Bitcoin Backbone and Consensus

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], Szabo's bit gold [1998], Karma by Vishnumurthy, Chandrakumar, Sirer [2003]. See also references in Nakamoto's Bitcoin whitepaper (Hashcash by Back and work on time-stamping by Haber and Stornetta).
- 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

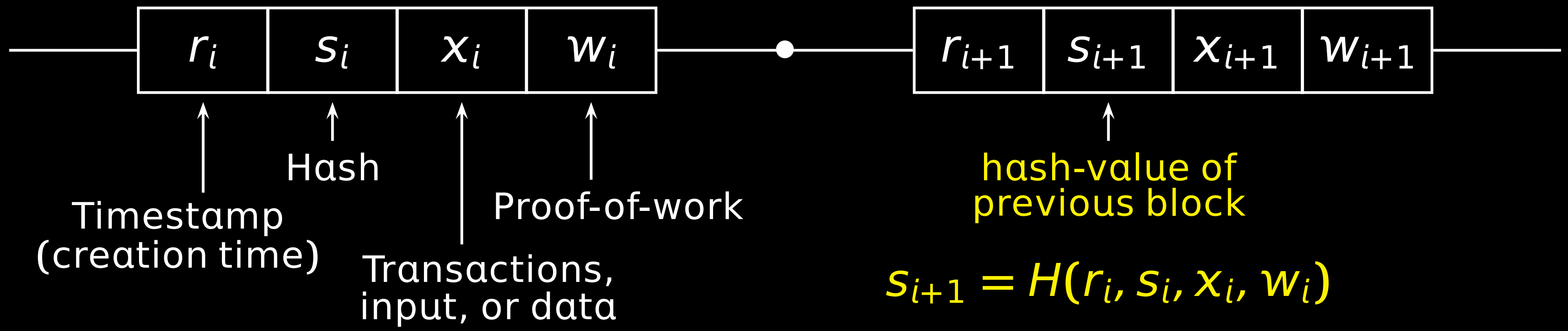
A moderately hard computational task: Given a hash-function $H(\cdot)$ with range $\{0,1\}^{\kappa}$ and a y, find x such that $H(x,y)$ begins with a lot of zeroes. More generaly, given a target T,

- 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



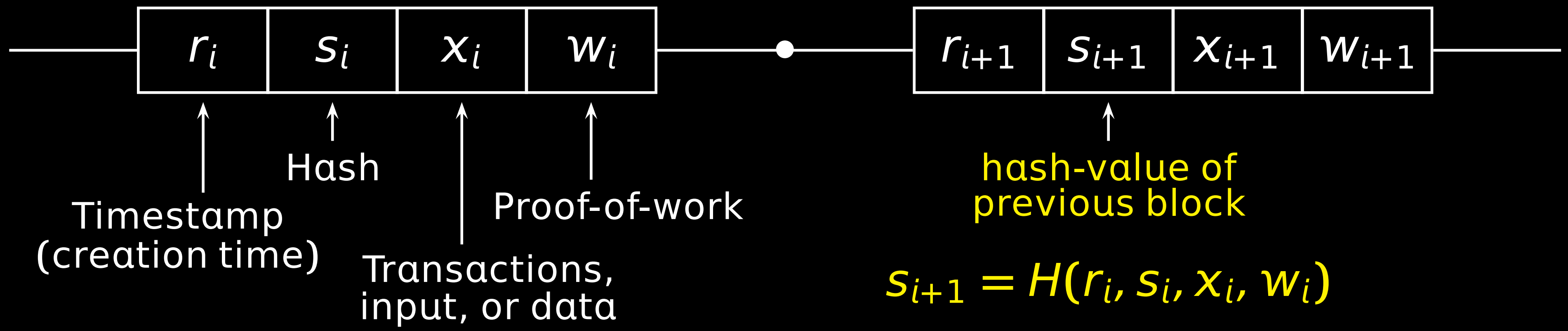
- A block (r, s, x, w) is valid if it has a small hash-value, providing a proof-of-work:

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

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

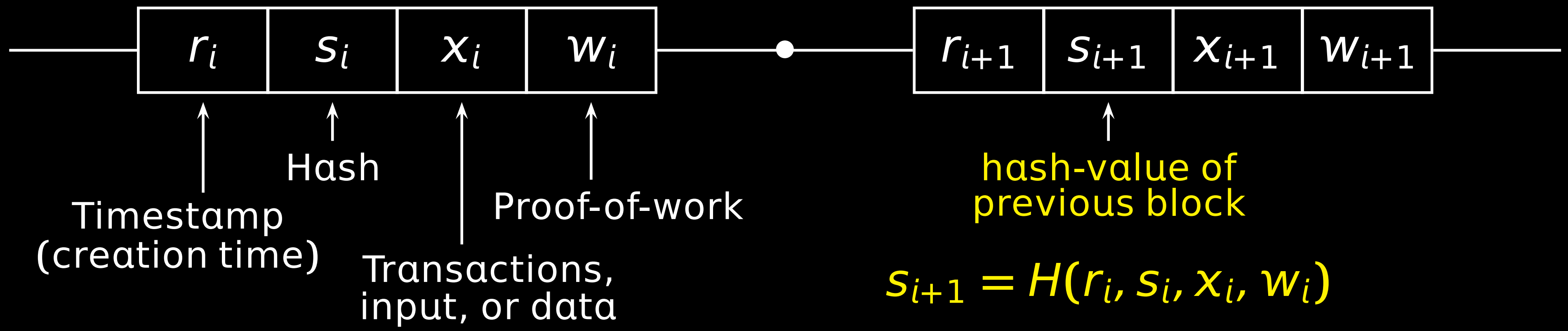
$$\text{for each } i, \quad s_{i+1} = H(r_i, s_i, x_i, w_i) \;\text{ and }\; r_{i+1} > r_i.$$

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

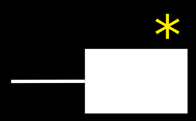
  $$H(r, s, x, w) < T.$$

  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.

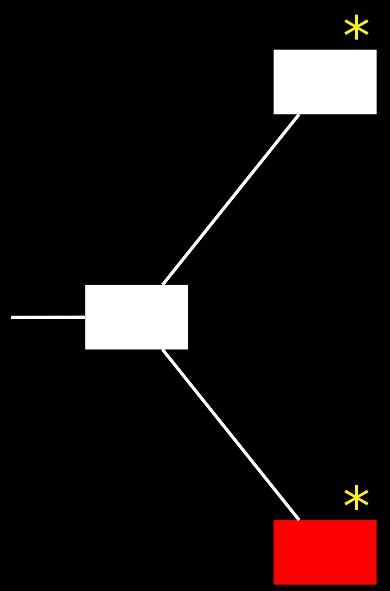
An execution example

An execution example



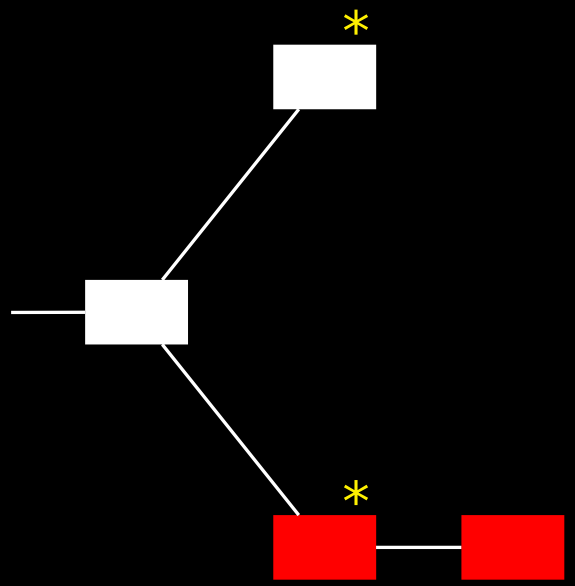
- White blocks have been computed by an honest party.
- Red blocks have been computed by the adversary.
- A star (∗) on a block means that an honest party has the chain ending with that block at the given round.

An execution example



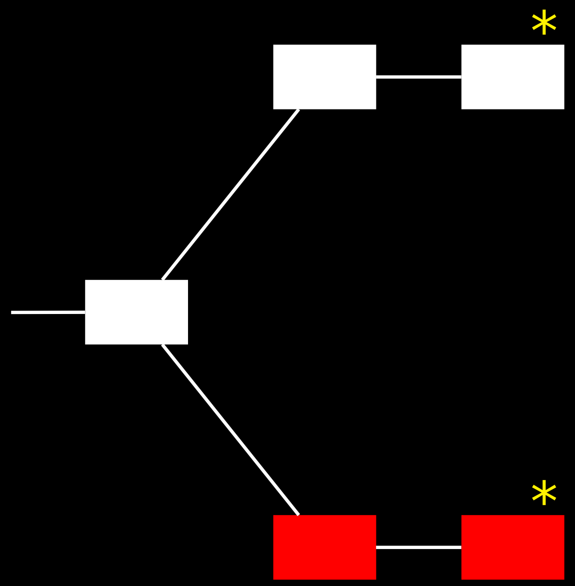
- White blocks have been computed by an honest party.
- Red blocks have been computed by the adversary.
- A star (∗) on a block means that an honest party has the chain ending with that block at the given round.

An execution example



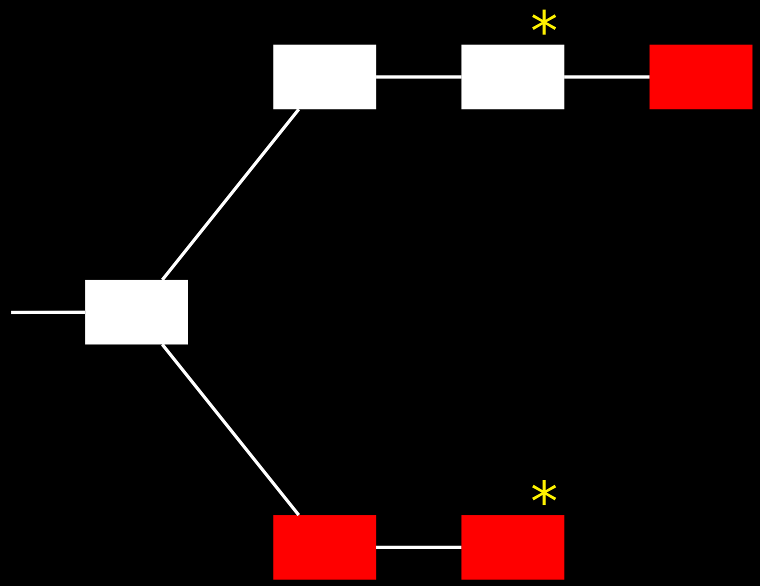
- White blocks have been computed by an honest party.
- Red blocks have been computed by the adversary.
- A star $(*)$ on a block means that an honest party has the chain ending with that block at the given round.

An execution example



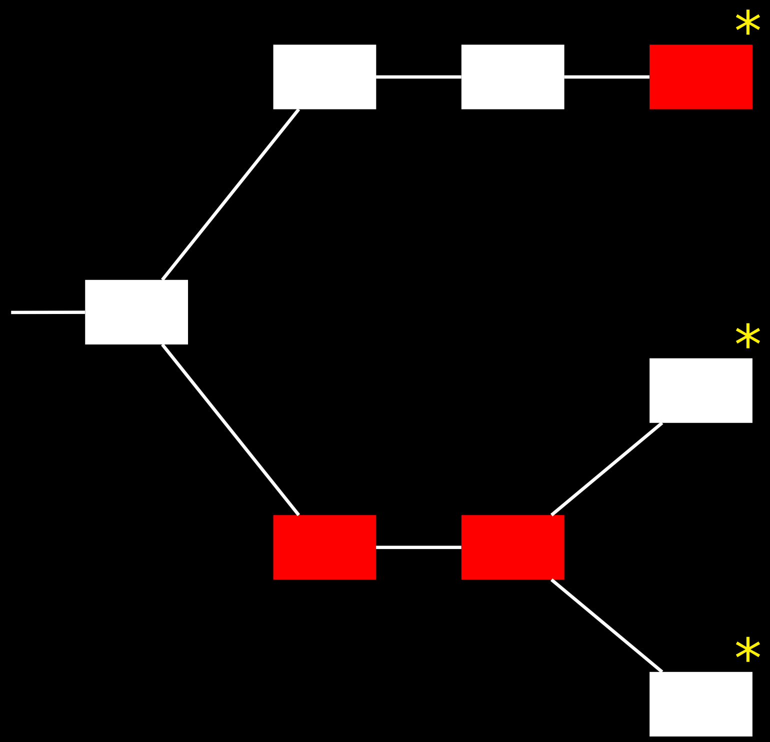
- White blocks have been computed by an honest party.
- Red blocks have been computed by the adversary.
- A star (*) on a block means that an honest party has the chain ending with that block at the given round.

An execution example



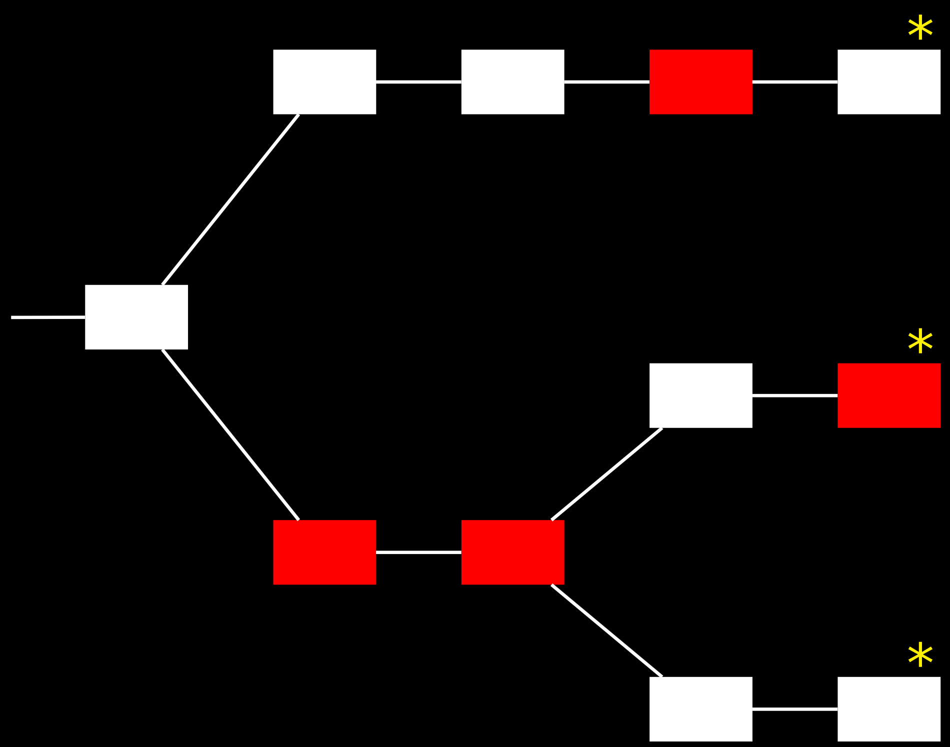
- White blocks have been computed by an honest party.
- Red blocks have been computed by the adversary.
- A star (∗) on a block means that an honest party has the chain ending with that block at the given round.

