Bitcoin Backbone and Consesus

Nikos Leonardos
National Technical University of Athens

Bitcoin info

- Bitcoin was the first decentralized cryptocurrency with no need for a trusted central authority.
  - Previous work: Pricing functions of Dwork and Naor [1992], MicroMint of Rivest and Shamir [1996], Hashcash of Back [1997,2002], Szabo's bit gold [1998], Karma by Vishnumurthy, Chandrakumar, Sirer [2003].

- Introduced in the 2008 paper "Bitcoin: A Peer-to-Peer Electronic Cash System" by Satoshi Nakamoto (a pseudonym).

- Released as open-source code in 2009; first block: 9, Jan 2009.
  - Nowadays there are more than than 800,000 blocks.

- The total number of bitcoins will not exceed 21 million and this limit is expected to be reached around 2140.
  - Nowadays there are more than 19 million bitcoins in circulation.
  - The smallest denomination is the satoshi, equal to 10^{-8} bitcoins.

Bitcoin: a solution to two problems

- Bitcoin was the first decentralized cryptocurrency, with no need for a trusted central authority.
- Bitcoin was a fresh solution at an old, fundamental, and well-studied problem in distributed computing, the consensus problem.

Bitcoin: a solution to two problems

- Bitcoin was the first decentralized cryptocurrency, with no need for a trusted central authority.
- Bitcoin was a fresh solution at an old, fundamental, and well-studied problem in distributed computing, the consensus problem.

Formal analysis

- A formal description of the model in which the problem and its solution can be described.
- The properties that a suggested solution should satisfy.
- A formal description of the protocol.
- Proof that Bitcoin backbone indeed has the desired properties.

The model

- Synchronous model.
  - Time is discrete and divided in rounds.
  - Global clock: round number is common knowledge.
  - All messages get delivered in the next round.
- A number of honest parties n and an adversary that controls t parties.
  - Honest parties act independently.
  - Parties controlled by the adversary collaborate.
- Parties communicate by broadcasting a message.

  The adversary can:
  - inject messages into a party's incoming messages.
  - reorder a party's incoming messages.
- Anonymous setting: parties cannot associate a message to a sender; they don't even know if two messages come from the same sender.

What is not in the model

- Honest parties losing messages or becoming eclipsed or becoming unable to know the current time.
  - Parties experiencing such issues are factored into the adversary.
- The honest parties' incentives.
  - On the other hand, adversarial parties wish to inflict the worst possible damage independently of utility.
- An adversary with computational power that even on occasion, exceeds that of honest parties.
- Attacks that exploit specific weaknesses of the underlying cryptographic primitives.

  [We will use idealized versions of hash functions and digital signatures].

Hash functions

A cryptographic hash function is a deterministic algorithm

$$H : \{0,1\}^* \to \{0,1\}^\kappa$$

with the following properties.

- Preimage resistance: Given $y \in \{0,1\}^\kappa$ it should be computationally infeasible to compute x such that $H(x) = y$.
- Second-preimage resistance: Given x and $y = H(x)$ it should be computationally infeasible to compute a $x' \neq x$ such that $H(x') = y$.
- Collision resistance: It should be computationally infeasible to compute $x \neq x'$ such that $H(x) = H(x')$.

For a meaningful formal definition one considers cryptographic hash families.

Proof-of-work in the random-oracle model

A moderately hard computational task: Given a hash-function $H(\cdot)$ with range $\{0,1\}^\kappa$ and a y, find x such that $H(x,y)$ begins with a lot of zeroes. More generaly, given a target T,

- find x such that $H(x,y) < T$.

Proof-of-work in the random-oracle model

A moderately hard computational task: Given a hash-function $H(\cdot)$ with range $\{0,1\}^\kappa$ and a y, find x such that $H(x,y)$ begins with a lot of zeroes. More generaly, given a target T,

- find x such that $H(x,y) < T$.

We'll work in the "random oracle" model. That is, we assume the existence of a hash-function $H(\cdot)$ that operates as follows.

- On a query x, the returned value $H(x)$ is a random number from the range of $H(\cdot)$, unless x has been queried before in which case $H(\cdot)$ is consistent (equal to the previous returned value).

Proof-of-work in the random-oracle model

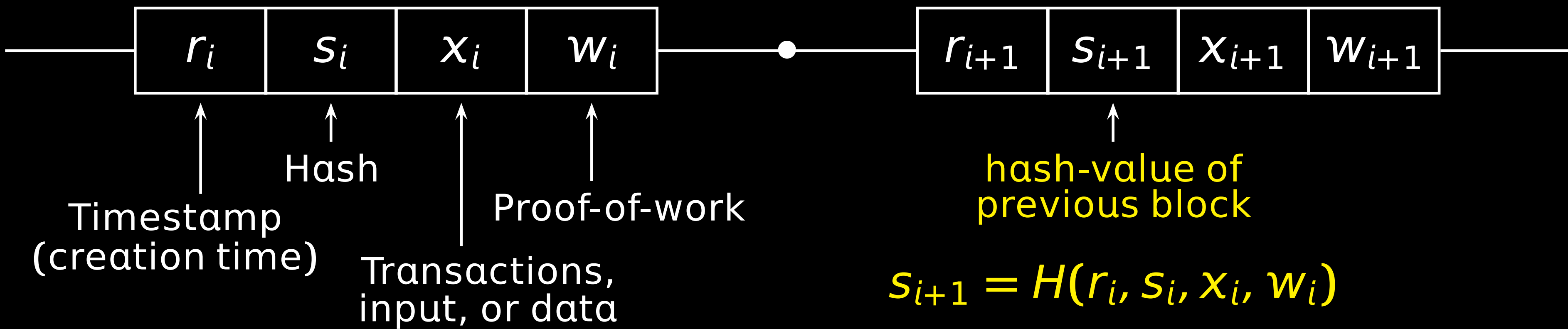
A moderately hard computational task: Given a hash-function $H(\cdot)$ with range $\{0, 1\}^\kappa$ and a y, find x such that $H(x, y)$ begins with a lot of zeroes. More generaly, given a target T,

- find x such that $H(x, y) < T$.

We'll work in the "random oracle" model. That is, we assume the existence of a hash-function $H(\cdot)$ that operates as follows.

- On a query x, the returned value $H(x)$ is a random number from the range of $H(\cdot)$, unless x has been queried before in which case $H(\cdot)$ is consistent (equal to the previous returned value).
- A query is successful with probability $\frac{T}{2^\kappa}$, and one needs in expectation $\frac{2^\kappa}{T}$ calls to the oracle $H(\cdot)$ for a proof-of-work.
- Among poly(k) queries, the probability of a collision (two distinct x and x' with $H(x) = H(x'))$ is exponentially small in κ.

