Provably Secure Blockchain Protocols from Distributed Proof-of-Deep-Learning

Xiangyu Su[1], Mario Larangeira[1, 2], and Keisuke Tanaka[1]

1. Tokyo Institute of Technology, Japan
2. Input Output Global, Singapore

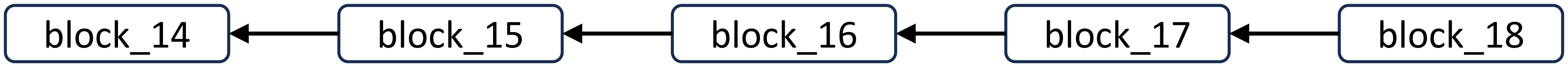
Tokyo Tech

Outline

- Background: Blockchain Basic
- Our Contributions:
  - Distributed Proof-of-Deep-Learning (D-PoDL) Scheme
  - Provably Secure D-PoDL-Based Blockchain Protocols
- Summary and Future Works

Blockchain Basic: Structure

- Data: message (transaction, tx) & block
- Structure: Hash chain of blocks



Blockchain Basic: Proof-of-Work (PoW)[1]

- Data: message (transaction, tx) ∈ block
- Structure: Chaining blocks with hash
- PoW: Block generation with parameter T

block_14 ← block_15 ← block_16 ← block_17 ← block_18

Find <u>nonce</u>, s.t., hash(prevBK, <u>nonce</u>) $\leq$ T

Blockchain Basic: PoW

- Data: message (transaction, tx) ∈ block
- Structure: Chaining blocks with hash
- PoW: Block generation with parameter T

block_14 ← block_15 ← block_16 ← block_17 ← block_18

Find <u>nonce</u>, s.t., hash(prevBK, <u>nonce</u>) ≤ T

- For hash: $\{0,1\}^*$->$\{0,1\}^n$, PoW is expected to require $\frac{2^n}{T}$-hash evaluations

Blockchain Basic: PoW

- Data: message (transaction, tx) ∈ block
- Structure: Chaining blocks with hash
- PoW: Block generation with parameter T

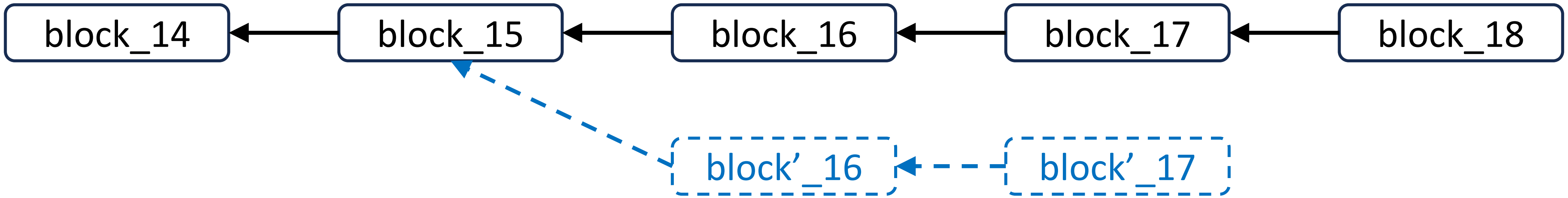
  => Have ``enough (?)'' time to send blocks

Blockchain Basic: PoW

- Data: message (transaction, tx) ∈ block
- Structure: Chaining blocks with hash
- PoW: Block generation with parameter T

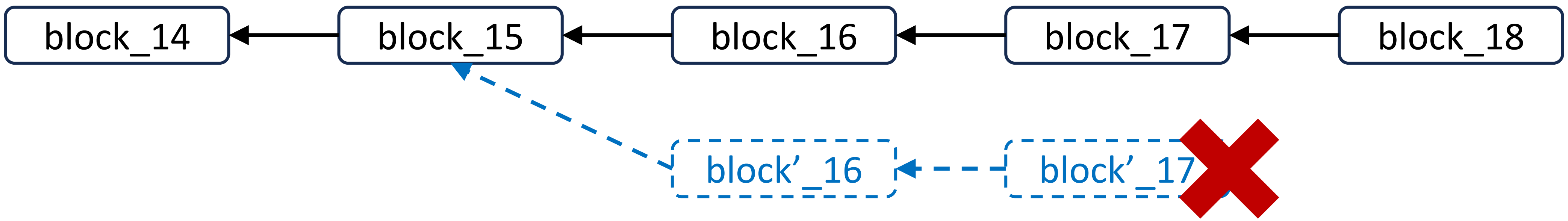
  => Have ``enough (?)'' time to send blocks

  => Forks



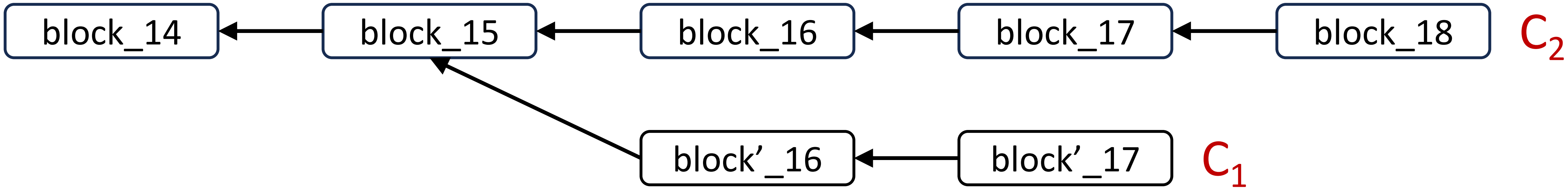
Blockchain Basic: Forks and Chain Selections

- The longest-chain-rule[2] and the weight-based selection[3,4]



- Why is the fork guaranteed to die out?
  - PoW is bounded by computing power
  - The honest majority assumption (Up-to 1/2 corruption)

Blockchain Basic: Security[2]



- Persistence: For any honest chain C_1 and C_2 in time slot $t_1<t_2$, after pruning several latest blocks in C_1, C_1 is the prefix of C_2

- Liveness: Any honest message (tx) will eventually be embedded in all honest users' blockchain

A Problem of the PoW

Find nonce, s.t., hash(prevBK, nonce) $\leq$ T

- That hash iterations seem quite wasteful
  - And rather meaningless outside the PoW
- Can we replace it with something more useful?

A Problem of the PoW

Find nonce, s.t., hash(prevBK, nonce) $\leq$ T

- That hash iterations seem quite wasteful
  - And rather meaningless outside the PoW
- Can we replace it with something more useful?

=> Proof-of-useful-work (PoUW)[5,6]

Given task, run Solve(task) -> proof, s.t.,
Verify(task, proof) = 1

The Useful Work

- Worst-case assumptions fine-grained complexity theory [5]
- Stochastic local search algorithm[6]
- Deep Learning (DL) Tasks
  - A model should be accurate enough to be useful
  - Training a model requires sufficient computing power

=> Proof-of-Deep-Learning (PoDL)[7-10]

General Setup from Existing Works

- Participants: Task publishers, provers, and verifiers
- Task: (dataset D, accuracy threshold T_{acc})
- Prover Goal: Find a model that has accuracy surpassing T_{acc}

A Few Drawbacks

- Strong Assumptions:
  - Separation between publisher and prover[7,9,10]
  - Strong synchronous to publish test dataset[7,8]

- Some Waste Computing Power:
  - ``Somehow'' trained models cannot be reused[7-10*]

- No Explicit Security Analysis[7-10*]

*10 considered pre-determined short-term targets;
*10 has proof against double spending attack.

