山东大学
SHANDONG UNIVERSITY

Security-Performance Trade-off in DAG-based Proof-of-Work Blockchain Protocols

Shichen Wu, Puwen Wei, Ren Zhang, Bowen Jiang

NDSS 2024
28/02/2024

Why we still focus on PoW?

- In 585 papers presented at top CS conferences from 2020 to 2022

Why we still focus on PoW?

- In 585 papers presented at top CS conferences from 2020 to 2022
  - 41 papers focus on PoW:
    - - Formal Analysis of Nakamoto Consensus (10)
    - - New Design: DAG-based Protocols (7)
    - - New Design: non-DAG-based Protocols (6)
    - - Mining Attacks and Ecosystem Analysis (18)
  - 23 papers involve PoS:
    - - Analysis (11)
    - - New Design (12)

Why we still focus on PoW?

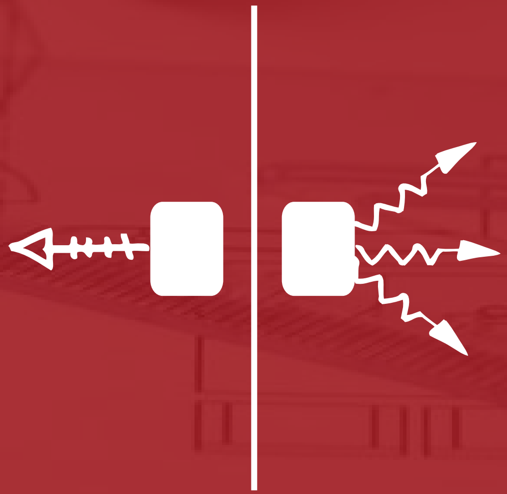
- In 585 papers presented at top CS conferences from 2020 to 2022
  - 41 papers focus on PoW:
    - - Formal Analysis of Nakamoto Consensus (10)
    - - New Design: DAG-based Protocols (7)
    - - New Design: non-DAG-based Protocols (6)
    - - Mining Attacks and Ecosystem Analysis (18)
  - 23 papers involve PoS:
    - - Analysis (11)
    - - New Design (12)
- To sum up:
  - Security Analysis
    - PoW: more secure than previously believed
    - PoS: more attack vectors discovered
  - New PoS Designs: not sure we can ever achieve PoW's security
  - PoS ecosystems: lack of studies raises concerns

1. NC & DAG

2. New Model

3. LP Attack

4. Examples & Simulation



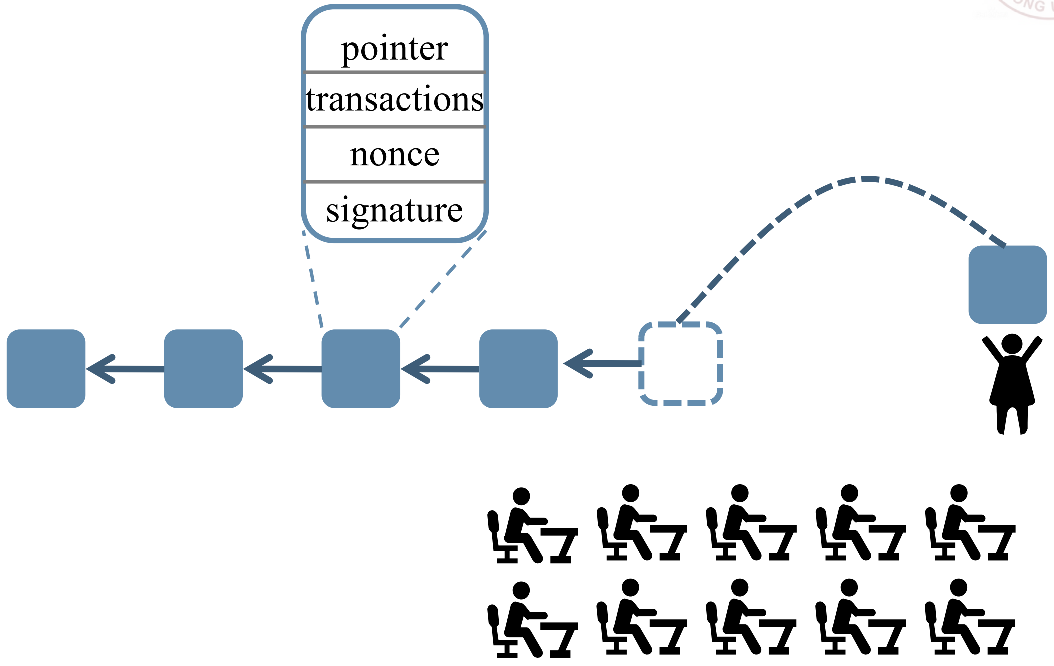
- Nakamoto Consensus and its limitation
- The solution: DAG-based blockchain
- Does DAG solve the problem?
- The phenomena in DAG blockchain

Nakamoto Consensus

- NC (Bitcoin and its variants)

Nakamoto Consensus

- NC (Bitcoin and its variants)
  - ledger: a chain of blocks
  - participants: miners

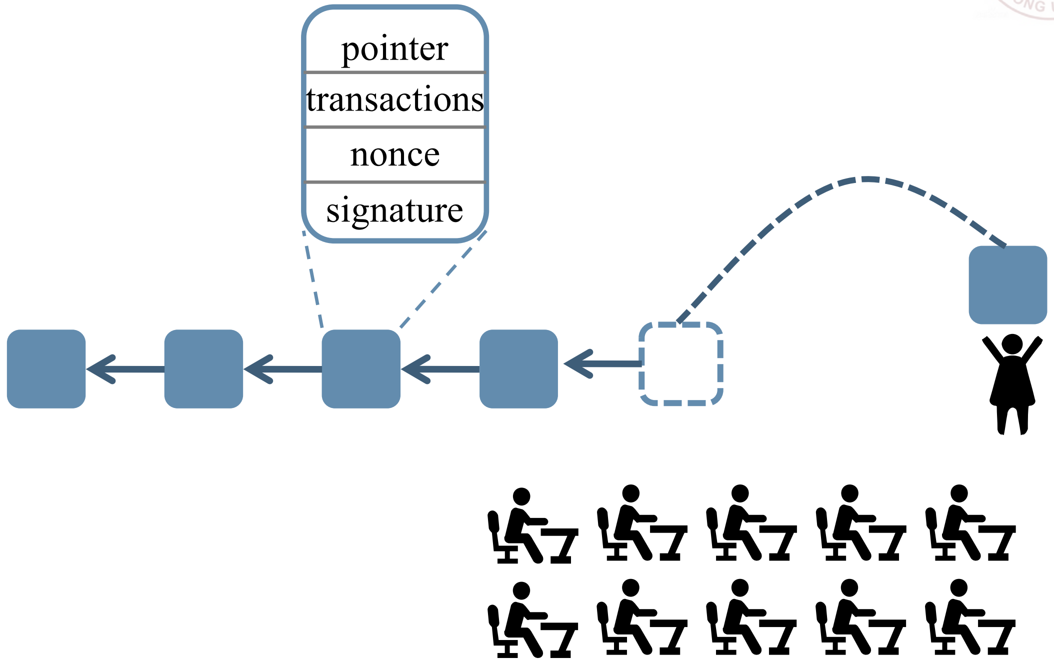


Nakamoto Consensus

- NC (Bitcoin and its variants)
  - ledger: a chain of blocks
  - participants: miners
    - generate block: Proof-of-Work

$$\boxed{Hash(pointer, tx, nonce) < \ Target}$$

change



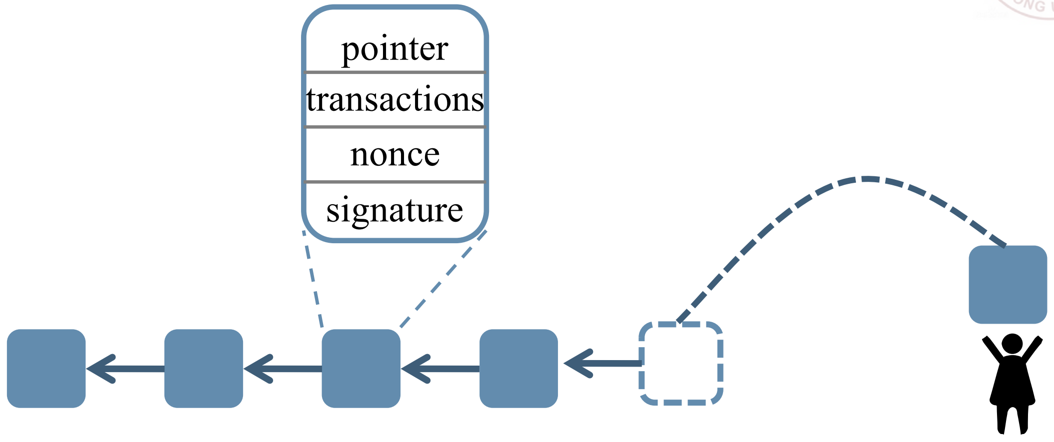
Nakamoto Consensus

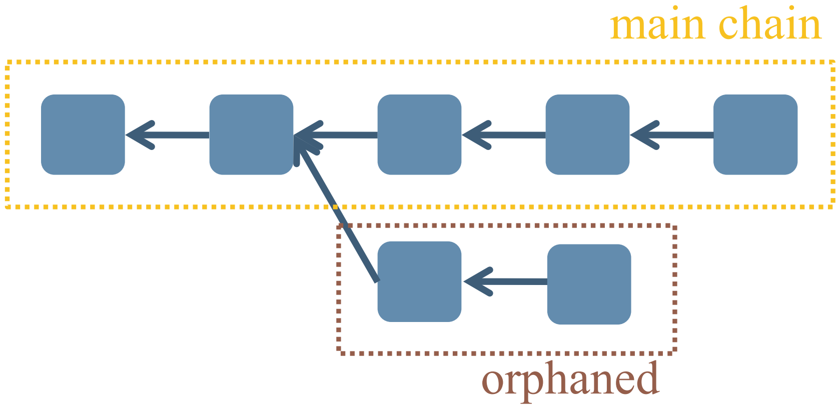
- NC (Bitcoin and its varients)
  - ledger: a chain of blocks
  - participants: miners
    - generate block: Proof-of-Work

$$\boxed{Hash(pointer, tx, nonce) < Target}$$

change

    - extend chain: Longest-Chain rule
      - the longest fork means the most mining power





Limitations of NC

- Security-Performance Tradeoff

Limitations of NC

- Security-Performance Tradeoff
  - security of NC is rooted in

"block generation interval >> the time for propagation"

- the smaller the gap, the worse the security

Y. Sompolinsky and A. Zohar, "Secure high-rate transaction processing in Bitcoin," in *Financial Cryptography and Data Security - 19th International Conference, FC 2015*, ser. Lecture Notes in Computer Science, vol. 8975. Springer, 2015, pp. 507–527.

J. A. Garay, A. Kiayias, and N. Leonardos, "The bitcoin backbone protocol: Analysis and applications," in *Advances in Cryptology - EUROCRYPT 2015 - 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, ser. Lecture Notes in Computer Science, vol. 9057. Springer, 2015, pp. 281–310.

——, "The bitcoin backbone protocol with chains of variable difficulty," in *Advances in Cryptology - CRYPTO 2017 - 37th Annual International Cryptology Conference*, ser. Lecture Notes in Computer Science, vol. 10401. Springer, 2017, pp. 291–323.

P. Gaži, A. Kiayias, and A. Russell, "Tight consistency bounds for bitcoin," in *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '20. ACM, 2020, p. 819–838.

R. Pass, L. Seeman, and A. Shelat, "Analysis of the blockchain protocol in asynchronous networks," in *Advances in Cryptology - EUROCRYPT 2017 - 36th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, ser. Lecture Notes in Computer Science, vol. 10211, 2017, pp. 643–673.

A. Dembo, S. Kannan, E. N. Tas, D. Tse, P. Viswanath, X. Wang, and O. Zeitouni, "Everything is a race and Nakamoto always wins," in *CCS '20: 2020 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2020, pp. 859–878.

L. Kiffer, R. Rajaraman, and A. Shelat, "A better method to analyze blockchain consistency," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, CCS 2018*, pp. 729–744.