Bitcoin's data structure: the blockchain



$$s_{i+1} = H(r_i, s_i, x_i, w_i)$$

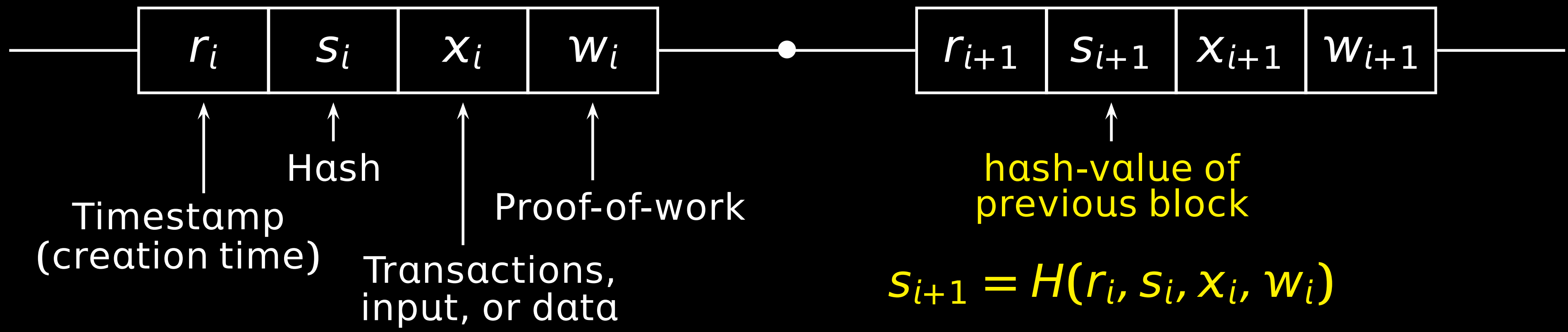
- A block (r, s, x, w) is valid if it has a small hash-value, providing a proof-of-work:

$$H(r, s, x, w) < T.$$

- A chain is valid if all its blocks provide a proof-of-work and each block extends the previous one:

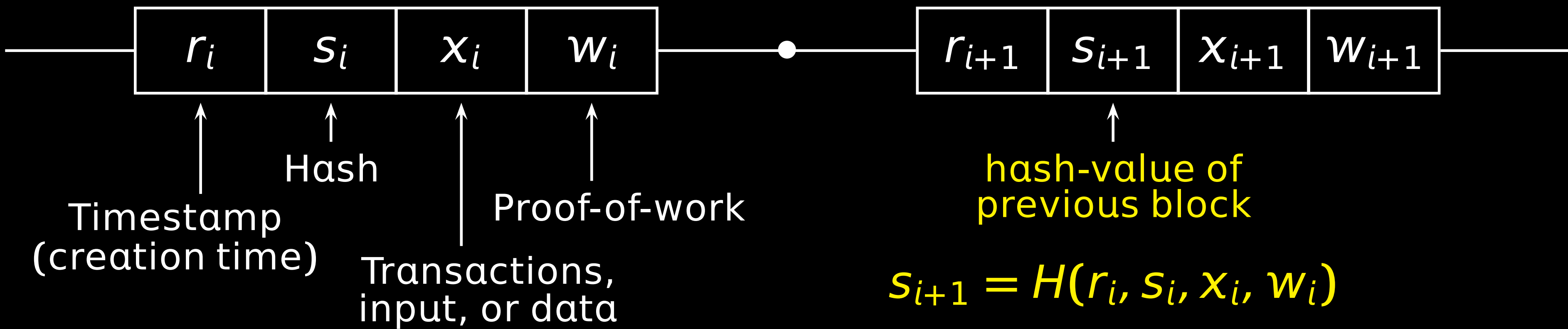
$$\text{for each } i, \quad s_{i+1} = H(r_i, s_i, x_i, w_i) \text{ and } r_{i+1} > r_i.$$

Comments on the blockchain



- To alter the contents of a block and preserve the length of the chain the adversary either has to discover a collision in $H(\cdot)$ or compute all the subsequent blocks.
  - Thus the adversary *cannot* delete, copy, inject, or predict blocks.
- By adjusting the target T we control how hard is computing a block: the lower the target the higher the difficulty, wlog $1/T$.

Transactions on the blockchain



$$s_{i+1} = H(r_i, s_i, x_i, w_i)$$

A transaction has the following form:

- "From the output (say 10BTC) of transaction i in block j (which was sent to public pk_0), send 2BTC to pk_1 and 7BTC to pk_2"--- signed with sk_0.
- Fees, coinbase transaction.
- Parties need to agree on which is the j-th block.

Bitcoin backbone: A distributed randomized algorithm

In each round r, each party with a chain C_0 performs the following:

- Receive from the network (block)chains $C_1, C_2, \ldots$
- Choose the first longest chain C among the valid ones in $\{C_0, C_1, C_2, \ldots\}$. (Order matters*.)
- Try to extend the longest chain C.

  This is modeled by a Bernoulli trial with a probability of success that depends on the target T.

  – Suppose its last block is the i-th one and equal to (r_i, s_i, x_i, w_i) with $s = H(r_i, s_i, x_i, w_i)$. Find $w \in \{1, 2, \ldots, q\}$ such that

  $$H(r, s, x, w) < T.$$

  If successful, let $C \leftarrow C \,\|\, (r, s, x, w)$.

- If $C \neq C_0$ (i.e., you computed or switched-to another (longer) chain), diffuse the new chain C.

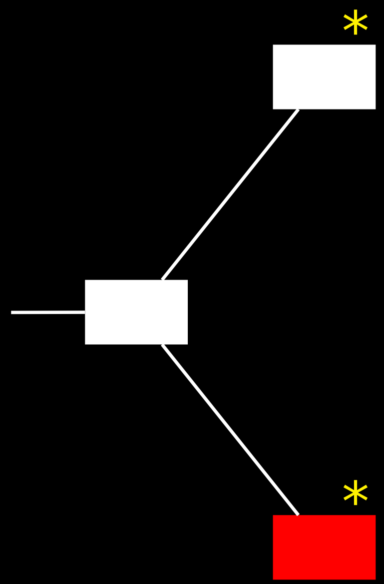
An execution example

$\emptyset$

An execution example

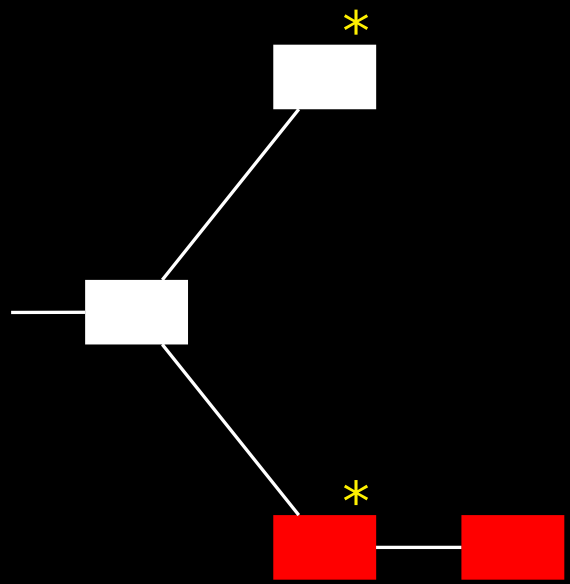
- White blocks have been computed by an honest party.
- Red blocks have been computed by the adversary.
- A star ($*$) on a block means that an honest party has the chain ending with that block at the given round.

