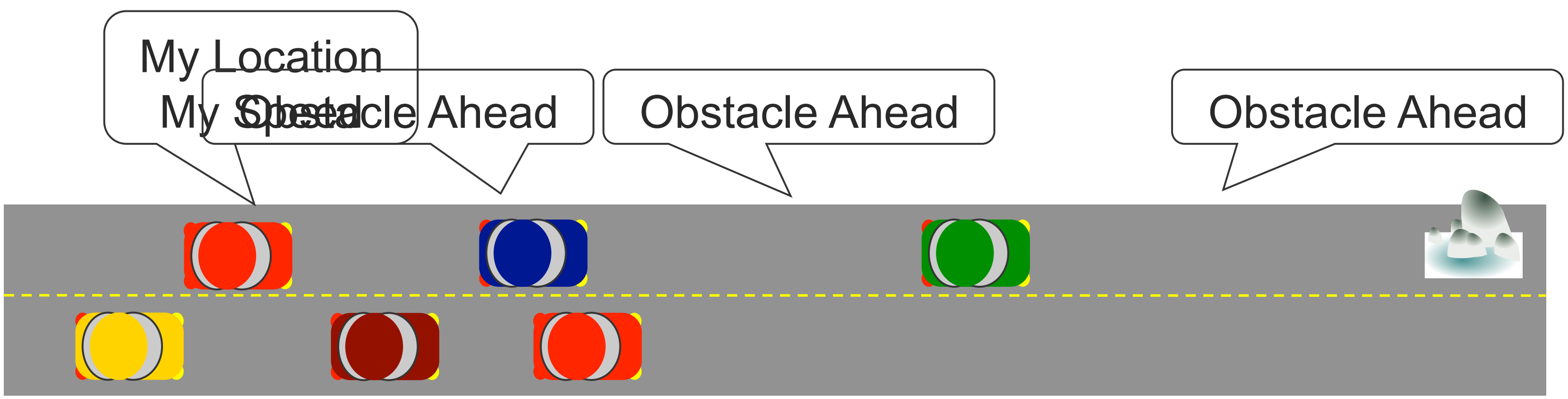
Carnegie Mellon CyLab
CONFIDENCE FOR A NETWORKED WORLD

Flooding-Resilient Broadcast Authentication for VANETs

<u>Hsu-Chun Hsiao</u>, Ahren Studer, Chen Chen, Adrian Perrig
Carnegie Mellon University
Fan Bai, Bhargav Bellur, Aravind Iyer
General Motors

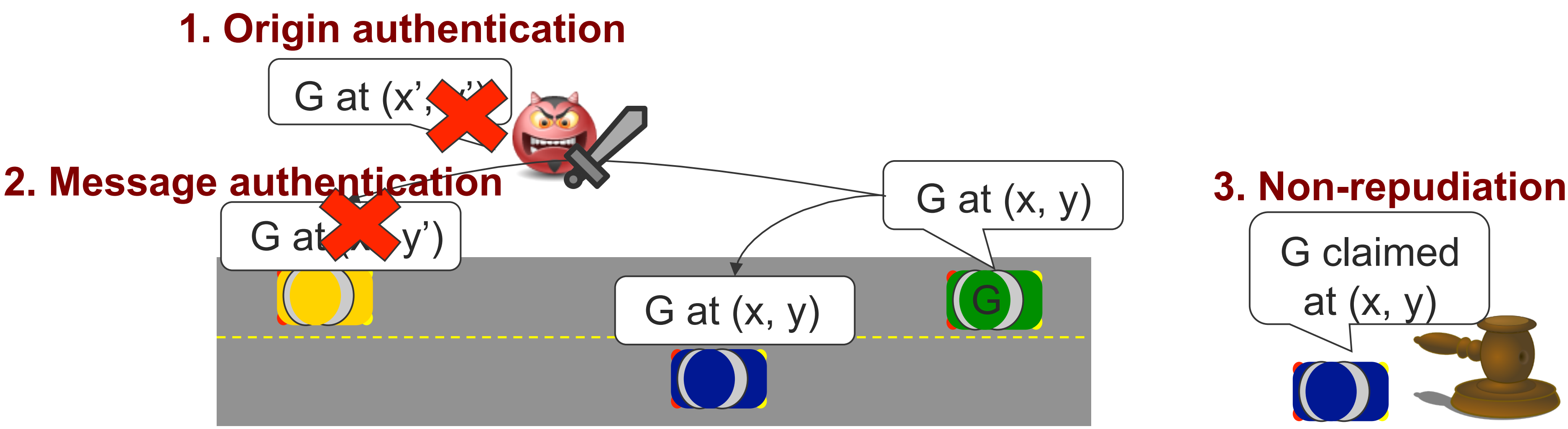
Vehicular Ad Hoc Network (VANET)

- Each vehicle possesses an On Board Unit (OBU)
  - Broadcasts info for safety & convenience



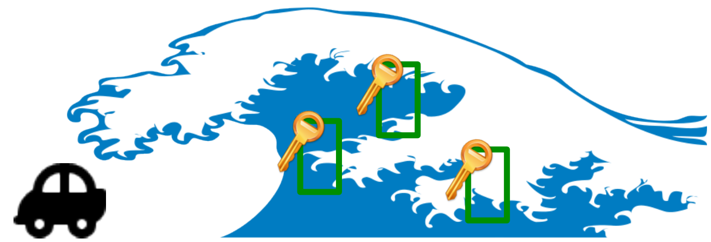
Broadcast Signatures

- Secure wireless communication



- IEEE 1609.2 VANET security standard
  - Digitally signs every message using ECDSA algorithm

Signature Flooding



- Expensive verification
  - 22 ms to verify ECDSA signature on 400MHz processor
- Many messages may arrive in a short time period
  - Every vehicle broadcasts location every 100ms
  - Verify 50 neighbors' location = 1100% processing cycle

⇒*Severely limits effectiveness of VANET applications*

Can we reduce overhead of VANET verification?