Limitations of NC

- **Security-Performance Tradeoff**
  - security of NC is rooted in

"block generation interval >> the time for propagation"

  - the smaller the gap, the worse the security

Y. Sompolinsky and A. Zohar, "Secure high-rate transaction processing in Bitcoin," in *Financial Cryptography and Data Security - 19th International Conference, FC 2015*, ser. Lecture Notes in Computer Science, vol. 8975. Springer, 2015, pp. 507–527.

J. A. Garay, A. Kiayias, and N. Leonardos, "The bitcoin backbone protocol: Analysis and applications," in *Advances in Cryptology - EUROCRYPT 2015 - 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, ser. Lecture Notes in Computer Science, vol. 9057. Springer, 2015, pp. 281–310.

——, "The bitcoin backbone protocol with chains of variable difficulty," in *Advances in Cryptology - CRYPTO 2017 - 37th Annual International Cryptology Conference*, ser. Lecture Notes in Computer Science, vol. 10401. Springer, 2017, pp. 291–323.

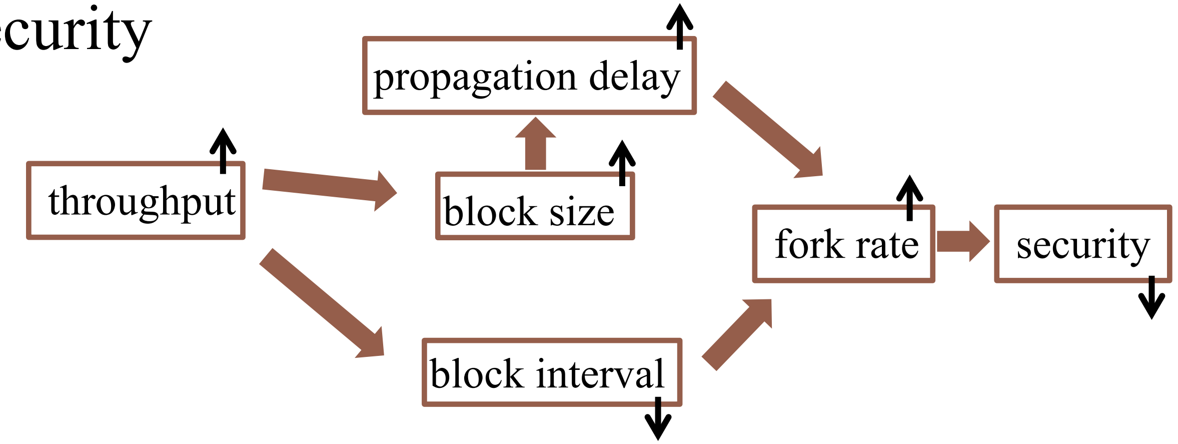
P. Gaži, A. Kiayias, and A. Russell, "Tight consistency bounds for bitcoin," in *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '20. ACM, 2020, p. 819–838.

R. Pass, L. Seeman, and A. Shelat, "Analysis of the blockchain protocol in asynchronous networks," in *Advances in Cryptology - EUROCRYPT 2017 - 36th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, ser. Lecture Notes in Computer Science, vol. 10211, 2017, pp. 643–673.

A. Dembo, S. Kannan, E. N. Tas, D. Tse, P. Viswanath, X. Wang, and O. Zeitouni, "Everything is a race and Nakamoto always wins," in *CCS '20: 2020 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2020, pp. 859–878.

L. Kiffer, R. Rajaraman, and A. Shelat, "A better method to analyze blockchain consistency," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, CCS 2018*, pp. 729–744.

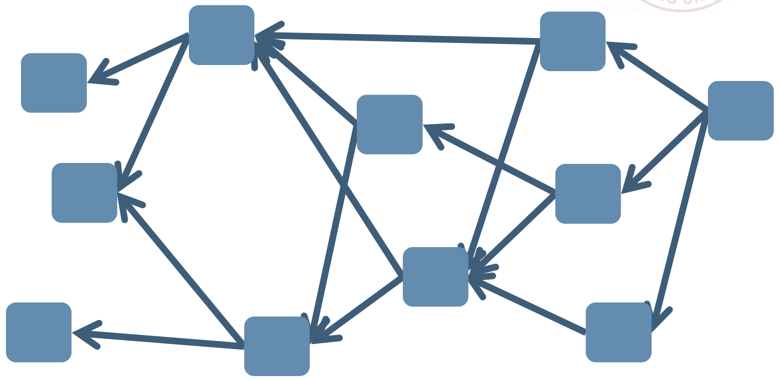
  - however!
  - higher throughput requires larger block and shorter block interval, which reduces the security



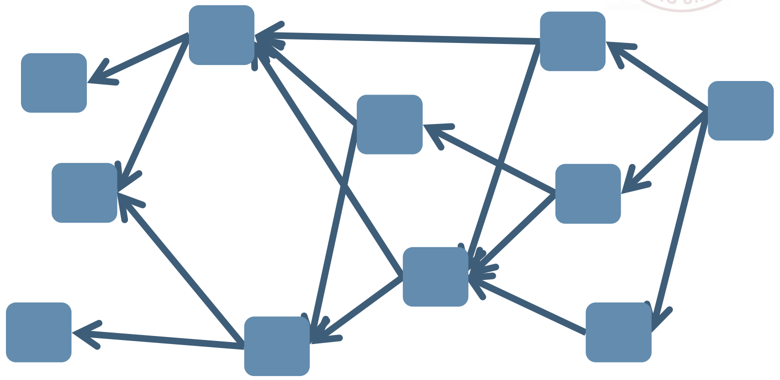
- **NC has to maintain a poor performance.**
  - 7 TPS

DAG-based Blockchain

- **Structure:** Chain → Directed Acyclic Graph



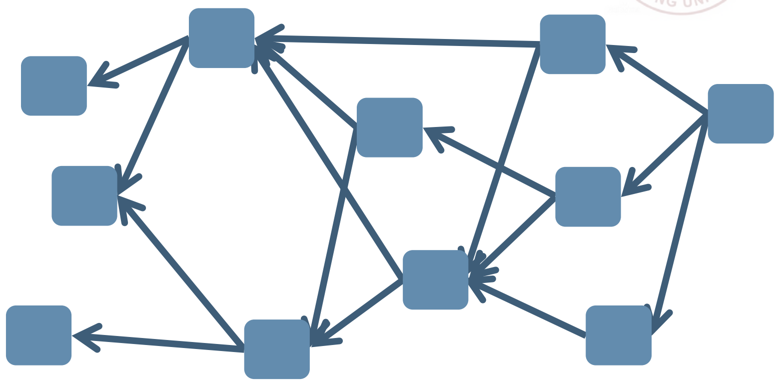
DAG-based Blockchain



- **Structure:** Chain → Directed Acyclic Graph
  - ➢ multiple predecessors
  - ➢ multiple concurrent blocks

- A large number of valid blocks result in a high throughput （thousands TPS）

DAG-based Blockchain



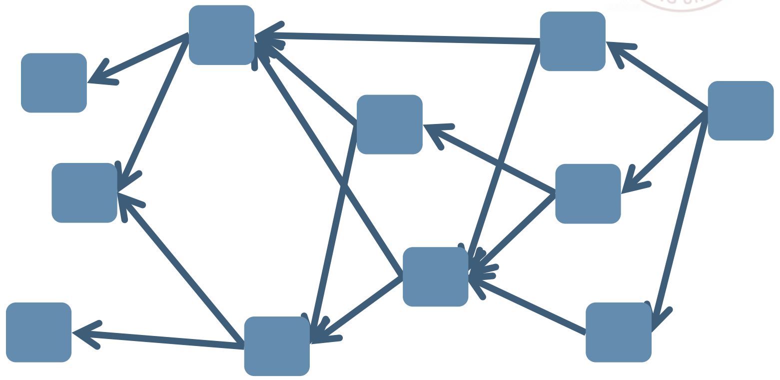
- **Structure:** Chain → Directed Acyclic Graph
  - multiple predecessors
  - multiple concurrent blocks

- A large number of valid blocks result in a high throughput （thousands TPS）

- Security is a concern for early protocols
  - weak security guarantees
    - Inclusive, Meshcash
  - partial security analyses
    - SPECTRE, PHANTOM, Conflux

DAG-based Blockchain

- **Structure:** Chain → Directed Acyclic Graph
  - multiple predecessors
  - multiple concurrent blocks



- A large number of valid blocks result in a high throughput （thousands TPS）

- Security is a concern for early protocols
  - weak security guarantees
    - Inclusive, Meshcash
  - partial security analyses
    - SPECTRE, PHANTOM, Conflux

State-of-the-art:

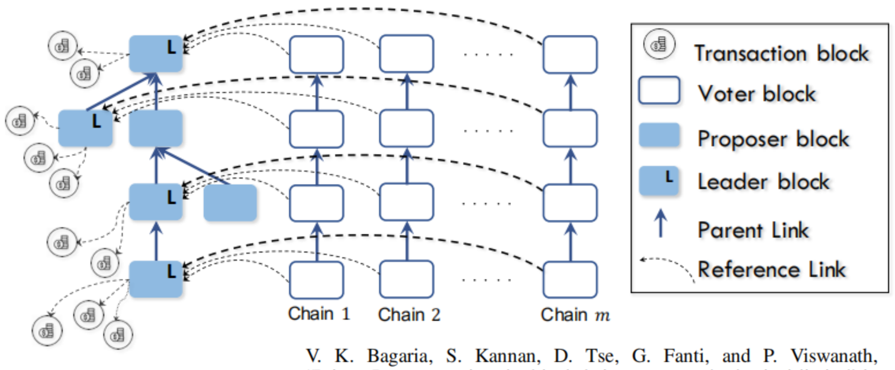
Prism (CCS’ 2019), OHIE (S&P 2020)

Prism & OHIE

- Structured DAG blockchain based on NC

Prism & OHIE

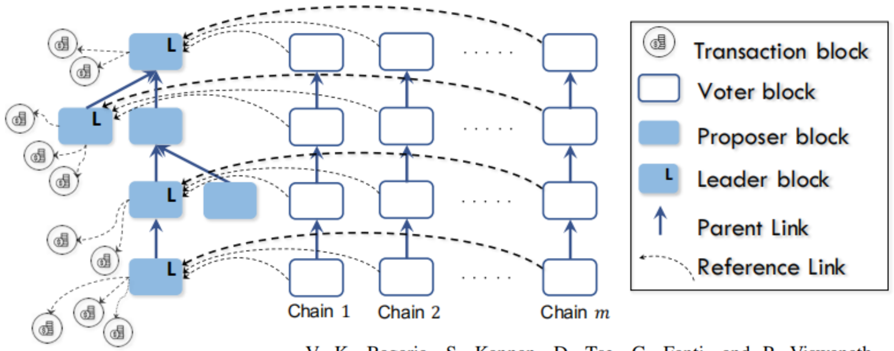
- Structured DAG blockchain based on NC
  - Prism [CCS'19] (three types of blocks)
    - tx blocks, proposer blocks, voter blocks



V. K. Bagaria, S. Kannan, D. Tse, G. Fanti, and P. Viswanath, "Prism: Deconstructing the blockchain to approach physical limits," in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, CCS 2019.* ACM, 2019, pp. 585–602.

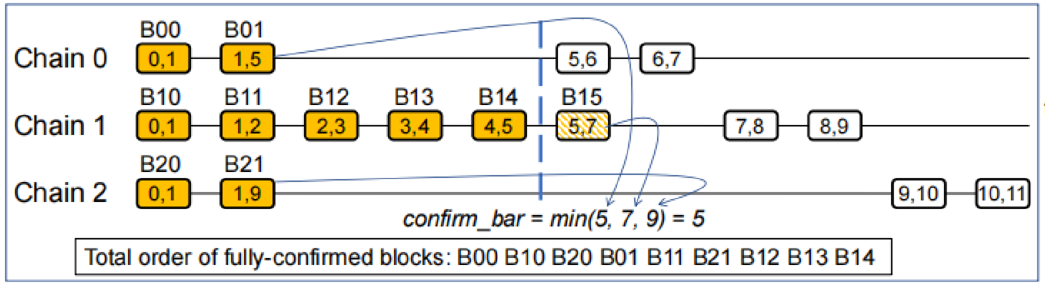
Prism & OHIE

- Structured DAG blockchain based on NC
  - Prism [CCS'19] (three types of blocks)
    - tx blocks, proposer blocks, voter blocks



V. K. Bagaria, S. Kannan, D. Tse, G. Fanti, and P. Viswanath, "Prism: Deconstructing the blockchain to approach physical limits," in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, CCS 2019.* ACM, 2019, pp. 585–602.

