But Why Does it Work?

A Rational Protocol Design Treatment of Bitcoin

Christian Badertscher ETH Zurich
Juan Garay Texas A&M
Ueli Maurer ETH Zurich
Daniel Tschudi ETH Zurich

Vassilis Zikas
University of Edinburgh & IOHK

EUROCRYPT 2018

The Evolution of Bitcoin: A Partial View

Time

The Evolution of Bitcoin: A Partial View

Time

2008/09 White Paper & Genesis

And Nakamoto said:
Let there be Bitcoin…

The Evolution of Bitcoin: A Partial View

Time

2008/09 — White Paper & Genesis

The Bitcoin community

And Nakamoto said:
Let there be Bitcoin…

The Evolution of Bitcoin: A Partial View

Time

2008/09 — White Paper & Genesis

And Nakamoto said:
Let there be Bitcoin…

The Bitcoin community

Rational analysis and attacks

- Selfish Mining: Bitcoin is **not an equilibrium** strategy [ES14]
- ….

The Evolution of Bitcoin: A Partial View

Time

2008/09 — White Paper & Genesis

And Nakamoto said:
Let there be Bitcoin…

— The Bitcoin community

— Rational analysis and attacks

- Selfish Mining: Bitcoin is **not an equilibrium** strategy [ES14]
- ….

— Cryptographic analysis: *Backbone* (consensus layer) is **secure *if and only if*** the computing power of adversarial nodes does not form a majority [GKL15, PSS17]

The Evolution of Bitcoin: A Partial View

Time

2008/09 — White Paper & Genesis

And Nakamoto said:
Let there be Bitcoin…

The Bitcoin community

Rational analysis and attacks

- Selfish Mining: Bitcoin is **not an equilibrium** strategy [ES14]
- ….

Cryptographic analysis: *Backbone* (consensus layer) is **secure *if and only if*** the computing power of adversarial nodes does not form a majority [GKL15, PSS17]

The Evolution of Bitcoin: A Partial View

Time

2008/09 — White Paper & Genesis

And Nakamoto said:
Let there be Bitcoin…

— The Bitcoin community

— Rational analysis and attacks

- Selfish Mining: Bitcoin is **not an equilibrium** strategy [ES14]
- ….

— Cryptographic analysis: *Backbone* (consensus layer) is **secure *if and only if*** the computing power of adversarial nodes does not form a majority [GKL15, PSS17]

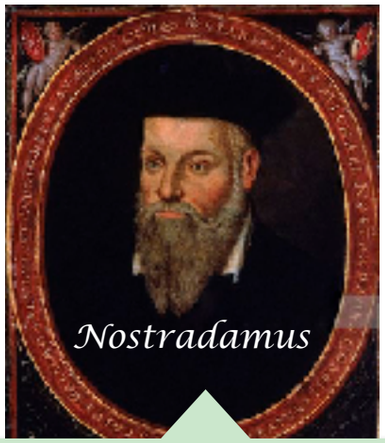
2018 — Bitcoin still works and no attack on its "backbone" has been observed!

Why Does it Work?

Why don’t the predicted attacks occur and entirely break it?

Why Does it Work?

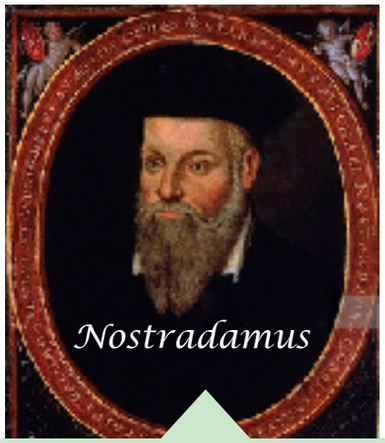
Why don’t the predicted attacks occur and entirely break it?



It doesn’t! Not an equilibrium. Just a temporary anomaly.

Why Does it Work?

Why don't the predicted attacks occur and entirely break it?

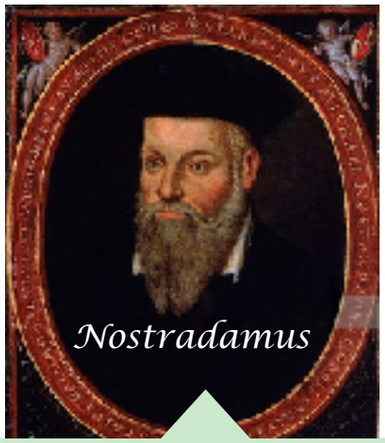


Because the majority of computing power is controlled by honest miners

It doesn't! Not an equilibrium. Just a temporary anomaly.

Why Does it Work?

Why don't the predicted attacks occur and entirely break it?



Because the majority of computing power is controlled by honest miners

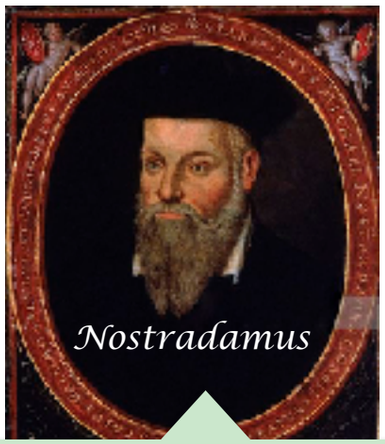
It doesn't! Not an equilibrium. Just a temporary anomaly.

In game-theoretic analysis

- Utilities = assumptions to explain/predict players behavior
- If predictions ≠ observable then utilities (and game?) can (should?) be rethought.

Why Does it Work?

Why don't the predicted attacks occur and entirely break it?



Because the majority of computing power is controlled by honest miners

but … why … ?

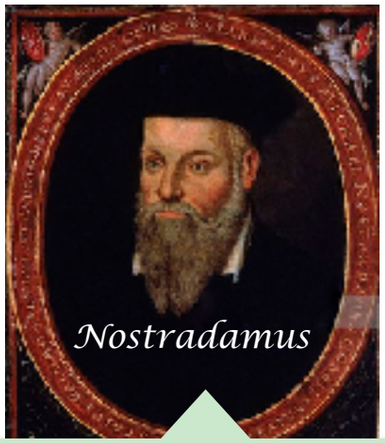
It doesn't! Not an equilibrium. Just a temporary anomaly.

In game-theoretic analysis

- Utilities = assumptions to explain/predict players behavior
- If predictions ≠ observable then utilities (and game?) can (should?) be rethought.

Why Does it Work?

Why don't the predicted attacks occur and entirely break it?



Because the majority of computing power is controlled by honest miners

but … why … ?

Can we back this up by a rational assumption?

- Because the adversary has no incentive to break it (either by corrupting majority or otherwise)

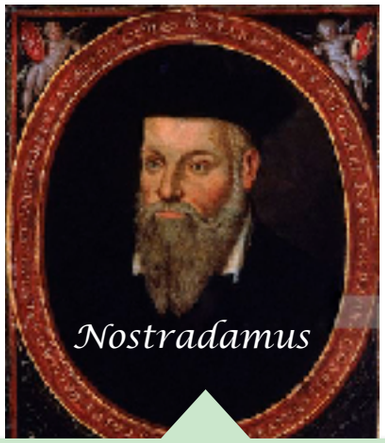
It doesn't! Not an equilibrium. Just a temporary anomaly.

In game-theoretic analysis

- Utilities = assumptions to explain/predict players behavior
- If predictions ≠ observable then utilities (and game?) can (should?) be rethought.

Why Does it Work?

Why don't the predicted attacks occur and entirely break it?



Because the majority of computing power is controlled by honest miners

but … why … ?

Can we back this up by a rational assumption?

- Because the adversary has no incentive to break it (either by corrupting majority or otherwise)

It doesn't! Not an equilibrium. Just a temporary anomaly.

In game-theoretic analysis

- Utilities = assumptions to explain/predict players behavior
- If predictions ≠ observable then utilities (and game?) can (should?) be rethought.

Calls for an alternative rational treatment

Our Contributions

- A new model for rational analysis of Bitcoin
- Applying the framework to analyze the Bitcoin backbone
  - A class of utilities reflecting “minimal” assumptions about the Bitcoin miners’ incentives.
  - Deriving predictions that match the observable.

Our Contributions

- A new model for rational analysis of Bitcoin

Blockchains

- Applying the framework to analyze the Bitcoin backbone
  - A class of utilities reflecting “minimal” assumptions about the Bitcoin miners’ incentives.
  - Deriving predictions that match the observable.

Our Contributions

- A new model for rational analysis of Bitcoin

Blockchains

- Applying the framework to analyze the Bitcoin backbone
  - A class of utilities reflecting “minimal” assumptions about the Bitcoin miners’ incentives.
  - Deriving predictions that match the observable.

Rational Protocol Design (RPD) [GKMTZ13]

Securely implementing a task against
an incentive-driven adversary

Rational Protocol Design (RPD) [GKMTZ13]

Securely implementing a task against
an incentive-driven adversary

The Attack Game

Protocol Designer

(n-party) task as an ideal functionality $\mathcal{F}$

Protocol Attacker

u_D

u_A

Rational Protocol Design (RPD) [GKMTZ13]

Securely implementing a task against
an incentive-driven adversary

The Attack Game

Protocol Designer

u_D

(n-party) task as an ideal functionality $\mathcal{F}$

(n-party) protocol π for $\mathcal{F}$

Protocol Attacker

u_A

Rational Protocol Design (RPD) [GKMTZ13]

Securely implementing a task against
an incentive-driven adversary

The Attack Game

(n-party) task as an ideal functionality $\mathcal{F}$

Protocol Designer

u_D

(n-party) protocol π for $\mathcal{F}$

Adversary $\mathcal{A}$ *for attacking* π

Protocol Attacker

u_A

Rational Protocol Design (RPD) [GKMTZ13]

Securely implementing a task against an incentive-driven adversary

The Attack Game

(n-party) task as an ideal functionality $\mathcal{F}$

Protocol Designer

u_D

(n-party) protocol π for $\mathcal{F}$

Adversary $\mathcal{A}$ *for attacking* π

Protocol Attacker

u_A

- Utilities are defined in the ideal world as payoffs of explicit "breaks" of $\mathcal{F}$
- zero-sum game (i.e., $u_D := - u_A$)

Rational Protocol Design (RPD) [GKMTZ13]

Flavors of Protocol Quality (security / stability)

- π is *(u_D, u_A, Π)-attack-payoff optimal* for $\mathcal{F}$ if any other protocol in Π allows for more rewarding attacks
  - π is a best-response strategy among protocols in Π
- π is *$(u_D, u_A, \mathbb{A})$-attack-payoff secure* for $\mathcal{F}$ if the best the attacker can do is play an adversary in $\mathbb{A}$
  - an $\mathbb{A}$-adversary is best response to π

In [GKMTZ13]**:**

✳ Π = The class of all poly-time protocols

✳ $\mathbb{A}$ = The class of all adversaries that honestly execute the protocol

Rational Protocol Design (RPD) [GKMTZ13]

Flavors of Protocol Quality (security / stability)

