Security 2024

Max Attestation Matters
: Making Honest Parties Lose Their Incentives in Ethereum PoS

Mingfei Zhang
Shandong University
mingfei.zh@outlook.com

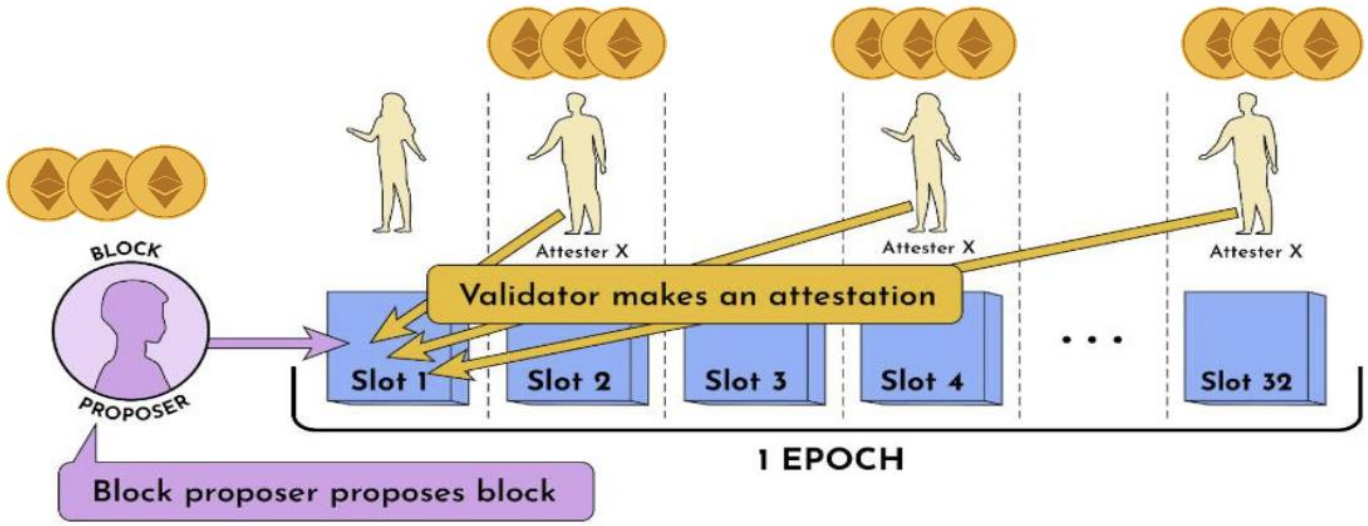
Rujia Li*
Tsinghua University
rujia@tsinghua.edu.cn

Sisi Duan*†
Tsinghua University
duansisi@tsinghua.edu.cn

2024. 10. 16
서울대 분산시스템연구실
석사과정 문보설

Motivation

- Ethereum incentive system
  - Attestation Incentives
    - Rewards
    - **Penalties**
  - Block Rewards
  - Sync Committee Incentives



Background

- Attestation
  - Source 체크포인트와 target 체크포인트의 연결성(link)에 대한 투표
- Last Justified Checkpoint
  - 마지막으로 justify된 체크포인트 블록
  - 마지막으로 2N/3 attestation을 받은 블록
  - Attestation의 source
- …

Attack overview

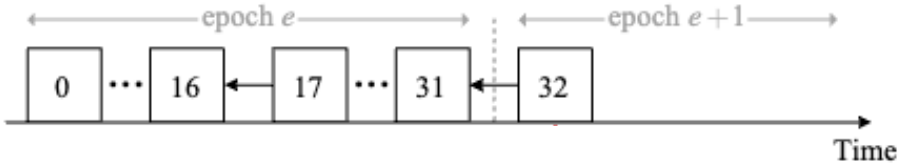
- Idea: LJ might change in the middle of an epoch

- **Case 1) LJ is updated at the first block**
  - Normal case
  - Assume LJ == cp at epoch e

Attack overview

- Idea: LJ might change in the middle of an epoch

- **Case 1) LJ is updated at the first block**
  - Normal case
  - Assume LJ == cp at epoch e

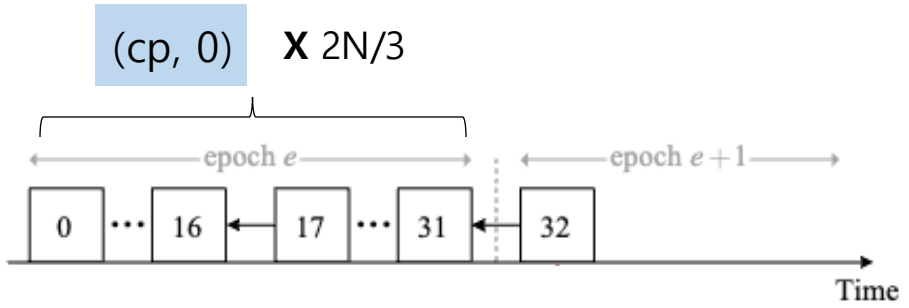


(a) v_1 receives block 32, the first block in epoch $e+1$. v_1 updates its LJ to block 0.

Attack overview

- Idea: LJ might change in the middle of an epoch

- **Case 1) LJ is updated at the first block**
  - Normal case
  - Assume LJ == cp at epoch e

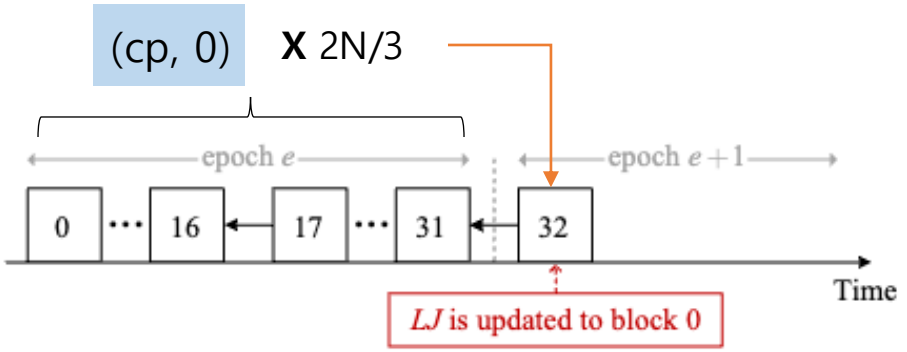


(a) v_1 receives block 32, the first block in epoch $e+1$. v_1 updates its *LJ* to block 0.