  - OHIE [S&P'20] (multiple parallel chains)
    - m parallel NC chains, m times throughput
    - security comparable to NC



H. Yu, I. Nikolic, R. Hou, and P. Saxena, "OHIE: blockchain scaling made simple," in *2020 IEEE Symposium on Security and Privacy, SP 2020.* IEEE, 2020, pp. 90–105.

DAG Breaks Trade-off

- Security-Performance tradeoff has been broken
  - Prism and OHIE achieve 90% and 50% bandwidth utilization
  - Both designs prove the same security properties as NC

DAG Breaks Trade-off

- Security-Performance tradeoff has been broken
  - Prism and OHIE achieve 90% and 50% bandwidth utilization
  - Both designs prove the same security properties as NC

- Security-Performance tradeoff really has been broken?

?

Problems of analyses for DAG-based blockchain

- Assumption of Decoupling
  - some priority blocks are small enough and enjoy a priority propagation policy
    - delay is always very small
    - always accept immediately

Problems of analyses for DAG-based blockchain

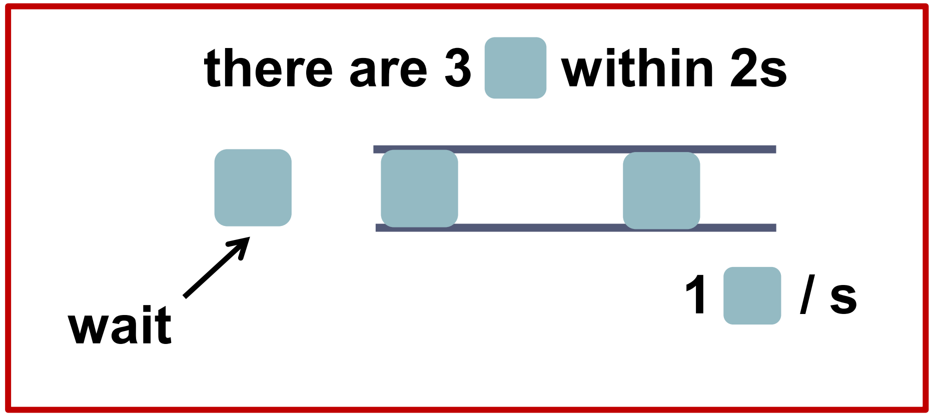
- **Assumption of Decoupling**
  - some priority blocks are small enough and enjoy a priority propagation policy
    - delay is always very small
    - always accept immediately
  - Security will be guaranteed if these priority blocks can always be "synchronized quickly"

Problems of analyses for DAG-based blockchain

- Assumption of Decoupling
  - some priority blocks are small enough and enjoy a priority propagation policy
    - delay is always very small
    - always accept immediately
  - Security will be guaranteed if these priority blocks can always be "synchronized quickly"
  - But it's not easy in a high-throughput DAG-based blockchain system

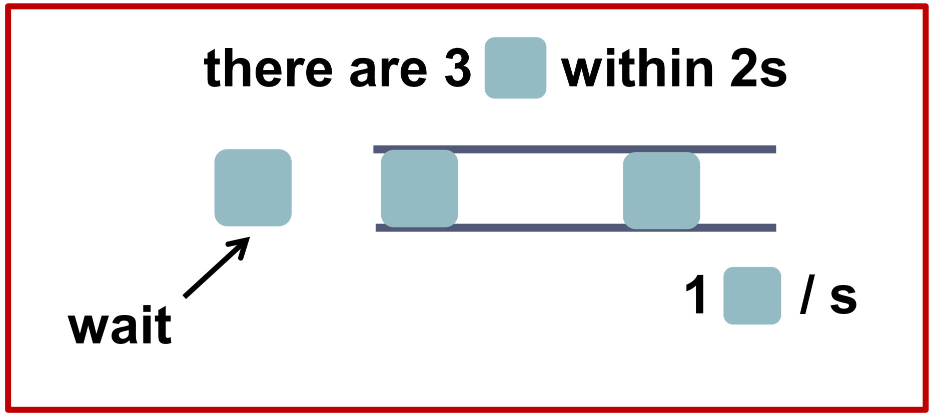
Block Jam

- If the total number of blocks propagated over a period of time **exceeds** the network's processing capacity, some blocks will have more propagation delay



Block Jam

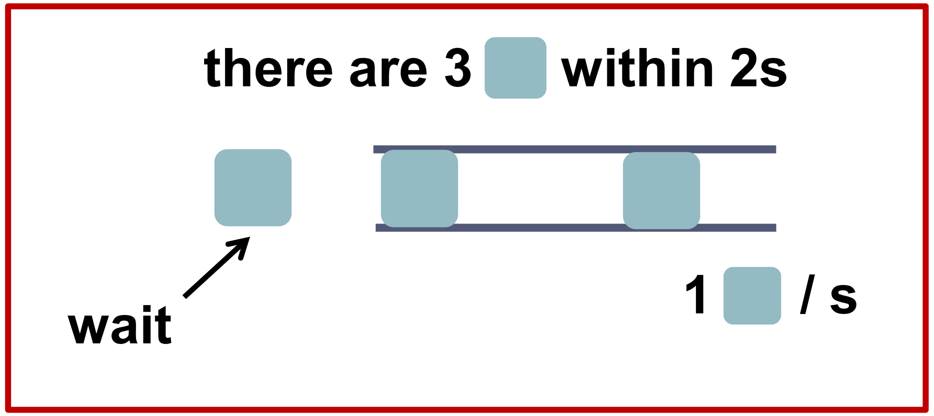
- If the total number of blocks propagated over a period of time **exceeds** the network's processing capacity, some blocks will have more propagation delay



- In DAG-based blockchain system
  - many blocks generated parallelly
  - network loads many blocks→block propagation delay vary and increases

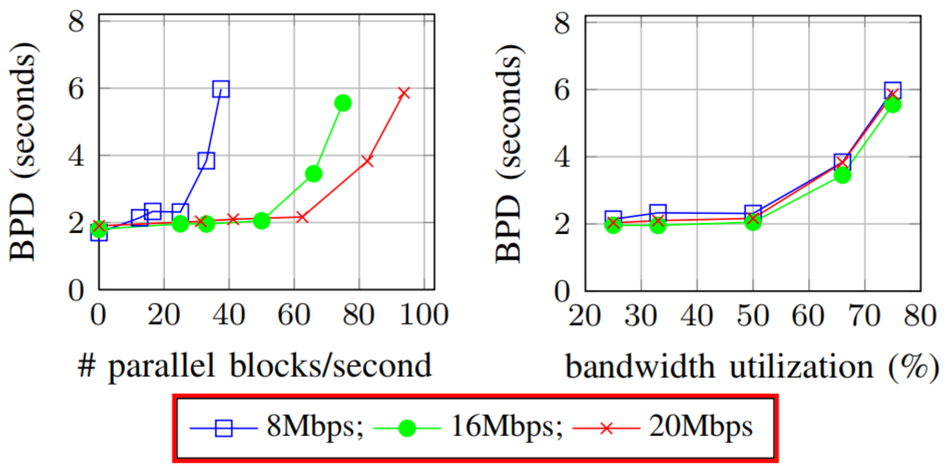
Block Jam

- If the total number of blocks propagated over a period of time **exceeds** the network's processing capacity, some blocks will have more propagation delay



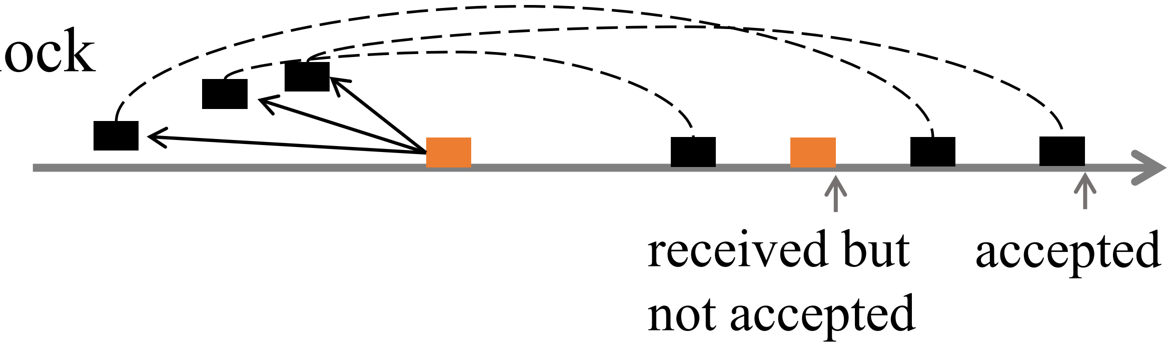
- In DAG-based blockchain system
  - many blocks generated parallelly
  - network loads many blocks → block propagation delay varies and increases

H. Yu, I. Nikolic, R. Hou, and P. Saxena, "OHIE: blockchain scaling made simple," in *2020 IEEE Symposium on Security and Privacy, SP 2020*. IEEE, 2020, pp. 90–105.



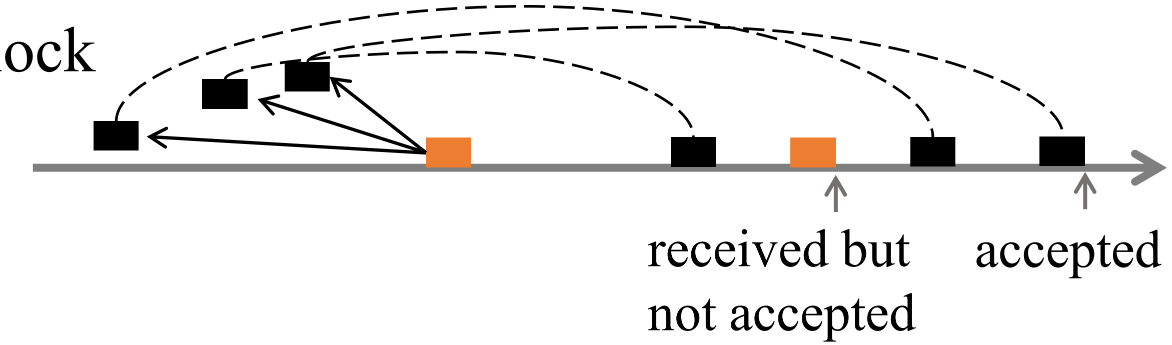
Late Predecessor

- If a miner receives a block but does not receive all its predecessors, the miner cannot accept the block



Late Predecessor

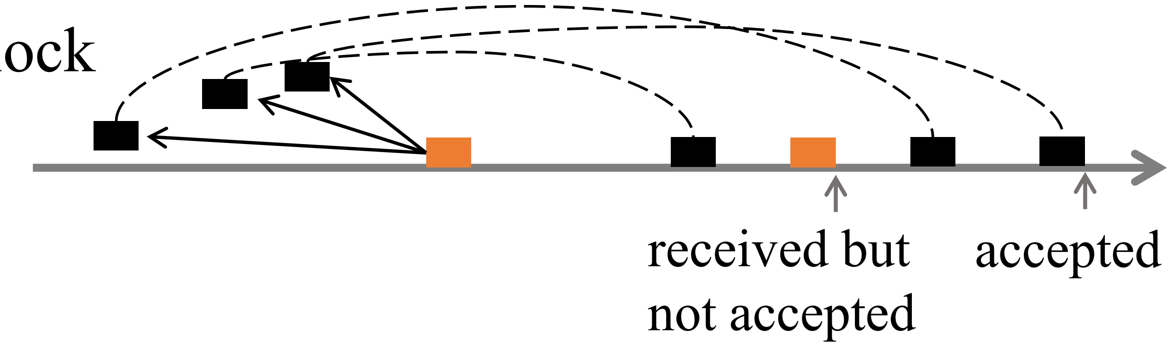
- If a miner receives a block but does not receive all its predecessors, the miner cannot accept the block



- The more pointers a block has, the more late predecessors it will have.