- π is *(u_D, u_A, Π)-attack-payoff optimal* for $\mathcal{F}$ if any other protocol in Π allows for more rewarding attacks
  - π is a best-response strategy among protocols in Π
- π is *$(u_D, u_A, \mathbb{A})$-attack-payoff secure* for $\mathcal{F}$ if the best the attacker can do is play an adversary in $\mathbb{A}$
  - an $\mathbb{A}$-adversary is best response to π

Rational Protocol Design (RPD)++

Flavors of Protocol Quality (security / stability)

- π is *$(u_D, u_A, \mathbb{\Pi})$-attack-payoff optimal* for $\mathcal{F}$ if any other protocol in $\mathbb{\Pi}$ allows for more rewarding attacks
  - π is a best-response strategy among protocols in $\mathbb{\Pi}$
- π is *$(u_D, u_A, \mathbb{A})$-attack-payoff secure* for $\mathcal{F}$ if the best the attacker can do is play an adversary in $\mathbb{A}$
  - an $\mathbb{A}$-adversary is best response to π

Rational Protocol Design (RPD)++

Flavors of Protocol Quality (security / stability)

- π is *$(u_D, u_A, \mathbb{\Pi})$-attack-payoff optimal* for $\mathcal{F}$ if any other protocol in $\mathbb{\Pi}$ allows for more rewarding attacks
  - π is a best-response strategy among protocols in $\mathbb{\Pi}$
- π is *$(u_D, u_A, \mathbb{A})$-attack-payoff secure* for $\mathcal{F}$ if the best the attacker can do is play an adversary in $\mathbb{A}$
  - an $\mathbb{A}$-adversary is best response to π
- π is *$(u_D, u_A, (\mathbb{A}, \mathbb{\Pi}))$-incentive compatible* for $\mathcal{F}$ if it is $(u_D, u_A, \mathbb{A})$-attack-payoff *secure* **AND** *$(u_D, u_A, \mathbb{\Pi})$-attack-payoff optimal*

Bitcoin in RPD++

Flavors of Protocol Quality (security / stability)

- π is *(u_D, u_A, Π)-attack-payoff optimal* for $\mathcal{F}$ if any other protocol in Π allows for more rewarding attacks
  - π is a best-response strategy among protocols in Π
- π is *$(u_D, u_A, \mathbb{A})$-attack-payoff secure* for $\mathcal{F}$ if the best the attacker can do is play an adversary in $\mathbb{A}$
  - an $\mathbb{A}$-adversary is best response to π
- π is *$(u_D, u_A, (\mathbb{A}, \Pi))$-incentive compatible* for $\mathcal{F}$ if it is *$(u_D, u_A, \mathbb{A})$-attack-payoff secure* **AND** *(u_D, u_A, Π)-attack-payoff optimal*

For Bitcoin

✳ Π = The class of protocols that use the Bitcoin infrastructure (circulate blocks and transactions of the right format)

✳ $\mathbb{A}$ = The class of semi-honest network-rushing adversaries

➡ *strongly (u_D, u_A)-attack-payoff secure*

Bitcoin in RPD++

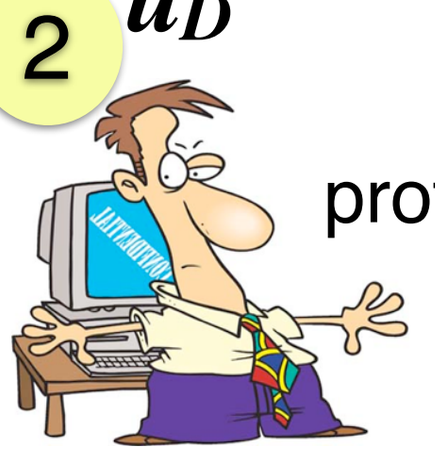
The Bitcoin Attack Game

(n-party) task as an ideal functionality $\mathcal{F}$

1

Protocol Designer

2 u_D



(n-party) protocol π for $\mathcal{F}$

Protocol Attacker

u_A 2

Adversary $\mathcal{A}$ *for attacking* π

- Utilities are defined in the ideal world as payoffs of explicit "breaks" of $\mathcal{F}$
- zero-sum game (i.e., $u_D := - u_A$)

Bitcoin in RPD++

The Bitcoin Attack Game

(n-party) task as an ideal functionality $\mathcal{F}$

Protocol Designer

u_D

2

(n-party) protocol π for $\mathcal{F}$

1

Protocol Attacker

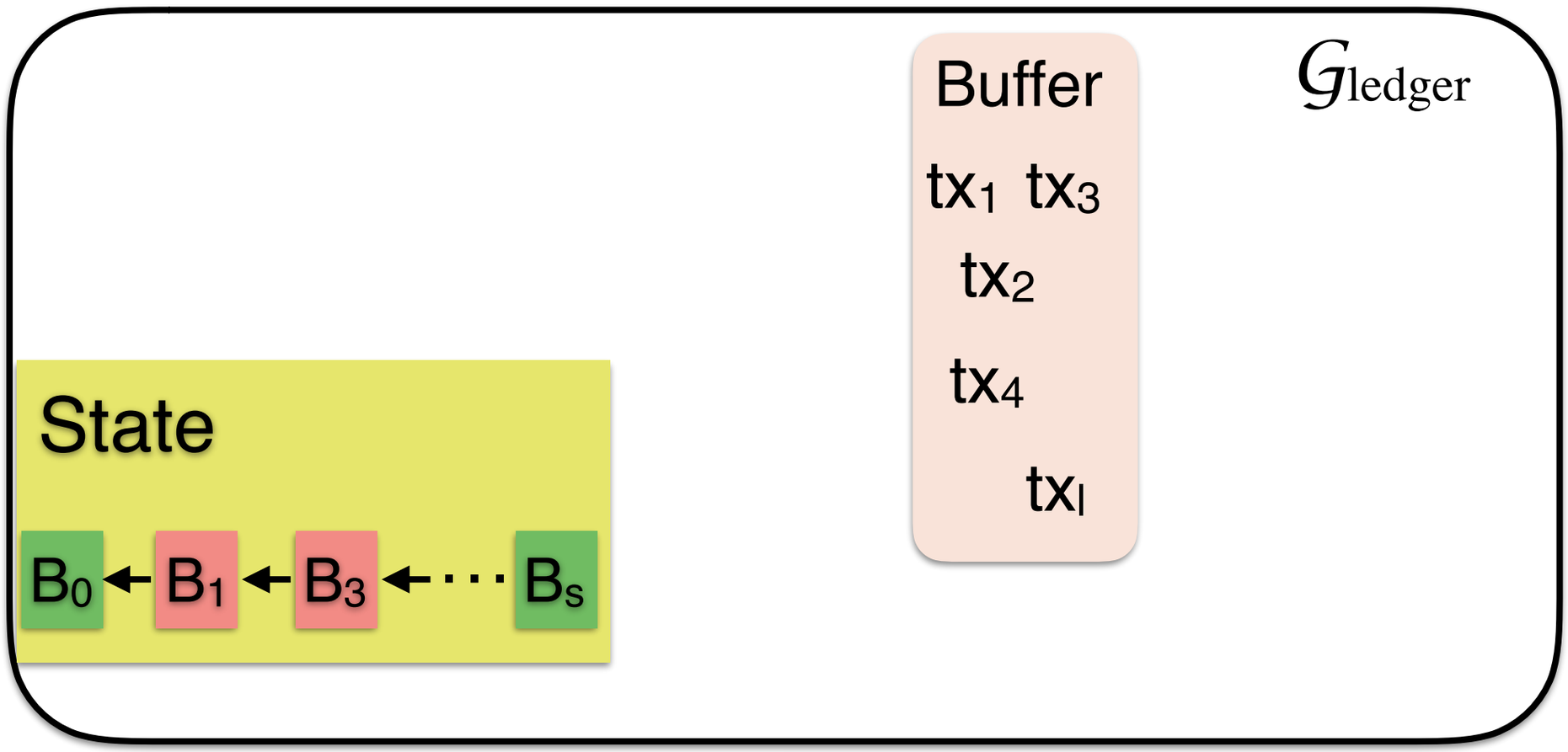
u_A

2

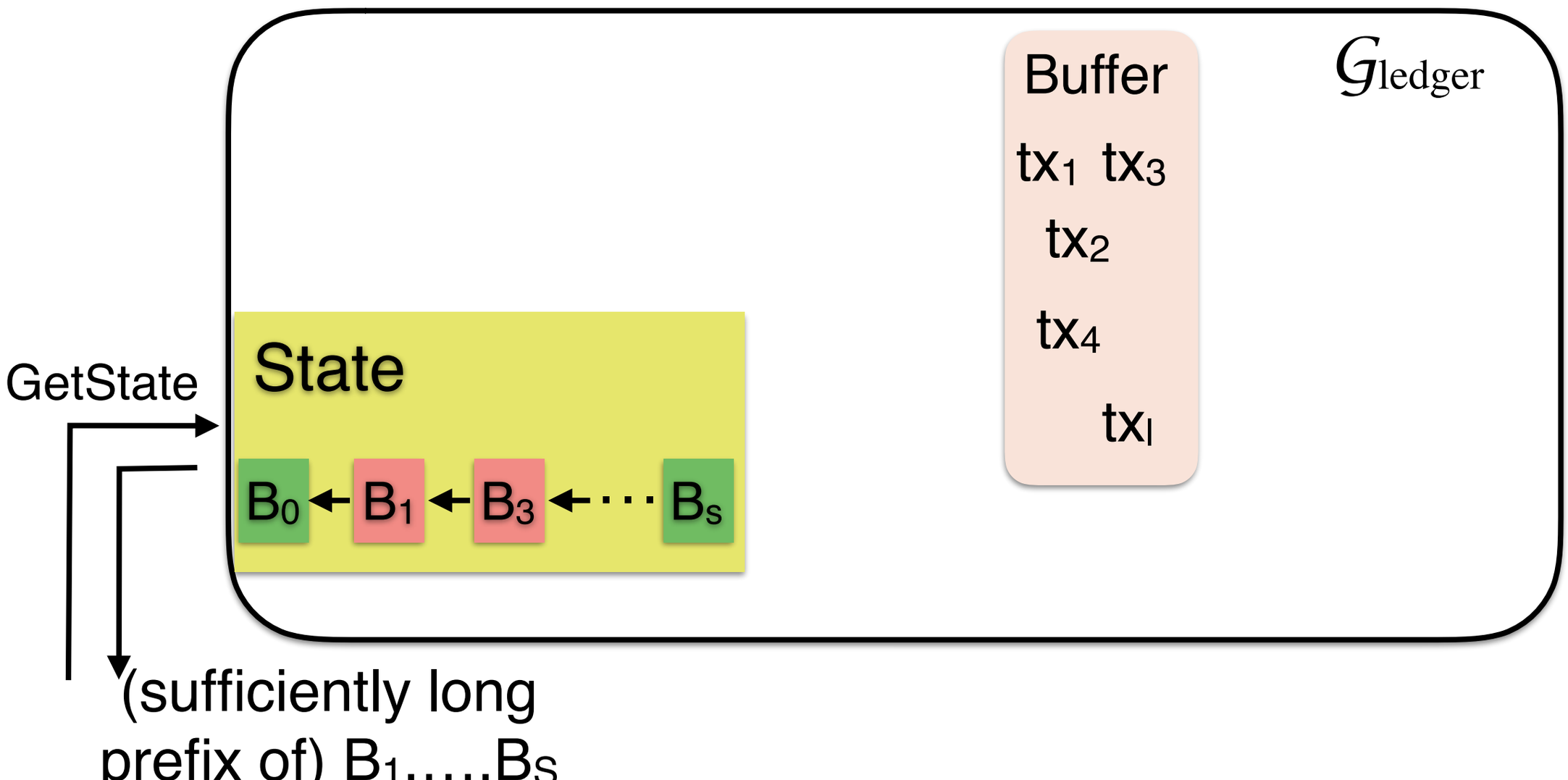
Adversary $\mathcal{A}$ *for attacking* π