Attack overview

- Idea: LJ might change in the middle of an epoch

- **Case 1) LJ is updated at the first block**
  - Normal case
  - Assume LJ == cp at epoch e

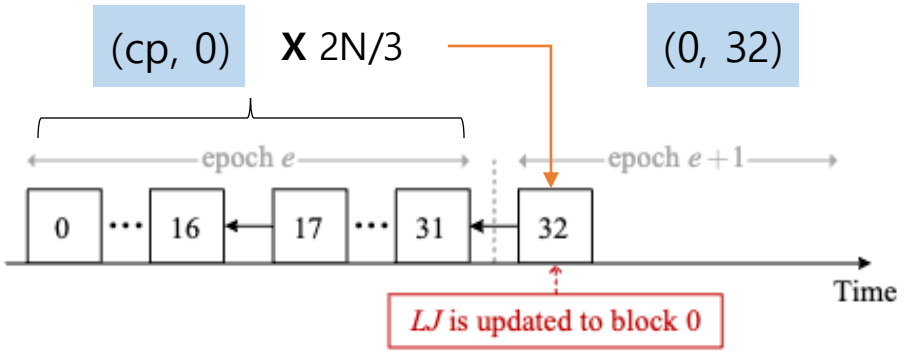


(a) v_1 receives block 32, the first block in epoch $e+1$. v_1 updates its *LJ* to block 0.

Attack overview

- Idea: LJ might change in the middle of an epoch

- **Case 1) LJ is updated at the first block**
  - Normal case
  - Assume LJ == cp at epoch e



(a) v_1 receives block 32, the first block in epoch $e+1$. v_1 updates its LJ to block 0.

Attack overview

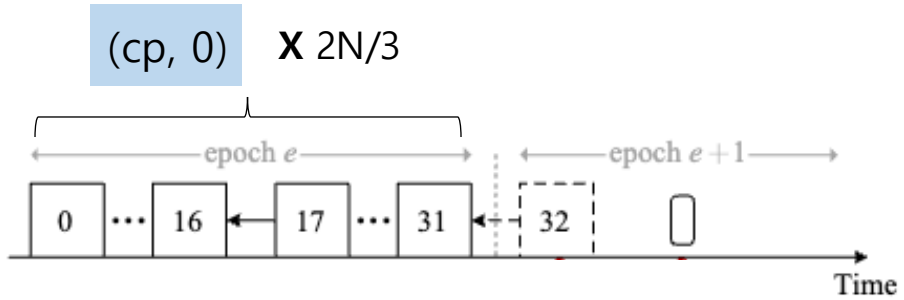
- Idea: LJ might change in the middle of an epoch

- **Case 2) LJ is updated before preparing an attestation**
  - Assume LJ == cp at epoch e
  - No blocks has been received in epoch e+1

Attack overview

- Idea: LJ might change in the middle of an epoch

- **Case 2) LJ is updated before preparing an attestation**
  - Assume LJ == cp at epoch e
  - No blocks has been received in epoch e+1

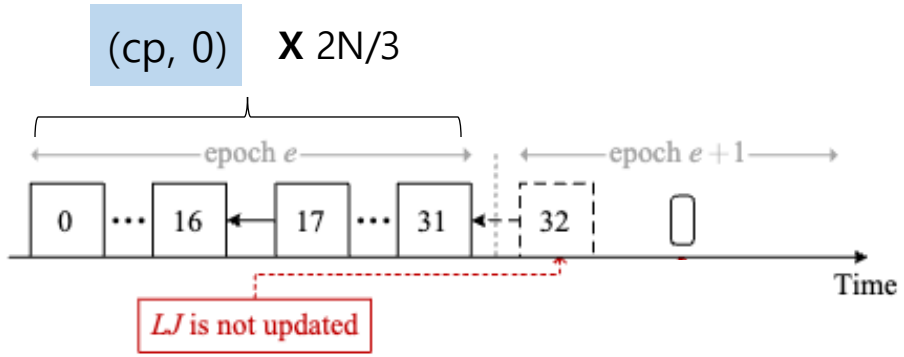


(b) v_2 has not received any blocks in epoch $e+1$ yet and does not update its *LJ* until it prepares an attestation. Before v_2 prepares its attestation, it updates its *source* as block 0 and *target* as block 31.

Attack overview

- Idea: LJ might change in the middle of an epoch

- **Case 2) LJ is updated before preparing an attestation**
  - Assume LJ == cp at epoch e
  - No blocks has been received in epoch e+1

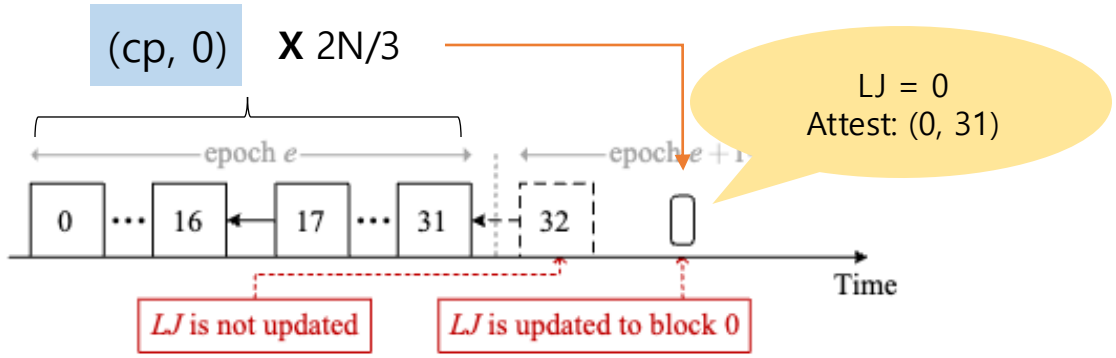


(b) v_2 has not received any blocks in epoch $e+1$ yet and does not update its *LJ* until it prepares an attestation. Before v_2 prepares its attestation, it updates its *source* as block 0 and *target* as block 31.

Attack overview

- Idea: LJ might change in the middle of an epoch

- **Case 2) LJ is updated before preparing an attestation**
  - Assume LJ == cp at epoch e
  - No blocks has been received in epoch e+1



(b) v_2 has not received any blocks in epoch $e+1$ yet and does not update its *LJ* until it prepares an attestation. Before v_2 prepares its attestation, it updates its *source* as block 0 and *target* as block 31.