An execution example



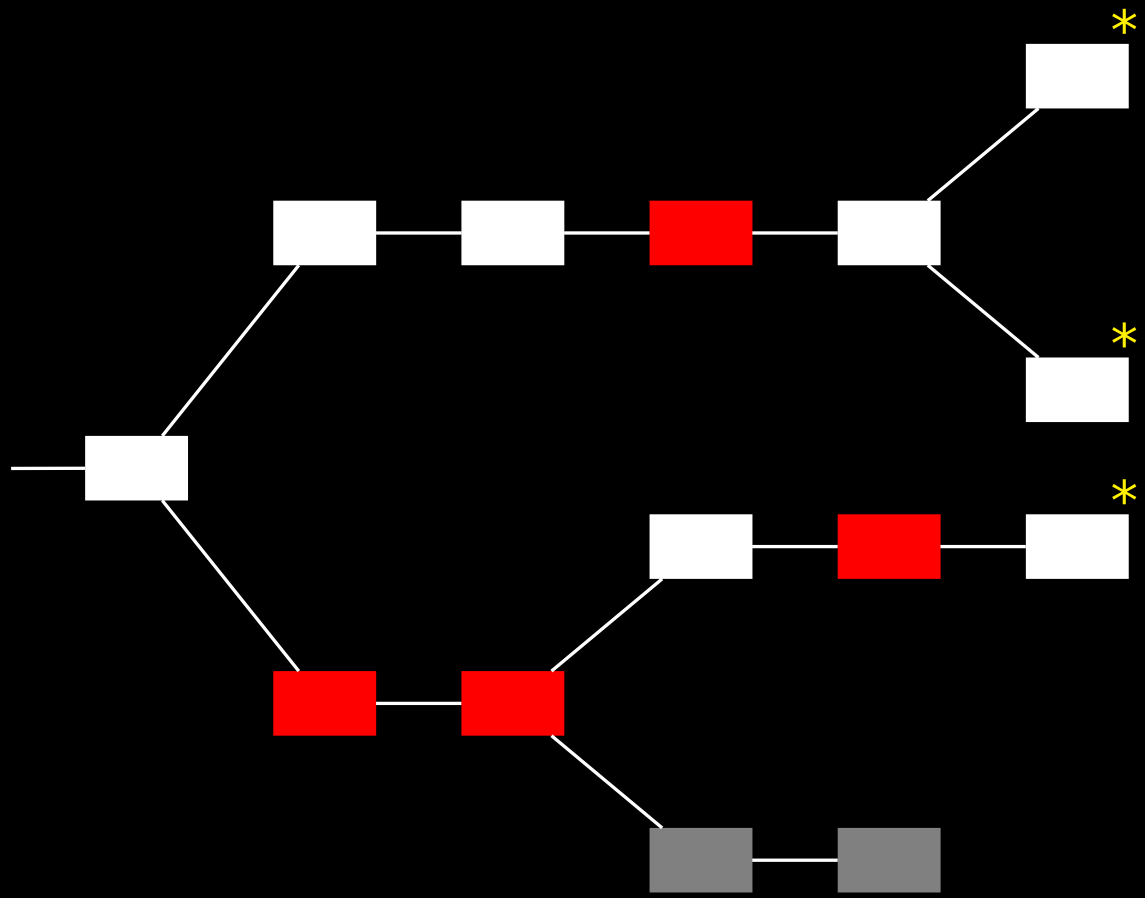
- White blocks have been computed by an honest party.
- Red blocks have been computed by the adversary.
- A star (∗) on a block means that an honest party has the chain ending with that block at the given round.

An execution example



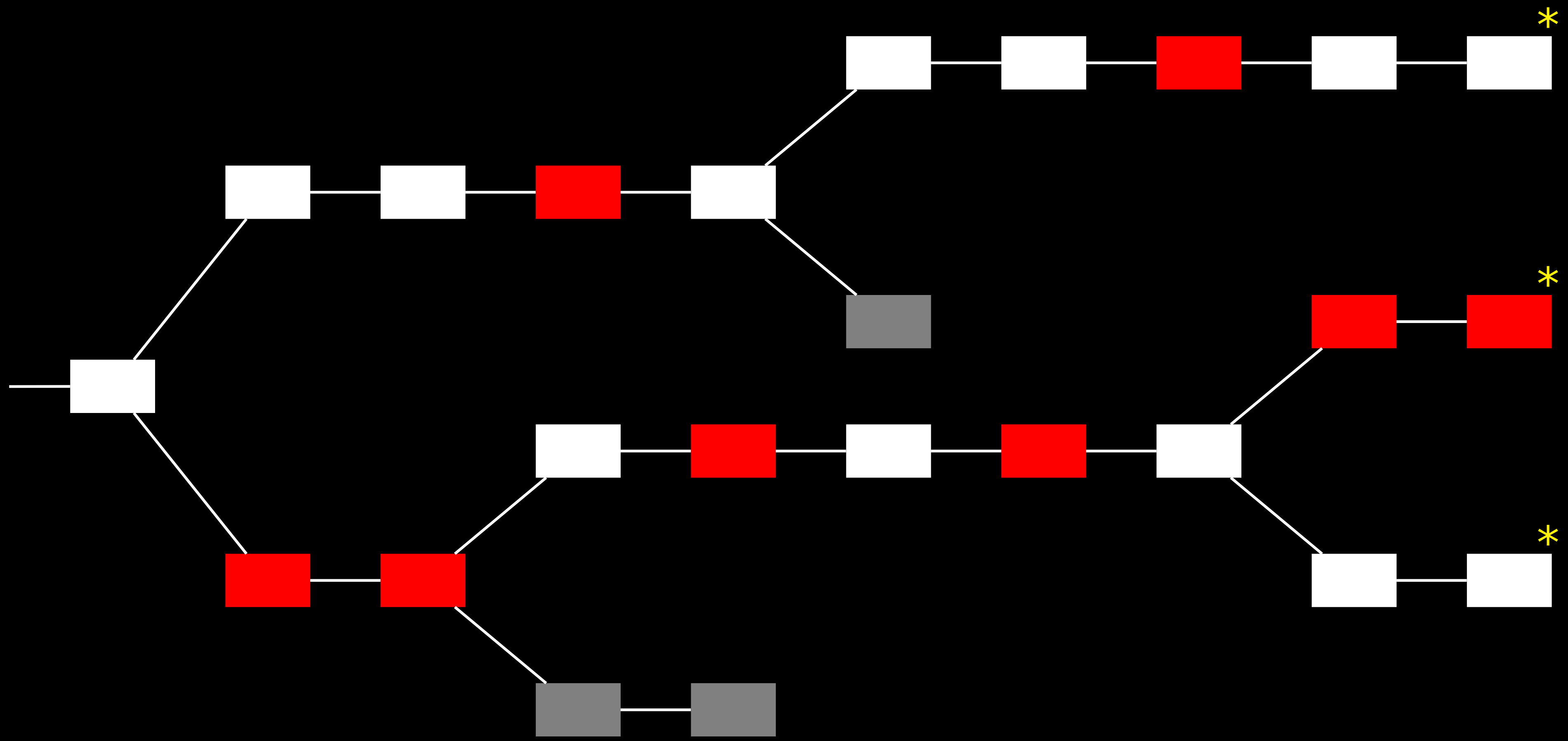
- White blocks have been computed by an honest party.
- Red blocks have been computed by the adversary.
- A star (∗) on a block means that an honest party has the chain ending with that block at the given round.

An execution example



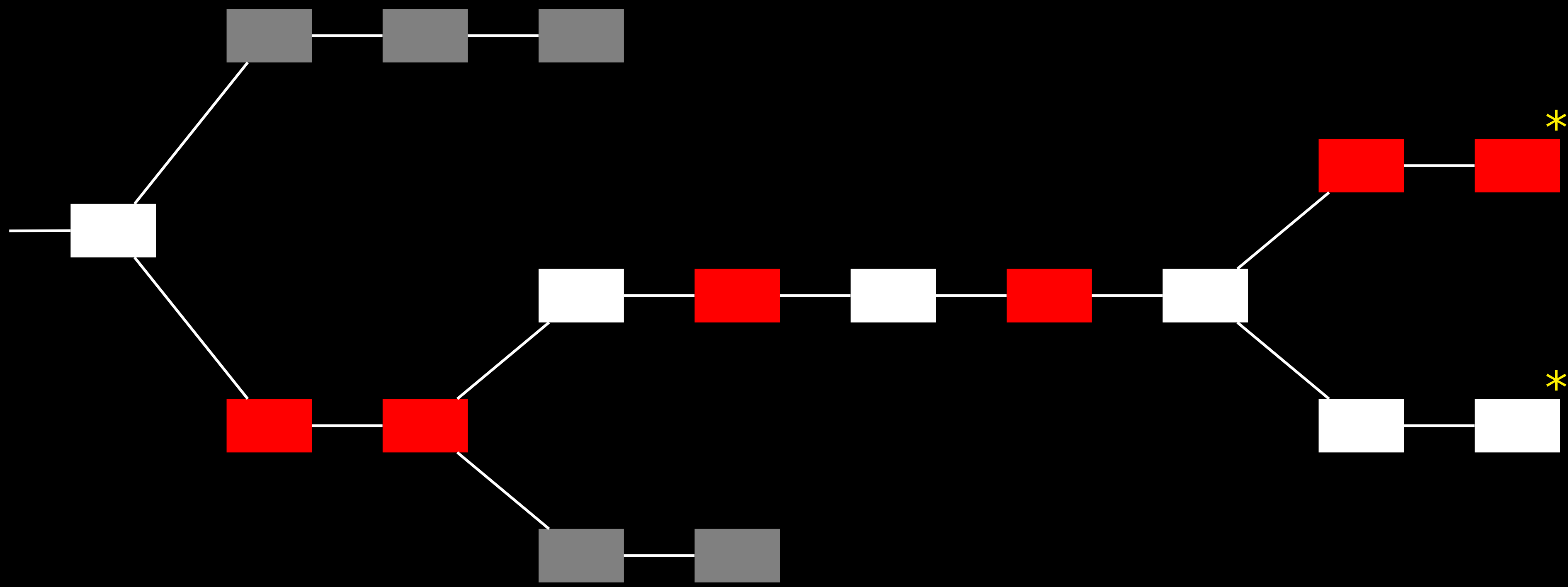
- White blocks have been computed by an honest party.
- Red blocks have been computed by the adversary.
- A star (∗) on a block means that an honest party has the chain ending with that block at the given round.

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 $r_1 \leq r_2$ 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.

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 $\delta \in (0, 1]$,*

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

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. ($\mathbf{E}[X_1 + \cdots + X_s] = fs \approx pns$.)

Chvátal's trick

Proof (Chvatal's trick). Let $X \sim Bin(n, p)$ and $k = (p + t)n$.

$$\Pr[X \geq k] = \sum_{i=k}^{n} \binom{n}{i} p^i (1-p)^{n-i}$$

□

Chvátal's trick

Proof (Chvatal's trick). Let $X \sim Bin(n,p)$ and $k = (p+t)n$.

$$\Pr[X \geq k] = \sum_{i=k}^{n} \binom{n}{i} p^i (1-p)^{n-i}$$

$$\leq \sum_{i=0}^{n} \binom{n}{i} p^i (1-p)^{n-i} x^{i-k} \quad (x \geq 1)$$

□

Chvátal's trick

Proof (Chvatal's trick). Let $X \sim Bin(n,p)$ and $k = (p+t)n$.

$$\Pr[X \geq k] = \sum_{i=k}^{n} \binom{n}{i} p^i (1-p)^{n-i}$$

$$\leq \sum_{i=0}^{n} \binom{n}{i} p^i (1-p)^{n-i} x^{i-k} \quad (x \geq 1)$$

$$= x^{-k} \sum_{i=0}^{n} \binom{n}{i} (px)^i (1-p)^{n-i}$$

□

Chvátal's trick

Proof (Chvatal's trick). Let $X \sim Bin(n,p)$ and $k=(p+t)n$.

$$\begin{aligned}
\Pr[X \geq k] &= \sum_{i=k}^{n} \binom{n}{i} p^i (1-p)^{n-i} \\
&\leq \sum_{i=0}^{n} \binom{n}{i} p^i (1-p)^{n-i} x^{i-k} \quad (x \geq 1) \\
&= x^{-k} \sum_{i=0}^{n} \binom{n}{i} (px)^i (1-p)^{n-i} \\
&= x^{-k} (1+(x-1)p)^n
\end{aligned}$$

□

Chvátal's trick

Proof (Chvatal's trick). Let $X \sim Bin(n, p)$ and $k = (p + t)n$.

$$\begin{aligned}
\Pr[X \geq k] &= \sum_{i=k}^{n} \binom{n}{i} p^i (1-p)^{n-i} \\
&\leq \sum_{i=0}^{n} \binom{n}{i} p^i (1-p)^{n-i} x^{i-k} \quad (x \geq 1) \\
&= x^{-k} \sum_{i=0}^{n} \binom{n}{i} (px)^i (1-p)^{n-i} \\
&= x^{-k} (1 + (x-1)p)^n \\
&= \left[\left(\frac{p}{p+t} \right)^{p+t} \left(\frac{1-p}{1-p-t} \right)^{1-p-t} \right]^n \quad \left(x = \frac{(1-p)(p+t)}{p(1-p-t)} \right)
\end{aligned}$$

□

Chvátal's trick

Proof (Chvatal's trick). Let $X \sim Bin(n, p)$ and $k = (p + t)n$.