- Utilities are defined in the ideal world as payoffs of explicit "breaks" of $\mathcal{F}$
- zero-sum game (i.e., $u_D := - u_A$)

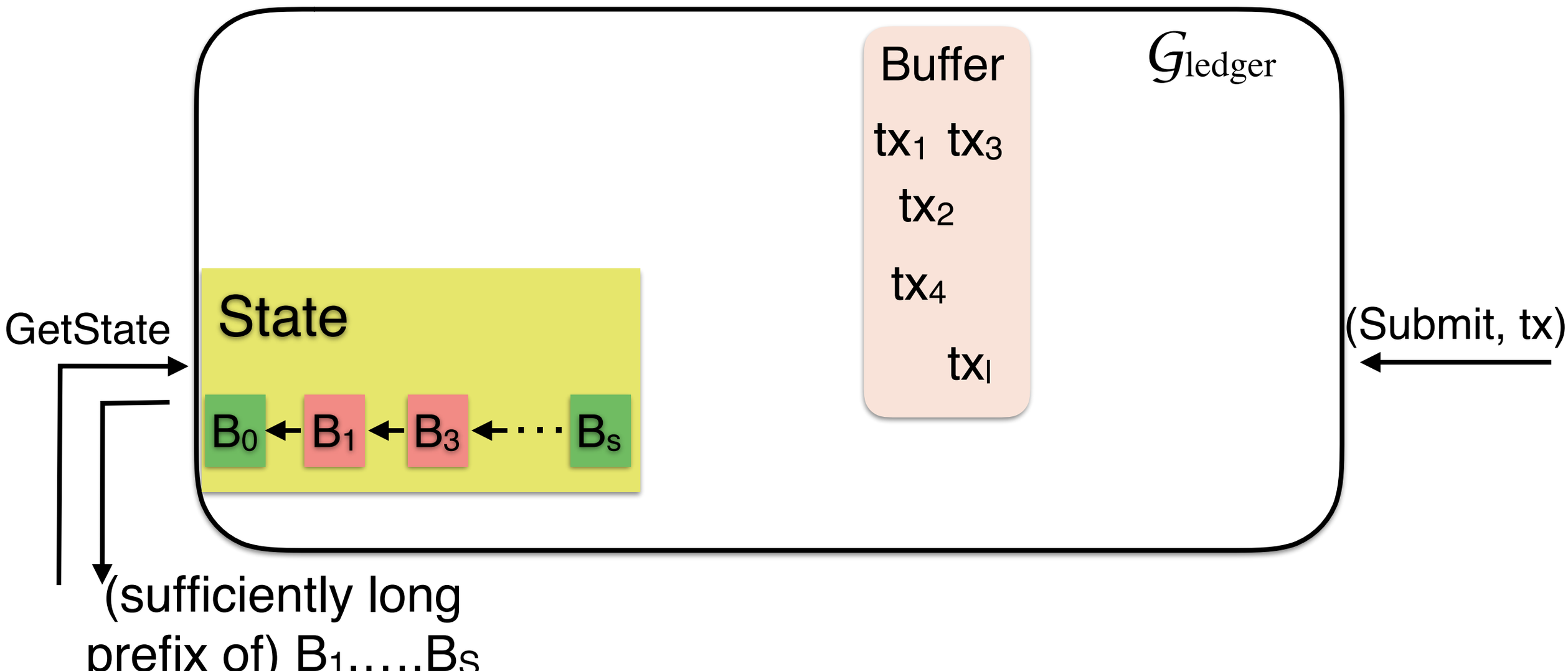
Bitcoin an a Transaction Ledger [BMTZ17]



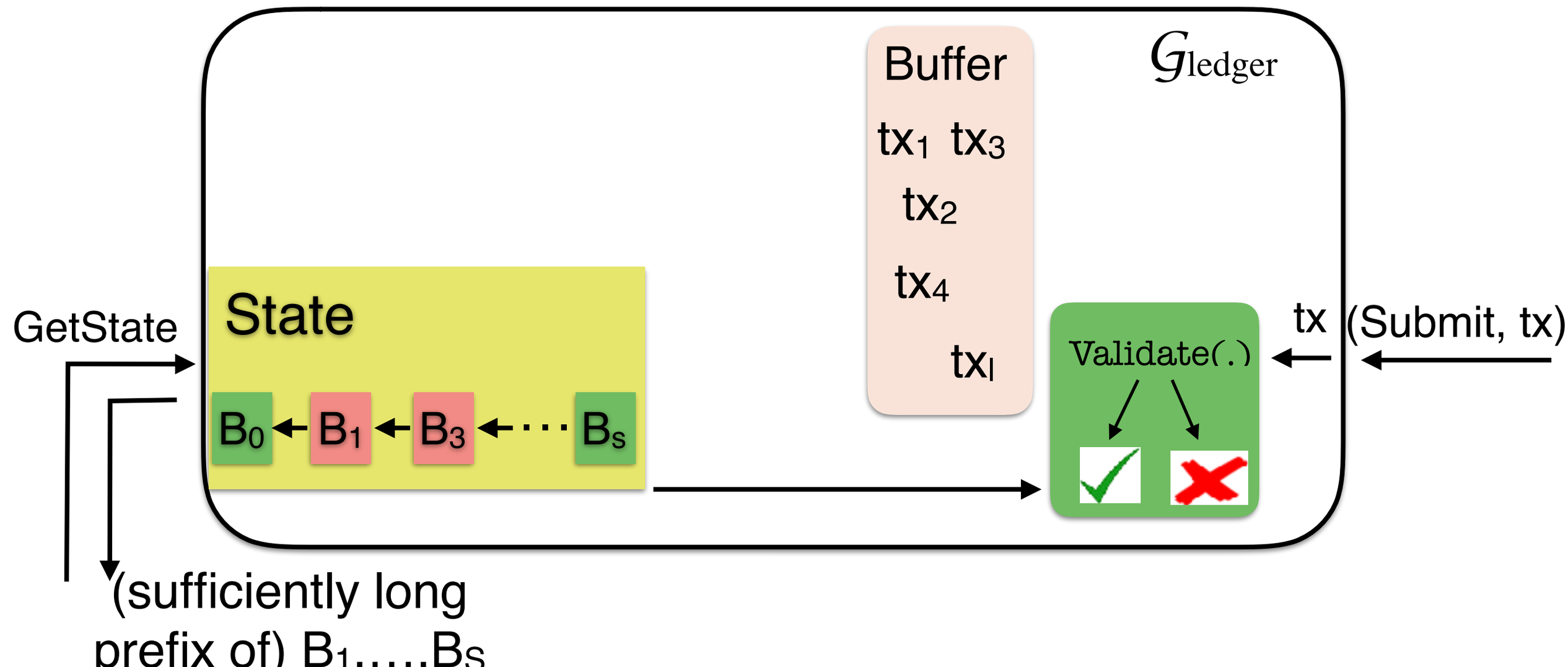
Bitcoin an a Transaction Ledger [BMTZ17]



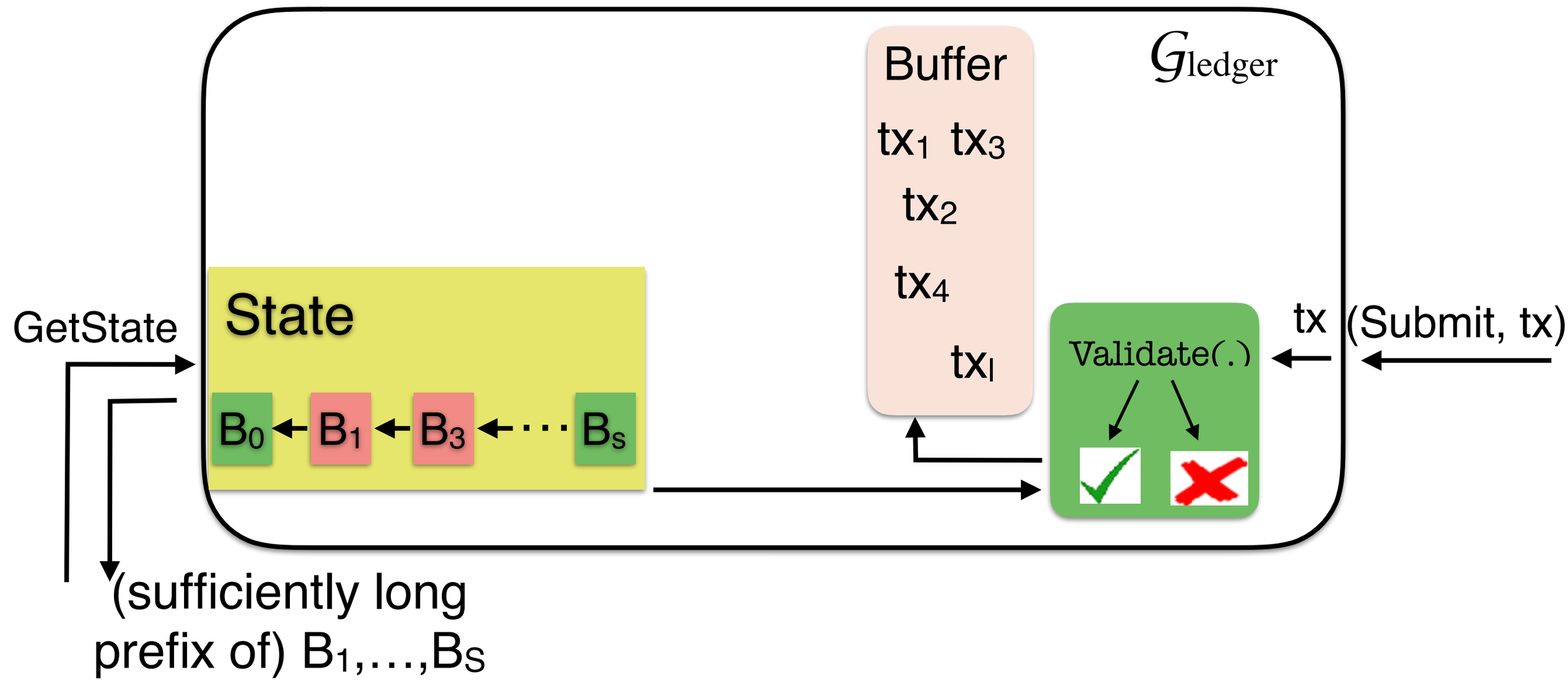
Bitcoin an a Transaction Ledger [BMTZ17]



