Analysis of the Blockchain Protocol in Asynchronous Networks

Wu, chun-chi   Li, jia-hao   Yang, Chih-kai

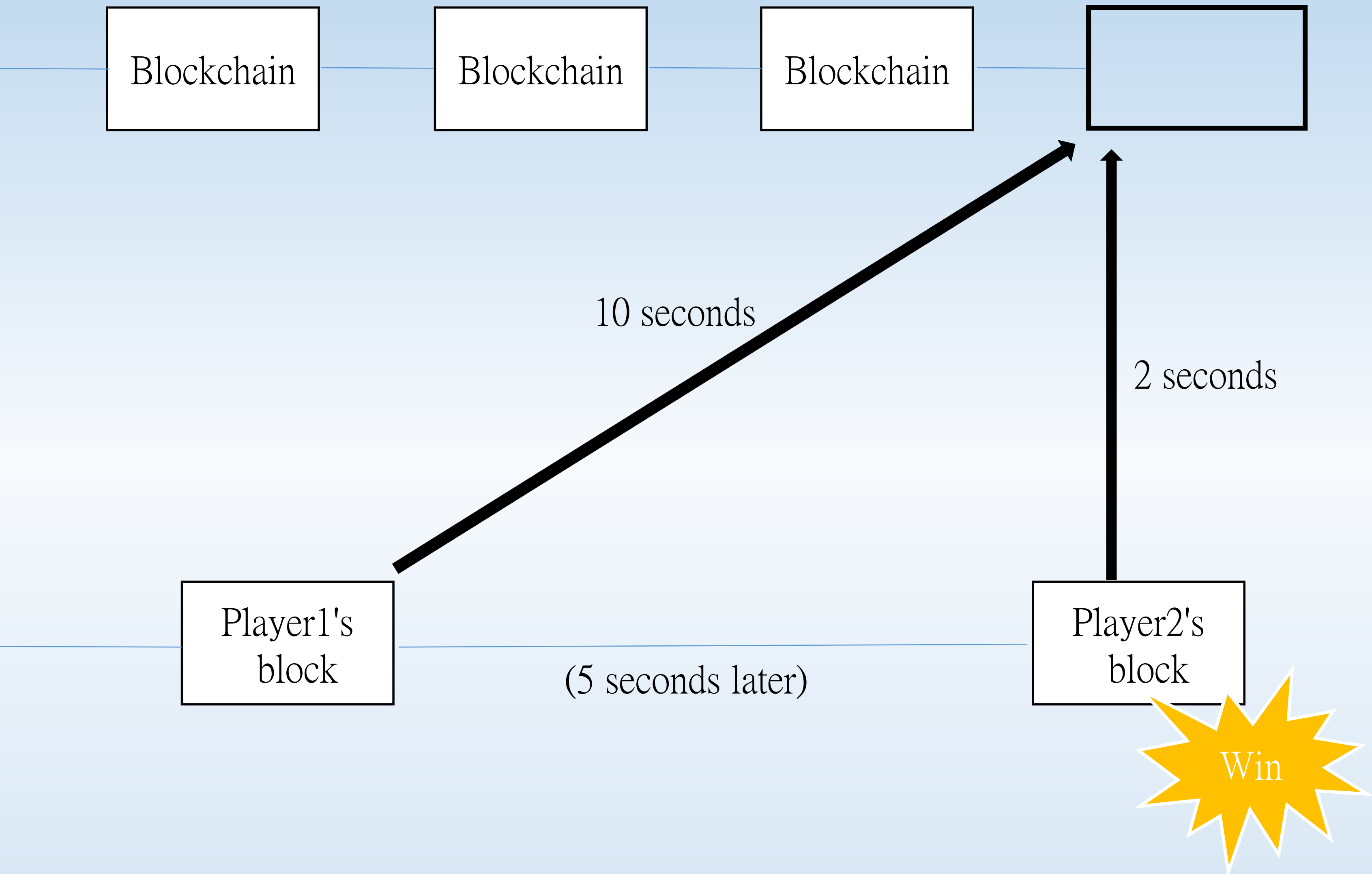
Rafael Pass and Lior Seeman and abhi shelat.
Analysis of the Blockchain Protocol in Asynchronous Networks.
IACR Cryptology ePrint Archive, 2016:454,2016

Outline

- Introduction
- Main Result
- Blockchain Protocol
- $\mathcal{F}_{tree}$ Hybrid Model
- Nakamoto's Model v.s. Hybrid Models
- Proof of the Consistency in Asynchronous Networks
- Conclusion

Introduction

- Motivation: Nakamoto's protocol is consistent in synchronous networks. How about in asynchronous networks?
- Nobody did the analysis in asynchronous networks before.


Blockchain
Blockchain
Blockchain
10 seconds
2 seconds
Player1's block
(5 seconds later)
Player2's block
Win

Introduction

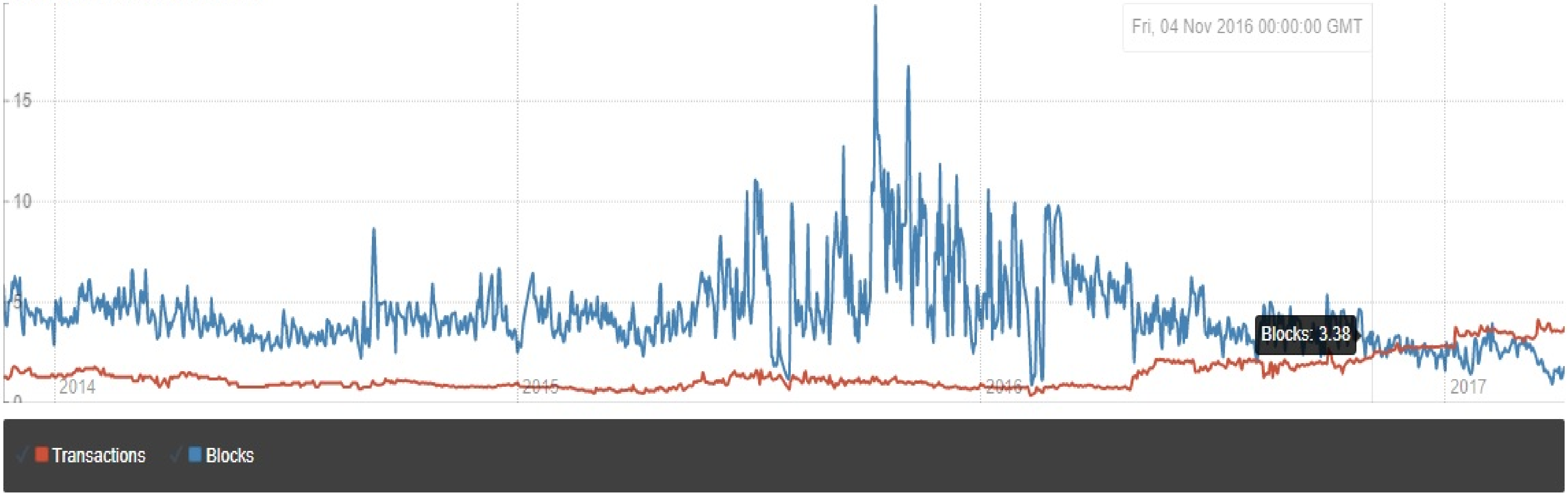
In fact, for a period during the summer of 2012,they computed the average e blocktime to be roughly 10.55m and the “weighted average” $\Delta \sim 11.37s$. [PSS16]

Data Propagation Daily Snapshots

The information exchange in the Bitcoin Network is all but instantaneous. But exactly how fast is information being propagated in the network?

Propagation evolution

The chart below shows the 50th percentile of the *inv*-messages received by peers, i.e., the plot shows the time since a transaction or block enters the network until a majority of nodes has received and processed it.



Take from : http://bitcoinstats.com/network/propagation/

Daily snapshot

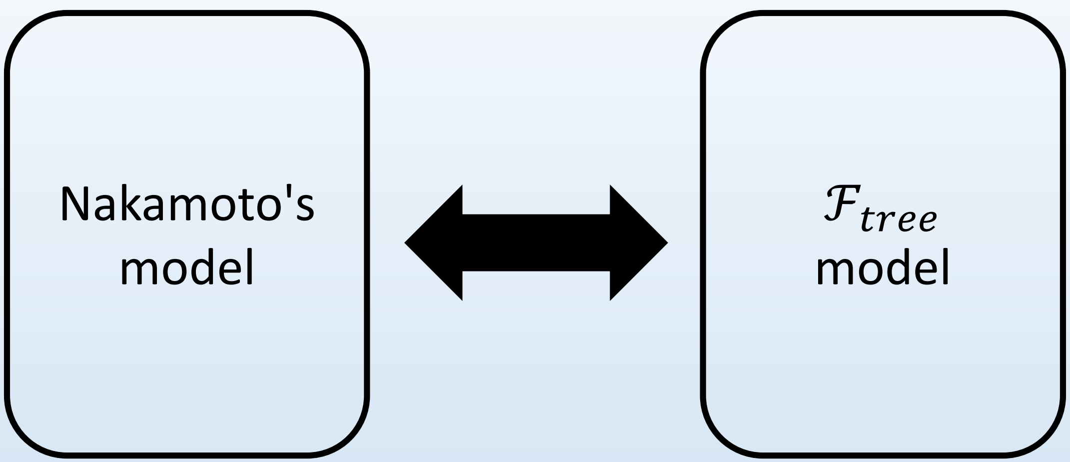
Date (Link)	Block 50th percentile	Block 90th percentile
2017/03/17	2.081 seconds	13.665 seconds
2017/03/18	2.135 seconds	15.905 seconds
2017/03/19	1.886 seconds	14.351 seconds
2017/03/20	1.558 seconds	9.347 seconds
2017/03/21	1.643 seconds	10.472 seconds
2017/03/22	1.517 seconds	9.309 seconds
2017/03/23	1.553 seconds	10.25 seconds
2017/03/24	1.3 seconds	8.249 seconds
2017/03/25	1.268 seconds	7.976 seconds
2017/03/26	0.947 seconds	4.941 seconds
2017/03/27	1.067 seconds	6.152 seconds
2017/03/28	1.621 seconds	10.446 seconds
2017/03/29	1.587 seconds	9.281 seconds
2017/03/30	1.643 seconds	9.436 seconds
2017/03/31	1.481 seconds	7.379 seconds
2017/04/01	1.771 seconds	9.803 seconds
2017/04/02	1.24 seconds	6.439 seconds
2017/04/03	1.282 seconds	6.778 seconds
2017/04/04	1.544 seconds	8.152 seconds
2017/04/05	1.818 seconds	12.828 seconds

Take from : http://bitcoinstats.com/network/propagation/

Introduction

- How to solve the problem?

Set up a model that can represent the protocol in asynchronous networks and did the analysis in the model!



Outline

- Introduction
- Main Result
- Blockchain Protocol
- $\mathcal{F}_{tree}$ Hybrid Model
- Nakamoto's Model v.s. Hybrid Models
- Proof of the Consistency in Asynchronous Networks
- Conclusion

Main Result

- Define an abstract of blockchain protocol and identify security of the protocol.
- Prove that Nakamoto’s protocol satisfies the protocol.
- Prove that the blockchain consensus mechanism satisfying **consistency** in an asynchronous network.

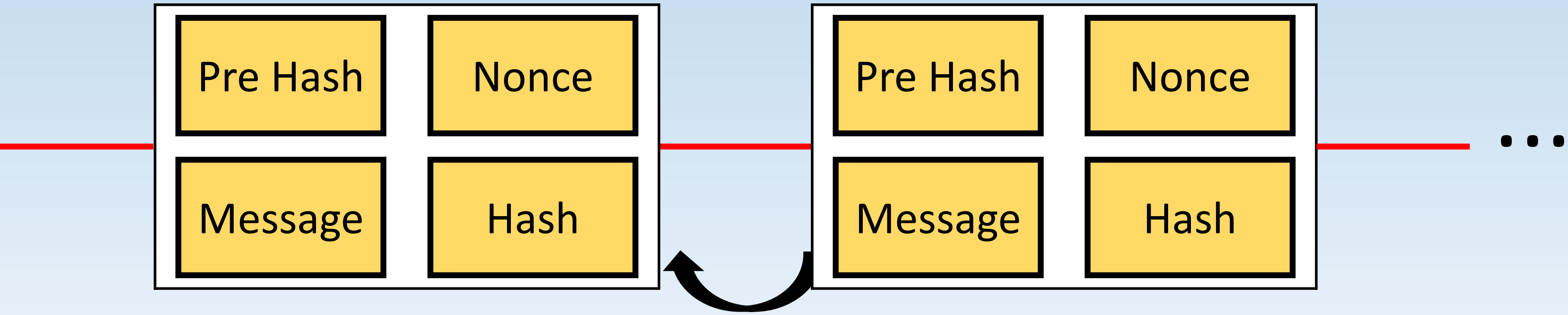
Outline

- Introduction
- Main Result
- Blockchain Protocol
- $\mathcal{F}_{tree}$ Hybrid Model
- Nakamoto's Model v.s. Hybrid Models
- Proof of the Consistency in Asynchronous Networks
- Conclusion

Environment

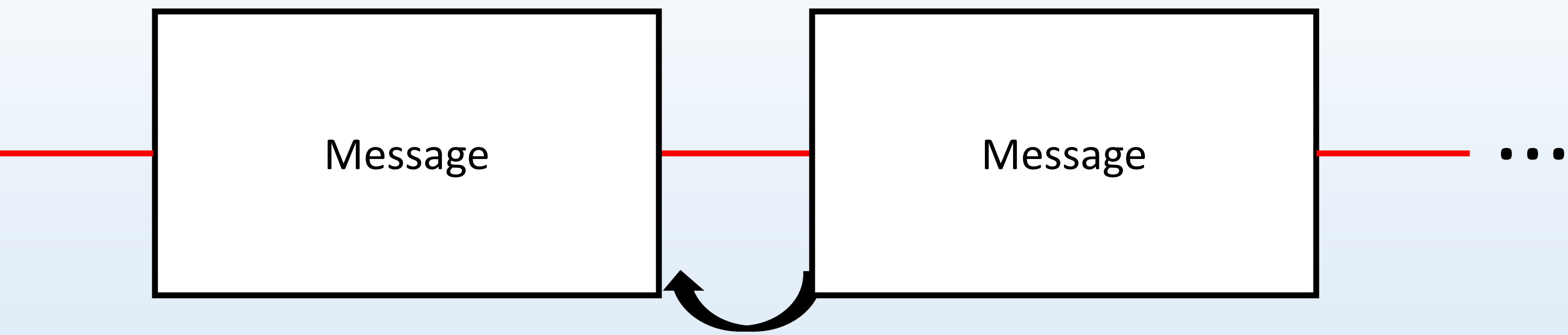
- Control oracle
- Communicate with the adversary
- Corrupt/uncorrupt players

