve probability of detection
- To improve storage efficiency, LDPC codes are used

[Yu '19]

Probability of failure affected by small stopping sets



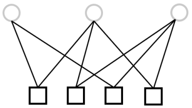
Ensure Data Availability: Light Node Sampling

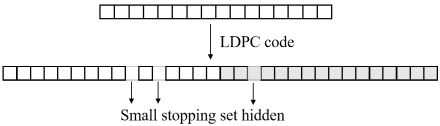


- Anonymously sample few random chunks of the block
- Erasure coding used to improve probability of detection
- To improve storage efficiency, LDPC codes are used [Yu '19]

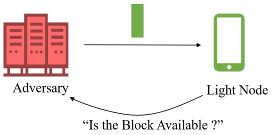
Probability of failure affected by small stopping sets

- If hidden, prevents a peeling decoder from decoding the block





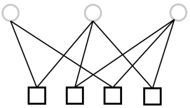
Ensure Data Availability: Light Node Sampling

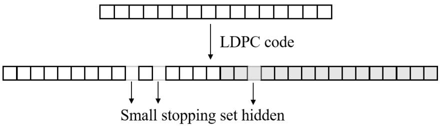


- Anonymously sample few random chunks of the block
- Erasure coding used to improve probability of detection
- To improve storage efficiency, LDPC codes are used [Yu '19]

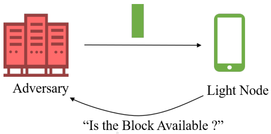
Probability of failure affected by small stopping sets

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





Ensure Data Availability: Light Node Sampling

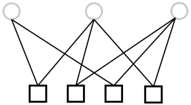


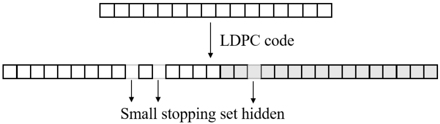
- Anonymously sample few random chunks of the block
- Erasure coding used to improve probability of detection
- To improve storage efficiency, LDPC codes are used

[Yu '19]

Probability of failure affected by small stopping sets

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

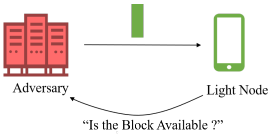




Probability of failure (2 samples):

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

Ensure Data Availability: Light Node Sampling

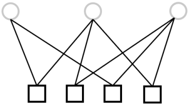


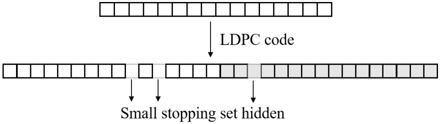
- Anonymously sample few random chunks of the block
- Erasure coding used to improve probability of detection
- To improve storage efficiency, LDPC codes are used

[Yu '19]

Probability of failure affected by small stopping sets

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





Probability of failure (2 samples):

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

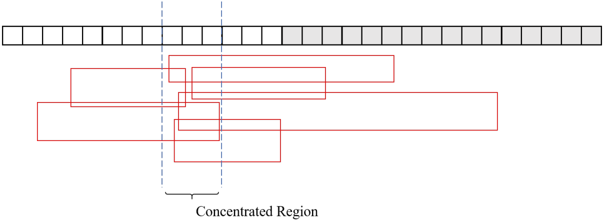
Our work: Design of specialized LDPC codes with a coupled sampling strategy to achieve a significantly lower probability of failure.

Concentrated Stopping Set Design

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

Concentrated Stopping Set Design

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

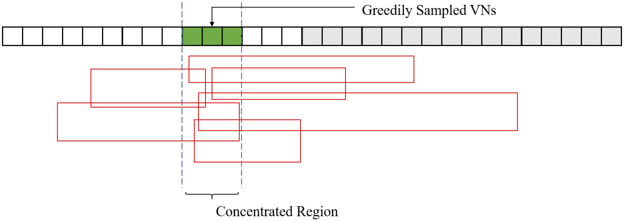


Code Design Idea:

- Concentrate stopping sets to a small section of VNs

Concentrated Stopping Set Design

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

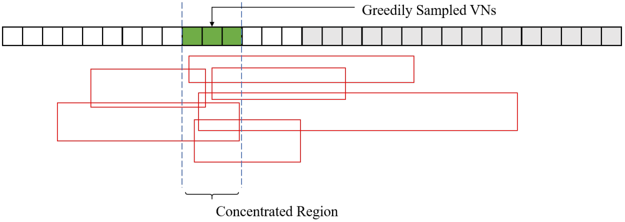


Code Design Idea:

- Concentrate stopping sets to a small section of VNs
- Greedily Sample this small section of VNs

Concentrated Stopping Set Design

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



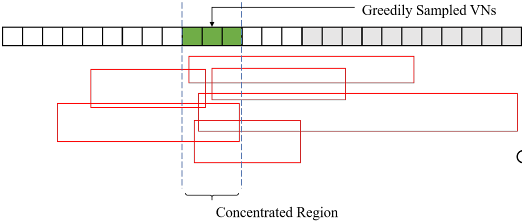
Code Design Idea:

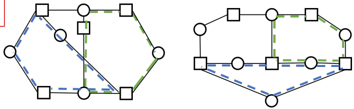
- Concentrate stopping sets to a small section of VNs
- Greedily Sample this small section of VNs

How to concentrate stopping sets?

Concentrated Stopping Set Design

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





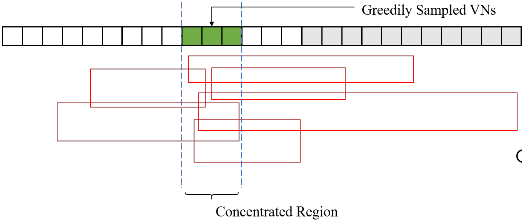
How to concentrate stopping sets?

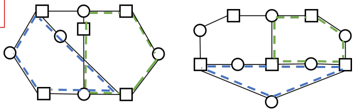
Code Design Idea:

- Concentrate stopping sets to a small section of VNs
- Greedily Sample this small section of VNs

Concentrated Stopping Set Design

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





Code Design Idea:

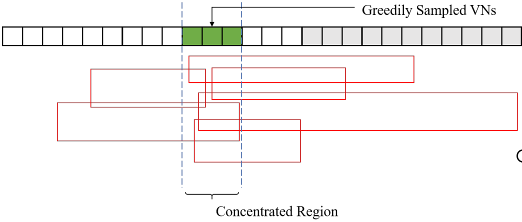
- Concentrate stopping sets to a small section of VNs
- Greedily Sample this small section of VNs

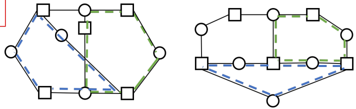
How to concentrate stopping sets?

- Concentrating cycles $\Longrightarrow$ Concentrating stopping sets

Concentrated Stopping Set Design

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





Code Design Idea:

- Concentrate stopping sets to a small section of VNs
- Greedily Sample this small section of VNs

How to concentrate stopping sets?

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

Entropy to Concentrate Cycles: EC-PEG Algorithm

Entropy to Concentrate Cycles: EC-PEG Algorithm

EC (Entropy Constrained)-PEG Algorithm

For each VN v_j

Expand Tanner Graph in a BFS fashion

If $\exists$ CNs not connected to v_j

- select a CN with min degree not connected to v_j

Else *New cycles created*

Entropy to Concentrate Cycles: EC-PEG Algorithm

EC (Entropy Constrained)-PEG Algorithm

For each VN v_j

Expand Tanner Graph in a BFS fashion

If $\exists$ 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

Entropy to Concentrate Cycles: EC-PEG Algorithm

EC (Entropy Constrained)-PEG Algorithm

For each VN v_j

Expand Tanner Graph in a BFS fashion

If $\exists$ 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 concentrates the cycle distribution

Entropy to Concentrate Cycles: EC-PEG Algorithm

EC (Entropy Constrained)-PEG Algorithm

For each VN v_j

Expand Tanner Graph in a BFS fashion

If $\exists$ 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 concentrates the cycle distribution

How do we select a CN that concentrates the cycle distribution?

Entropy to Concentrate Cycles: EC-PEG Algorithm

For distribution $p = (p_1, p_2, \ldots, p_n)$, Entropy $\mathcal{H}(p) = \sum_{i=1}^{n} p_i \log \frac{1}{p_i}$

EC (Entropy Constrained)-PEG Algorithm

For each VN v_j

Expand Tanner Graph in a BFS fashion

If $\exists$ CNs not connected to v_j

- select a CN with min degree not connected to v_j

Else *New cycles created*

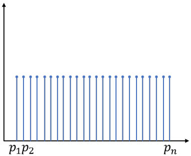
- Find CNs most distant to v_j
- Select CN that concentrates the cycle distribution

How do we select a CN that concentrates the cycle distribution?