Outline

- Introduction
- Core idea: entropy-aware authentication
- Proposed flooding-resilient schemes
  - **FastAuth** secures single-hop periodic messages
  - **SelAuth** secures multi-hop messages
- Related Work
- Conclusion

Entropy-Aware Authentication

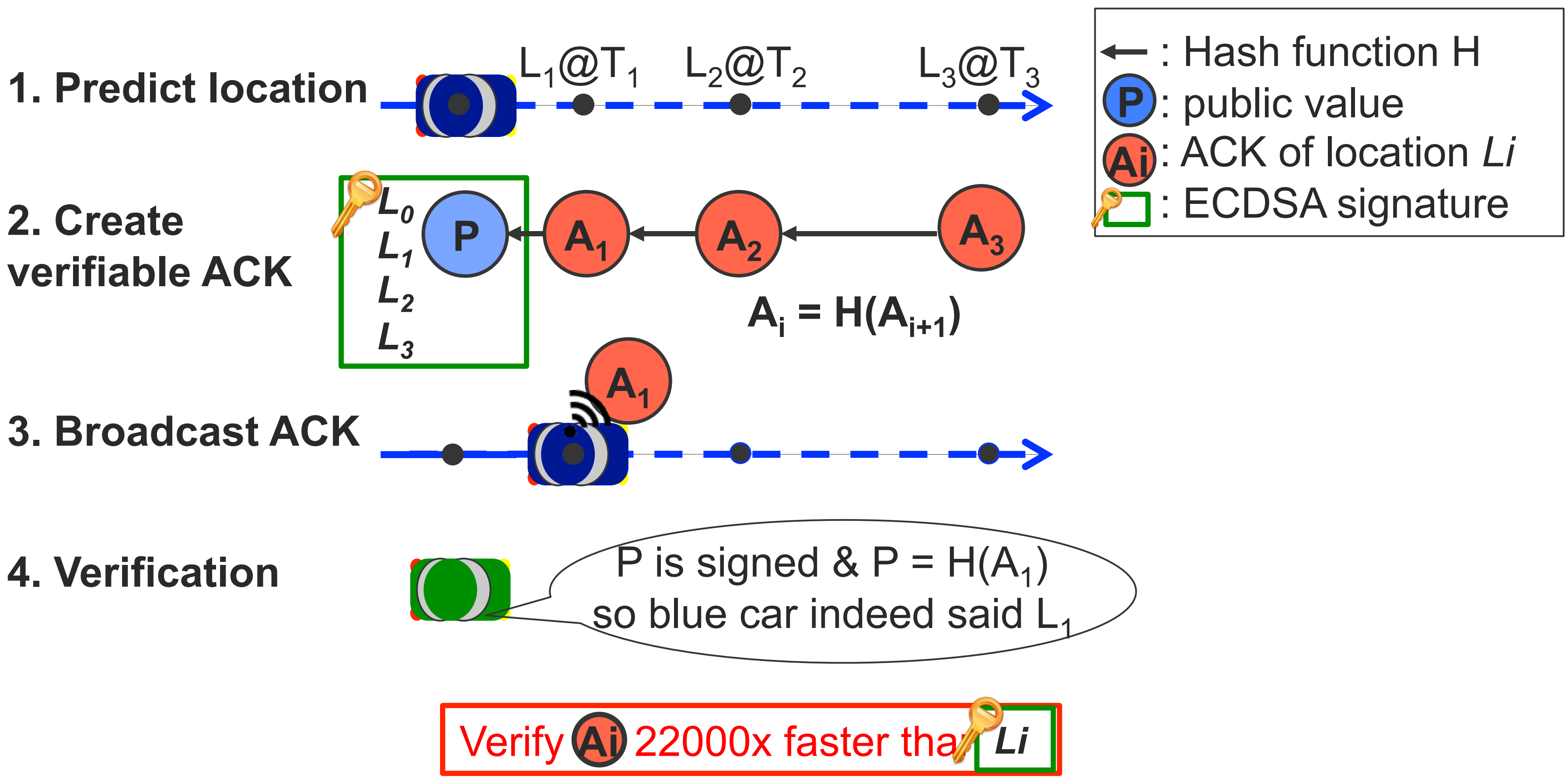
Scheme's overhead should match the *entropy of broadcast messages*

- FastAuth – exploits predictability of future messages
  - Replaces expensive ECDSA sigs with efficient hash ops
- SelAuth – selective verification before forwarding
  - Avoid checking sigs with high certainty of validity

FastAuth: First Attempt

Verifying location updates sent at 10Hz rate

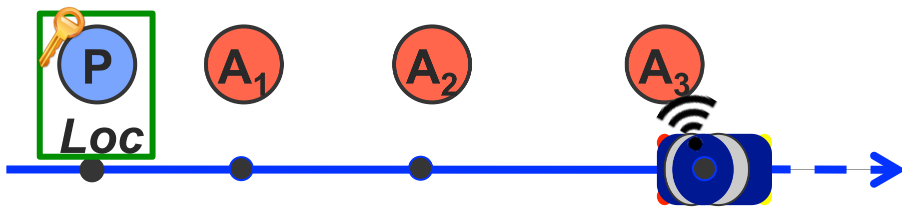
- Lightweight hash operation (1us) instead of expensive ECDSA verification (22ms)