$$\Pr[X \geq k] = \sum_{i=k}^{n} \binom{n}{i} p^i (1-p)^{n-i}$$

$$\leq \sum_{i=0}^{n} \binom{n}{i} p^i (1-p)^{n-i} x^{i-k} \quad (x \geq 1)$$

$$= x^{-k} \sum_{i=0}^{n} \binom{n}{i} (px)^i (1-p)^{n-i}$$

$$= x^{-k} (1 + (x-1)p)^n$$

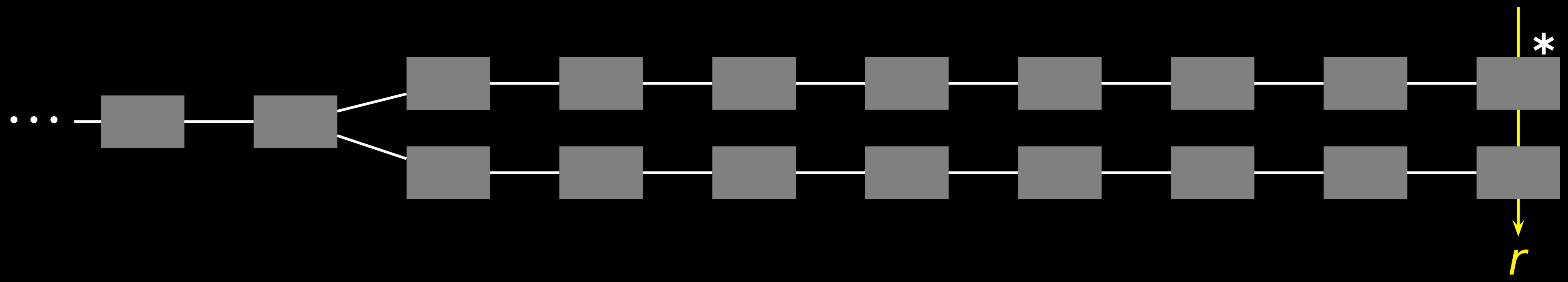
$$= \left[\left(\frac{p}{p+t} \right)^{p+t} \left(\frac{1-p}{1-p-t} \right)^{1-p-t} \right]^n \quad \left(x = \frac{(1-p)(p+t)}{p(1-p-t)} \right)$$

$\ldots$

$\ldots$ (Calculus) $\ldots \quad \leq e^{-2t^2 n}$ □

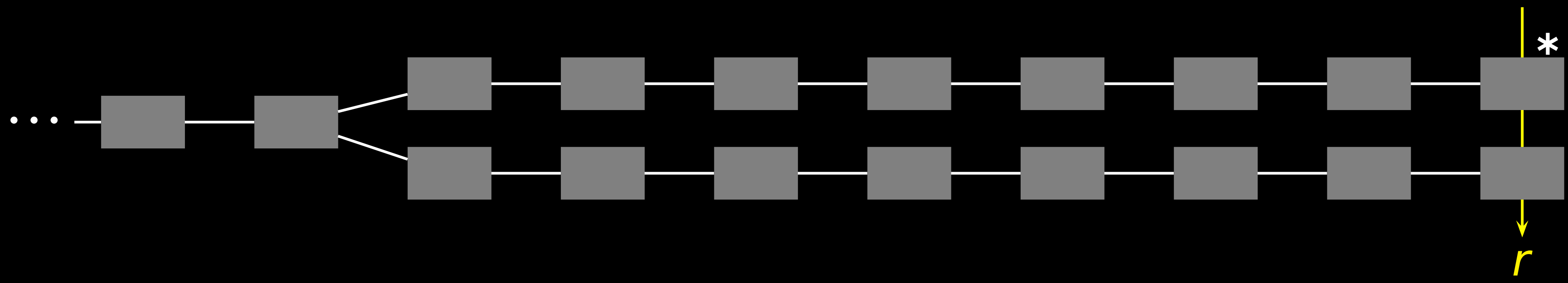
Common-Prefix Lemma

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

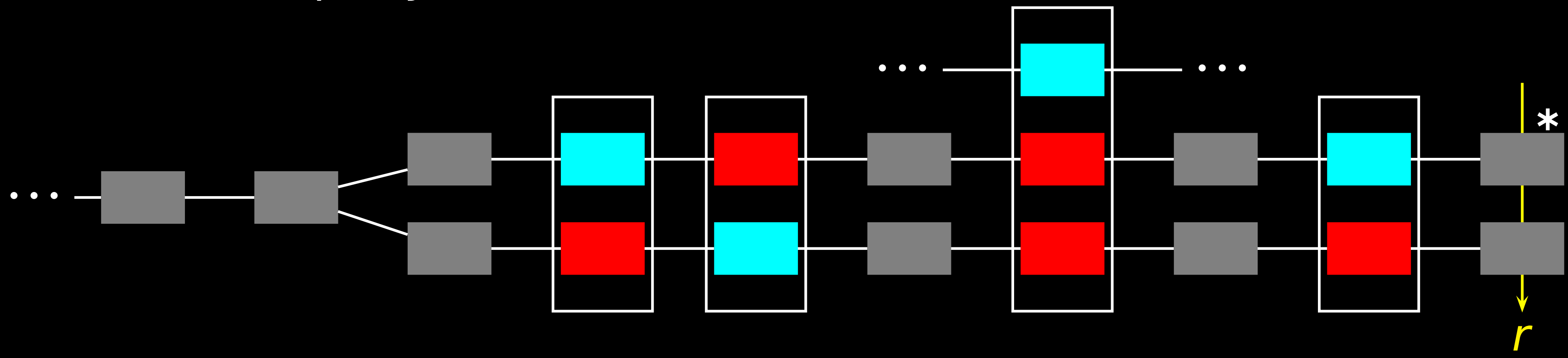
Common-Prefix Lemma. *The probability that at a given round two parties have chains that disagree in the last k blocks, is at most $e^{-\Omega(k)}$. (The party with the shortest chain should be honest.)*



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]

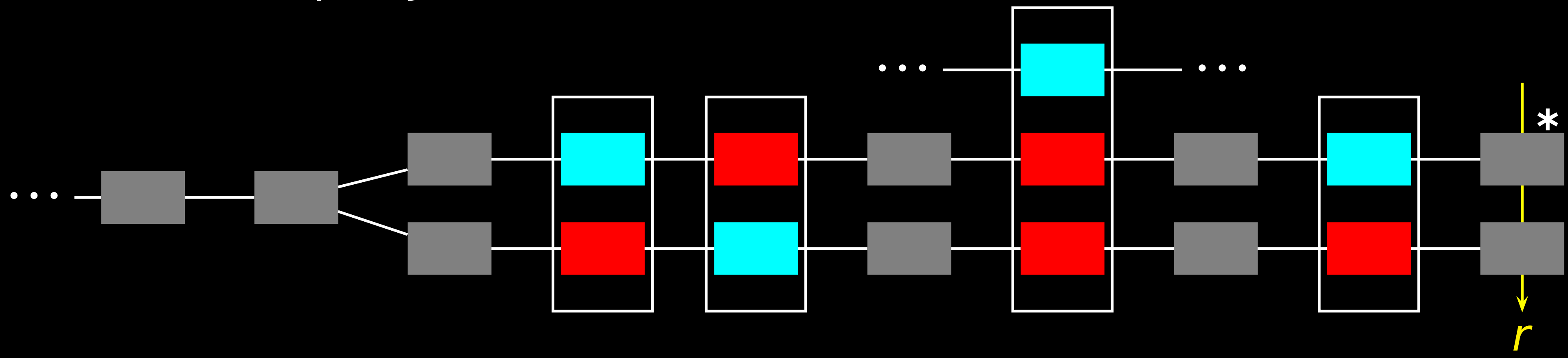
Common-Prefix Lemma. *The probability that at a given round two parties have chains that disagree in the last* k *blocks, is at most* $e^{-\Omega(k)}$. *(The party with the shortest chain should be honest.)*



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.)*

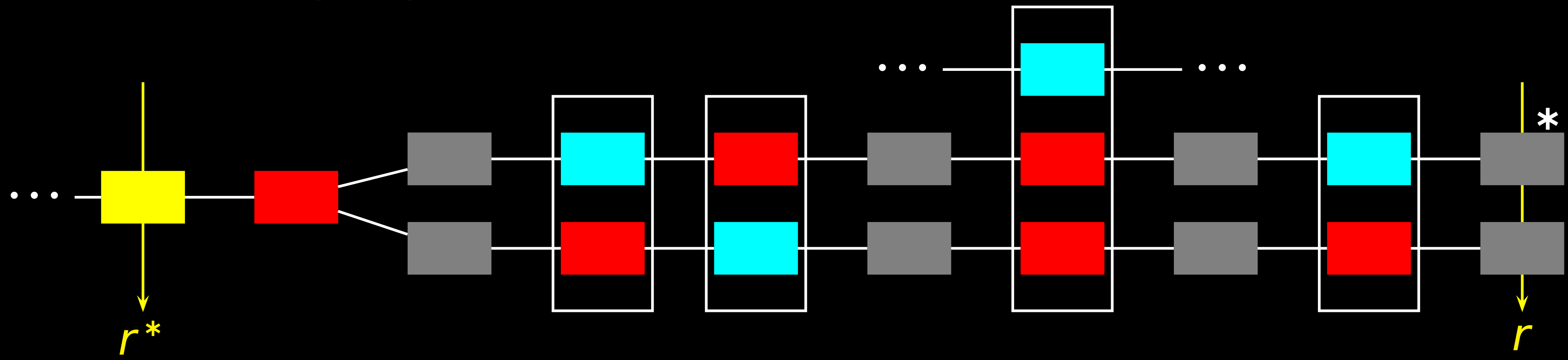


Observation. *Suppose the ℓ-the block of a chain was computed by an honest party in a uniquely successful round. Then any other ℓ-th block has been computed by the adversary.*

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]

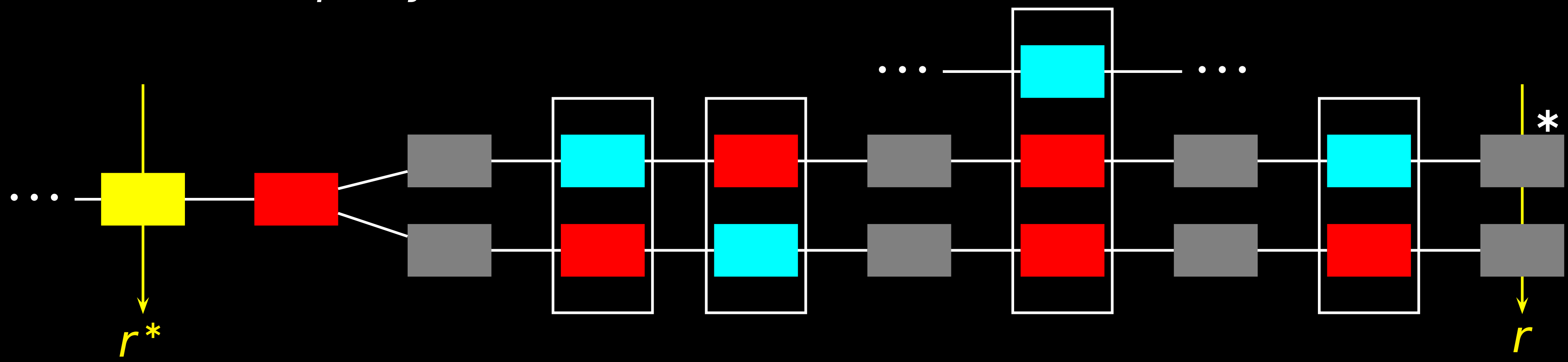
Common-Prefix Lemma. *The probability that at a given round two parties have chains that disagree in the last* k *blocks, is at most* $e^{-\Omega(k)}$. *(The party with the shortest chain should be honest.)*



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

Proof of the common-prefix lemma [GKL15]

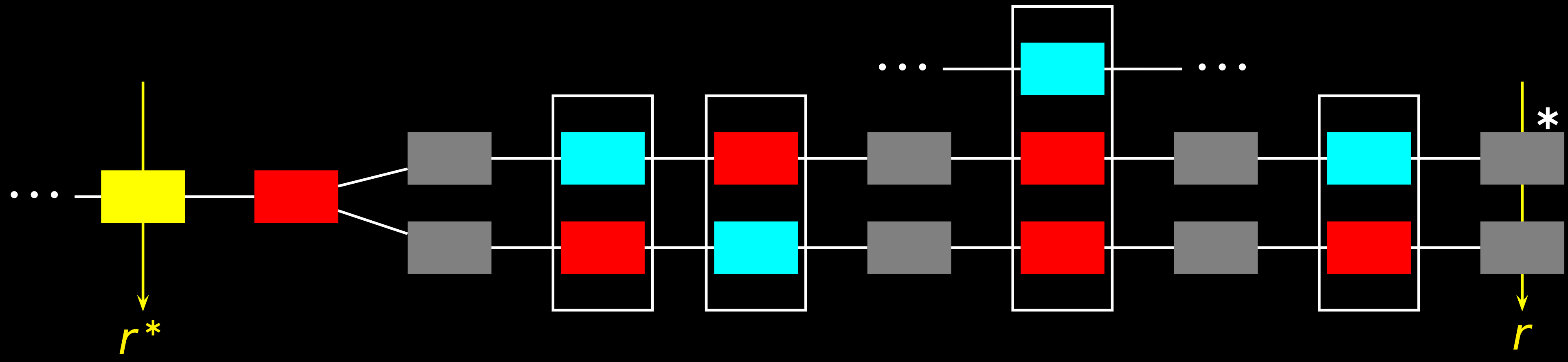
Common-Prefix Lemma. *The probability that at a given round two parties have chains that disagree in the last* k *blocks, is at most* $e^{-\Omega(k)}$. *(The party with the shortest chain should be honest.)*



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

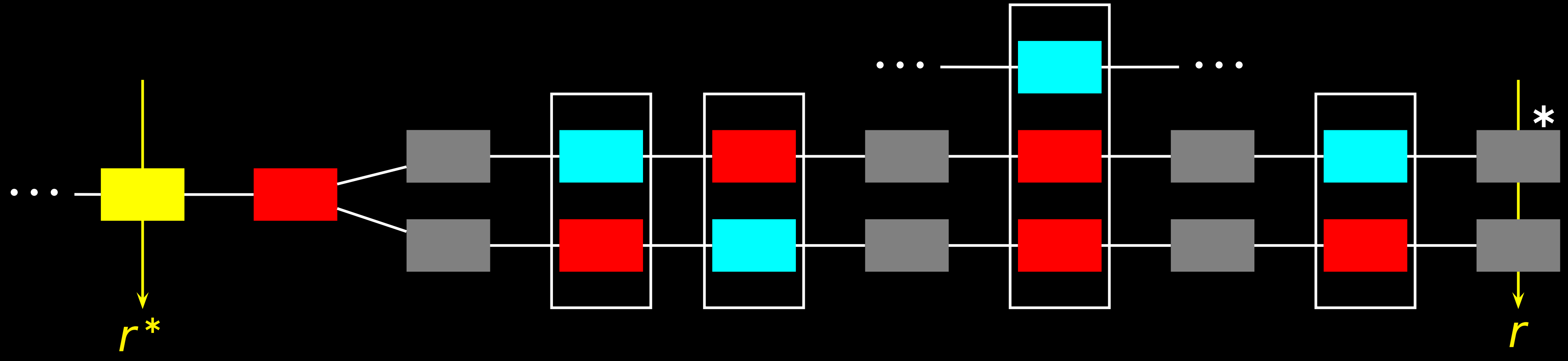


Proof. Let r^* be the last round in which a block before the fork was computed by an honest party. Set $S = \{r^* + 1, \ldots, r - 1\}$. By the Observation, to every uniquely successful round in S corresponds an adversarial block computed in S. It follows that

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.
$E[\sum Y_i] \approx pn(1-f)\|S\|$		$E[\sum Z_i] = pt\|S\|$.

