# Introduction to Blockchain

## Lecture 2: Bitcoin and Cryptocurrency

Ras Dwivedi

Indian Institute of Technology Kanpur

May 22, 2018

# Outline


1. Recap
2. Hash Functions
3. Digital Signature
4. Cryptocurrency

# Outline

1. Recap
2. Hash Functions
3. Digital Signature
4. Cryptocurrency

# RSA

# RSA

- proposed by Rivest,Shamir,Adleman

# RSA

- proposed by Rivest,Shamir,Adleman
- choose two large distinct prime number $p, q$

# RSA

- proposed by Rivest,Shamir,Adleman
- choose two large distinct prime number $p, q$
- calculate $n = pq$

# RSA

- proposed by Rivest,Shamir,Adleman
- choose two large distinct prime number $p, q$
- calculate $n = pq$
- calculate $\phi = lcm(p-1, q-1)$

# RSA

- proposed by Rivest,Shamir,Adleman
- choose two large distinct prime number $p, q$
- calculate $n = pq$
- calculate $\phi = lcm(p-1, q-1)$
- choose $e$ such that $gcd(e, \phi) = 1$

# RSA

- proposed by Rivest,Shamir,Adleman
- choose two large distinct prime number $p, q$
- calculate $n = pq$
- calculate $\phi = lcm(p-1, q-1)$
- choose $e$ such that $gcd(e, \phi) = 1$
- calculate $d$ such that $d = e^{-1} mod \phi$

# RSA

- proposed by Rivest,Shamir,Adleman
- choose two large distinct prime number $p, q$
- calculate $n = pq$
- calculate $\phi = lcm(p-1, q-1)$
- choose $e$ such that $gcd(e, \phi) = 1$
- calculate $d$ such that $d = e^{-1} mod \phi \longrightarrow e \times d = 1 \ mod \ \phi$

# RSA

- proposed by Rivest,Shamir,Adleman
- choose two large distinct prime number $p, q$
- calculate $n = pq$
- calculate $\phi = lcm(p-1, q-1)$
- choose $e$ such that $gcd(e, \phi) = 1$
- calculate $d$ such that $d = e^{-1} mod \phi \longrightarrow e \times d = 1 \ mod \ \phi$
- Idea: $m^{e \times d} = m^{e^d} = m \ modulo \ n$

# RSA

- proposed by Rivest,Shamir,Adleman
- choose two large distinct prime number $p, q$
- calculate $n = pq$
- calculate $\phi = lcm(p-1, q-1)$
- choose $e$ such that $gcd(e, \phi) = 1$
- calculate $d$ such that $d = e^{-1} mod \phi \longrightarrow e \times d = 1 \ mod \ \phi$
- Idea: $m^{e \times d} = m^{e^d} = m \ modulo \ n$
- **encryption:** $c = m^e \ mod \ n$

# RSA

- proposed by Rivest,Shamir,Adleman
- choose two large distinct prime number $p, q$
- calculate $n = pq$
- calculate $\phi = lcm(p-1, q-1)$
- choose $e$ such that $gcd(e, \phi) = 1$
- calculate $d$ such that $d = e^{-1} mod \phi \longrightarrow e \times d = 1 \ mod \ \phi$
- Idea: $m^{e \times d} = m^{e^d} = m \ modulo \ n$
- **encryption:** $c = m^e \ mod \ n$
- **decryption:** $p = c^d \ mod \ n$

# RSA

- proposed by Rivest,Shamir,Adleman
- choose two large distinct prime number $p, q$
- calculate $n = pq$
- calculate $\phi = lcm(p-1, q-1)$
- choose $e$ such that $gcd(e, \phi) = 1$
- calculate $d$ such that $d = e^{-1} mod \phi \longrightarrow e \times d = 1 \ mod \ \phi$
- Idea: $m^{e \times d} = m^{e^d} = m \ modulo \ n$
- **encryption:** $c = m^e \ mod \ n$
- **decryption:** $p = c^d \ mod \ n$

# Digital signature Attempt 1

- **AIM:** Convince everybody that Alice have signed the document

# Digital signature Attempt 1

- **AIM:** Convince everybody that Alice have signed the document
- nobody should be able to forge the document

# Digital signature Attempt 1

- **AIM:** Convince everybody that Alice have signed the document
- nobody should be able to forge the document
- simple pasting a copy of signature do not work

# Digital signature Attempt 1

- **AIM:** Convince everybody that Alice have signed the document
- nobody should be able to forge the document
- simple pasting a copy of signature do not work
- IDEA: USE RSA