Attack overview

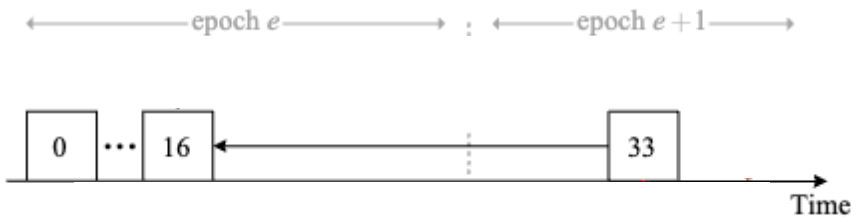
- Idea: LJ might change in the middle of an epoch

- **Case 3) LJ is updated in the middle of an epoch**
  - Assume LJ == cp at epoch e
  - Blocks from previous epoch are withheld / delayed

Attack overview

- Idea: LJ might change in the middle of an epoch

- **Case 3) LJ is updated in the middle of an epoch**
  - Assume LJ == cp at epoch e
  - Blocks from previous epoch are withheld / delayed

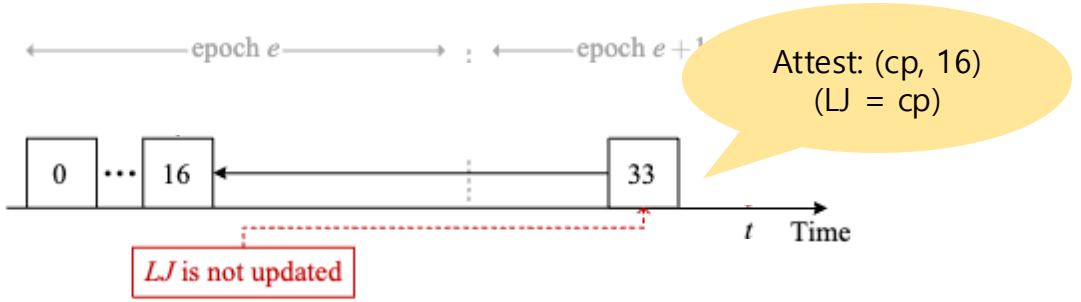


(c) v_3 receives block 33 and the chain led by block 33 does not include $2N/3$ attestations. v_3 does not update its LJ. When v_3 later receives blocks 17-31, the chain led by block 31 includes $2N/3$ attestations with *target* as block 0. v_3 updates LJ to block 0.

Attack overview

- Idea: LJ might change in the middle of an epoch

- **Case 3) LJ is updated in the middle of an epoch**
  - Assume LJ == cp at epoch e
  - Blocks from previous epoch are withheld / delayed

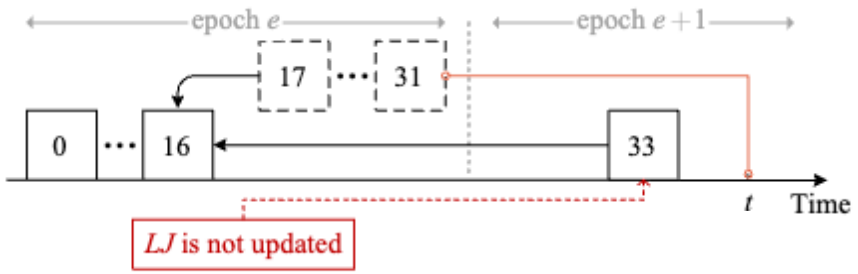


(c) v_3 receives block 33 and the chain led by block 33 does not include $2N/3$ attestations. v_3 does not update its LJ. When v_3 later receives blocks 17-31, the chain led by block 31 includes $2N/3$ attestations with *target* as block 0. v_3 updates LJ to block 0.

Attack overview

- Idea: LJ might change in the middle of an epoch

- **Case 3) LJ is updated in the middle of an epoch**
  - Assume LJ == cp at epoch e
  - Blocks from previous epoch are withheld / delayed

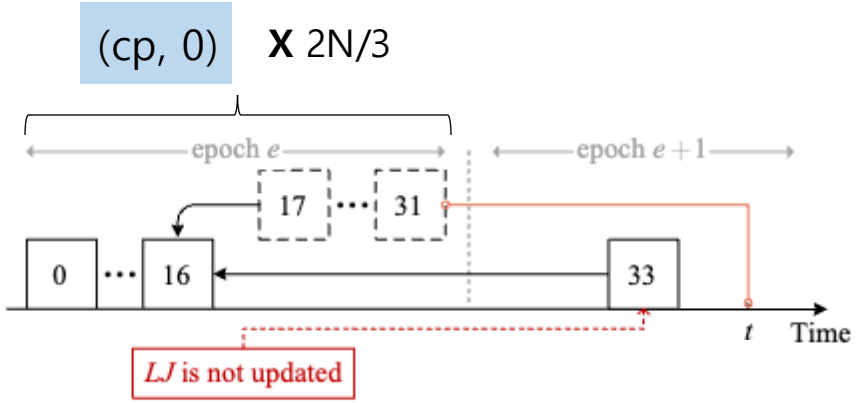


(c) v_3 receives block 33 and the chain led by block 33 does not include $2N/3$ attestations. v_3 does not update its LJ. When v_3 later receives blocks 17-31, the chain led by block 31 includes $2N/3$ attestations with *target* as block 0. v_3 updates LJ to block 0.

Attack overview

- Idea: LJ might change in the middle of an epoch

- **Case 3) LJ is updated in the middle of an epoch**
  - Assume LJ == cp at epoch e
  - Blocks from previous epoch are withheld / delayed

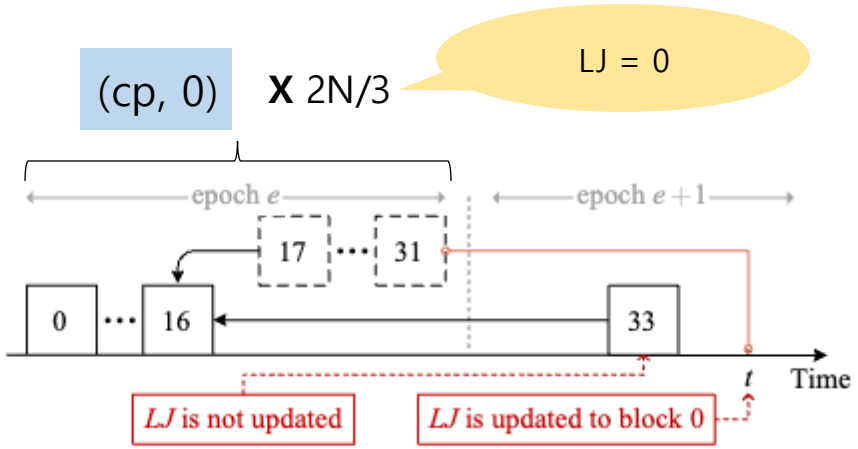


(c) v_3 receives block 33 and the chain led by block 33 does not include $2N/3$ attestations. v_3 does not update its LJ. When v_3 later receives blocks 17-31, the chain led by block 31 includes $2N/3$ attestations with *target* as block 0. v_3 updates LJ to block 0.