Location Uncertainty

Ideal case: perfect prediction

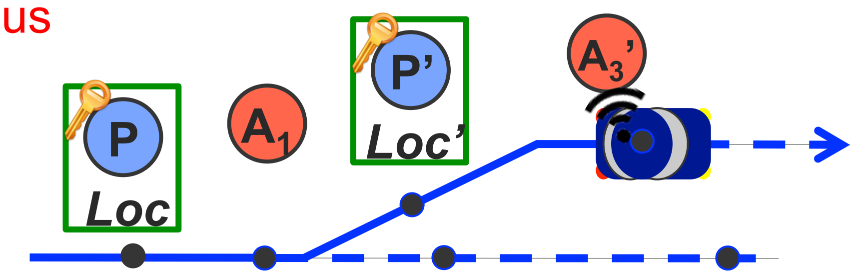
Avg overhead ➔ 1 us



Verification time
Ai : 1 us
: 22000 us

Unfortunately… incorrect prediction requires re-prediction

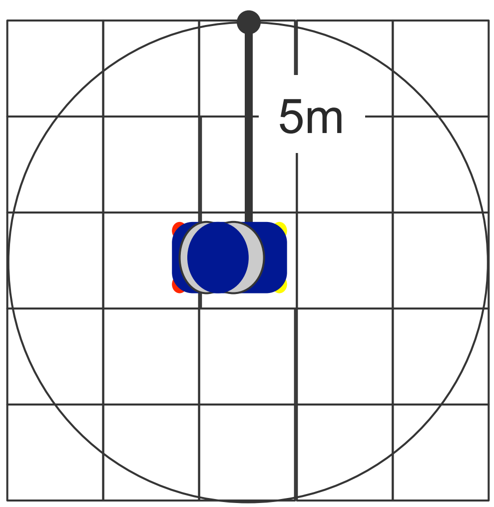
Avg overhead >> 1 us



Challenge: commit all possible movements into ACKs

1. Location Prediction

- Sender predicts it own movements
- Narrow down possible movements for efficiency
  - Sender's speed limits
    - e.g., slower than 180km or 112mile per hr ➔ cannot move > 5m per 0.1s
  - Sender's location measurement accuracy

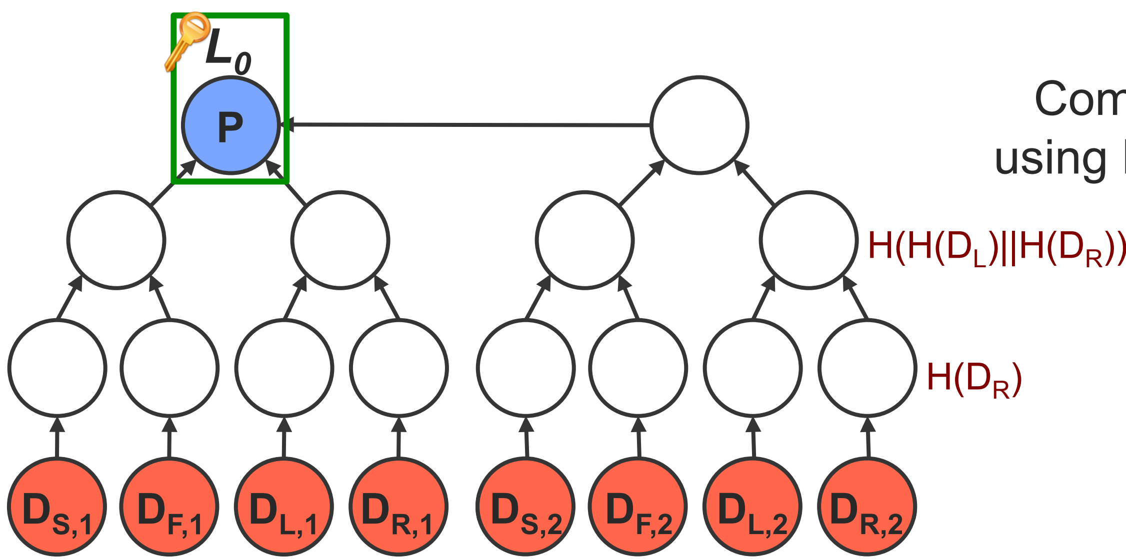


Possible Movement In 0.1s ($L_{i+1} - L_i$)
Stay (D_S)
Forward (D_F)
Forward left (D_L)
Forward right (D_R)
…
…   …

