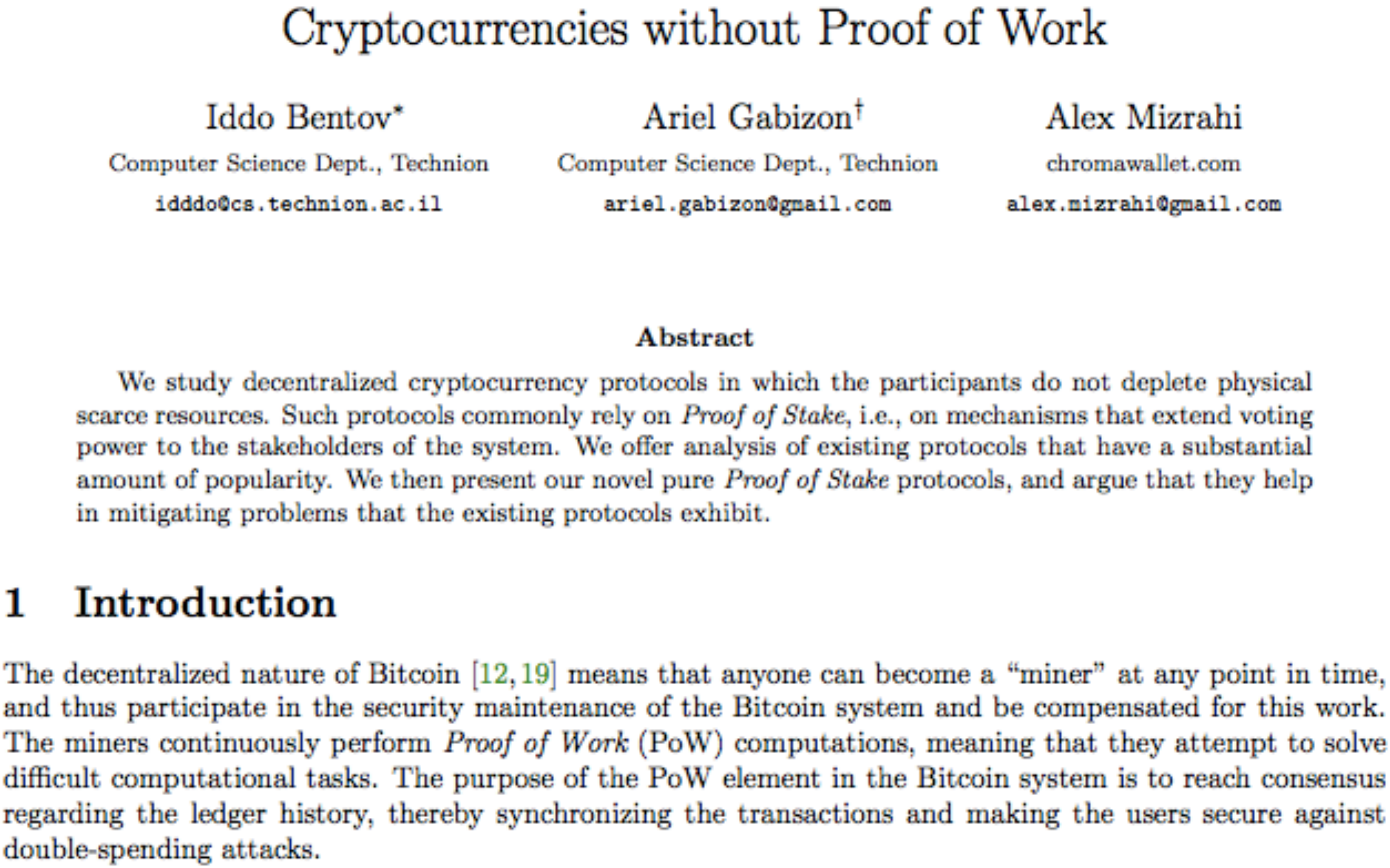
Proof of Stake 2

Murat Osmanoglu

Chains of Activity



Cryptocurrencies without Proof of Work

Iddo Bentov*
Computer Science Dept., Technion
idddo@cs.technion.ac.il

Ariel Gabizon†
Computer Science Dept., Technion
ariel.gabizon@gmail.com

Alex Mizrahi
chromawallet.com
alex.mizrahi@gmail.com

Abstract

We study decentralized cryptocurrency protocols in which the participants do not deplete physical scarce resources. Such protocols commonly rely on *Proof of Stake*, i.e., on mechanisms that extend voting power to the stakeholders of the system. We offer analysis of existing protocols that have a substantial amount of popularity. We then present our novel pure *Proof of Stake* protocols, and argue that they help in mitigating problems that the existing protocols exhibit.

1 Introduction

The decentralized nature of Bitcoin [12, 19] means that anyone can become a "miner" at any point in time, and thus participate in the security maintenance of the Bitcoin system and be compensated for this work. The miners continuously perform *Proof of Work* (PoW) computations, meaning that they attempt to solve difficult computational tasks. The purpose of the PoW element in the Bitcoin system is to reach consensus regarding the ledger history, thereby synchronizing the transactions and making the users secure against double-spending attacks.

Chains of Activity

- a pure Proof of Stake protocol that aims to prevent the rational forks by which the only a single stakeholder identity can create the next block

Chains of Activity

- a pure Proof of Stake protocol that aims to prevent the rational forks by which the only a single stakeholder identity can create the next block

- a party who possesses p fraction of the total amount of coins in circulation will be the one who creates the next block with the probability p

Chains of Activity

- a pure Proof of Stake protocol that aims to prevent the rational forks by which the only a single stakeholder identity can create the next block

- a party who possesses p fraction of the total amount of coins in circulation will be the one who creates the next block with the probability p

- there are two difficulties associated with pure Proof of Stake system:

Chains of Activity

- a pure Proof of Stake protocol that aims to prevent the rational forks by which the only a single stakeholder identity can create the next block

- a party who possesses p fraction of the total amount of coins in circulation will be the one who creates the next block with the probability p

- there are two difficulties associated with pure Proof of Stake system:
  - Fair initial distribution of the money supply to the parties
  - Network fragility if the nodes are rational rather than altruistic

Chains of Activity

- a pure Proof of Stake protocol that aims to prevent the rational forks by which the only a single stakeholder identity can create the next block

- a party who possesses p fraction of the total amount of coins in circulation will be the one who creates the next block with the probability p

- there are two difficulties associated with pure Proof of Stake system:

  - Fair initial distribution of the money supply to the parties

  PoW solves this hurdle by converting physical resources into coins

  - Network fragility if the nodes are rational rather than altruistic

Chains of Activity

- time is divided into sequence of segments, called epoch

Chains of Activity

- time is divided into sequence of segments, called epoch

- each epoch is divided into L discrete unites, called slot

Chains of Activity

- time is divided into sequence of segments, called epoch
- each epoch is divided into L discrete unites, called slot
- each slot is associated with a single block that is generated by a single stakeholder

Chains of Activity

- the identity of this stakeholder is fixed and publicly known. He collects transactions that are broadcasted over the network, then creates a block

Chains of Activity

- the identity of this stakeholder is fixed and publicly known. He collects transactions that are broadcasted over the network, then creates a block

- The leaders of the current epoch will form a seed as $S^L = \mathrm{comb}(b_1,\ldots,b_L)$ where $b_i = \mathrm{Hash}(B_i)$

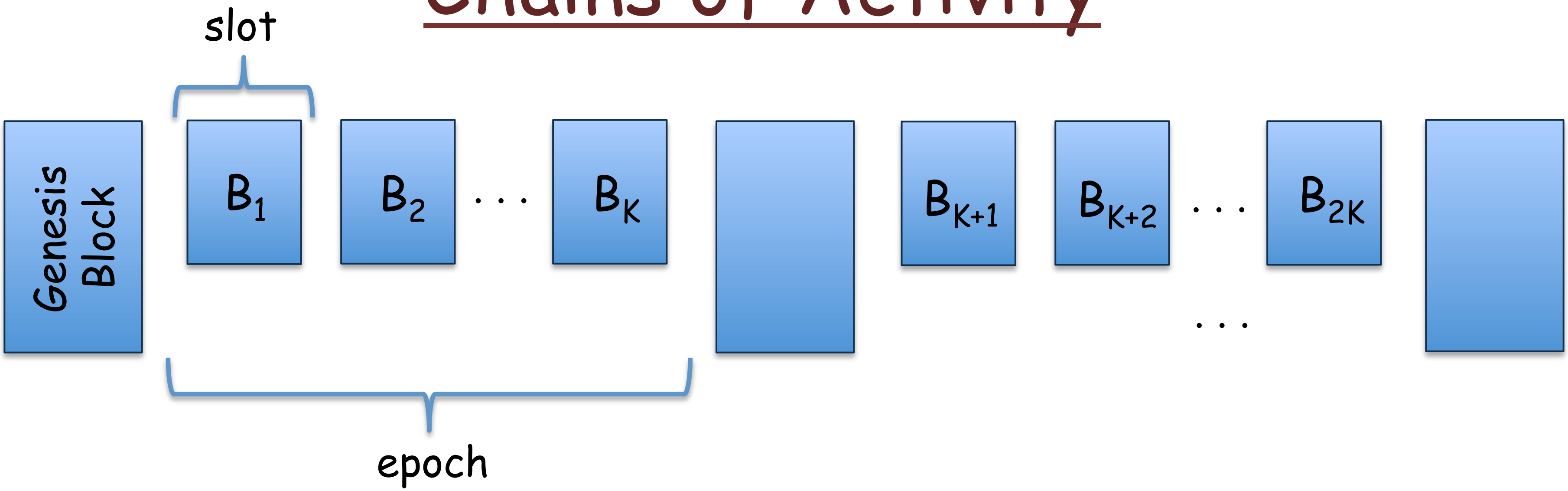
Chains of Activity

- the identity of this stakeholder is fixed and publicly known. He collects transactions that are broadcasted over the network, then creates a block