Attack overview

- Idea: LJ might change in the middle of an epoch

- **Case 3) LJ is updated in the middle of an epoch**
  - Assume LJ == cp at epoch e
  - Blocks from previous epoch are withheld / delayed

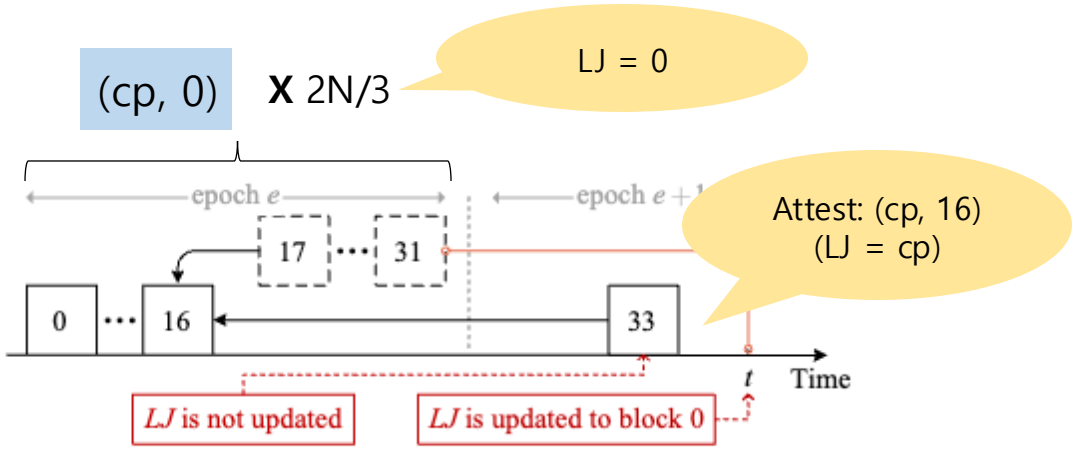


(c) v_3 receives block 33 and the chain led by block 33 does not include $2N/3$ attestations. v_3 does not update its LJ. When v_3 later receives blocks 17-31, the chain led by block 31 includes $2N/3$ attestations with *target* as block 0. v_3 updates LJ to block 0.

Attack overview

- Idea: LJ might change in the middle of an epoch

- **Case 3) LJ is updated in the middle of an epoch**
  - Assume LJ == cp at epoch e
  - Blocks from previous epoch are withheld / delayed

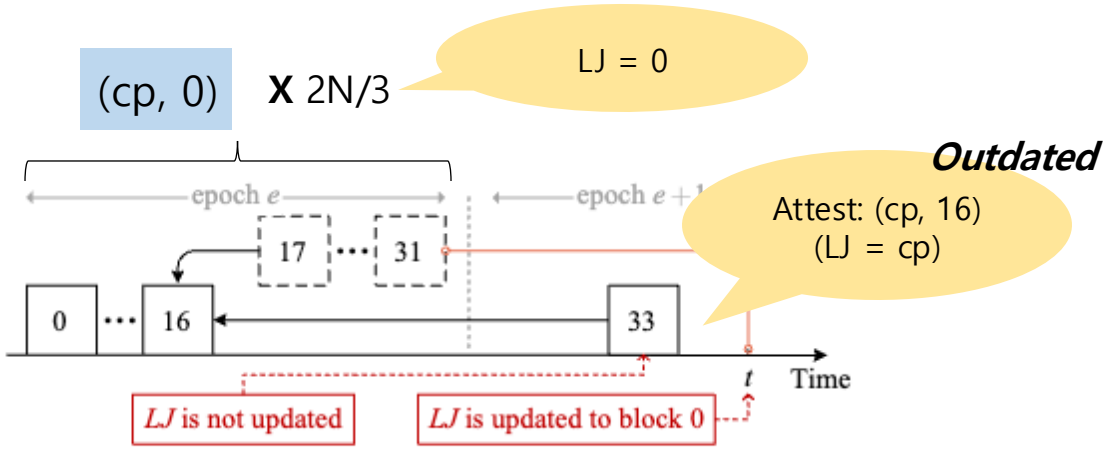


(c) v_3 receives block 33 and the chain led by block 33 does not include $2N/3$ attestations. v_3 does not update its LJ. When v_3 later receives blocks 17-31, the chain led by block 31 includes $2N/3$ attestations with *target* as block 0. v_3 updates LJ to block 0.

Attack overview

- Idea: LJ might change in the middle of an epoch

- **Case 3) LJ is updated in the middle of an epoch**
  - Assume LJ == cp at epoch e
  - Blocks from previous epoch are withheld / delayed