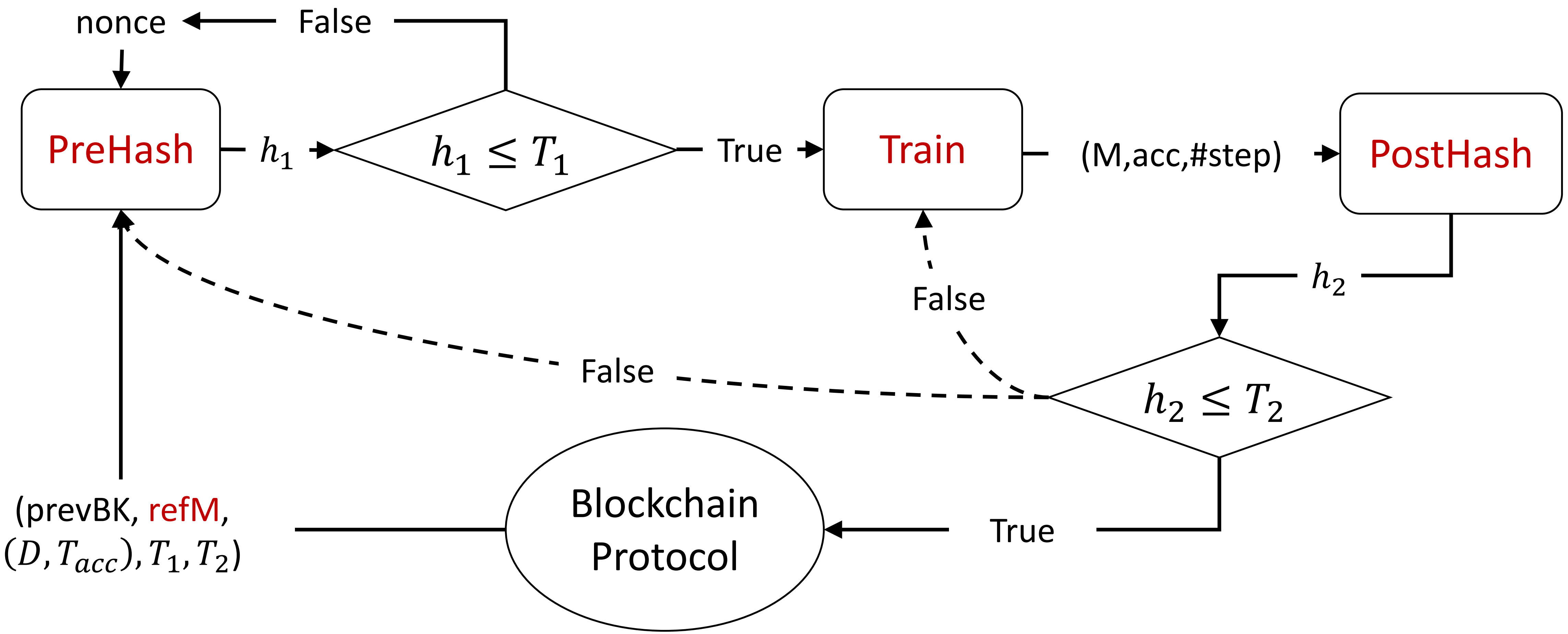
Additional Requirements

- No-grinding attack (cherry-picking parameters)
- Pre-computation resilience
- Adjustable difficulty
- Efficient verification
- Usefulness measurement

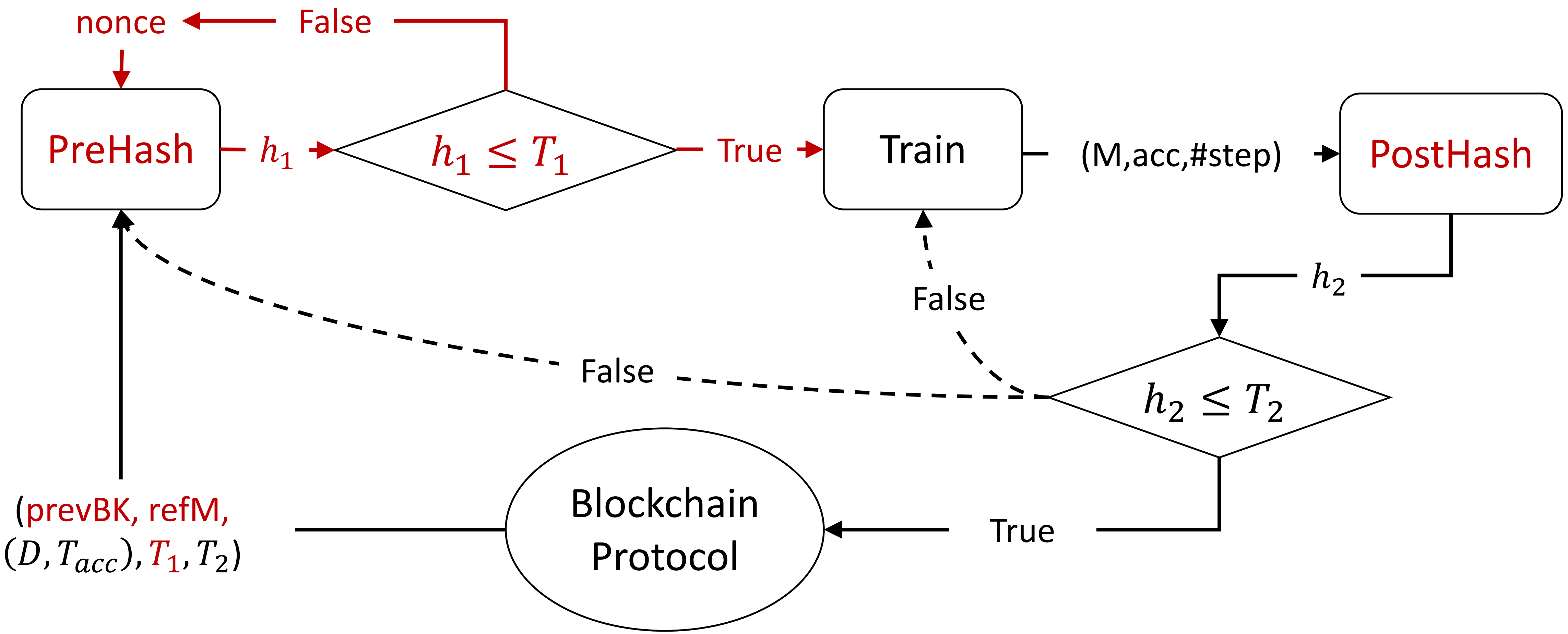
Our Approach (Intuition)

- Setting:
  - Focus on training dataset and accuracy
  - AND consider test ones to prevent overfitting (in protocol)
- Goal: Distribute task solving among provers (D-PoDL)
  - Hash-training-hash structure
  - Model-referencing mechanism

Scheme Overview

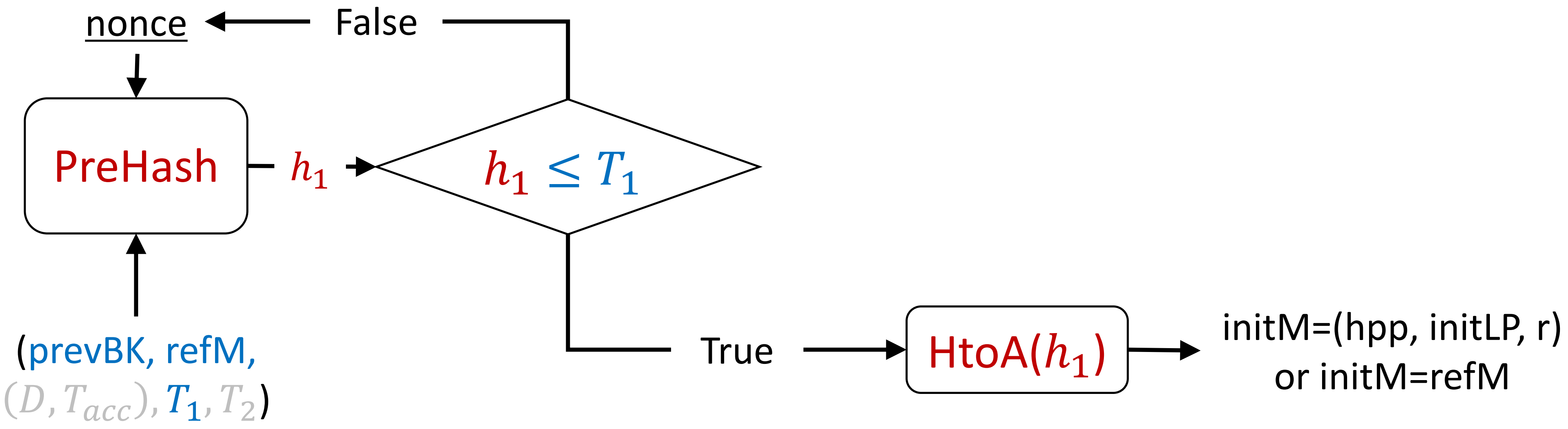


Scheme Overview



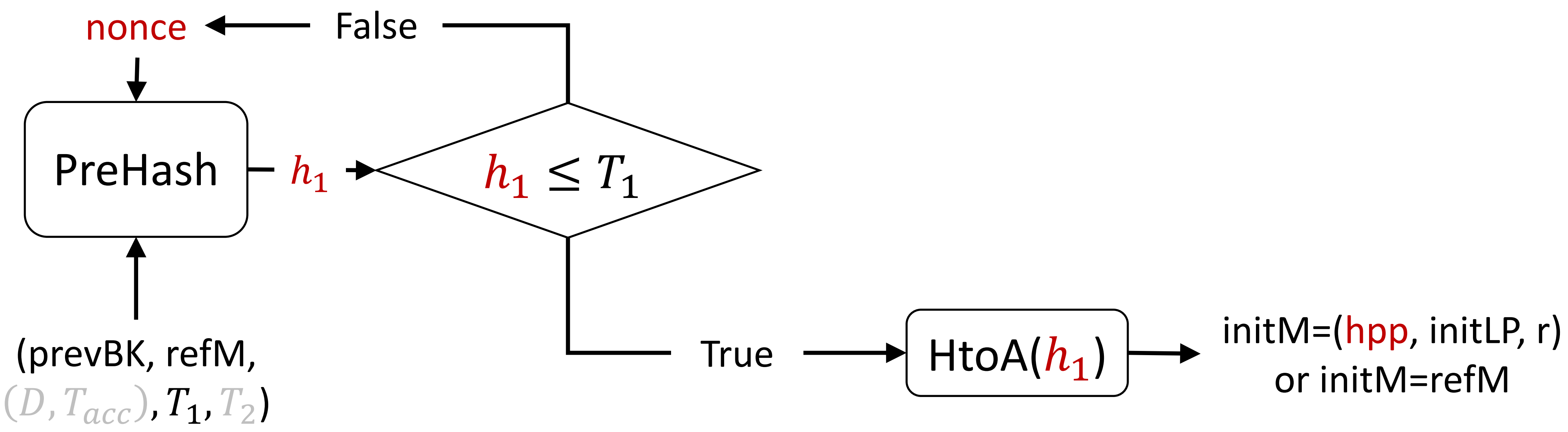
The PreHash Algorithm

- A PoW with low difficulty T_1
- Hash-to-architecture mapping: HtoA(h_1)=(hpp, initLP, r)
- Prevent grinding attack and pre-computation attack