Nakamoto’s blockchain:



If the hash value less than D_P

$\mathcal{F}_{tree}$ blockchain:

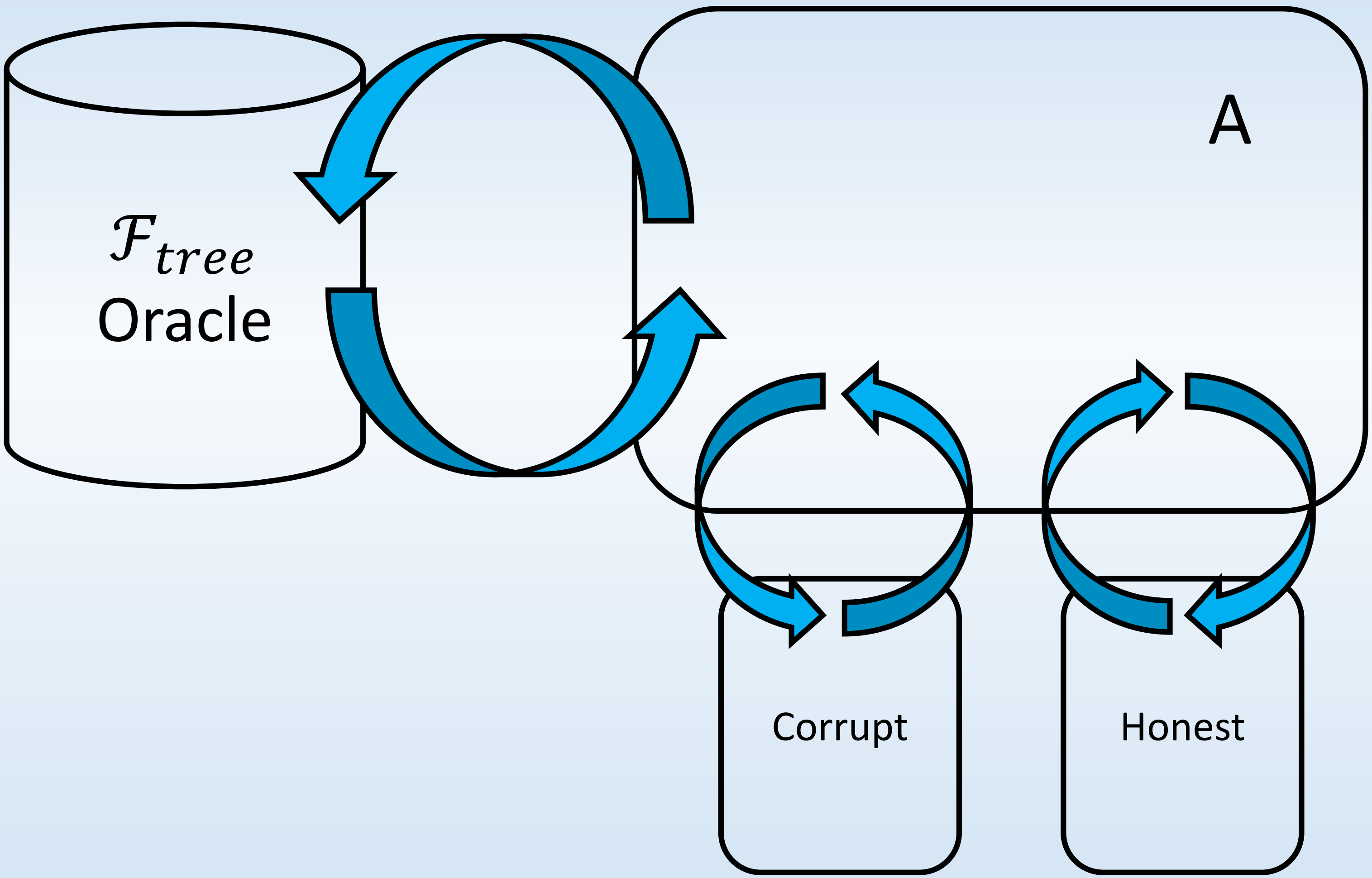


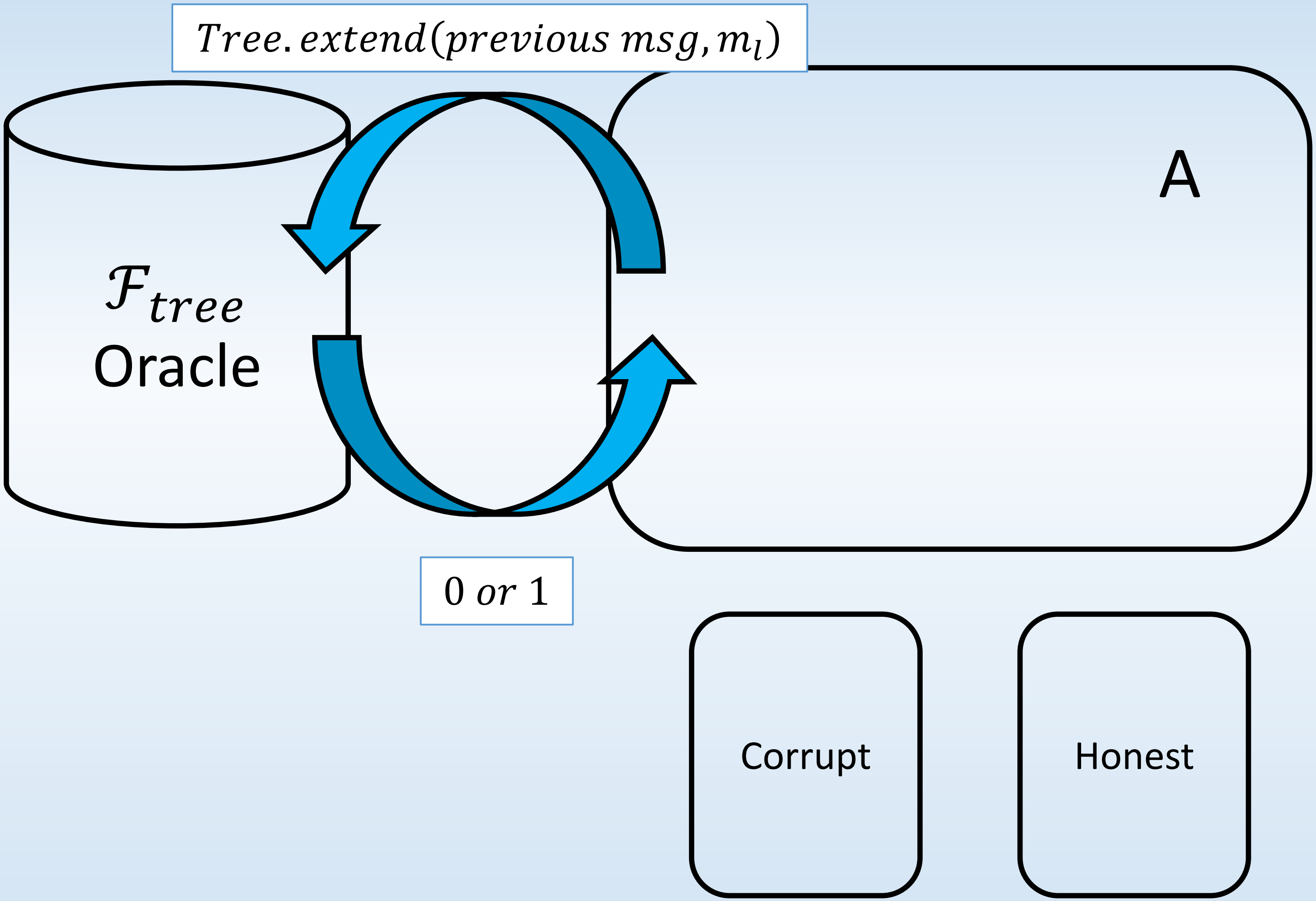
If in probability $p(\kappa) = \frac{D_p}{2^{\kappa}}$

Outline

- Introduction
- Main Result
- Blockchain Protocol
- $\mathcal{F}_{tree}$ Hybrid Model
- Nakamoto's Model v.s. Hybrid Models
- Proof of the Consistency in Asynchronous Networks
- Conclusion

$\mathcal{F}_{tree}$ Hybrid Model

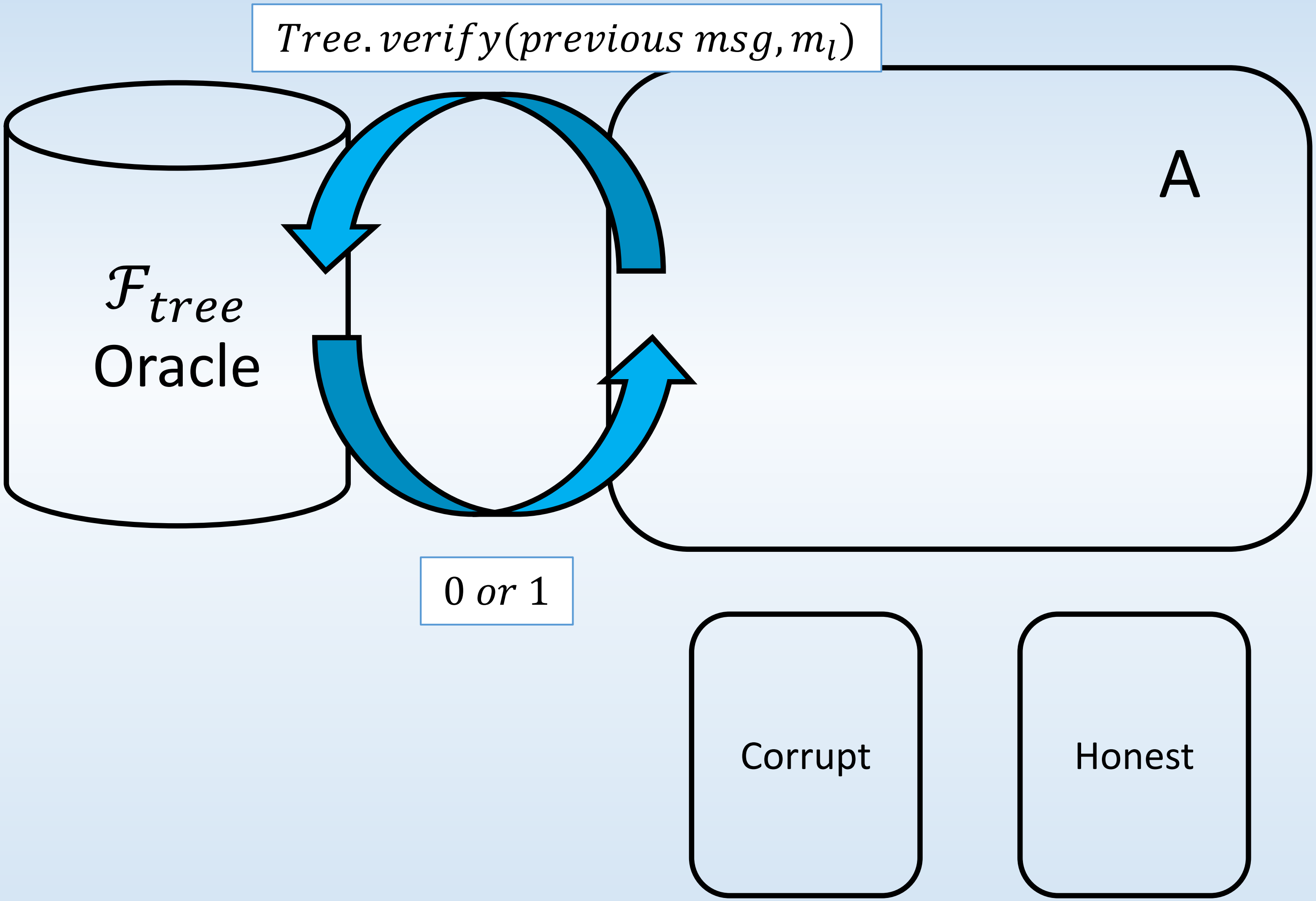



$Tree.extend(previous\ msg, m_l)$
$\mathcal{F}_{tree}$
Oracle
A
$0\ or\ 1$
Corrupt
Honest

$Tree.extend(previous\ msg, m_l)$

return 1 if $(previous\ msg, m_l)$ keeps track
and with probability $p(\kappa)$

return 0 otherwise


$Tree.verify(previous\ msg, m_l)$
$\mathcal{F}_{tree}$
Oracle
A
$0\ or\ 1$
Corrupt
Honest

$Tree.verify(previous\ msg, m_l)$

return 1 if $(\perp, m_1, m_2, \ldots, m_l)$ keeps track

return 0 otherwise

Nakamoto's blockchain in F_{tree}

- In Nakamoto's blockchain each query is picking a nonce and calculate a hash

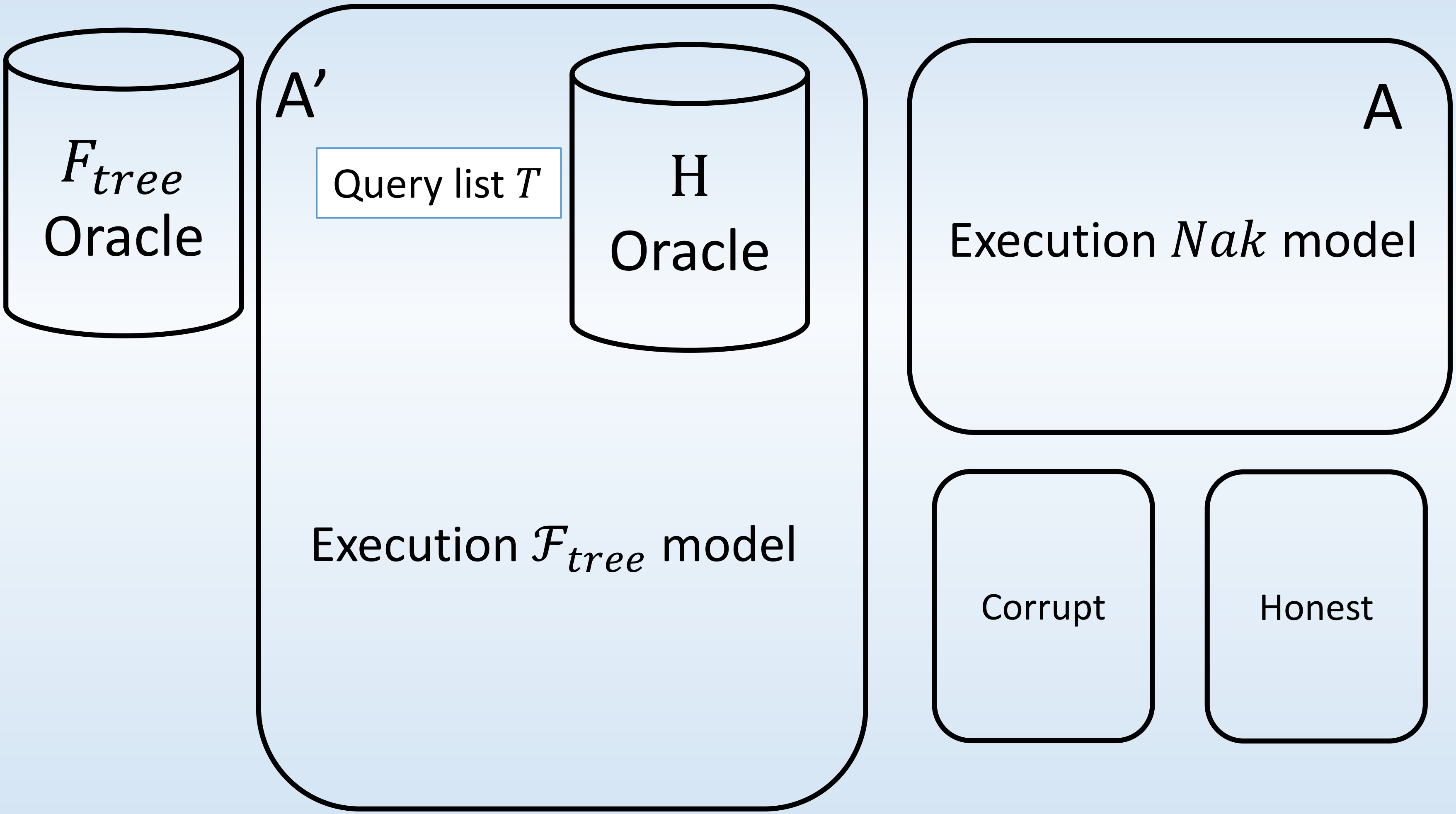
  i.e. each round of input in **Nak** are all different