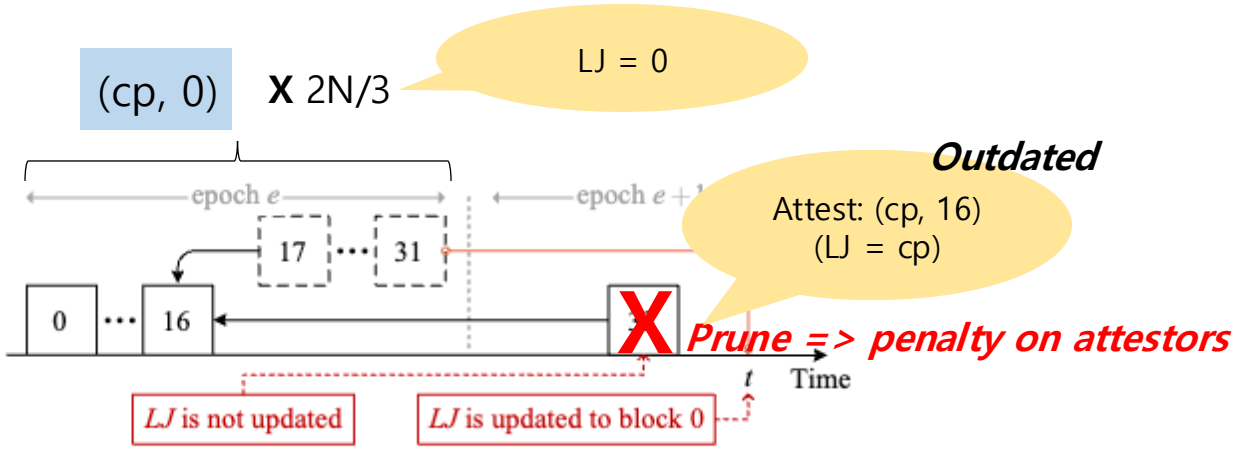


(c) v_3 receives block 33 and the chain led by block 33 does not include $2N/3$ attestations. v_3 does not update its LJ. When v_3 later receives blocks 17-31, the chain led by block 31 includes $2N/3$ attestations with *target* as block 0. v_3 updates LJ to block 0.

Attack overview

- Idea: LJ might change in the middle of an epoch

- **Case 3) LJ is updated in the middle of an epoch**
  - Assume LJ == cp at epoch e
  - Blocks from previous epoch are withheld / delayed



(c) v_3 receives block 33 and the chain led by block 33 does not include $2N/3$ attestations. v_3 does not update its LJ. When v_3 later receives blocks 17-31, the chain led by block 31 includes $2N/3$ attestations with *target* as block 0. v_3 updates LJ to block 0.

Warm up attack

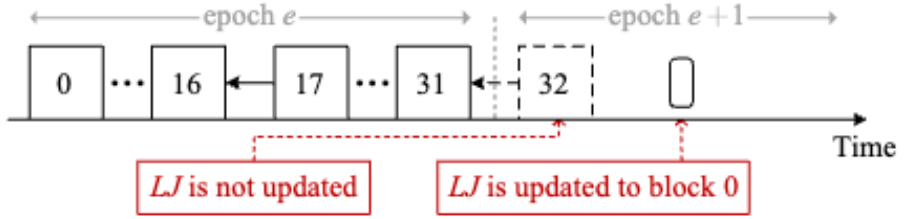
- **Goal: Make (N-f)/32 attestations to be discarded**

- 1) Attacker waits for an **opportune** epoch
  - 해당 에포크의 첫 슬롯의 제안자가 되는 에포크
- 2) Attacker creates and withholds its block *b_i*

Warm up attack

- **Goal: Make (N-f)/32 attestations to be discarded**

- 1) Attacker waits for an **opportune** epoch
  - 해당 에포크의 첫 슬롯의 제안자가 되는 에포크
- 2) Attacker creates and withholds its block *b_i* **<= Case 2!!**
  - Attestor들이 LJ를 업데이트 한 후, (0, 31)에 attest

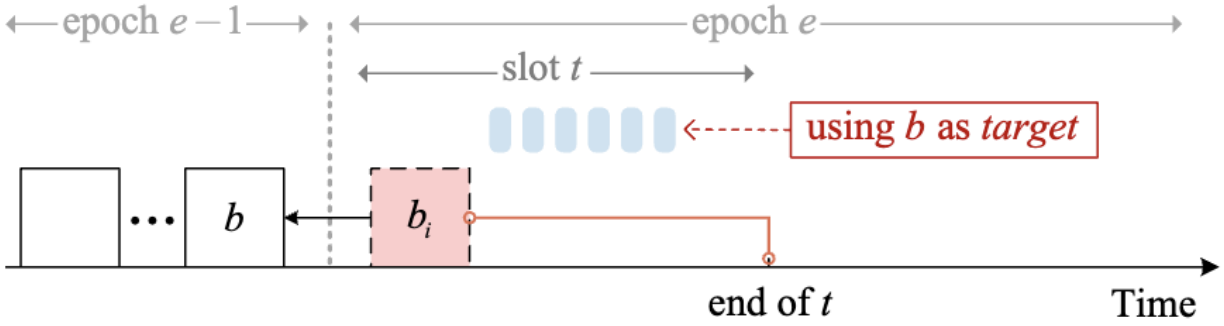


(b) v_2 has not received any blocks in epoch $e+1$ yet and does not update its *LJ* until it prepares an attestation. Before v_2 prepares its attestation, it updates its *source* as block 0 and *target* as block 31.

Warm up attack

- **Goal: Make (N-f)/32 attestations to be discarded**

- 1) Attacker waits for an **opportune** epoch
  - 해당 에포크의 첫 슬롯의 제안자가 되는 에포크
- 2) Attacker creates and withholds its block *b_i*
  - Attestor들이 LJ를 cp{e-1}로 업데이트 한 후, (cp{e-1}, b)에 attest

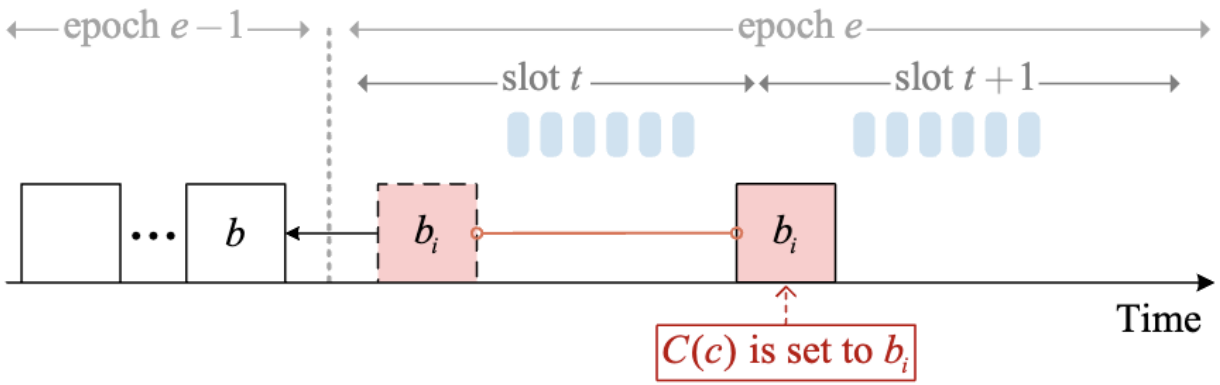


(a) Step 1: v_i withholds block b_i. Honest attestors in slot t create attestation with b as *target*.