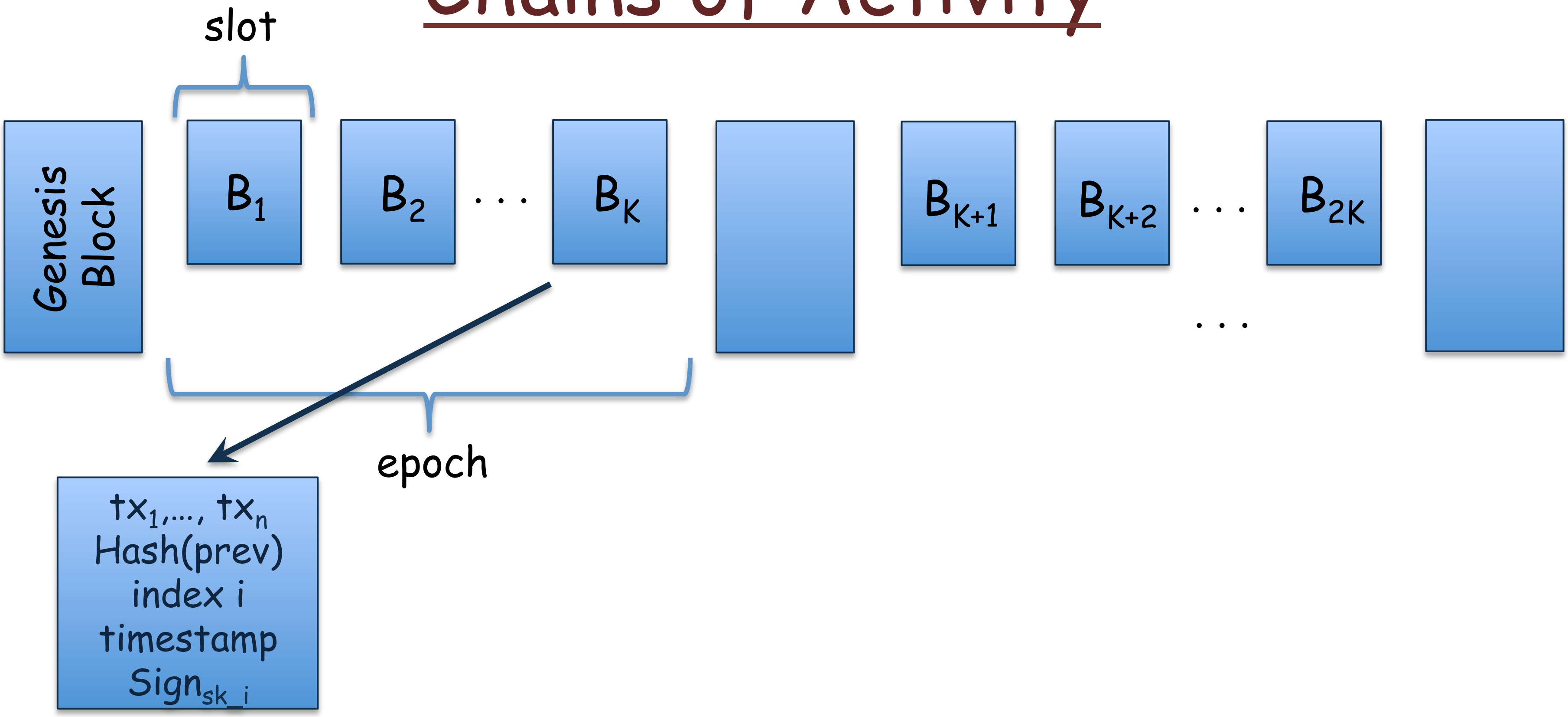
- The leaders of the current epoch will form a seed as S^L = comb(b_1,…,b_L) where b_i = Hash(B_i)

- These is then used to derive the identities of the next L stakeholders via follow-the-satoshi