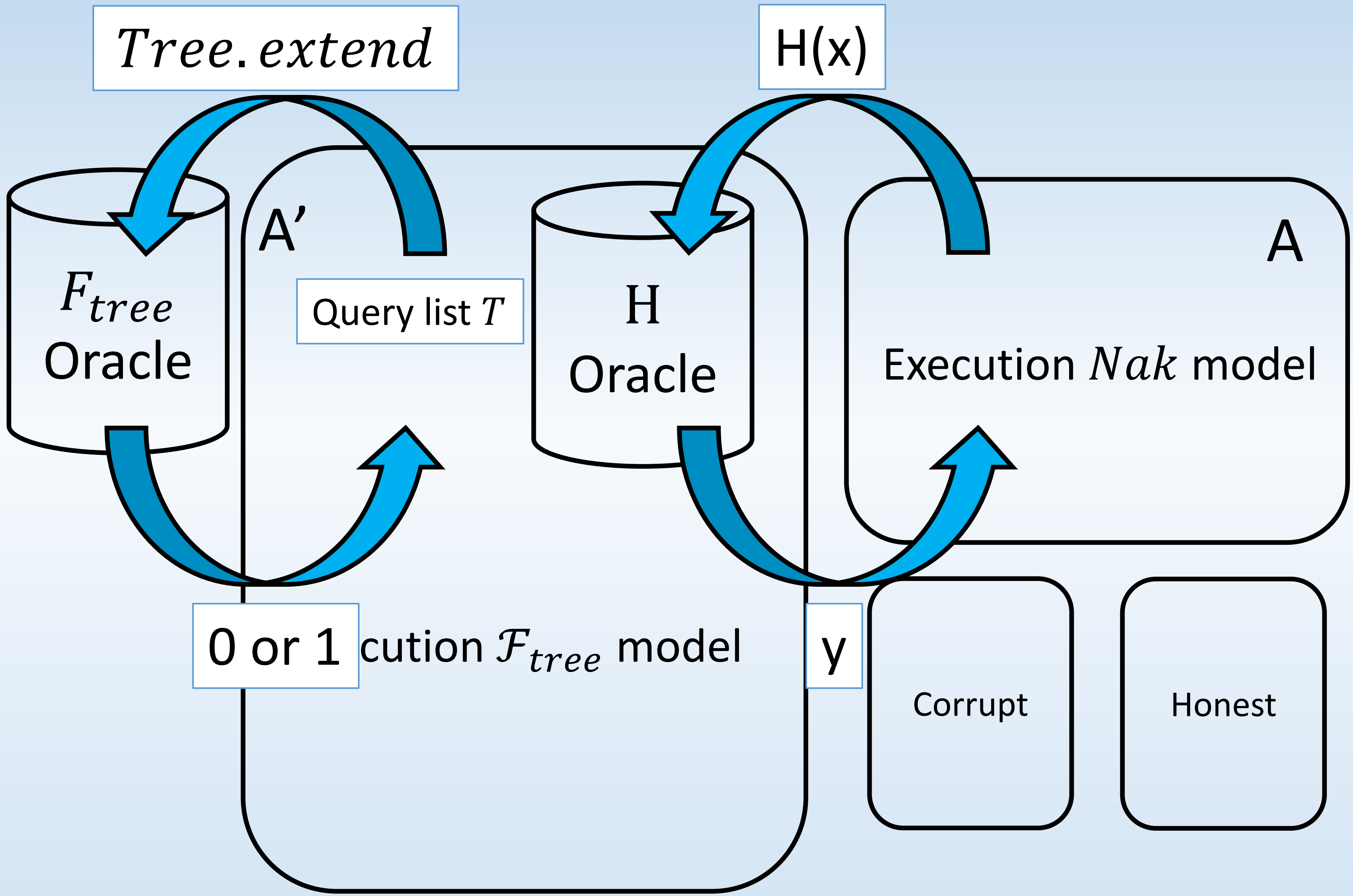
- In F_{tree} no matter what message we query, we mine a block just cause to the probability p

Outline

- Introduction
- Main Result
- Blockchain Protocol
- $\mathcal{F}_{tree}$ Hybrid Model
- Nakamoto's Model v.s. Hybrid Models
- Proof of the Consistency in Asynchronous Networks
- Conclusion

$(\Pi^V_{Nak}, \mathcal{C}_{Nak})$ “*as security as*” $(\Pi^V_{Tree}, \mathcal{C}_{Tree})$





$Tree.extend$
H(x)
F_{tree}
Oracle
A'
Query list T
H
Oracle
A
Execution Nak model
0 or 1
cution $\mathcal{F}_{tree}$ model
y
Corrupt
Honest

$H(x) = y$

- If $(x, y) \in T$ return y
- If x has the form (h_{l-1}, η_l, m_l)

  If $Tree.extend(\overrightarrow{m}, m_l) = 1$

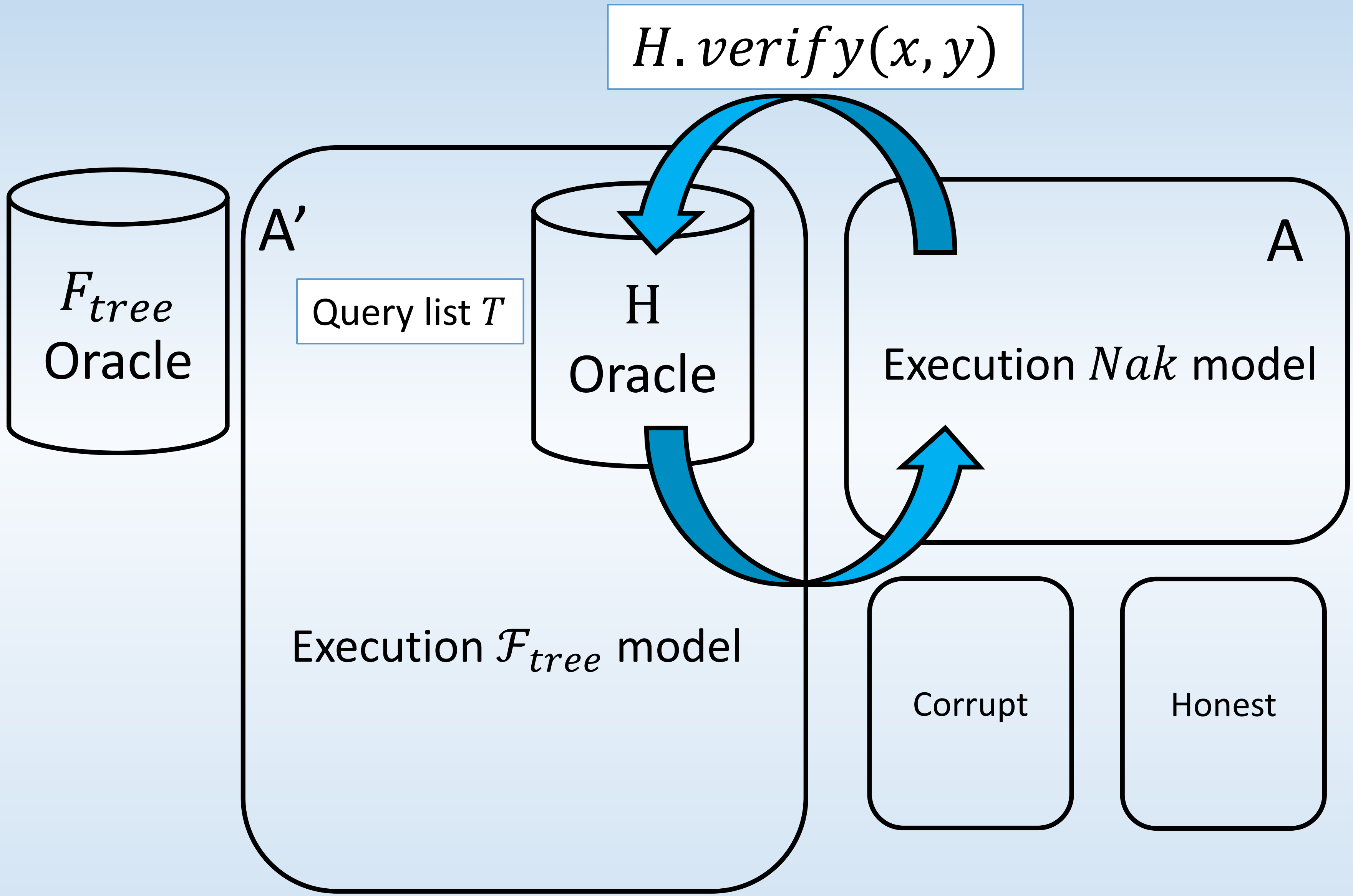
  $y \xleftarrow{\$} \{0,1\}^\kappa$ $with\ y < D_p$

  If $Tree.extend(\overrightarrow{m}, m_l) = 0$

  $y \xleftarrow{\$} \{0,1\}^\kappa$ $with\ y \geq D_p$
- Else $y \xleftarrow{\$} \{0,1\}^\kappa$

Insert (x, y) into T, and then output y

Abort if $Tree.ver(\overrightarrow{m}) \neq 1$ or $Collision$



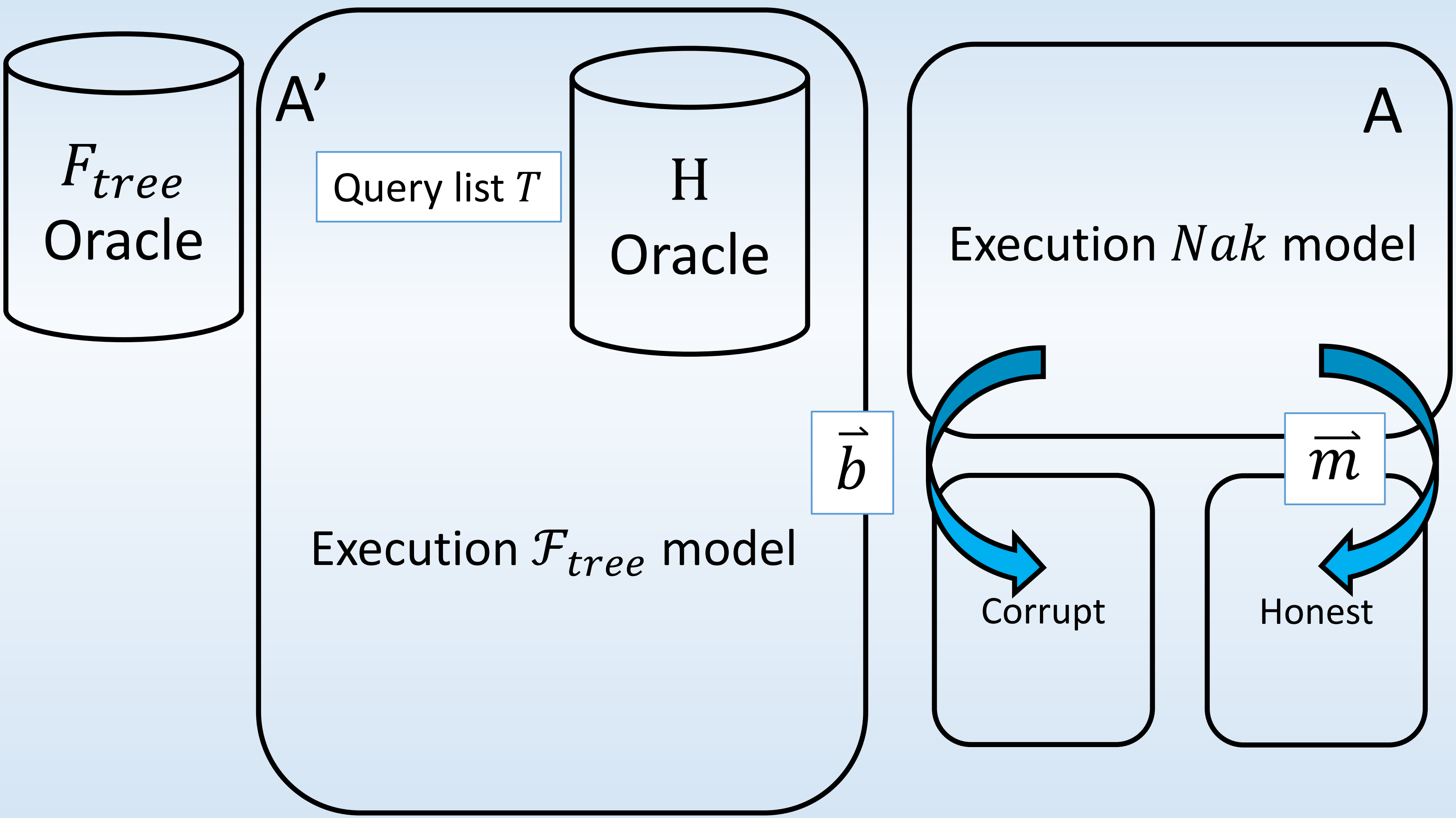
$H.verify(x, y)$
F_{tree}
Oracle
A'
Query list T
H
Oracle
A
Execution Nak model
Execution $\mathcal{F}_{tree}$ model
Corrupt
Honest