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,

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

Similarly

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

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,

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

Similarly

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

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

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

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

□

Chain Quality

Chain Quality. *For any* ℓ *(sufficiently many) blocks in the chain of an honest party, the ratio of adversarial blocks is at most*

$$(1+\epsilon)\cdot\frac{t}{n}.$$

Chain Quality

Chain Quality. *For any ℓ (sufficiently many) blocks in the chain of an honest party, the ratio of adversarial blocks is at most*

$$(1+\epsilon)\cdot\frac{t}{n}.$$

Compare to the ideal ratio $t/(n+t)$.

Chain Quality

Chain Quality. *For any ℓ (sufficiently many) blocks in the chain of an honest party, the ratio of adversarial blocks is at most*

$$(1+\epsilon)\cdot\frac{t}{n}.$$

Corollary. *If $t < (1-\epsilon)n$, there is at least one honest block among any ℓ consecutive blocks in the chain of an honest party.*

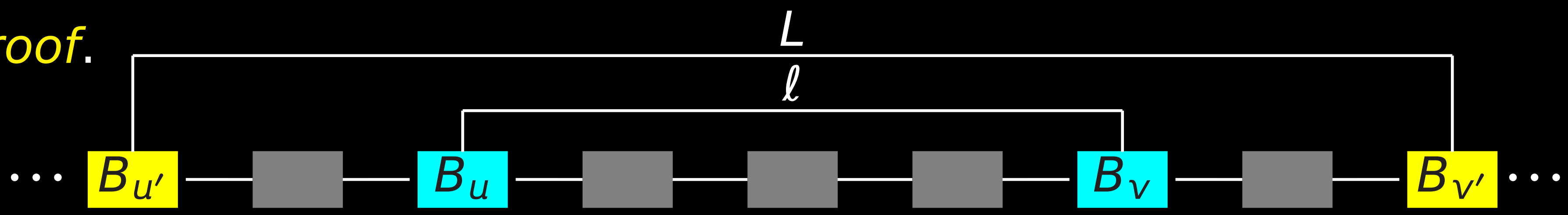
Proof. The ratio of adversarial blocks is less than $(1+\epsilon)(1-\epsilon) < 1$.

Chain Quality

Chain Quality. *For any ℓ (sufficiently many) blocks in the chain of an honest party, the ratio of adversarial blocks is at most*

$$(1+\epsilon)\cdot\frac{t}{n}.$$

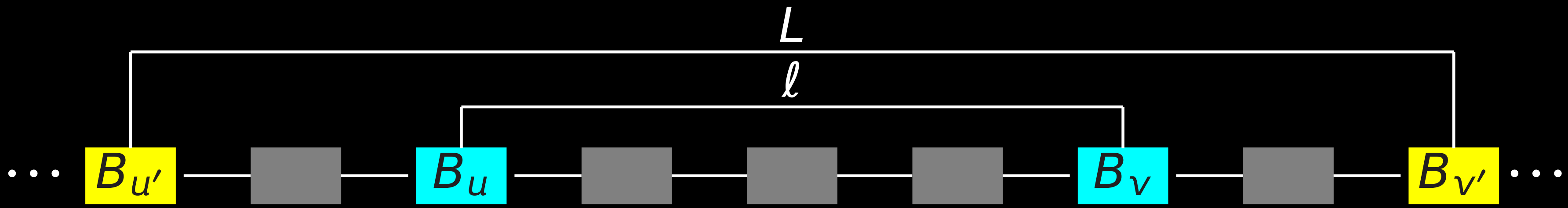
Proof.



- u' is greatest such that $B_{u'}$ was computed by an honest party.
- v' is least such that there exists a round at which an honest party was trying to extend the chain ending at block $B_{v'}$.
- r_1 is the round that $B_{u'}$ was created.
- r_2 first round that an honest party attempts to extend $B_{v'}$.
- $S = \{r : r_1 < r < r_2\}$.

Proof of Chain-Quality Property

Proof Cont'd.



We may assume that all the L blocks have been computed during the rounds in the set S.

- The number of successful rounds is at least $X \geq (1 - \frac{\epsilon}{3})pn|S|$.
- The number of adversarial blocks is at most $Z \leq (1 + \frac{\epsilon}{3})pt|S|$.
- Chain growth implies that $L \geq X$.
- The fraction of adversarial blocks is at most

$$\frac{Z}{L} \leq \frac{Z}{X} \leq \frac{1 + \frac{\epsilon}{3}}{1 - \frac{\epsilon}{3}} \cdot \frac{t}{n} \leq (1 + \epsilon) \cdot \frac{t}{n}.$$

□

Tightness of Chain Quality

Theorem. *There exists an adversary such that, with probability at least* $1 - e^{-\Omega(\epsilon^2 \ell)}$ *($\ell = \Omega(1/\epsilon)$), there will be* ℓ *consecutive blocks in the chain of every honest party in which the fraction of adversarial blocks is at least*

$$\frac{t}{n} - 2\epsilon.$$

Tightness of Chain Quality

Theorem. *There exists an adversary such that, with probability at least* $1 - e^{-\Omega(\epsilon^2 \ell)}$ *(*$\ell = \Omega(1/\epsilon)$*), there will be* ℓ *consecutive blocks in the chain of every honest party in which the fraction of adversarial blocks is at least*

$$\frac{t}{n} - 2\epsilon.$$

A selfish mining attack.

- The adversary keeps on extending a private chain.
- Whenever an honest party finds a solution, the (rushing) adversary releases one block from the private chain.
- If the private chain is depleted the adversary returns to the public chain.

Tightness of Chain Quality

Theorem. *There exists an adversary such that, with probability at least* $1 - e^{-\Omega(\epsilon^2 \ell)}$ *(*$\ell = \Omega(1/\epsilon)$*), there will be* ℓ *consecutive blocks in the chain of every honest party in which the fraction of adversarial blocks is at least*

$$\frac{t}{n} - 2\epsilon.$$

A selfish mining attack.

- The adversary keeps on extending a private chain.
- Whenever an honest party finds a solution, the (rushing) adversary releases one block from the private chain.
- If the private chain is depleted the adversary returns to the public chain.

Assumption. Ties between chains of equal length always favor the adversary.

Analysis of the Selfish Mining Attack

- Consider a set S of at least $\ell/(1-\epsilon)pn$ consecutive rounds.
  - This implies $X(S) \leq \ell$ (recall Chain-Growth Property).
- The number Z of adversarial blocks is at least $\frac{t}{n} \cdot \ell$.
- The number Z' of orphaned adversarial blocks computed in S is at most $\epsilon\ell$ with high probability.
  - k adversarial blocks may be orphaned, only if an honest party computes $k+1$ sequential blocks.
- The number Z'' of adversarial blocks not released in S is at most $\epsilon^2\ell$ with high probability.
  - k adversarial blocks are not released, only if no honest party computed a block in the meantime.

The ratio of adversarial blocks is at least

$$\frac{Z - Z' - Z''}{X} \geq \frac{\frac{t}{n} \cdot \ell - \epsilon\ell - \epsilon^2\ell}{\ell} \geq \frac{t}{n} - 2\epsilon \qquad \square$$

Byzantine agreement (consensus)

A set of parties $\{1, ..., n\}$, t of which are controlled and coordinated by an adversary. Parties have inputs $x_1, ..., x_n \in \{0, 1\}$ and want to decide on outputs $v_1, ..., v_n$ so that the following conditions are satisfied.

- **Agreement:** All honest parties decide on the same value (i.e., if i and j are honest, then $v_i = v_j$).
- **Validity:** If all honest parties have the same input value x, then all honest parties decide x (i.e., if i is honest, then $v_i = x$).
- **Termination:** All honest processes should terminate.

Byzantine agreement (consensus)

A set of parties $\{1, ..., n\}$, t of which are controlled and coordinated by an adversary. Parties have inputs $x_1, ..., x_n \in \{0, 1\}$ and want to decide on outputs $v_1, ..., v_n$ so that the following conditions are satisfied.

- **Agreement:** All honest parties decide on the same value (i.e., if i and j are honest, then $v_i = v_j$).
- **Validity:** If all honest parties have the same input value x, then all honest parties decide x (i.e., if i is honest, then $v_i = x$).
- **Termination:** All honest processes should terminate.

Remark. *Note that n here is the total number of parties.*

Byzantine Agreement: Fundamental Results

- One of the classical problems in distributed computing, a variant of which was first introduced in "Reaching Agreement in the Presence of Faults" [Pease-Shostak-Lamport 1980].
- Requires $n > 3t$, unless cryptography is used [PSL].
- Even with cryptographic tools, at least $t + 1$ rounds are needed [Fischer-Lynch and Dolev-Strong 1982].
- In an asynchronous or anonymous network no deterministic protocol exists [Fischer-Lynch-Paterson 1985]. But possible with probability 1 [Ben-Or 1983].
- Bit complexity is $\Omega(nt)$ [Dolev-Reischuk 1985].
- Fully Polynomial: There exists a protocol for all $t < \frac{n}{3}$, that terminates in $t+1$ rounds, and both computation and communication are polynomial in n. [Garay, Moses, "Fully polynomial Byzantine agreement for $n > 3t$ processors in $t + 1$ rounds." 1998]

Byzantine Agreement: Toy Proof

When 1 party out of n might be Byzantine, at least 2 rounds are needed.

- Upon receiving $00\ldots001$, an honest party should output 0.
  - Because of validity, since party p_n could be Byzantine.
- Upon receiving $00\ldots011$, an honest party should output 0.
  - Because party p_{n-1} could be Byzantine, and some parties might have received $00\ldots001$ and going to answer 0.
- Upon receiving $00\ldots0111$, an honest party should output 0.
  - Because party p_{n-2} could be Byzantine, and some parties might have received $00\ldots011$ and going to answer 0.

$$\vdots$$

- Upon receiving $01\ldots111$, an honest party should output 0.

Contradiction! Because the first party could be Byzantine.

Consensus: $t < n/2$ necessary (even with crypto)

Proof. On input $0 \ldots 0, 1 \ldots, 1$, where there are $n/2$ zeroes and $n/2$ ones and all parties are honest, the protocol terminates in one of the following three states.

A. All honest parties output 0.

B. All honest parties output 1.

C. Honest parties have mixed outputs.

The adversary chooses a strategy as follows.

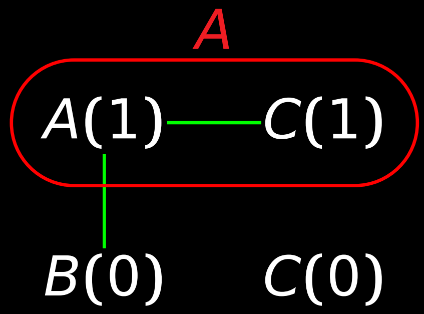
- In case A, he corrupts the first half of parties and behaves honestly. Validity fails.
- In case B, he corrupts the second half of parties and behaves honestly. Validity fails.
- In case C, he does not corrupt any party. Agreement fails.

Consensus: $t < n/3$ is necessary with delays

$A(mute)$ $B(0)$ $C(0)$	Since B and C have 0 and A might have crashed, at some time t_A parties B and C should terminate with 0.
$A(1)$ $B(mute)$ $C(1)$	Since A and C have 1 and B might have crashed, at some time t_B parties A and C should terminate with 1.
$A(1)$ $B(0)$ $C(*)$	Adversary C talks to A as if he has a 1 and to B as if he has a 0. Meanwhile, he holds messages $A \longleftrightarrow B$ for $t_C > t_A + t_B$ rounds.

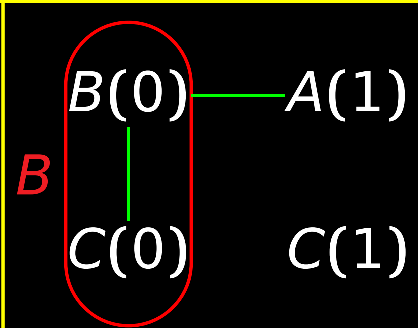
Consensus: $t < n/3$ is necessary (no delays)

World 1



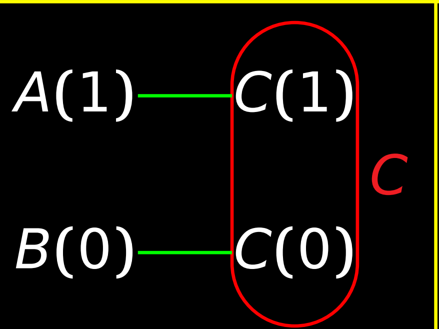
Adversary *A* tries to confuse *B* by acting as if *C* is Byzantine and talks to *A* as if its input is 1 and to *B* as if it is 0.

World 2



Adversary *B* tries to confuse *A* by acting as if *C* is Byzantine and talks to *B* as if its input is 0 and to *A* as if it is 1.