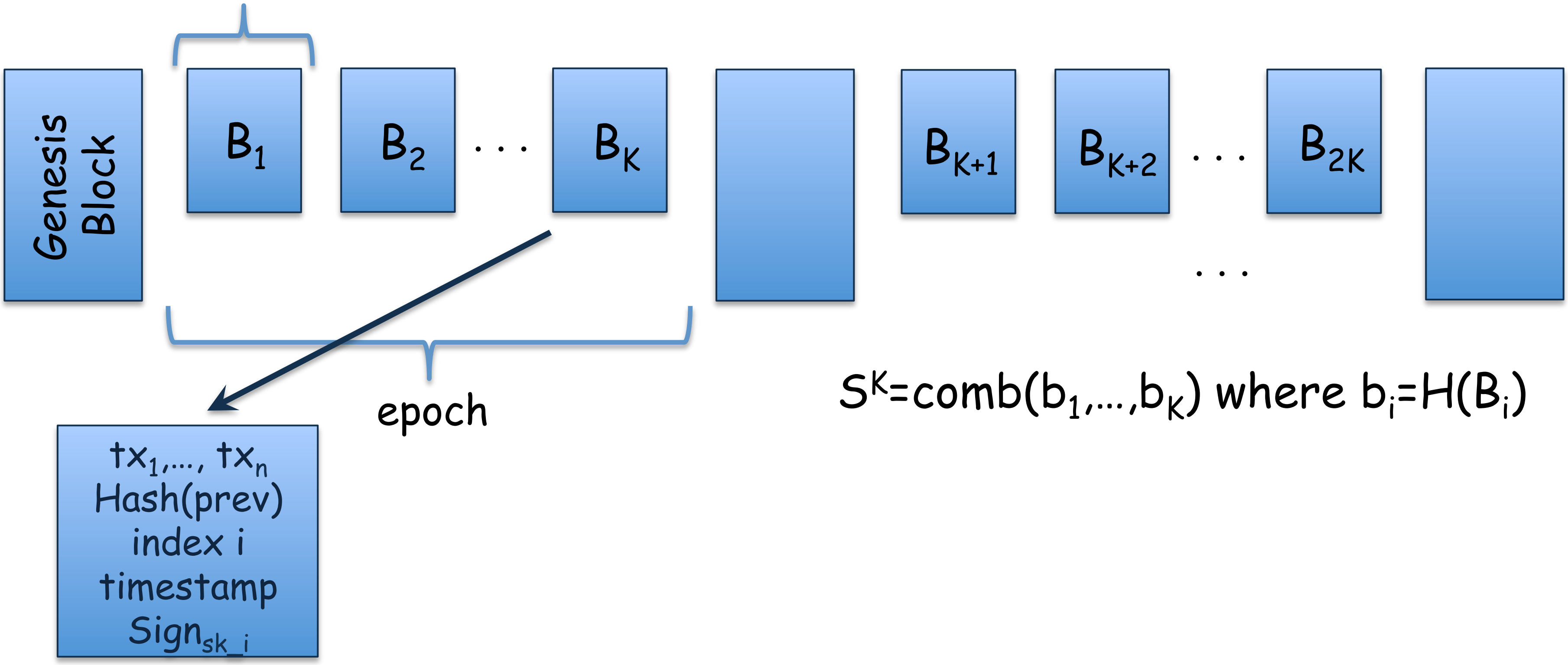
Chains of Activity



Chains of Activity

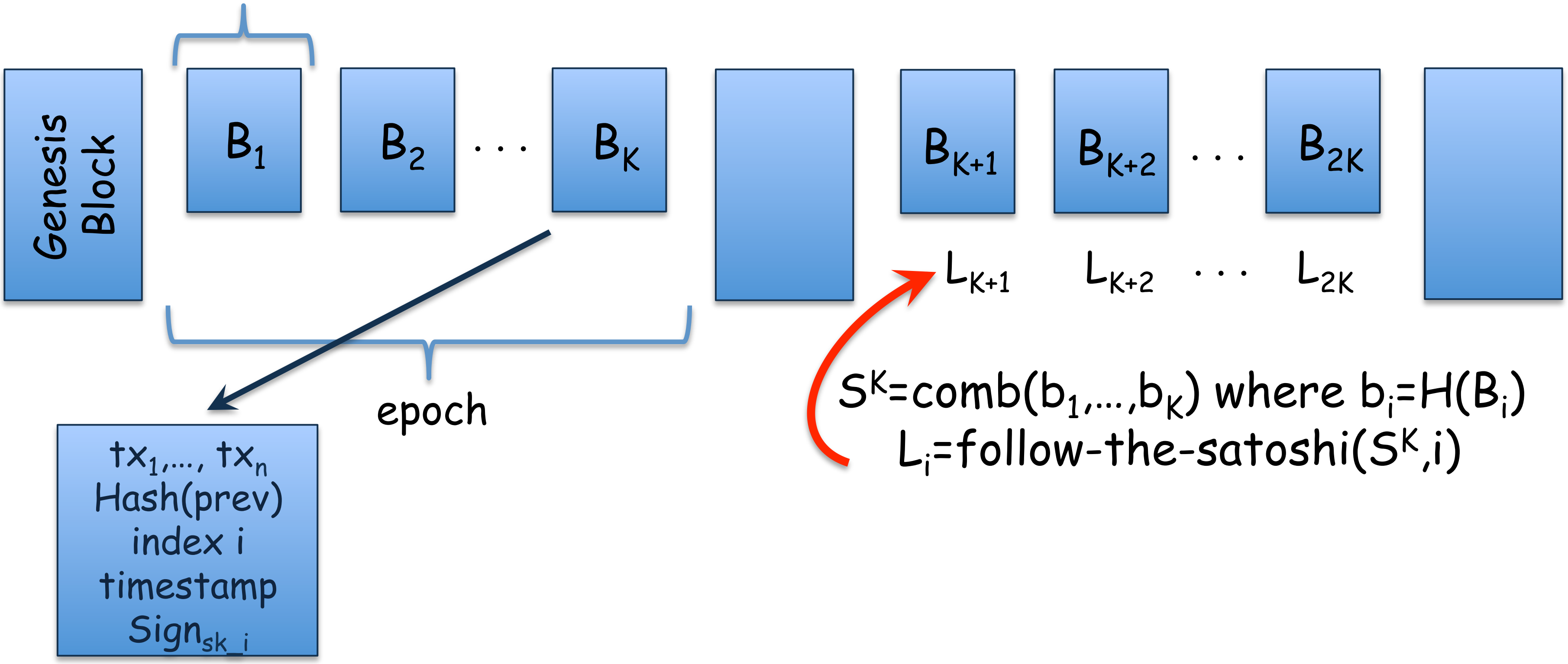


Chains of Activity



$S^K = \text{comb}(b_1,\dots,b_K)$ where $b_i = H(B_i)$

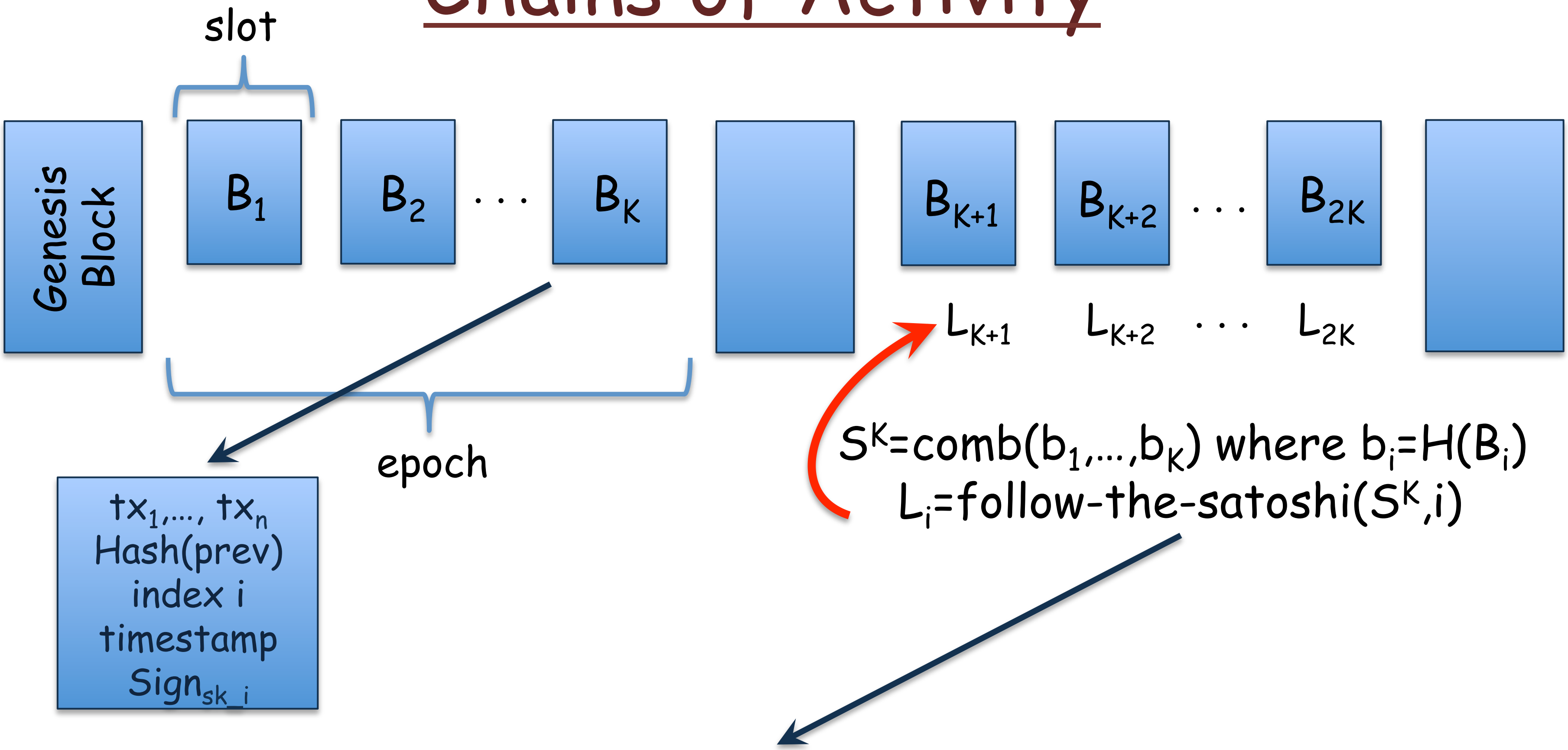
Chains of Activity



$tx_1, \ldots, tx_n$
Hash(prev)
index i
timestamp
Sign_{sk_i}

$S^K = \mathrm{comb}(b_1, \ldots, b_K)$ where $b_i = H(B_i)$
$L_i = \text{follow-the-satoshi}(S^K, i)$

Chains of Activity



$tx_1, \ldots, tx_n$
Hash(prev)
index i
timestamp
$Sign_{sk\_i}$

$$S^K = comb(b_1, \ldots, b_K) \text{ where } b_i = H(B_i)$$

$$L_i = \text{follow-the-satoshi}(S^K, i)$$

follow-the-satoshi : it takes an index of a satoshi as input, and fetches the block of ledger data in which this satoshi minted, and tracks the transactions that moved this satoshi to subsequent addresses until the last one, and outputs this address

Chains of Activity

- If K is very large, then attacker may try to gain possession of future consecutive satoshis in order to mount a double-spending attack.

Chains of Activity

- If K is very large, then attacker may try to gain possession of future consecutive satoshis in order to mount a double-spending attack.

  If K is very small, then it is easier to make coalitions to influence the future identities.

Chains of Activity

- If K is very large, then attacker may try to gain possession of future consecutive satoshis in order to mount a double-spending attack.

  If K is very small, then it is easier to make coalitions to influence the future identities.

  it is suggested that K = 459

Chains of Activity

- If K is very large, then attacker may try to gain possession of future consecutive satoshis in order to mount a double-spending attack.

  If K is very small, then it is easier to make coalitions to influence the future identities.

  it is suggested that K = 459

- The performance of the protocol : blocks get created in intervals of less than a specific value G_0 minutes.

Chains of Activity

- If K is very large, then attacker may try to gain possession of future consecutive satoshis in order to mount a double-spending attack.

  If K is very small, then it is easier to make coalitions to influence the future identities.

  it is suggested that K = 459

- The performance of the protocol : blocks get created in intervals of less than a specific value G_0 minutes.

  Each eligible stakeholder would wish to earn fees by collecting transactions nearly until the next G_0 tick

Chains of Activity

- If K is very large, then attacker may try to gain possession of future consecutive satoshis in order to mount a double-spending attack.

  If K is very small, then it is easier to make coalitions to influence the future identities.

  it is suggested that K = 459

- The performance of the protocol : blocks get created in intervals of less than a specific value G_0 minutes.

  Each eligible stakeholder would wish to earn fees by collecting transactions nearly until the next G_0 tick

  This value avoids the risk that the next stakeholder will extend an earlier block

  it is suggested to fix it as G_0 = 5 min

Chains of Activity

- Three strikes rule : an output txout_A was eligible to create a block 3 consecutive times but the stakeholder didn’t show up, then txout_A becomes blacklisted for the future blocks

Chains of Activity

- Three strikes rule : an output txout_A was eligible to create a block 3 consecutive times but the stakeholder didn’t show up, then txout_A becomes blacklisted for the future blocks

  if follow-the-satoshi chooses txout_A again, honest nodes will skip that particular block (won’t accept it)

Chains of Activity

- Three strikes rule : an output txout_A was eligible to create a block 3 consecutive times but the stakeholder didn't show up, then txout_A becomes blacklisted for the future blocks

  if follow-the-satoshi chooses txout_A again, honest nodes will skip that particular block (won't accept it)

  if it is spent via regular tx, the satoshis of txout_A are no longer blacklisted

Chains of Activity

- Three strikes rule : an output txout_A was eligible to create a block 3 consecutive times but the stakeholder didn't show up, then txout_A becomes blacklisted for the future blocks

  if follow-the-satoshi chooses txout_A again, honest nodes will skip that particular block (won't accept it)

  if it is spent via regular tx, the satoshis of txout_A are no longer blacklisted

- Stakeholders may wish to collude and skip the last several blocks as they did not exist, and extend the blockchain from an earlier block, and gain the fees that went to previous stakeholders

Chains of Activity

- Three strikes rule : an output txout_A was eligible to create a block 3 consecutive times but the stakeholder didn't show up, then txout_A becomes blacklisted for the future blocks

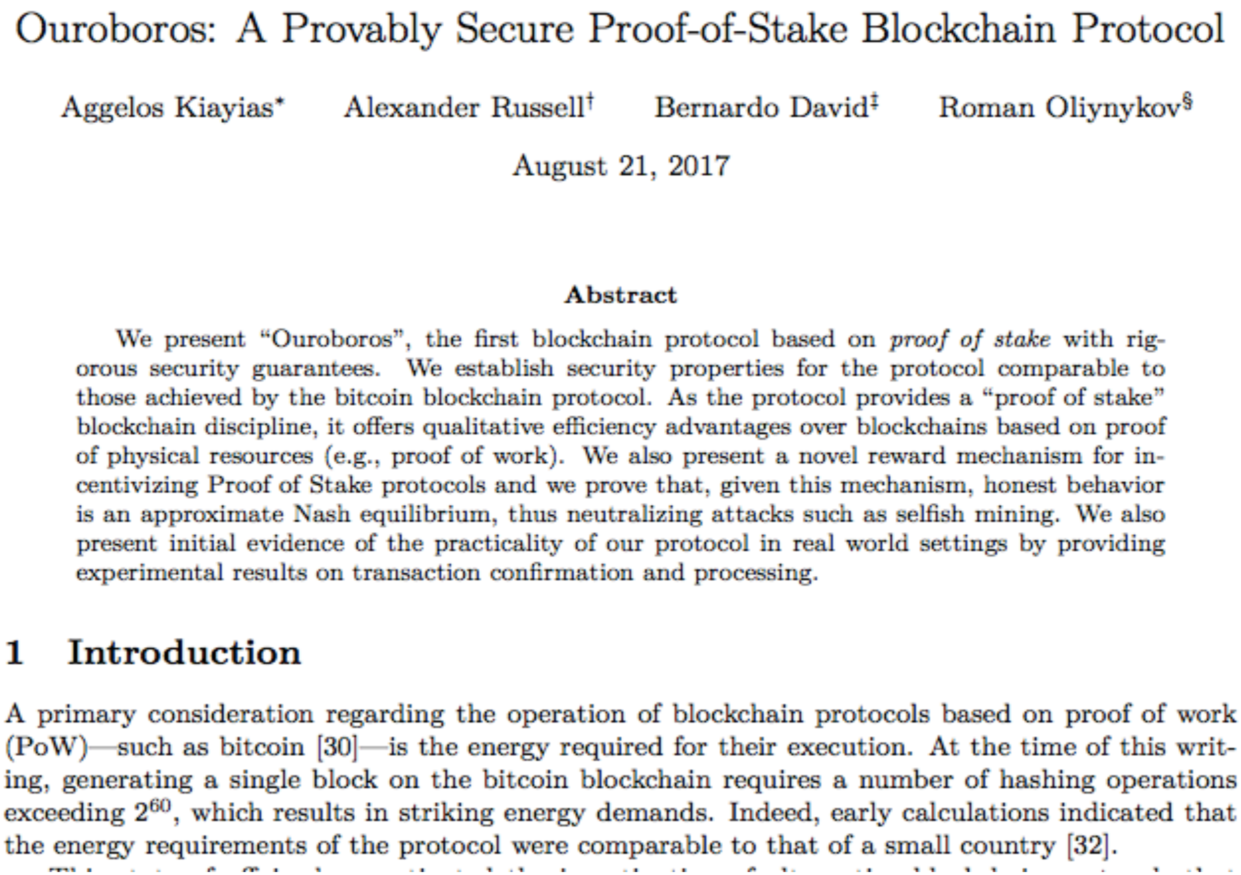
  if follow-the-satoshi chooses txout_A again, honest nodes will skip that particular block (won't accept it)

  if it is spent via regular tx, the satoshis of txout_A are no longer blacklisted

- Stakeholders may wish to collude and skip the last several blocks as they did not exist, and extend the blockchain from an earlier block, and gain the fees that went to previous stakeholders

  it can be avoided by including in each transaction the index of the latest block that the user who made this transaction is aware of