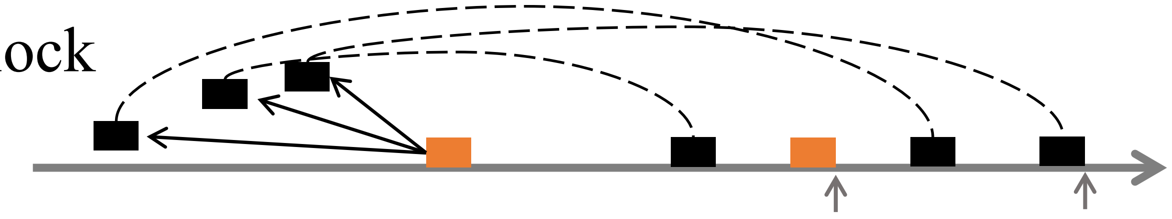
Late Predecessor

- If a miner receives a block but does not receive all its predecessors, the miner cannot accept the block
- The more pointers a block has, the more late predecessors it will have.
- Late predecessor phenomenon is common in DAG-based protocols, but it has been overlooked in previous analyses



Late Predecessor

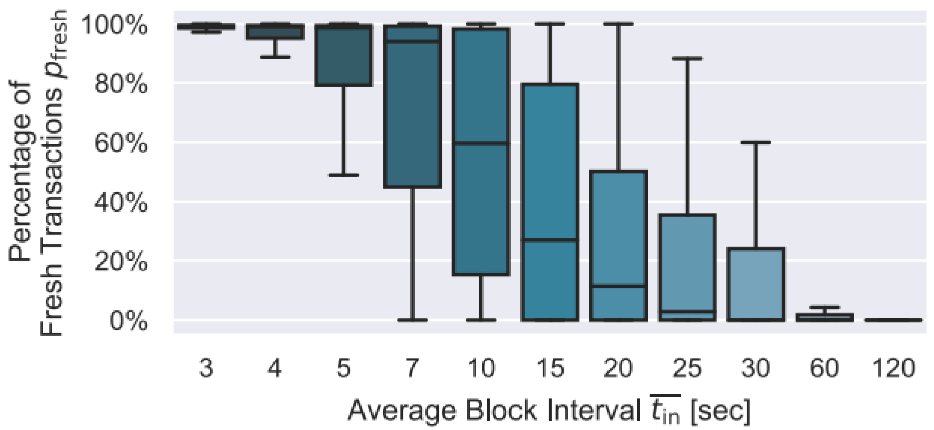
- If a miner receives a block but does not receive all its predecessors, the miner cannot accept the block



- The more pointers a block has, the more late predecessors it will have.

R. Zhang, D. Zhang, Q. Wang, S. Wu, J. Xie, and B. Preneel, "NC-Max: Breaking the security-performance tradeoff in Nakamoto consensus," *The Network and Distributed System Security (NDSS) Symposium*, 2022.

- Late predecessor phenomenon is common in DAG-based protocols, but it has been overlooked in previous analyses



1. NC & DAG

2. New Model

3. LP Attack

4. Examples & Simulation

- Why we need a new model?
- Characteristics of CBM
- Apply CBM to DAG-based blockchain

Why we need a new model ?

Why we need a new model ?

- For DAG-based blockchain
  - multiple types of blocks
  - overlaps in block propagation

delay is complex and diverse

Why we need a new model ?

- For DAG-based blockchain
  - multiple types of blocks
  - overlaps in block propagation

delay is complex and diverse

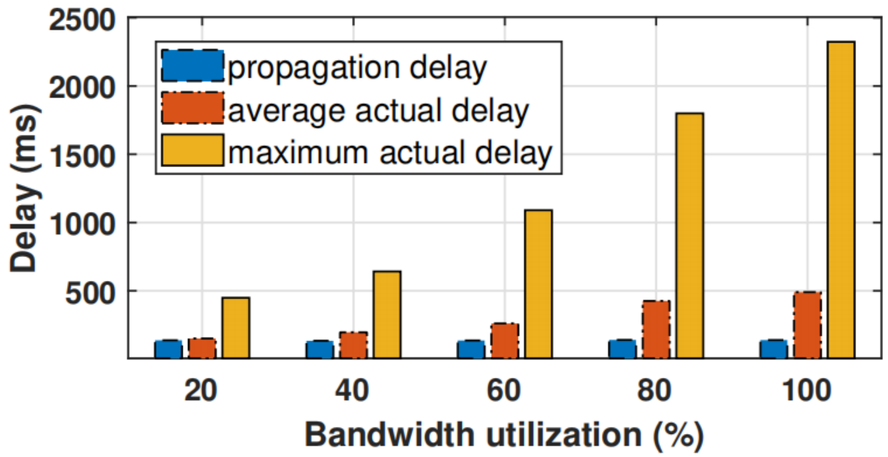
- UDBM
  - a uniform upper bound of delay on all blocks
  - adversary strategy: delay all receivers to the bound

Why we need a new model ?

- For DAG-based blockchain
  - multiple types of blocks
  - overlaps in block propagation

delay is complex and diverse

Delay of proposer blocks



*actual delay is the interval between the block's generation and the arrival of its latest predecessor at a certain node

- UDBM
  - a uniform upper bound of delay on all blocks
  - adversary strategy: delay all receivers to the bound

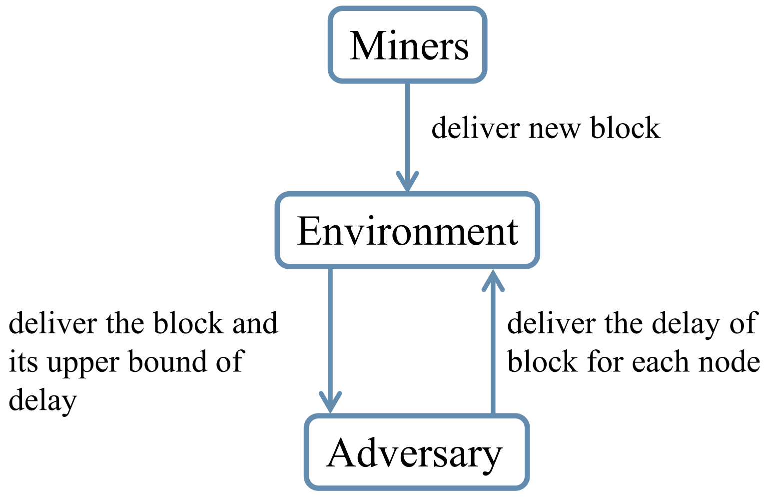
- We deploy Prism with SimBlock
  - a maximum delay bound would overestimate the security requirement

Congestible Blockchain Model

- CBM

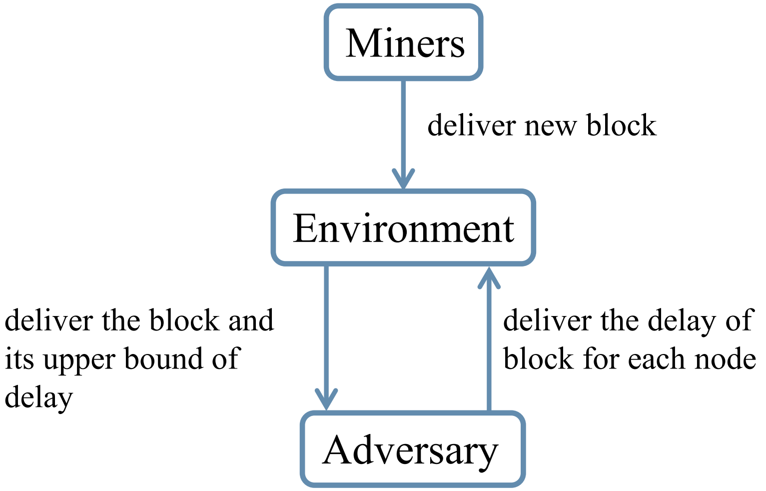
Congestible Blockchain Model

- CBM
  - time assumption
    - the upper bound may be different
    - specify delay of each node

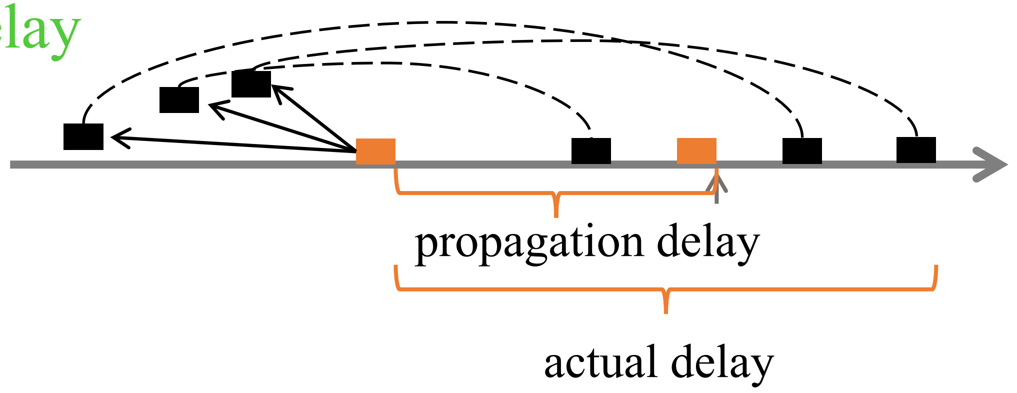


Congestible Blockchain Model

- CBM
  - time assumption
    - the upper bound may be different
    - specify delay of each node



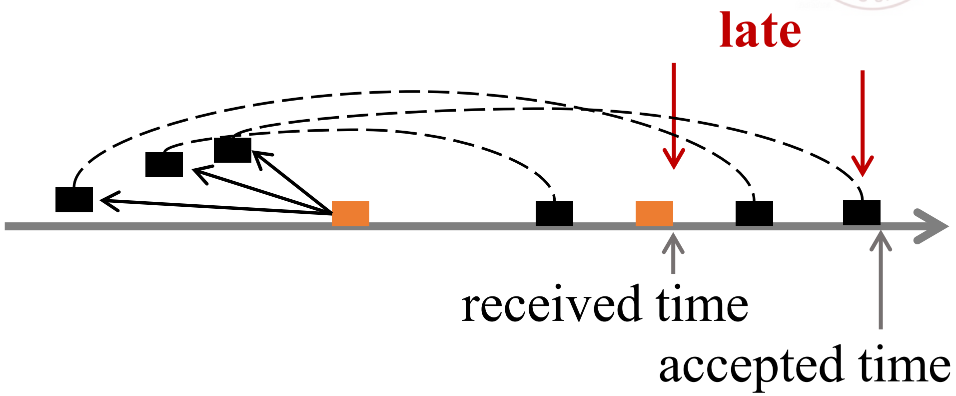
  - block processing: distinguish the status
    - received, accepted, confirmed, orphaned
    - actual delay = propagation delay + processing delay



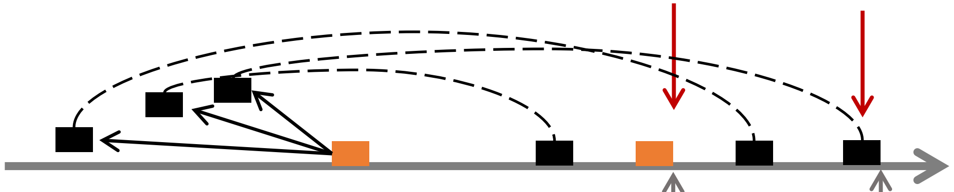
  - same adversary & mining & security property

Apply CBM to DAG-based blockchain

- Defining late-predecessors (LP)
  - B^* is late if $B^* \leftarrow B^+$, but B^+ is received first
  - interval between receiving B^* and B^+ is lag time



Apply CBM to DAG-based blockchain



- Defining late-predecessors (LP)
  - B^* is late if $B^* \leftarrow B^+$, but B^+ is received first
  - interval between receiving B^* and B^+ is lag time

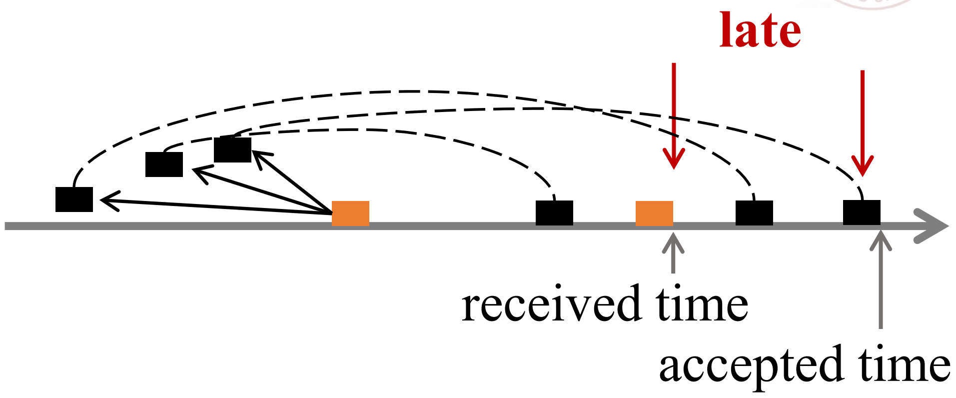
- Bounding the Actual Delay
  - max actual delay =

    propagation delay (itself) + max lag time (predecessors)