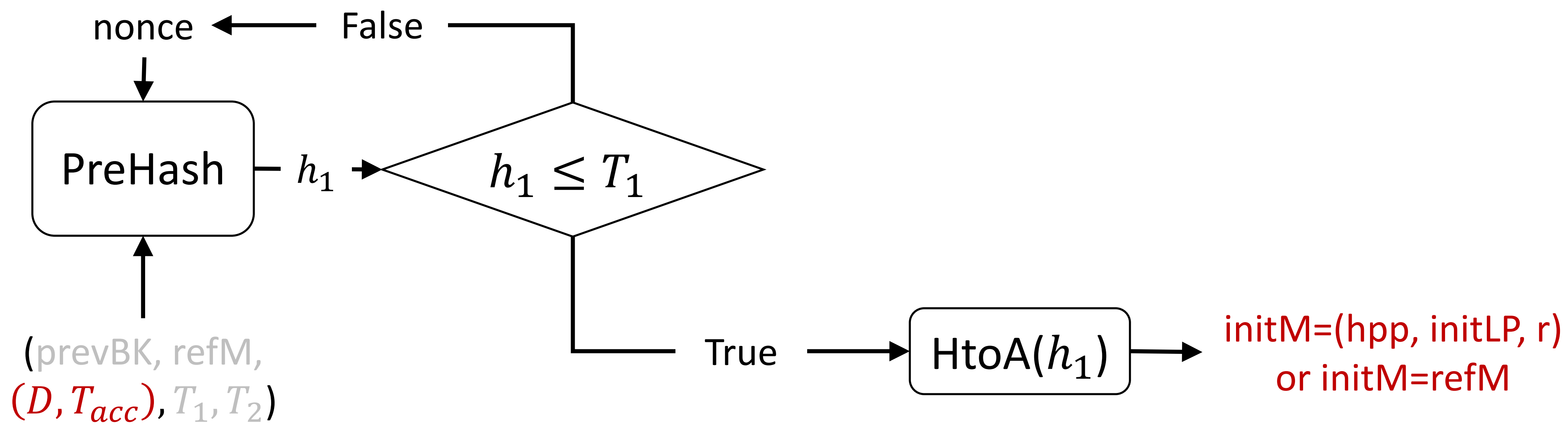
The PreHash Algorithm

- A PoW with low difficulty T_1
- Hash-to-architecture mapping: HtoA(h_1)=(hpp, initLP, r)
- Prevent grinding attack and pre-computation attack



The PreHash Algorithm

- A PoW with low difficulty T_1
- Hash-to-architecture mapping: HtoA(h_1)=(hpp, initLP, r)
- Prevent grinding attack and pre-computation attack

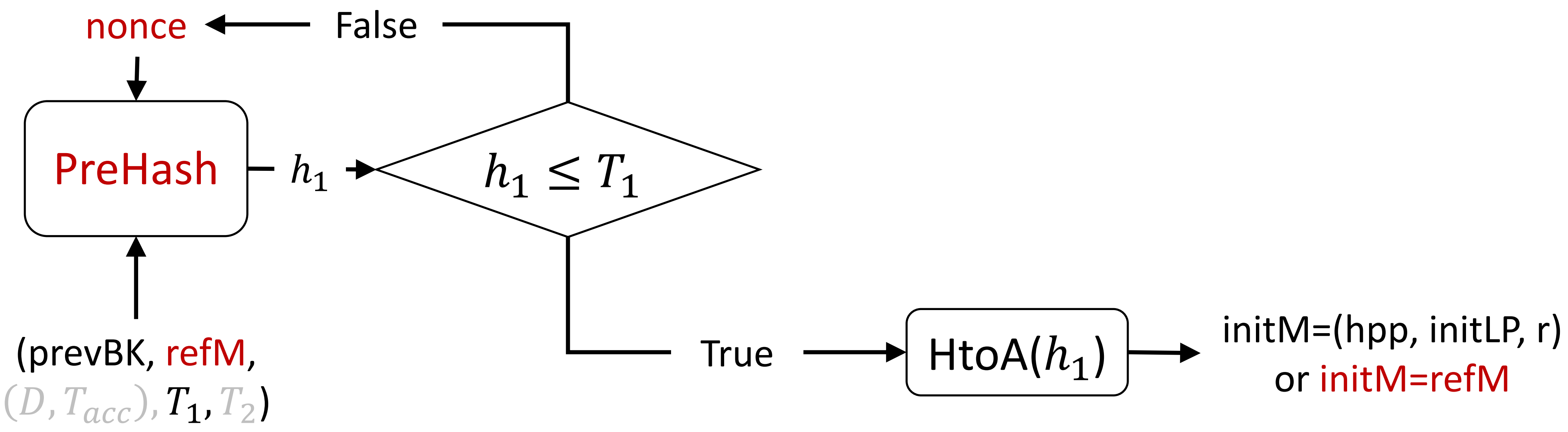


Model-Referencing in PreHash

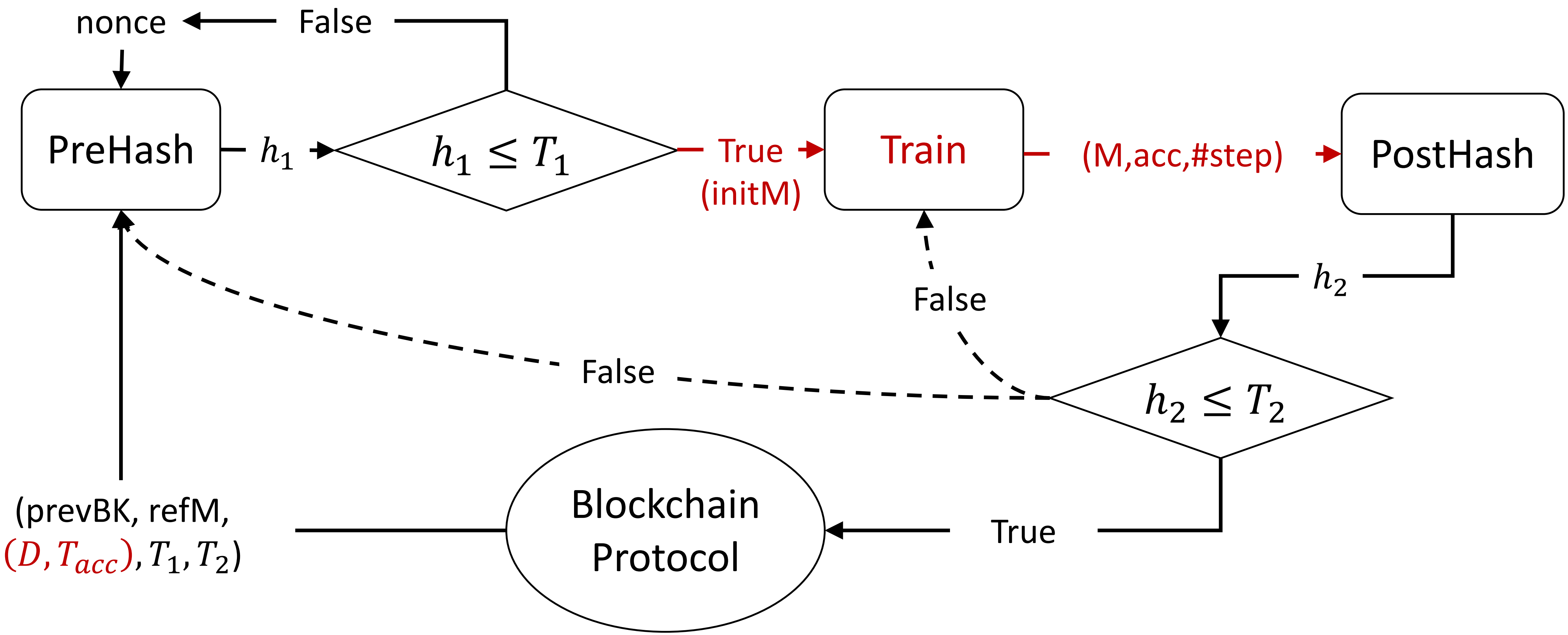
- Usually, training others model is forbidden
  - Achieve similar accuracy with less training iterations

