Introduction to Blockchain - Day III + Day IV

8.01.2025/ 13.01.2025

Problem

In ScroogeCoin, suppose Mallory has banking id of Alice and she can login using that banking id provided she gets hold of the password of Alice. Consider that password of Alice is an 8 digit number, each digit can be between 0 to 9. She tries generating the password until it matches the password of Alice. How long will it take before she succeeds, on average? What will happen if she can crack the password?

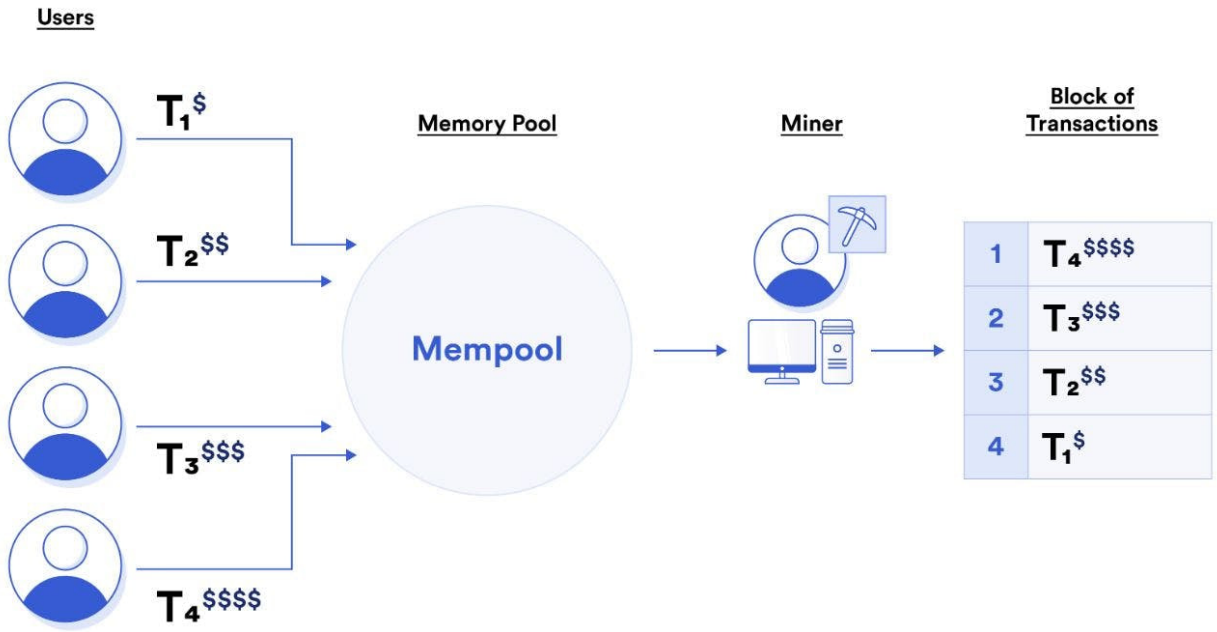
Challenges of DCP in Bitcoin

There are number of technical problems with the approach

- Consensus in general is a hard problem since nodes might crash or be outright malicious
- In the Bitcoin context, P2P network is highly imperfect.
- Not all pairs of nodes are connected to each other.
- There could be faults in the network because of poor Internet connectivity
- A lot of latency in the system because it's distributed all over the Internet.

How do nodes in P2P Network receive transaction?

Either they put their own transaction or receive from some other source.



Ordering I

- Blockchain transactions (or blocks containing them) must be ordered, unambiguously, and without the need for a trusted third party.
- But if transactions are generated by anonymous participants all over the world, and no central party is responsible for organizing the list, how can it be done?

  For example transactions (or blocks) could include timestamps, but how could these timestamps be trusted?

Ordering II

- Relying on a timestamp to determine event order is not possible in a decentralized, geographically dispersed system.
- What we need is a mechanism by which we can verify that one event took place before another or perhaps concurrently.

Transactions, UTXO, mempool I

A transaction, in simple terms, is the transfer of value from one person to another.

For example, when I have a physical bill of money and hand it to another person, I am carrying out a financial transaction. Similarly, in the banking system, when I transfer funds from my bank account to another, I am executing a transaction.