An execution example



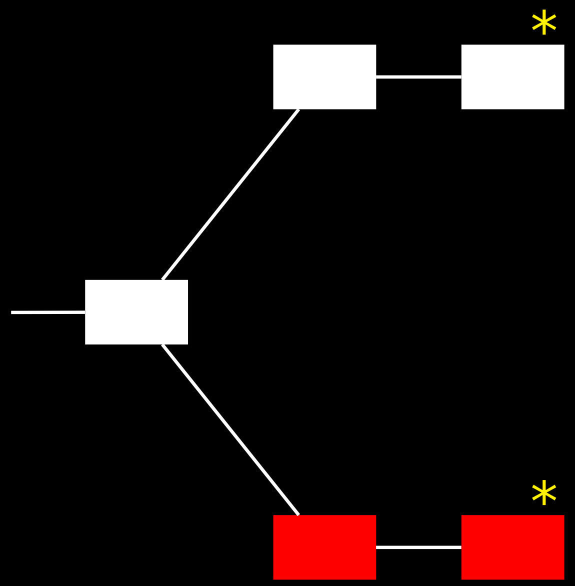
- White blocks have been computed by an honest party.
- Red blocks have been computed by the adversary.
- A star ($*$) on a block means that an honest party has the chain ending with that block at the given round.

An execution example



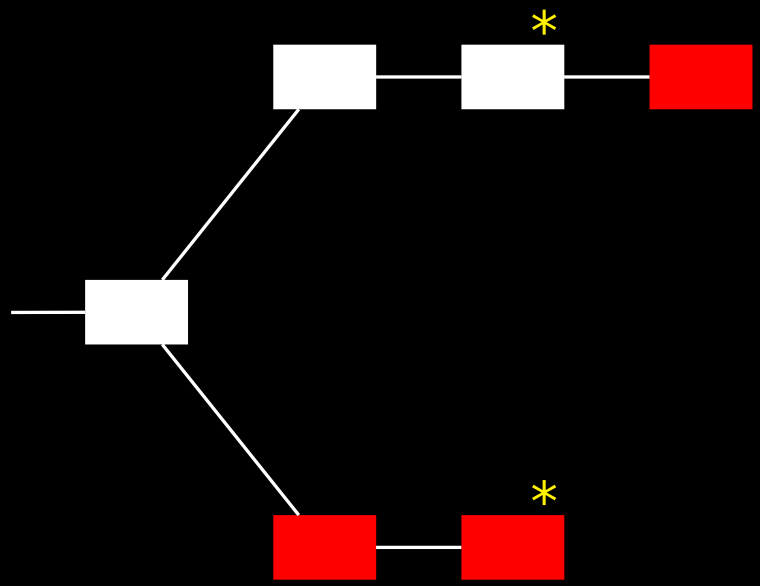
- White blocks have been computed by an honest party.
- Red blocks have been computed by the adversary.
- A star (∗) on a block means that an honest party has the chain ending with that block at the given round.

An execution example



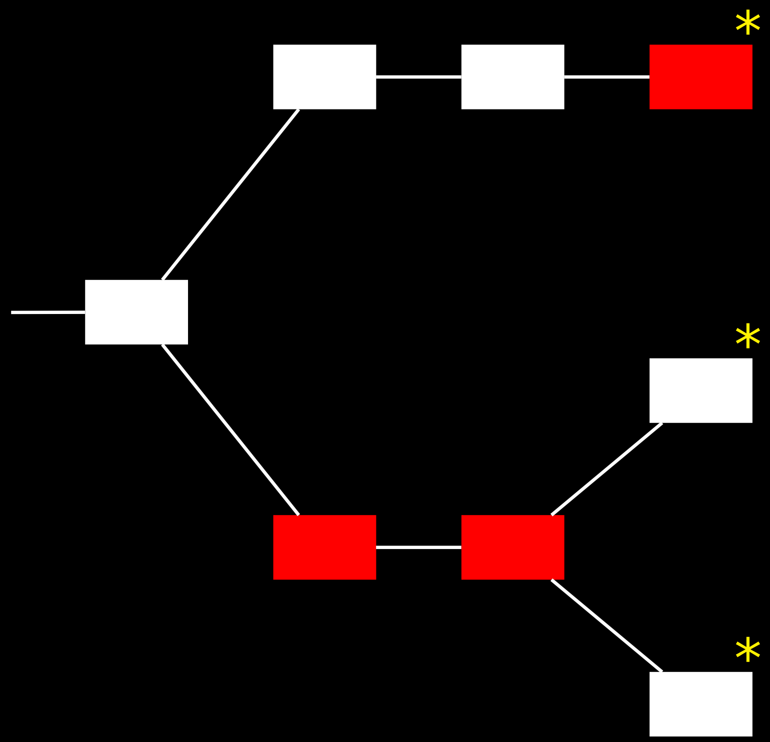
- White blocks have been computed by an honest party.
- Red blocks have been computed by the adversary.
- A star $(*)$ on a block means that an honest party has the chain ending with that block at the given round.

An execution example



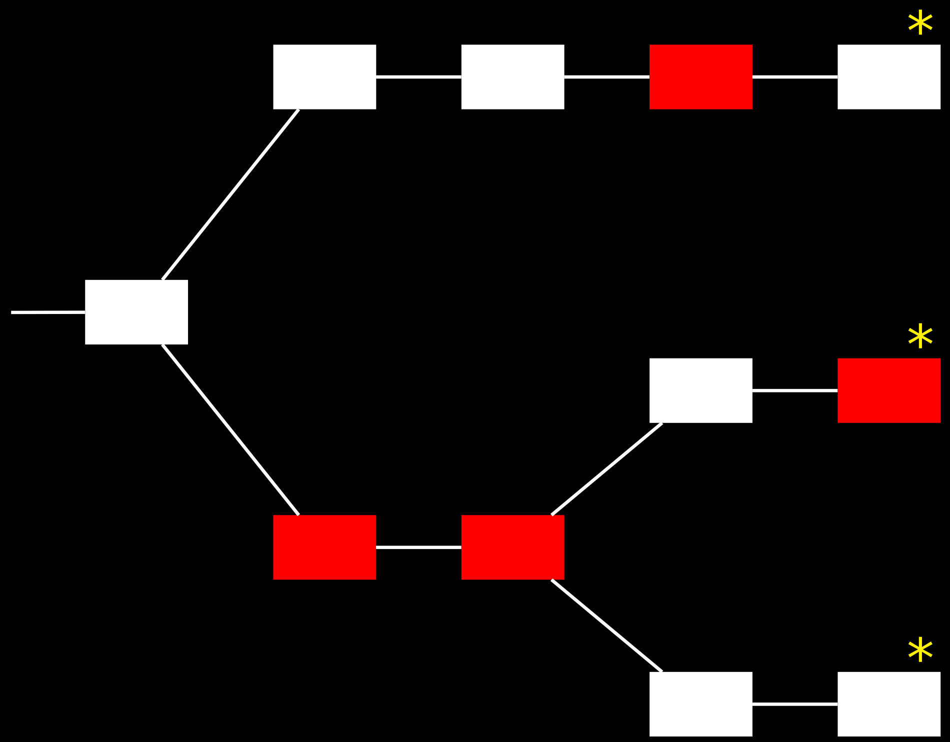
- White blocks have been computed by an honest party.
- Red blocks have been computed by the adversary.
- A star ($*$) on a block means that an honest party has the chain ending with that block at the given round.