World 3

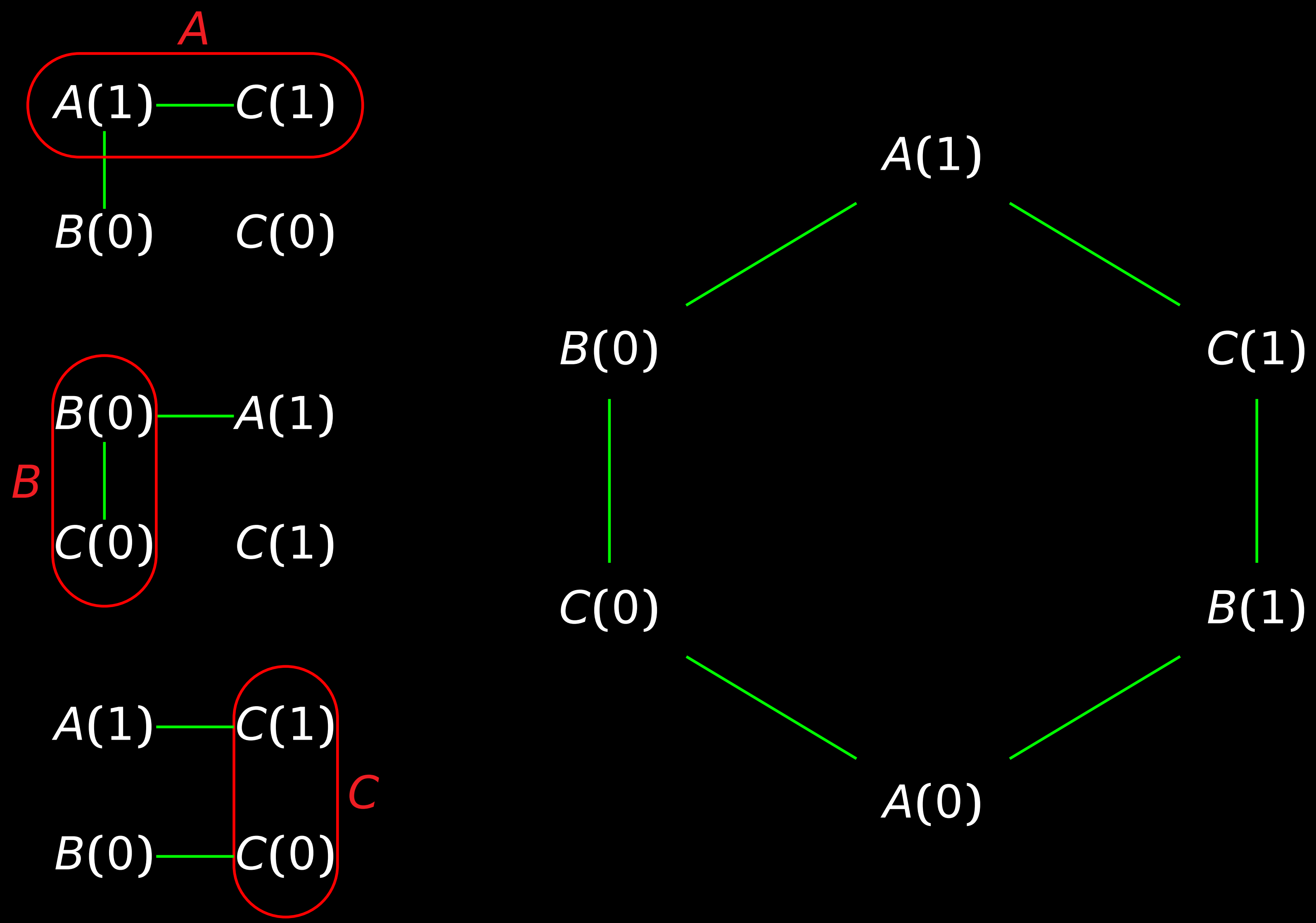


Adversary *C* talks to *A* as if he has a 1 and to *B* as if he has a 0, in a way that:

- *A* cannot distinguish between worlds 2 and 3
- *B* cannot distinguish between worlds 1 and 3

Contradiction! If Validity holds in worlds 1 and 2, then Agreement fails in world 3.

Feasibility of the strategy. The Hexagon idea!



Nakamoto's insight

Re: Bitcoin P2P e-cash paper

Satoshi Nakamoto | Thu, 13 Nov 2008 19:34:25 -0800

```
James A. Donald wrote:
> It is not sufficient that everyone knows X. We also
> need everyone to know that everyone knows X, and that
> everyone knows that everyone knows that everyone knows X
> - which, as in the Byzantine Generals problem, is the
> classic hard problem of distributed data processing.

The proof-of-work chain is a solution to the Byzantine Generals' Problem.  I'll
try to rephrase it in that context.

A number of Byzantine Generals each have a computer and want to attack the
King's wi-fi by brute forcing the password, which they've learned is a certain
number of characters in length.  Once they stimulate the network to generate a
packet, they must crack the password within a limited time to break in and
erase the logs, otherwise they will be discovered and get in trouble.  They
only have enough CPU power to crack it fast enough if a majority of them attack
at the same time.

They don't particularly care when the attack will be, just that they all agree.
 It has been decided that anyone who feels like it will announce a time, and
whatever time is heard first will be the official attack time.  The problem is
```

https://www.mail-archive.com/cryptography@metzdowd.com/msg09997.html

Byzantine Agreement Protocol

Theorem [GKL2015]. *Assuming $t < n/3$, the following protocol terminates after $\Theta(k)$ rounds in expectation and solves consensus with probability at least $1 - e^{-\Omega(k)}$.*

Byzantine Agreement Protocol

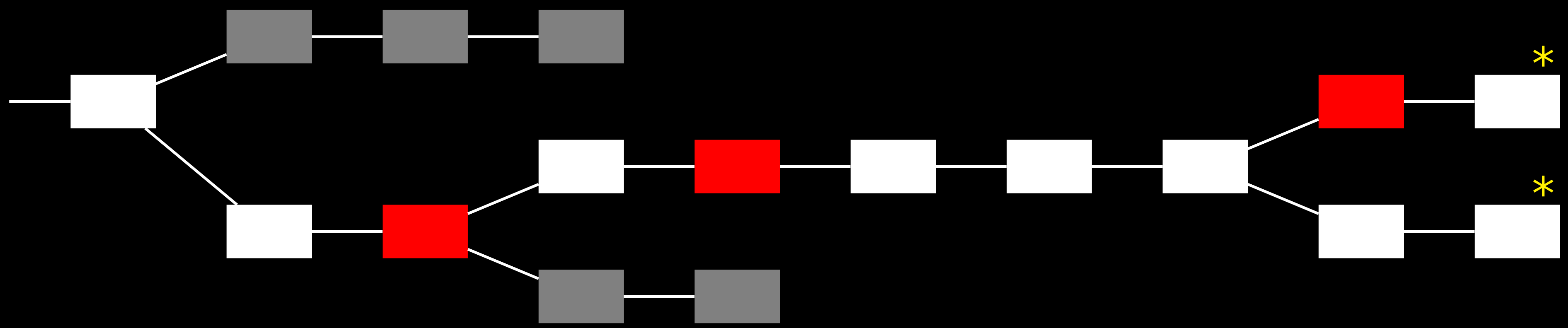
Theorem [GKL2015]. *Assuming $t < n/3$, the following protocol terminates after $\Theta(k)$ rounds in expectation and solves consensus with probability at least $1 - e^{-\Omega(k)}$.*

1) Parties run the Bitcoin protocol, putting their own input-bit in every block they compute.
2) When they obtain a chain with length $\geq 2k$ they halt (after they diffuse it).
3) Each party decides on the output equal to the majority of the inputs recorded in the first k blocks.

Byzantine Agreement Protocol

Theorem [GKL2015]. *Assuming $t < n/3$, the following protocol terminates after $\Theta(k)$ rounds in expectation and solves consensus with probability at least $1 - e^{-\Omega(k)}$.*

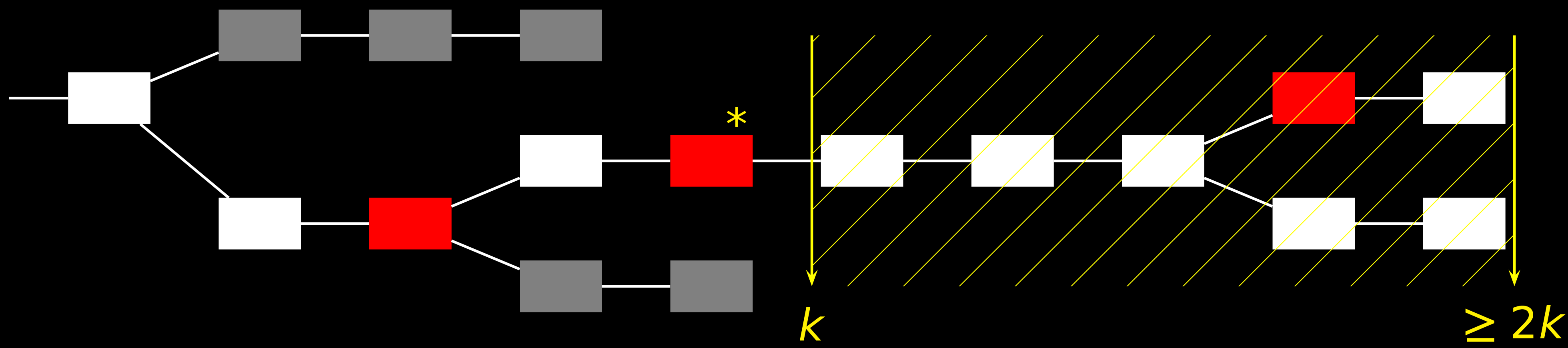
1) Parties run the Bitcoin protocol, putting their own input-bit in every block they compute.
2) When they obtain a chain with length $\geq 2k$ they halt (after they diffuse it).
3) Each party decides on the output equal to the majority of the inputs recorded in the first k blocks.



Byzantine Agreement Protocol

Theorem [GKL2015]. *Assuming $t < n/3$, the following protocol terminates after $\Theta(k)$ rounds in expectation and solves consensus with probability at least $1 - e^{-\Omega(k)}$.*

1) Parties run the Bitcoin protocol, putting their own input-bit in every block they compute.
2) When they obtain a chain with length $\geq 2k$ they halt (after they diffuse it).
3) Each party decides on the output equal to the majority of the inputs recorded in the first k blocks.



Proof of Agreement and Validity

- By the common-prefix property, if the adversary has less than half of the total computational power, Agreement is satisfied with high probability.

  This is because every honest party will output the majority of the input-bits included in the common prefix of their (possibly different) chains. (Consider the first time an honest party has a chain of length at least $2k$.)

Proof of Agreement and Validity

- By the common-prefix property, if the adversary has less than half of the total computational power, Agreement is satisfied with high probability.

  This is because every honest party will output the majority of the input-bits included in the common prefix of their (possibly different) chains. (Consider the first time an honest party has a chain of length at least $2k$.)

- By the chain-quality property, if the adversary has less than one third of the total computational power, Validity is satisfied with high probability.

  This is because out of the k bits of the common prefix, the adversary has computed less than half of them. Therefore, if all the honest parties have the same input x, the majority of the bits in the common prefix will be x.

2-for-1 PoWs

Idea. Two kinds of blocks with a single query. Recall $H : \{0,1\}^* \to \{0,1\}^\kappa$.

- **Normal blocks:** $H(x) < T = 2^a$.
- **Input blocks:** $[H(x)]^R < T' = 2^b$.

Here, $[y]^R$ is the number with binary expansion the reverse of y.

2-for-1 PoWs

Idea. Two kinds of blocks with a single query. Recall $H : \{0,1\}^* \to \{0,1\}^\kappa$.

- Normal blocks: $H(x) < T = 2^a$.
- Input blocks: $[H(x)]^R < T' = 2^b$.

Here, $[y]^R$ is the number with binary expansion the reverse of y.

Observation. As long as $a + b > \kappa$, the probabilities of obtaining a block of each kind are independent.

Proof. Let U random over $\{0,1\}^\kappa$. Conditioning on $U < T$ leaves the a least significant bits of U random, while fixing the remaining $\kappa - a$ bits. Thus, the $a > \kappa - b$ most significant bits of U^R are random. It follows that

$$\Pr[U^R < T' | U < T] = \frac{2^{a-(\kappa-b)}}{2^a} = \frac{2^b}{2^\kappa} = \frac{T'}{2^\kappa} = \Pr[U^R < T'].$$

1/2-resilient consensus

Theorem [GKL2015]. *Assuming* $t < n/2$, *the following protocol terminates after* $\Theta(k)$ *rounds in expectation and solves consensus with probability at least* $1 - e^{-\Omega(k)}$.

1/2-resilient consensus

Theorem [GKL2015]. *Assuming $t < n/2$, the following protocol terminates after $\Theta(k)$ rounds in expectation and solves consensus with probability at least $1 - e^{-\Omega(k)}$.*

1) Parties run the Bitcoin protocol, putting their own input-bit in every input block they compute and referencing all input blocks from the normal blocks they compute.

2) When they obtain a chain with at least $\frac{3k}{\delta} + 2k$ normal blocks they halt (after they diffuse it).

3) Each party decides on the output equal to the majority of the unique inputs recorded in the first $\frac{3k}{\delta} + k$ normal blocks.

1/2-resilient consensus

Theorem [GKL2015]. *Assuming $t < n/2$, the following protocol terminates after $\Theta(k)$ rounds in expectation and solves consensus with probability at least $1 - e^{-\Omega(k)}$.*

1) Parties run the Bitcoin protocol, putting their own input-bit in every input block they compute and referencing all input blocks from the normal blocks they compute.

2) When they obtain a chain with at least $\frac{3k}{\delta} + 2k$ normal blocks they halt (after they diffuse it).

3) Each party decides on the output equal to the majority of the unique inputs recorded in the first $\frac{3k}{\delta} + k$ normal blocks.

Agreement follows from Common-Prefix Property because at least k blocks are pruned.

Proof for validity (sketch)

- Let C denote the prefix of the first $\frac{3k}{\delta} + 2k$ normal blocks.
- By Chain-Quality Property, the last k of C contain an honest normal block B, computed at some round r.
- Note that B contains all honest input blocks computed in $S = \{1, 2, \ldots, r\}$. Let $X(S)$ denote their number and $Z(S')$ the adversarial input blocks referenced.
- Thus,

$$\frac{Z(S')}{X(S)} < \frac{(1+\epsilon)pt|S'|}{(1-\epsilon)f|S|} < \frac{(1+\epsilon)(1-\delta)pn|S'|}{(1-\epsilon)(1-f)pn|S|} \leq \cdots \leq 1,$$

as long as δ and S are big enough.

Bounded-Delay Model

The adversary may delay the delivery of a message for at most Δ rounds. That is, a message diffuse at round r may be delivered at round $r + \Delta$ (but not later).

Bounded-Delay Model

The adversary may delay the delivery of a message for at most Δ rounds. That is, a message diffuse at round r may be delivered at round $r+\Delta$ (but not later).

Δ-isolated uniquely-successful round.

$$Y_i' = 1 \text{ if } Y_i = 1 \text{ and } X_j = 0 \text{ for } j \neq i \text{ with } |j-i| < \Delta$$

$$\mathbf{E}[Y_i'] \geq f(1-f)^{2\Delta-1} \geq f[1-(2\Delta-1)f]$$

Bounded-Delay Model

The adversary may delay the delivery of a message for at most Δ rounds. That is, a message diffuse at round r may be delivered at round $r+\Delta$ (but not later).