Transactions, UTXO, mempool II

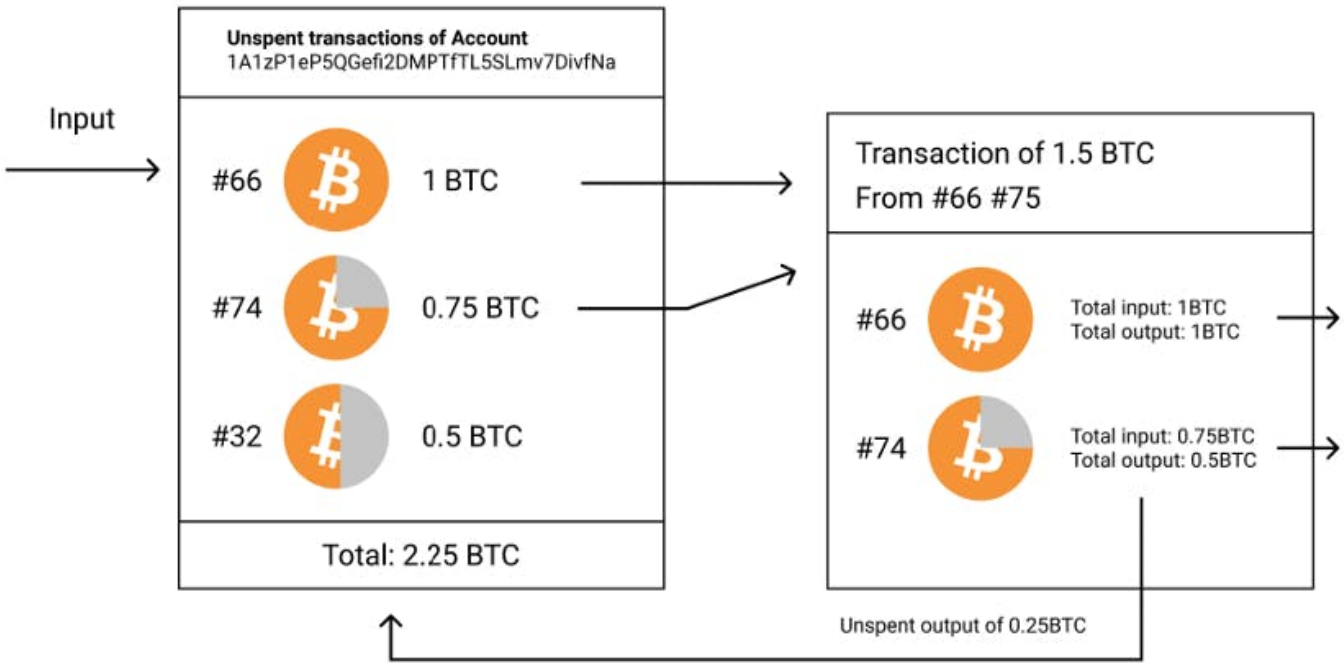
- When someone sends Bitcoin from one address to another, they are not simply transferring a "balance" but rather spending one or more outputs from previous transactions.
- If Kaká sends 1 BTC to Carol, she may be spending an outflow from a previous transaction she received from Fernanda. This transaction output, previously received from Fernanda, will now be used as input in the new transaction sent by Kaká to Carol.

Transactions, UTXO, mempool III

Transaction creation: The sender specifies the recipient's address and the amount of Bitcoin they want to send. The transaction will include one or more "inputs" and one or more "outputs"

The inputs are references to previous UTXOs (Unspent Transaction Outputs), that is, outputs from previous transactions that have not yet been spent. So, basically, these are the "currencies" that the sender is using to fund the transaction.

Transactions, UTXO, mempool IV



Transactions, UTXO, mempool V

UTXO Set: All transaction outputs that have not yet been spent—constitute what is known as the UTXO set. A UTXO can only be utilized once. This is vital for the proper functioning of the blockchain system, as it ensures that the same digital currency is not spent more than once, thereby avoiding the issue of double spending.

Nodes, or validating participants in the Bitcoin network, maintain and constantly update this set. They do this to keep a clear record of which Bitcoin funds are available to be spent and, crucially, who has the authority to spend them.

Transactions, UTXO, mempool VI

To check if a transaction can enter the mempool even if the UTXO set doesn't contain an entry for its input, you need to verify the transaction's inputs against the blockchain explorer to ensure they are unspent outputs (UTXOs) that are currently in the unconfirmed transaction pool (mempool); if they are, then the transaction can be broadcast and should enter the mempool despite not appearing in your local UTXO set, as long as the transaction is valid and has a sufficient fee.

Transactions, UTXO, mempool VII

Prevent Double Spending: With the UTXO set, Bitcoin has found an effective way to combat double spending. This happens because whenever a transaction is made, nodes check the pool to ensure that the UTXOs being used as inputs for the transaction have not yet been spent. Thus, if a UTXO has already been used, the transaction will be considered invalid and rejected by the network.

Transactions, UTXO, mempool VIII

Mempool: The mempool is a collection of unconfirmed transactions waiting to be included in a block. Transactions are broadcast to the network and stored in each node's mempool. A block proposer select transactions from the mempool to include in their candidate blocks based on fee and other criteria

Transactions, UTXO, mempool IX

To check if a transaction is eligible to enter the mempool using the UTXO set, you need to verify that all the input UTXOs referenced in the transaction are currently unspent and that the transaction adheres to the network rules regarding fees and transaction size, essentially confirming that the transaction is valid and can be added to the pending pool for potential inclusion in a block.

Transactions, UTXO, mempool X

In a mempool , if there are two transactions having input that refer to same output, then reject the transaction that arrived later. Technically, every node checks this condition before adding the transaction in mempool. This prevents Double Spending.

Transactions, UTXO, mempool XI

Do you update UTXO set when transaction enter mempool?
No, when a transaction enters the mempool, you do not update the UTXO set; the UTXO set is only updated once the transaction is confirmed and included in a block on the blockchain, at which point the spent UTXOs are removed and new ones created based on the transaction outputs.

Transactions, UTXO, mempool XII

A mempool is updated whenever a new transaction is added to it by a node/participant on the network, and then subsequently updated when a transaction is removed from the mempool by being included in a block mined by a miner; essentially, the mempool is constantly updating as new transactions arrive and existing ones are confirmed and removed to be added to the blockchain

DCP in Bitcoin

Assumptions in the traditional model for consensus that Bitcoin violates

DCP in Bitcoin

Assumptions in the traditional model for consensus that Bitcoin violates

- Introduces the idea of incentive

DCP in Bitcoin

Assumptions in the traditional model for consensus that Bitcoin violates

- Introduces the idea of incentive
- Bitcoin embraces the notion of randomness

DCP in Bitcoin

Assumptions in the traditional model for consensus that Bitcoin violates

- Introduces the idea of incentive
- Bitcoin embraces the notion of randomness
  - It's consensus algorithm does away with specific starting point and ending point for consensus
  - Consensus happens for a longer period of time
  - At the end of the period, node can't be certain that any particular transaction has made into the ledger
  - As time goes on, the probability that your view of any block matches with the consensus view increases.