Entropy to Concentrate Cycles: EC-PEG Algorithm

For distribution $p = (p_1, p_2, \ldots, p_n)$, Entropy $\mathcal{H}(p) = \sum_{i=1}^{n} p_i \log \frac{1}{p_i}$

- Uniform distributions have high entropy



High Entropy

EC (Entropy Constrained)-PEG Algorithm

For each VN v_j

Expand Tanner Graph in a BFS fashion

If $\exists$ CNs not connected to v_j

- select a CN with min degree not connected to v_j

Else *New cycles created*

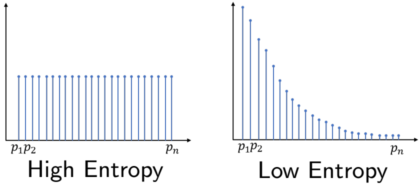
- Find CNs most distant to v_j
- Select CN that concentrates the cycle distribution

How do we select a CN that concentrates the cycle distribution?

Entropy to Concentrate Cycles: EC-PEG Algorithm

For distribution $p = (p_1, p_2, \ldots, p_n)$, Entropy $\mathcal{H}(p) = \sum_{i=1}^{n} p_i \log \frac{1}{p_i}$

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



EC (Entropy Constrained)-PEG Algorithm

For each VN v_j

Expand Tanner Graph in a BFS fashion

If $\exists$ CNs not connected to v_j

- select a CN with min degree not connected to v_j

Else *New cycles created*

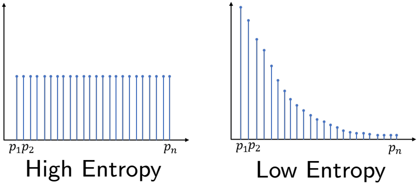
- Find CNs most distant to v_j
- Select CN that concentrates the cycle distribution

How do we select a CN that concentrates the cycle distribution?

Entropy to Concentrate Cycles: EC-PEG Algorithm

For distribution $p = (p_1, p_2, \dots, p_n)$, Entropy $\mathcal{H}(p) = \sum_{i=1}^{n} p_i \log \frac{1}{p_i}$

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



EC (Entropy Constrained)-PEG Algorithm

For each VN v_j

Expand Tanner Graph in a BFS fashion

If $\exists$ CNs not connected to v_j

- select a CN with min degree not connected to v_j

Else *New cycles created*

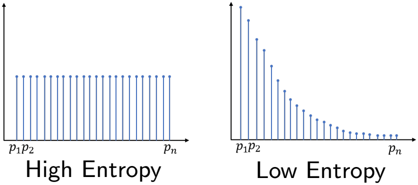
- Find CNs most distant to v_j
- Select CN that concentrates the cycle distribution

We want the cycle distributions to be concentrated

Entropy to Concentrate Cycles: EC-PEG Algorithm

For distribution $p = (p_1, p_2, \ldots, p_n)$, Entropy $\mathcal{H}(p) = \sum_{i=1}^{n} p_i \log \frac{1}{p_i}$

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



EC (Entropy Constrained)-PEG Algorithm

For each VN v_j

Expand Tanner Graph in a BFS fashion

If $\exists$ 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 concentrates the cycle distribution

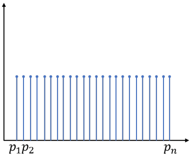
We want the cycle distributions to be concentrated

$\rightarrow$ Select CNs such that the entropy of the cycle distribution is minimized

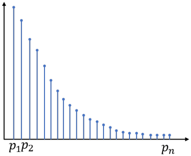
Entropy to Concentrate Cycles: EC-PEG Algorithm

For distribution $p = (p_1, p_2, \ldots, p_n)$, Entropy $\mathcal{H}(p) = \sum_{i=1}^{n} p_i \log \frac{1}{p_i}$

- 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 $\exists$ 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

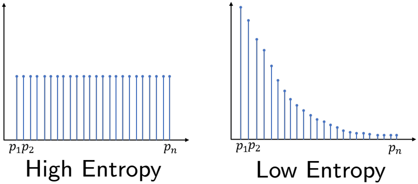
We want the cycle distributions to be concentrated

$\rightarrow$ Select CNs such that the entropy of the cycle distribution is minimized

Entropy to Concentrate Cycles: EC-PEG Algorithm

For distribution $p = (p_1, p_2, \ldots, p_n)$, Entropy $\mathcal{H}(p) = \sum_{i=1}^{n} p_i \log \frac{1}{p_i}$