Δ-isolated uniquely-successful round.

$$Y_i' = 1 \text{ if } Y_i = 1 \text{ and } X_j = 0 \text{ for } j \neq i \text{ with } |j-i| < \Delta$$

$$\mathbf{E}[Y_i'] \geq f(1-f)^{2\Delta-1} \geq f[1-(2\Delta-1)f]$$

Δ-isolated successful round.

$$X_i' = 1 \text{ if } X_i = 1 \text{ and } X_j = 0 \text{ for } i-\Delta < j < i$$

$$\mathbf{E}[X_i'] \geq f(1-f)^{\Delta-1} \geq f[1-(\Delta-1)f]$$

Bounded-Delay Model

The adversary may delay the delivery of a message for at most Δ rounds. That is, a message diffuse at round r may be delivered at round $r+\Delta$ (but not later).

Δ-isolated uniquely-successful round.

$$Y_i' = 1 \text{ if } Y_i = 1 \text{ and } X_j = 0 \text{ for } j \neq i \text{ with } |j-i| < \Delta$$

$$\mathbf{E}[Y_i'] \geq f(1-f)^{2\Delta-1} \geq f[1-(2\Delta-1)f]$$

Δ-isolated successful round.

$$X_i' = 1 \text{ if } X_i = 1 \text{ and } X_j = 0 \text{ for } i-\Delta < j < i$$

$$\mathbf{E}[X_i'] \geq f(1-f)^{\Delta-1} \geq f[1-(\Delta-1)f]$$

Remark. These definitions are not tight. In particular, we could do with a set of uniquely successful rounds such that any two are Δ-far away from each other.

Chain-Growth Lemma

Chain-Growth Lemma. *Suppose that at round u an honest party has a chain of length ℓ. Then, by round $v \geq u + \Delta - 1$, every honest party has adopted a chain of length at least* $\ell' = \ell + X'_u + \cdots + X'_{v-\Delta}$.

Chain-Growth Lemma

Chain-Growth Lemma. *Suppose that at round u an honest party has a chain of length ℓ. Then, by round $v \geq u + \Delta - 1$, every honest party has adopted a chain of length at least* $\ell' = \ell + X'_u + \cdots + X'_{v-\Delta}$.

Proof. By induction on v.

Basis ($v = u + \Delta - 1$). If at round u an honest party has a chain C of length ℓ, then that party diffuses C at a round earlier than u. It follows that every honest party will receive C by round $u - 1 + \Delta = v$.

Case $X'_{v-\Delta} = 0$. By hypothesis, every honest party has received a chain of length at least $\ell + X'_u + \cdots + X'_{v-\Delta-1} = \ell'$ by round $v - 1$.

Chain-Growth Lemma

Chain-Growth Lemma. *Suppose that at round* u *an honest party has a chain of length* ℓ*. Then, by round* $v \geq u + \Delta - 1$*, every honest party has adopted a chain of length at least* $\ell' = \ell + X'_u + \cdots + X'_{v-\Delta}$*.*

Proof. By induction on v.

Basis ($v = u + \Delta - 1$). If at round u an honest party has a chain C of length ℓ, then that party diffuses C at a round earlier than u. It follows that every honest party will receive C by round $u - 1 + \Delta = v$.

Case $X'_{v-\Delta} = 0$. By hypothesis, every honest party has received a chain of length at least $\ell + X'_u + \cdots + X'_{v-\Delta-1} = \ell'$ by round $v - 1$.

Case $X'_{v-\Delta} = 1$. By hypothesis, by round $v - \Delta$, every honest party has adopted a chain of length at least

$$\ell + X'_u + \cdots + X'_{v-2\Delta} = \ell + X'_u + \cdots + X'_{v-\Delta-1} = \ell' - 1.$$

Hence, all honest parties successful at round $v - \Delta$ broadcast a chain of length at least ℓ'. This chain will be received by every honest party by round v.

Concentration for Lipschitz functions

- Note that Y'_i and Y'_j are not independent anymore when $|i - j| < 2\Delta$ and the standard Chernoff bound does not apply. (Similarly for X'_i and X'_j.)

Concentration for Lipschitz functions

- Note that Y'_i and Y'_j are not independent anymore when $|i - j| < 2\Delta$ and the standard Chernoff bound does not apply. (Similarly for X'_i and X'_j.)

A function $f(x_1, \ldots, x_n)$ is k-Lipschitz if $|f(x) - f(x')| \leq k$, whenever x and x' differ in at most one coordinate.

Theorem. *If f is k-Lipschitz and $X_1, \ldots, X_n$ are independent random variables, then*

$$\Pr[f > \mathbf{E}f + t] \leq \exp\left(-\frac{2t^2}{nk^2}\right) \quad \textit{and} \quad \Pr[f < \mathbf{E}f - t] \leq \exp\left(-\frac{2t^2}{nk^2}\right).$$

Concentration for Lipschitz functions

A function $f(x_1, ..., x_n)$ is k-Lipschitz if $|f(x) - f(x')| \leq k$, whenever x and x' differ in at most one coordinate.

Theorem. *If f is k-Lipschitz and $X_1, ..., X_n$ are independent random variables, then*

$$\Pr[f > \mathbf{E}f + t] \leq \exp\left(-\frac{2t^2}{nk^2}\right) \quad \textit{and} \quad \Pr[f < \mathbf{E}f - t] \leq \exp\left(-\frac{2t^2}{nk^2}\right).$$

- Each X'_i is a function of $X_{i-\Delta}, ..., X_i$.
- Thus, the sum $\sum_{i=\Delta}^{r} X'_i$ is a function of the independent random variables $X_1, X_2, ..., X_r$.
- Moreover, $\sum_{i=\Delta}^{r} X'_i$ is 2-Lipschitz. This is because X_j affects X'_i only if $j \leq i < j + \Delta$ and there can be at most two X'_i equal to 1 in an interval of length Δ.

Chain Quality

Chain Quality. *For any ℓ blocks in the chain of an honest party, the ratio of adversarial blocks is at most*

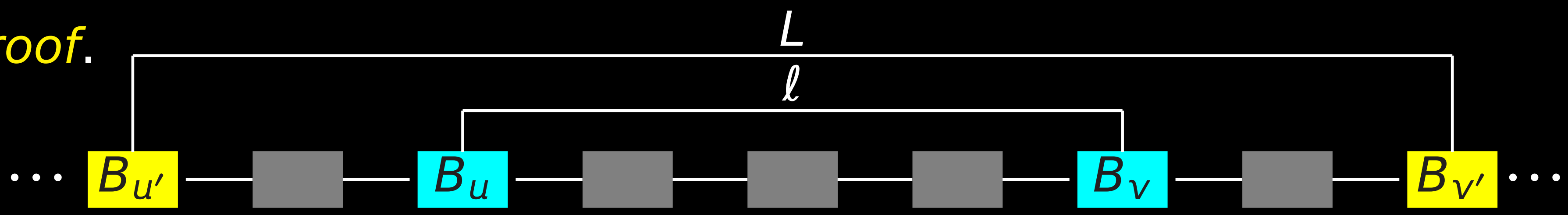
$$(1+\epsilon) \cdot \frac{t}{n}.$$

Chain Quality

Chain Quality. *For any ℓ blocks in the chain of an honest party, the ratio of adversarial blocks is at most*

$$(1+\epsilon)\cdot\frac{t}{n}.$$

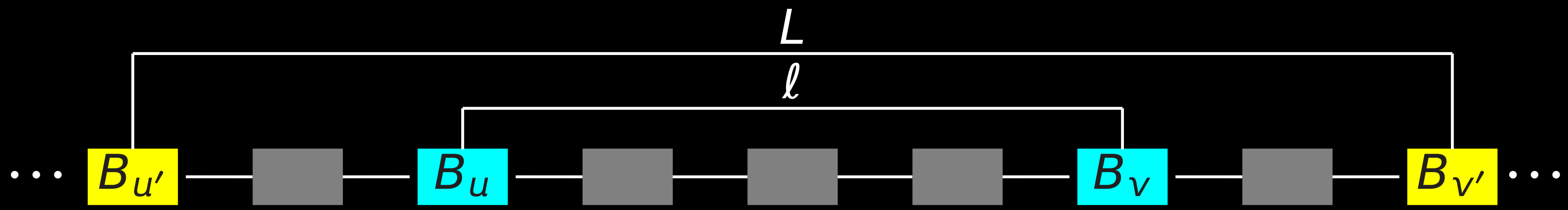
Proof.



- u' is greatest such that $B_{u'}$ was computed by an honest party.
- v' is least such that there exists a round at which an honest party was trying to extend the chain ending at block $B_{v'}$.
- r_1 is the round that $B_{u'}$ was created.
- r_2 first round that an honest party attempts to extend $B_{v'}$.
- $S = \{r : r_1 < r < r_2\}$.

Proof of Chain-Quality Property

Proof Cont'd.



We may assume that all the L blocks have been computed during the rounds in the set S.

- The number of successful rounds is at least $X \geq (1 - \frac{\epsilon}{3})pn(|S| - \Delta)$.
- The number of adversarial blocks is at most $Z \leq (1 + \frac{\epsilon}{3})pt|S|$.
- Chain growth implies that $L \geq X$.
- The fraction of adversarial blocks is at most

$$\frac{Z}{L} \leq \frac{Z}{X} \leq \frac{1 + \frac{\epsilon}{3}}{1 - \frac{\epsilon}{3}} \cdot \frac{t}{n} \cdot \left(1 - \frac{\Delta}{|S|}\right) \leq \cdots .$$

Choose ℓ large enough so that $\Delta/|S|$ is small enough.

Proof of the 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.)*

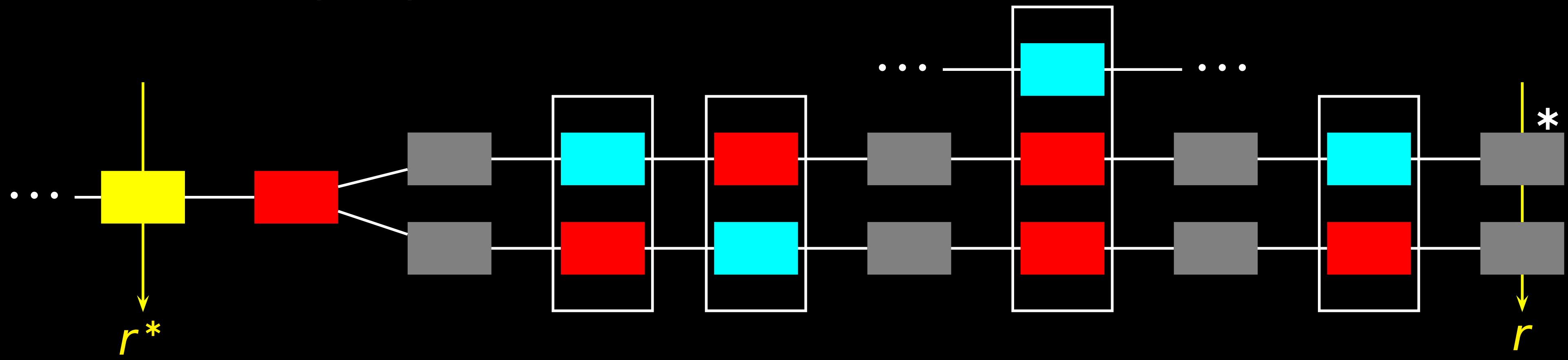
Observation. *Suppose the ℓ-the block of a chain was computed by an honest party in an isolated uniquely successful round. Then any other ℓ-th block has been computed by the adversary.*

Proof. Suppose a block of height ℓ was computed by an honest party at a round u with $Y'_u = 1$. This implies $X_r = 0$ for and $r \neq u$ with $|r - u| < \Delta$.

- Thus, no honest party could compute another block at a round r with $|r - u| < \Delta$.
- If any honest party computed a block of height ℓ at any round $r < u - \Delta$, then any honest party is trying to extend a chain of length at least ℓ at round u.
- Similarly for $r > u + \Delta$.

Proof of the 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.)*

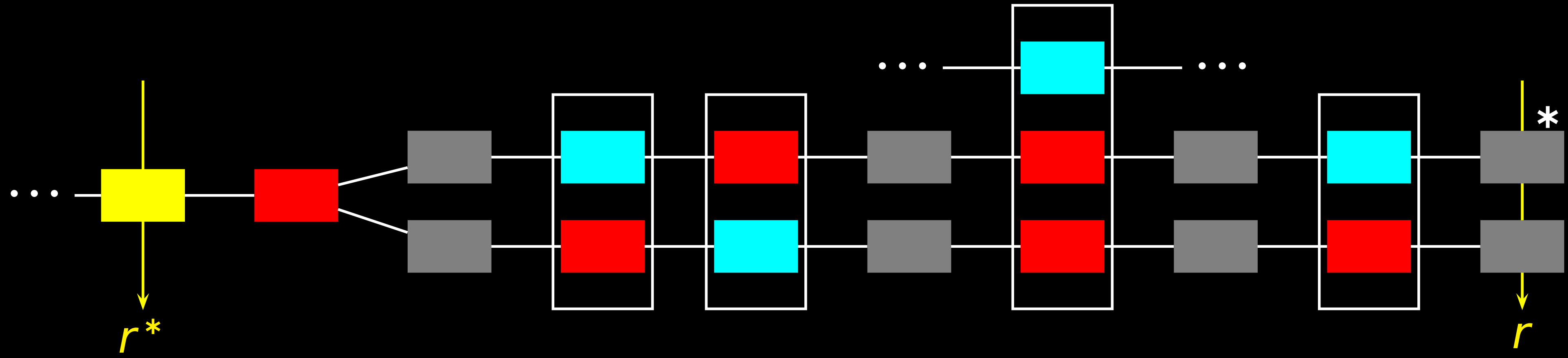


Proof. Let r^* be the last round before the fork that was computed by an honest party. Set

$$S = \{r^* + 1, \dots, r - 1\} \quad \text{and} \quad S' = \{r^* + 1 + \Delta, \dots, r - 1 - \Delta\}.$$

Proof of the 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.)*



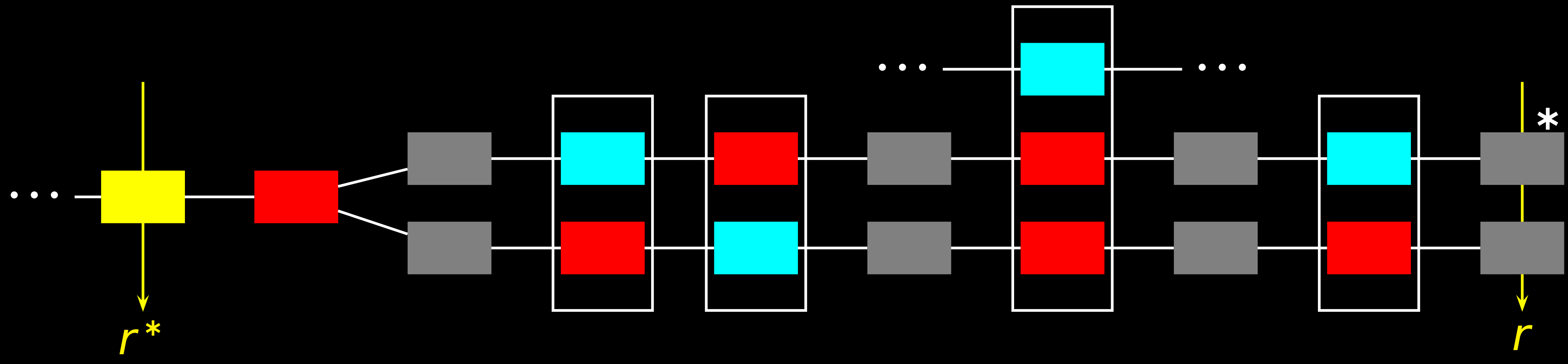
Proof. Let r^* be the last round before the fork that was computed by an honest party. Set

$$S = \{r^* + 1, \ldots, r - 1\} \quad \text{and} \quad S' = \{r^* + 1 + \Delta, \ldots, r - 1 - \Delta\}.$$

By the Observation, to every Δ-isolated uniquely successful round in S' corresponds an adversarial block computed in S.