Consensus without Identity

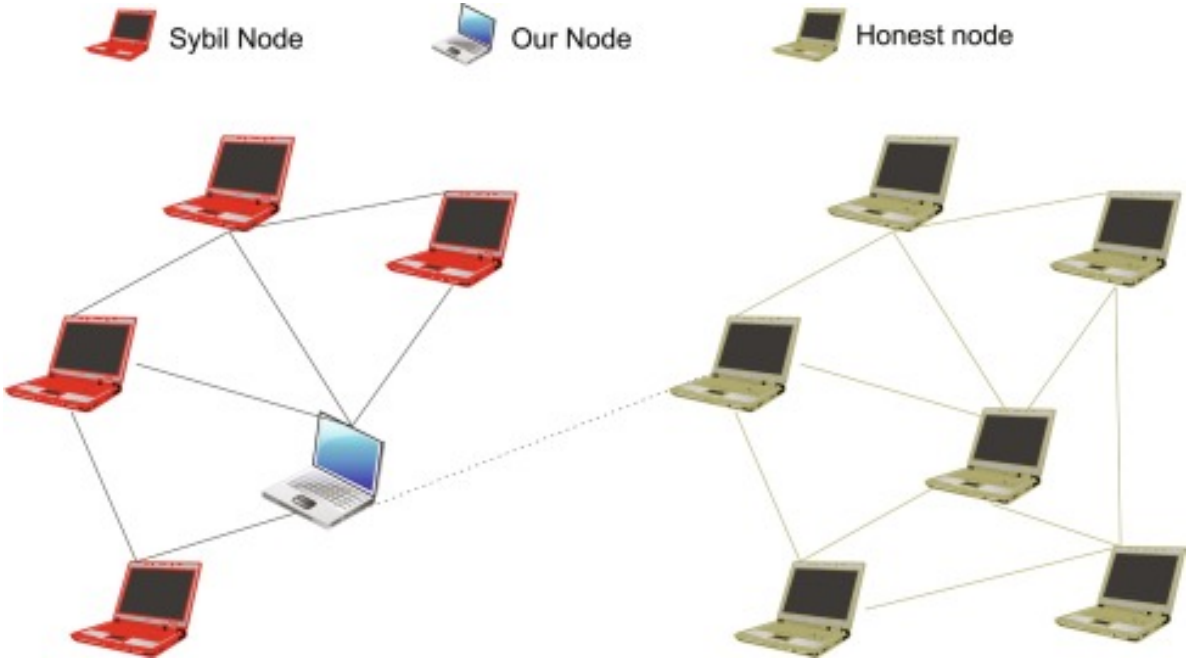
- Bitcoin nodes do not have persistent, long-term identities due to P2P network – difference from traditional DCP.
- No central authority to assign identities to participants and verify that they're not creating new nodes at will.
- Pseudonymity is inherently a goal of Bitcoin.
- Lack of identity introduces *Sybil attack*

What is Sybil?

Sybil is a 1973 book by Flora Rheta Schreiber about the treatment of Sybil Dorsett (a pseudonym for Shirley Ardell Mason) for dissociative identity disorder (then referred to as multiple personality disorder) by her psychoanalyst, Cornelia B. Wilbur.

Mason is given the pseudonym "Sybil" by her therapist to protect her privacy. Sybil manifests sixteen personalities.

Sybil Attack



Consensus Protocol with Identity

- Each node has a separate identity
- Identities would allow us to put in the protocol instructions of the form,

  "Now the node with the lowest numerical ID should take some step"

- Without identities, the set of possible instructions is more constrained.

A much more serious reason for nodes to have identities is for security

For both of these reasons, the lack of identities introduce difficulties for the consensus protocol in Bitcoin.

How Consensus can be reached without Identity? Naive Way

- *Like lottery!* We consider a system that assigns random token id to individual node.
- *Assumption:* We further assume, for the moment, that this token generation and distribution algorithm is sufficiently smart so that Sybil attack does not take place.
- That is if the adversary is going to try to create a lot of Sybil nodes, all of those Sybils together will get only one token.

This results to an implicit consensus

How Consensus can be reached without Identity

Bitcoin Consensus Algorithm with Random Token Id

Bitcoin consensus algorithm (simplified)

This algorithm is simplified in that it assumes the ability to select a random node in a manner that is not vulnerable to Sybil attacks.

1. New transactions are broadcast to all nodes
2. Each node collects new transactions into a block
3. In each round a random node gets to broadcast its block
4. Other nodes accept the block only if all transactions in it are valid (unspent, valid signatures)
5. Nodes express their acceptance of the block by including its hash in the next block they create

How Consensus can be reached without Identity

Bitcoin Consensus Algorithm with Random Token Id

Bitcoin consensus algorithm (simplified)

This algorithm is simplified in that it assumes the ability to select a random node in a manner that is not vulnerable to Sybil attacks.

1. New transactions are broadcast to all nodes
2. Each node collects new transactions into a block
3. In each round a random node gets to broadcast its block
4. Other nodes accept the block only if all transactions in it are valid (unspent, valid signatures)
5. Nodes express their acceptance of the block by including its hash in the next block they create

How a malicious adversary can subvert the process ?

How Consensus can be reached without Identity

Bitcoin Consensus Algorithm with Random Token Id

Bitcoin consensus algorithm (simplified)

This algorithm is simplified in that it assumes the ability to select a random node in a manner that is not vulnerable to Sybil attacks.