# Digital signature Attempt 1

- **AIM:** Convince everybody that Alice have signed the document
- nobody should be able to forge the document
- simple pasting a copy of signature do not work
- IDEA: USE RSA
- for document $m$ Alice uses $s = m^d$ as her digital signature. To verify, verifier calculates $s^e$ and if $m = s^e \ mod \ n$, signature is genuine

# Digital signature Attempt 1

- **AIM:** Convince everybody that Alice have signed the document
- nobody should be able to forge the document
- simple pasting a copy of signature do not work
- IDEA: USE RSA
- for document $m$ Alice uses $s = m^d$ as her digital signature. To verify, verifier calculates $s^e$ and if $m = s^e \ mod \ n$, signature is genuine
- $d$ is called Alice's secret key and $e$ is called Alice's Public key

# Digital signature Attempt 1

- **AIM:** Convince everybody that Alice have signed the document
- nobody should be able to forge the document
- simple pasting a copy of signature do not work
- IDEA: USE RSA
- for document $m$ Alice uses $s = m^d$ as her digital signature. To verify, verifier calculates $s^e$ and if $m = s^e \ mod \ n$, signature is genuine
- $d$ is called Alice's secret key and $e$ is called Alice's Public key

# Is the Scheme secure?

# Is the Scheme secure?

No!

# Is the Scheme secure?

No!

- given $(n, e)$ private key of Alice cannot be calculated

# Is the Scheme secure?

No!

- given $(n, e)$ private key of Alice cannot be calculated
- given document $m$, $s = m^d$ could not be guessed.

# Is the Scheme secure?

No!

- given $(n, e)$ private key of Alice cannot be calculated
- given document $m$, $s = m^d$ could not be guessed.
- **Problem:** forging given $m_1, m_2$ as two document, and $s_1, s_2$ as their digital signature, one can find the valid signature of $m_1.m_2$ as $s_1.s_2$

# Is the Scheme secure?

No!

- given $(n, e)$ private key of Alice cannot be calculated
- given document $m$, $s = m^d$ could not be guessed.
- **Problem:** forging given $m_1, m_2$ as two document, and $s_1, s_2$ as their digital signature, one can find the valid signature of $m_1.m_2$ as $s_1.s_2$
- Also the length of the signature is proportional to the size of the document

# Is the Scheme secure?

No!

- given $(n, e)$ private key of Alice cannot be calculated
- given document $m$, $s = m^d$ could not be guessed.
- **Problem:** forging given $m_1, m_2$ as two document, and $s_1, s_2$ as their digital signature, one can find the valid signature of $m_1.m_2$ as $s_1.s_2$
- Also the length of the signature is proportional to the size of the document
- slow

# Is the Scheme secure?

No!

- given $(n, e)$ private key of Alice cannot be calculated
- given document $m$, $s = m^d$ could not be guessed.
- **Problem:** forging given $m_1, m_2$ as two document, and $s_1, s_2$ as their digital signature, one can find the valid signature of $m_1.m_2$ as $s_1.s_2$
- Also the length of the signature is proportional to the size of the document
- slow

**What could we do now?**

# SHA: Secure Hash Functions

An Ideal Hash function is one which has following properties

- given $f(x)$ it is impossible to guess $x$

# SHA: Secure Hash Functions

An Ideal Hash function is one which has following properties

- given $f(x)$ it is impossible to guess $x$
- given $x_1$ its is impossible to find $x_2$ such that $f(x_1) = f(x_2)$

# SHA: Secure Hash Functions

An Ideal Hash function is one which has following properties

- given $f(x)$ it is impossible to guess $x$
- given $x_1$ its is impossible to find $x_2$ such that $f(x_1) = f(x_2)$
- it is impossible to find $x_1, x_2$, such that $x_1 \neq x_2$ and $f(x_1) = f(x_2)$

# Merkle Demgrad Construction

Need of Padding message $m$?

# Merkle Demgrad Construction

Need of Padding message $m$?

- $m$ is prefix of $PAD(m)$

# Merkle Demgrad Construction

Need of Padding message $m$?

- $m$ is prefix of $PAD(m)$
- if $|m_1| = |m_2|$ then $|PAD(m_1)| = |PAD(m_2)|$

# Merkle Demgrad Construction

Need of Padding message $m$?

- $m$ is prefix of $PAD(m)$
- if $|m_1| = |m_2|$ then $|PAD(m_1)| = |PAD(m_2)|$
- if $|m_1| \neq |m_2|$ then the last block of $PAD(m_1) \neq PAD(m_2)$

# Merkle Demgrad Construction

Need of Padding message $m$?