An execution example



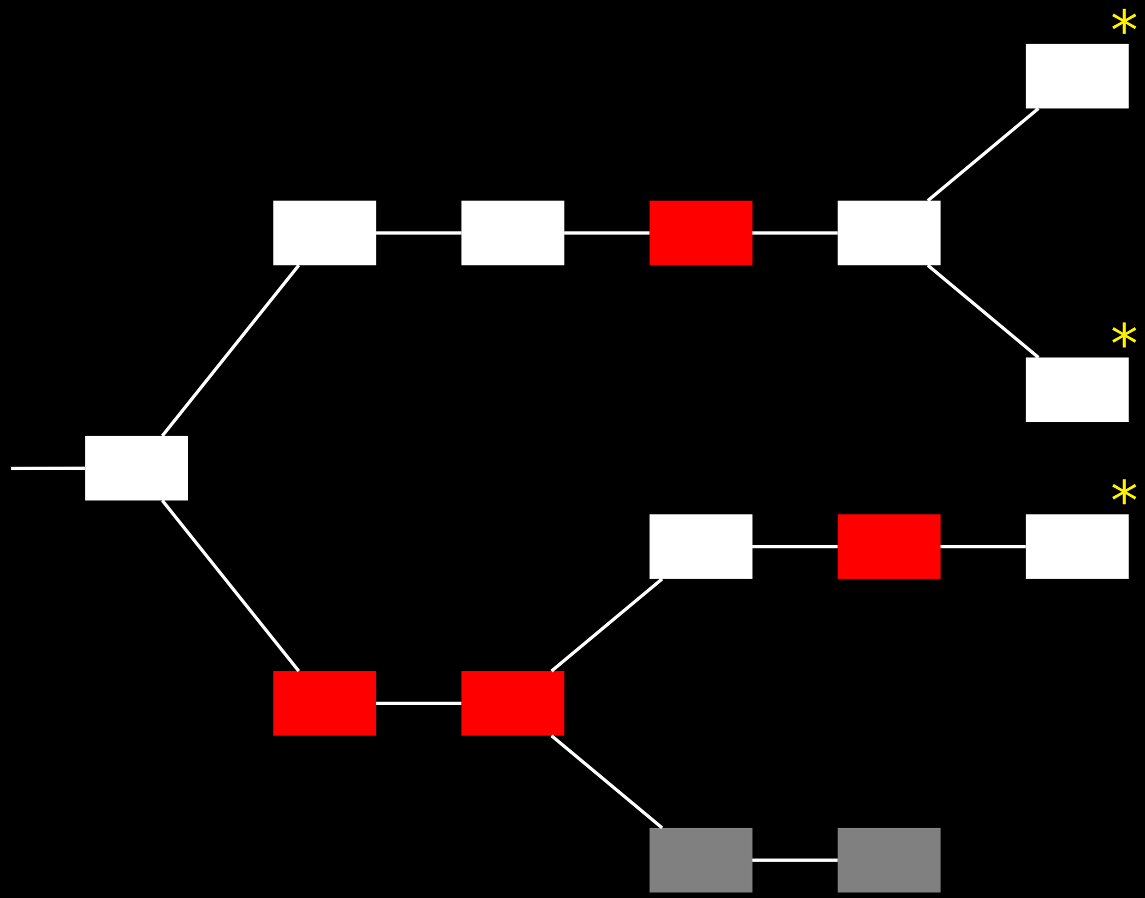
- White blocks have been computed by an honest party.
- Red blocks have been computed by the adversary.
- A star ($*$) on a block means that an honest party has the chain ending with that block at the given round.

An execution example



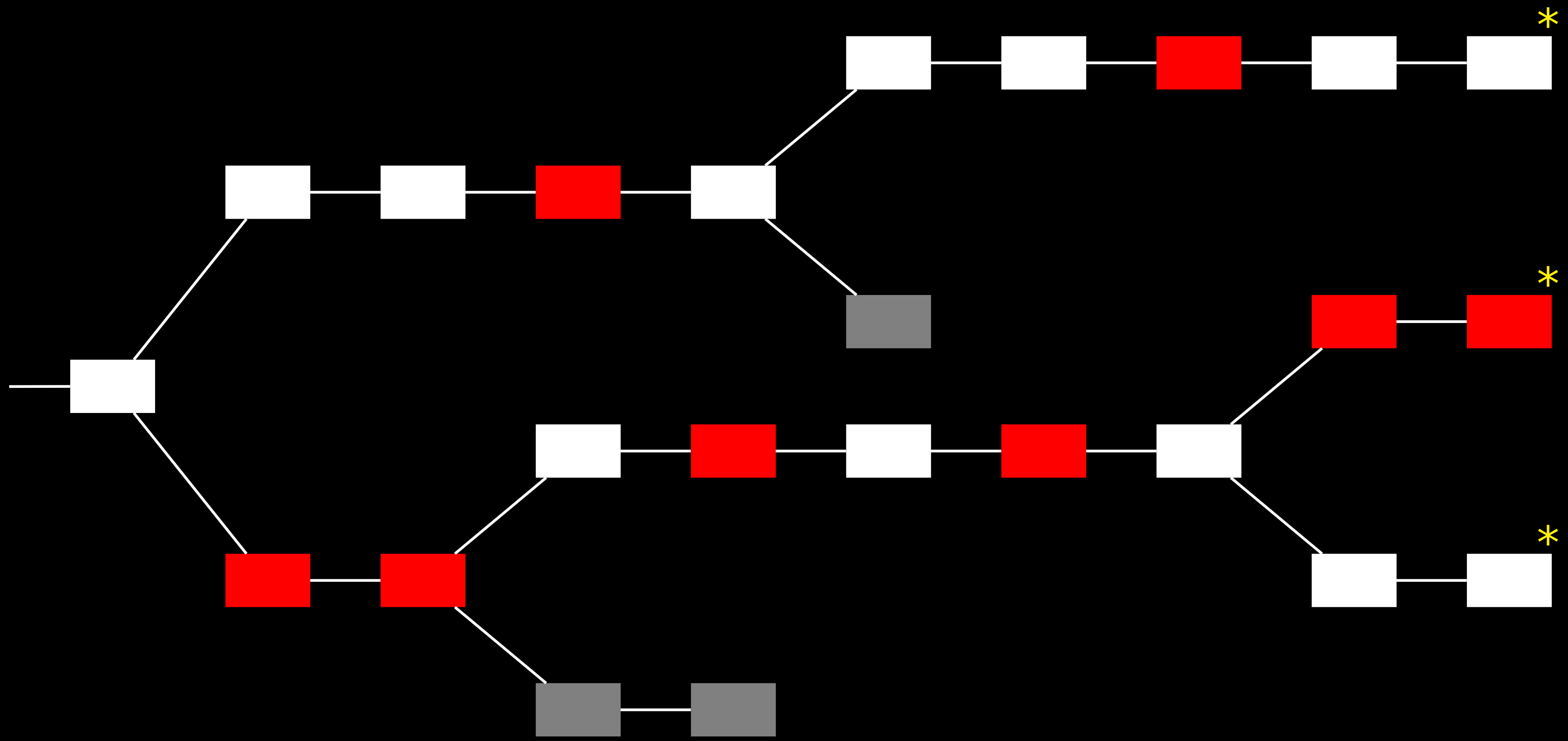
- White blocks have been computed by an honest party.
- Red blocks have been computed by the adversary.
- A star ($*$) on a block means that an honest party has the chain ending with that block at the given round.