1. New transactions are broadcast to all nodes
2. Each node collects new transactions into a block
3. In each round a random node gets to broadcast its block
4. Other nodes accept the block only if all transactions in it are valid (unspent, valid signatures)
5. Nodes express their acceptance of the block by including its hash in the next block they create

How a malicious adversary can subvert the process ?

- Can he steal the bitcoin ?

How Consensus can be reached without Identity

Bitcoin Consensus Algorithm with Random Token Id

Bitcoin consensus algorithm (simplified)

This algorithm is simplified in that it assumes the ability to select a random node in a manner that is not vulnerable to Sybil attacks.

1. New transactions are broadcast to all nodes
2. Each node collects new transactions into a block
3. In each round a random node gets to broadcast its block
4. Other nodes accept the block only if all transactions in it are valid (unspent, valid signatures)
5. Nodes express their acceptance of the block by including its hash in the next block they create

How a malicious adversary can subvert the process ?

- Can he steal the bitcoin ? ?–No–

How Consensus can be reached without Identity

Bitcoin Consensus Algorithm with Random Token Id

Bitcoin consensus algorithm (simplified)

This algorithm is simplified in that it assumes the ability to select a random node in a manner that is not vulnerable to Sybil attacks.

1. New transactions are broadcast to all nodes
2. Each node collects new transactions into a block
3. In each round a random node gets to broadcast its block
4. Other nodes accept the block only if all transactions in it are valid (unspent, valid signatures)
5. Nodes express their acceptance of the block by including its hash in the next block they create

How a malicious adversary can subvert the process ?

- Can he steal the bitcoin ? ?–No–
- Can he mount DoS attack ?

How Consensus can be reached without Identity

Bitcoin Consensus Algorithm with Random Token Id

Bitcoin consensus algorithm (simplified)

This algorithm is simplified in that it assumes the ability to select a random node in a manner that is not vulnerable to Sybil attacks.

1. New transactions are broadcast to all nodes
2. Each node collects new transactions into a block
3. In each round a random node gets to broadcast its block
4. Other nodes accept the block only if all transactions in it are valid (unspent, valid signatures)
5. Nodes express their acceptance of the block by including its hash in the next block they create

How a malicious adversary can subvert the process ?

- Can he steal the bitcoin ? ?–No–
- Can he mount DoS attack ? –No–

How Consensus can be reached without Identity

Bitcoin Consensus Algorithm with Random Token Id

Bitcoin consensus algorithm (simplified)

This algorithm is simplified in that it assumes the ability to select a random node in a manner that is not vulnerable to Sybil attacks.

1. New transactions are broadcast to all nodes
2. Each node collects new transactions into a block
3. In each round a random node gets to broadcast its block
4. Other nodes accept the block only if all transactions in it are valid (unspent, valid signatures)
5. Nodes express their acceptance of the block by including its hash in the next block they create

How a malicious adversary can subvert the process ?

- Can he steal the bitcoin ? ?–No–
- Can he mount DoS attack ? –No–
- Can he mount Double Spending attack ?

Double Spending Attack on Simplified Bitcoin Consensus Protocol

- Alice is an online customer – perhaps malicious
- Bob runs a server, sells softwares in exhange of bitcoins.
- Alice adds an item to her shopping cart on Bob's website and the server requests payment.
- Alice creates a Bitcoin transaction from her address to Bob's and broadcasts it to the network.
- Some honest node creates the next block, and includes this transaction in that block.
- This transaction contains a hash pointer that refers to some previous tx o/p that Alice received.
- There are two different types of hash pointers here : (i) blocks include a hash pointer to the previous block (ii) transactions include one or more hash pointers to previous transaction outputs that are being redeemed.

Double Spending Attack on Simplified Bitcoin Consensus Protocol

- The latest block was generated by an honest node and includes a Alice $\rightarrow$ Bob TX.
- Upon seeing the TX included in the block chain, Bob concludes that Alice has paid him and allows him to download the software.
- Suppose the next random node that is selected in the next round happens to be controlled by Alice.
- Alice proposes a block that ignores the block that contains the payment to Bob and instead contains a pointer to the previous block.
- In that block that Alice includes a TX that transfers the very coins that she was sending to Bob to herself.

Double Spending Attack on Simplified Bitcoin Consensus Protocol

- The latest block was generated by an honest node and includes a Alice $\rightarrow$ Bob TX.
- Upon seeing the TX included in the block chain, Bob concludes that Alice has paid him and allows him to download the software.
- Suppose the next random node that is selected in the next round happens to be controlled by Alice.
- Alice proposes a block that ignores the block that contains the payment to Bob and instead contains a pointer to the previous block.
- In that block that Alice includes a TX that transfers the very coins that she was sending to Bob to herself.