- $m$ is prefix of $PAD(m)$
- if $|m_1| = |m_2|$ then $|PAD(m_1)| = |PAD(m_2)|$
- if $|m_1| \neq |m_2|$ then the last block of $PAD(m_1) \neq PAD(m_2)$

# Outline

1 Recap

2 Hash Functions

3 Digital Signature

4 Cryptocurrency

# Hash Function:

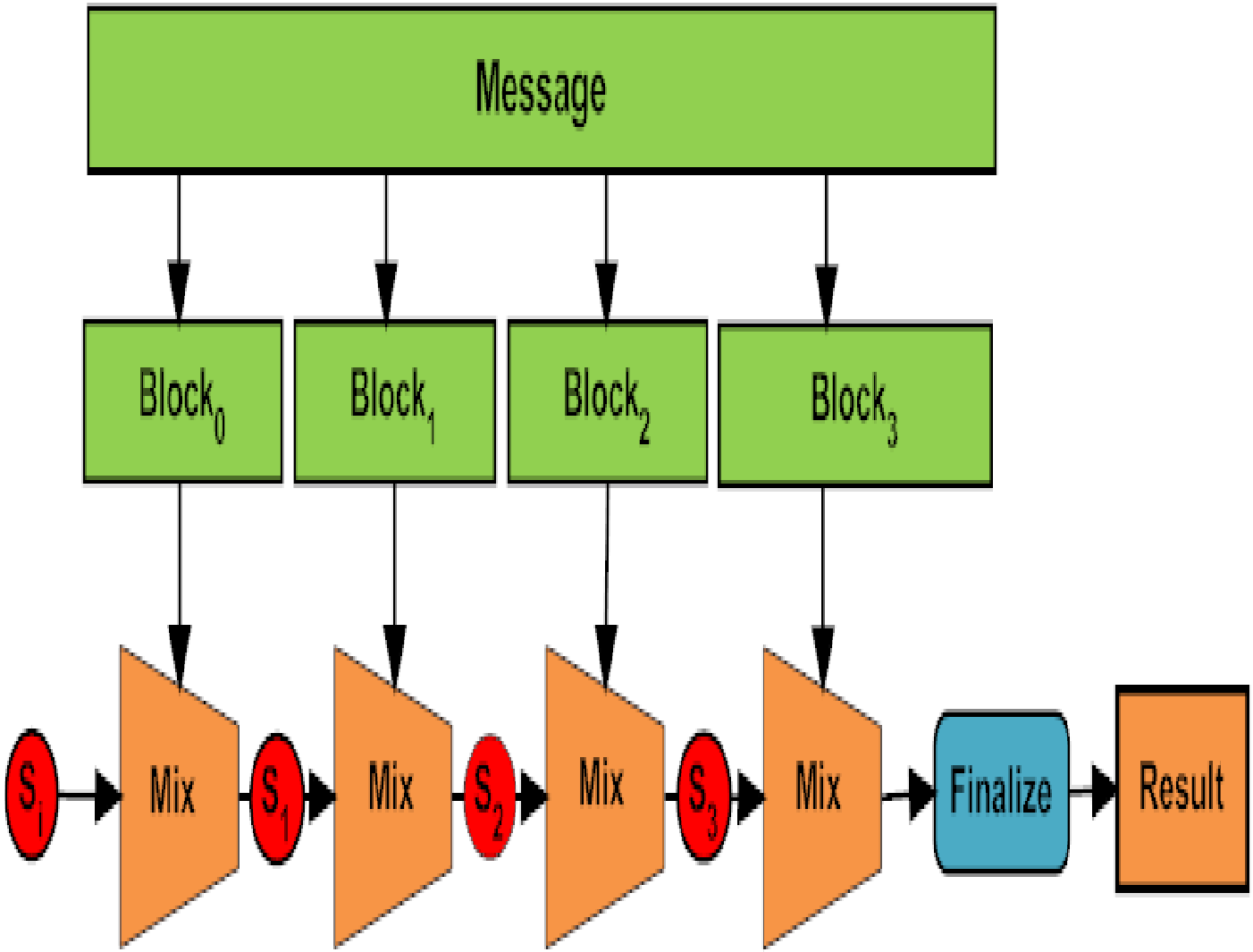



Figure: Hash Function

# Document integrity

# Document integrity

If I have a document "$m$" and I publish its hash "$H(m)$"

# Document integrity

If I have a document "$m$" and I publish its hash "$H(m)$"
Can I change document later?