    $\leq$ max propagation delay (predecessors)

Apply CBM to DAG-based blockchain

- Defining late-predecessors (LP)
  - B^* is late if $B^* \leftarrow B^+$, but B^+ is received first
  - interval between receiving B^* and B^+ is lag time

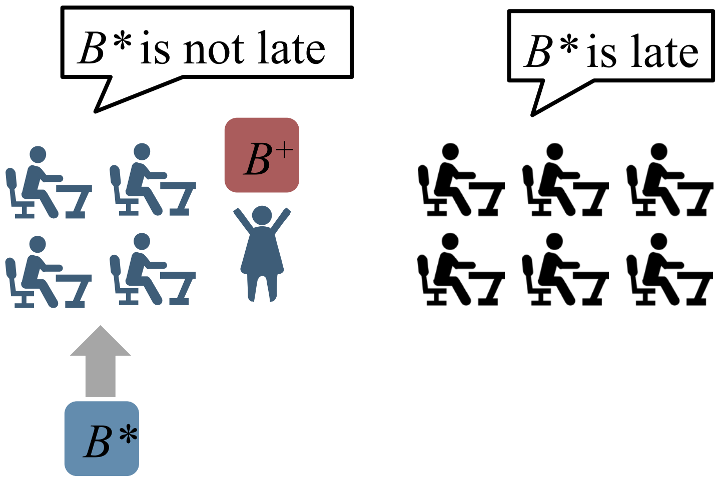


- Bounding the Actual Delay
  - max actual delay =

    propagation delay (itself) + max lag time (predecessors)

    $\leq$ max propagation delay (predecessors)

  - Max actual delay cannot be reached for all nodes



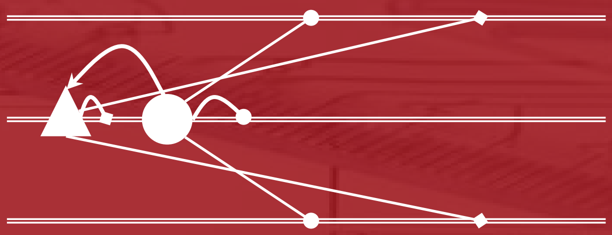
- Only the maximum actual delay is insufficient

1. NC & DAG

2. New Model

3. LP Attack

4. Examples & Simulation



- Adversary’s capability and target
- Simple case: one predecessor
- General case: Concrete attack strategy
- Results and security analysis

Defining the attacker's utility

- Consider two group of blocks:

Defining the attacker's utility

- Consider two group of blocks:

  potential late predecessor B^*

  - large and many, such as transaction blocks

Defining the attacker's utility

- Consider two group of blocks:

  potential late predecessor B^*

  - large and many, such as transaction blocks

  affected block B^+

  - has small delay, such as proposer blocks

$$\boxed{\delta^*_{max} > \delta^+_{max}}$$

Defining the attacker's utility

- Consider two groups of blocks:

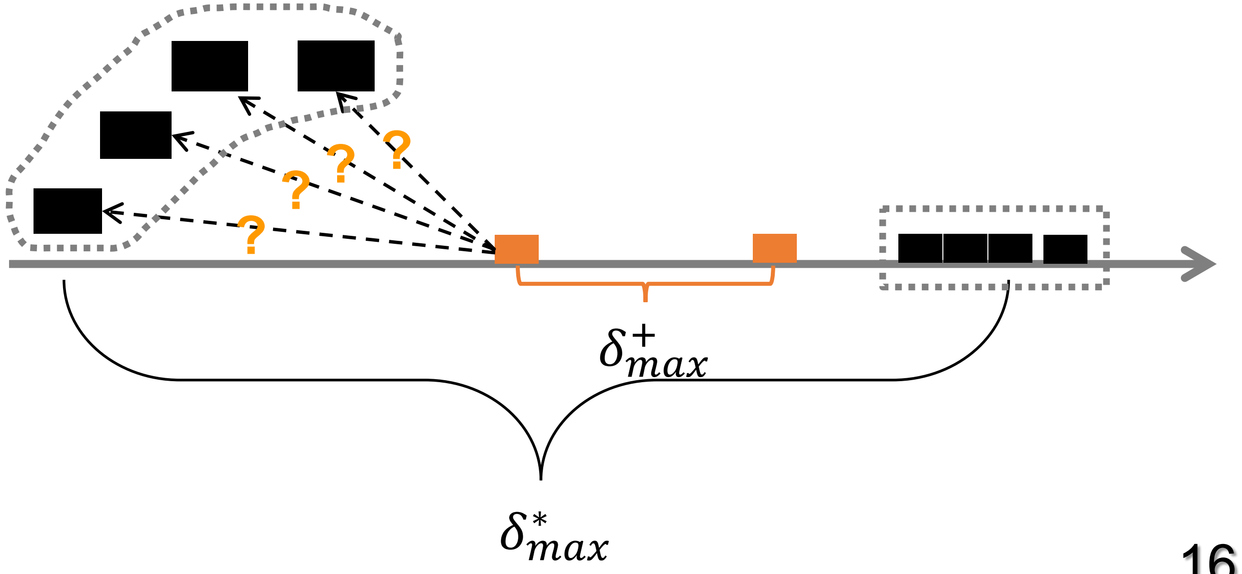
  potential late predecessor B^*

  ➢ large and many, such as transaction blocks

  affected block B^+

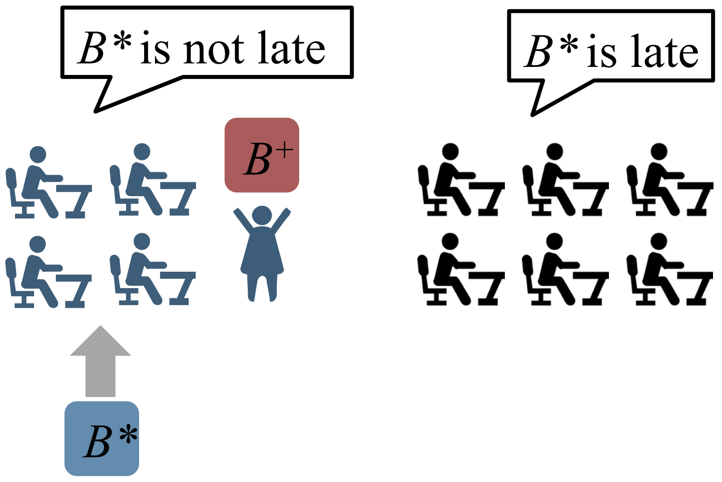
  ➢ have small delays, such as proposer blocks

$$\boxed{\delta_{max}^* > \delta_{max}^+}$$



Defining the attacker's utility

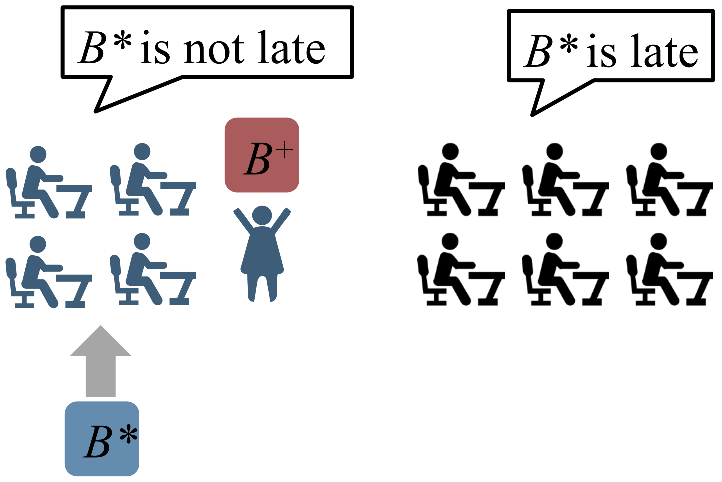
- Adversary's target
  - average actual delay of B^+
    - the average delay of a block accepted by every node
    - reflects the wasted computing power



Defining the attacker's utility

- Adversary's target
  - average actual delay of B^+
    - the average delay of a block accepted by every node
    - reflects the wasted computing power

- Since block mining process is random and unpredictable, adversary maximizes the expectation.



Propagating one potential Late-predecessor

- Maximize the “damage” of one potential LP

Propagating one potential Late-predecessor

- Maximize the "damage" of one potential LP

given B^* (mined earlier) and B^+

- the probability of $B^* \leftarrow B^+$
- the lag time of B^* and B^+ for each node

Propagating one potential Late-predecessor

- Maximize the "damage" of one potential LP

given B^* (mined earlier) and B^+

  - the probability of $B^* \leftarrow B^+$
  - the lag time of B^* and B^+ for each node

- Optimal strategy:

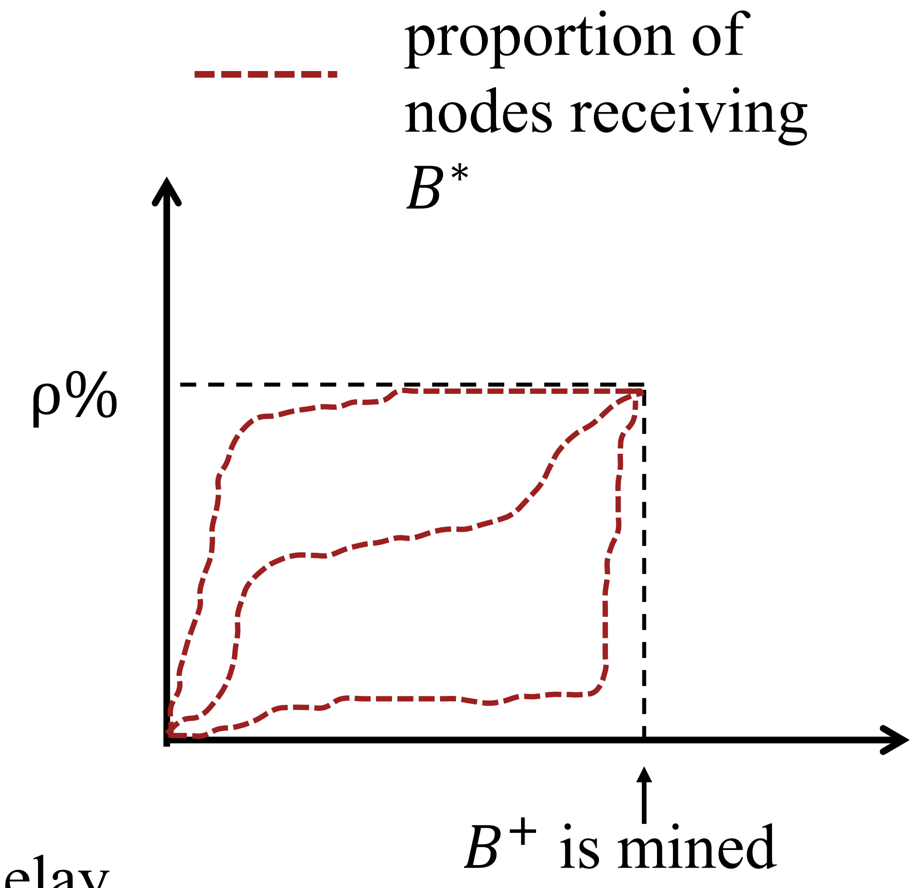
  - some nodes (ρ%) receive B^* before B^+ is mined
  - other nodes receive B^* at the maximum propagation delay

Propagating one potential Late-predecessor

- Maximize the "damage" of one potential LP

given B^* (mined earlier) and B^+

  - the probability of $B^* \leftarrow B^+$
  - the lag time of B^* and B^+ for each node