$H.verify(x, y)$

Return 1 if $(x, y) \in T$

Return 0 otherwise

A delivers $\vec{b}$

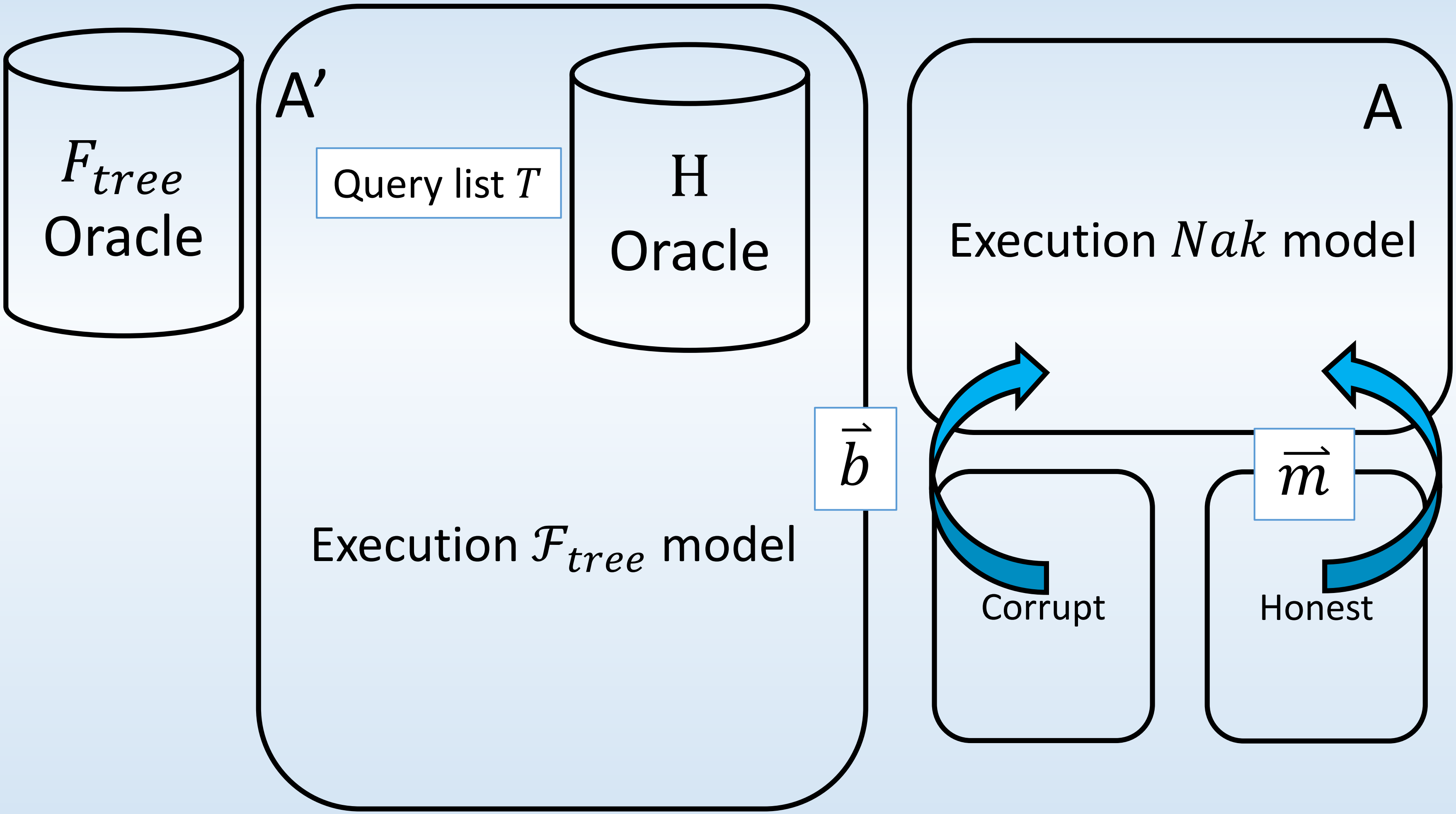


A delivers $\vec{b}$

$\vec{b} = (b_0, b_1, \ldots, b_l)$ *where* $b_i = ((h_{i-1}, \eta_i, m_i), h_i)$

Send $\vec{m} = (previous\ msg, m_l)$ to honest party j

Broadcasts $\overrightarrow{m}$, $\vec{b}$



Honest j broadcasts $\vec{m}$

A replace it to $\vec{b}$

With $b_l = (h_{l-1}, \eta_l, m_l, h_l)$

Where

$$H\big((h_{l-1}, \eta_l, m_l)\big) = h_l$$

and let $Tree.extend = 1$

$\left(\Pi_{Nak}^{V}, \mathcal{C}_{Nak}\right)$ "*as security as*" $\left(\Pi_{Tree}^{V}, \mathcal{C}_{Tree}\right)$

Prove by Hybrid Argument

$H(\cdot)$ randomly output

Discard Fail query

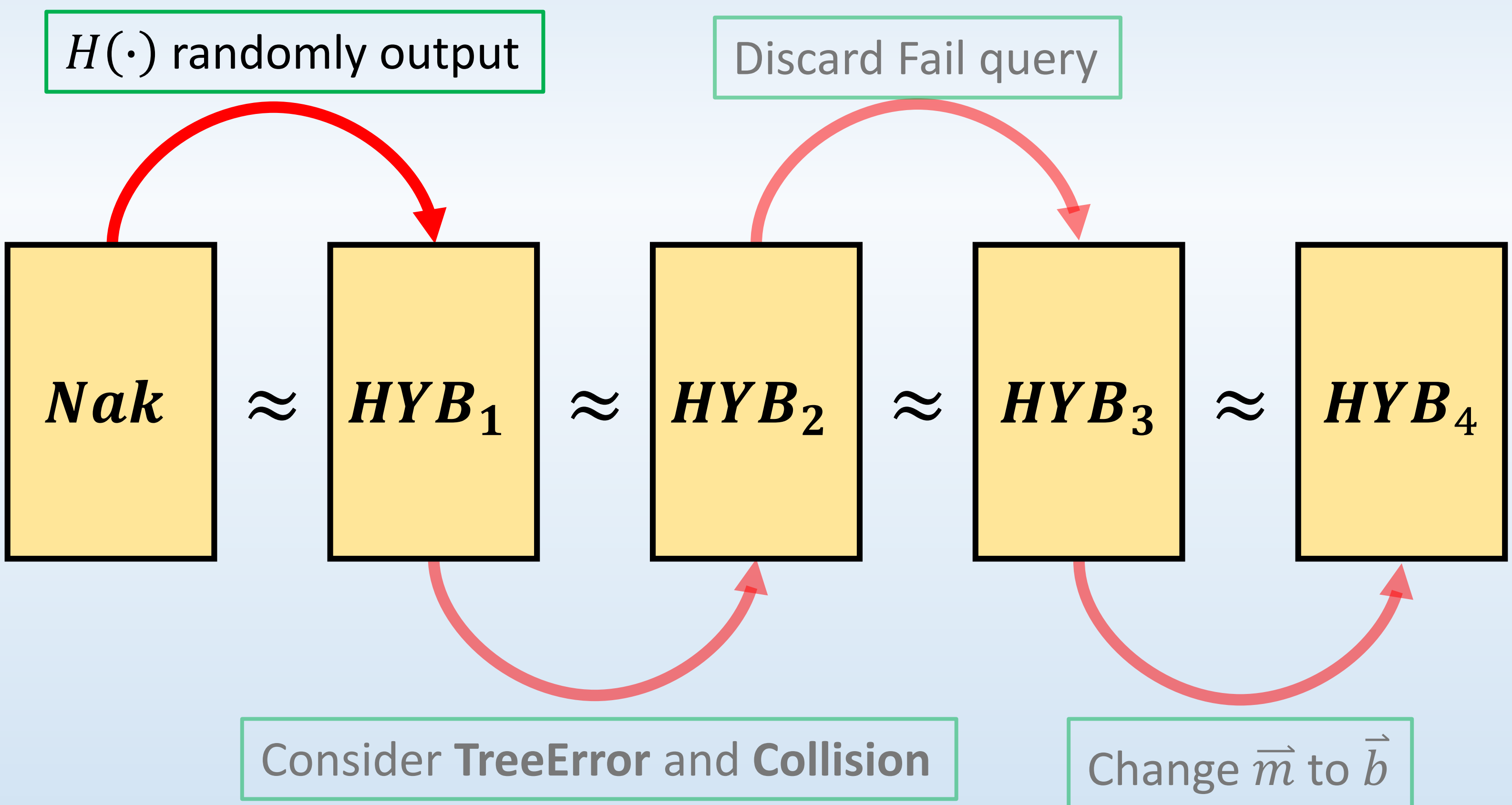
$$\boldsymbol{Nak} \approx \boldsymbol{HYB_1} \approx \boldsymbol{HYB_2} \approx \boldsymbol{HYB_3} \approx \boldsymbol{HYB_4}$$

Consider **TreeError** and **Collision**

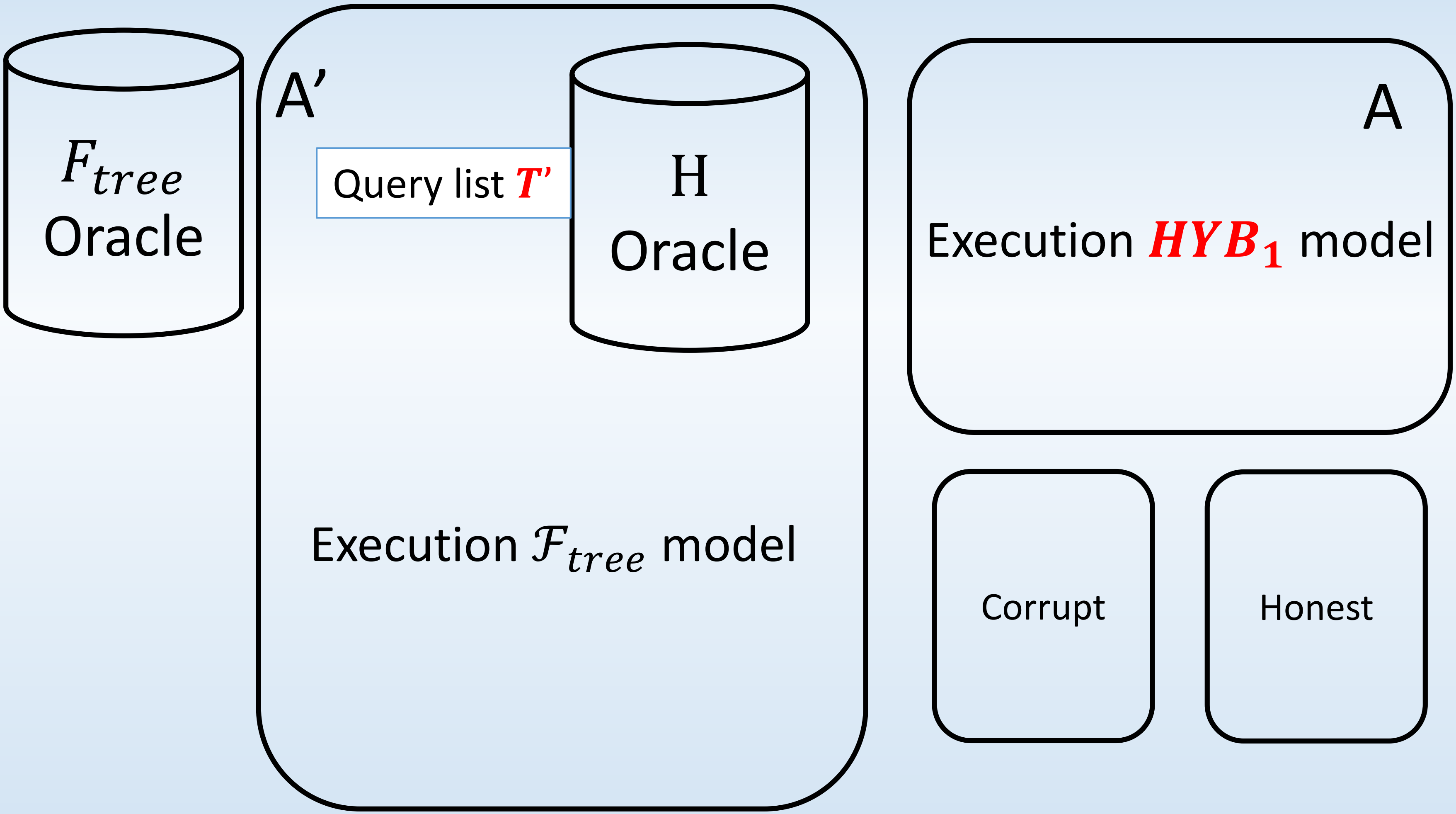
Change $\vec{m}$ to $\vec{b}$

$\left(\Pi^{V}_{Nak}, \mathcal{C}_{Nak}\right)$ "*as security as*" $\left(\Pi^{V}_{Tree}, \mathcal{C}_{Tree}\right)$