Warm up attack

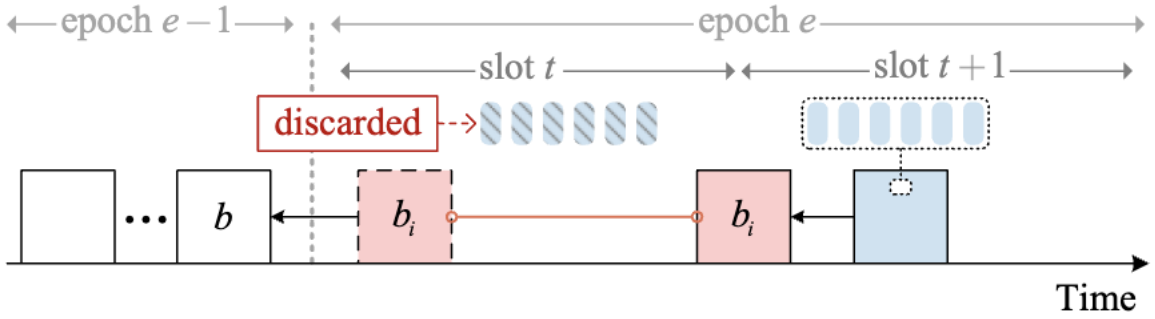
- **Goal: Make (N-f)/32 validators to be penalized**

- 1) Attacker waits for an **opportune** epoch
  - 해당 에포크의 첫 슬롯의 제안자가 되는 에포크
- 2) Attacker creates and withholds its block *b_i*
  - Attestor들이 LJ를 cp{e-1}로 업데이트 한 후, (cp{e-1}, b)에 attest
- 3) At the end of slot, the attacker sends *b_i* to all validators
  - All validators update their target to *b_i*



Warm up attack

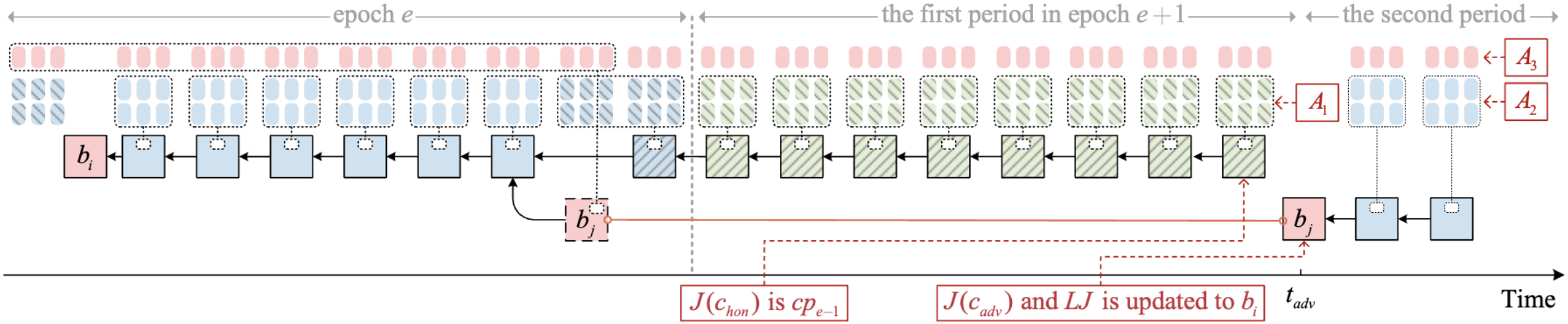
- **Goal: Make (N-f)/32 validators to be penalized**



- 1) Attacker waits for an **opportune** epoch
  - 해당 에포크의 첫 슬롯의 제안자가 되는 에포크
- 2) Attacker creates and withholds its block *b_i*
  - Attestor들이 LJ를 cp{e-1}로 업데이트 한 후, (cp{e-1}, b)에 attest
- 3) At the end of slot, the attacker sends *b_i* to all validators
  - All validators update their target to *b_i*
- 4) After slot t, any attestations created by honest attestors in slot t will be discarded
  - (N-f)/32 attestations

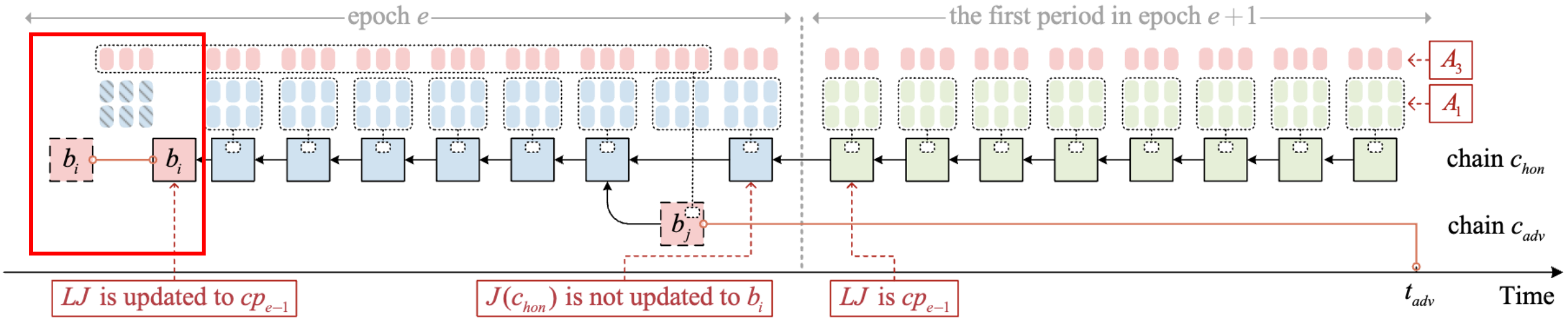
Staircase Attack

- **Idea**
  - **1)** By discarding (N-f)/32 votes through the warm-up attack, it prevents gathering 2N/3 proofs by the end of the epoch.
  - **2)** Due to the LJ value maintained by 1), it updates LJ in the middle of the epoch (Case 3), causing the honest branch to be pruned.