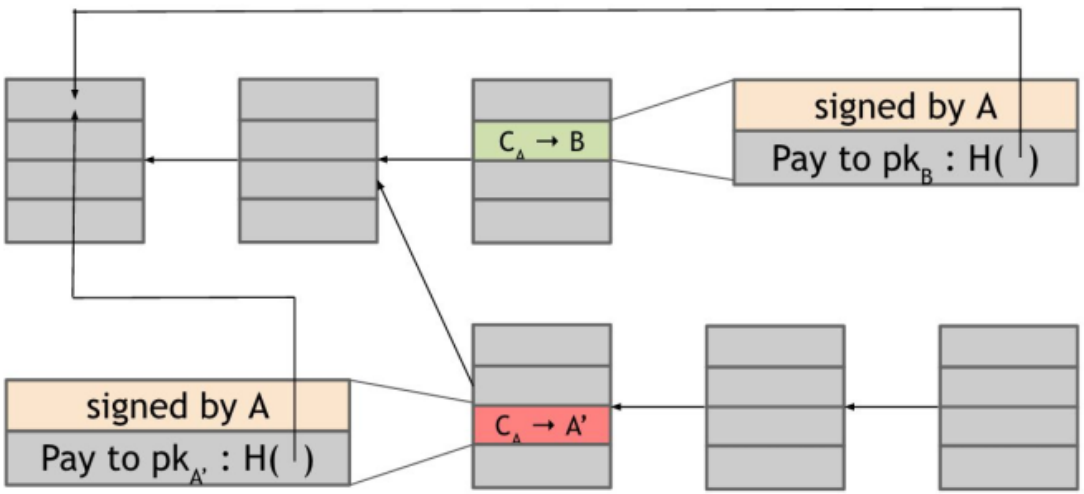
This is a classic double-spend pattern.

Why Double Spending Attack occur?

- Since the two transactions spend the same coins, only one of them can be included in the block chain.
- If Alice succeeds in including the payment to her own address in the block chain, then the transaction in which she pays Bob is useless as it can never be included later in the block chain.

Why Double Spending Attack occur?

- Since the two transactions spend the same coins, only one of them can be included in the block chain.
- If Alice succeeds in including the payment to her own address in the block chain, then the transaction in which she pays Bob is useless as it can never be included later in the block chain.



Double Spending Attack on Simplified Bitcoin Consensus Protocol

How to Determine whether a Double Spend Attack Succeeds or not

- It depends on which block will ultimately end up on the long-term consensus chain
- What determines which block will be included?
  - Honest nodes follow the policy of *extending the longest valid branch*
  - So which branch will they extend?
- If Alice succeeds in including the payment to her own address in the blockchain, then the transaction in which she pays Bob is useless!

Double Spending Attack on Simplified Bitcoin Consensus Protocol

- From the moral point of view, there is a difference between the two blocks
- From technical point of views, both of the blocks are valid.
- In practice, nodes often follow a heuristic of extending the block that they first heard about on the P2P network.

  Caveat: due to network latency, invalid block gets to extended.
- Alice could further try to increase the likelihood of this happening by bribing the next node to do so.

Double Spending Attack on Simplified Bitcoin Consensus Protocol

- If the next node does build on the double-spend block (what is the probability?), then this chain will now be longer than the one that includes the transaction to Bob.

Double Spending Attack on Simplified Bitcoin Consensus Protocol

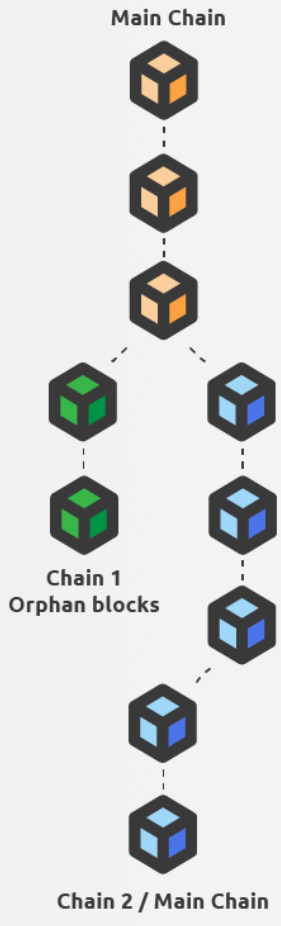
- If the next node does build on the double-spend block (what is the probability?), then this chain will now be longer than the one that includes the transaction to Bob.
- At this point, the next honest node is much more likely to continue to build on this chain since it is longer. This process will continue, and it will become increasingly likely that the block containing the double-spend will be part of the long-term consensus chain.

Double Spending Attack on Simplified Bitcoin Consensus Protocol

- If the next node does build on the double-spend block (what is the probability?), then this chain will now be longer than the one that includes the transaction to Bob.
- At this point, the next honest node is much more likely to continue to build on this chain since it is longer. This process will continue, and it will become increasingly likely that the block containing the double-spend will be part of the long-term consensus chain.

This results to orphan block

Orphan Block

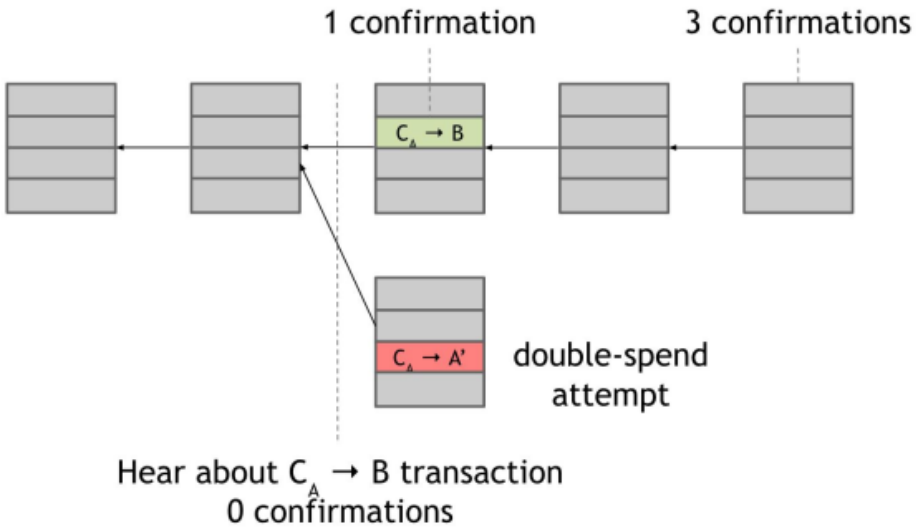


How to Mitigate Double Spending

This introduces the concept of confirmations.