Ouroboros



Ouroboros: A Provably Secure Proof-of-Stake Blockchain Protocol

Aggelos Kiayias* Alexander Russell[†] Bernardo David[‡] Roman Oliynykov[§]

August 21, 2017

Abstract

We present "Ouroboros", the first blockchain protocol based on *proof of stake* with rigorous security guarantees. We establish security properties for the protocol comparable to those achieved by the bitcoin blockchain protocol. As the protocol provides a "proof of stake" blockchain discipline, it offers qualitative efficiency advantages over blockchains based on proof of physical resources (e.g., proof of work). We also present a novel reward mechanism for incentivizing Proof of Stake protocols and we prove that, given this mechanism, honest behavior is an approximate Nash equilibrium, thus neutralizing attacks such as selfish mining. We also present initial evidence of the practicality of our protocol in real world settings by providing experimental results on transaction confirmation and processing.

1 Introduction

A primary consideration regarding the operation of blockchain protocols based on proof of work (PoW)—such as bitcoin [30]—is the energy required for their execution. At the time of this writing, generating a single block on the bitcoin blockchain requires a number of hashing operations exceeding 2^{60}, which results in striking energy demands. Indeed, early calculations indicated that the energy requirements of the protocol were comparable to that of a small country [32].

The first blockchain protocol based on Proof of Stake with rigorous security guarantees.

Ouroboros

- A fundamental problem for PoS is to simulate the leader election process.

Ouroboros

- A fundamental problem for PoS is to simulate the leader election process.

- To achieve a fair randomized election among stakeholders, entropy must be introduced

Ouroboros

- A fundamental problem for PoS is to simulate the leader election process.

- To achieve a fair randomized election among stakeholders, entropy must be introduced

- An adversary controlling a set of stakeholders may attempt to simulate the protocol execution trying different sequence of stakeholders participants so that it finds a protocol continuation that favors him

Ouroboros

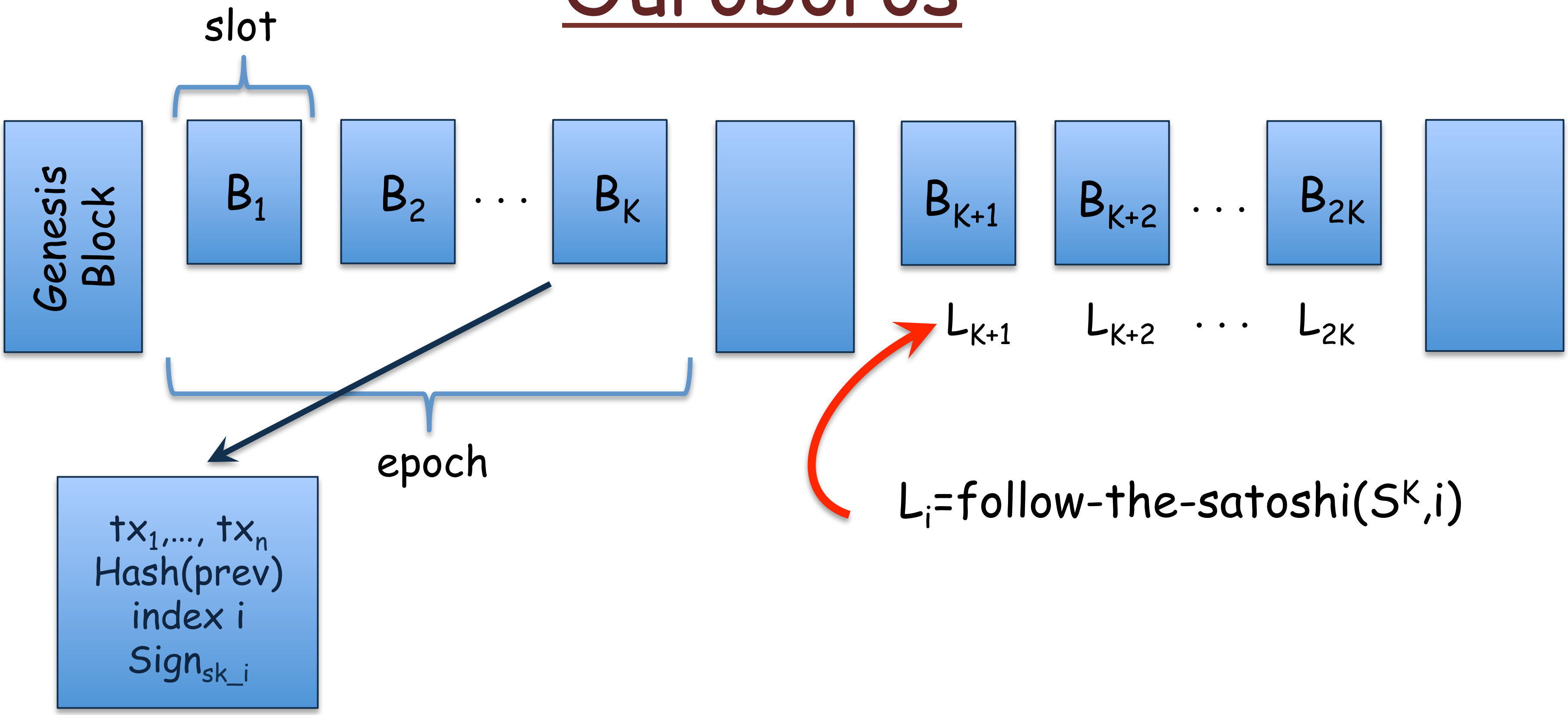
- A fundamental problem for PoS is to simulate the leader election process.

- To achieve a fair randomized election among stakeholders, entropy must be introduced

- An adversary controlling a set of stakeholders may attempt to simulate the protocol execution trying different sequence of stakeholders participants so that it finds a protocol continuation that favors him

- it is called grinding vulnerability where malicious parties may use computational resources to bias the leader election

Ouroboros



slot

Genesis Block

B_1 B_2 . . . B_K

B_{K+1} B_{K+2} . . . B_{2K}

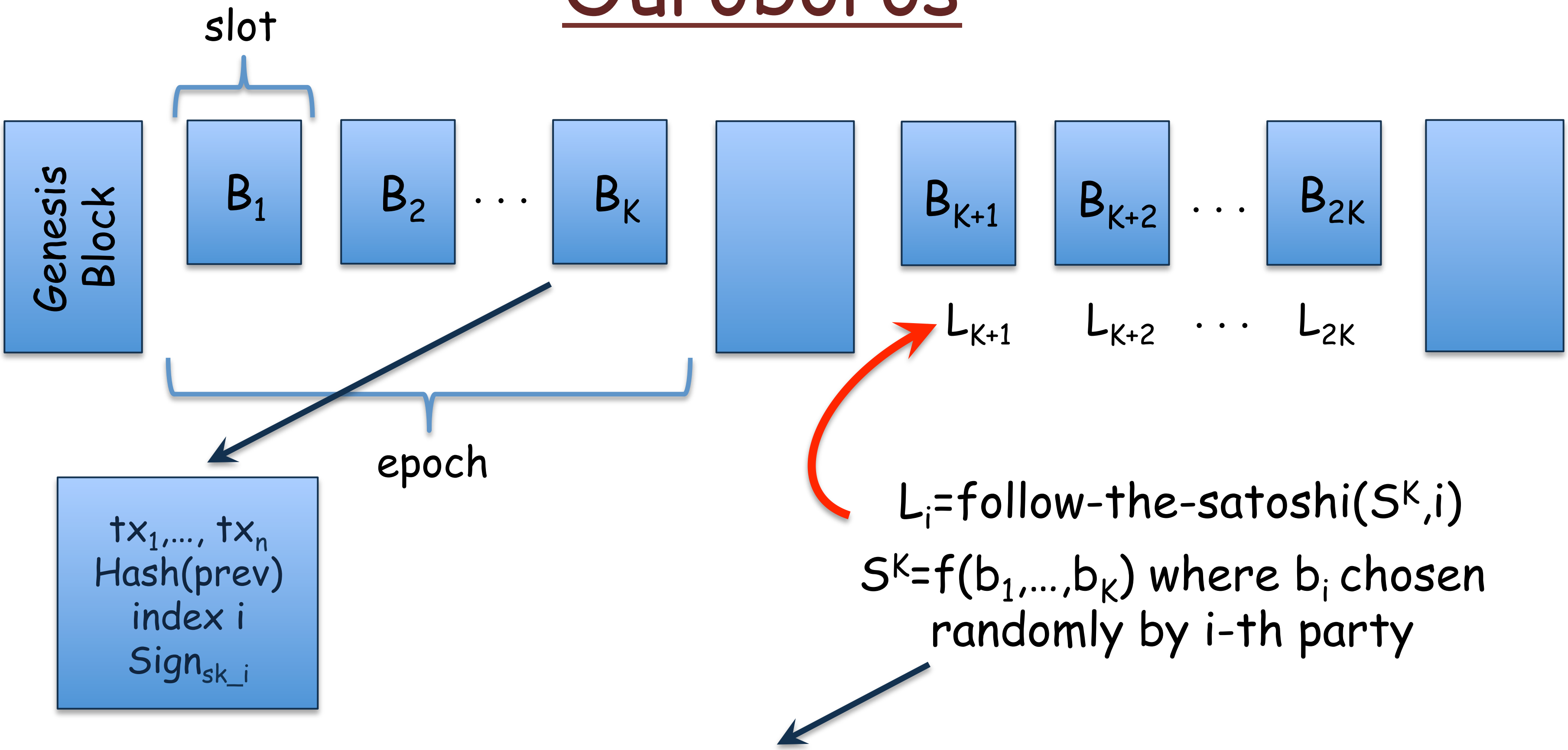
L_{K+1} L_{K+2} . . . L_{2K}

epoch

$tx_1, \ldots, tx_n$
Hash(prev)
index i
Sign_{sk_i}

L_i=follow-the-satoshi(S^K,i)