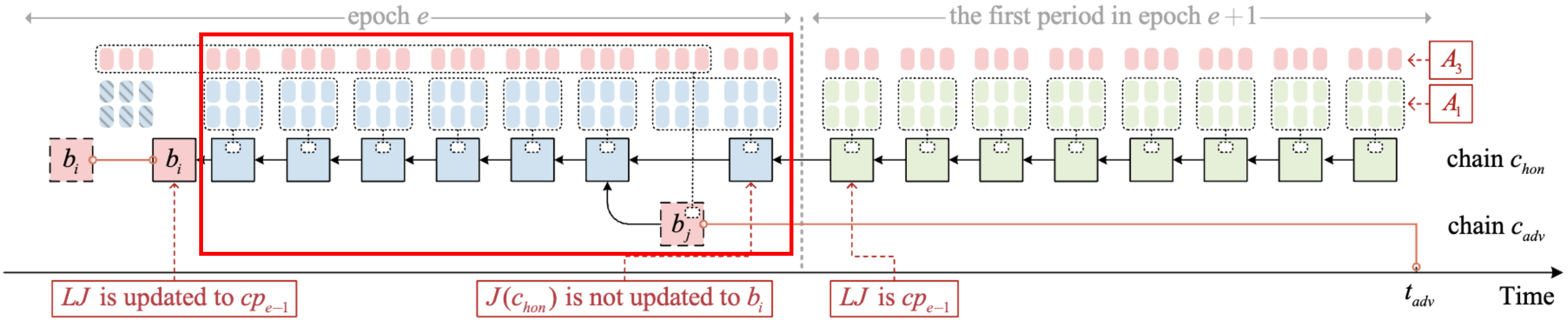
Staircase Attack

- **1) Warm-up Attack**
  - Slot t: Discard (N-f)/32 attestations using Warm-up attack



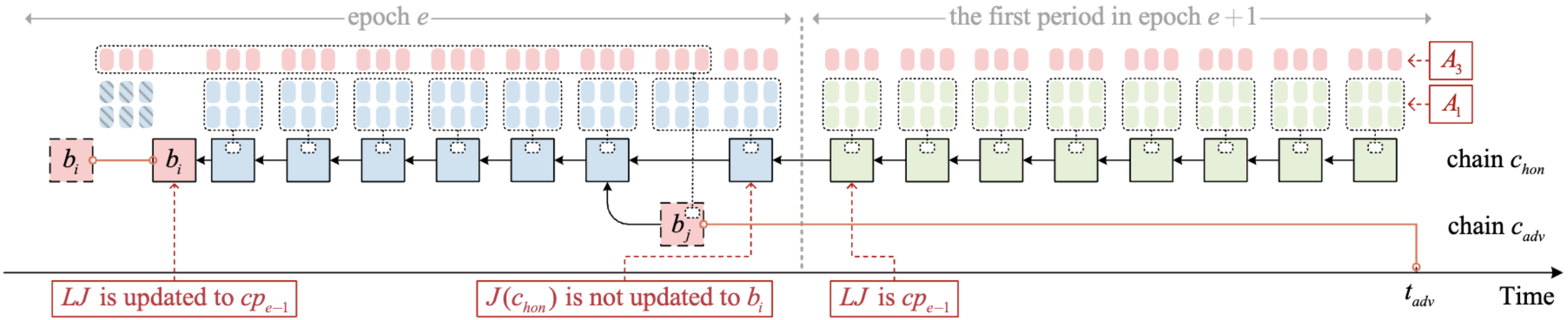
Staircase Attack

- **2) Byzantine validators hide their attestations/last block**
  - (1) All Byzantine validators : Withhold attestations
  - (2) Last Byzantine proposer (e.g. slot t_j) : Withhold b_j



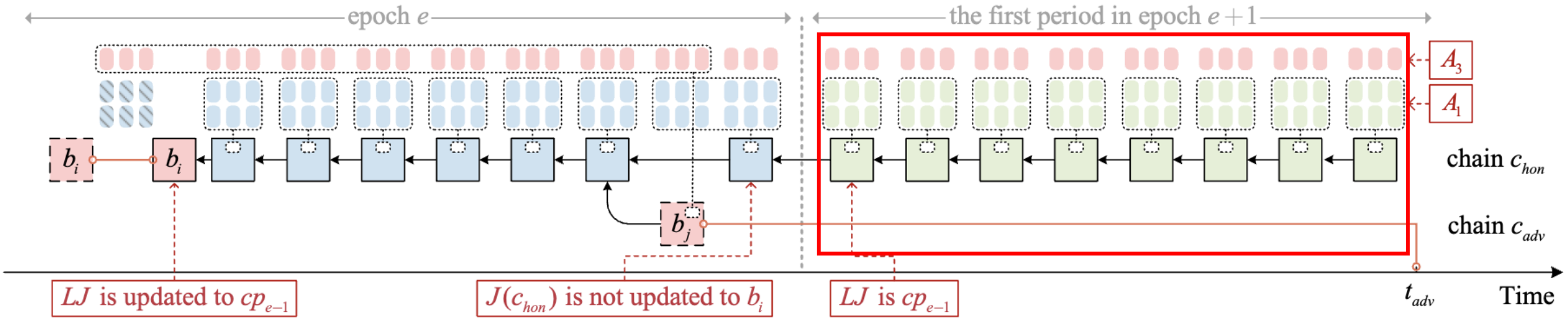
Staircase Attack

- **3) At the end of epoch e, LJ remains as cp{e-1}**
  - # of attestations for (cp_{e-1}, b_i) = 31(N-f)/32 < 2N/3 (f = N/3)