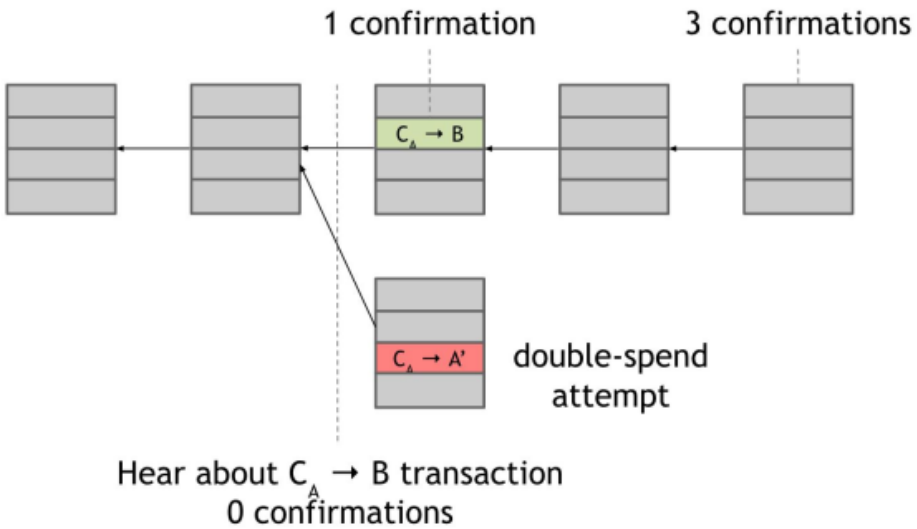
How to Mitigate Double Spending

This introduces the concept of confirmations.



How to Mitigate Double Spending

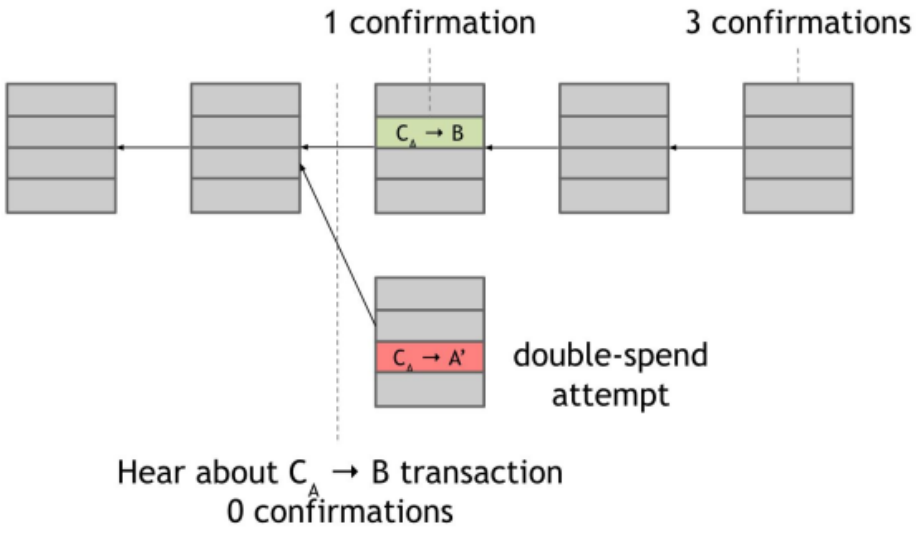
This introduces the concept of confirmations.



- Double-spend probability decreases exponentially with the number of confirmations.

How to Mitigate Double Spending

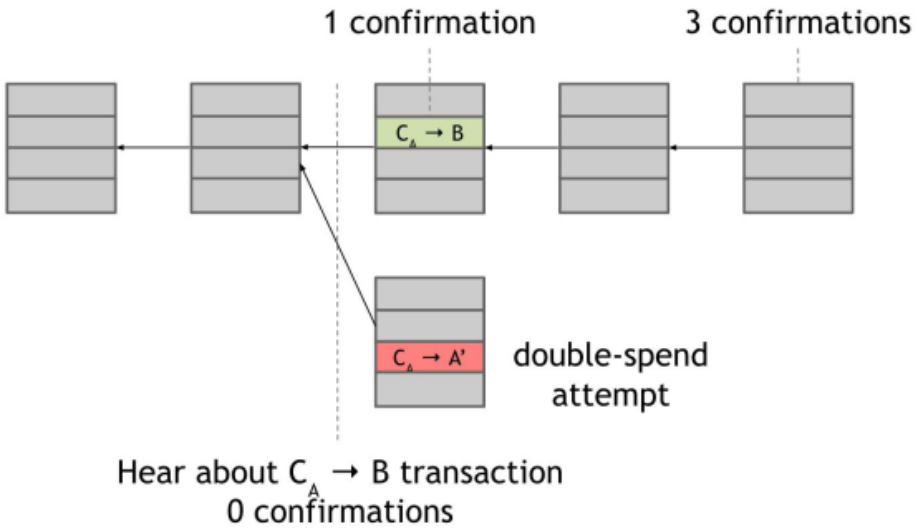
This introduces the concept of confirmations.



- Double-spend probability decreases exponentially with the number of confirmations.
- If the transaction that you are interested in has received k confirmations, then the probability that a double-spend transaction will end up on the long-term consensus chain goes down exponentially as a function of k.