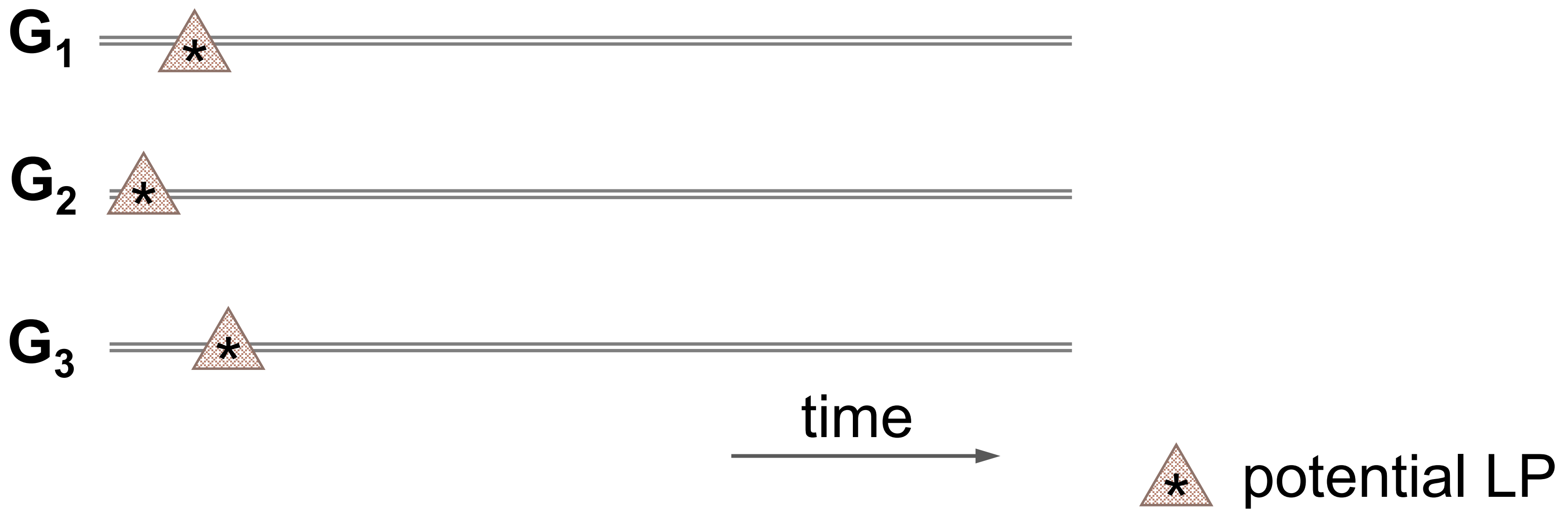
- Optimal strategy:
  - some nodes (ρ%) receive B^* before B^+ is mined
  - other nodes receive B^* at the maximum propagation delay

Consider a sequence of potential B^*

- maximize the probability of LP appearing: each node has a potential B^*

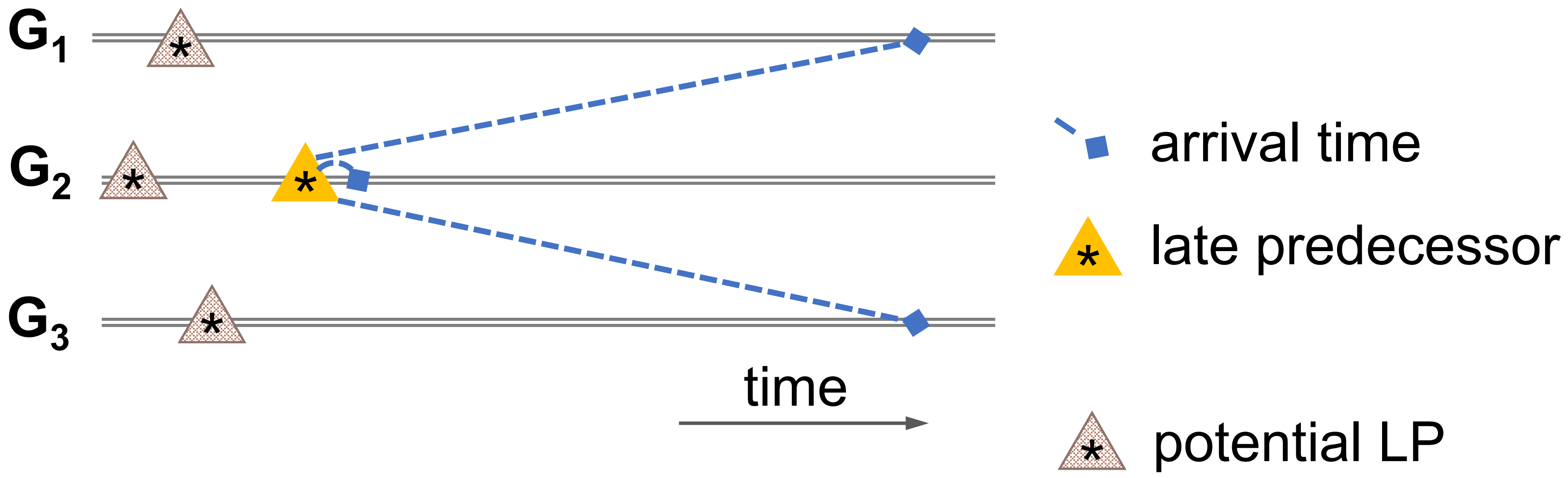
Consider a sequence of potential B^*

- maximize the probability of LP appearing: each node has a potential B^*



Consider a sequence of potential B^*

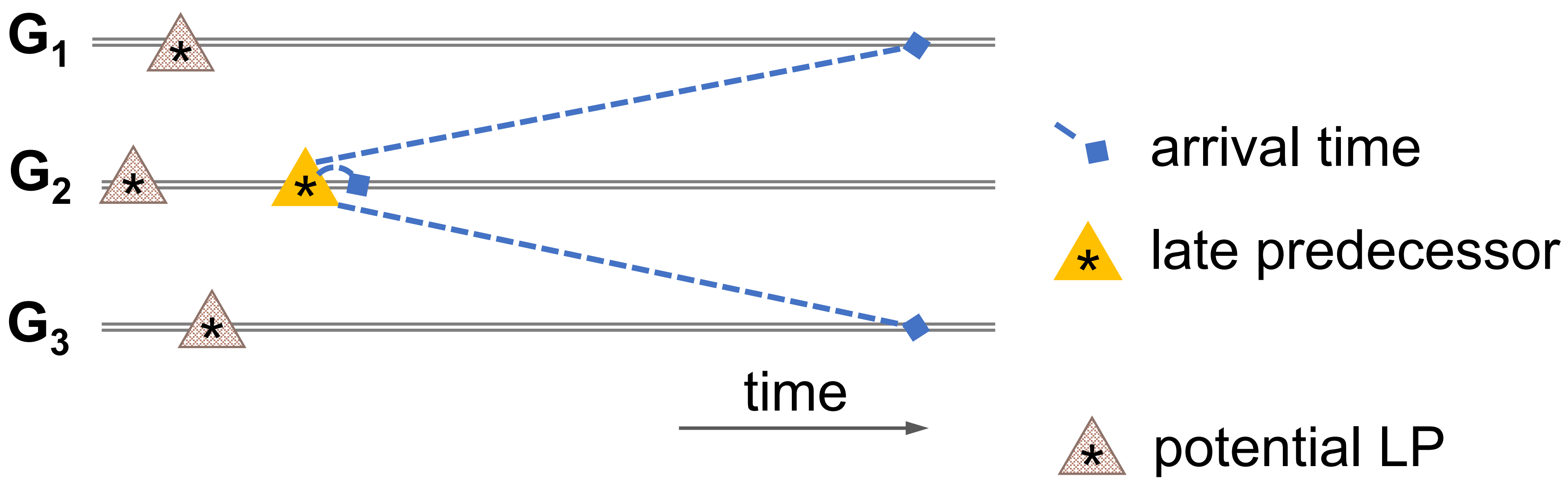
➢ maximize the probability of LP appearing: each node has a potential B^*



Consider a sequence of potential B^*

- maximize the probability of LP appearing: each node has a potential B^*
- increase lag time: shorten the interval to obtain B^*
- increase number of affected nodes: reduce group size

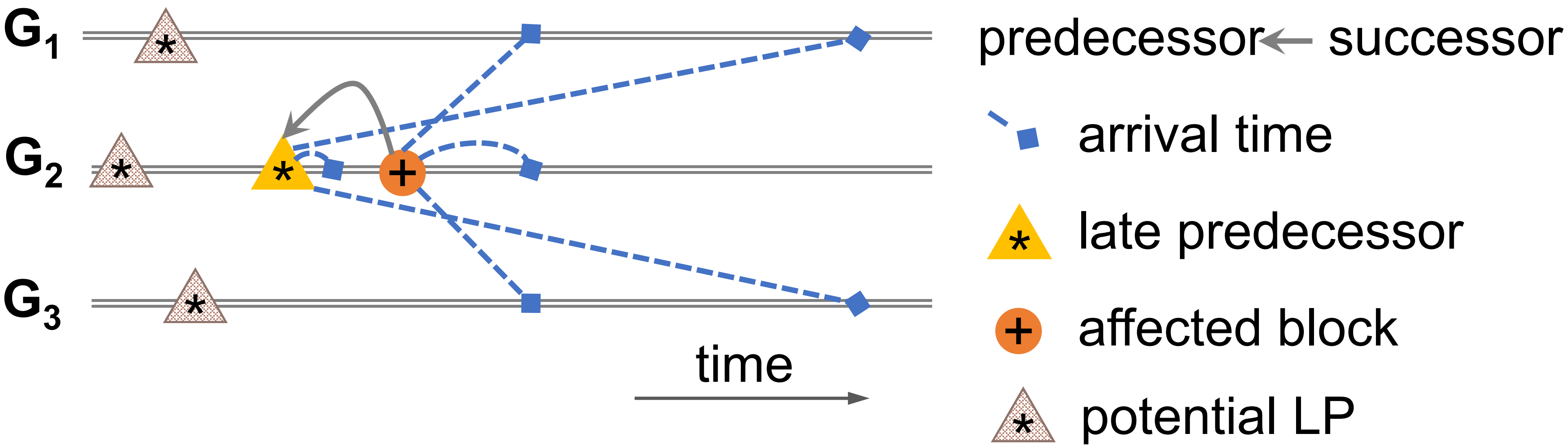
trade-off



Consider a sequence of potential B^*

- maximize the probability of LP appearing: each node has a potential B^*
- increase lag time: shorten the interval to obtain B^*
- increase number of affected nodes: reduce group size

trade-off



Attack Results

- Optimal ***s***

TABLE I: The optimal s that maximizes $\mathbb{E}[\Delta^+]$, where k is the expected number of in-propagation blocks in $\mathcal{B}^*$ in a round.

k	(0.5,2.53)	[2.54,9.81)	[9.82,18.64)	[18.65,20]
s	2	3	4	5

- Computing the result

$$\boxed{\mathbb{E}[\overline{\Delta}^+] = \delta^+_{\max} + (1 - 1/s)(k - s(1-\omega))f^*}$$

$$k = f^* \cdot (\delta^*_{\max} - \delta^+_{\max}) \qquad \omega = (1 - f^*/s)^{\delta^*_{\max} - \delta^+_{\max}}$$

Attack Results

- Optimal *s*

TABLE I: The optimal s that maximizes $\mathbb{E}[\Delta^+]$, where k is the expected number of in-propagation blocks in $\mathcal{B}^*$ in a round.

k	(0.5,2.53)	[2.54,9.81)	[9.82,18.64)	[18.65,20]
s	2	3	4	5

- Computing the result

$$\boxed{\mathbb{E}[\overline{\Delta}^+] = \delta^+_{\max} + (1 - 1/s)(k - s(1-\omega))f^*}$$

$$k = f^* \cdot (\delta^*_{\max} - \delta^+_{\max}) \qquad \omega = (1 - f^*/s)^{\delta^*_{\max} - \delta^+_{\max}}$$

- longer propagation delay of LP
- higher generation rate of LP

⊳ longer actual delays

Security Properties in the Presence of an LP Attacker

- As nodes have different delays for the same late predecessor, we cannot replace the delay in existing UDBM analyses.

- Chain growth
  - using average actual delay to compute discounted computing power

- Chain quality
  - comparing the discounted chain growth with the adversary’s computing power

- Common prefix
  - probability of splitting nodes to work on two distinct chains with different block delays

Security Properties in the Presence of an LP Attacker

- As nodes have different delays for the same late predecessor, we cannot replace the delay in existing UDBM analyses.