# Document integrity

If I have a document "$m$" and I publish its hash "$H(m)$"
Can I change document later?
Can I have two document with same Hash?

# Document integrity

If I have a document "$m$" and I publish its hash "$H(m)$"
Can I change document later?
Can I have two document with same Hash?
Can I verify a document with incorrect Hash

# Document integrity

If I have a document "$m$" and I publish its hash "$H(m)$"
Can I change document later?
Can I have two document with same Hash?
Can I verify a document with incorrect Hash

# Outline

1 Recap

2 Hash Functions

3 Digital Signature

4 Cryptocurrency

# Digital signature Attempt 1

- **AIM:** Convince everybody that Alice have signed the document

# Digital signature Attempt 1

- **AIM:** Convince everybody that Alice have signed the document
- nobody should be able to forge the document

# Digital signature Attempt 1

- **AIM:** Convince everybody that Alice have signed the document
- nobody should be able to forge the document
- simple pasting a copy of signature do not work

# Digital signature Attempt 1

- **AIM:** Convince everybody that Alice have signed the document
- nobody should be able to forge the document
- simple pasting a copy of signature do not work
- IDEA: USE RSA

# Digital signature Attempt 1

- **AIM:** Convince everybody that Alice have signed the document
- nobody should be able to forge the document
- simple pasting a copy of signature do not work
- IDEA: USE RSA
- for document $m$ Alice uses $s = m^d$ as her digital signature. To verify, verifier calculates $s^e$ and if$m = s^e \ mod \ n$, signature is genuine

# Digital signature Attempt 1

- **AIM:** Convince everybody that Alice have signed the document
- nobody should be able to forge the document
- simple pasting a copy of signature do not work
- IDEA: USE RSA
- for document $m$ Alice uses $s = m^d$ as her digital signature. To verify, verifier calculates $s^e$ and if$m = s^e \ mod \ n$, signature is genuine
- $d$ problem: given $s(m_1)$ and $s(m_2)$ one could calculate $s(m_1.m_2)$

# Digital Signature: Another try

- **AIM:** Convince everybody that Alice have signed the document
- nobody should be able to forge the document

# Digital Signature: Another try

- **AIM:** Convince everybody that Alice have signed the document
- nobody should be able to forge the document
- IDEA: publish hash of document $m$ as $h = H(m)$

# Digital Signature: Another try

- **AIM:** Convince everybody that Alice have signed the document
- nobody should be able to forge the document
- IDEA: publish hash of document $m$ as $h = H(m)$
- IDEA: USE RSA

# Digital Signature: Another try

- **AIM:** Convince everybody that Alice have signed the document
- nobody should be able to forge the document
- IDEA: publish hash of document $m$ as $h = H(m)$
- IDEA: USE RSA
- for document $m$ Alice uses $s = h^d$ as her digital signature.

# Digital Signature: Another try

- **AIM:** Convince everybody that Alice have signed the document
- nobody should be able to forge the document
- IDEA: publish hash of document $m$ as $h = H(m)$
- IDEA: USE RSA
- for document $m$ Alice uses $s = h^d$ as her digital signature.
- To verify, verifier calculates $s^e$ and if$H(m) = s^e \ mod \ n$, signature is genuine

# Digital Signature: Another try

- **AIM:** Convince everybody that Alice have signed the document
- nobody should be able to forge the document
- IDEA: publish hash of document $m$ as $h = H(m)$
- IDEA: USE RSA
- for document $m$ Alice uses $s = h^d$ as her digital signature.
- To verify, verifier calculates $s^e$ and if$H(m) = s^e \ mod \ n$, signature is genuine
- $d$ given $s(m_1)$ and $s(m_2)$ one can not calculate $s(m_1.m_2)$

# Summary

- Symmetric vs asymmetric encryption system

# Summary

- Symmetric vs asymmetric encryption system
- RSA

# Summary

- Symmetric vs asymmetric encryption system
- RSA
- SHA as tool for data integrity

# Summary

- Symmetric vs asymmetric encryption system
- RSA
- SHA as tool for data integrity
- Digital signature

# Summary

- Symmetric vs asymmetric encryption system
- RSA
- SHA as tool for data integrity
- Digital signature