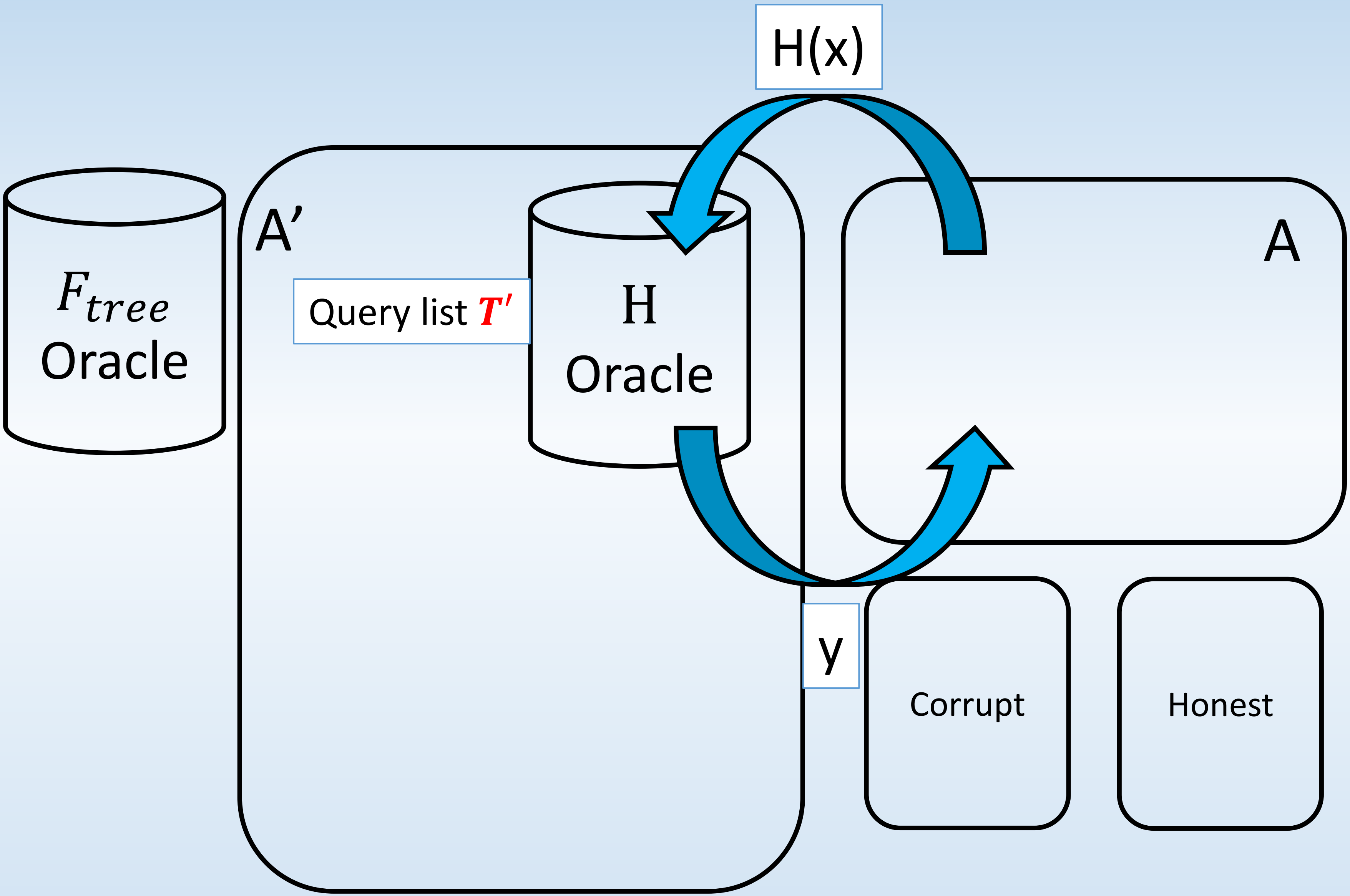
Prove by Hybrid Argument



HYB1



F_{tree}
Oracle
A'
Query list T'
H
Oracle
Execution $\mathcal{F}_{tree}$ model
A
Execution HYB_1 model
Corrupt
Honest


H(x)
A'
F_{tree}
Oracle
Query list T'
H
Oracle
A
y
Corrupt
Honest

$H(x) = y$

- If $(x, y) \in T'$ return y
- If x has the form (h_{l-1}, η_l, m_l)

  If $Tree.extend(\overrightarrow{m}, m_l) = 1$

  $y \overset{\$}{\leftarrow} \{0,1\}^{\kappa}$ $with\ y < D_p$

  If $Tree.extend(\overrightarrow{m}, m_l) = 0$

  $y \overset{\$}{\leftarrow} \{0,1\}^{\kappa}$ $with\ y \geq D_p$
- Else $y \overset{\$}{\leftarrow} \{0,1\}^{\kappa}$

Insert (x, y) into T', and then output y

Abort if $Tree.verify(\overrightarrow{m}) \neq 1$ or $Collision$

$H(x) = y$

Nak Model

- If $(x, y) \in T$ return y
- If x has the form (h_{l-1}, η_l, m_l)

  If $Tree.extend(\overrightarrow{m}, m_l) = 1$

  $y \overset{\$}{\leftarrow} \{0,1\}^\kappa$ $with\ y < D_p$

  If $Tree.extend(\overrightarrow{m}, m_l) = 0$

  $y \overset{\$}{\leftarrow} \{0,1\}^\kappa$ $with\ y \geq D_p$
- Else $y \overset{\$}{\leftarrow} \{0,1\}^\kappa$

HYB_1 Model

- If $(x, y) \in T'$ return y
- Else $y \overset{\$}{\leftarrow} \{0,1\}^\kappa$

$H.verify(x, y)$

Nak Model

Return 1 if $(x, y) \in T$

Return 0 otherwise

HYB_1 Model

If $x \notin T'$

$$y \xleftarrow{\$} \{0,1\}^{\kappa}$$

Return 1 if $(x, y) \in T$

Return 0 otherwise

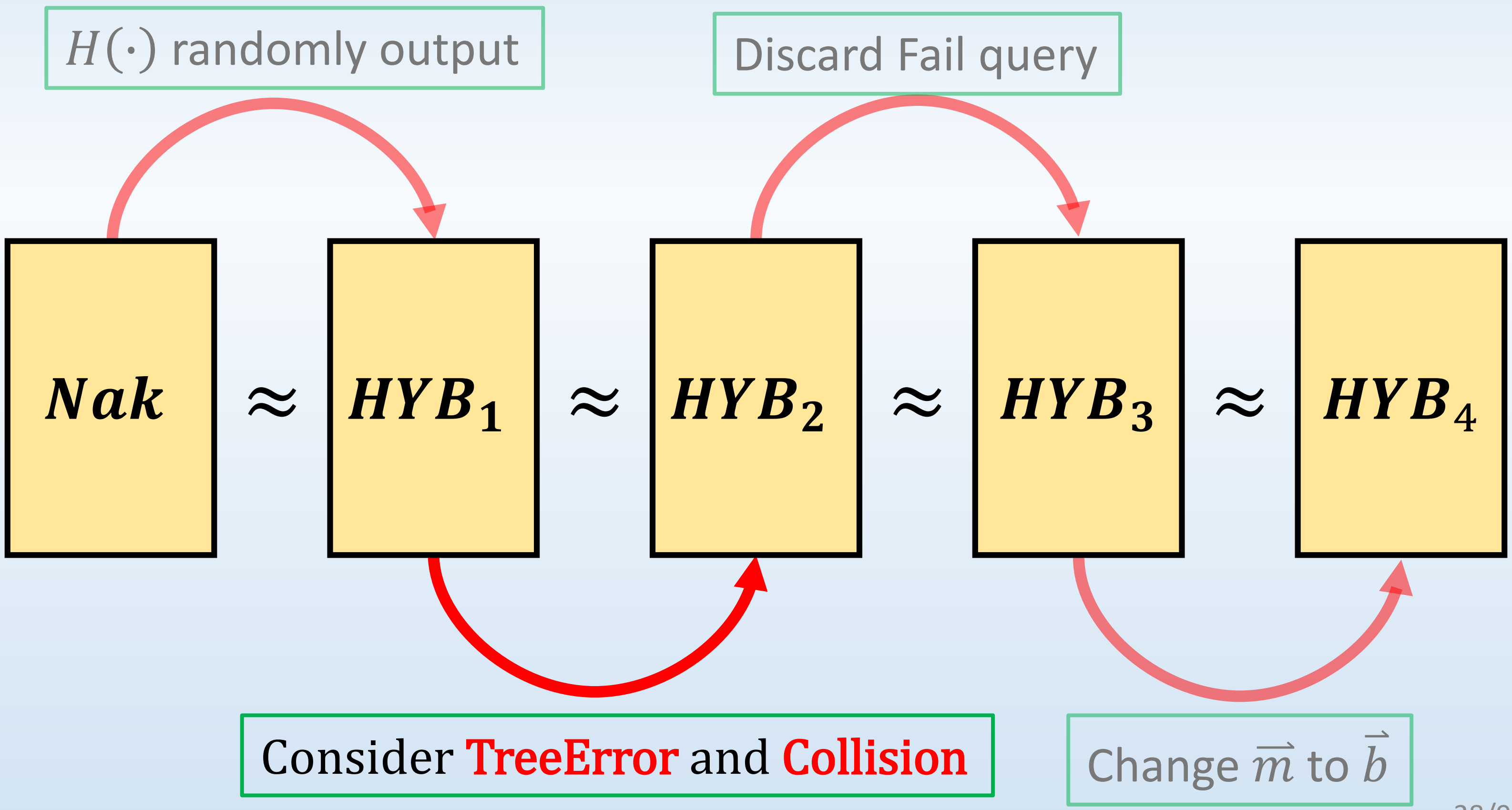
The probability that

$H.verify(x, \cdot)$ return 1 without request $H(x)$

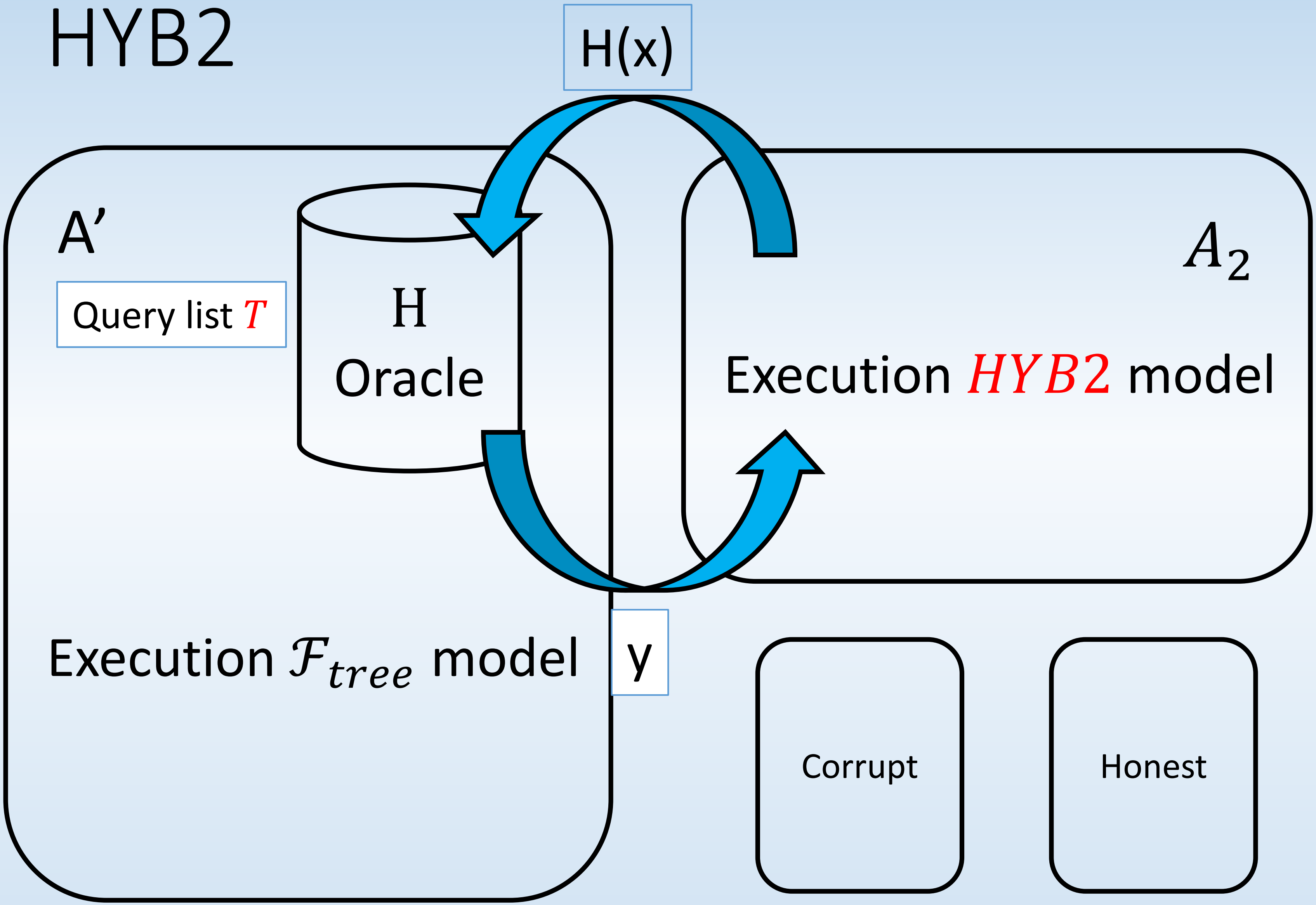
is **negligible**

$\left(\Pi^{V}_{Nak}, \mathcal{C}_{Nak}\right)$ "*as security as*" $\left(\Pi^{V}_{Tree}, \mathcal{C}_{Tree}\right)$

Prove by Hybrid Argument



HYB2


H(x)
A'
Query list T
H
Oracle
A_2
Execution $HYB2$ model
Execution $\mathcal{F}_{tree}$ model
y
Corrupt
Honest

$H(x) = y$

- If $(x, y) \in T$ return y
- If x has the form (h_{l-1}, η_l, m_l)

  If $Tree.extend(\overrightarrow{m}, m_l) = 1$

  $y \xleftarrow{\$} \{0,1\}^{\kappa}$ $with\ y < D_p$

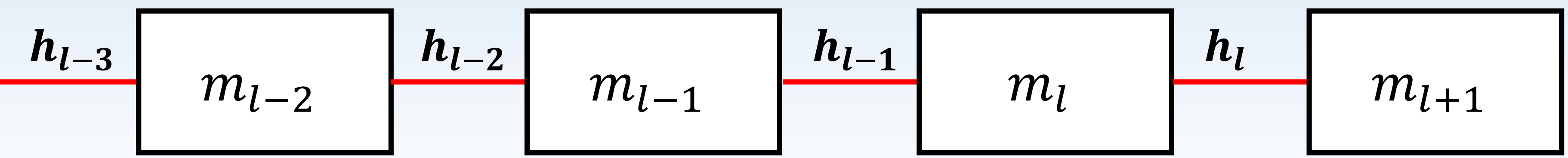
  If $Tree.extend(\overrightarrow{m}, m_l) = 0$

  $y \xleftarrow{\$} \{0,1\}^{\kappa}$ $with\ y \geq D_p$
- Else $y \xleftarrow{\$} \{0,1\}^{\kappa}$