An execution example



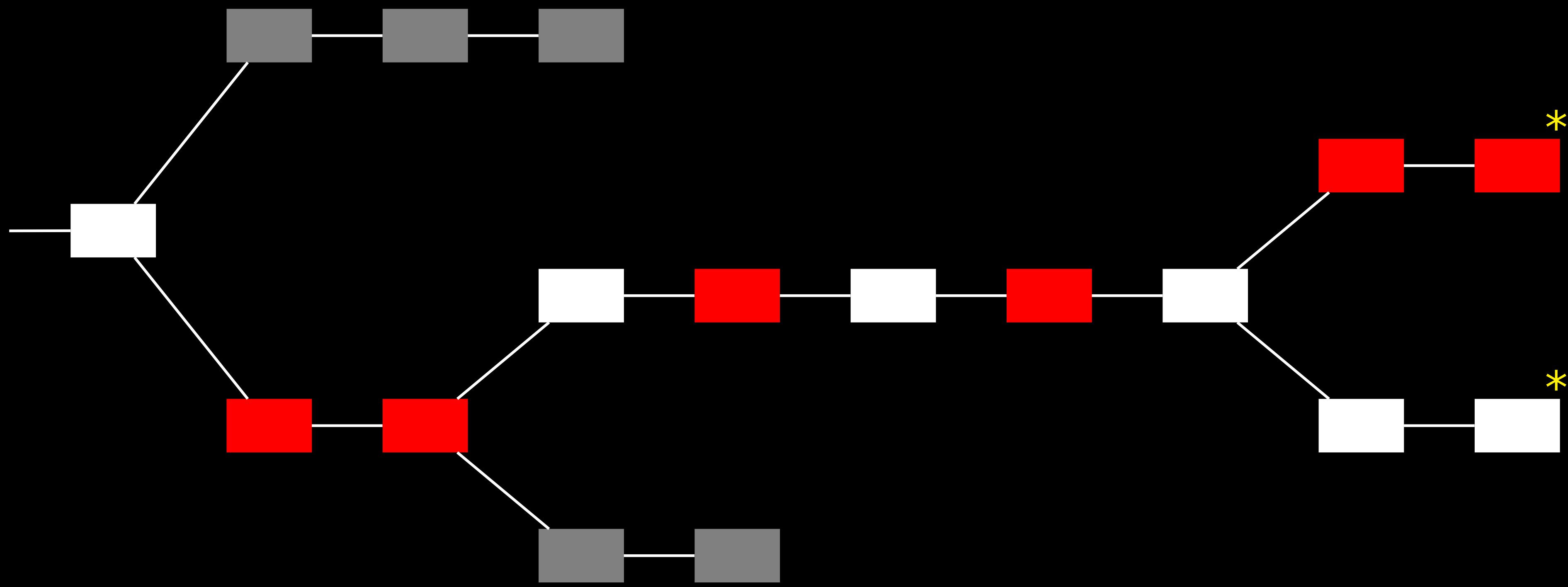
- White blocks have been computed by an honest party.
- Red blocks have been computed by the adversary.
- A star (∗) on a block means that an honest party has the chain ending with that block at the given round.

An execution example



- White blocks have been computed by an honest party.
- Red blocks have been computed by the adversary.
- A star ($*$) on a block means that an honest party has the chain ending with that block at the given round.

An execution example



- White blocks have been computed by an honest party.
- Red blocks have been computed by the adversary.
- A star (∗) on a block means that an honest party has the chain ending with that block at the given round.

Properties of the transaction ledger

Persistence. If a transaction is confirmed by an honest party, no honest party will ever disagree about the position of that transaction in the ledger.

Properties of the transaction ledger

Persistence. If a transaction is confirmed by an honest party, no honest party will ever disagree about the position of that transaction in the ledger.

Liveness. If a transaction is diffused, it will eventually become confirmed by all honest parties.

Properties of the transaction ledger

Persistence. If a transaction is confirmed by an honest party, no honest party will ever disagree about the position of that transaction in the ledger.

Liveness. If a transaction is diffused, it will eventually become confirmed by all honest parties.

Properties of the blockchain

Common-Prefix Property. Any two honest parties' chains have a large common prefix.

More formally: For any pair of honest parties adopting chains C_1 and C_2 at rounds $r_1 \leq r_2$ respectively, it holds $C_1^{\lceil k} \preceq C_2$.

Properties of the transaction ledger

Persistence. If a transaction is confirmed by an honest party, no honest party will ever disagree about the position of that transaction in the ledger.

Liveness. If a transaction is diffused, it will eventually become confirmed by all honest parties.

Properties of the blockchain

Common-Prefix Property. Any two honest parties' chains have a large common prefix.

More formally: For any pair of honest parties adopting chains C_1 and C_2 at rounds $r_1 \leq r_2$ respectively, it holds $C_1^{\lceil k} \preceq C_2$.

Chain-Quality Property. Any sufficiently large segment of an honest party's chain, will contain some blocks computed from honest parties.

Properties of the transaction ledger

Persistence. If a transaction is confirmed by an honest party, no honest party will ever disagree about the position of that transaction in the ledger.

Liveness. If a transaction is diffused, it will eventually become confirmed by all honest parties.

Properties of the blockchain

Common-Prefix Property. Any two honest parties' chains have a large common prefix.

More formally: For any pair of honest parties adopting chains C_1 and C_2 at rounds $r_1 \le r_2$ respectively, it holds $C_1^{\lceil k} \preceq C_2$.

Chain-Quality Property. Any sufficiently large segment of an honest party's chain, will contain some blocks computed from honest parties.

Chain-Growth Property. The chain of any honest party grows at least at a steady rate.

Analysis: Random Variables

Successful Round. A round r in which at least one honest party computes a block.

- Recall that a single query is successful with probability $p := T/2^\kappa$.

$$X_r = 1 \iff r \text{ is a successful round}$$
$$f := \mathbf{E}[X_r] = 1 - (1-p)^n \approx pn$$

Analysis: Random Variables

Successful Round. A round r in which at least one honest party computes a block.

- Recall that a single query is successful with probability $p := T/2^\kappa$.

$$X_r = 1 \iff r \text{ is a successful round}$$
$$f := \mathbf{E}[X_r] = 1 - (1-p)^n \approx pn$$

Uniquely Successful Round. A round r in which exactly one honest party computes a block.

$$Y_r = 1 \iff r \text{ is a uniqly successful round}$$
$$\mathbf{E}[Y_r] = np(1-p)^{n-1} > np(1-pn) \geq f(1-f)$$

Analysis: Random Variables

Successful Round. A round r in which at least one honest party computes a block.

- Recall that a single query is successful with probability $p := T/2^\kappa$.

$$X_r = 1 \iff r \text{ is a successful round}$$
$$f := \mathbf{E}[X_r] = 1 - (1-p)^n \approx pn$$

Uniquely Successful Round. A round r in which exactly one honest party computes a block.

$$Y_r = 1 \iff r \text{ is a uniqly successful round}$$
$$\mathbf{E}[Y_r] = np(1-p)^{n-1} > np(1-pn) \geq f(1-f)$$

Adversary. For each **query** j,

$$Z_j = 1 \iff \text{the adversary computed a block with his } j\text{-th query}$$
$$\mathbf{E}[Z_r] = \mathbf{E}[Z_1 + \cdots + Z_t] = \mathbf{E}[Z_r] = \mathbf{E}[Z_1] + \cdots + \mathbf{E}[Z_t] = pt$$