# Outline

1 Recap

2 Hash Functions

3 Digital Signature

4 Cryptocurrency

# Trade

Figure: Barter Trade

## Problem: Meet of Demand

# how to Meet Demand

- Medium of Exchange

# how to Meet Demand

- Medium of Exchange
- Gold

# Meet of Demand

Figure: Coins

# Meet of Demand

Figure: Coins

Confidence of metal inside

# Meet of Demand

Figure: Coins

Confidence of metal inside

Problem: Difficult to carry

# Solution: Paper note


Figure: paper note

# Solution: Paper note


Figure: paper note

problem: Paper alone do not have value

# What gives currency power?

- Backed by RBI

# What gives currency power?

- Backed by RBI
- RBI controls the flow

# What gives currency power?

- Backed by RBI
- RBI controls the flow
- strong regulation against counterfeiting

# What gives currency power?

- Backed by RBI
- RBI controls the flow
- strong regulation against counterfeiting

# Digital currency

# Digital currency

Figure: PoS machine

# Digital currency

# Digital currency

- Backed by Central Bank to give legitimacy

# Digital currency

- Backed by Central Bank to give legitimacy
- problem: Double spending

# Digital currency

- Backed by Central Bank to give legitimacy
- problem: Double spending
- solution:

# Digital currency

- Backed by Central Bank to give legitimacy
- problem: Double spending
- solution:Central server to record all the transaction

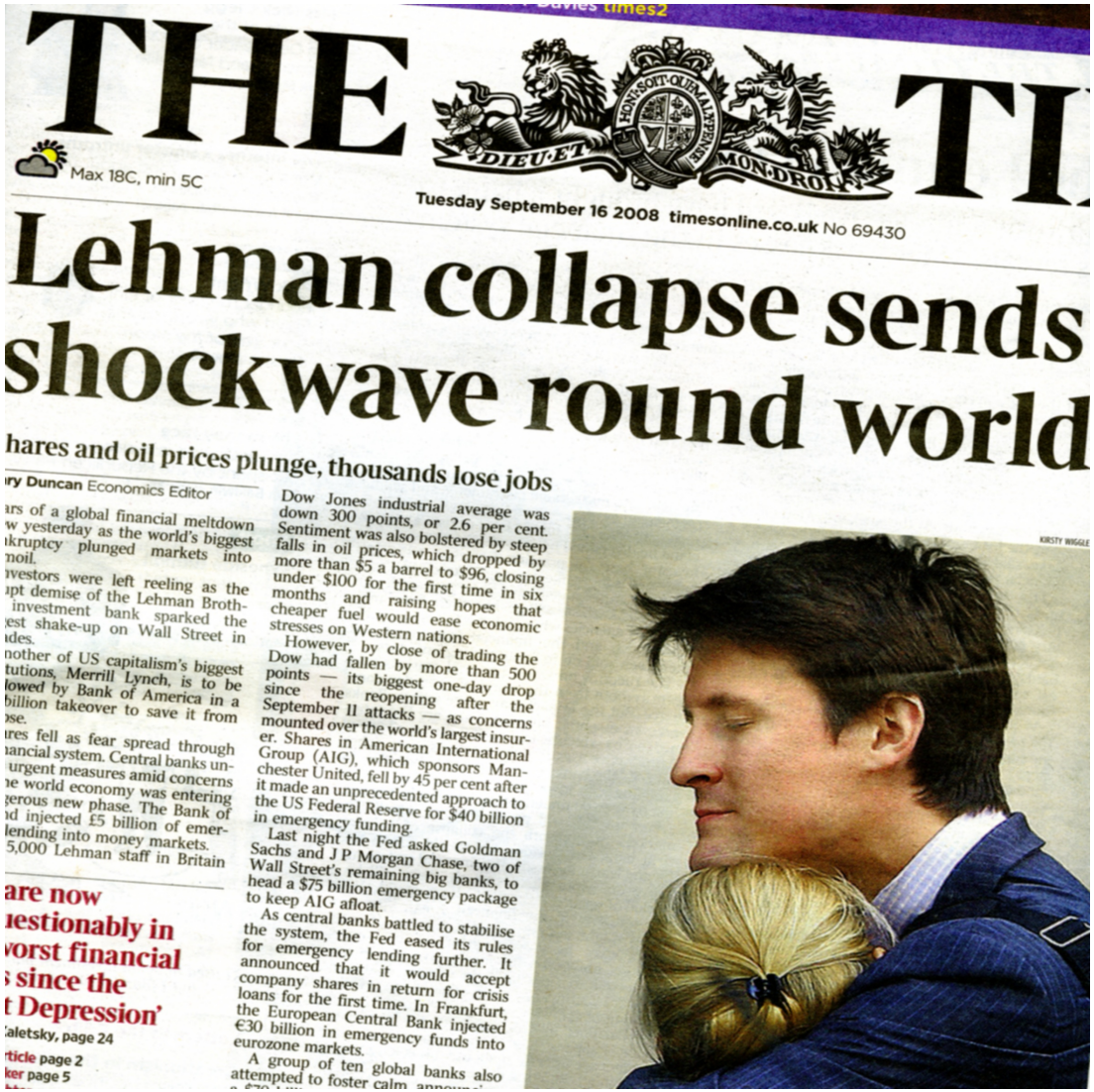
# Digital currency

- Backed by Central Bank to give legitimacy
- problem: Double spending
- solution:Central server to record all the transaction
- Enter Banks: act as centralized server