Insert (x, y) into T, and then output y

Abort if $Tree.verify(\overrightarrow{m}) \neq 1$ or $Collision$

Case of $Tree.verify(\overrightarrow{m}) \neq 1$



$Tree.verify(\overrightarrow{m}) \neq 1$

When $\exists i < l$ s.t. $h_i = h_l$

$H(x) = y$

HYB_1 Model

- If $(x, y) \in T'$ return y
- Else $y \xleftarrow{\$} \{0,1\}^{\kappa}$

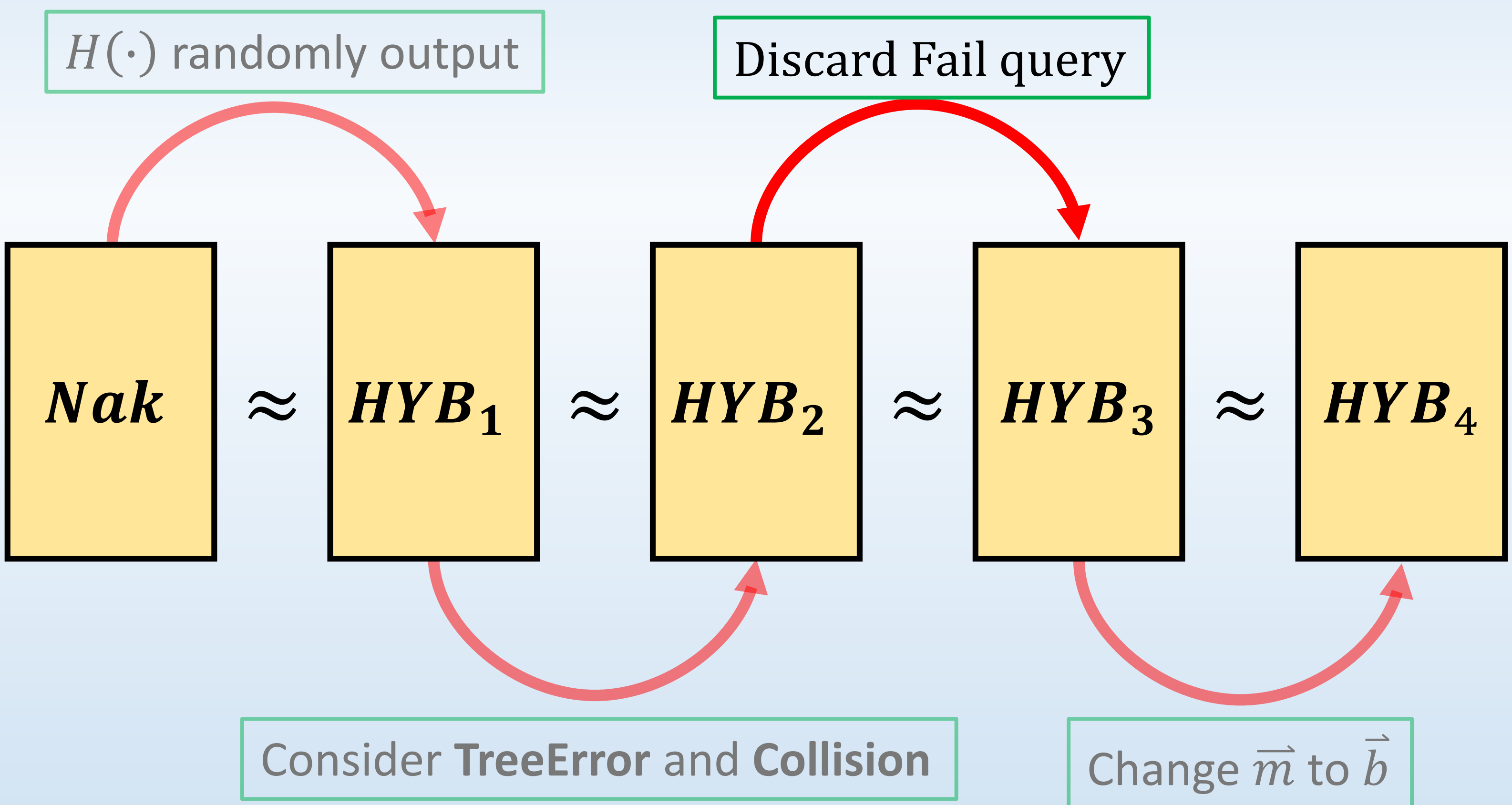
HYB_2 Model

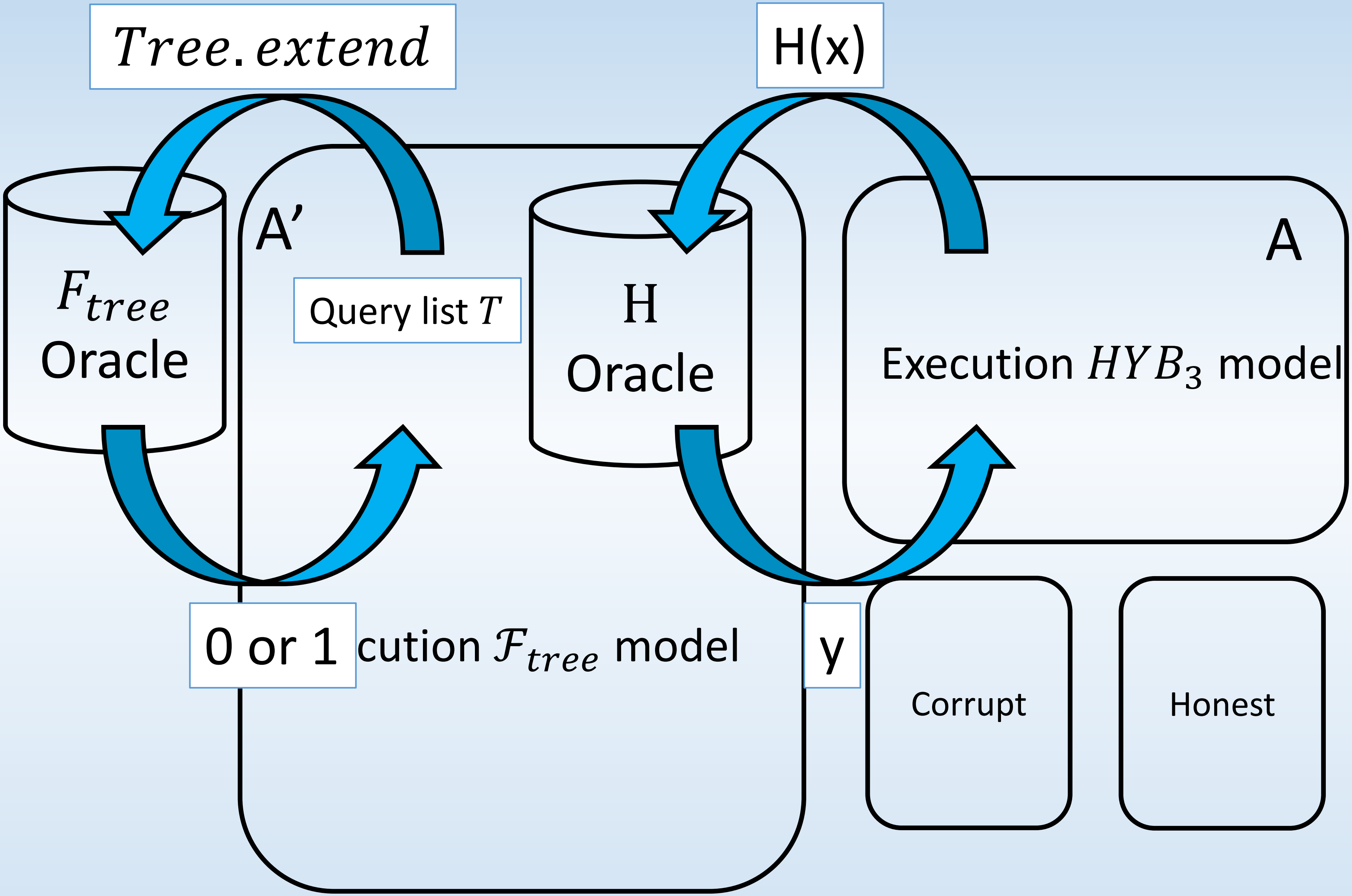
- If $(x, y) \in T'$ return y
- Else $y \xleftarrow{\$} \{0,1\}^{\kappa}$
- Abort if **Tree error** or **Collision**

The probability that events happened is negligible

$(\Pi^{V}_{Nak}, \mathcal{C}_{Nak})$ "*as security as*" $(\Pi^{V}_{Tree}, \mathcal{C}_{Tree})$

Prove by Hybrid Argument




$Tree.extend$
H(x)
F_{tree}
Oracle
A'
Query list T
H
Oracle
A
Execution HYB_3 model
0 or 1
cution $\mathcal{F}_{tree}$ model
y
Corrupt
Honest

$H(x) = y$

- If $(x, y) \in T$ return y
- Else $y \xleftarrow{\$} \{0,1\}^{\kappa}$

Output y, and insert (x, y) into T

Insert (x, y) into T ,if one of the event Happened:

- x is request by corrupt player
- $Tree.extend(\overrightarrow{m}, m_l) = 1$

Same x will never be queried again

Abort if $Tree.verify(\overrightarrow{m}) \neq 1$ or $Collision$

$H(x) = y$

HYB_2 Model

- If $(x, y) \in T'$ return y
- Else $y \xleftarrow{\$} \{0,1\}^{\kappa}$

- Abort if **Tree error** or **Collision**

HYB_3 Model

- If $(x, y) \in T$ return y
- Else $y \xleftarrow{\$} \{0,1\}^{\kappa}$

Insert (x, y) into T ,if

x is not a fail query by honest

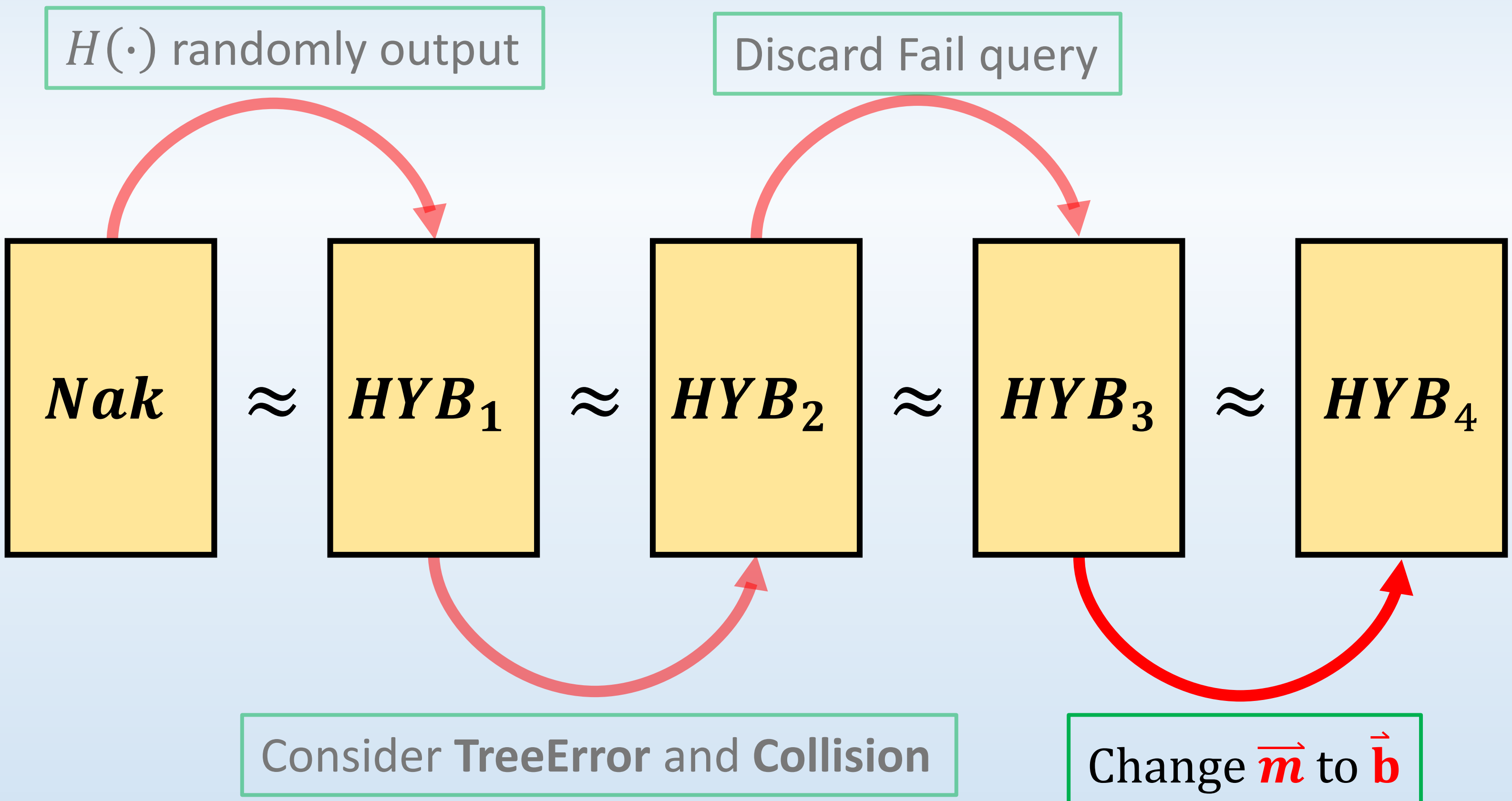
Same x won't be queried again

- Abort if **Tree error** or **Collision**

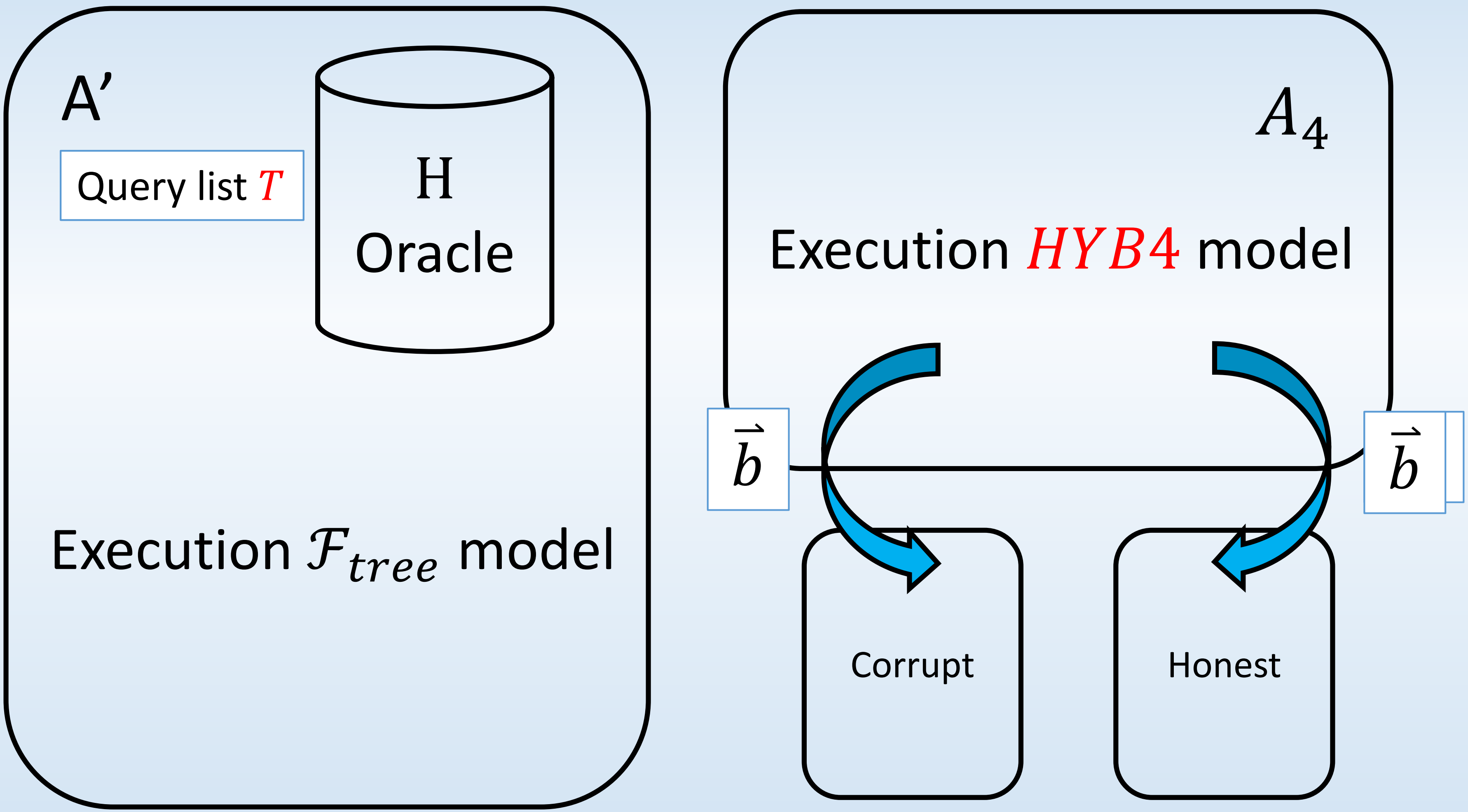
The probability that events happened is negligible