  => However, pre-trained models are wasted

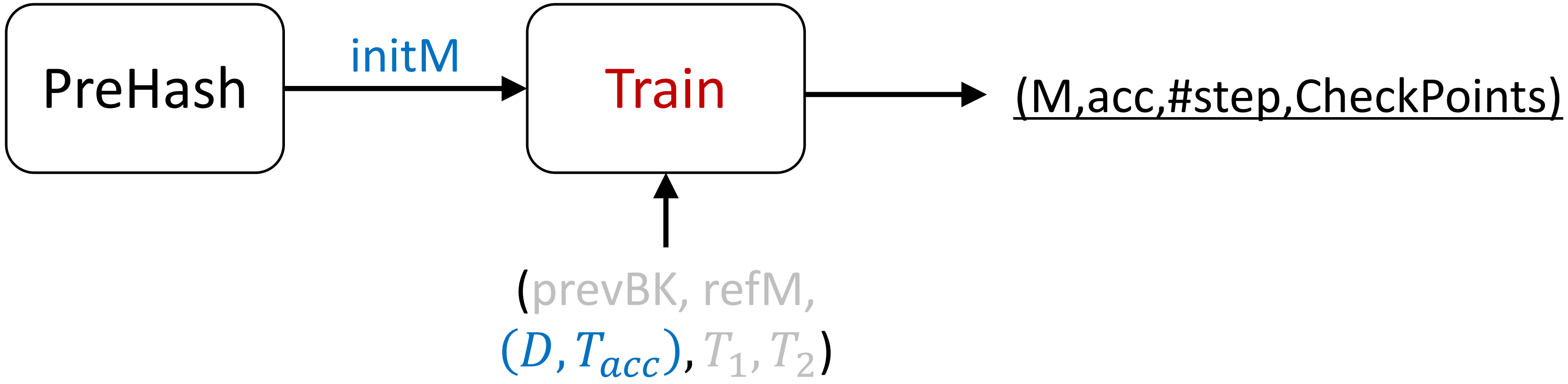
=> Provers should make clear references



Scheme Overview

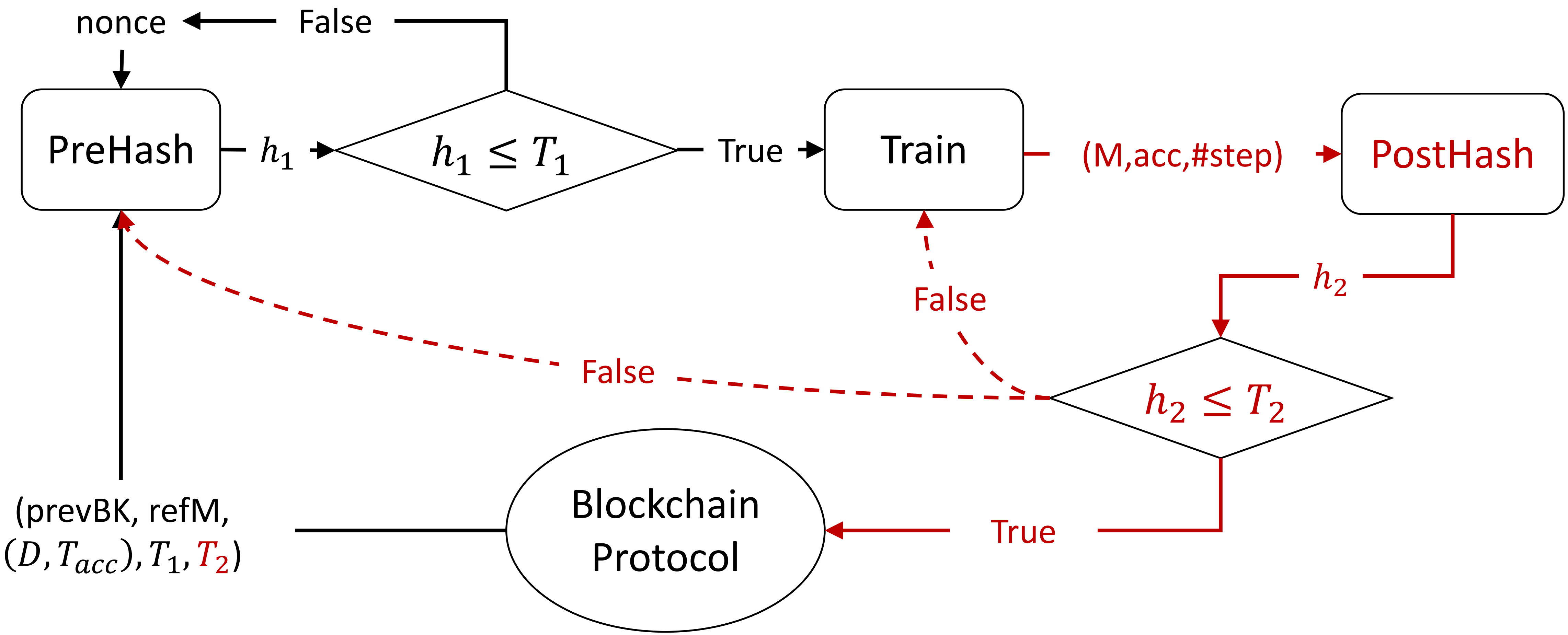


Main Training Algorithm



- Choose training algorithm e.g., the SGD algorithm
- Result model: M=(hpp, lp*),
- Corresponding accuracy and step number: (acc, #step)
- Checkpoints: CPs={(M_i, acc_i, #$step_i$)}