- Chain growth
  - using average actual delay to compute discounted computing power
- Chain quality
  - comparing the discounted chain growth with the adversary's computing power
- Common prefix
  - probability of splitting nodes to work on two distinct chains with different block delays

higher average actual delay
leads to
lower security level

1. NC & DAG

2. New Model

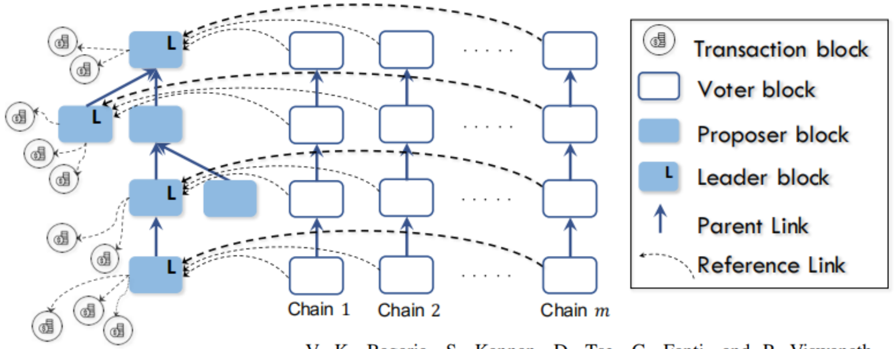
3. LP Attack

4. Examples & Simulation

- Prism’s security-performance trade-off
- OHIE’s security-performance trade-off
- Simulation of Prism and OHIE

Prism's security-performance trade-off

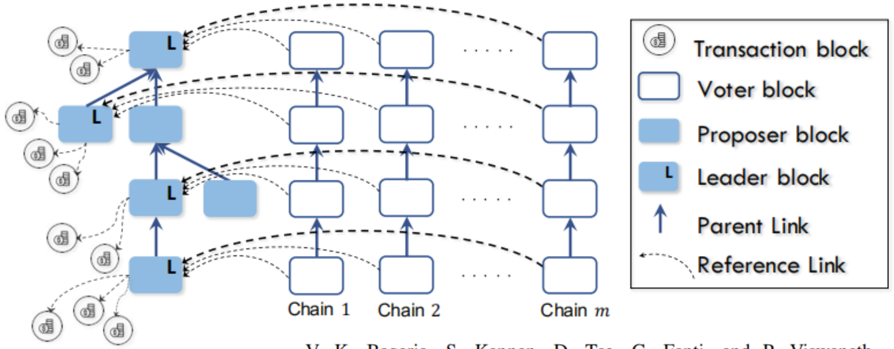
- Original paper of Prism claims that
  - changing the parameters of transaction blocks (size and rate) doesn't affect the security



V. K. Bagaria, S. Kannan, D. Tse, G. Fanti, and P. Viswanath, "Prism: Deconstructing the blockchain to approach physical limits," in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, CCS 2019.* ACM, 2019, pp. 585–602.

Prism's security-performance trade-off

- Original paper of Prism claims that
  - changing the parameters of transaction blocks (size and rate) doesn't affect the security



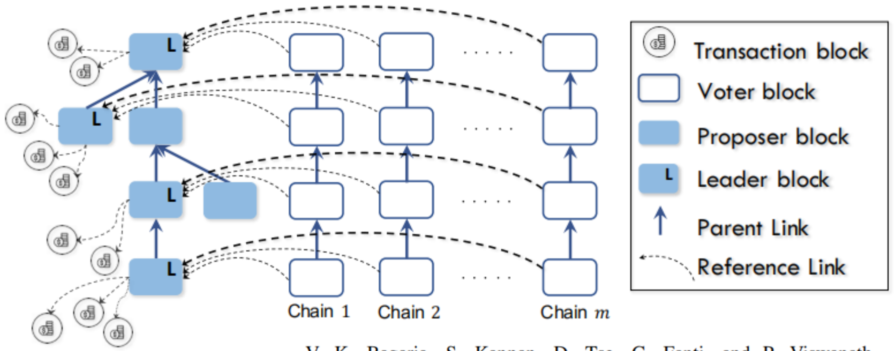
V. K. Bagaria, S. Kannan, D. Tse, G. Fanti, and P. Viswanath, "Prism: Deconstructing the blockchain to approach physical limits," in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, CCS 2019.* ACM, 2019, pp. 585–602.

- Apply our analyses to Prism
  - delay of proposer blocks is related to tx block's
    - propagation delay
    - generation rate

i.e. Throughput

Prism's security-performance trade-off

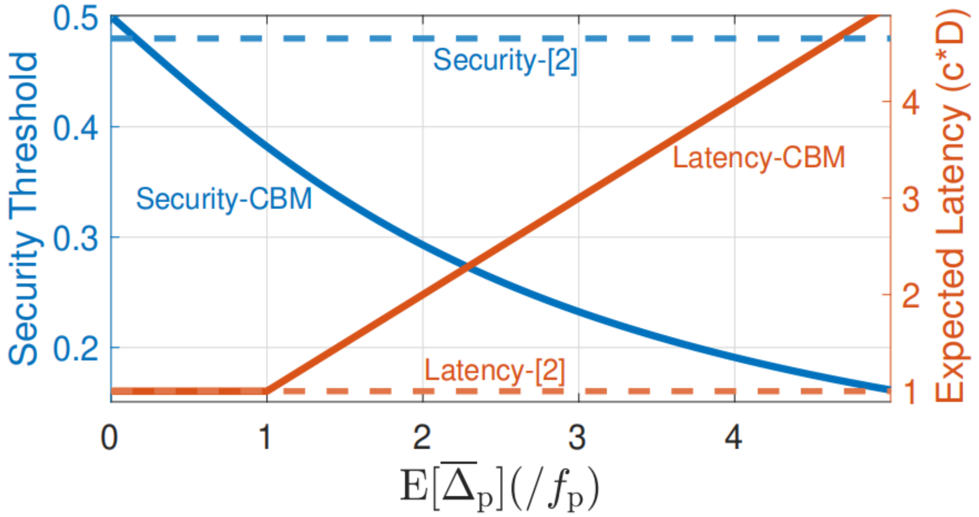
- Original paper of Prism claims that
  - changing the parameters of transaction blocks (size and rate) doesn't affect the security



V. K. Bagaria, S. Kannan, D. Tse, G. Fanti, and P. Viswanath, "Prism: Deconstructing the blockchain to approach physical limits," in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, CCS 2019.* ACM, 2019, pp. 585–602.

- Apply our analyses to Prism
  - delay of proposer blocks is related to tx block's
    - propagation delay
    - generation rate

i.e. Throughput

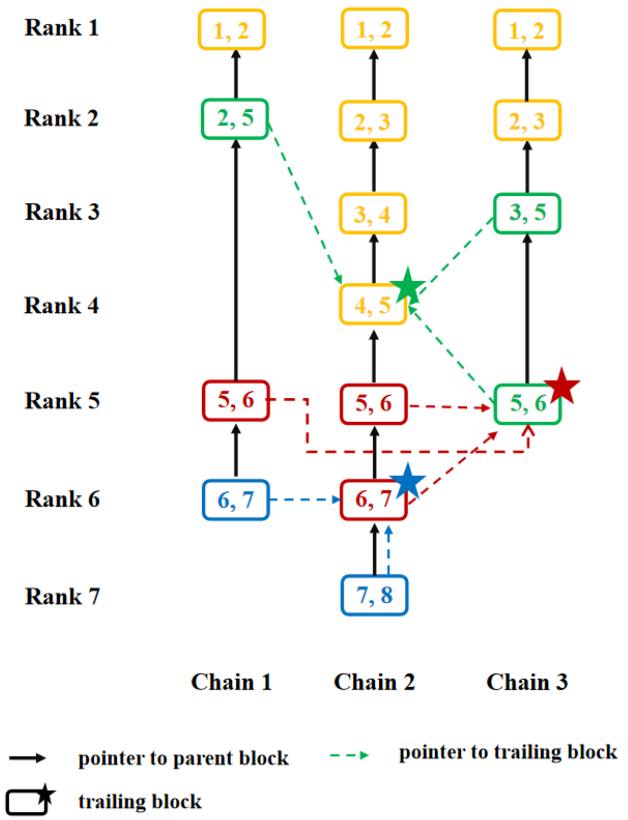


- Security-performance trade-off in Prism still exists
  - throughput ↑ security ↓ latency ↑

OHIE's security-performance trade-off

- OHIE's performance relies on the short and stable block propagation delay.
  - More than 50% of the network capacity

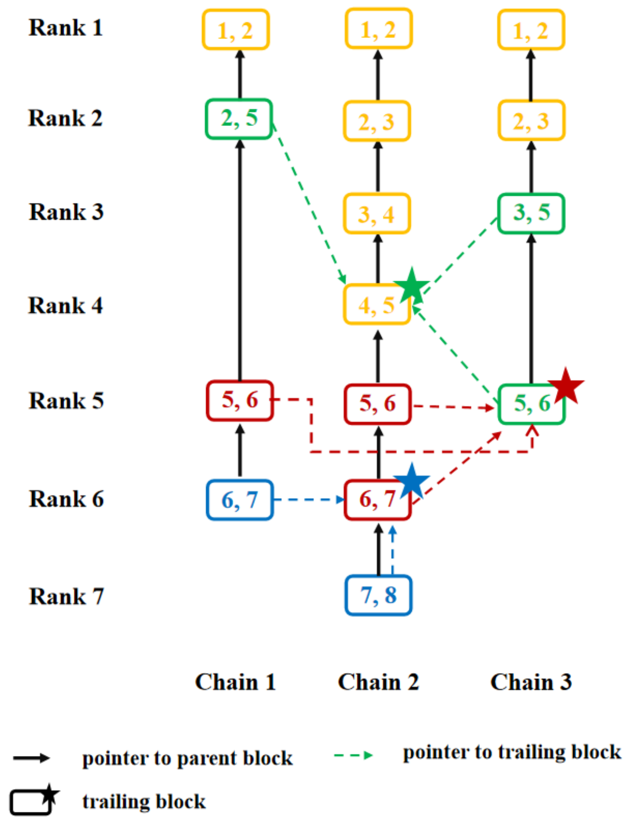
    → propagation delay increases



OHIE's security-performance trade-off

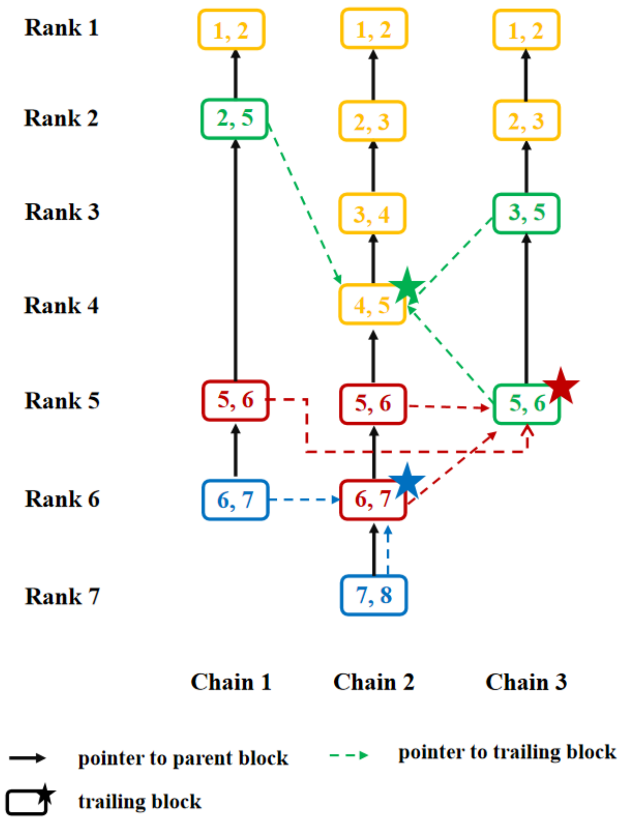
- OHIE's performance relies on the short and stable block propagation delay.
  - More than 50% of the network capacity
    └→ propagation delay increases