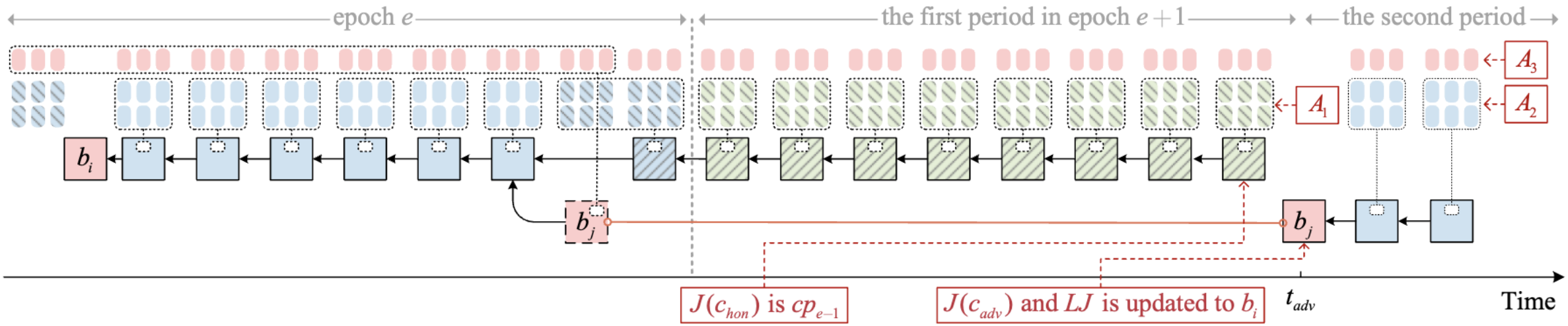
Staircase Attack

- **4) Byzantine validators withheld attestations for (b_i, b_j)**
  - While honest validators vote for (cp_{e-1}, b_i)
  - Byzantine proposer의 차례(t_adv)가 될 때까지 반복 (the first period)



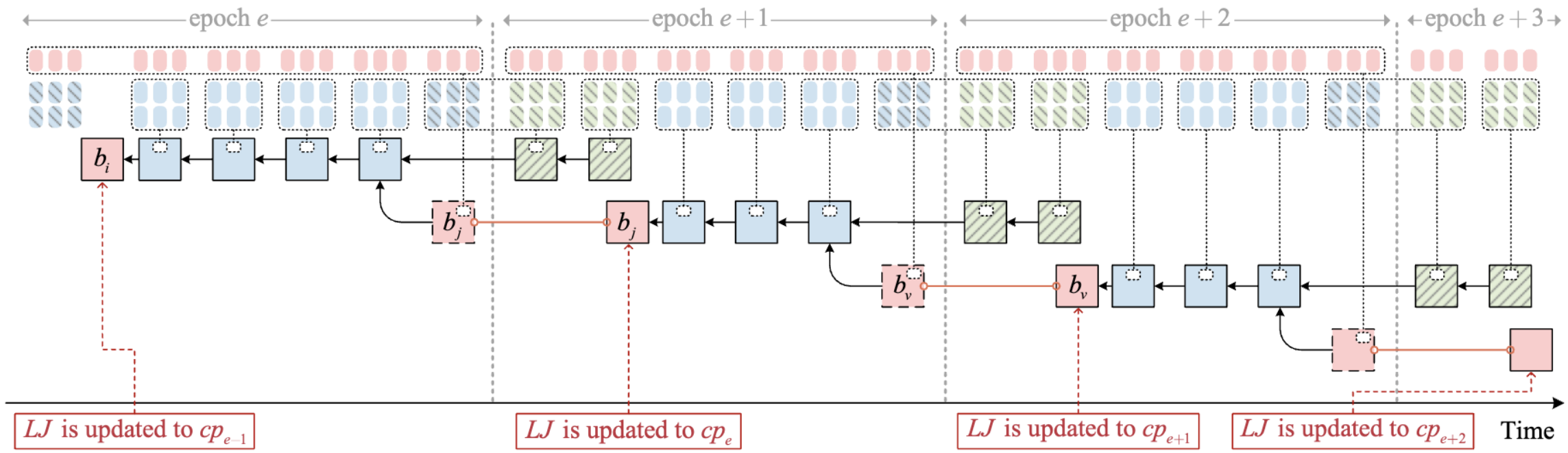
Staircase Attack

- **4) Release block b_j and attestations at the second period**
  - Byzantine proposer의 차례(t_adv)가 오면 b_j 및 숨겨진 attestation들을 공개
  - LJ를 b_i로 만들고 honset branch를 pruning
    - t_j-t-1 > 2/3 인 상황에서 성립
    - Pr[t_j-t-1 > 2/3] >= 98.84%



Staircase Attack

- **Attacker can repeat the attack in every epoch**



Evaluation

- Prysm Capella(Golang)
- 1000 validators connected by LAN

- F: # of Byzantine validators
- Incentive Loss Rate
  - (fair share – incentive) / fair share * 100

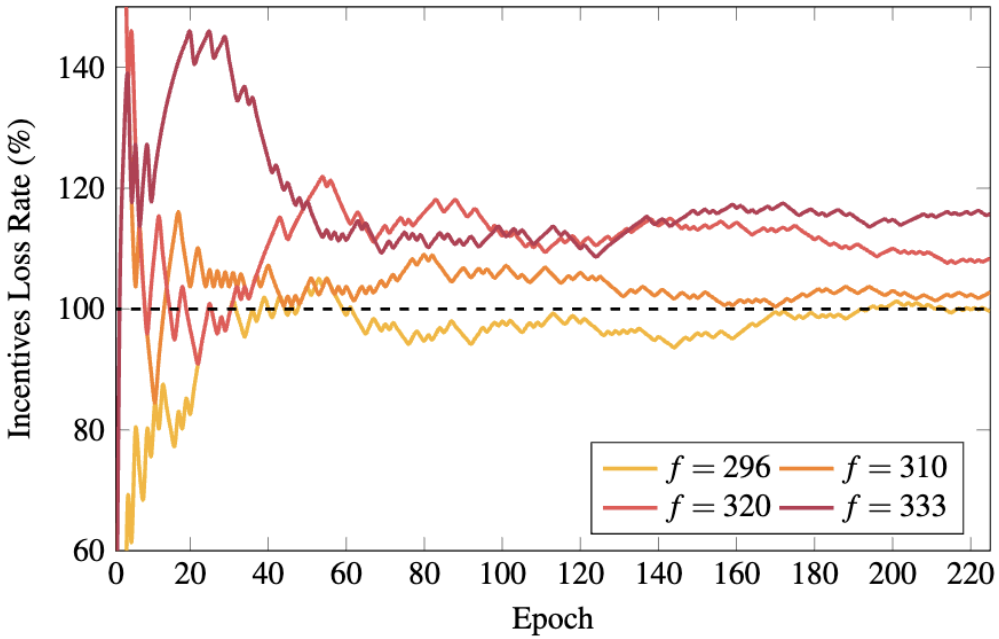


Figure 10: The incentives loss rate of an honest validator for experiments with 1,000 validators and f Byzantine validators. Each experiment is launched for one day (225 epochs).

Conclusion

- An adversary that controls 29.6% stake can make all honest validators lose their incentive
- Larger MAX_ATTESTATIONS make higher incentive loss