Chain-Growth Lemma

Chain-Growth Lemma. *Suppose that at round r an honest party has a chain of length ℓ. Then, by round $s \geq r$, every honest party has adopted a chain of length at least*

$$\ell + X_r + \cdots + X_{s-1}.$$

Chain-Growth Lemma

Chain-Growth Lemma. *Suppose that at round r an honest party has a chain of length ℓ. Then, by round $s \geq r$, every honest party has adopted a chain of length at least*

$$\ell + X_r + \cdots + X_{s-1}.$$

Chernoff Bound. *Suppose $\{X_i : i \in [n]\}$ are mutually independent Boolean random variables, with $\Pr[X_i = 1] = p$, for all $i \in [n]$. Let $X = \sum_{i=1}^{n} X_i$ and $\mu = pn$. Then, for any $\delta \in (0, 1]$,*

$$\Pr[X \leq (1-\delta)\mu] \leq e^{-\delta^2\mu/2} \quad \text{and} \quad \Pr[X \geq (1+\delta)\mu] \leq e^{-\delta^2\mu/3}.$$

Chain-Growth Lemma

Chain-Growth Lemma. *Suppose that at round r an honest party has a chain of length ℓ. Then, by round $s \geq r$, every honest party has adopted a chain of length at least*

$$\ell + X_r + \cdots + X_{s-1}.$$

Chernoff Bound. *Suppose $\{X_i : i \in [n]\}$ are mutually independent Boolean random variables, with $\Pr[X_i = 1] = p$, for all $i \in [n]$. Let $X = \sum_{i=1}^{n} X_i$ and $\mu = pn$. Then, for any $\delta \in (0, 1]$,*

$$\Pr[X \leq (1-\delta)\mu] \leq e^{-\delta^2\mu/2} \quad \textit{and} \quad \Pr[X \geq (1+\delta)\mu] \leq e^{-\delta^2\mu/3}.$$

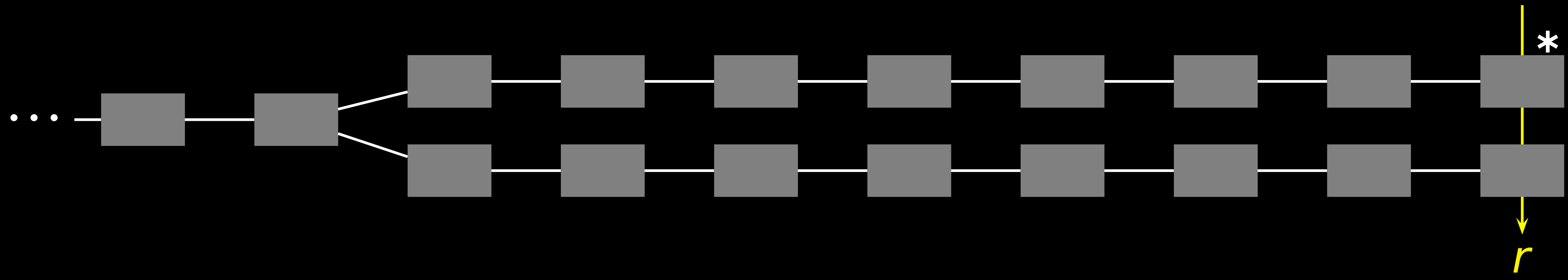
Chain-growth property. *With probability at least $1-e^{-\Omega(\epsilon^2 fs)}$, the chain of any honest party increases by at least*

$$(1-\epsilon)fs \approx (1-\epsilon)pns$$

blocks after s consecutive rounds. ($\mathbf{E}[X_1 + \cdots + X_s] = fs \approx pns$.)

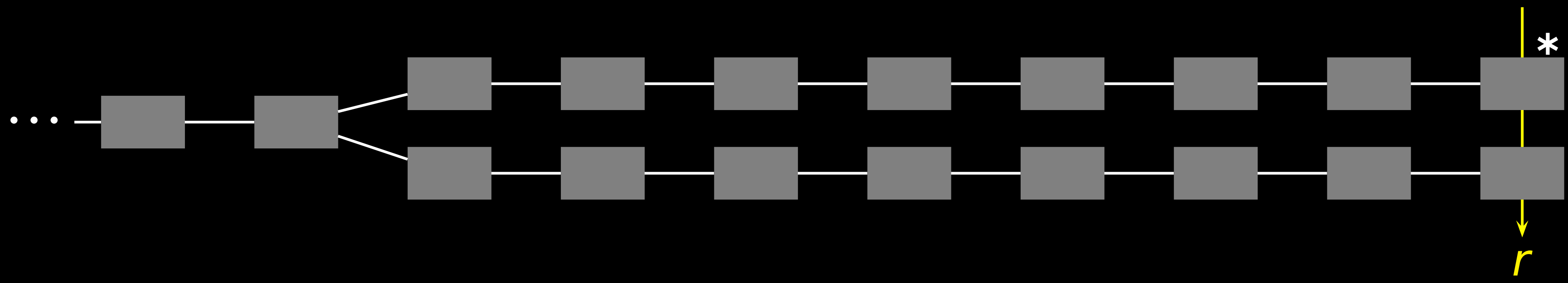
Common-Prefix Lemma

Common-Prefix Lemma. *The probability that at a given round two parties have chains that disagree in the last* k *blocks, is at most* $e^{-\Omega(k)}$. *(The party with the shortest chain should be honest.)*



Common-Prefix Lemma

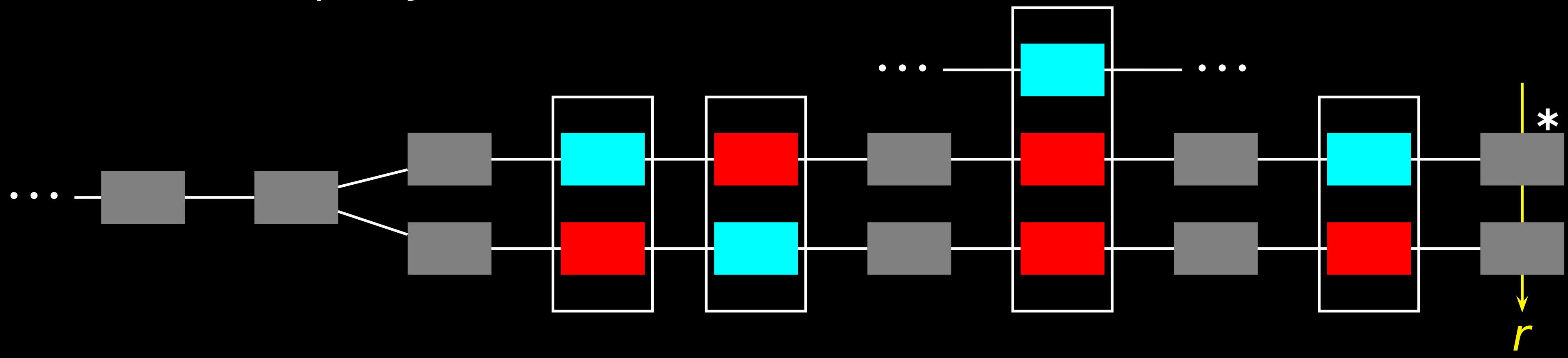
Common-Prefix Lemma. *The probability that at a given round two parties have chains that disagree in the last k blocks, is at most $e^{-\Omega(k)}$. (The party with the shortest chain should be honest.)*



Observation. *Suppose the ℓ-the block of a chain was computed by an honest party in a uniquely successful round. Then any other ℓ-th block has been computed by the adversary.*

Proof of the common-prefix lemma [GKL15]