$(\Pi^{V}_{Nak}, \mathcal{C}_{Nak})$ "*as security as*" $(\Pi^{V}_{Tree}, \mathcal{C}_{Tree})$

Prove by Hybrid Argument



HYB4


A'
Query list T
H
Oracle
Execution $\mathcal{F}_{tree}$ model
A_4
Execution $HYB4$ model
$\vec{b}$
$\vec{b}$
Corrupt
Honest

Outline

- Introduction
- Main Result
- Blockchain Protocol
- $\mathcal{F}_{tree}$ Hybrid Model
- Nakamoto's Model v.s. Hybrid Models
- Proof of the Consistency in Asynchronous Networks
  - Chain Growth Property
  - Chain Quality Property
  - Consistency
- Conclusion

Chain Growth Property

- The rate of chain growth

$$g_{\delta}^{p}(n, \rho, \Delta) = (1 - \delta)\gamma$$

α means the probability honest parties mine a block in a round

$\gamma = \frac{\alpha}{1+\alpha\Delta}$ is the probability that honest party mine a chain

with Δ round delay

Chain growth lower bound in HYB_r

We prove it by **Chernoff Bound**

Chernoff Bound:

Let $X_1, \dots, X_n$ be independent Bernoulli random variable ,and $X = \Sigma X_i$, $\mathrm{E}[x] = \mu$

Then, for any $\delta \in (0,1]$ statisfies:

$$\Pr[X > (1 + \delta)\mu] < e^{-\Omega(\delta^2 \mu)}$$
$$\Pr[X < (1 - \delta)\mu] < e^{-\Omega(\delta^2 \mu)}$$

Chain growth lower bound in HYB_r

Define a Bernoulli random variable W

$$W = \begin{cases} 1\text{ , if any honset party mine a block} \\ 0\text{ , if no honset party mine a block} \end{cases}$$

Clearly, $E[W] = \alpha = 1 - (1-p)^{(1-\rho)n}$

α means the probability honest parties mine a block in a round

ρ is the fraction of corrupt party

n is the number of total parties

Chain growth lower bound in HYB_r

Consider in $\boldsymbol{t}$ rounds, the chain grow **less then** $\boldsymbol{c}$

That is, honest parties **"freeze" at most** $\boldsymbol{c\Delta}$ **rounds**

Equivalently, they **compute at least** $\boldsymbol{t - c\Delta}$ **rounds**

That means:

$$\mathrm{E}[\text{Chain–growth in } t \text{ rounds}] \geq E[W^{t-c\Delta}]$$

Chain growth lower bound in HYB_r

Consider when $\mathrm{E}\left[W^{t-c\Delta}\right] = c$

Then, $\mathrm{E}\left[W^{t-c\Delta}\right] = \alpha(t - c\Delta) = c$

$$c = \frac{\alpha t}{1+\alpha\Delta} := \gamma t \quad \text{where } \gamma = \frac{\alpha}{1+\alpha\Delta}$$

Where γ is the probability that honest party mine a chain

with Δ round delay

By Chernoff bound,

we have for any $\delta \in (0,1]$

$$\Pr\left\{W^{t-c\Delta} < (1-\delta)\gamma t\right\} < e^{-\Omega\left(\delta^2 \gamma t\right)}$$

Chain growth lower bound in F_{Tree}

Since we know:

- $\Pr\{len^{r+t}(HYB_r) < len^{r}(HYB_r) + \boldsymbol{(1-\delta)\gamma t}\} < e^{-\Omega(\delta^2 \gamma t)}$
- max chain length in $F_{tree} \geq$ max chain length in HYB_r

We have:

$$\Pr\{len^{r+t}(F_{\mathrm{Tree}}) < len^{r}(F_{\mathrm{Tree}}) + \boldsymbol{(1-\delta)\gamma t}\} < e^{-\Omega(\delta^2 \gamma t)}$$

Chain growth Property in F_{Tree}

Define $\boldsymbol{chain_i^r}$ as the chain that honest party i seems at r

Since,

$$\min_{i,j}\left(\left|chain_j^{r+t}\right| - |chain_i^r|\right) = \min_j\left|chain_j^{r+t}\right| - \max_i|chain_i^r|$$

Also, we know:

$$\min_j\left|chain_j^{r+t}\right| \geq \max_j\left|chain_j^{r+t-\Delta}\right|$$

Combine two equation,

$$\min_{i,j}\left(\left|chain_j^{r+t}\right| - |chain_i^r|\right) \geq \max_j\left|chain_j^{r+t-\Delta}\right| - \max_i|chain_i^r|$$

$$= len^{r+t-\Delta}(F_{\text{Tree}}) - len^r(F_{\text{Tree}})$$

Chain growth Property in F_{Tree}

Since,

- $\min_{i,j}\left(\left|chain_j^{r+t}\right| - |chain_i^r|\right) = len^{r+t-\Delta}(F_{\text{Tree}}) - len^r(F_{\text{Tree}})$
- $\Pr\left\{len^{r+t-\Delta}(F_{\text{Tree}}) < len^r(F_{\text{Tree}}) + (1-\delta')\gamma t\right\} < e^{-\Omega\left(\left(\delta'\right)^2\gamma(t-\Delta)\right)}$

Since $\gamma\Delta = \frac{\Delta\alpha}{1+\Delta\alpha} < 1$

If we pick sufficient small δ', there exists $\delta \in (0,1]$

$s.t.\ \ \Pr\left\{len^{r+t-\Delta}(F_{\text{Tree}}) < len^r(F_{\text{Tree}}) + (1-\delta)\gamma t\right\} < e^{-\Omega(\delta^2\gamma t)}$

Chain growth Property in F_{Tree}

Since,

- $\min_{i,j}\left(\left|chain_j^{r+t}\right| - |chain_i^r|\right) = len^{r+t-\Delta}(F_{\text{Tree}}) - len^r(F_{\text{Tree}})$
- $\Pr\left\{len^{r+t-\Delta}(F_{\text{Tree}}) < len^r(F_{\text{Tree}}) + (1-\delta)\gamma t\right\} < e^{-\Omega\left(\delta^2\gamma t\right)}$

Combine two equation:

$$\Pr\left\{\left|chain_j^{r+t}\right| - |chain_i^r| < (1-\delta)\gamma t\right\} < e^{-\Omega\left(\delta^2\gamma t\right)}$$

Chain growth Property in F_{Tree}

Since

$$\Pr\left\{\left|chain_j^{r+t}\right| - |chain_i^r| < (1-\delta)\gamma t\right\} < e^{-\Omega(\delta^2\gamma t)}$$

Therefore, we have the chain growth property:

For any $\delta \in (0,1]$,

We have the chain growth rate $\boldsymbol{(1-\delta)\gamma}$

Outline

- Introduction
- Main Result
- Blockchain Protocol
- $\mathcal{F}_{tree}$ Hybrid Model
- Nakamoto's Model v.s. Hybrid Models
- Proof of the Consistency in Asynchronous Networks
  - Chain Growth Property
  - Chain Quality Property
  - Consistency
- Conclusion

Chain quality property

- The fraction of the block contributed by honest players

$$\frac{\#\text{ blocks by honest}}{\#\text{ all blocks}} = 1 - \frac{\#\text{ blocks by Adversary}}{\#\text{ all blocks}}$$

Chain quality property

- Calculate the maximum value of fraction of the block contributed by Adversary
- So we consider the condition that Adversary mines blocks in a row

Chain quality property

Consider a **subchain**

$$-b_j - b_{j+1} - \cdots - b_{j+T-1} - b_{j+T} -$$

where b_{j-1} mined at rounds r'

and b_{j+T+1} mined at rounds $r' + t$

were created by honest party

The upper bound on blocks

Since ρ is the fraction of corrupt parties

By Chernoff bound, in t rounds:

For any $\delta' \in (0,1)$

$$\Pr\{M_{\mathcal{A}}^{t'} > (1+\delta')\beta t'\} < e^{-\Omega\left(\delta'^2 \beta t'\right)}$$

Where $M_{\mathcal{A}}^{t}$means

the maximum block mined by Adversary in t rounds

β means the probability that $\boldsymbol{\mathcal{A}}$ mine a block in a round

The upper bound on blocks

The upper bound of $M_{\mathcal{A}}^{t'}$ is:

$$(1+\delta')\beta t'$$

Also, by Chain growth property:

$$T \geq (1-\delta)\gamma t'$$

We have:

$$(1+\delta')\beta t' \leq \frac{(1+\delta')}{(1-\delta)}\frac{\beta}{\gamma}T \leq (1+\delta^*)\frac{\beta}{\gamma}T$$

Proof of chain quality

$$\frac{\text{\# blocks by honest}}{\text{\# all blocks}} = 1 - \frac{\text{\# blocks by Adversary}}{\text{\# all blocks}}$$

$$= 1 - \frac{(1+\delta^*)\frac{\beta}{\gamma}T}{\mathrm{T}}$$

$$= \ 1 - (1 + \delta^*)\,\frac{\beta}{\gamma}$$

Where $\gamma = \frac{\alpha}{1+\Delta\alpha}$

Outline

- Introduction
- Main Result
- Blockchain Protocol
- $\mathcal{F}_{tree}$ Hybrid Model
- Nakamoto's Model v.s. Hybrid Models
- Proof of the Consistency in Asynchronous Networks
    - Chain Growth Property
    - Chain Quality Property
    - Consistency
- Conclusion

Consistency property

- The blockchain seen by honest players in different round should be identical except the last specific blocks

Proof ideas

- What actions may break consistency ?

Selfish mining(long block withholding)

Adversary mines a chain as long as the longest chain accepted by honest player

- Prove that the chain seen by honest players will not diverge under these conditions

Proof of no withholding

- Theorem:

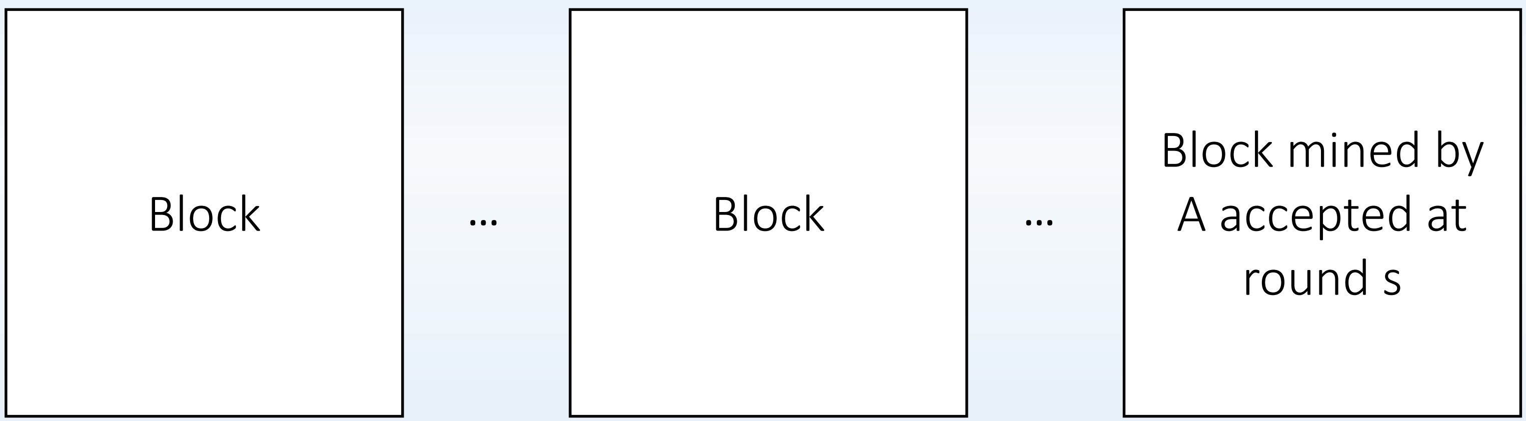
when $\gamma \geq (1+\delta)\beta$, with δ, ω are constants $\in (0,1)$:

The probability that block withholding time$\geq \omega t$ and the block can still be accepted is negligible

- We prove it by contradiction

Proof of no withholding

- Assume a condition: k blocks are mined by Adversary from round r to round s ($s - r \geq \omega t$)



- If the probability is not negligible…

Proof of no withholding

- By lower bound of chain growth:

$$k \geq (1 - \delta')\gamma\omega t$$

By upper bound of adversarial blocks:

$$k \leq (1 + \delta'')\beta\omega t$$

Proof of no withholding

- By choosing proper δ', δ'', $\gamma \le \frac{1+\delta''}{1-\delta'}\beta < (1+\delta)\beta$
- Theorem:

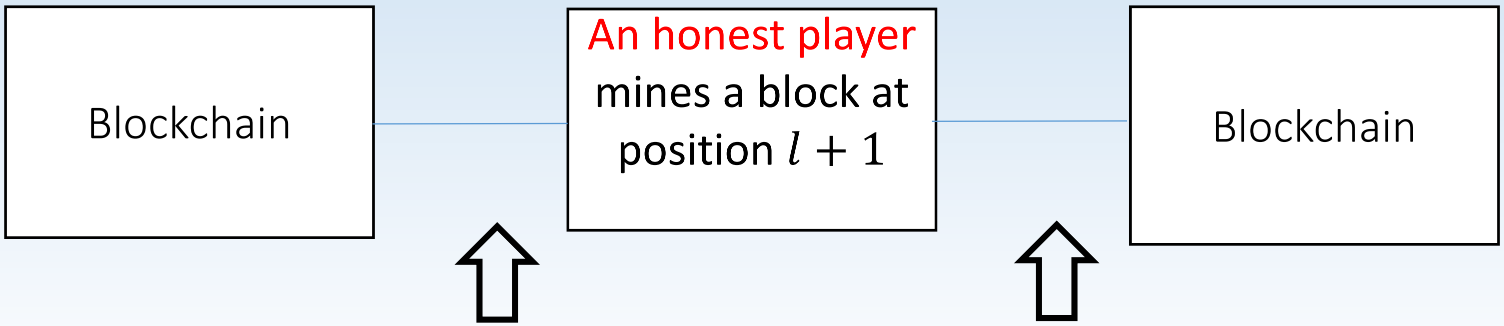
The probability that block withholding time$\geq \omega t$ and the block can still be accepted is negligible

for $\gamma \geq (1+\delta)\beta$, with δ, ω are constants $\in (0,1)$

- Contradiction!!