- Apply our analyses to OHIE
  - actual delay of all blocks increases



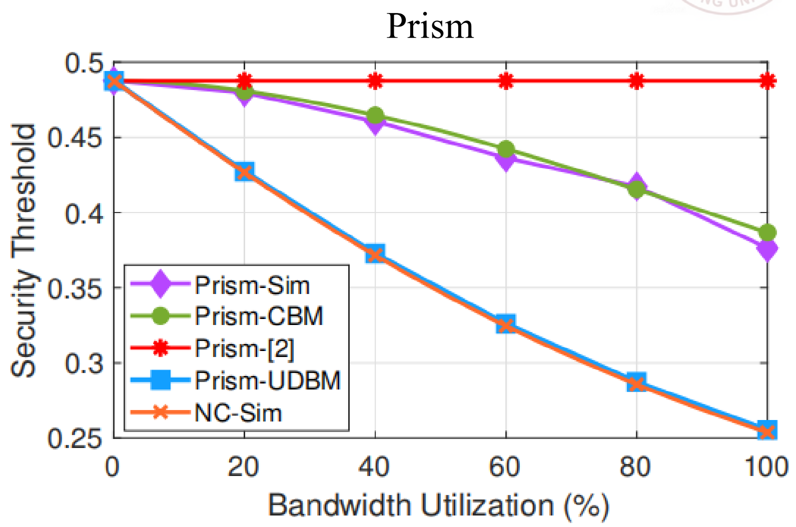
OHIE's security-performance trade-off

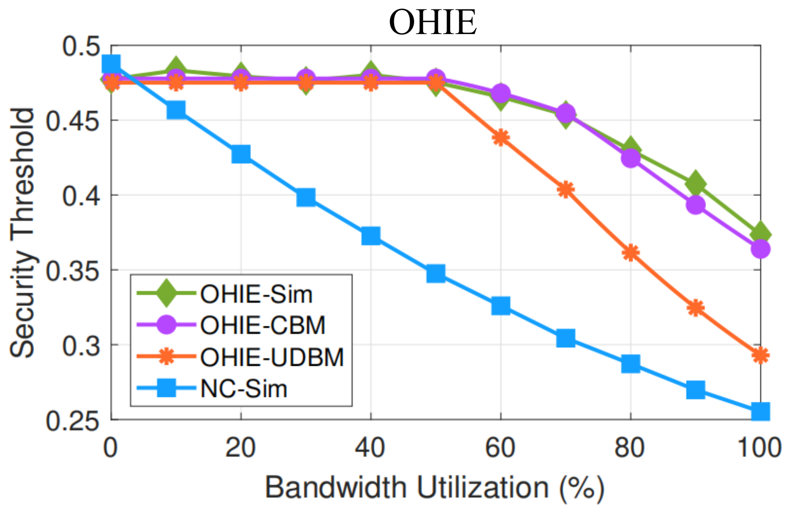
- OHIE's performance relies on the short and stable block propagation delay.
  - More than 50% of the network capacity
    - → propagation delay increases
- Apply our analyses to OHIE
  - actual delay of all blocks increases
- Security is lower when increasing throughput of OHIE by
  - increasing the block size
  - increasing the number of parallel chains (more frequent trailing blocks)



Simulation

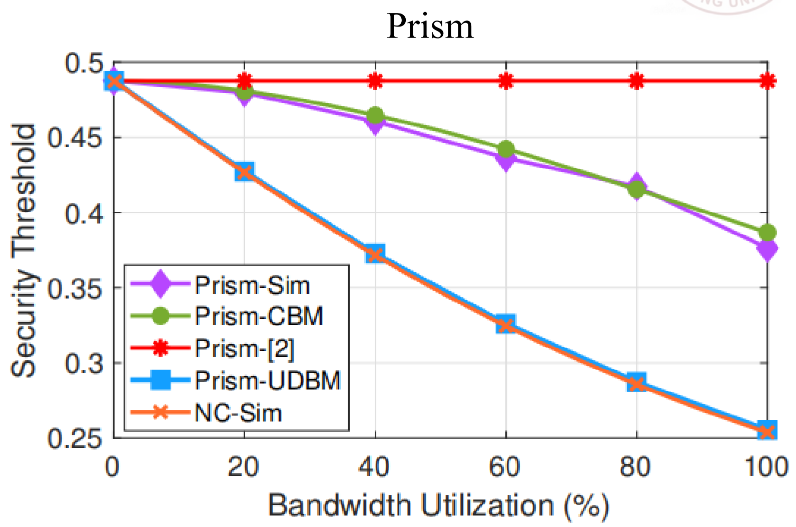
- We modify SimBlock by adding 1000 LoC to evaluate Prism's and OHIE
- Results

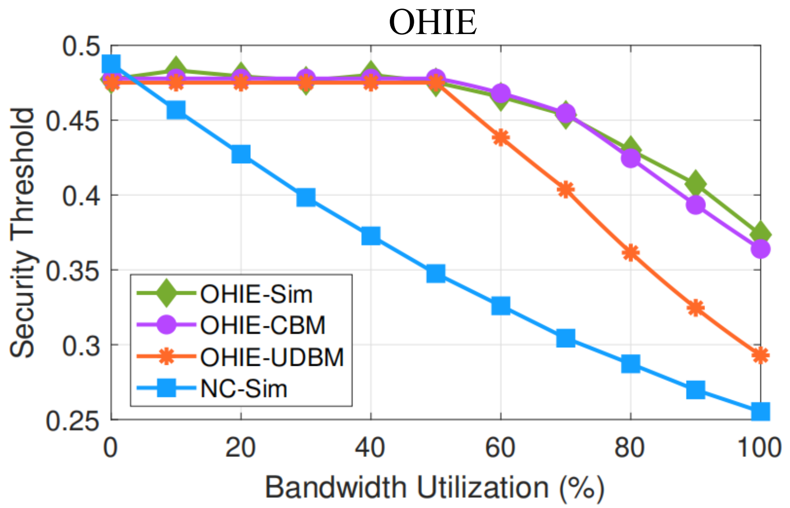




Simulation

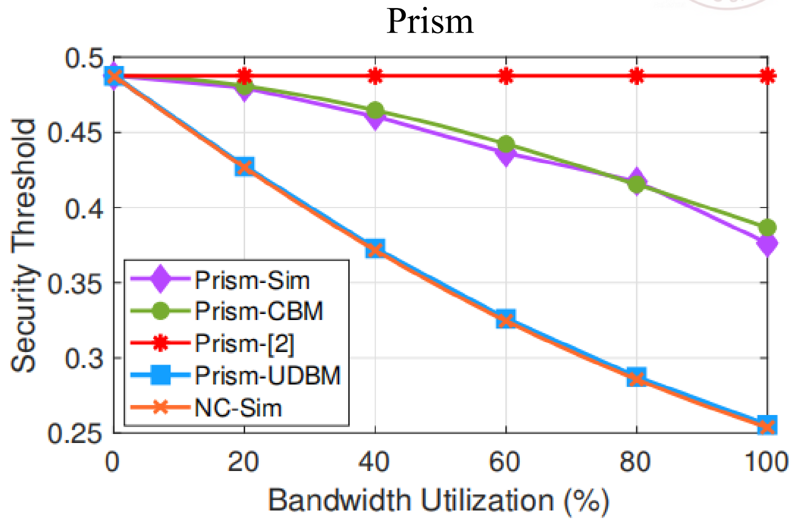
- We modify SimBlock by adding 1000 LoC to evaluate Prism's and OHIE
- Results (Prism as an example)
  - our theoretical analysis is precise
    - original paper is 0.48, simulation is 0.39

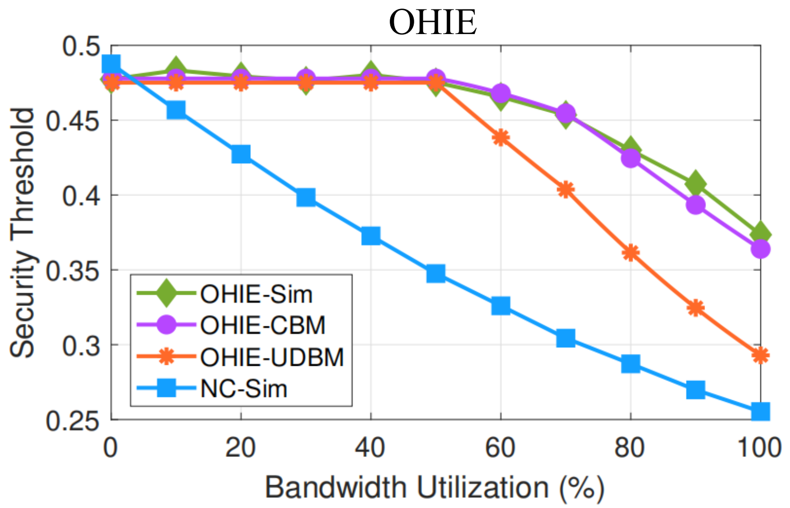




Simulation

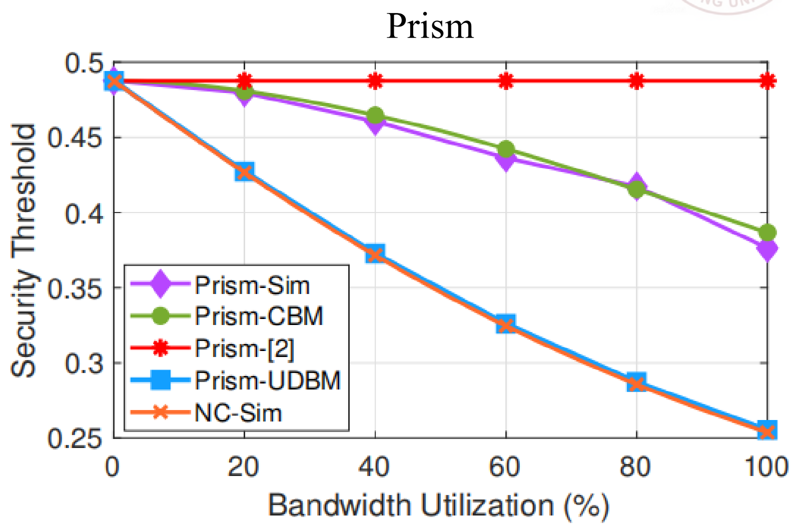
- We modify SimBlock by adding 1000 LoC to evaluate Prism's and OHIE
- Results (Prism as an example)
  - our theoretical analysis is precise
    - original paper is 0.48, simulation is 0.39
  - UDBM downgrades the security
    - UBDM is 0.27

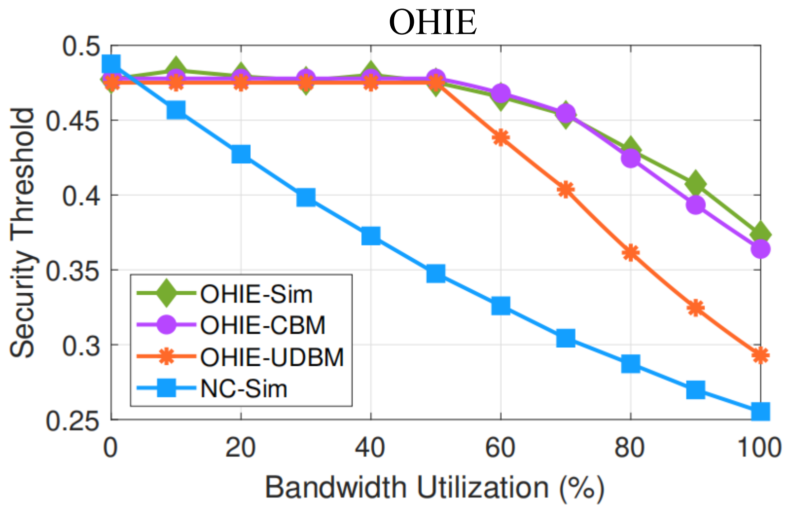




Simulation

- We modify SimBlock by adding 1000 LoC to evaluate Prism’s and OHIE
- Results (Prism as an example)
  - our theoretical analysis is precise
    - original paper is 0.48, simulation is 0.39
  - UDBM downgrades the security
    - UBDM is 0.27
- Existing DAG-based protocols still have not overcome the trade-off between security and performance





5

Conclusion & Future works

Our works:

- identified vulnerabilities in previous works
- proposed a new model called CBM
- presented a sound attack strategy
- exemplified analysis on Prism and OHIE.

5

Conclusion & Future works

- Future works:
  - Generalizability of CBM
  - Practicality and Optimality of Our Attack
  - Generalizability of the Tradeoff
  - Improving DAG-based Protocols

Thank you!

shichenw@mail.sdu.edu.cn

Shichen Wu, Puwen Wei, Ren Zhang, Bowen Jiang

NDSS 2024
28/02/2024