# Digital currency

- Backed by Central Bank to give legitimacy
- problem: Double spending
- solution:Central server to record all the transaction
- Enter Banks: act as centralized server

# 2008 Recession



THE TI

Max 18C, min 5C

Tuesday September 16 2008 timesonline.co.uk No 69430

## Lehman collapse sends shockwave round world

### hares and oil prices plunge, thousands lose jobs

ry Duncan Economics Editor

ars of a global financial meltdown w yesterday as the world's biggest ıkruptcy plunged markets into noil.

ıvestors were left reeling as the ıpt demise of the Lehman Broth- investment bank sparked the est shake-up on Wall Street in ades.

nother of US capitalism's biggest tutions, Merrill Lynch, is to be lowed by Bank of America in a billion takeover to save it from se.

ares fell as fear spread through ancial system. Central banks un- urgent measures amid concerns e world economy was entering gerous new phase. The Bank of d injected £5 billion of emer- lending into money markets. 5,000 Lehman staff in Britain

**are now uestionably in vorst financial since the t Depression'**

aletsky, page 24

rticle page 2
ker page 5

Dow Jones industrial average was down 300 points, or 2.6 per cent. Sentiment was also bolstered by steep falls in oil prices, which dropped by more than $5 a barrel to $96, closing under $100 for the first time in six months and raising hopes that cheaper fuel would ease economic stresses on Western nations.

However, by close of trading the Dow had fallen by more than 500 points — its biggest one-day drop since the reopening after the September 11 attacks — as concerns mounted over the world's largest insurer. Shares in American International Group (AIG), which sponsors Manchester United, fell by 45 per cent after it made an unprecedented approach to the US Federal Reserve for $40 billion in emergency funding.

Last night the Fed asked Goldman Sachs and J P Morgan Chase, two of Wall Street's remaining big banks, to head a $75 billion emergency package to keep AIG afloat.

As central banks battled to stabilise the system, the Fed eased its rules for emergency lending further. It announced that it would accept company shares in return for crisis loans for the first time. In Frankfurt, the European Central Bank injected €30 billion in emergency funds into eurozone markets.