How to Mitigate Double Spending

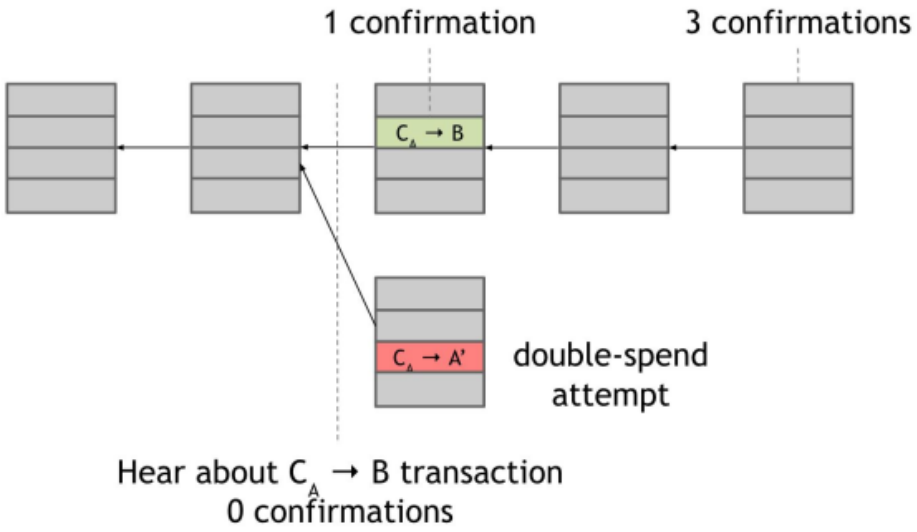
This introduces the concept of confirmations.



- The most common heuristic is to wait for six confirmations.

How to Mitigate Double Spending

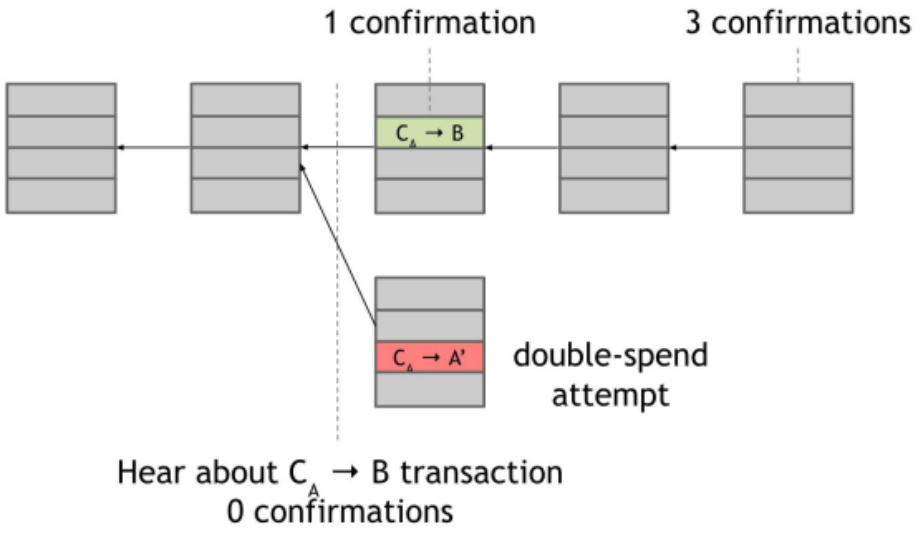
This introduces the concept of confirmations.



- The most common heuristic is to wait for six confirmations.
- It's a good tradeoff between the amount of time you have to wait and your guarantee that the transaction you are interested in ends up on the consensus block chain.

How to Mitigate Double Spending

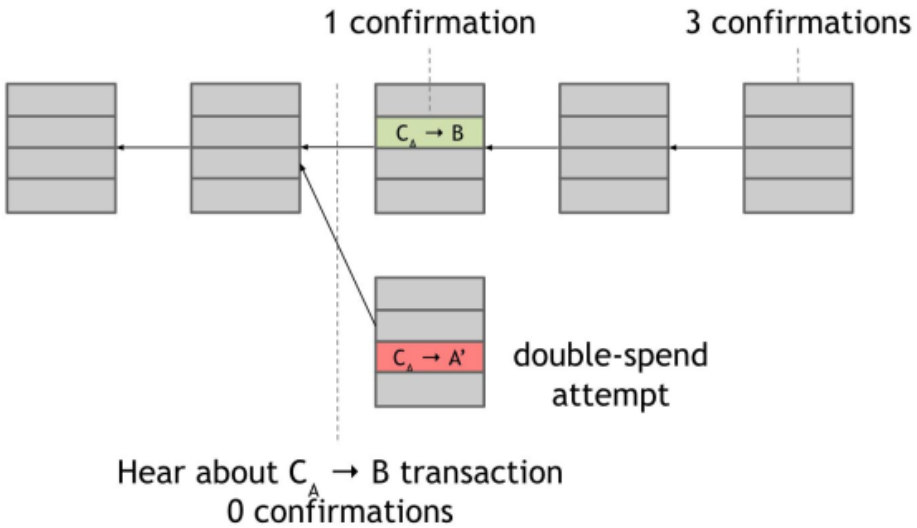
This introduces the concept of confirmations.



- However, you are never 100% sure that a transaction you're interested in is on the consensus branch.