Scheme Overview

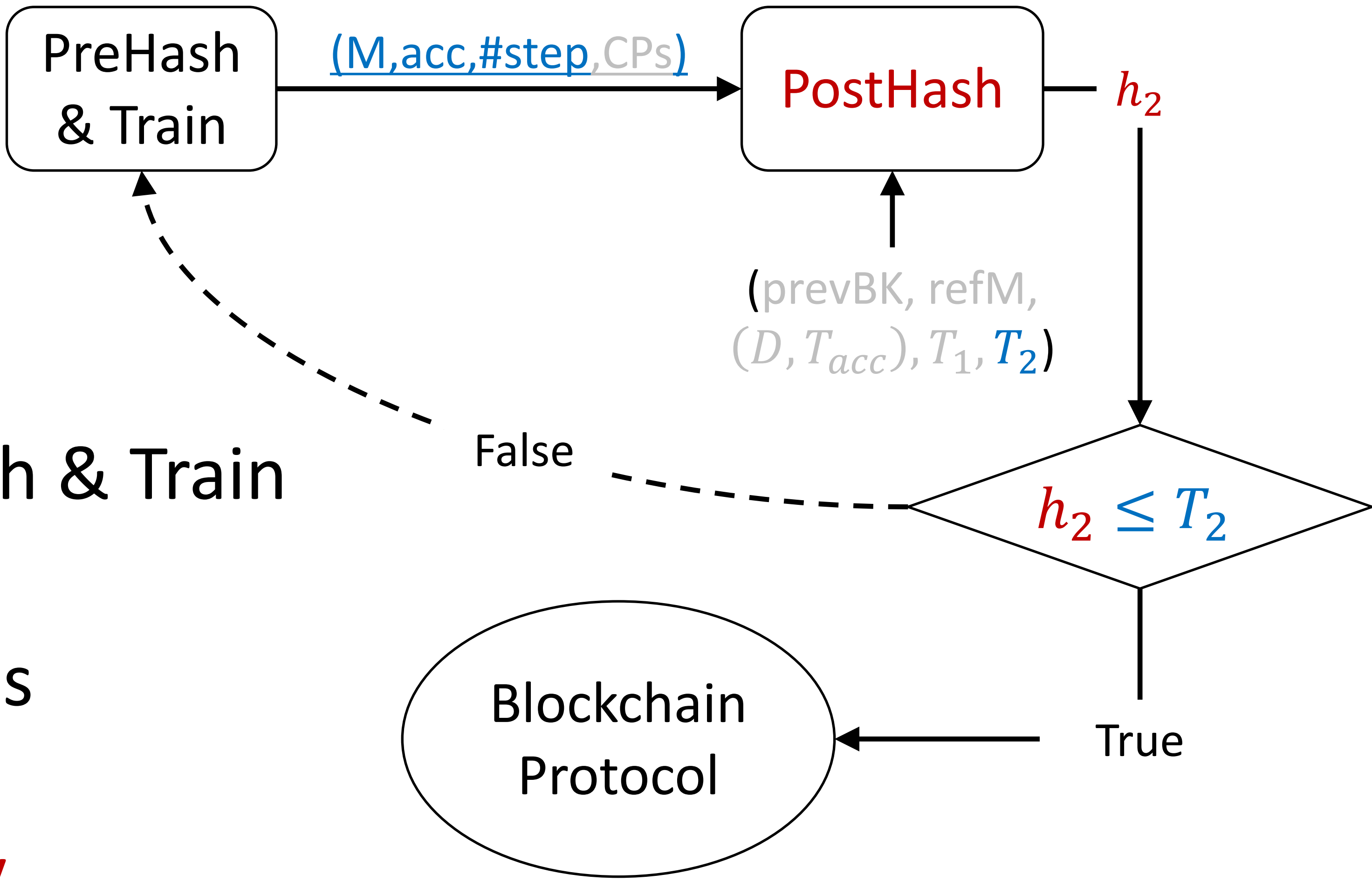


The PostHash Algorithm

- One hash check
- If true, publish
- If false, return to PreHash & Train

=> More training iterations

=> Adjust overall difficulty



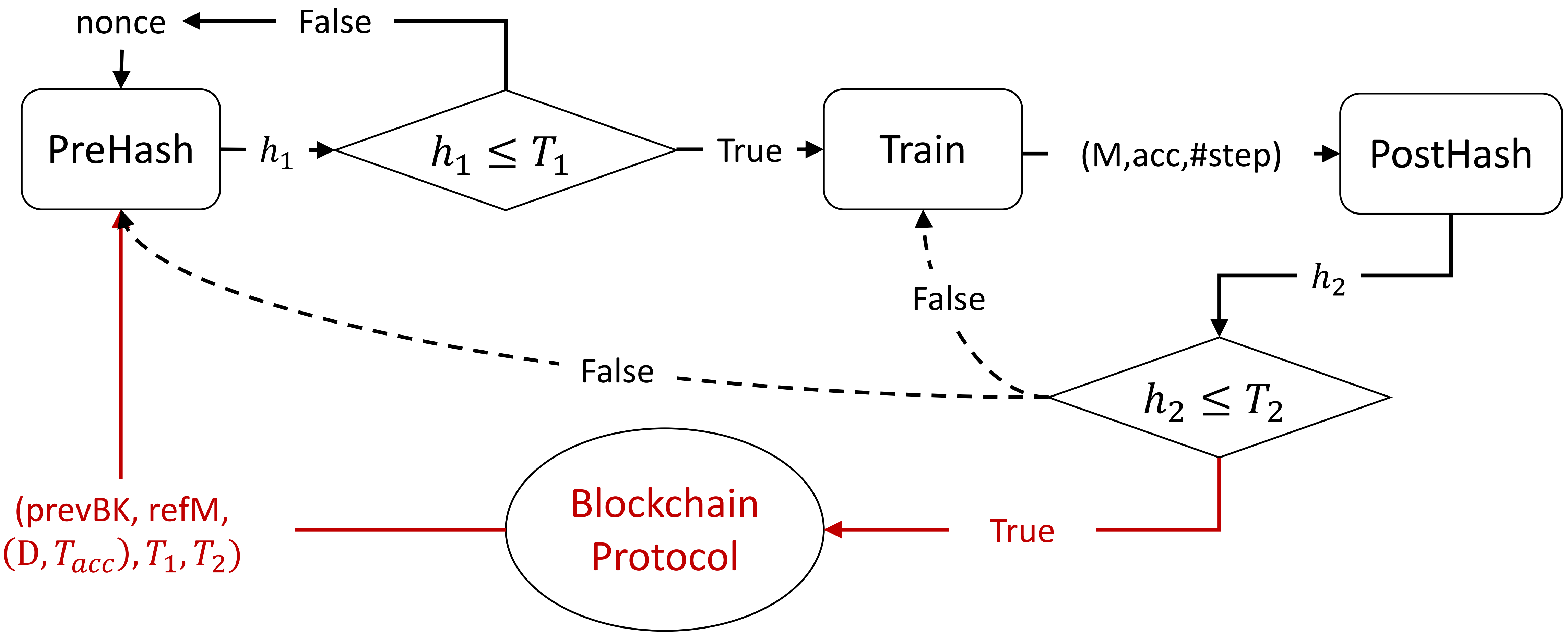
Model Verification

- Naïve approach: Reproduce the whole training

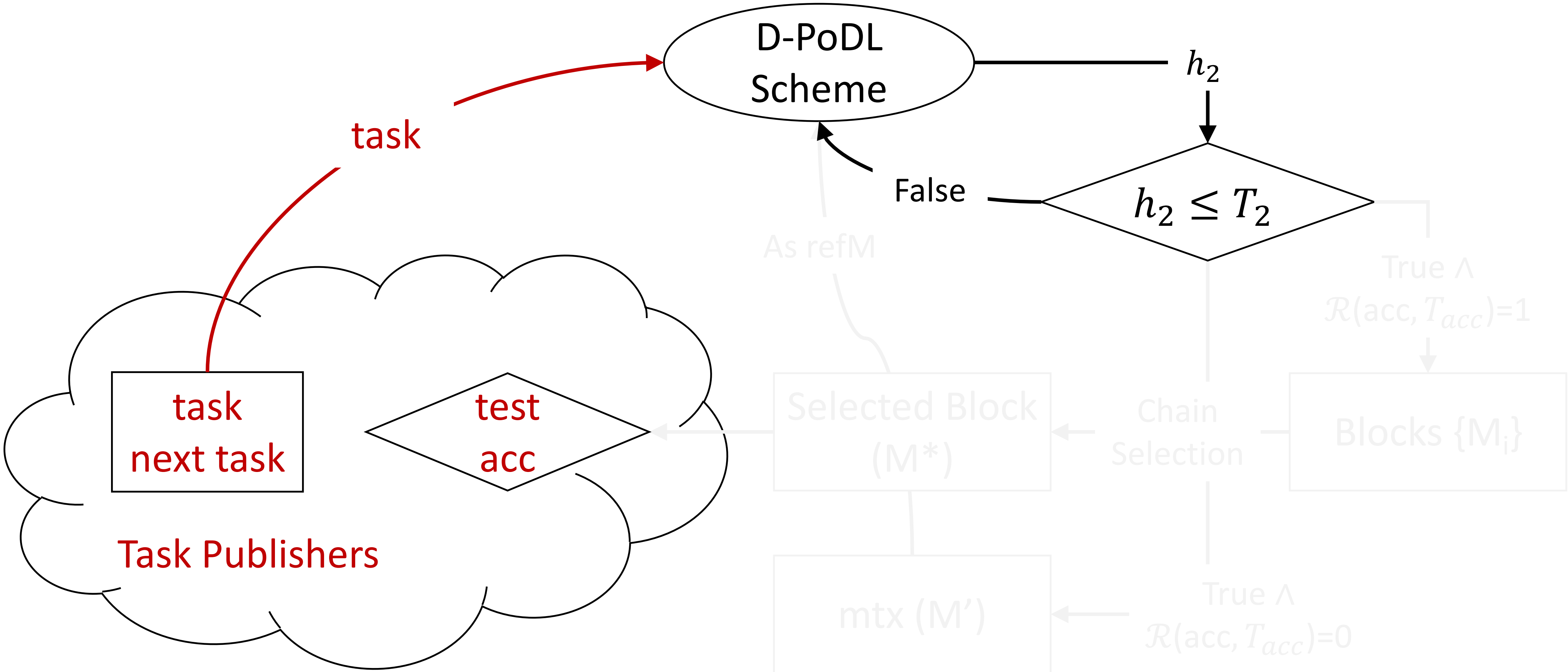
- Considering efficiency

=> Merkle-tree-based verification on checkpoints[10,11]