Ouroboros

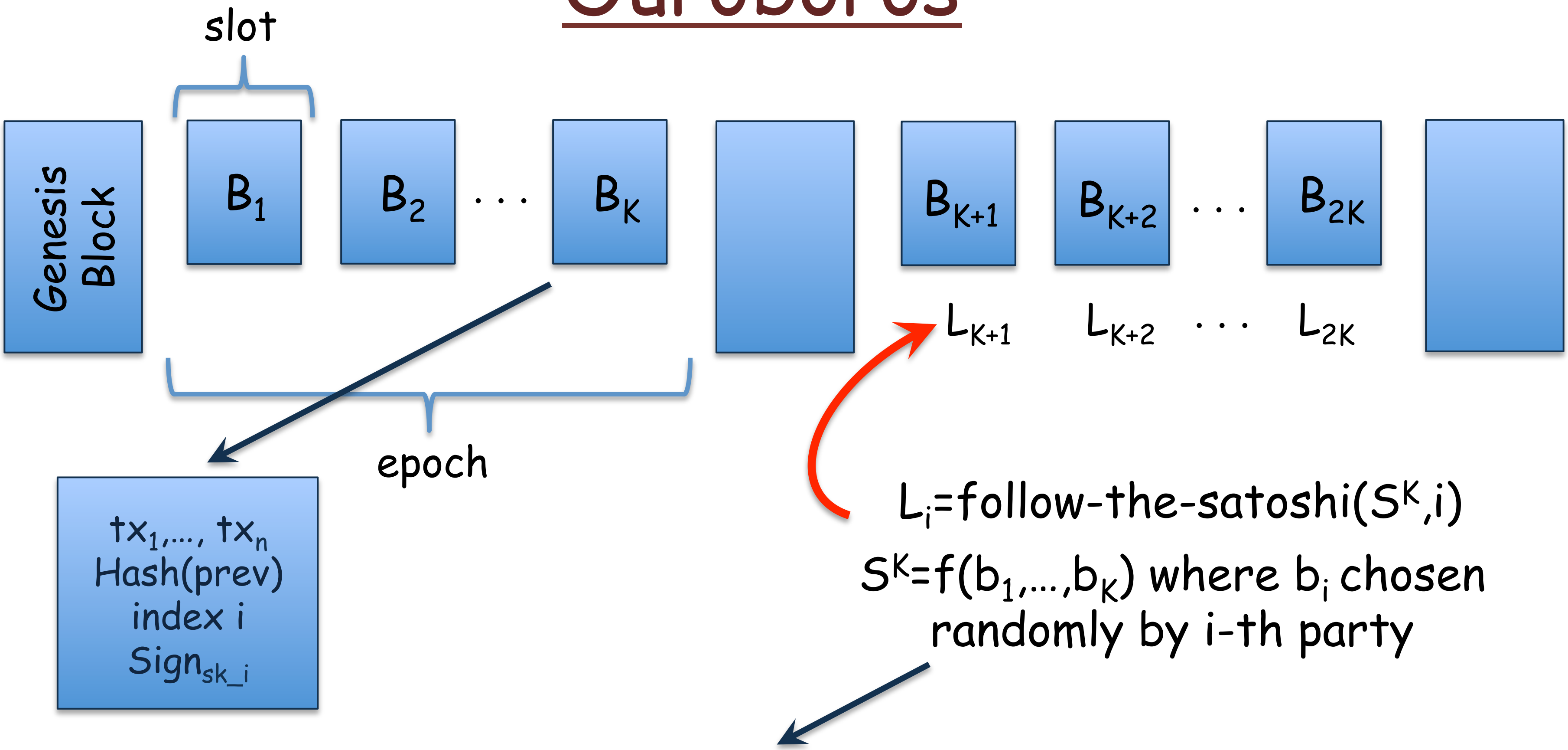


$tx_1, \ldots, tx_n$
Hash(prev)
index i
Sign_{sk_i}

L_i=follow-the-satoshi(S^K,i)

$S^K=f(b_1,\ldots,b_K)$ where b_i chosen randomly by i-th party

Ouroboros



L_i=follow-the-satoshi(S^K,i)

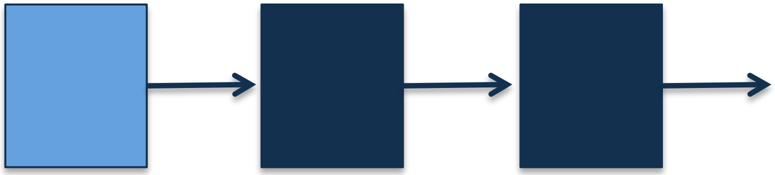
S^K=f(b_1,…,b_K) where b_i chosen randomly by i-th party

Secure Multiparty Computation: the leaders of an epoch run a secure multi-party computation to produce the randomness used to choose the leaders of the next epoch during the current epoch

PoW

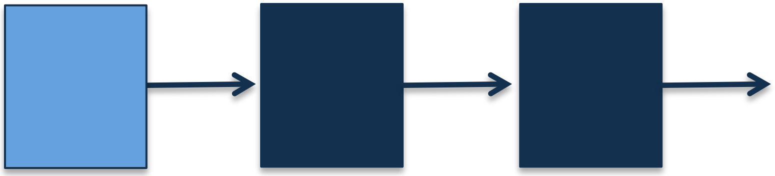
PoS

PoW



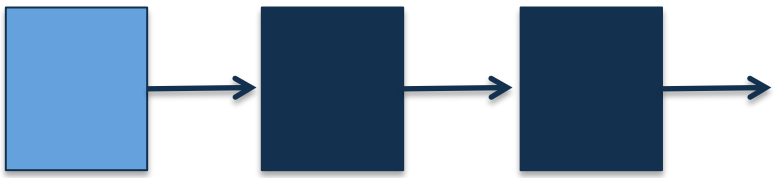
There is a genesis block which is extended by parties as they find Proof of Work

PoS



There is a genesis block which is extended by parties with Proof of Stake

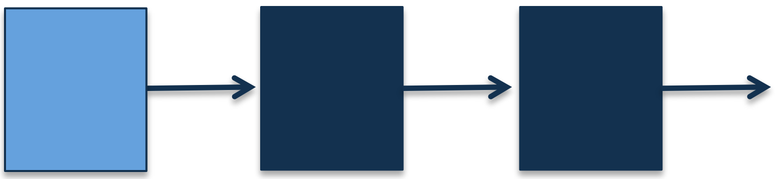
PoW



There is a genesis block which is extended by parties as they find Proof of Work

Each block that is appended to the chain, contains transactions

PoS



There is a genesis block which is extended by parties with Proof of Stake

Each block that is appended to the chain, contains transactions

PoW

There is a genesis block which is extended by parties as they find Proof of Work

Each block that is appended to the chain, contains transactions

At any given moment, one of the parties is elected to create the next block based on the hash power the parties have

PoS

There is a genesis block which is extended by parties with Proof of Stake

Each block that is appended to the chain, contains transactions

One of the stakeholder is randomly elected to create the next block based on the stake that is recorded in the blockchain

PoW

There is a genesis block which is extended by parties as they find Proof of Work

Each block that is appended to the chain, contains transactions

At any given moment, one of the parties is elected to create the next block based on the hash power the parties have

Random sampling by those trying to create the block

PoS

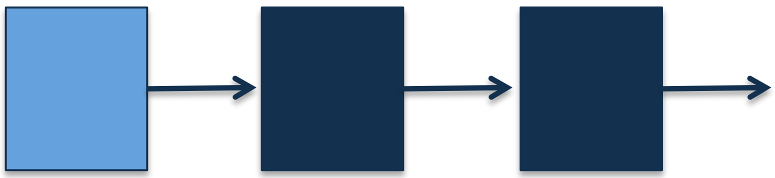
There is a genesis block which is extended by parties with Proof of Stake

Each block that is appended to the chain, contains transactions

One of the stakeholder is randomly elected to create the next block based on the stake that is recorded in the blockchain

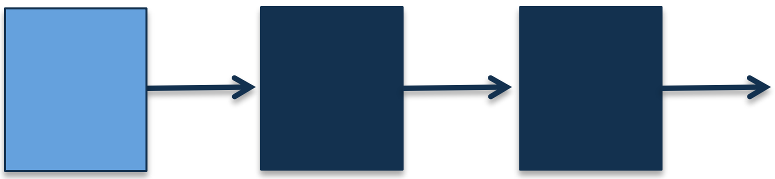
Random sampling by those trying to create the block

PoW



The genesis block, as a requirement, have to be provided to the parties as a point of reference

PoS



The genesis block, as a requirement, have to be provided to the parties as a point of reference

PoW

The genesis block, as a requirement, have to be provided to the parties as a point of reference

This point of reference specifies the initial block and setting the difficulty level

PoS

The genesis block, as a requirement, have to be provided to the parties as a point of reference

This point of reference specifies the initial block and the initial stakeholder distribution

PoW

The genesis block, as a requirement, have to be provided to the parties as a point of reference

This point of reference specifies the initial block and setting the difficulty level

the genesis block should know something about the status the world the blockchain is initiated

It should hit the right level of difficulty that reflects how many parties will be creating blocks when the blockchain starts

PoS

The genesis block, as a requirement, have to be provided to the parties as a point of reference

This point of reference specifies the initial block and the initial stakeholder distribution

The initial stakeholder distribution should be coded into the genesis block

PoW

The genesis block, as a requirement, have to be provided to the parties as a point of reference

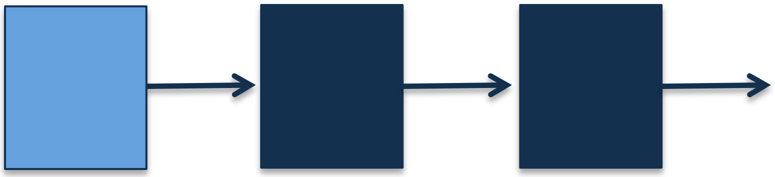
This point of reference specifies the initial block and setting the difficulty level

the genesis block should know something about the status the world the blockchain is initiated

It should hit the right level of difficulty that reflects how many parties will be creating blocks when the blockchain starts