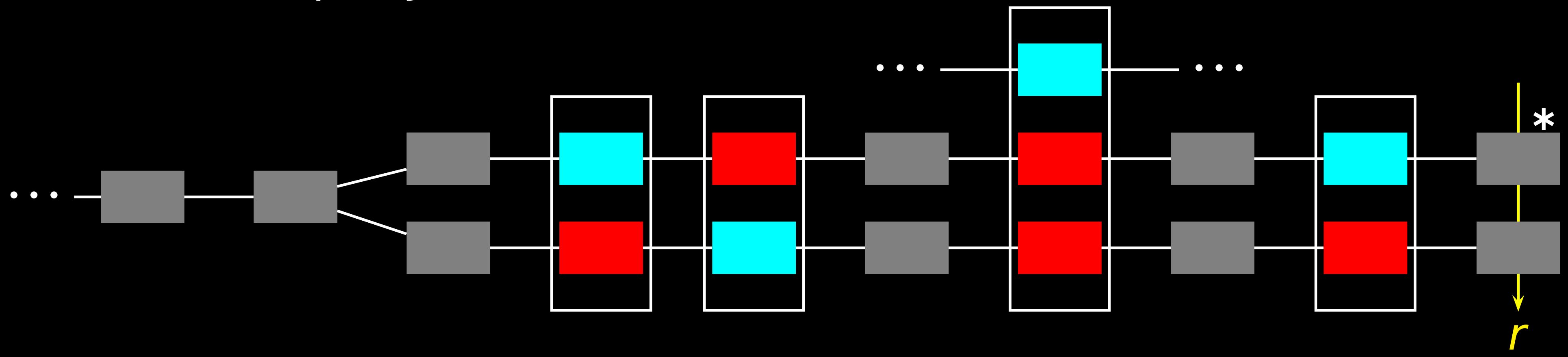
Common-Prefix Lemma. *The probability that at a given round two parties have chains that disagree in the last k blocks, is at most $e^{-\Omega(k)}$. (The party with the shortest chain should be honest.)*



Observation. *Suppose the ℓ-the block of a chain was computed by an honest party in a uniquely successful round. Then any other ℓ-th block has been computed by the adversary.*

Proof of the common-prefix lemma [GKL15]

Common-Prefix Lemma. *The probability that at a given round two parties have chains that disagree in the last k blocks, is at most $e^{-\Omega(k)}$. (The party with the shortest chain should be honest.)*

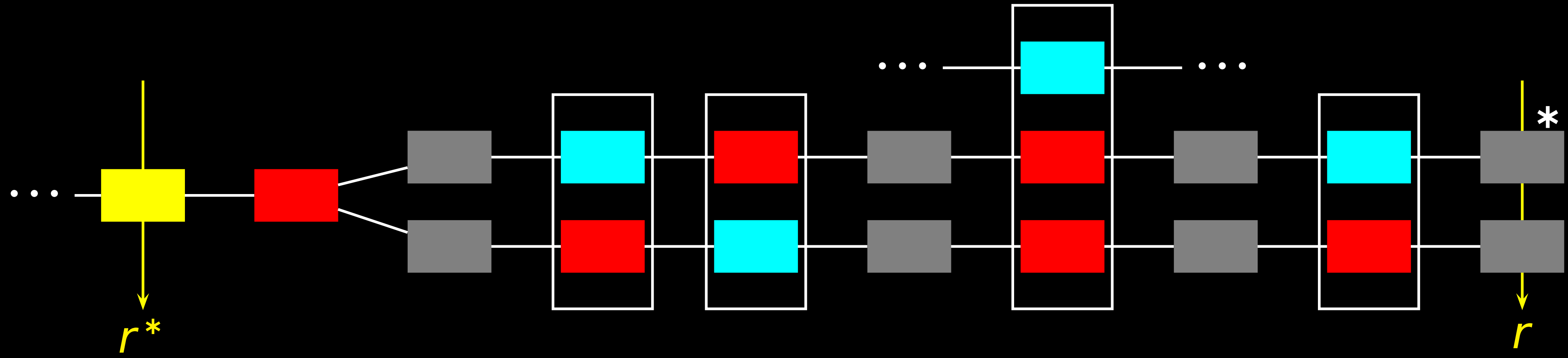


Observation. *Suppose the ℓ-the block of a chain was computed by an honest party in a uniquely successful round. Then any other ℓ-th block has been computed by the adversary.*

Proof. Suppose a block of height ℓ was computed by an honest party at a round u with $Y_u = 1$. If any honest party computed a block of height ℓ at any round $r < u$, then any honest party is trying to extend a chain of length at least ℓ at round u. Similarly for $r > u$.

Proof of the common-prefix lemma [GKL15]

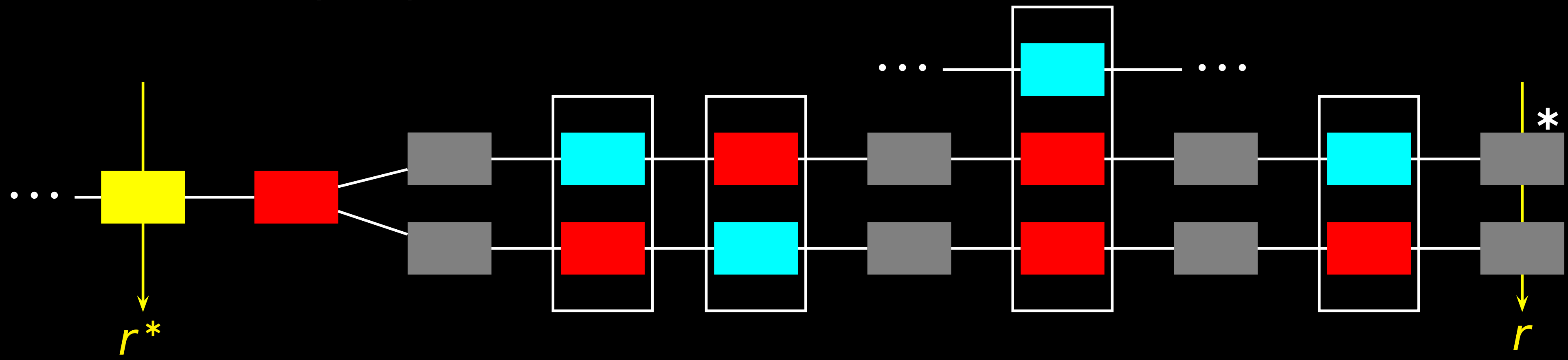
Common-Prefix Lemma. *The probability that at a given round two parties have chains that disagree in the last* k *blocks, is at most* $e^{-\Omega(k)}$. *(The party with the shortest chain should be honest.)*



Proof. Let r^* be the last round in which a block before the fork was computed by an honest party. Set $S = \{r^* + 1, \dots, r - 1\}$.