A group of ten global banks also attempted to foster calm, announc

KIRSTY WIGGLE

Figure: Caption

# Cryptocurrency: The Begining

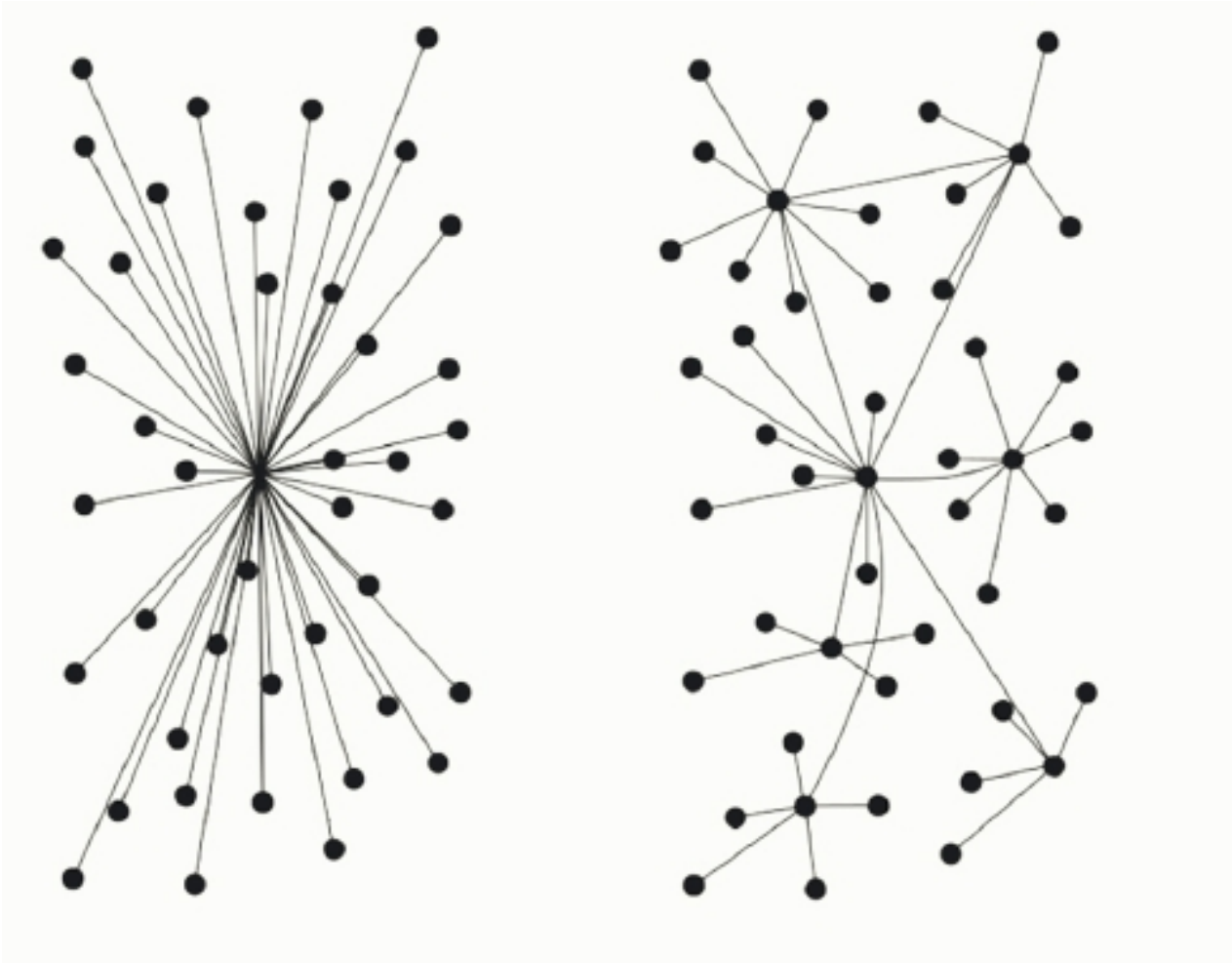
- problem: cannot trust bank

# Cryptocurrency: The Begining

- problem: cannot trust bank and their ledger

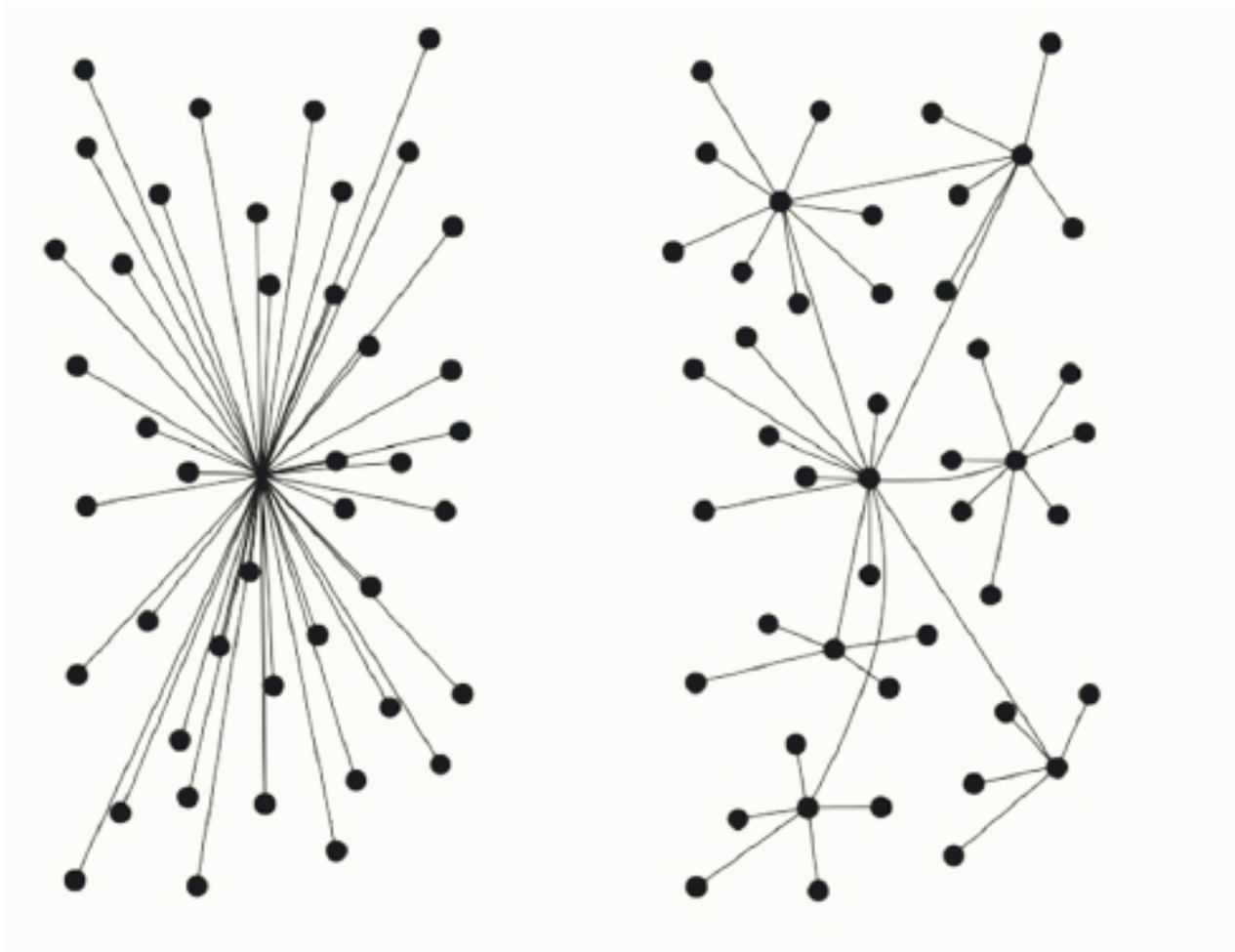
# Cryptocurrency: The Begining

- problem: cannot trust bank and their ledger
- solution: Make decentralized ledger



# Cryptocurrency: The Begining

- problem: cannot trust bank and their ledger
- solution: Make decentralized and permissionless ledger



# Problem solved?

# Problem solved?

No!

# Problem solved?

No!

It just exploded.

# Problem 1: Identity

# Problem 1: Identity

- bank used to do the verification and assign an identity to the account number

# Problem 1: Identity

- bank used to do the verification and assign an identity to the account number
- for India KYC norms

# Problem 1: Identity

- bank used to do the verification and assign an identity to the account number
- for India KYC norms
- Who would do identity management here?

# Problem 1: Identity

- bank used to do the verification and assign an identity to the account number
- for India KYC norms
- Who would do identity management here?
- Think RSA !!

# Problem 1: Identity

- bank used to do the verification and assign an identity to the account number
- for India KYC norms
- Who would do identity management here?
- Think RSA !!

# Identity: RSA a quick look

# Identity: RSA a quick look

- Public key: Known to all

# Identity: RSA a quick look

- Public key: Known to all
- Secret key: known to none

# Identity: RSA a quick look

- Public key: Known to all
- Secret key: known to none
- Anybody can use my public key to encrypt the text and only I can decrypt it

# Identity: RSA a quick look

- Public key: Known to all