Bitcoin an a Transaction Ledger [BMTZ17]



Bitcoin an a Transaction Ledger [BMTZ17]



Bitcoin an a Transaction Ledger [BMTZ17]

tx

Buffer

tx_1 tx_3

tx_2

tx_4

tx tx_l

$\mathcal{G}_{\text{ledger}}$

Validate(.)

tx (Submit, tx)

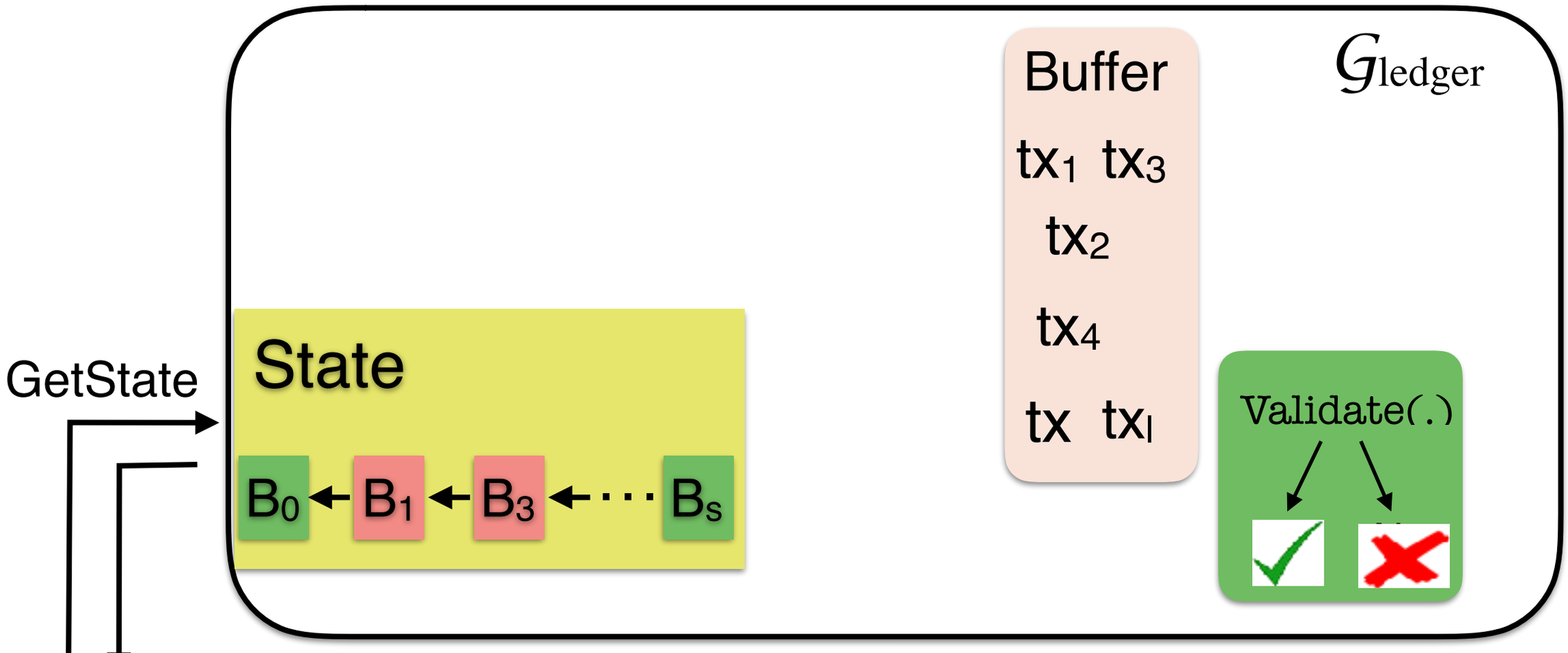
GetState

State

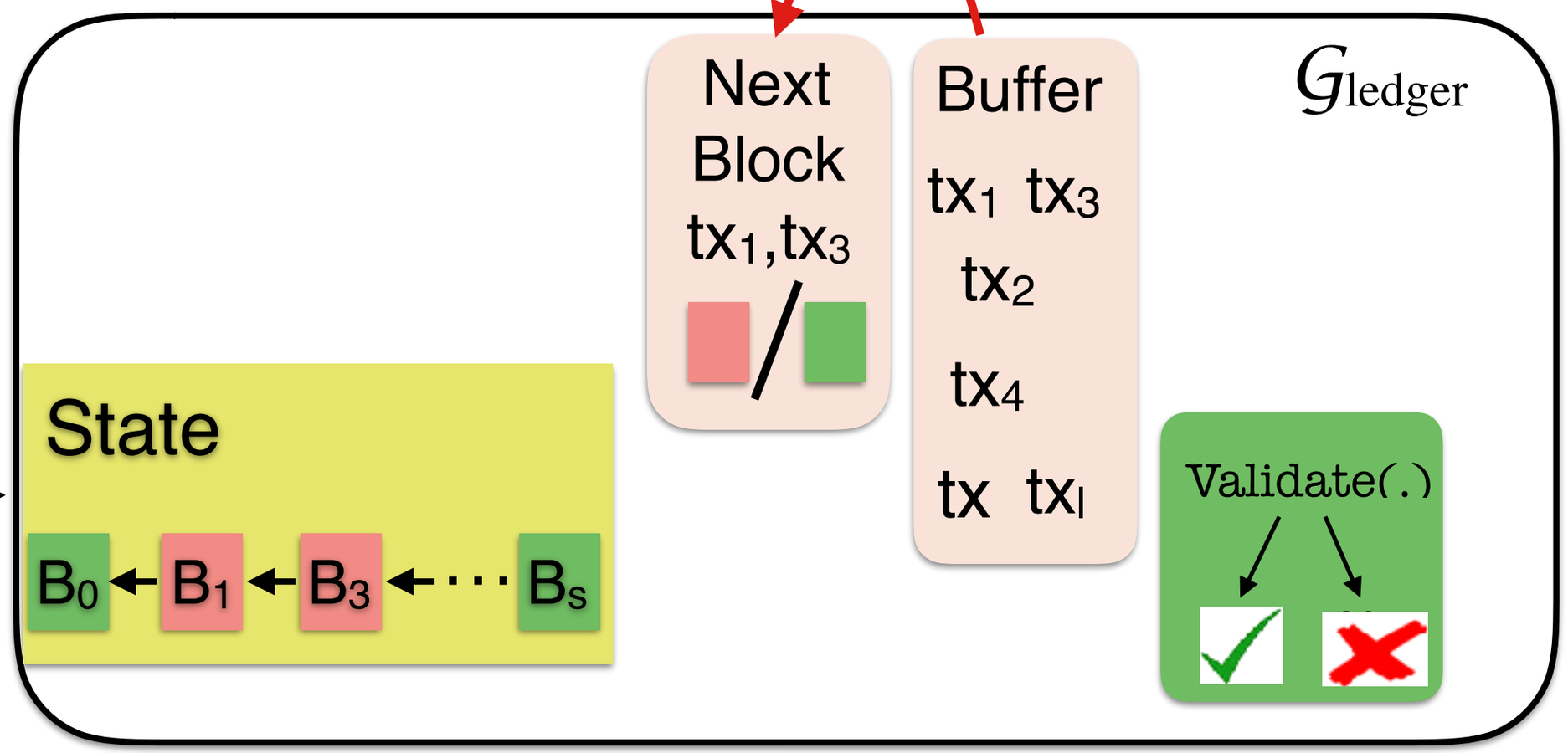
$B_0 \leftarrow B_1 \leftarrow B_3 \leftarrow \cdots B_s$

(sufficiently long prefix of) $B_1,\ldots,B_S$

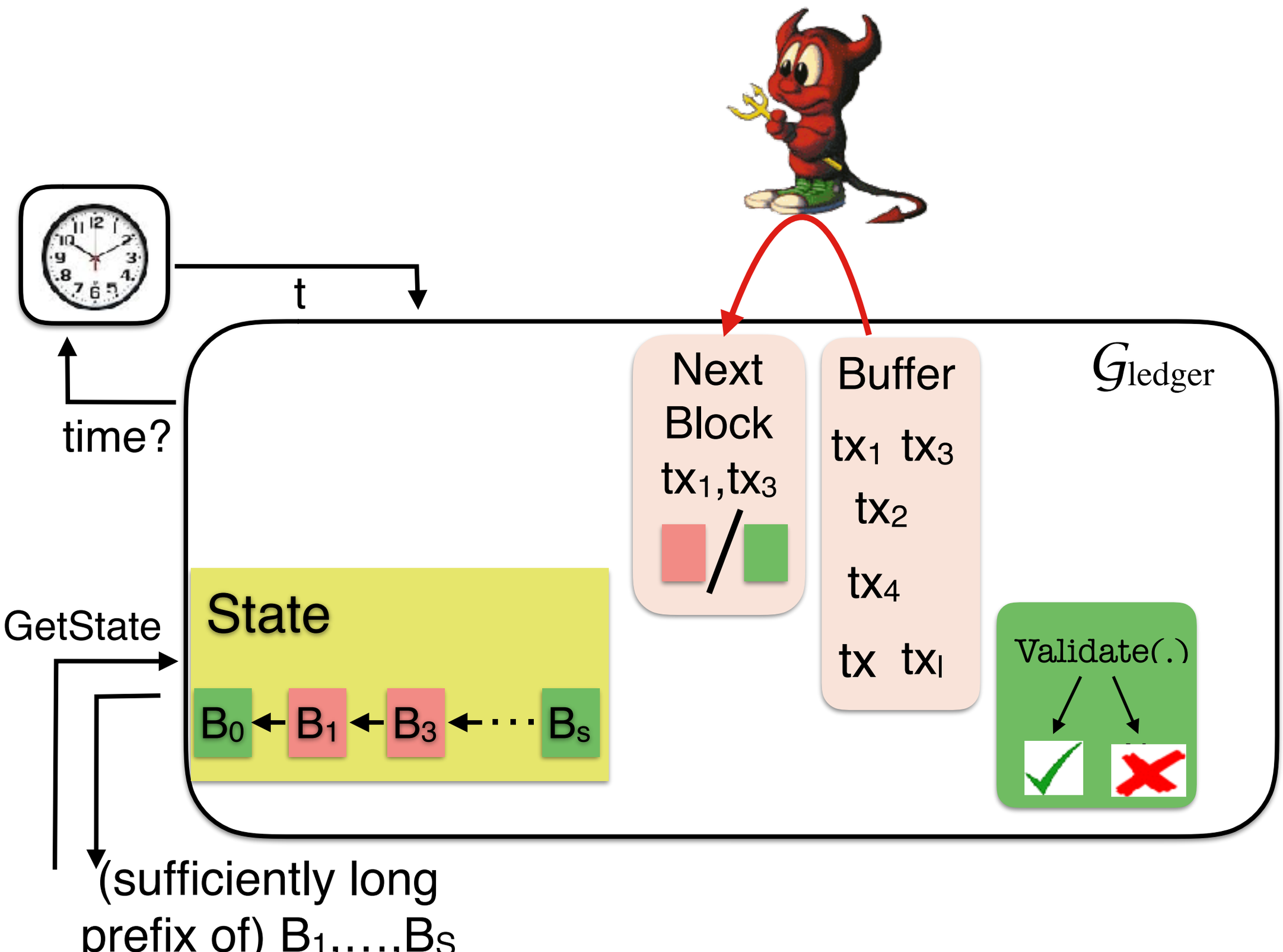
Bitcoin an a Transaction Ledger [BMTZ17]