2. Verifiable ACK Construction

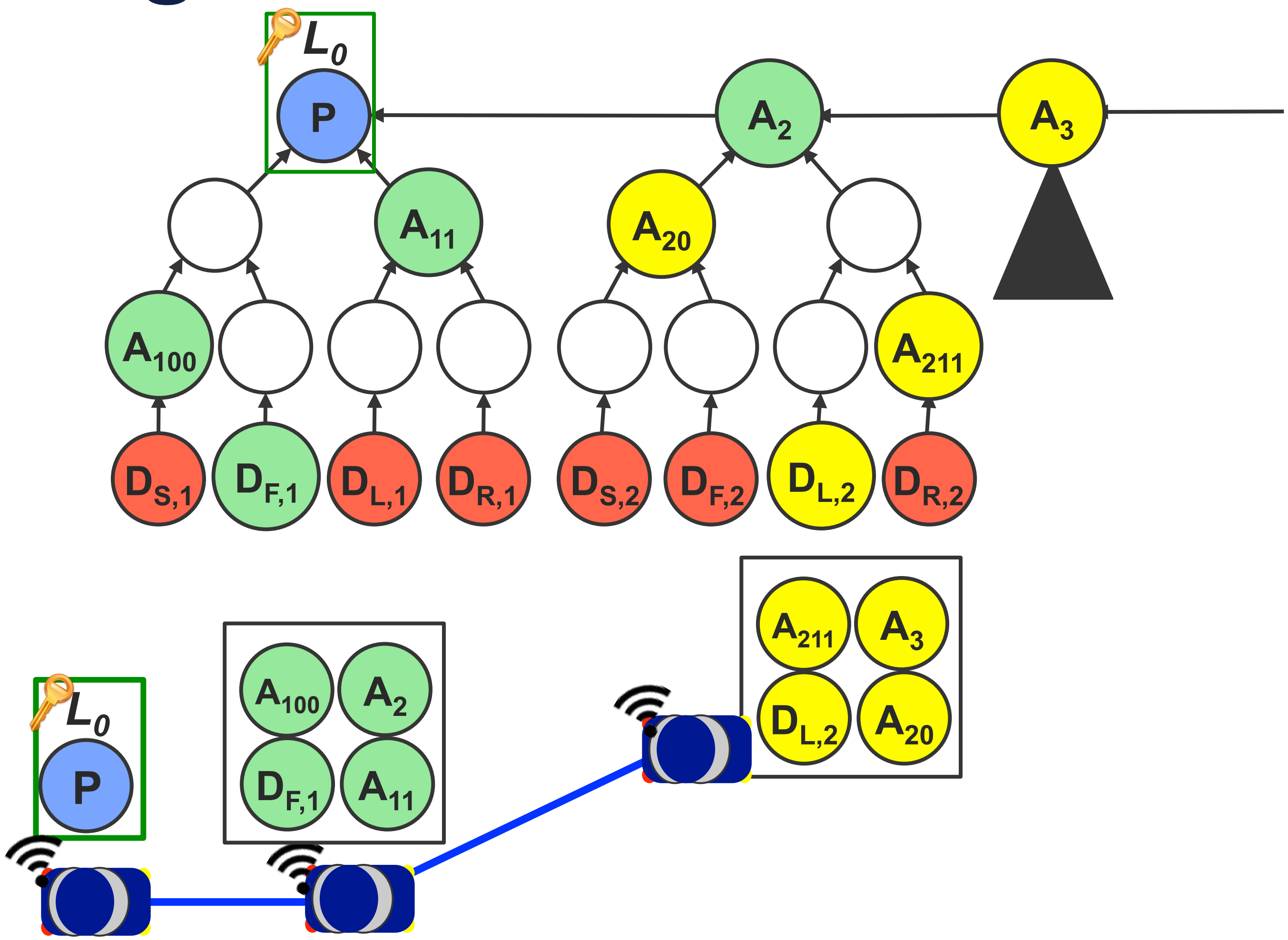
Possible Movement ($L_i - L_{i-1}$)
Stay (D_S)
Forward (D_F)
Forward left (D_L)
Forward right (D_R)

- ↑ : Hash function H
- Blue circle: public value
- Red circle: ACK of location *Li*
- Green box: ECDSA signature



Commit movements using Merkle Hash Tree

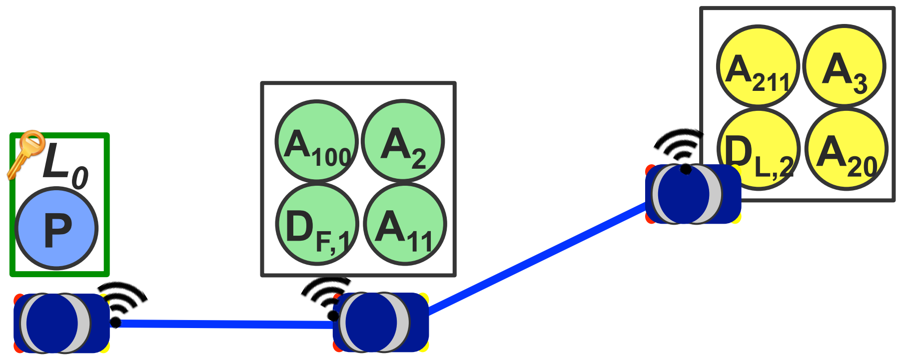
3. Signed Location Broadcast



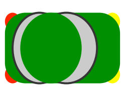
Movement committed to ACK tree => No re-prediction needed!