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



EC (Entropy Constrained)-PEG Algorithm

For each VN v_j

Expand Tanner Graph in a BFS fashion

If $\exists$ 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

$\rightarrow$ Select 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

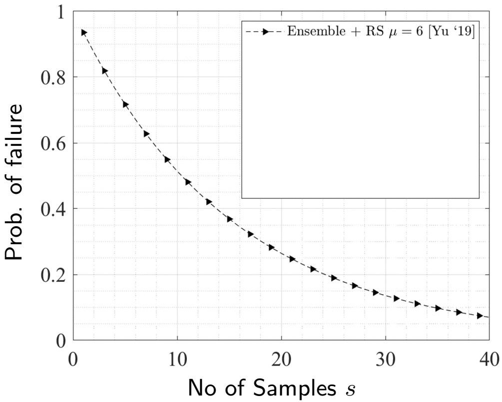
Simulation Results

Probability of failure for a stopping set of size μ

Simulation Results

Probability of failure for a stopping set of size μ

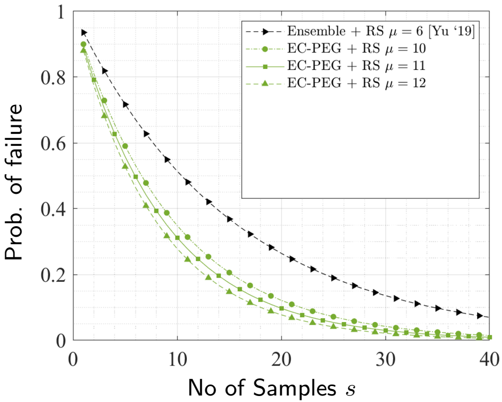
RS: Random Sampling



Simulation Results

Probability of failure for a stopping set of size μ

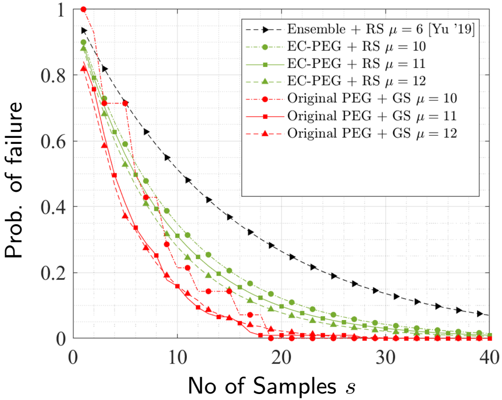
RS: Random Sampling



Simulation Results

Probability of failure for a stopping set of size μ

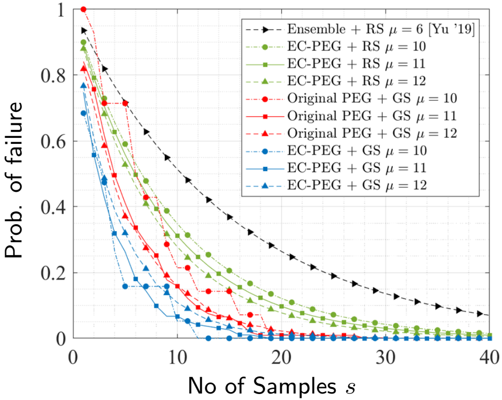
RS: Random Sampling
GS: Greedy Sampling



Simulation Results

Probability of failure for a stopping set of size μ

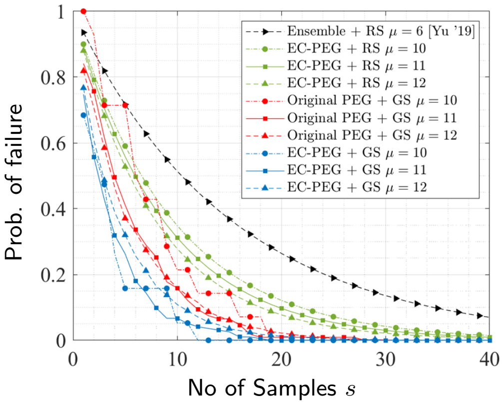
RS: Random Sampling
GS: Greedy Sampling



Simulation Results

Probability of failure for a stopping set of size μ

RS: Random Sampling
GS: Greedy Sampling



- Concentrated LDPC codes with Greedy sampling 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.
- (Xiao '05) X.Y. Hu, et al., *"Regular and irregular progressive edge-growth tanner graphs,"* IEEE Transactions of Information Theory, vol. 51, no. 1, 2005.
- (Tian '03) T. Tian, et al., *"Construction of irregular LDPC codes with low error floors,"* IEEE International Conference on Communications, May 2003.