How to Mitigate Double Spending

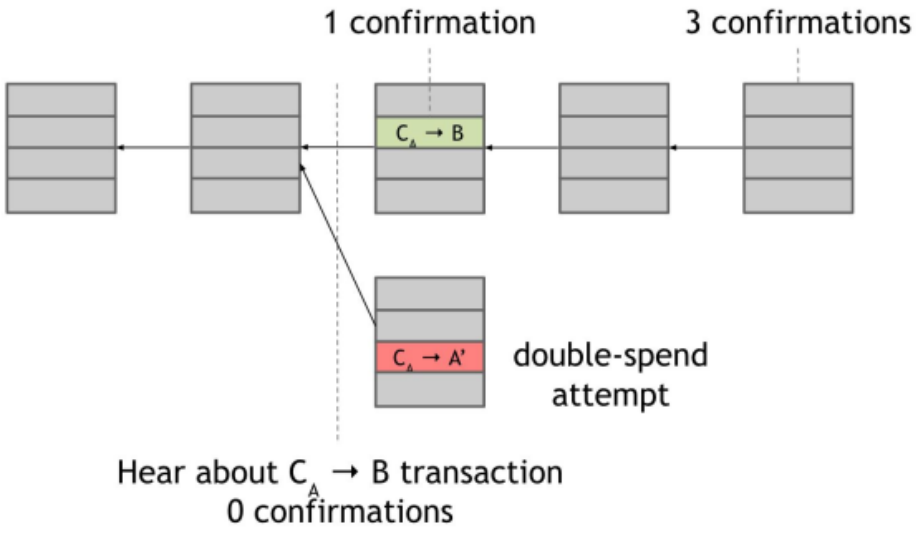
This introduces the concept of confirmations.



- However, you are never 100% sure that a transaction you're interested in is on the consensus branch.
- But, this exponential probability guarantee is rather good.

How to Mitigate Double Spending

This introduces the concept of confirmations.



- However, you are never 100% sure that a transaction you're interested in is on the consensus branch.
- But, this exponential probability guarantee is rather good.
- After about six transactions, there's virtually no chance that you're going to go wrong.

Incentivization of Bitcoin

- We assumed that we're able to pick a random node and, perhaps more problematically, that at least 50 percent of the time, the process will pick an honest node.

Incentivization of Bitcoin

- We assumed that we're able to pick a random node and, perhaps more problematically, that at least 50 percent of the time, the process will pick an honest node.
- This assumption of honesty is problematic to ensure in the face of financial incentives for participants to subvert the process

Incentivization of Bitcoin

- We assumed that we're able to pick a random node and, perhaps more problematically, that at least 50 percent of the time, the process will pick an honest node.
- This assumption of honesty is problematic to ensure in the face of financial incentives for participants to subvert the process

The question then becomes: can we give nodes an incentive for behaving honestly?

Incentivization of Bitcoin

- Can we somehow penalize the node that created the block with the double-spend transaction after the first confirmation ?

Incentivization of Bitcoin

- Can we somehow penalize the node that created the block with the double-spend transaction after the first confirmation ?
- We flip the question around. Can we reward each of the nodes that created the blocks that did end up on the long-term consensus chain?

Incentivization of Bitcoin

- Can we somehow penalize the node that created the block with the double-spend transaction after the first confirmation ?
- We flip the question around. Can we reward each of the nodes that created the blocks that did end up on the long-term consensus chain?

We're going to use bitcoins to incentivize the nodes that created these blocks.

Incentivization of Bitcoin

- Can we somehow penalize the node that created the block with the double-spend transaction after the first confirmation ?
- We flip the question around. Can we reward each of the nodes that created the blocks that did end up on the long-term consensus chain?

We're going to use bitcoins to incentivize the nodes that created these blocks.

- Block reward

Incentivization of Bitcoin

- Can we somehow penalize the node that created the block with the double-spend transaction after the first confirmation ?
- We flip the question around. Can we reward each of the nodes that created the blocks that did end up on the long-term consensus chain?

We're going to use bitcoins to incentivize the nodes that created these blocks.

- Block reward
- Transaction fees

Incentivization of Bitcoin

Block Reward

- Node that creates a block gets to include a special transaction – *coinbase transaction (generates coins!)* and the node can also choose the recipient address of this transaction as its own.
- *Avoid Inflation?* - There will only ever be 21 million Bitcoins minted *But Why?*
- Does block reward enforces nodes to act honestly ?

In the year 2140, there will be no block reward and the generation of bitcoin will be stopped.

Incentivization of Bitcoin

Block Reward

- Node that creates a block gets to include a special transaction – *coinbase transaction (generates coins!)* and the node can also choose the recipient address of this transaction as its own.
- *Avoid Inflation?* - There will only ever be 21 million Bitcoins minted *But Why?*
- Does block reward enforces nodes to act honestly ?

In the year 2140, there will be no block reward and the generation of bitcoin will be stopped.

Does it make the system insecure as there is no incentive for the nodes to act honestly ?

Incentivization of Bitcoin

Transaction Fees

- The creator of any transaction can choose to make

$$\text{total o/p value of the transaction} < \text{total i/p value of transactio} \tag{1}$$

- Whoever creates the block that puts that transaction into the block chain gets to collect the difference, which acts a transaction fee.
- The transaction fee is purely voluntary. But as the block reward starts to run out, it will become almost mandatory

Problem with selection of node

Problems to resolve in bitcoin consensus

Problem with selection of node

Problems to resolve in bitcoin consensus

- How to pick a random node ?

Problem with selection of node

Problems to resolve in bitcoin consensus

- How to pick a random node ?
- Incentivization can make the system unstable

Problem with selection of node

Problems to resolve in bitcoin consensus

- How to pick a random node ?
- Incentivization can make the system unstable
- What happens if an adversary creates a large number of Sybil nodes