4. Verification

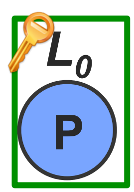
Sender



Receiver

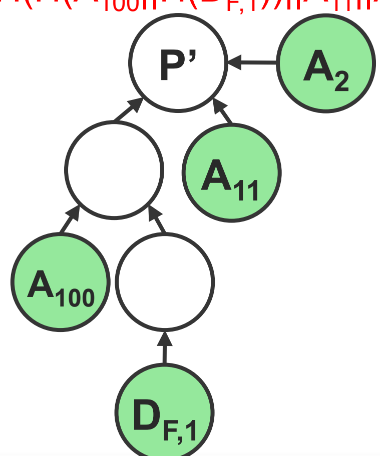


Verify ECDSA sig

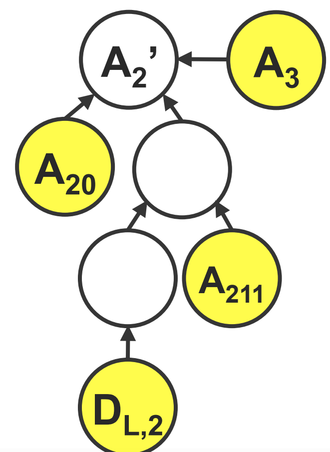


Compute P'
Verify if P = P'
$L_1 = L_0 + D_F$

P' = $H(H(A_{100}||H(D_{F,1}))||A_{11}||A_2)$



Compute A_2'
Verify if $A_2' = A_2$
$L_2 = L_1 + D_L$



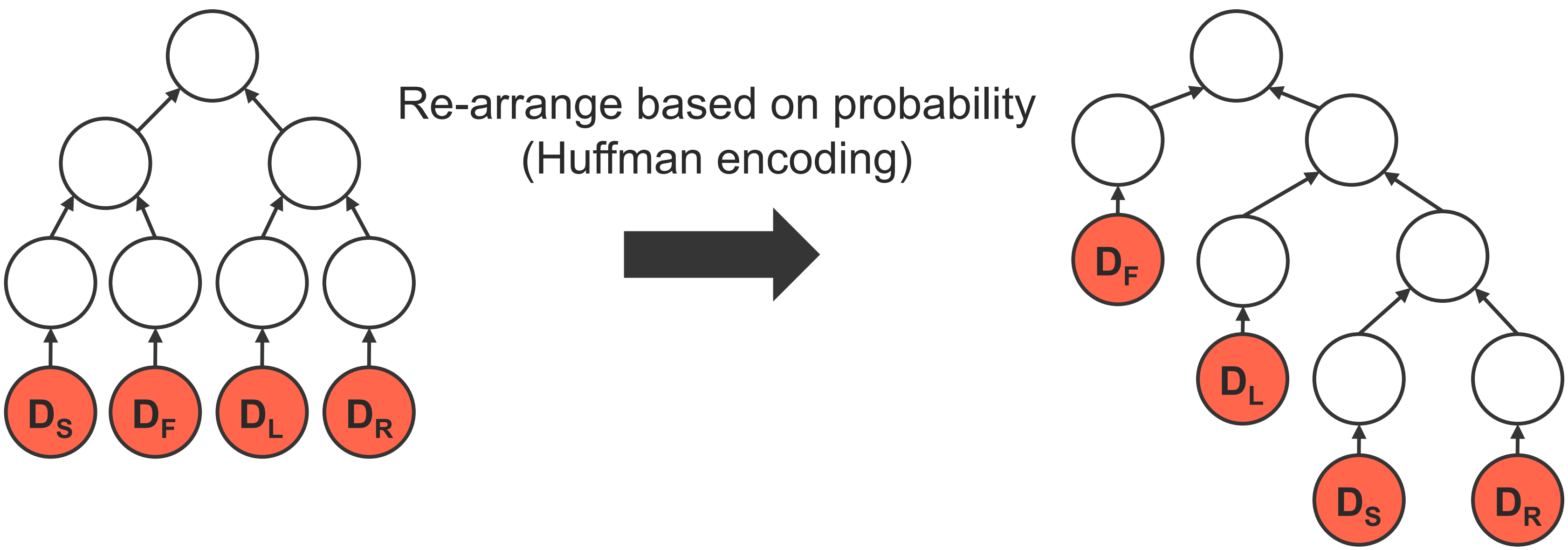
Further Improvement

- We have reduced verification overhead
  - Expensive sig verification => lightweight hash ops
- Can we also reduce comm. overhead?
  - Yes. Not fully leveraged location predictability yet

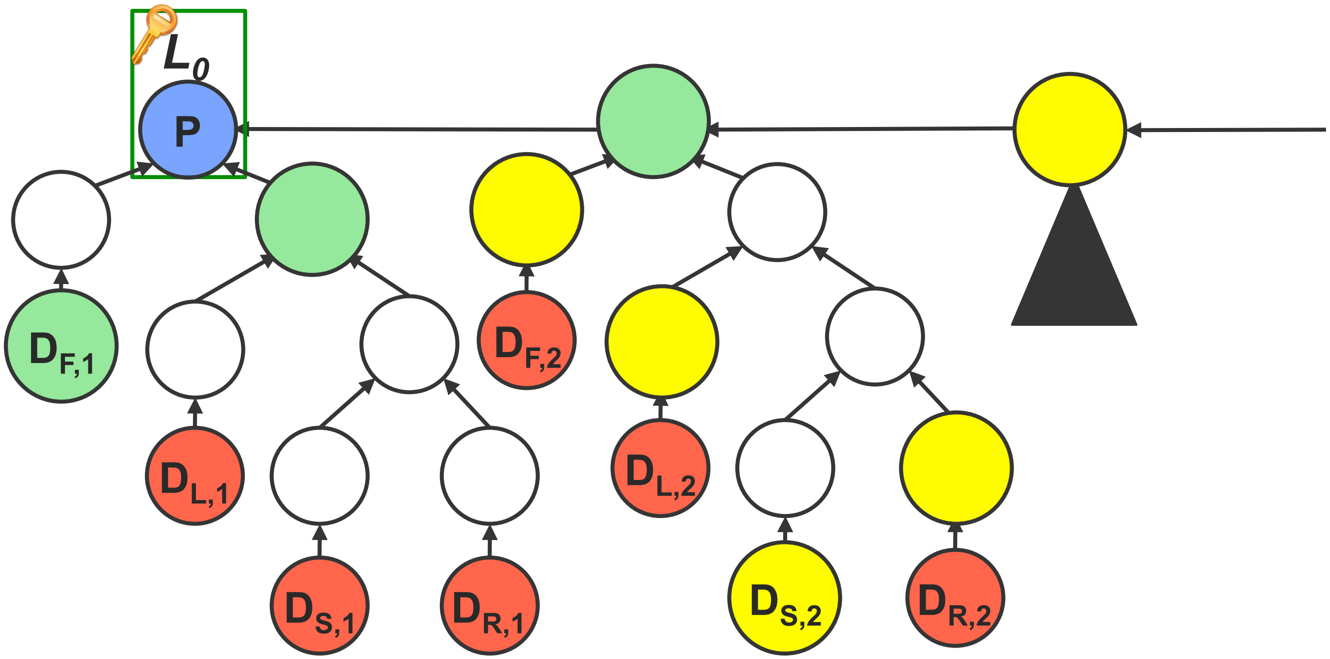
Possible Movement ($L_i - L_{i-1}$)	Probability
Stay (Ds)	?
Forward (Df)	?
Forward left (Dl)	?
Forward right (Dr)	?