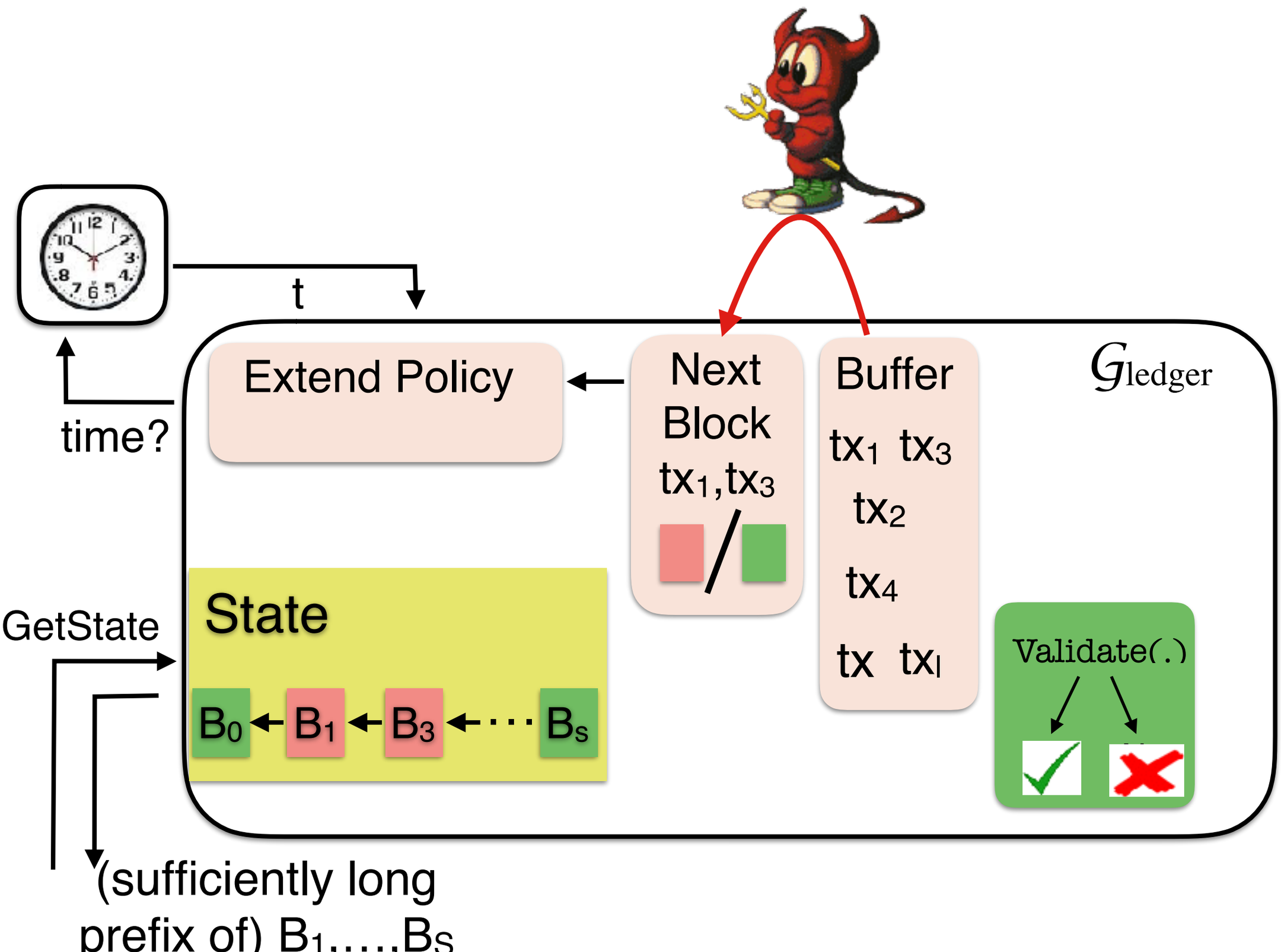
Bitcoin an a Transaction Ledger [BMTZ17]



Bitcoin an a Transaction Ledger [BMTZ17]



Bitcoin an a Transaction Ledger [BMTZ17]



Bitcoin an a Transaction Ledger [BMTZ17]

- chain growth, chain quality,… [GKL15,PSS17]

t

time?

$\mathcal{G}_{\text{ledger}}$

Extend Policy

Next Block

tx_1,tx_3

Buffer

tx_1 tx_3

tx_2

tx_4

tx tx_l

Validate(.)

State

$B_0 \leftarrow B_1 \leftarrow B_3 \leftarrow \cdots B_s$

GetState

(sufficiently long prefix of) $B_1,\ldots,B_S$

Bitcoin an a Transaction Ledger [BMTZ17]

- chain growth, chain quality,… [GKL15,PSS17]

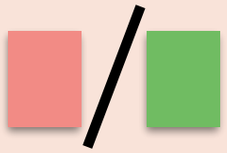
t

time?

Extend Policy

Next Block

tx_1, tx_3



Buffer

tx_1 tx_3

tx_2

tx_4

tx tx_l

$\mathcal{G}_{\text{ledger}}$

Validate(.)

State

$B_0 \leftarrow B_1 \leftarrow B_3 \leftarrow \cdots B_S$

GetState

(sufficiently long prefix of) $B_1,\ldots,B_S$

Bitcoin an a Transaction Ledger [BMTZ17]

- chain growth, chain quality,… [GKL15,PSS17]

t

time?

$\mathcal{G}_{\text{ledger}}$

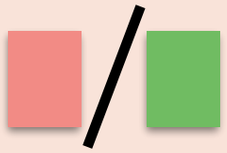
Extend Policy

Blockify

Next Block

Block

tx_1, tx_3



Buffer

tx_1 tx_3

tx_2

tx_4

tx tx_l

Validate(.)

State

$B_0 \leftarrow B_1 \leftarrow B_3 \leftarrow \cdots B_s$

GetState

(sufficiently long prefix of) $B_1,\ldots,B_S$

Bitcoin an a Transaction Ledger [BMTZ17]

- chain growth, chain quality,… [GKL15,PSS17]

t

time?

$\mathcal{G}_{\text{ledger}}$

Extend Policy ✓

Next Block

Block tx_1, tx_3

Buffer

tx_1 tx_3

tx_2

tx_4

tx tx_l

Blockify → B_{s+1}

State

$B_0 \leftarrow B_1 \leftarrow B_3 \leftarrow \cdots B_s$

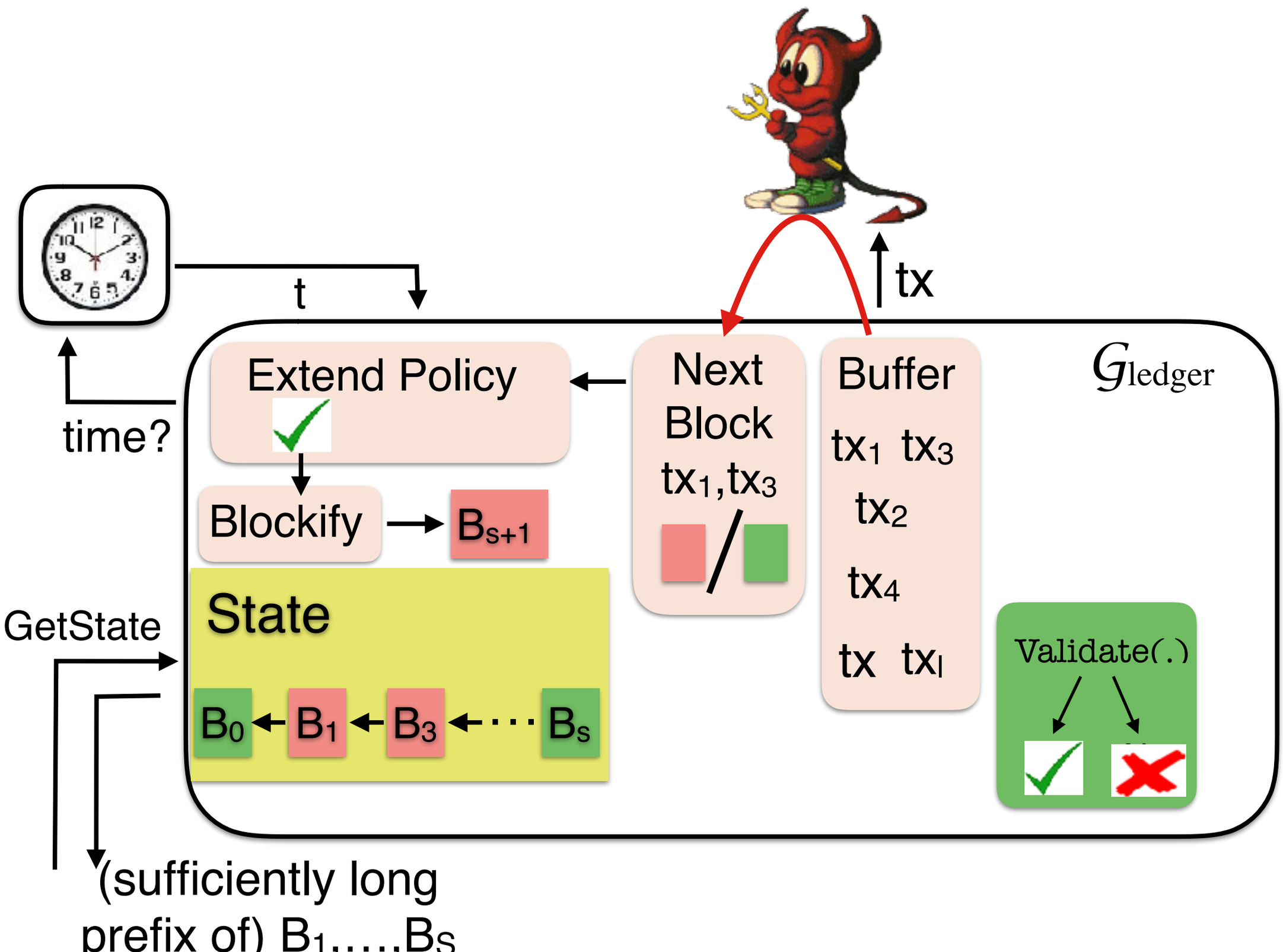
Validate(.)