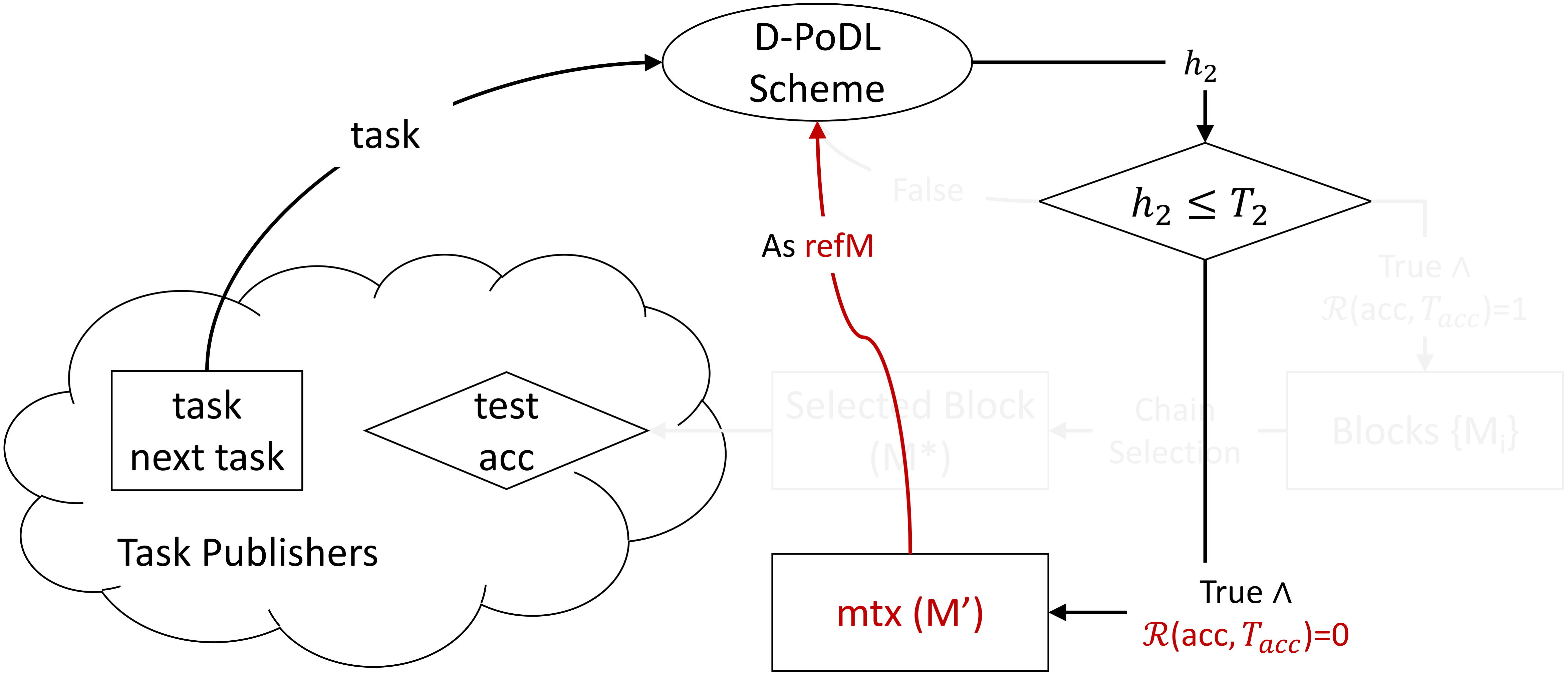
Transform to Blockchain Protocol



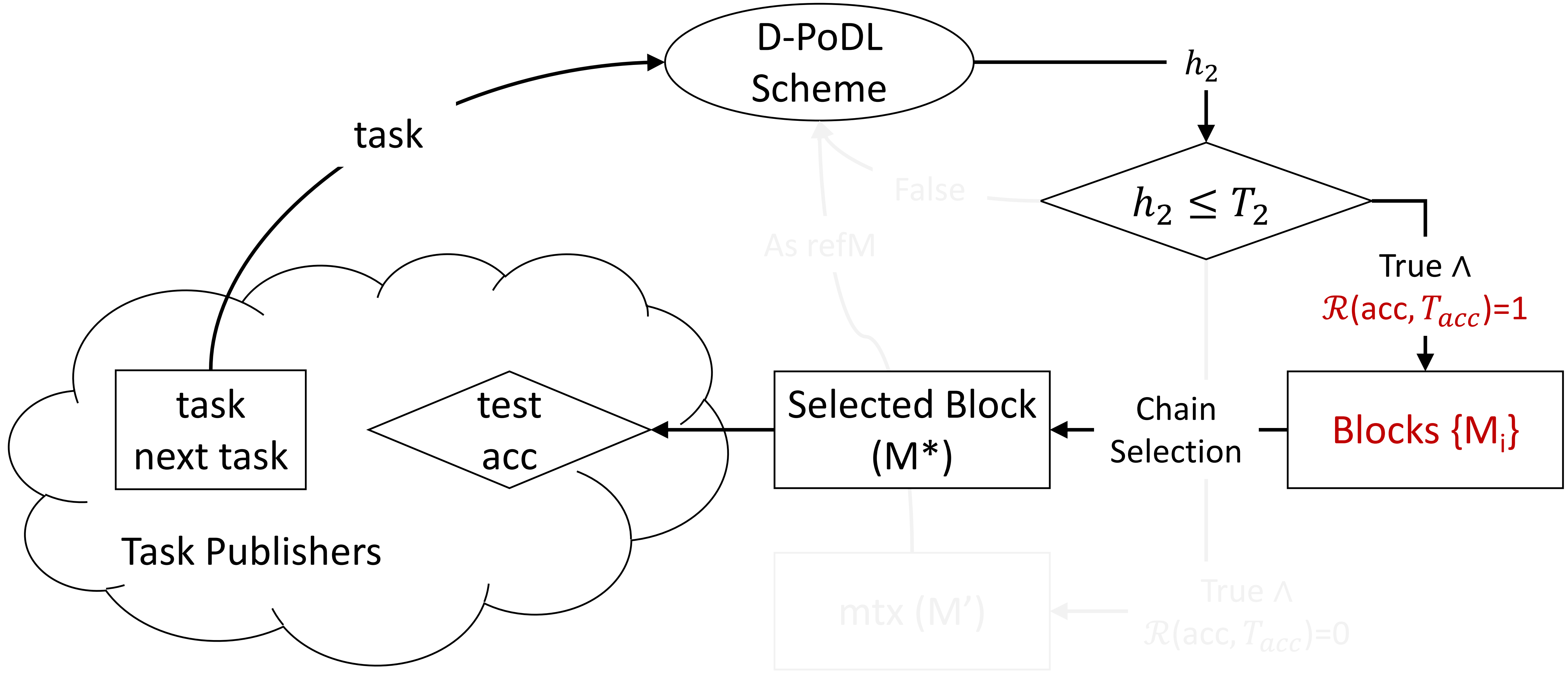
Publishers and Tasks



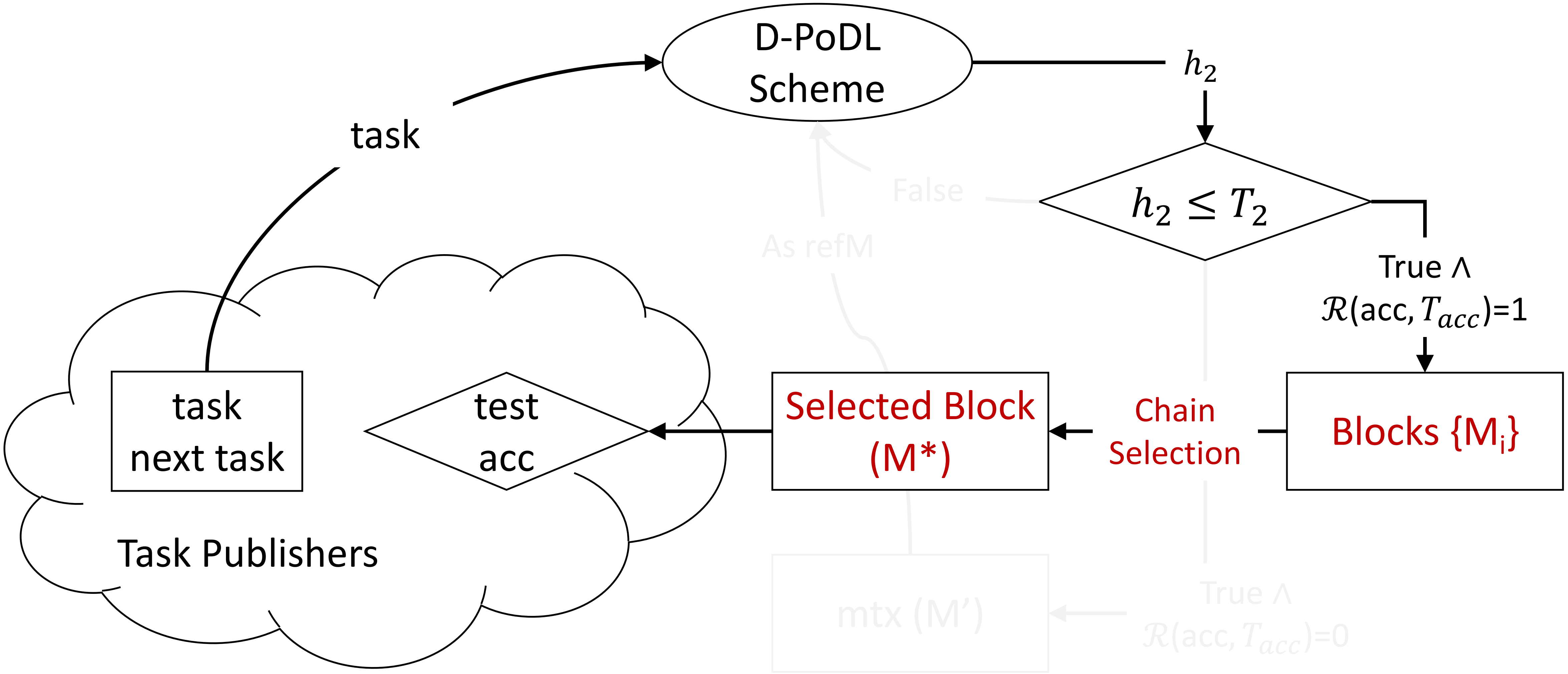
Referred Models: Model Transaction (mtx)