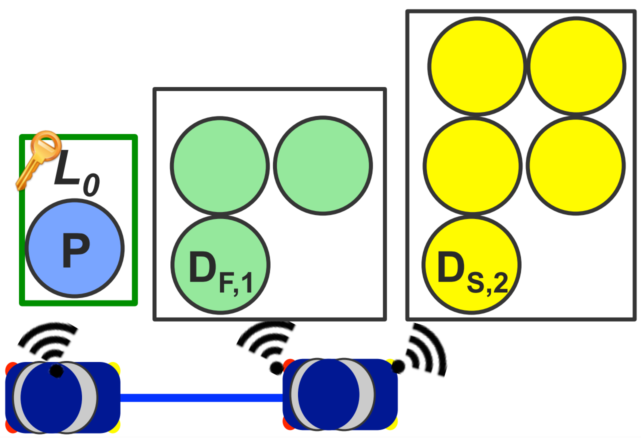
Huffman Tree + Hash Tree

Possible Movement ($L_i - L_{i-1}$)	**Probability**
Stay (D_S)	P_S
Forward (D_F)	P_F
Forward left (D_L)	P_L
Forward right (D_R)	P_R



Reduced Communication





Inaccurate probability model
⇒Larger sigs but still no re-prediction needed!

Discussion

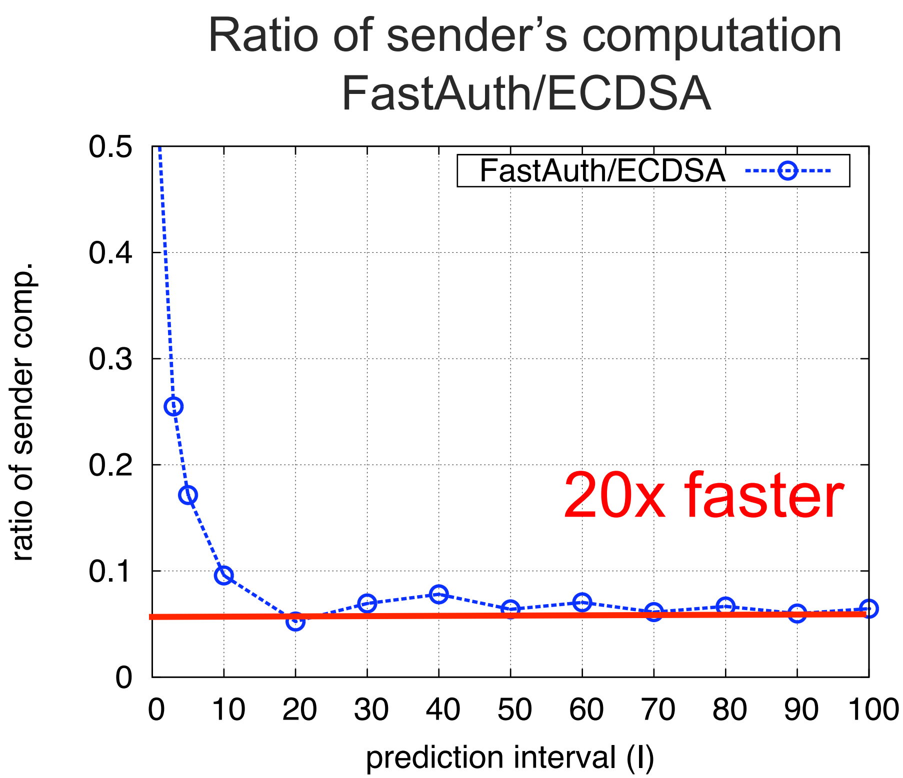
- Tradeoffs
  - Pros: instant verification, low comp. & low comm.
  - Cons: low update frequency
- Low update frequency due to verification dependency
  - Missing msg prevents verification of subsequent msgs
- To increase update frequency
  - Error correction codes to mitigate packet loss
  - Occasionally sign messages using ECDSA signatures

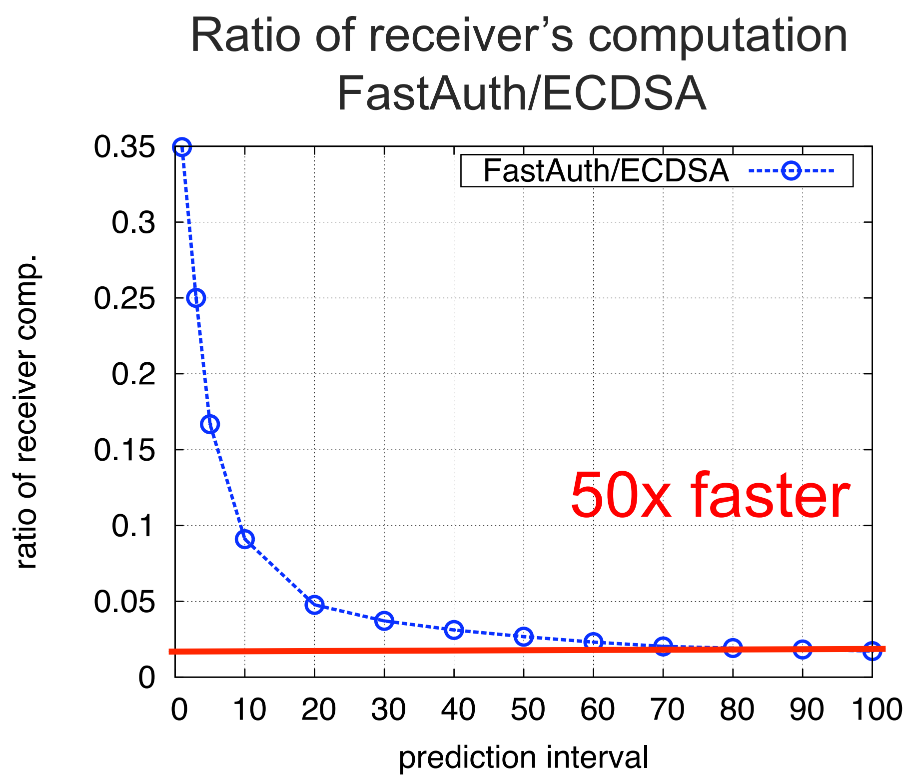
FastAuth: Evaluation Settings

Does FastAuth mitigate Signature Flooding?