✓ ✗

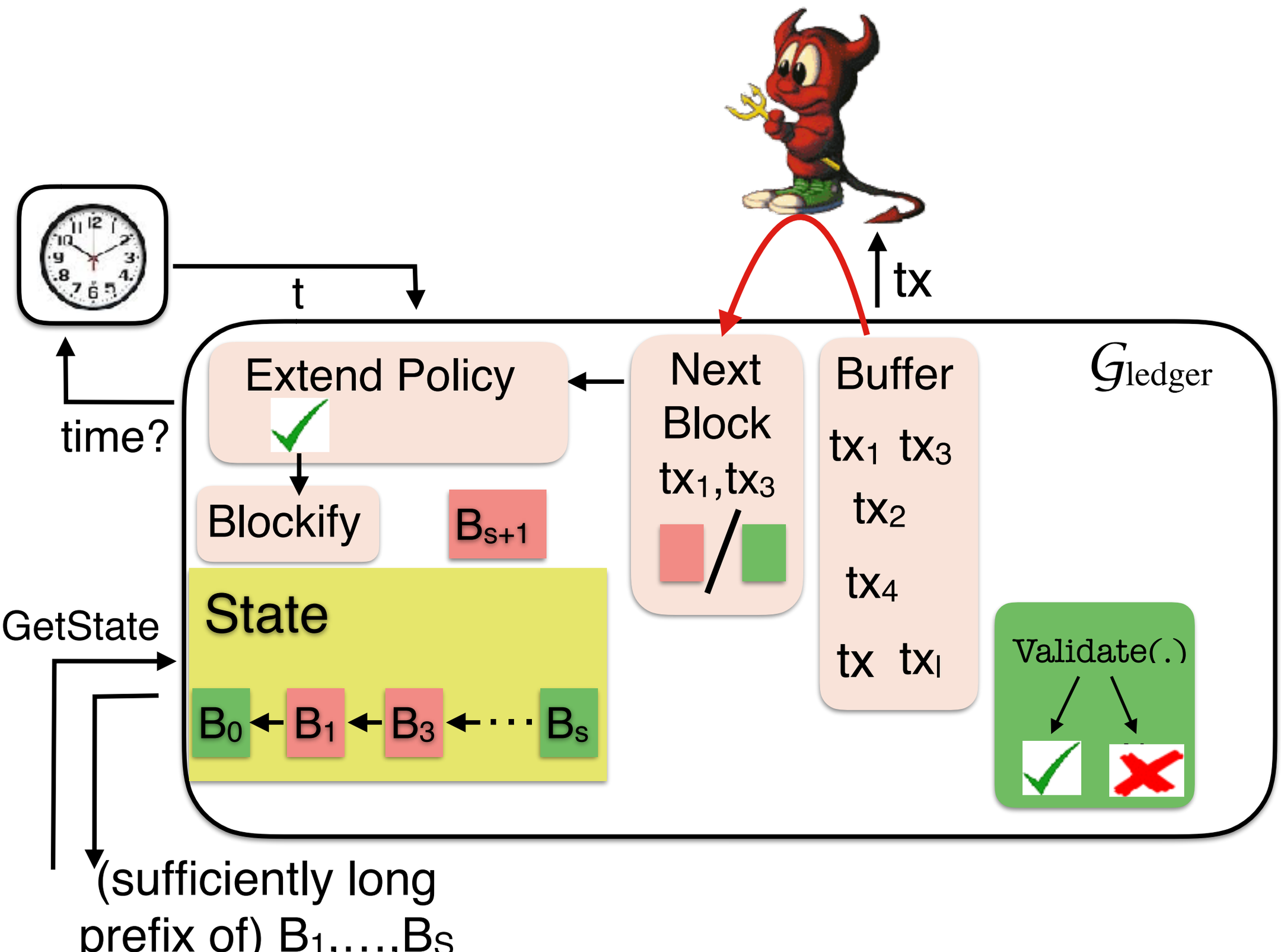
GetState

(sufficiently long prefix of) $B_1,\ldots,B_S$

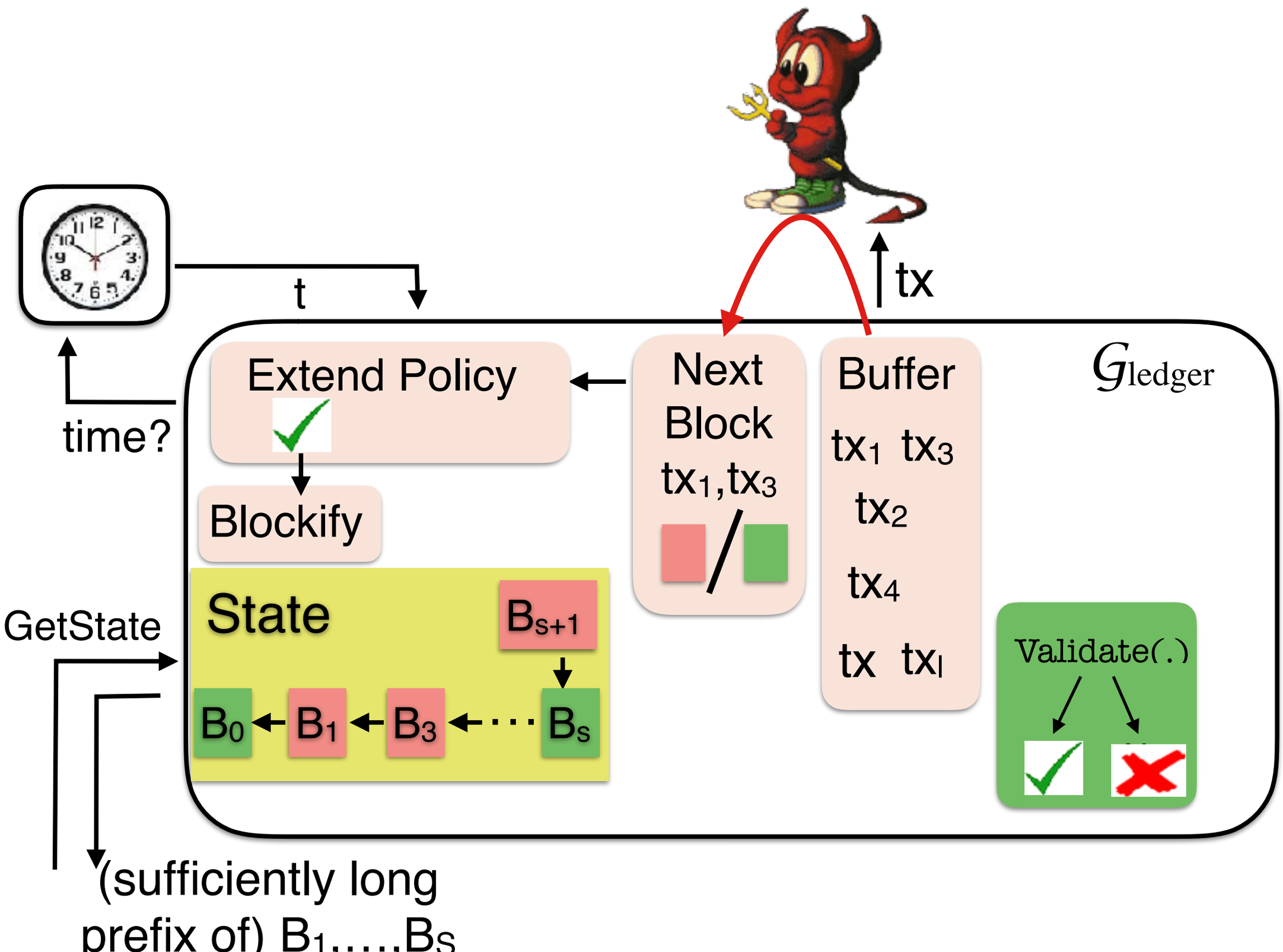
Bitcoin an a Transaction Ledger [BMTZ17]



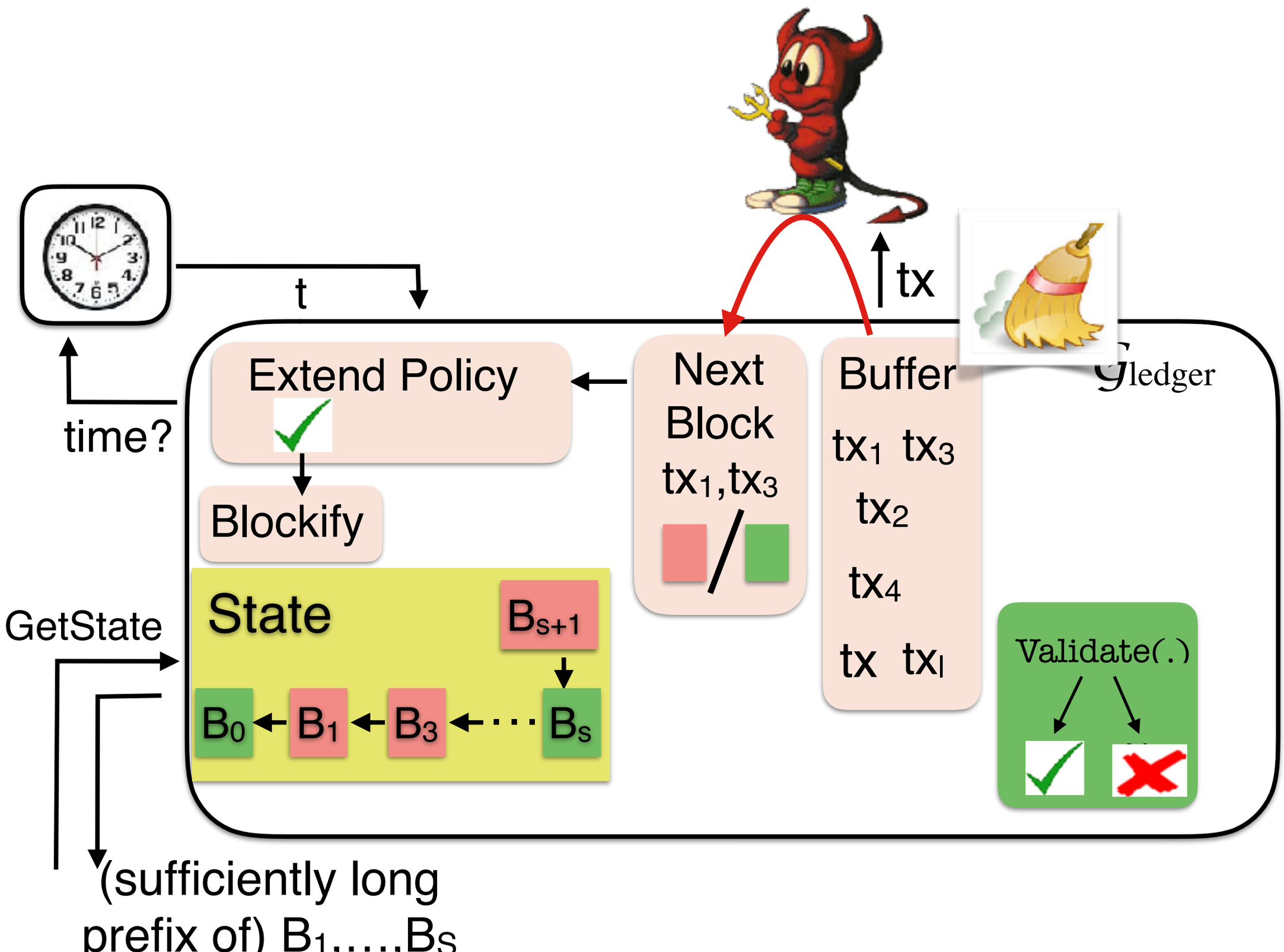
Bitcoin an a Transaction Ledger [BMTZ17]