To work in proper way, the protocol assumes the honest majority of hashing power

PoS



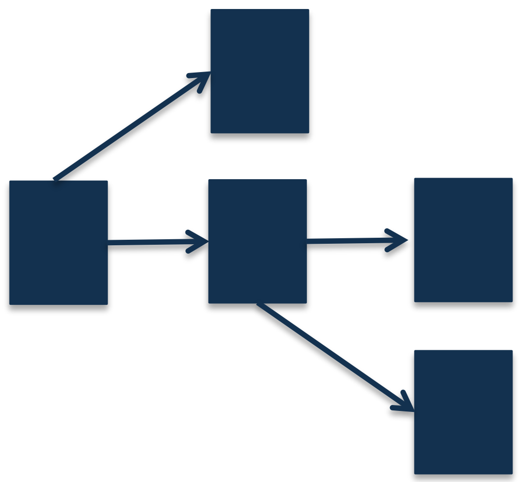
The genesis block, as a requirement, have to be provided to the parties as a point of reference

This point of reference specifies the initial block and the initial stakeholder distribution

The initial stakeholder distribution should be coded into the genesis block

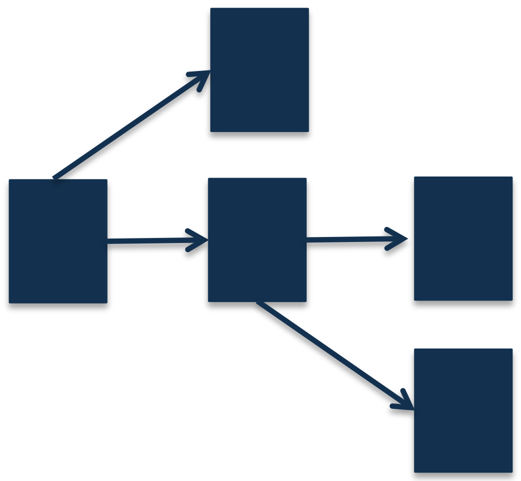
To work in proper way, the protocol assumes the honest majority of stake

PoW



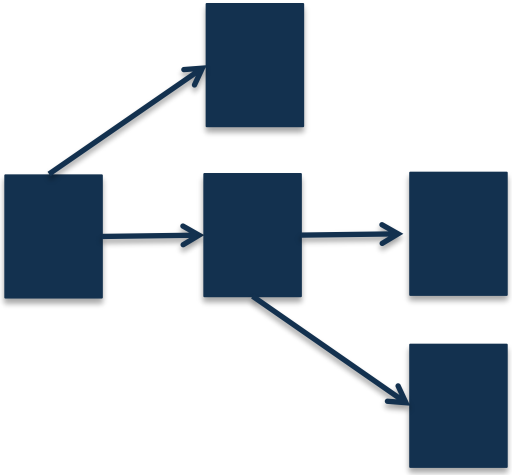
Multiple blockchains can coexist since they don't run the protocol in a coordinated way

PoS



Similar to PoW, Multiple blockchains can coexist

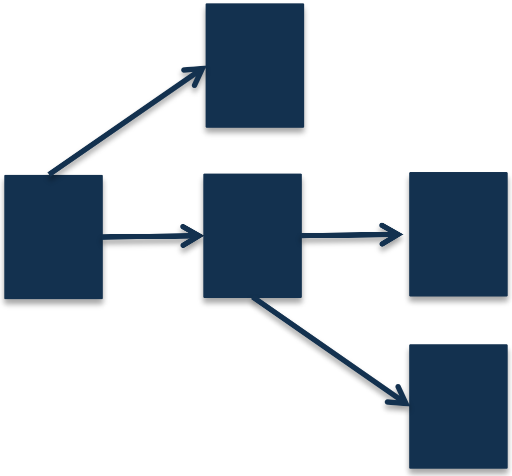
PoW



Multiple blockchains can coexist since they don't run the protocol in a coordinated way

The protocol follows the chain that has the biggest amount of difficulty

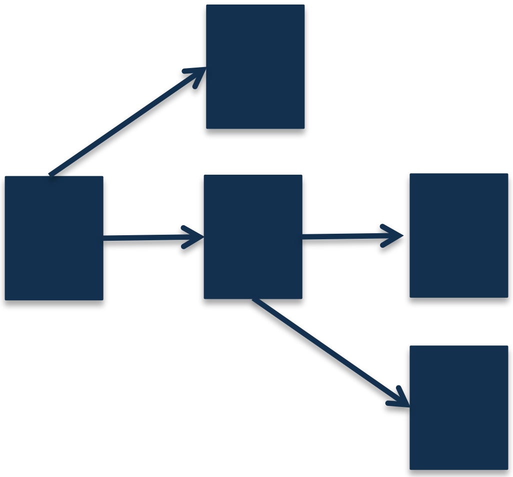
PoS



Similar to PoW, Multiple blockchains can coexist

The protocol follows the chain that has the biggest amount of stake

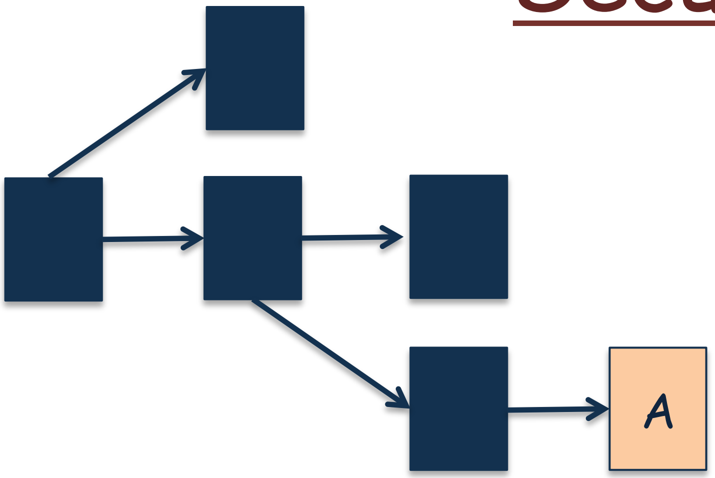
Security of PoW



Multiple blockchains can coexist since they don't run the protocol in a coordinated way

The protocol follows the chain that has the biggest amount of difficulty

Security of PoW

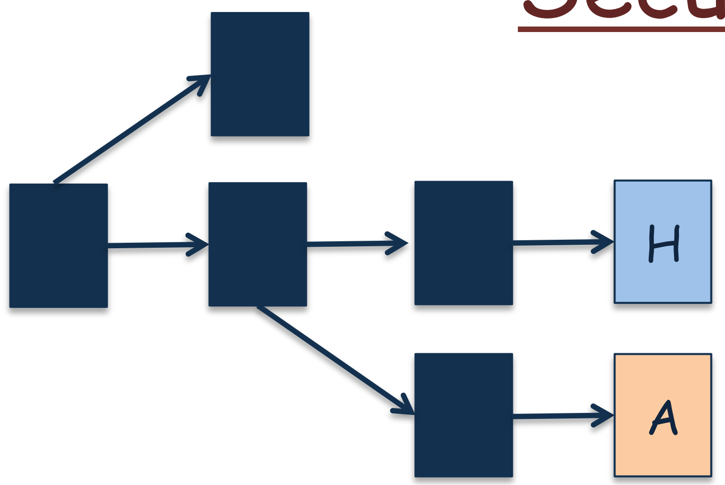


Multiple blockchains can coexist since they don't run the protocol in a coordinated way

The protocol follows the chain that has the biggest amount of difficulty

Since solving PoW is moderately hard problem, there is a moment that the block is produced uniquely by a single honest party

Security of PoW



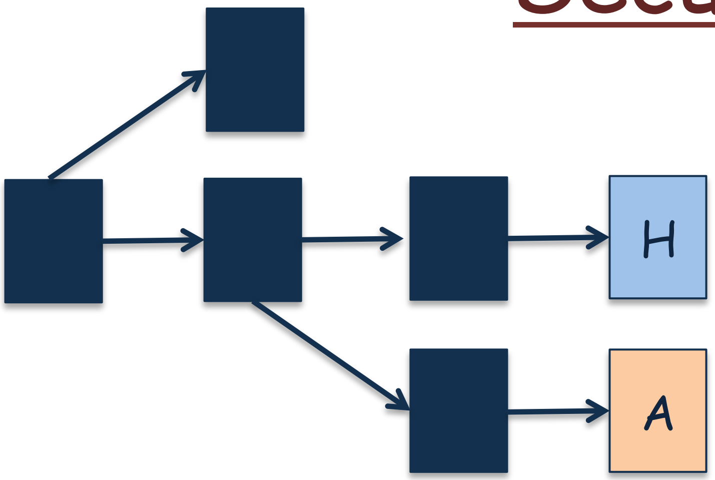
Multiple blockchains can coexist since they don't run the protocol in a coordinated way

The protocol follows the chain that has the biggest amount of difficulty

Since solving PoW is moderately hard problem, there is a moment that the block is produced uniquely by a single honest party

Since there is only one such block at that moment, this block will be adopted by all other honest parties unless the adv issues another block and splits the honest parties

Security of PoW



Multiple blockchains can coexist since they don't run the protocol in a coordinated way

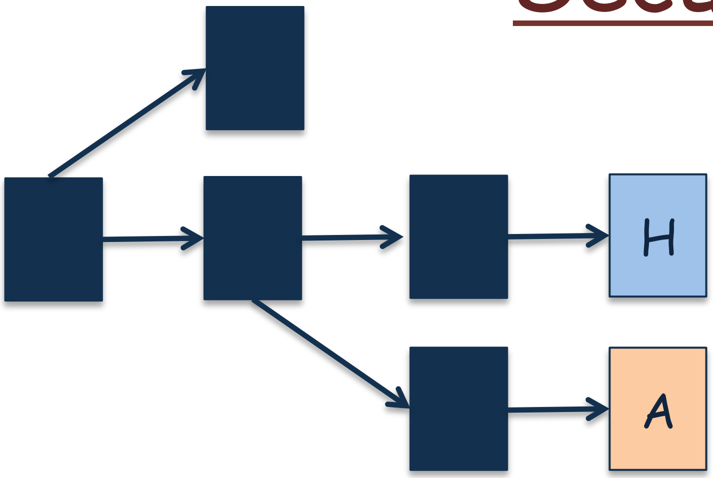
The protocol follows the chain that has the biggest amount of difficulty

Since solving PoW is moderately hard problem, there is a moment that the block is produced uniquely by a single honest party

Since there is only one such block at that moment, this block will be adopted by all other honest parties unless the adv issues another block and splits the honest parties

The rate of uniquely successful round should be bigger than the rate of blocks produced by the adv.