Proof of the 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.)*



Proof. Let r^* be the last round before the fork that was computed by an honest party. Set

$$S = \{r^*+1, \ldots, r-1\} \quad \text{and} \quad S' = \{r^*+1+\Delta, \ldots, r-1-\Delta\}.$$

By the Observation, to every Δ-isolated uniquely successful round in S' corresponds an adversarial block computed in S. Thus,

Isolated uniqely successful rounds in S' $\leq$ Adversarial successes in S.

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

Recall $\mathbf{E}[Y_i'] > f(1-f)^{2\Delta-1}$. We can argue that $Y'(S') = \sum_{r\in S'} Y_r'$ is 2-Lipschitz. By the Concentration bound for Lipschitz functions,

$$\Pr[Y'(S') \leq (1-\epsilon)f(1-f)^{2\Delta-1}|S'|] = e^{-\Omega(|S|)}.$$

Similarly

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

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

Recall $\mathbf{E}[Y_i'] > f(1-f)^{2\Delta-1}$. We can argue that $Y'(S') = \sum_{r\in S'} Y_r'$ is 2-Lipschitz. By the Concentration bound for Lipschitz functions,

$$\Pr[Y'(S') \le (1-\epsilon)f(1-f)^{2\Delta-1}|S'|] = e^{-\Omega(|S|)}.$$

Similarly

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

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

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

Recall $\mathbf{E}[Y'_i] > f(1-f)^{2\Delta-1}$. We can argue that $Y'(S') = \sum_{r\in S'} Y'_r$ is 2-Lipschitz. By the Concentration bound for Lipschitz functions,

$$\Pr[Y'(S') \le (1-\epsilon)f(1-f)^{2\Delta-1}|S'|] = e^{-\Omega(|S|)}.$$

Similarly

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

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

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

$$\begin{aligned} Z(S) &< (1+\epsilon)pt|S| \\ &< (1+\epsilon)(1-\delta)pn|S'| \qquad (t < (1-\delta)n) \\ &< (1+\epsilon)(1-\delta)\cdot\frac{f}{1-f}\cdot|S'|\left(1+\frac{2\Delta}{|S'|}\right) \qquad ((1-f)pn < f) \\ &\ldots \text{(Making } 2\Delta/|S'| \text{ sufficiently small)} \ldots \\ &< Y'(S') \end{aligned}$$

Dynamic execution: number of parties is changing

- The proof in the static case (fixed number of parties) breaks.
  - As block-production rate goes to 1, persistence breaks.
  - As block-production rate goes to 0, liveness is hurt.

Dynamic execution: number of parties is changing

- The proof in the static case (fixed number of parties) breaks.
  - As block-production rate goes to 1, persistence breaks.
  - As block-production rate goes to 0, liveness is hurt.
- Actually, Bitcoin strives to maintain constant block-production rate of about 1 block per 10 mins.

Dynamic execution: number of parties is changing

- The proof in the static case (fixed number of parties) breaks.
  - As block-production rate goes to 1, persistence breaks.
  - As block-production rate goes to 0, liveness is hurt.
- Actually, Bitcoin strives to maintain constant block-production rate of about 1 block per 10 mins.
- The difficulty of producing a block can be calibrated by changing the target T.

  Note that we want to use this in a distributed manner.

  Bitcoin achieves (approximately) constant rate by having the target of the to-be-computed block determined (locally) by a fixed number of previous blocks.

Dynamic execution: number of parties is changing

- The proof in the static case (fixed number of parties) breaks.
  - As block-production rate goes to 1, persistence breaks.
  - As block-production rate goes to 0, liveness is hurt.
- Actually, Bitcoin strives to maintain constant block-production rate of about 1 block per 10 mins.
- The difficulty of producing a block can be calibrated by changing the target T.

  Note that we want to use this in a distributed manner.

  Bitcoin achieves (approximately) constant rate by having the target of the to-be-computed block determined (locally) by a fixed number of previous blocks.
- Each block now is associated with a target T and difficulty $\frac{1}{T}$.

  Parties now follow the heaviest chain.

Naive target recalculation

- The target is recalculated every m blocks.

  Bitcoin uses $m = 2016$ and calls the period between two recalculation points an epoch.

  If one wants to extend a chain of length λm, first determines T by the last m blocks.

Naive target recalculation

- The target is recalculated every m blocks.

  Bitcoin uses $m = 2016$ and calls the period between two recalculation points an epoch.

  If one wants to extend a chain of length λm, first determines T by the last m blocks.

- Informally, if the last m blocks were calculated quickly, then increase difficulty (decrease T), otherwise decrease difficulty (increase T).

Naive target recalculation

- The target is recalculated every m blocks.

  Bitcoin uses $m = 2016$ and calls the period between two recalculation points an epoch.

  If one wants to extend a chain of length λm, first determines T by the last m blocks.

- Informally, if the last m blocks were calculated quickly, then increase difficulty (decrease T), otherwise decrease difficulty (increase T).

- Suppose the last m blocks were computed in Λ rounds for target T. If we want to have m blocks in every $\frac{m}{f}$ rounds, set

$$T' = \frac{\Lambda}{m/f} \cdot T, \qquad (f = \text{block-production rate}).$$

  This is justified because for small f the relation between f and T is approximately linear.

Bahack's difficulty raising attack

- Suppose that at some round r the honest parties have a chain of length λm.
- The adversary builds the next epoch all by himself with fake timestamps, resulting in huge difficulty for the next epoch.
- His strategy is to set T' so small, so that if he computes the 1st block (a superblock of difficulty $\frac{1}{T'}$) of the next epoch fast (say half the expected time), he obtains a chain heavier than the chain the honest parties are expected to have by that time.
- This works with constant probability!

Bahack's difficulty raising attack

- Suppose that at some round r the honest parties have a chain of length λm.
- The adversary builds the next epoch all by himself with fake timestamps, resulting in huge difficulty for the next epoch.
- His strategy is to set T' so small, so that if he computes the 1st block (a superblock of difficulty $\frac{1}{T'}$) of the next epoch fast (say half the expected time), he obtains a chain heavier than the chain the honest parties are expected to have by that time.
- This works with constant probability!

But, Nakamoto knew this!!!

Analysis of the attack (sketch)

To see why this works, let us fix a target T for the honest parties and suppose the honest parties advance with success probability f and the adversary with $\frac{1}{1+\delta} \cdot f$ (for some $\delta < 1/2$).

- If the adversary sets $T' = \frac{T}{2\delta m}$, then with constant probability he finishes his attack (i.e., $(m+1)$ blocks) in

$$(1+\delta) \cdot \frac{m}{f} + (1+\delta) \cdot \frac{T}{T'} \cdot \frac{1}{3f}$$

  rounds and has collected difficulty

$$\frac{m}{T} + \frac{1}{T'} = \frac{m}{T} + \frac{2\delta m}{T} = (1+2\delta) \cdot \frac{m}{T}.$$

- The honest parties have collected (in expectation)

$$(1+\delta)\left(\frac{m}{T} + \frac{1}{3T'}\right) = (1+\delta)\left(\frac{m}{T} + \frac{2\delta m}{3T}\right) < (1+2\delta) \cdot \frac{m}{T}.$$

The adversary wins with constant probability!

Bitcoin's target recalculation

- Suppose the last m blocks were computed in Λ rounds for target T. If we want to have m blocks in every $\frac{m}{f}$ rounds, set

$$T' = \frac{\Lambda}{m/f} \cdot T,$$

Bitcoin's target recalculation

- Suppose the last m blocks were computed in Λ rounds for target T. If we want to have m blocks in every $\frac{m}{f}$ rounds, set

$$T' = \frac{\Lambda}{m/f} \cdot T,$$

unless $T' < T/4$ or $T' > 4T$, in which case set $T' = T/4$ or $T' = 4T$ accordingly.

Bitcoin's target recalculation

- Suppose the last m blocks were computed in Λ rounds for target T. If we want to have m blocks in every $\frac{m}{f}$ rounds, set

$$T' = \frac{\Lambda}{m/f} \cdot T,$$

unless $T' < T/4$ or $T' > 4T$, in which case set $T' = T/4$ or $T' = 4T$ accordingly.

- If the number of parties keeps increasing by a large factor per epoch, then target recalculation won't catch up.

Bitcoin's target recalculation

- Suppose the last m blocks were computed in Λ rounds for target T. If we want to have m blocks in every $\frac{m}{f}$ rounds, set

$$T' = \frac{\Lambda}{m/f} \cdot T,$$

  unless $T' < T/4$ or $T' > 4T$, in which case set $T' = T/4$ or $T' = 4T$ accordingly.

- If the number of parties keeps increasing by a large factor per epoch, then target recalculation won't catch up.

- Can we prove security under the assumption that the number of parties does not fluctuate wildly?

Bitcoin's target recalculation

- Suppose the last m blocks were computed in Λ rounds for target T. If we want to have m blocks in every $\frac{m}{f}$ rounds, set

$$T' = \frac{\Lambda}{m/f} \cdot T,$$

  unless $T' < T/4$ or $T' > 4T$, in which case set $T' = T/4$ or $T' = 4T$ accordingly.

- If the number of parties keeps increasing by a large factor per epoch, then target recalculation won't catch up.

- Can we prove security under the assumption that the number of parties does not fluctuate wildly?

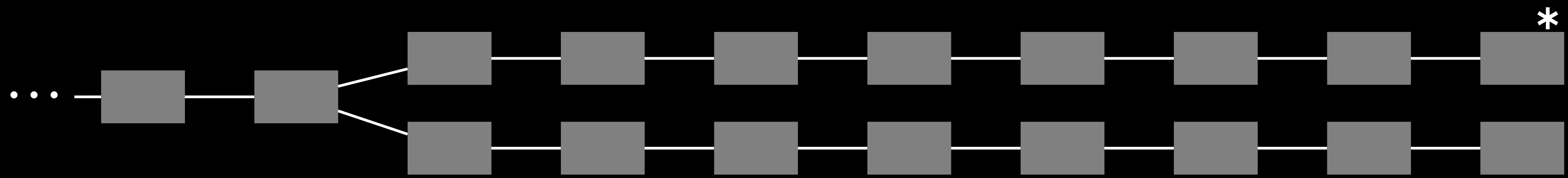
Theorem. *If, for appropriate parameters s and γ,*

$$\forall r, r' \qquad |r - r'| \leq s \quad \Longrightarrow \quad \frac{n_r}{\lambda} \leq n_{r'} \leq \lambda n_r,$$

then common prefix and chain quality hold (assuming adversarial minority and appropriate initialization).

The common-prefix lemma in the dynamic case

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.)*

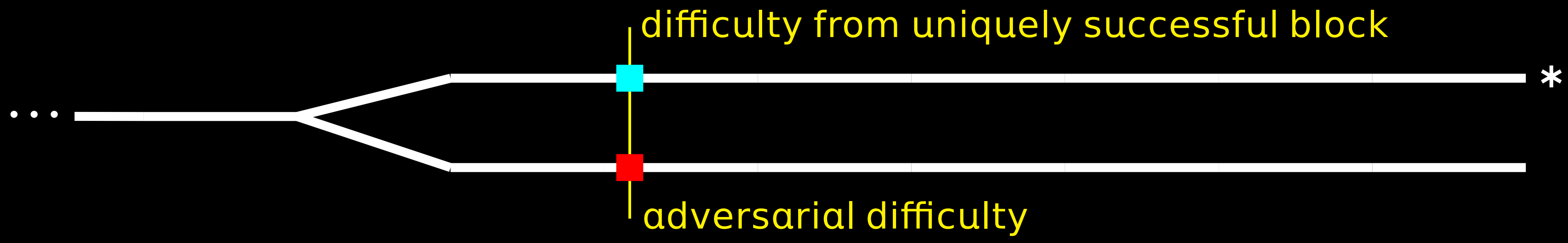


The common-prefix lemma in the dynamic case

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.)*

The common-prefix lemma in the dynamic case

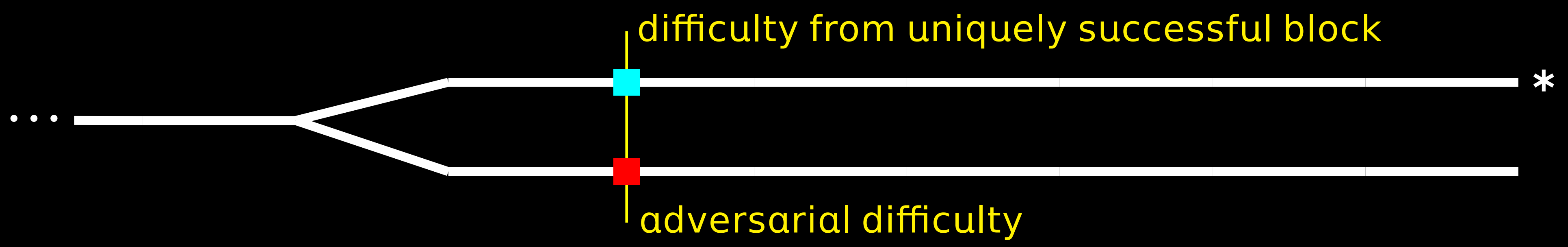
Common-Prefix Lemma. *The probability that at a given round two parties have chains that disagree in the last k blocks, is at most $e^{-\Omega(k)}$. (The party with the shortest chain should be honest.)*



Observation. *Suppose difficulty d of a chain belongs to a block that was computed by an honest party in a uniquely successful round. Then any other block that contains difficulty d has been computed by the adversary.*

The common-prefix lemma in the dynamic case

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 difficulty d of a chain belongs to a block that was computed by an honest party in a uniquely successful round. Then any other block that contains difficulty d has been computed by the adversary.*

Difficulty accumulated in uniqely successful rounds in a set S $\leq$ Difficulty accumulated by the adversary during rounds in S □

Concentration bounds in the dynamic case

In the dynamic case we still have Bernoulli trials. However, the success probabilities are random variables depending on the strategy of the adversary.