- Evaluate receiver's & sender's computational overhead
- Data collection
  - 4 traces, each by driving along a 2-mile path for 2 hours
- Additional evaluation metrics
  - Communication, update frequency
- Impacting factors
  1. Is FastAuth sensitive to *prediction accuracy?*
  2. How does *packet loss* affect FastAuth?

FastAuth: Computation





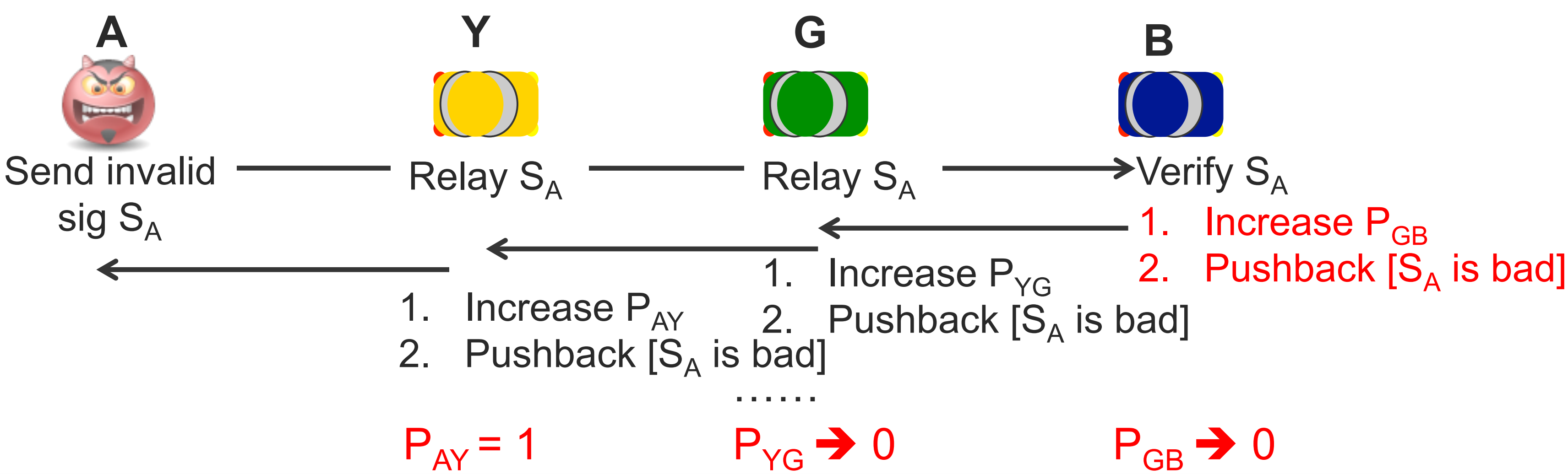
CyLab www.cylab.cmu.edu

Outline

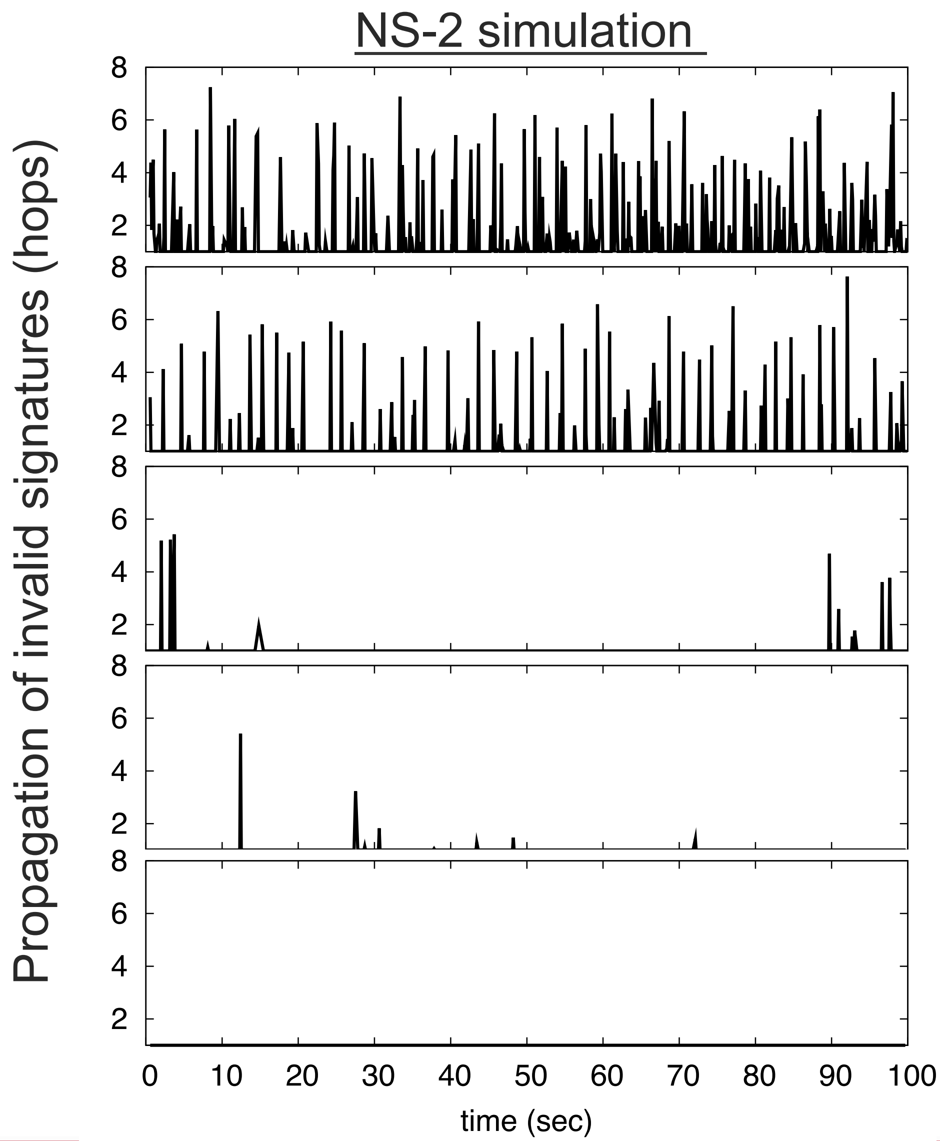
- Introduction
- Core idea: entropy-aware authentication
- Proposed flooding-resilient schemes
  - **FastAuth** secures single-hop periodic messages
  - **SelAuth** secures multi-hop messages
- Related Work
- Conclusion