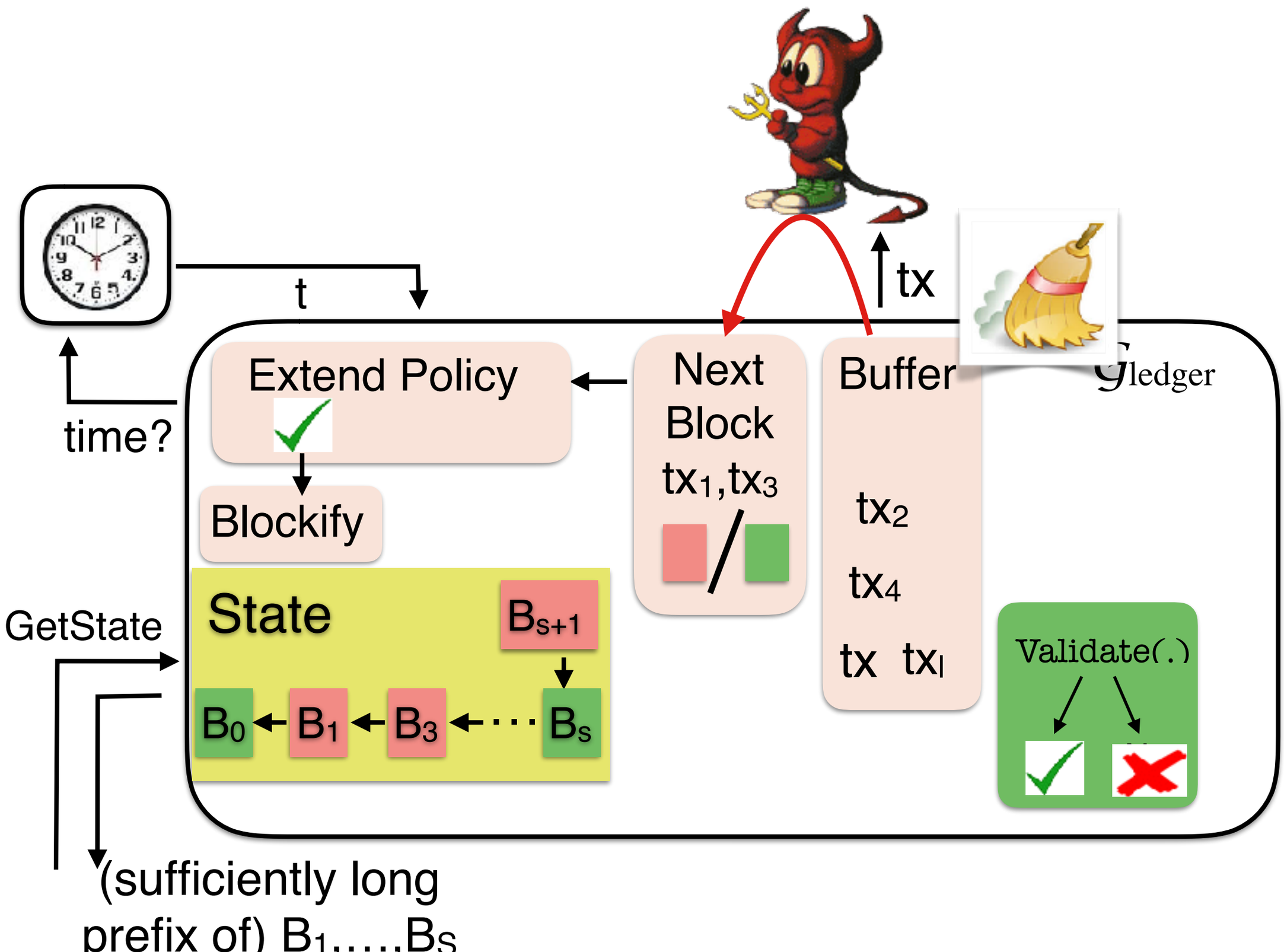
Bitcoin an a Transaction Ledger [BMTZ17]



Bitcoin an a Transaction Ledger [BMTZ17]

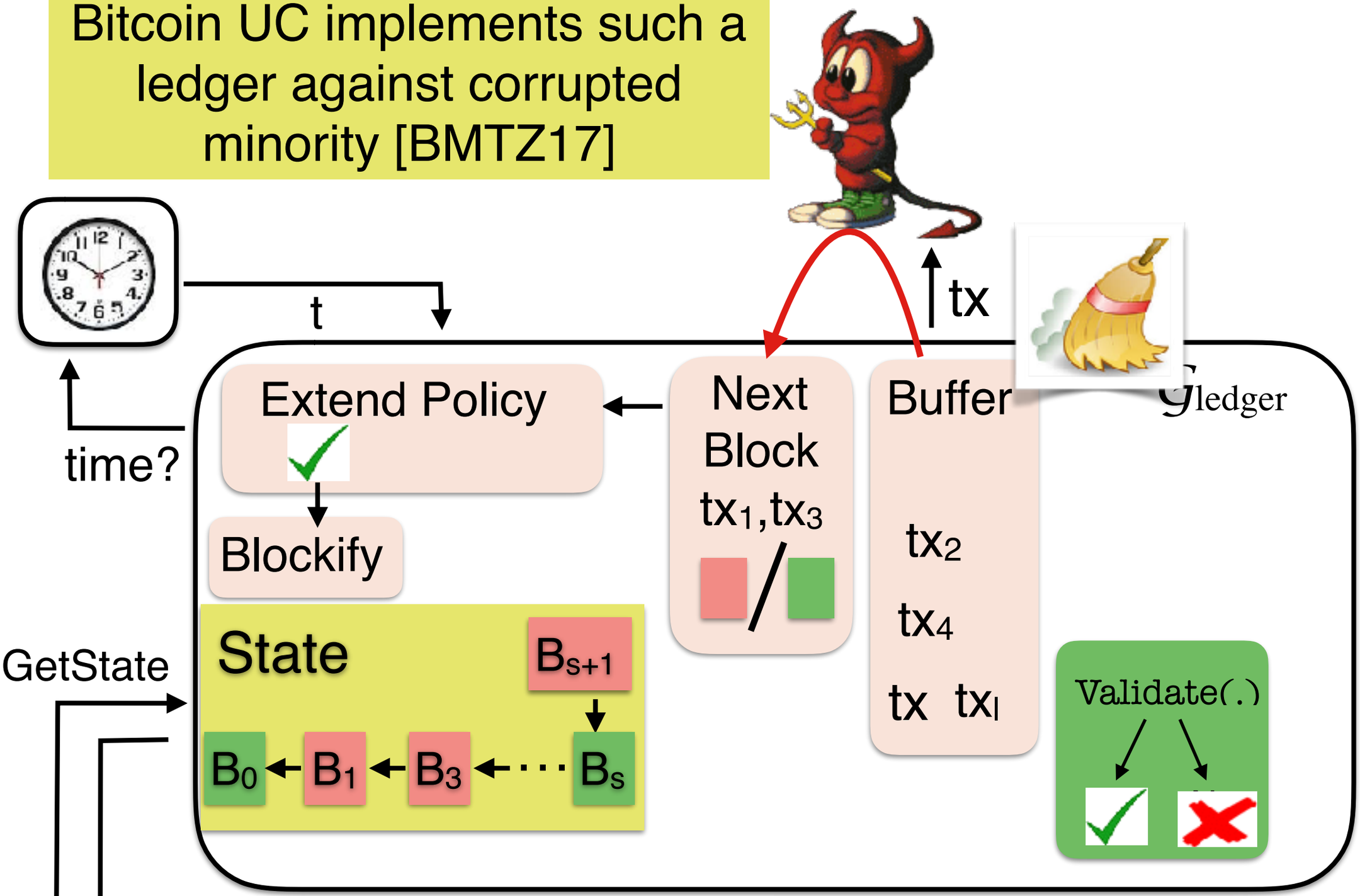


Bitcoin an a Transaction Ledger [BMTZ17]



Bitcoin an a Transaction Ledger [BMTZ17]

Bitcoin UC implements such a ledger against corrupted minority [BMTZ17]



Bitcoin in RPD++

The Bitcoin Attack Game

Protocol Designer

(n-party) task as an ideal functionality ~~$\mathcal{F}$~~ $\mathcal{F}_{\text{Ledger}}$ 1

Protocol Attacker

u_D 2

(n-party) protocol π for $\mathcal{F}$

u_A 2

Adversary $\mathcal{A}$ *for attacking* π

- Utilities are defined in the ideal world as payoffs of explicit "breaks" of $\mathcal{F}$
- zero-sum game (i.e., $u_D := - u_A$)

Bitcoin in RPD++

The Bitcoin Attack Game

(n-party) task as an ideal functionality ~~$\mathcal{F}$~~ $\mathcal{F}_{\text{Ledger}}$

1

Protocol Designer

u_D

2

(n-party) protocol π for $\mathcal{F}$

Adversary $\mathcal{A}$ *for attacking* π

Protocol Attacker

u_A

2

- Utilities are defined in the ideal world as payoffs of explicit "breaks" of $\mathcal{F}$
- zero-sum game (i.e., $u_D := - u_A$)

Bitcoin in RPD++

The Bitcoin Attack Game

(n-party) task as an ideal functionality ~~$\mathcal{F}$~~ $\mathcal{F}_{\text{Ledger}}$

1

Protocol Designer

u_D

2

(n-party) protocol π for $\mathcal{F}$

Protocol Attacker

u_A

2

Adversary $\mathcal{A}$ *for attacking* π

- Utilities are defined in the ideal world as payoffs of explicit "breaks" of $\mathcal{F}$
- ~~zero-sum game~~ (i.e., $u_D := -u_A$)

Our Contributions

- A new model for rational analysis of Bitcoin

Blockchains

- Applying the framework to analyze the Bitcoin backbone

  - A class of utilities reflecting “minimal” assumptions about the Bitcoin miners’ incentives.