Concentration bounds in the dynamic case

In the dynamic case we still have Bernoulli trials. However, the success probabilities are random variables depending on the strategy of the adversary.

- Consider the following two-player game.
  - In the beginning of round i, player A chooses a bias $p_i \in \mathbb{R}$.
  - Player B flips a coin with bias p_i.
  - If heads, B earns $X_i = \frac{1}{p_i}$ bitcoins; otherwise $X_i = 0$.

Concentration bounds in the dynamic case

In the dynamic case we still have Bernoulli trials. However, the success probabilities are random variables depending on the strategy of the adversary.

- Consider the following two-player game.
  - In the beginning of round i, player A chooses a bias $p_i \in \mathbb{R}$.
  - Player B flips a coin with bias p_i.
  - If heads, B earns $X_i = \frac{1}{p_i}$ bitcoins; otherwise $X_i = 0$.
- B is expected to earn $\mathbf{E}[X_i] = 1$฿ in round i. Thus, B is expected to earn k฿ in k rounds.
- How concentrated around their expectation are B's earnings? Does it hold

$$\Pr\left[\sum_{i=1}^{k} X_i < (1-\epsilon)k\right] = e^{-\Omega(\epsilon^2 k)} \quad ?$$

Martingale bounds

Theorem. *Let f be a function of the n random variables $X_1, \ldots, X_n$. Let*

$$D_i = \mathbf{E}[f|X_1, \ldots, X_i] - \mathbf{E}[f|X_1, \ldots, X_{i-1}],$$

$$V = \sum_{1\le i\le n} \mathrm{Var}(D_i|X_1,..,X_{i-1}) \quad \textit{and} \quad b = \max_{1\le i\le n} \sup(D_i|X_1,..,Y_{i-1})$$

(sup is taken over all possible assignments to $X_1, \ldots, X_{i-1}$). Then, for any $t, v \ge 0$,

$$\Pr\Big[\, f \ge \mathbf{E}f + t \wedge V \le v \,\Big] \le \exp\Big\{-\frac{t^2}{2v + 2bt/3}\Big\}.$$

Martingale bounds

Theorem. *Let f be a function of the n random variables $X_1, \ldots, X_n$. Let*

$$D_i = \mathbf{E}[f|X_1, \ldots, X_i] - \mathbf{E}[f|X_1, \ldots, X_{i-1}],$$

$$V = \sum_{1 \le i \le n} \mathrm{Var}(D_i|X_1,..,X_{i-1}) \quad \textit{and} \quad b = \max_{1 \le i \le n} \sup(D_i|X_1,..,Y_{i-1})$$

(sup is taken over all possible assignments to $X_1, \ldots, X_{i-1}$). Then, for any $t, v \ge 0$,

$$\Pr\Big[\, f \ge \mathbf{E}f + t \wedge V \le v \,\Big] \le \exp\left\{-\frac{t^2}{2v + 2bt/3}\right\}.$$

Proof application: Show that if an execution begins with good initial parameters (in particular, $V \le v$) and at some point deviates from the desired block-production rate, then concentration was violated while $V \le v$.

Proof roadmap (high level)

Assuming the execution begins with good initial parameters—i.e., in the beginning the block-production rate is very close to the (desired) f—we show that with high probability the following hold.

Proof roadmap (high level)

Assuming the execution begins with good initial parameters—i.e., in the beginning the block-production rate is very close to the (desired) f—we show that with high probability the following hold.

If a chain C is adopted by an honest party, then C:

- was never abandoned by honest parties for $\epsilon m/f$ rounds,

Proof roadmap (high level)

Assuming the execution begins with good initial parameters—i.e., in the beginning the block-production rate is very close to the (desired) f—we show that with high probability the following hold.

If a chain C is adopted by an honest party, then C:

- was never abandoned by honest parties for $\epsilon m/f$ rounds,
- is $\epsilon m/f$-accurate—each of its blocks has a timestamp that is $\epsilon m/f$) rounds away from its real creation time,

Proof roadmap (high level)

Assuming the execution begins with good initial parameters—i.e., in the beginning the block-production rate is very close to the (desired) f—we show that with high probability the following hold.

If a chain C is adopted by an honest party, then C:

- was never abandoned by honest parties for $\epsilon m/f$ rounds,
- is $\epsilon m/f$-accurate—each of its blocks has a timestamp that is $\epsilon m/f$) rounds away from its real creation time,
- has "very good" recalculation points: $\frac{\gamma f}{C} \leq \mathbf{E}[X_i] \leq \frac{Cf}{\gamma}$;

Proof roadmap (high level)

Assuming the execution begins with good initial parameters—i.e., in the beginning the block-production rate is very close to the (desired) f—we show that with high probability the following hold.

If a chain C is adopted by an honest party, then C:

- was never abandoned by honest parties for $\epsilon m/f$ rounds,
- is $\epsilon m/f$-accurate—each of its blocks has a timestamp that is $\epsilon m/f$) rounds away from its real creation time,
- has "very good" recalculation points: $\frac{\gamma f}{C} \leq \mathbf{E}[X_i] \leq \frac{Cf}{\gamma}$;
- has blocks with "good" targets: $\frac{f}{C} \leq \mathbf{E}[X_i] \leq Cf$.

Proof roadmap (high level)

Assuming the execution begins with good initial parameters—i.e., in the beginning the block-production rate is very close to the (desired) f—we show that with high probability the following hold.

If a chain C is adopted by an honest party, then C:

- was never abandoned by honest parties for $\epsilon m/f$ rounds,
- is $\epsilon m/f$-accurate—each of its blocks has a timestamp that is $\epsilon m/f$) rounds away from its real creation time,
- has "very good" recalculation points: $\frac{\gamma f}{C} \leq \mathbf{E}[X_i] \leq \frac{Cf}{\gamma}$;
- has blocks with "good" targets: $\frac{f}{C} \leq \mathbf{E}[X_i] \leq Cf$.

Theorem. *Every block in a chain that is ever adopted by an honest party, has "accurate" timestamp and "good" target.*

What is missing

- In the analysis we assume the clocks of the miners are synchronized, which is not realistic.
  - We may assume that any two clocks are within Φ rounds.

What is missing

- In the analysis we assume the clocks of the miners are synchronized, which is not realistic.
  - We may assume that any two clocks are within Φ rounds.
- We now should accept blocks with a timestamp in the future!

  But not too far into the future, because target recalculation may lead to targets artificially large.
  - Bitcoin considers a block to be valid if its timestamp is at most $\Delta' = 2$ hours ahead. (We should set $\Delta' \geq \Phi$.)

What is missing

- In the analysis we assume the clocks of the miners are synchronized, which is not realistic.
  - We may assume that any two clocks are within Φ rounds.
- We now should accept blocks with a timestamp in the future!

  But not too far into the future, because target recalculation may lead to targets artificially large.
  - Bitcoin considers a block to be valid if its timestamp is at most $\Delta' = 2$ hours ahead. (We should set $\Delta' \geq \Phi$.)
- Similarly, we shouldn't accept blocks with timestamps too far in the past, because target recalculation may lead to a small target.
  - Bitcoin considers a block to be valid if its timestamp is at least the median of the last 11 timestamps.

Analysis without synchronized clocks

- Any two clocks are within Φ rounds.
- Bitcoin considers a block to be valid if its timestamp is at most $\Delta' = 2$ hours ahead. (We should set $\Delta' \geq \Phi$.)
- Bitcoin considers a block to be valid if its timestamp is at least the median of the last $k_{\text{med}} = 11$ timestamps.

Analysis without synchronized clocks

- Any two clocks are within Φ rounds.
- Bitcoin considers a block to be valid if its timestamp is at most $\Delta' = 2$ hours ahead. (We should set $\Delta' \geq \Phi$.)
- Bitcoin considers a block to be valid if its timestamp is at least the median of the last $k_{\text{med}} = 11$ timestamps.

How honest are the medians?

- If k_{med} is large, then majority is not enough for honest median (recall selfish mining).
- We need k_{med} to be small enough to argue that honest medians appear sufficiently often.

Back in the simplest model

- Consider a streak of 6 *strongly* isolated honest successes (without adversarial successes among them).

  Under what assumptions will it form an honest median?

Back in the simplest model

- Consider a streak of 6 *strongly* isolated honest successes (without adversarial successes among them).

  Under what assumptions will it form an honest median?

- We can argue that if no block in the streak belongs in a set of rounds in which the adversary obtained at least as many blocks as the honest parties, then the streak will become an honest median.

Back in the simplest model

- Consider a streak of 6 *strongly* isolated honest successes (without adversarial successes among them).

  Under what assumptions will it form an honest median?

- We can argue that if no block in the streak belongs in a set of rounds in which the adversary obtained at least as many blocks as the honest parties, then the streak will become an honest median.

- Consider the oracle outputs as votes for the two candidates: honest and adversary.

Back in the simplest model

- Consider a streak of 6 *strongly* isolated honest successes (without adversarial successes among them).

  Under what assumptions will it form an honest median?

- We can argue that if no block in the streak belongs in a set of rounds in which the adversary obtained at least as many blocks as the honest parties, then the streak will become an honest median.

- Consider the oracle outputs as votes for the two candidates: honest and adversary.

- The assumption can be reworded as follows.

  We want a permutation of the votes so that during the counting of the votes the honest candidate is always ahead of the adversary.

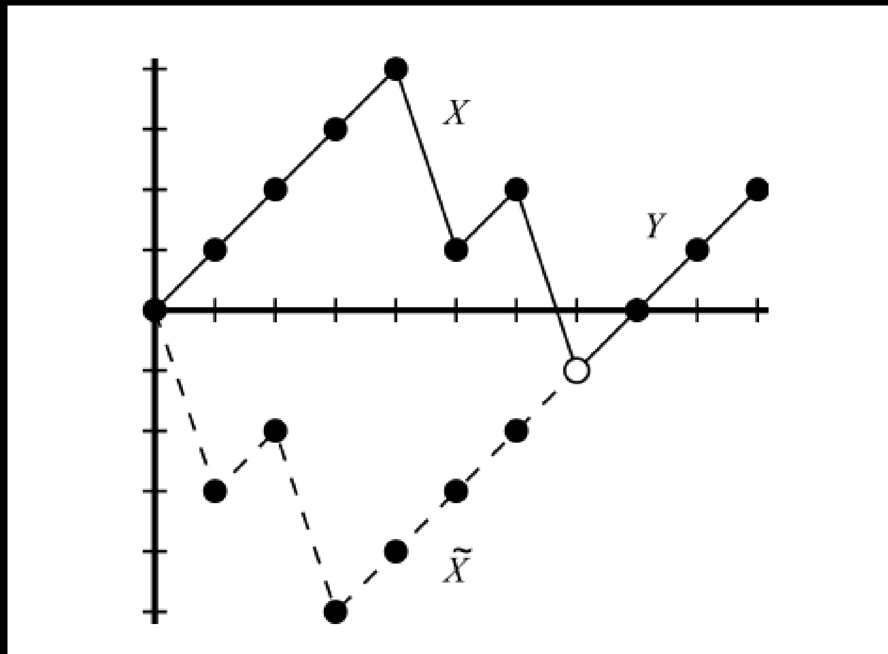
Ballot theorems

Ballot theorem. Suppose candidates A and B received a and b votes respectively. The probability candidate A was always ahead during the counting of the votes is

$$\frac{a-b}{a+b}.$$

Proof by reflection.

Four Proofs of the Ballot Theorem, Marc Renault, Mathematics Magazine, Vol 80, No 5 (Dec 2007).



Ballot theorems

Theorem. Let $X_1, X_2, \ldots$ be an infinite sequence of iid integer random variables with mean $\mu > 0$ and maximum value 1 and for any $i \geq 1$ let $S_i = X_1 + \cdots + X_i$. Then

$$Pr[S_i > 0 \text{ for } n = 1, 2, \ldots] = \mu.$$

Addario-Berry and Reed. Ballot Theorems, Old and New. 2008.

A general ballot theorem

Warnke 2016. Let $X = (X_1, \ldots, X_N)$ be a family of independent random variables with X_j taking values in a set Λ_j and let $\Gamma = \prod_{j\in[N]} \Gamma_j$ where $\Gamma_j \subseteq \Lambda_j$. Assume there are numbers $(c_j)_{j\in[N]}$ so that $f : \prod_{j\in[N]} \Lambda_j \to \mathbb{R}$ satisfies the following. Whenever $x, x' \in \prod_{j\in[N]} \Gamma_j$ differ only in the j-th coordinate and $x, x' \in \Gamma$ we have $|f(x) - f(x')| \le c_j$ and $|f(x) - f(x')| \le d$ for all $x, x' \in \prod_{j\in[N]} \Lambda_j$ that differ in at least one coordinate. Then, for all $t \ge 0$,

$$\Pr\Big[f(x) \le \mathbf{E}[f(X)] - t - d\Pr[X \notin \Gamma]\Big] \le \exp\left\{-\frac{2t^2}{\sum_{j\in[N]} c_j^2}\right\} + \Pr[X \notin \Gamma].$$

References

Joint works with Aggelos Kiayias and Juan Garay.

- The Bitcoin Backbone Protocol: Analysis and Applications.
  https://eprint.iacr.org/2014/765
- The Bitcoin Backbone Protocol with Chains of Variable Difficulty.
  https://eprint.iacr.org/2016/1048
- Full Analysis of Nakamoto Consensus in Bounded-Delay Networks.
  https://eprint.iacr.org/2020/277

Thank you for listening