Block Generation



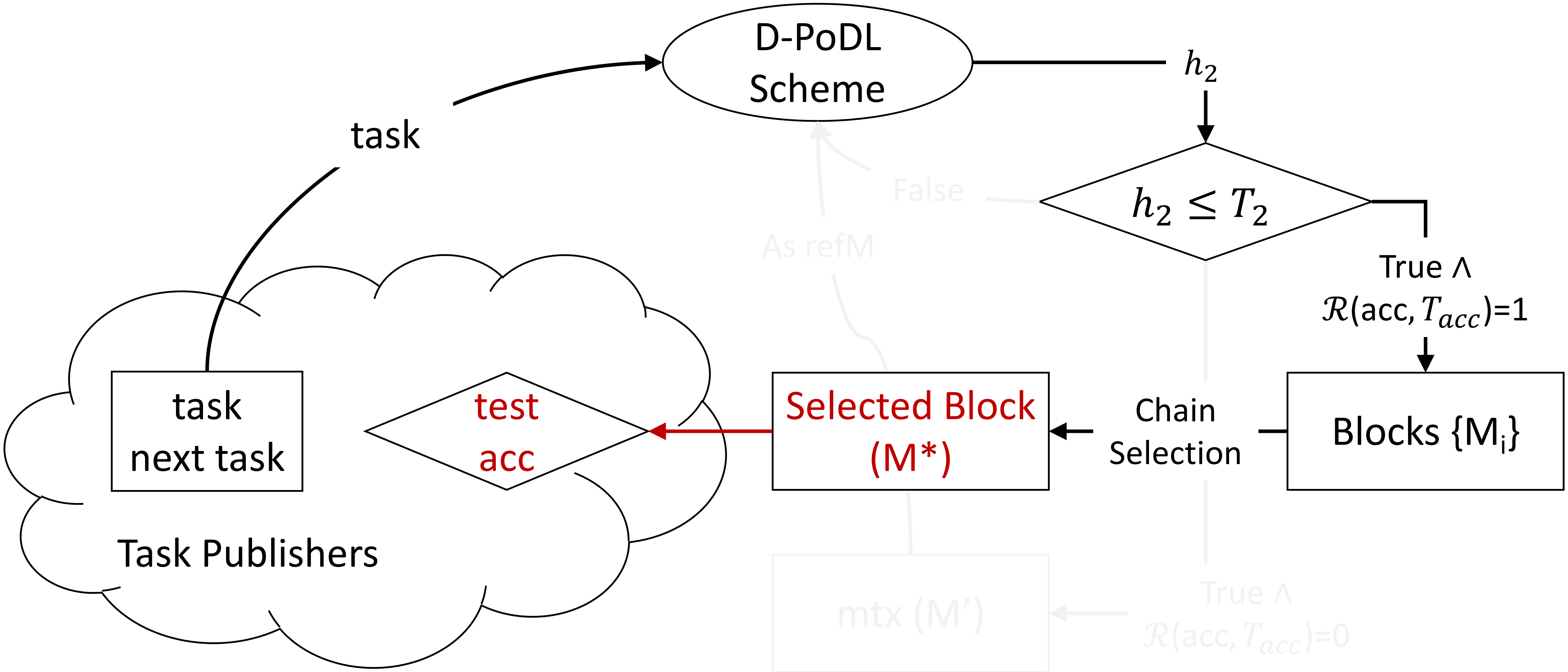
Chain Selection



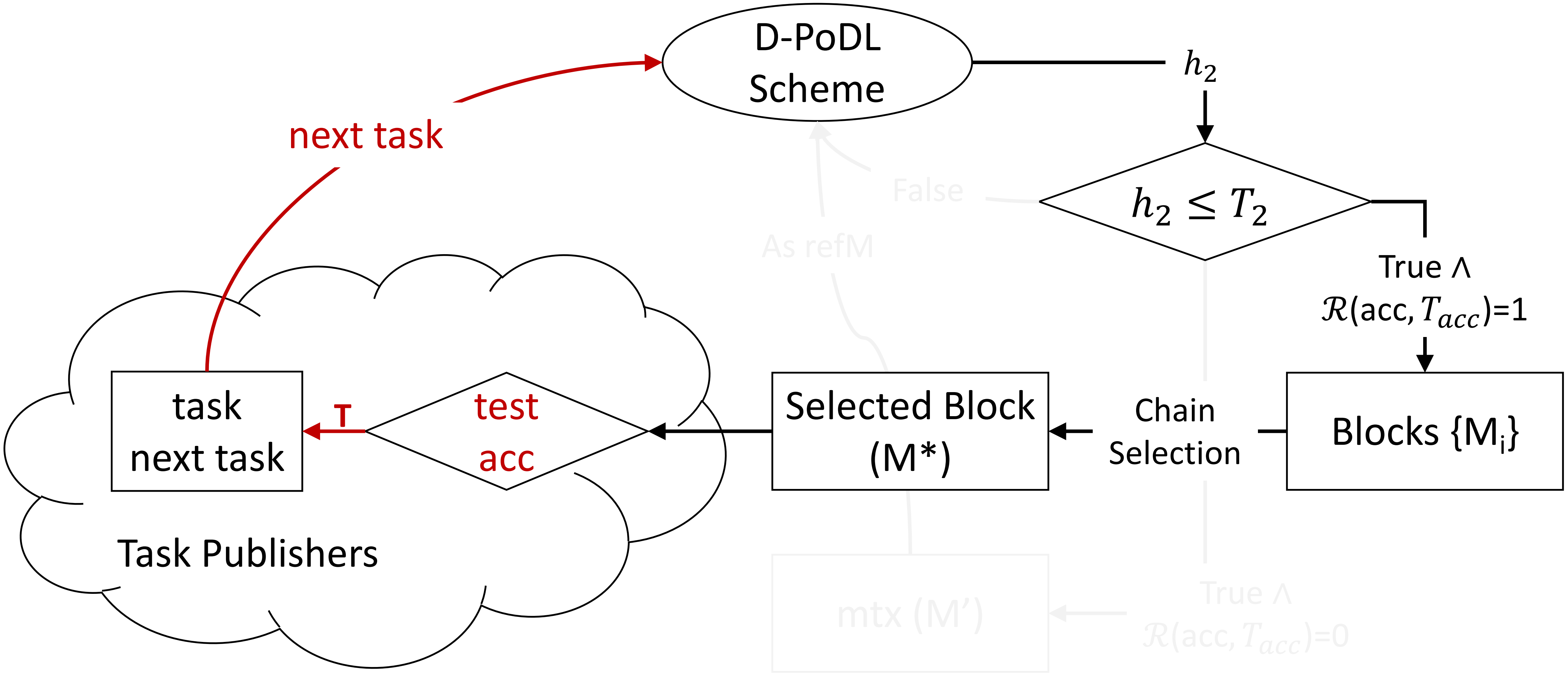
Concrete Chain Selection Rules

- Longest-Chain Rule[2]:
  - $\mathcal{R}(\text{acc}, T_{acc})$=1 if acc $\geq T_{acc}$; Otherwise $\mathcal{R}(\text{acc}, T_{acc})$=0
  - Miners choose the longest blockchain
- Weight-Based Framework[4]:
  - Assign weight to blocks according to $\mathcal{R}(\text{acc}, T_{acc})$
  - Lower accuracy has lower weight
  - Miners choose the heaviest blockchain

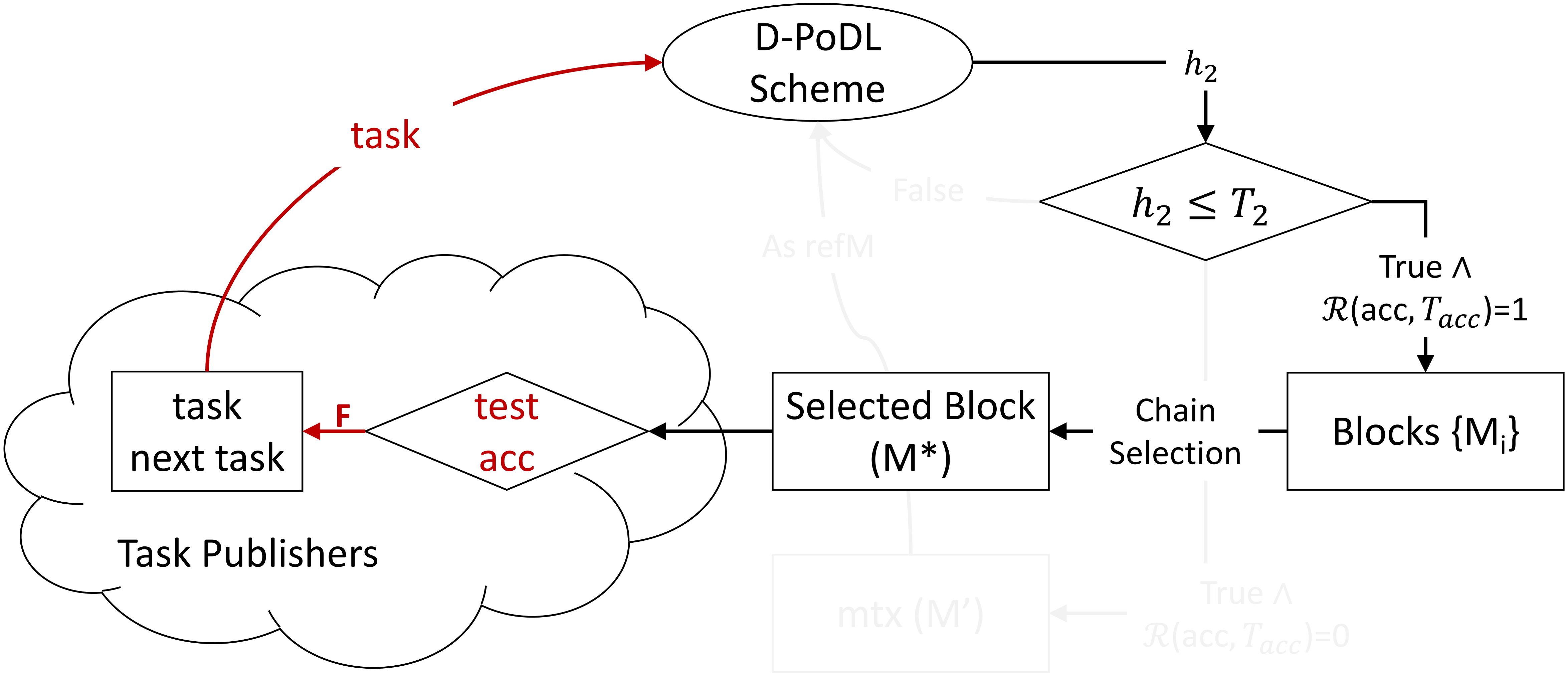
Consider Test Dataset after Chain Selection