Proof idea of no divergence

We try to prove this by induction



no honest player mines a block in Δ rounds

no honest player mines a block in Δ rounds

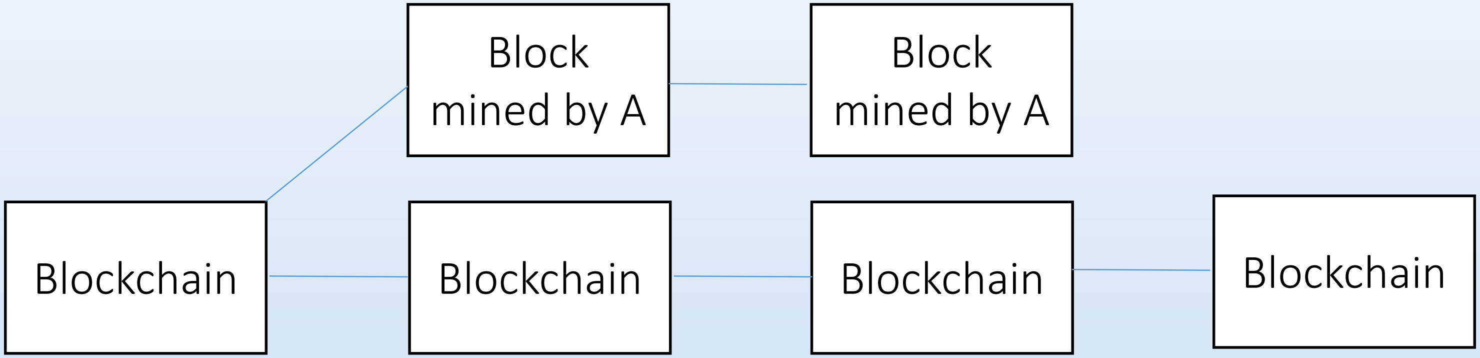
All will agree the new block after Δ rounds, so no honest players will try to mine a block at position $l+1$

If the pattern occurs, that means the chain will converge if $\boldsymbol{\mathcal{A}}$ doesn't mine a $(l+1)$ length chain

Proof idea of no divergence

- If Adversary mines a chain of length $l + 1$, the chain will be divergent
- There exists a unique block on each position under such patterns unless:

Adversary mines a chain of length $l + 1$ in each pattern



Proof of no divergence

- Calculate the block mined by honest player in t rounds
- Calculate the number of pattern in t rounds.
- Adversary should mine a block in each pattern to make the chain divergent, but the probability it occurs is negligible

Proof of no divergence

- Under the circumstance that

$$\alpha\big(1 - 2\alpha(\Delta + 1)\big) \geq (1 + \delta)\beta$$

- By the Chernoff bound,

blocks mined by honest parties $\geq L$

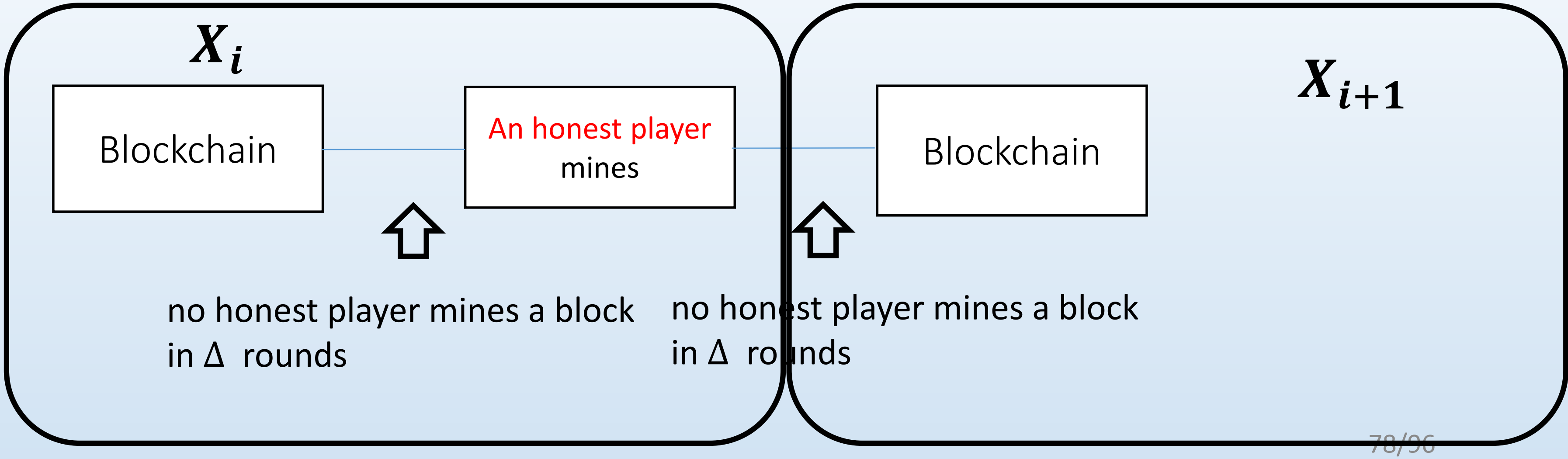
- $L = (1 - \delta')\alpha t$ for δ' is a constant $\in (0,1)$, it means the lower bound of blocks mined by honest players in t rounds

Proof of no divergence

- Let $\boldsymbol{X_i} = 1$ if

  the round between the ith block and the $(i + 1)$th block mined by honest players is more than Δ rounds and exactly one honest player mines a block

Otherwise, $\boldsymbol{X_i} = 0$



Proof of no divergence

- $Pr[X_i = 0] \geq (\Delta + 1)\alpha$
- Thus, $\Pr[X_i = 1] \leq 1 - (\Delta + 1)\alpha$
- We let $\boldsymbol{X} = \Sigma_{i=1}^{L} \boldsymbol{X_i}$
- By Chernoff bound,

$$\Pr[\boldsymbol{X} < X] \leq e^{-\Omega(\delta''^2 X)}$$

- $X = (1 - \delta'')(1 - (\Delta + 1)\alpha)L$ is the lower bound of $\boldsymbol{X}$

Proof of no divergence

- Let $\boldsymbol{Y_i} = 1$ if

  $\boldsymbol{X_i} = 1$ and $\boldsymbol{X_{i+1}} = 1$

  otherwise $\boldsymbol{Y_i} = 0$
- $\boldsymbol{Y_i}$ means the whether the pattern occurs. If the pattern occurs, it is a convergence opportunity
- Let $\boldsymbol{Y} = \sum_{\mathrm{i}=1}^{\mathrm{L}} \boldsymbol{Y_{\mathrm{i}}}$, it means the number of patterns occured in t rounds

Proof of no divergence

- $\boldsymbol{X_i} = 0$ will ruin at most 2 convergence opportunity

$$\mathbf{Y} \geq \Sigma_{\mathrm{i}=1}^{\mathrm{L}} 1 - 2(1 - \mathbf{X_i}) = 2\boldsymbol{X} - L$$
$$\geq ((1 - \delta')\alpha t)(1 - 2\delta'' - 2\alpha(\Delta + 1))$$

- Choosing proper $\delta'$and δ'',

$$\mathbf{Y} \geq (1 - \delta''')\alpha t(1 - 2\alpha(\Delta + 1))$$

- It means Adversary should mine at least

$(1 - \delta''')\alpha t(1 - 2\alpha(\Delta + 1))$ new blocks in t rounds

Proof of no divergence

- Now we calculate the maximum number of blocks mined by Adversary
- By no long block withholding and chain growth upper bound,

$\forall\ constant\ \omega, \omega'$, Adversary can mine up to $(1+\omega')(1+\omega)(t+1)\beta$ blocks in $t+1$ rounds except negligible probability

Proof of no divergence

- By the condition that $\alpha\big(1 - 2\alpha(\Delta + 1)\big) \geq (1 + \delta)\beta$, Adversary can mine at most

$\frac{(1+\omega')(1+\omega)}{(1+\delta)}\ (1 - 2\alpha(\Delta + 1))\ \alpha(t + 1)$ blocks

But Adversary should mine $(1 - \delta''')\alpha t(1 - 2\alpha(\Delta + 1))$ blocks

By picking proper constant, the number will be smaller than the blocks Adversary should mine

Consistency property

- Discuss the condition may diverge the blockchain

(A mines a blocks by withholding or during freeze round)

- Prove no long block withholding
- Prove that : the probability that A succeed in mines a blocks in each pattern is negligible

Outline

- Introduction
- Main Result
- Blockchain Protocol
- $\mathcal{F}_{tree}$ Hybrid Model
- Nakamoto's Model v.s. Hybrid Models
- Proof of the Consistency in Asynchronous Networks
- Conclusion

Conclusion

- Analyze the blockchain protocol in asynchronous network
- Prove that it satisfies consistency in asynchronous network

Conclusion : How to prove it?

- Make an abstract of blockchain protocol $\mathcal{F}_{tree}$ and prove that it is as secure as Nakamoto's protocol by using Hybrid models
- Prove the chain-growth property and chain quality property in $\mathcal{F}_{tree}$
- Prove the consistency in $\mathcal{F}_{tree}$ by using chain-growth property and chain quality property

Conclusion

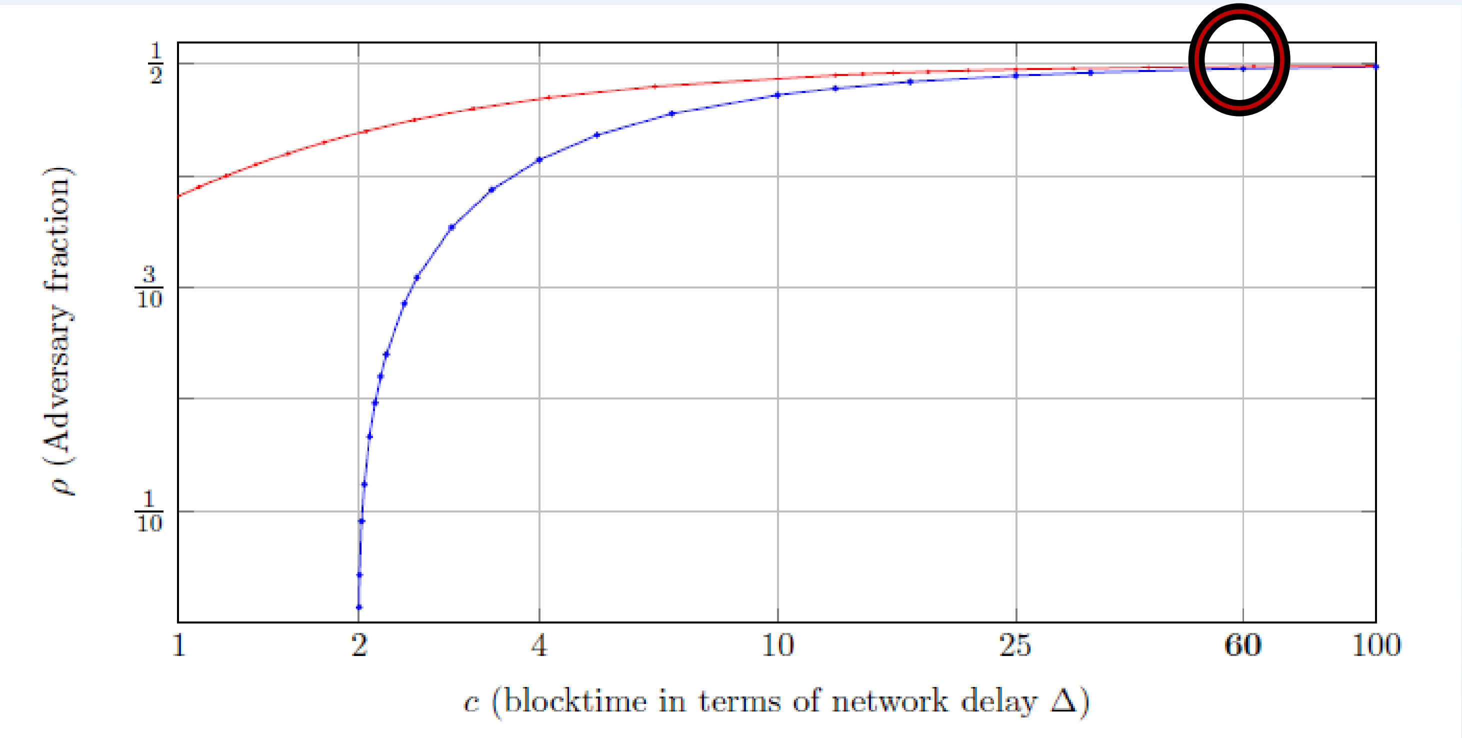
By the result of Consistency property,

we have that the chain consistency holds when:

$$\alpha\big(1 - 2\alpha(\Delta + 1)\big) \geq (1 + \delta)\beta$$

Conclusion

Consider the probability $p = \frac{1}{cn\Delta}$. If we have $n = 10^5, \Delta = 10^{13}, \mathrm{c} = 60$, we allow A has 49.57% computational power [PSS16]



Take from : [PSS16]