Security of PoW



Multiple blockchains can coexist since they don't run the protocol in a coordinated way

The protocol follows the chain that has the biggest amount of difficulty

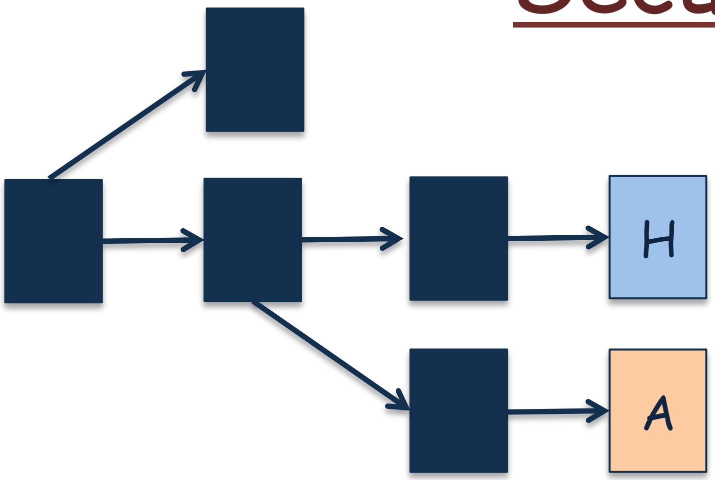
Since solving PoW is moderately hard problem, there is a moment that the block is produced uniquely by a single honest party

Since there is only one such block at that moment, this block will be adopted by all other honest parties unless the adv issues another block and splits the honest parties

The rate of uniquely successful round should be bigger than the rate of blocks produced by the adv.

If this is the case, the adv cannot maintain the fork

Security of PoW



Multiple blockchains can coexist since they don't run the protocol in a coordinated way

The protocol follows the chain that has the biggest amount of difficulty

Since solving PoW is moderately hard problem, there is a moment that the block is produced uniquely by a single honest party

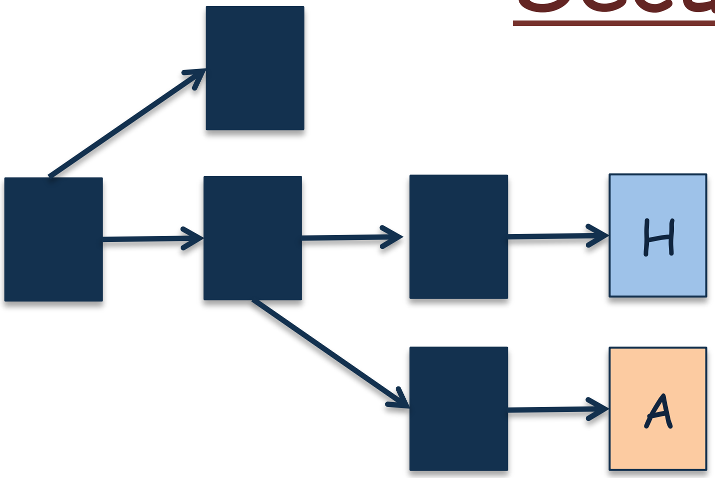
Since there is only one such block at that moment, this block will be adopted by all other honest parties unless the adv issues another block and splits the honest parties

The rate of uniquely successful round should be bigger than the rate of blocks produced by the adv.

If this is the case, the adv cannot maintain the fork

This inability of the adv implies 'persistence'

Security of PoW



Multiple blockchains can coexist since they don't run the protocol in a coordinated way

The protocol follows the chain that has the biggest amount of difficulty

Since solving PoW is moderately hard problem, there is a moment that the block is produced uniquely by a single honest party

Since there is only one such block at that moment, this block will be adopted by all other honest parties unless the adv issues another block and splits the honest parties

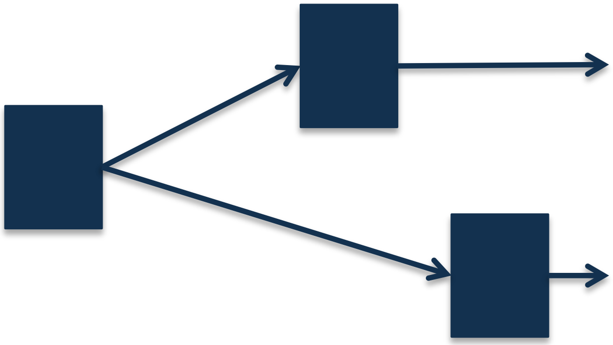
The rate of uniquely successful round should be bigger than the rate of blocks produced by the adv.

If this is the case, the adv cannot maintain the fork

This inability of the adv implies 'persistence'

In the long term, the rate of uniquely successful round overcomes the rate of the adversarial blocks

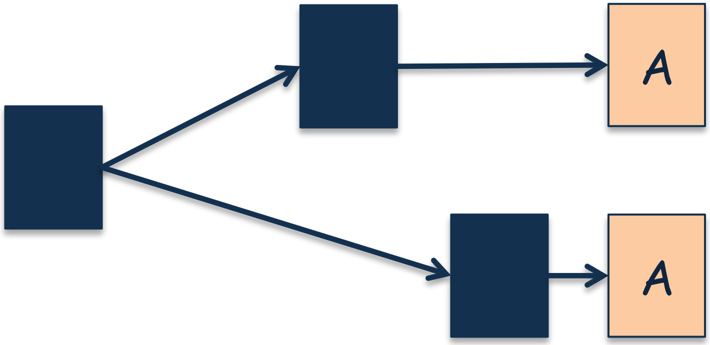
Security of PoS



Multiple blockchains can coexist since they don't run the protocol in a coordinated way

The protocol follows the chain that has the biggest amount of stake

Security of PoS

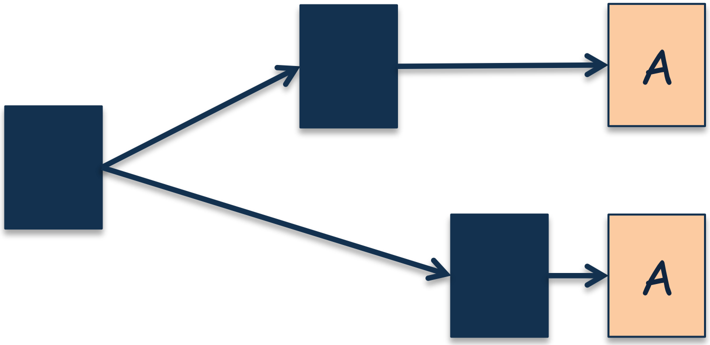


Multiple blockchains can coexist since they don’t run the protocol in a coordinated way

The protocol follows the chain that has the biggest amount of stake

The adv by being elected to issue the next block, he is capable of adding the new block to more than one chain

Security of PoS

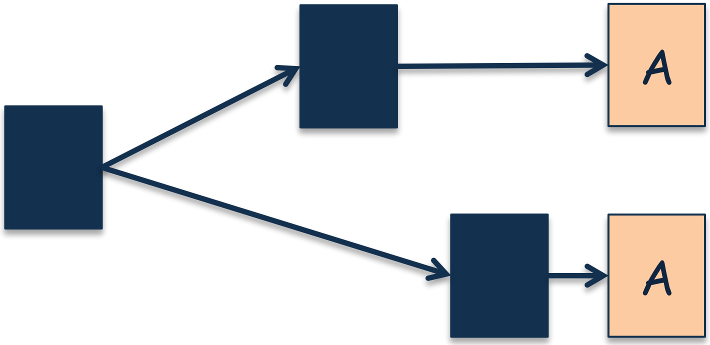


Multiple blockchains can coexist since they don't run the protocol in a coordinated way

The protocol follows the chain that has the biggest amount of stake

The adv by being elected to issue the next block, he is capable of adding the new block to more than one chain (nothing-at-stake)

Security of PoS



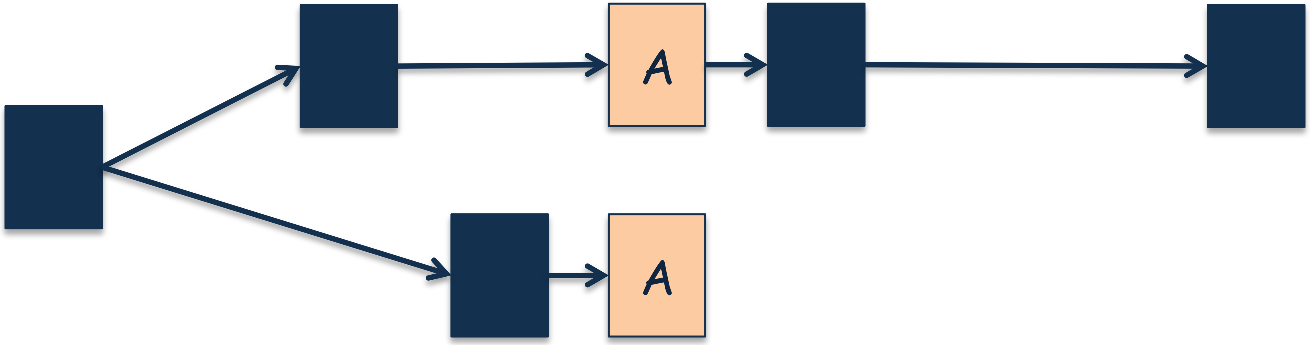
Multiple blockchains can coexist since they don't run the protocol in a coordinated way

The protocol follows the chain that has the biggest amount of stake

The adv by being elected to issue the next block, he is capable of adding the new block to more than one chain (nothing-at-stake)

So the security argument for PoW cannot be applied here

Security of PoS



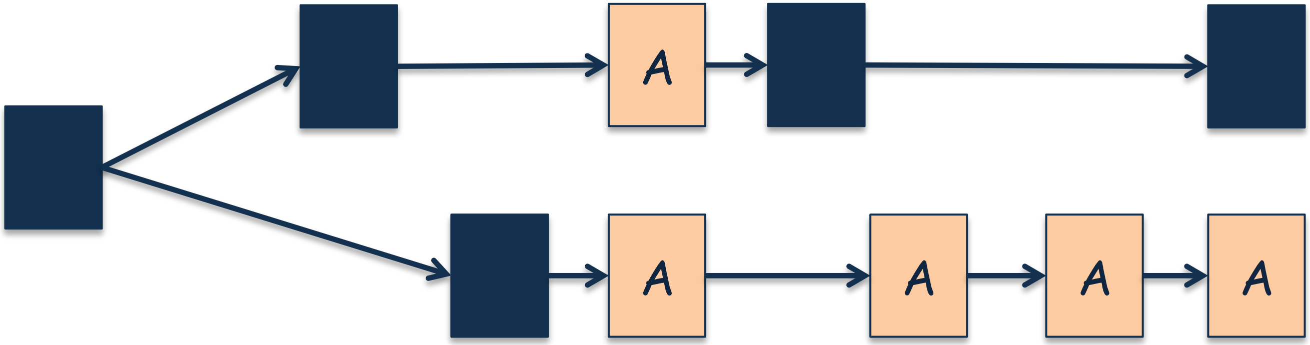
Multiple blockchains can coexist since they don't run the protocol in a coordinated way