Consider Test Dataset after Chain Selection

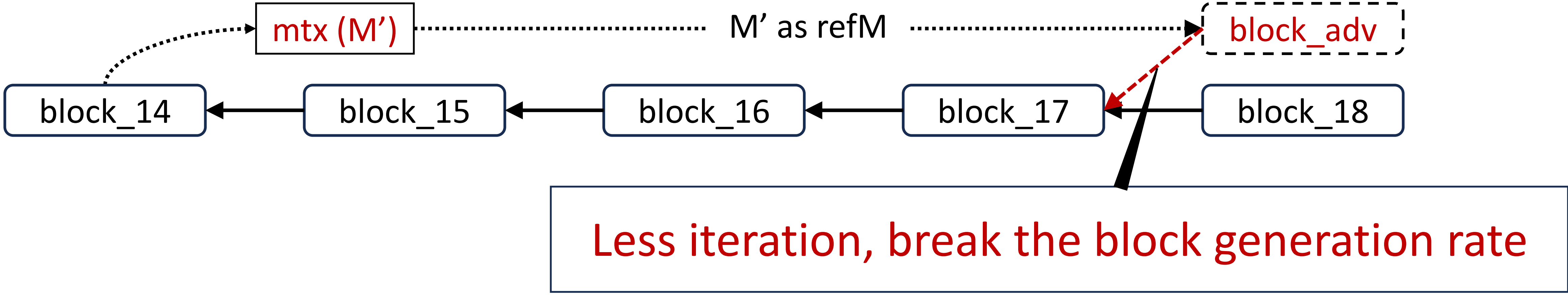


Consider Test Dataset after Chain Selection



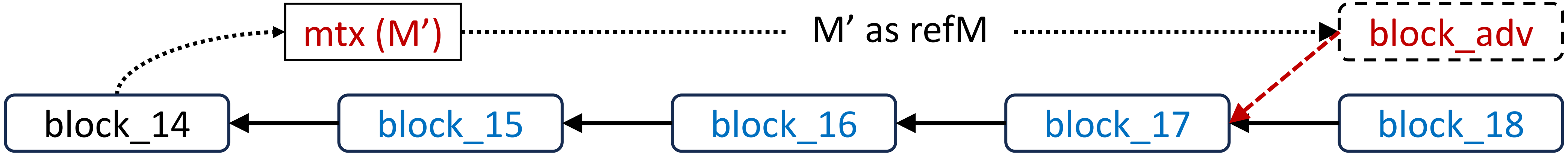
Cross Time Slot Attacks

- Refer to old/new models, and extend new/old blocks



Cross Time Slot Attacks

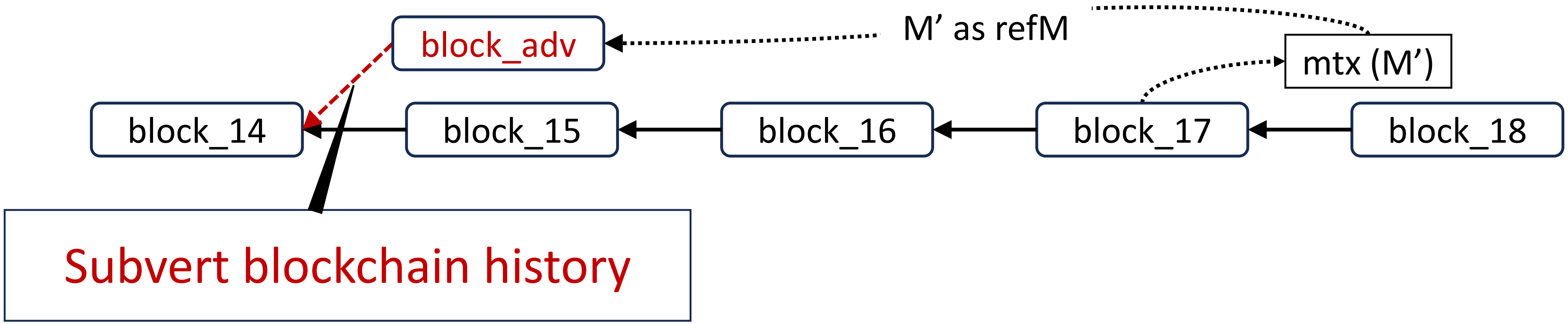
- Refer to old/new models, and extend new/old blocks



- Mitigation: Restrict step number in block_adv
- (#step_adv + #step_M') cannot be significantly less

Cross Time Slot Attacks

- Refer to old/new models, and extend new/old blocks



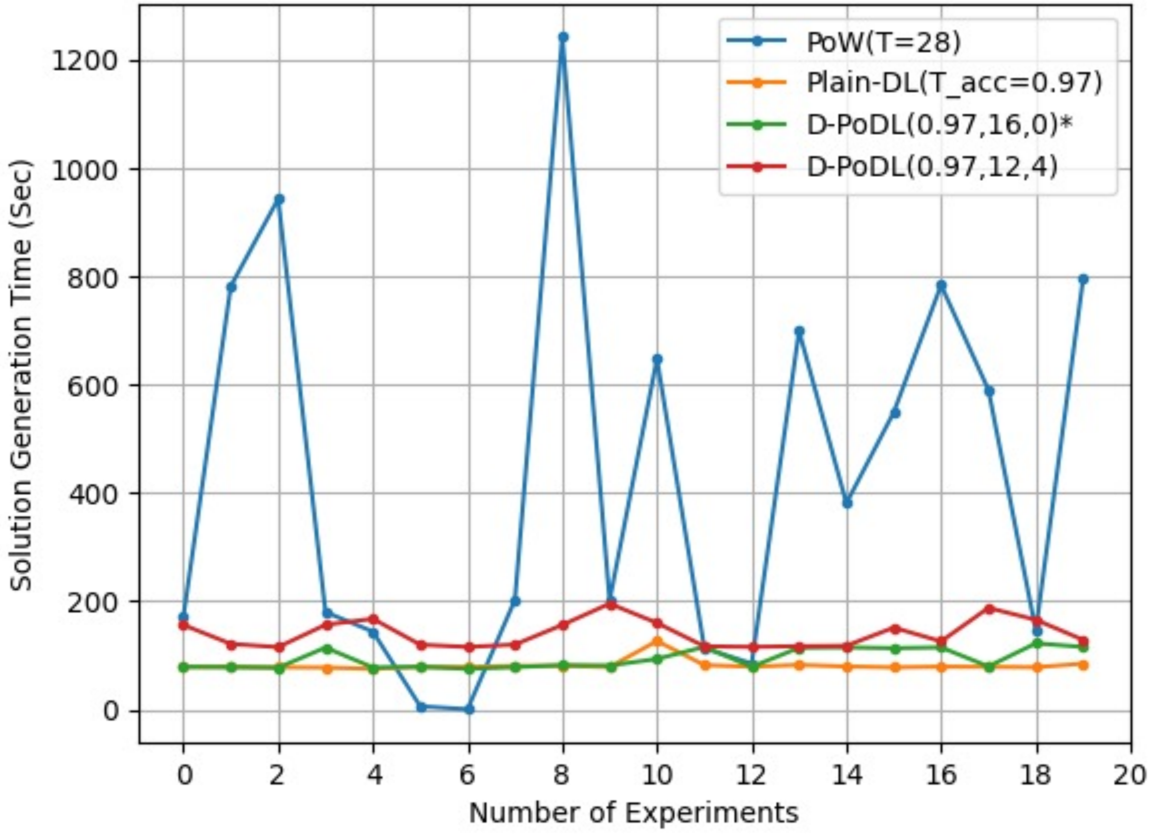
- <u>Mitigation</u>: Restrict reference

Security for D-PoDL-Based Blockchain

- Good period: Block generation follows an expected rate
- Good period guarantees persistence and liveness[2]
- Probability of periods being good is 1 minus negligible

Implementation of D-PoDL Scheme

- Compare to PoW and plain DL tasks (MNIST dataset)
- Stable rate with enough randomness to prevent domination



*D-PoDL parameter follows (T_{acc}, T_1, T_2)

Summary

- A design for distributed PoUW based on DL tasks (D-PoDL)
- Blockchain with different selection rules from the D-PoDL
- Prove security for the protocol and implement the scheme

Future Works

- Checkpoints are storage-demanding

=> Potential for proof-of-space[12]

- Parameter adjustment is hard

=> Feedback loops

- Incentive model and rational analysis

Thank you!

References

[1] Pricing via Processing or Combatting Junk Mail. Cynthia Dwork and Moni Naor.

[2] The Bitcoin Backbone Protocol: Analysis and Applications. Juan A. Garay et al.

[3] Weight-Based Nakamoto-Style Blockchains. Simon Holmgaard Kamp et al.

[4] Bootstrapping the blockchain, with applications to consensus and fast PKI setup. Juan A. Garay et al.

[5] Proofs of Work From Worst-Case Assumptions. Marshall Ball et al.

[6] Ofelimos: Combinatorial Optimization via Proof-of-Useful-Work, A Provably Secure Blockchain Protocol. Matthias Fitzi et al.

[7] Energy-recycling Blockchain with Proof-of-Deep-Learning. Changhao Chenli et al.

[8] Proof of Learning (PoLe): Empowering Machine Learning with Consensus Building on Blockchains (Demo). Yixiao Lan et al.

[9] Exploiting Computation Power of Blockchain for Biomedical Image Segmentation. Boyang Li et al.

[10] DLchain: Blockchain with Deep Learning as Proof-of-Useful-Work. Changhao Chenli et al.

[11] An (Almost) Constant-Effort Solution-Verification Proof-of-Work Protocol Based on Merkle Trees. Fabien Coelho.

[12] Proofs of Space. Stefan Dziembowski et al.