  - Deriving predictions that match the observable.

The Utilities

Parameters

(+)	**(+)**	**(-)**
Block Reward	Transaction Fees	Hashing Cost
$\boldsymbol{BR}$ (Unit : BTC)	$\boldsymbol{TF}$ (Unit : BTC)	$\boldsymbol{HC}$ (Unit : KWh)

Utility = expected rewards - expected costs

The Utilities

Parameters

(+)	**(+)**	**(-)**
Block Reward	Transaction Fees	Hashing Cost
$\boldsymbol{BR}$ (Unit : BTC)	$\boldsymbol{TF}$ (Unit : BTC)	$\boldsymbol{HC}$ (Unit : KWh)

Utility = expected rewards - expected costs

1KWh = $\boldsymbol{CR}$ BTC

The Utilities

Parameters

(+)	**(+)**	**(-)**
Block Reward BR (Unit : BTC)	Transaction Fees TF (Unit : BTC)	Hashing Cost HC (Unit : KWh)

Utility = expected rewards - expected costs

1KWh = CR BTC

The Utilities

Parameters

(+)	**(+)**	**(-)**
Block Reward BR (Unit : BTC)	Transaction Fees TF (Unit : BTC)	Hashing Cost HC (Unit : KWh)

Utility = expected rewards - expected costs

1KWh = CR BTC

The attacker's (expected) utility $u_A^{\ddot{\mathbb{B}}}$: Wants to make profit

- For each block a corrupted inserts into the state: $(BR + TF)$ BTCs
- For each hash query a corrupted makes: $-(HC \times CR)$ BTCs

The Utilities

Parameters

(+)	**(+)**	**(-)**
Block Reward BR (Unit : BTC)	Transaction Fees TF (Unit : BTC)	Hashing Cost HC (Unit : KWh)

Utility = expected rewards - expected costs

1KWh = CR BTC

The attacker's (expected) utility $u_A^{\ddot{\mathbb{B}}}$: Wants to make profit

- For each block a corrupted inserts into the state: $(BR + TF)$ BTCs
- For each hash query a corrupted makes: $-(HC \times CR)$ BTCs

Can be defined in the ideal experiment
(explicit in the functionality)

The Utilities

Parameters

(+)	**(+)**	**(-)**
Block Reward BR (Unit : BTC)	Transaction Fees TF (Unit : BTC)	Hashing Cost HC (Unit : KWh)

Utility = expected rewards - expected costs

1KWh = CR BTC

The designer's (expected) utility $u_D^{\ddot{\mathbb{B}}}$: Wants to preserve consensus *and* make profit while doing so

- For each block an honest inserts into the state: $(BR + TF)$ BTCs
- For each hash query an honest makes: **-** $(HC \times CR)$ BTCs
- If the state (permanent part) of the ledger forks then **- exp** BTCs

Bitcoin in RPD++

Advantages over standard rational analysis

- **Simpler (Stackelberg) game to analyze**
  - 2-party 2-move metagame among unbounded agents
- **Most Bitcoin miners will not cheat and will follow the protocol if it is profitable for them**
- **Utilities are defined in the cleaner ideal world**
  - Can define them based on the fixed ledger state rather than local views of parties
- **Automatic composition with crypto** [GKMTZ13]
- **Easily captures adaptive corruption**
  - Example: bribery attacks [Bon16]

Our Contributions

- A new model for rational analysis of Bitcoin

Blockchains

- Applying the framework to analyze the Bitcoin backbone
  - A class of utilities reflecting “minimal” assumptions about the Bitcoin miners’ incentives.
  - Deriving predictions that match the observable.

Stability/Security: No Transaction Fees (TF=0)

Bitcoin is strongly $(u_D^{\ddot{\mathbb{B}}}, u_A^{\ddot{\mathbb{B}}})$-attack-payoff secure for $\mathcal{F}_{\text{Ledger}}$

Stability/Security: No Transaction Fees (TF=0)

Bitcoin is strongly $(u_D^{\ddot{\mathbb{B}}}, u_A^{\ddot{\mathbb{B}}})$-attack-payoff secure for $\mathcal{F}_{\text{Ledger}}$

Recall: This is the semi-honest network-rushing adversary

Stability/Security: No Transaction Fees (TF=0)

Bitcoin is strongly $(u_D^{\ddot{\mathbb{B}}}, u_A^{\ddot{\mathbb{B}}})$-attack-payoff secure for $\mathcal{F}_{\text{Ledger}}$

Recall: This is the semi-honest network-rushing adversary

Proof Idea:

- The adversary controls the network

- Any non-network related attack involves hashing
  - If the finds a solution to the puzzle he is better off pushing it to the network
  - Otherwise, the hash is useless (and costly)

Stability/Security: No Transaction Fees (TF=0)

Bitcoin is strongly $(u_D^{\ddot{\mathbb{B}}}, u_A^{\ddot{\mathbb{B}}})$-attack-payoff secure for $\mathcal{F}_{\text{Ledger}}$

Recall: This is the semi-honest network-rushing adversary

Proof Idea:

- The adversary controls the network
- Any non-network related attack involves hashing
  - If the finds a solution to the puzzle he is better off pushing it to the network
  - Otherwise, the hash is useless (and costly)

Fixed difficulty

Stability/Security: No Transaction Fees (TF=0)

Bitcoin is $(u_D^{\text{฿}}, u_A^{\text{฿}}, (\mathbb{A}, \mathbb{\Pi}))$ -incentive-compatible for $\mathcal{F}_{\text{Ledger}}$ if and only if CR is "high enough"

Stability/Security: No Transaction Fees (TF=0)

Bitcoin is $(u_D^{\text{฿}}, u_A^{\text{฿}}, (\mathbb{A}, \mathbb{\Pi}))$ -incentive-compatible for $\mathcal{F}_{\text{Ledger}}$ if and only if $\boldsymbol{CR}$ is "high enough"

$$\boldsymbol{BR} \cdot \boldsymbol{CR} < \frac{\boldsymbol{HC}}{p}$$

$$\boldsymbol{BR} \cdot \boldsymbol{CR} > \boldsymbol{HC} \cdot \frac{1}{p \cdot (1-p)^{n-1}}$$

* p = Probability of finding a valid block in 1 hash query

Stability/Security: No Transaction Fees (TF=0)

Bitcoin is $(u_D^{\text{฿}}, u_A^{\text{฿}}, (\mathbb{A}, \mathbb{\Pi}))$ -incentive-compatible for $\mathcal{F}_{\text{Ledger}}$ if and only if $\boldsymbol{CR}$ is "high enough"

$$\boldsymbol{BR} \cdot \boldsymbol{CR} < \frac{\boldsymbol{HC}}{p}$$

$$\boldsymbol{BR} \cdot \boldsymbol{CR} > \boldsymbol{HC} \cdot \frac{1}{p \cdot (1-p)^{n-1}}$$

Proof Idea:
On expectation, the cost of mining till you find a block is more than the profit (even if the block would make it)

* p = Probability of finding a valid block in 1 hash query

Stability/Security: No Transaction Fees (TF=0)

Bitcoin is $(u_D^{\text{฿}}, u_A^{\text{฿}}, (\mathbb{A}, \mathbb{\Pi}))$ -incentive-compatible for $\mathcal{F}_{\text{Ledger}}$ if and only if $\boldsymbol{CR}$ is "high enough"

$$\boldsymbol{BR} \cdot \boldsymbol{CR} < \frac{\boldsymbol{HC}}{p}$$ 

$$\boldsymbol{BR} \cdot \boldsymbol{CR} > \boldsymbol{HC} \cdot \frac{1}{p \cdot (1-p)^{n-1}}$$ 

Proof Idea:
On expectation, the cost of mining till you are the only one that finds a block is less than the profit.

* p = Probability of finding a valid block in 1 hash query

Stability/Security: With Transaction Fees

Bitcoin is $(u_D^{\text{₿}}, u_A^{\text{₿}}, (\mathbb{A}, \mathbb{\Pi}))$ -incentive-compatible for $\mathcal{F}_{\text{Ledger}}$

if $\boldsymbol{CR}$ is "high enough" *and …*

Stability/Security: With Transaction Fees

Bitcoin is $(u_D^{\text{฿}}, u_A^{\text{฿}}, (\mathbb{A}, \mathbb{\Pi}))$ -incentive-compatible for $\mathcal{F}_{\text{Ledger}}$

if $\boldsymbol{CR}$ is "high enough" *and …*

- no incentive to circulate high-fee transactions to the network

Stability/Security: With Transaction Fees

Bitcoin is $(u_D^{\text{฿}}, u_A^{\text{฿}}, (\mathbb{A}, \mathbb{\Pi}))$ -incentive-compatible for $\mathcal{F}_{\text{Ledger}}$

if $\boldsymbol{CR}$ is "high enough" *and …*

- there is an upper bound on total fees
- all parties get enough transactions to reach this bound

- no incentive to circulate high-fee transactions to the network

Stability/Security: With Transaction Fees

Bitcoin is $(u_D^{\text{฿}}, u_A^{\text{฿}}, (\mathbb{A}, \mathbb{\Pi}))$ -incentive-compatible for $\mathcal{F}_{\text{Ledger}}$ if $\boldsymbol{CR}$ is "high enough" *and …*

- there is an upper bound on total fees
- all parties get enough transactions to reach this bound

Proposal for when rewards approach zero

- no incentive to circulate high-fee transactions to the network

Conclusions

Our Results

- Simple and Crypto-compatible rational model for blockchains
- Rational treatment of the **Bitcoin backbone** with **fixed difficulty** under natural minimal utilities
- The effect of exchange on stability/security
- Proposal for coping with diminishing rewards
- *Also in the paper:* Rationality as a fallback to honest majority

Conclusions

Our Results

- Simple and Crypto-compatible rational model for blockchains
- Rational treatment of the **Bitcoin backbone** with **fixed difficulty** under natural minimal utilities
- The effect of exchange on stability/security
- Proposal for coping with diminishing rewards
- *Also in the paper:* Rationality as a fallback to honest majority

Future Directions

- Variable difficulty
- Utilities capturing other factors might affect the decision:
    - Detection of a 50% attack might be a deterrence
    - Mining pools' incentives
- A rational analysis of Bitcoin **as cryptocurrency**
    - The contents of transactions might affect the utilities…

Conclusions

Our Results

- Simple and Crypto-compatible rational model for blockchains
- Rational treatment of the **Bitcoin backbone** with **fixed difficulty** under natural minimal utilities
- The effect of exchange on stability/security
- Proposal for coping with diminishing rewards
- *Also in the paper:* Rationality as a fallback to honest majority

Future Directions

- Variable
- Utilities
  - D
  - M
- A rational analysis of Bitcoin **as cryptocurrency**
  - The contents of transactions might affect the utilities…

Thank you!