SelAuth Overview

- SelAuth is about
  - Finds balance between *Verify-on-Demand* & *Verify-All*
  - Promptly isolates malicious parties
    - Invalid sigs cannot spread out consuming comm. bandwidth
  - Quickly adjusts P_{xy} s.t.
    - Pxy = Pr[y verifies signatures forwarded by x]
    - Pxy → 0 for benign x & Pxy → 1 for malicious x



Fast Isolation of Mobile Attacker



One verification prob. for all neighbors

One verification prob. for all neighbors
+ Pushback warning

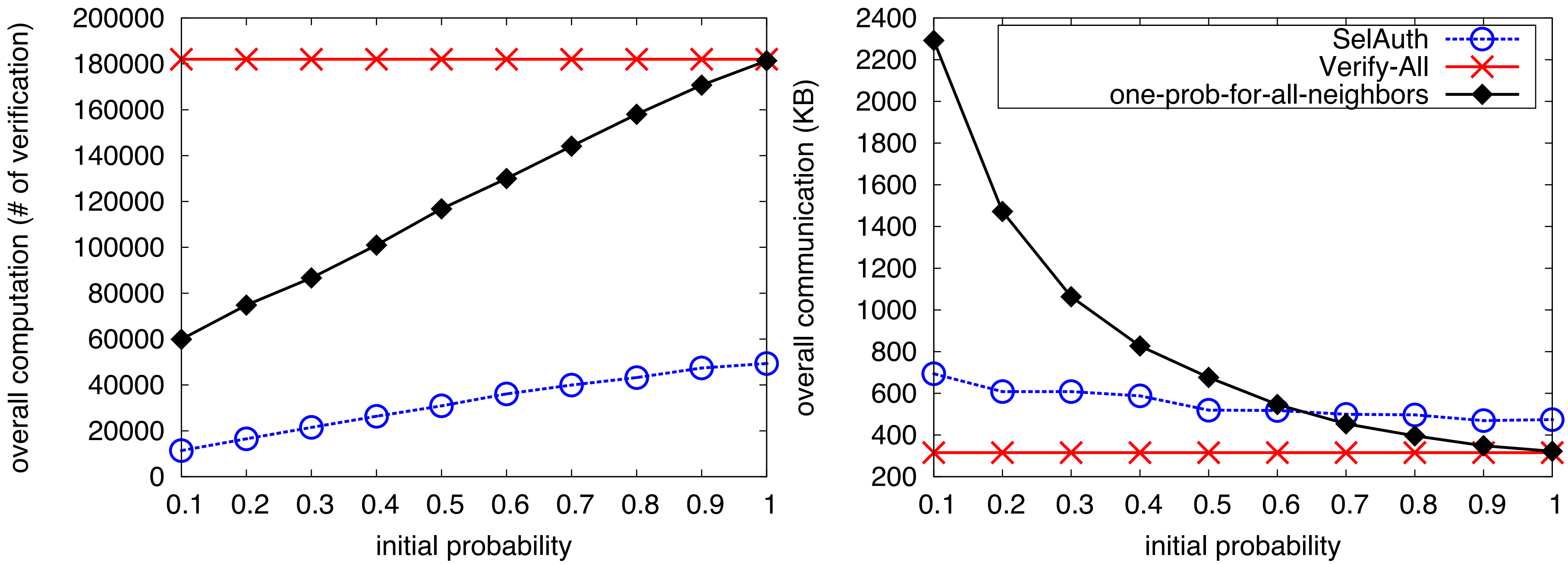
Per-neighbor verification prob.

SelAuth
Per-neighbor verification prob.
+ Pushback warning

Verify every signature with p = 1

SelAuth: Low Overhead

NS-2 simulation: 336 vehicles in 1kmx1km downtown Manhattan



Related Work

- Efficient broadcast authentication
  - Avoid expensive asymmetric cryptographic ops
    - Use symmetric crypto instead
      - One-time signatures: [Lamport, Merkle, Gennaro & Rohatgi]
      - One-way hash chains: [Perrig et al., Hu et al., Studer et al.]
    - Less crypto work when threat level is low
      - [Gunter et al., Khanna et al., Wang et al., Ristanovic et al., Li et al.]

Conclusion

- Flooding-resilient broadcast signatures
  - Required for timely verification of safety messages
  - Unachievable in current standard even in benign settings
- Entropy-aware authentication to mitigate flooding
  - FastAuth: instant verification for one-hop messages
    - Leverages message predictability
    - 50x faster computation compared to current standard
  - SelAuth: selective authentication for multi-hop messages
    - Enables fast isolation of malicious senders
    - 15%-30% computational overhead