Proof of the common-prefix lemma [GKL15]

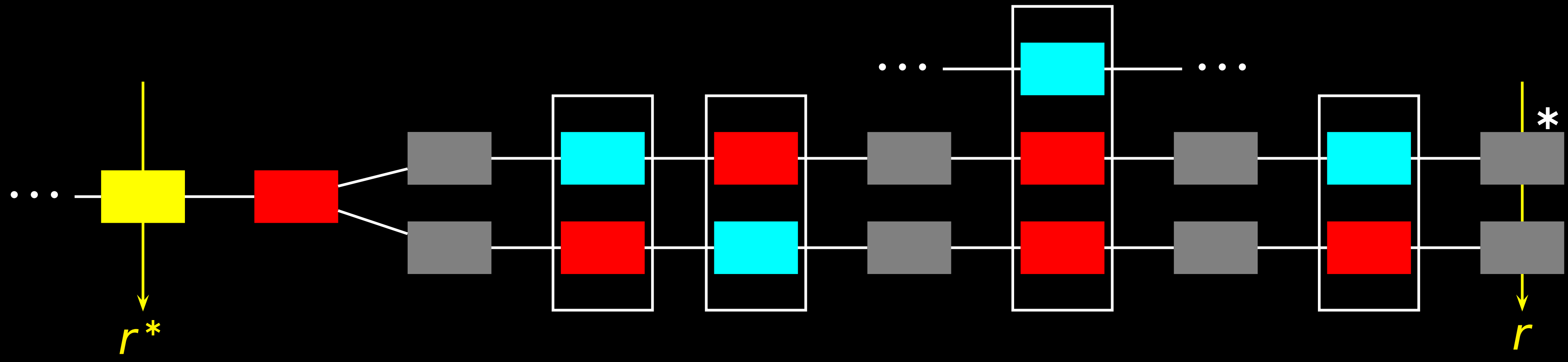
Common-Prefix Lemma. *The probability that at a given round two parties have chains that disagree in the last* k *blocks, is at most* $e^{-\Omega(k)}$. *(The party with the shortest chain should be honest.)*



Proof. Let r^* be the last round in which a block before the fork was computed by an honest party. Set $S = \{r^* + 1, \ldots, r - 1\}$. By the Observation, to every uniquely successful round in S corresponds an adversarial block computed in S.

Proof of the common-prefix lemma [GKL15]

Common-Prefix Lemma. *The probability that at a given round two parties have chains that disagree in the last k blocks, is at most $e^{-\Omega(k)}$. (The party with the shortest chain should be honest.)*

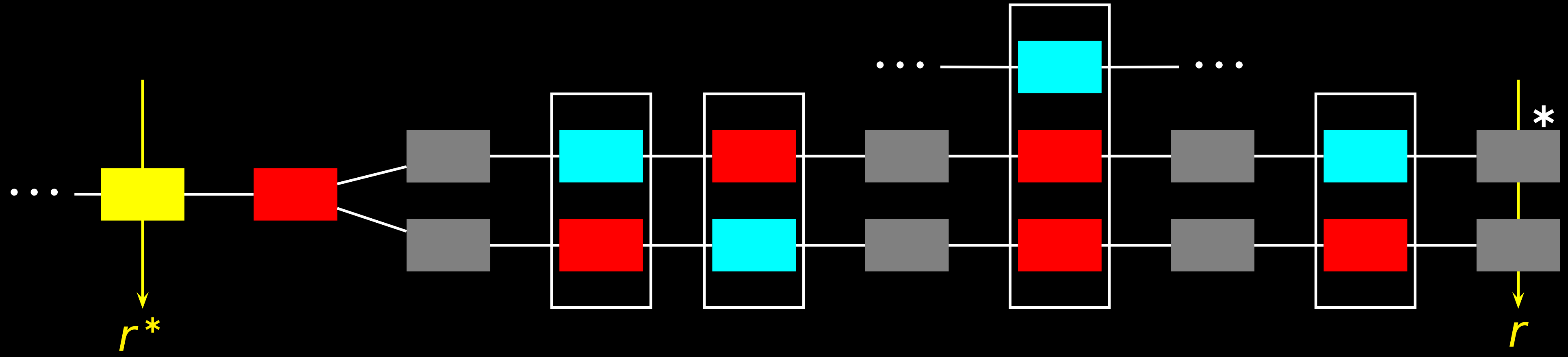


Proof. Let r^* be the last round in which a block before the fork was computed by an honest party. Set $S = \{r^* + 1, \ldots, r - 1\}$. By the Observation, to every uniquely successful round in S corresponds an adversarial block computed in S. It follows that

Uniqely successful rounds in S $\leq$ Adversarial successes in S.

Proof of the common-prefix lemma [GKL15]

Common-Prefix Lemma. *The probability that at a given round two parties have chains that disagree in the last k blocks, is at most $e^{-\Omega(k)}$. (The party with the shortest chain should be honest.)*



Proof. Let r^* be the last round in which a block before the fork was computed by an honest party. Set $S = \{r^* + 1, \ldots, r - 1\}$. By the Observation, to every uniquely successful round in S corresponds an adversarial block computed in S. It follows that

Uniqely successful rounds in S	$\leq$	Adversarial successes in S.
$E[\sum Y_i] \approx pn(1-f)\lvert S\rvert$		$E[\sum Z_i] = pt\lvert S\rvert.$

Proof of the common-prefix lemma (cont'd)

Recall that $\mathbf{E}[Y_i] > f(1-f)$. Let $Y(S) = \sum_{r \in S} Y_r$. Then, since $\mathbf{E}[Y(S)] = \sum_{r \in S} f(1-f) = f(1-f)|S|$, by the Chernoff bound,

$$\Pr[Y(S) \le (1-\epsilon)f(1-f)|S|] = e^{-\Omega(|S|)}.$$

Similarly

$$\Pr[Z(S) \ge (1+\epsilon)pt|S|] = e^{-\Omega(|S|)}.$$

Proof of the common-prefix lemma (cont'd)

Recall that $\mathbf{E}[Y_i] > f(1-f)$. Let $Y(S) = \sum_{r\in S} Y_r$. Then, since $\mathbf{E}[Y(S)] = \sum_{r\in S} f(1-f) = f(1-f)|S|$, by the Chernoff bound,

$$\Pr[Y(S) \le (1-\epsilon)f(1-f)|S|] = e^{-\Omega(|S|)}.$$

Similarly

$$\Pr[Z(S) \ge (1+\epsilon)pt|S|] = e^{-\Omega(|S|)}.$$

Honest Majority Assumption. $t < (1-\delta)n$ *for* $\delta > 3\epsilon + 3f$.

Proof of the common-prefix lemma (cont'd)

Recall that $\mathbf{E}[Y_i] > f(1-f)$. Let $Y(S) = \sum_{r \in S} Y_r$. Then, since $\mathbf{E}[Y(S)] = \sum_{r \in S} f(1-f) = f(1-f)|S|$, by the Chernoff bound,

$$\Pr[Y(S) \leq (1-\epsilon)f(1-f)|S|] = e^{-\Omega(|S|)}.$$

Similarly

$$\Pr[Z(S) \geq (1+\epsilon)pt|S|] = e^{-\Omega(|S|)}.$$

Honest Majority Assumption. *$t < (1-\delta)n$ for $\delta > 3\epsilon + 3f$.*

Assuming these bad events don't occur (union bound) and the Honest Majority Assumption

$$\begin{aligned} Z(S) &< (1+\epsilon)pt|S| \\ &< (1+\epsilon)(1-\delta)pn|S| && \{\ t < (1-\delta)n\ \} \\ &< (1+\epsilon)(1-\delta) \cdot \frac{f}{1-f} \cdot |S| && \{\ (1-f)pn < f\ \} \\ &< (1-\epsilon)f|S| && \{\ \delta > 3\epsilon + 3f\ \} \\ &< Y(S) \end{aligned}$$

$\square$