The protocol follows the chain that has the biggest amount of stake

The adv by being elected to issue the next block, he is capable of adding the new block to more than one chain (nothing-at-stake)

So the security argument for PoW cannot be applied here

Security of PoS



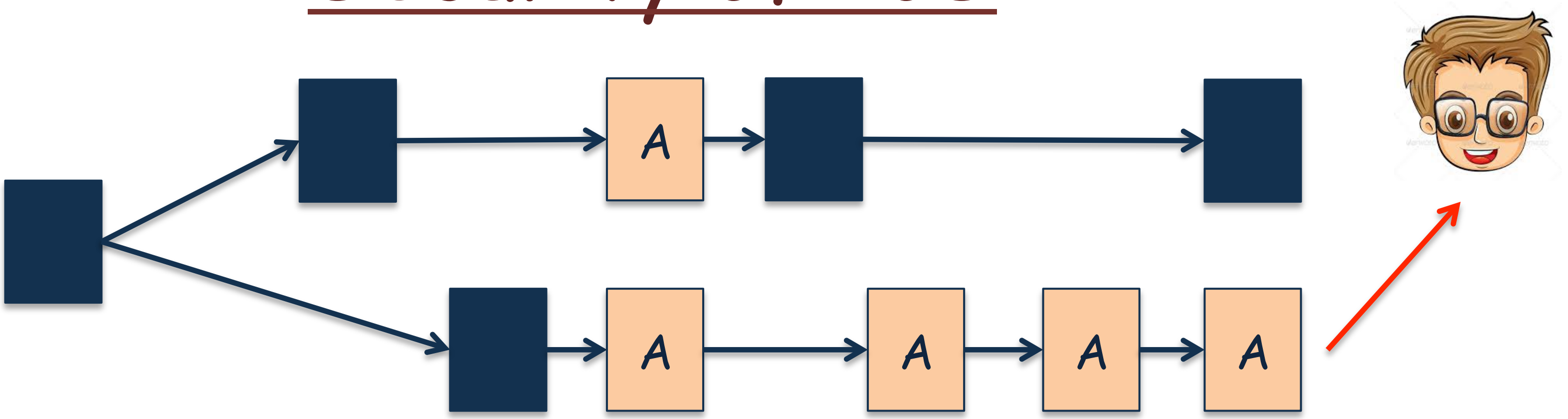
Multiple blockchains can coexist since they don’t run the protocol in a coordinated way

The protocol follows the chain that has the biggest amount of stake

The adv by being elected to issue the next block, he is capable of adding the new block to more than one chain (nothing-at-stake)

So the security argument for PoW cannot be applied here

Security of PoS



Multiple blockchains can coexist since they don't run the protocol in a coordinated way

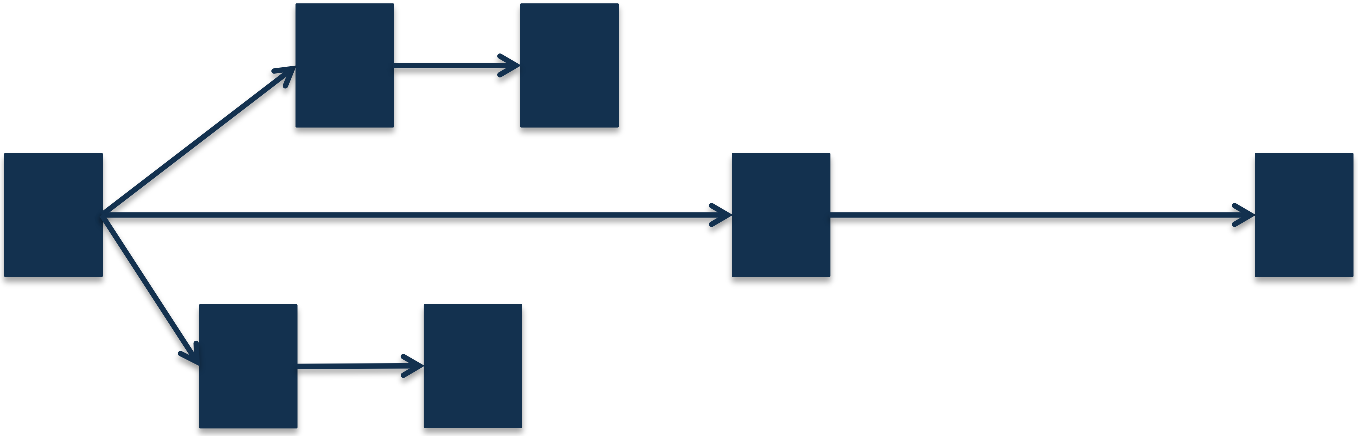
The protocol follows the chain that has the biggest amount of stake

The adv by being elected to issue the next block, he is capable of adding the new block to more than one chain (nothing-at-stake)

So the security argument for PoW cannot be applied here

This is a fork which undesired situation for the protocol

Security of PoS



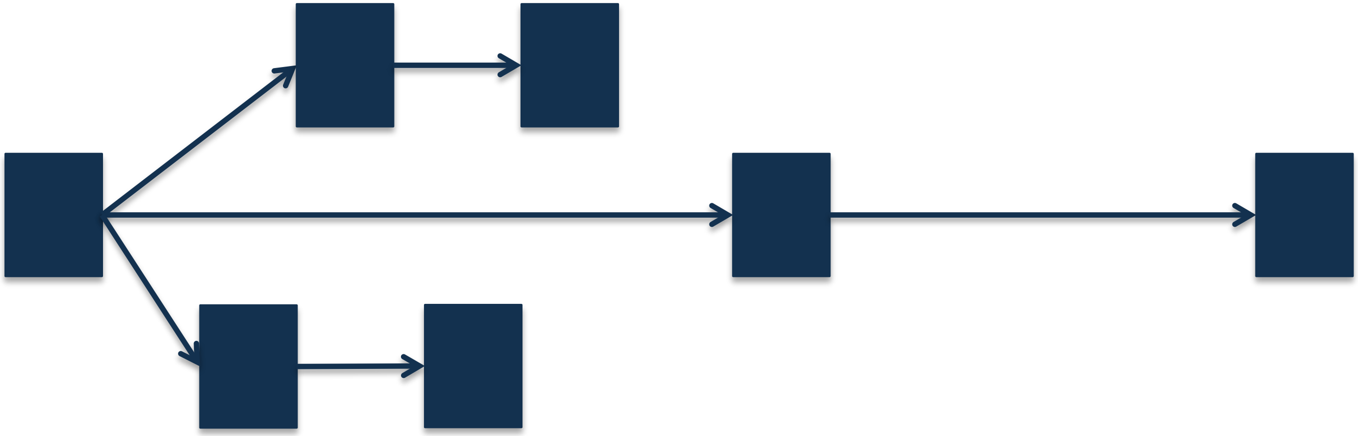
Multiple blockchains can coexist since they don't run the protocol in a coordinated way

The protocol follows the chain that has the biggest amount of stake

The adv by being elected to issue the next block, he is capable of adding the new block to more than one chain (nothing-at-stake)

So the security argument for PoW cannot be applied here

Security of PoS



Multiple blockchains can coexist since they don’t run the protocol in a coordinated way

The protocol follows the chain that has the biggest amount of stake

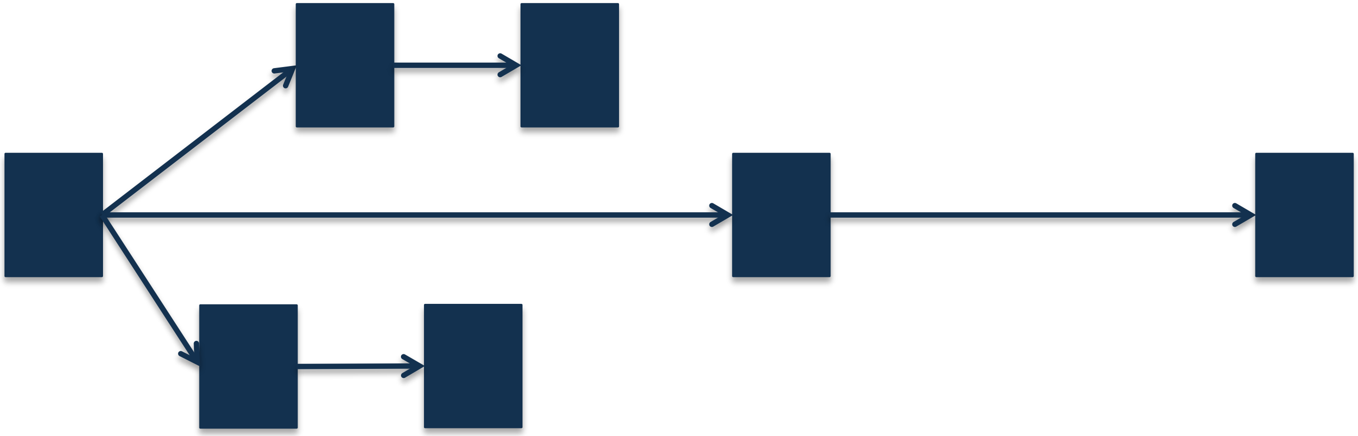
The adv by being elected to issue the next block, he is capable of adding the new block to more than one chain (nothing-at-stake)

So the security argument for PoW cannot be applied here

What we want the protocol execution has a single long chain, and any other disjoint chains are too short for the adv to be able to reach the longest one

So, the honest part adopts the longest one easily

Security of PoS



Multiple blockchains can coexist since they don't run the protocol in a coordinated way

The protocol follows the chain that has the biggest amount of stake

The adv by being elected to issue the next block, he is capable of adding the new block to more than one chain (nothing-at-stake)

So the security argument for PoW cannot be applied here

What we want the protocol execution has a single long chain, and any other disjoint chains are too short for the adv to be able to reach the longest one

So, the honest part adopts the longest one easily

Ouroboros proved that this happens almost all the time.