- Secret key: known to none
- Anybody can use my public key to encrypt the text and only I can decrypt it
- Idea:

# Identity: RSA a quick look

- Public key: Known to all
- Secret key: known to none
- Anybody can use my public key to encrypt the text and only I can decrypt it
- Idea: What if I make Public key as Identity

# Identity: RSA a quick look

- Public key: Known to all
- Secret key: known to none
- Anybody can use my public key to encrypt the text and only I can decrypt it
- Idea: What if I make Public key as Identity
- nobody would know my real identity,

# Identity: RSA a quick look

- Public key: Known to all
- Secret key: known to none
- Anybody can use my public key to encrypt the text and only I can decrypt it
- Idea: What if I make Public key as Identity
- nobody would know my real identity,but does that really matters?

# Identity: RSA a quick look

- Public key: Known to all
- Secret key: known to none
- Anybody can use my public key to encrypt the text and only I can decrypt it
- Idea: What if I make Public key as Identity
- nobody would know my real identity,but does that really matters?
- Anonymity comes for free

# Problem 2: Authorization

How does Banks do it?

# Problem 2: Authorization

How does Banks do it?
They provide you with some secret key like: Password, PIN, OTP, which you can use to verify your identity

# Problem 2: Authorization

How does Banks do it?
They provide you with some secret key like: Password, PIN, OTP, which you can use to verify your identity
**problem**: No trusted Banks
**Solution:**

# Problem 2: Authorization

How does Banks do it?
They provide you with some secret key like: Password, PIN, OTP, which you can use to verify your identity
**problem**: No trusted Banks
**Solution:** use Digital signature with secret key of RSA

# Problem 2: Authorization

How does Banks do it?
They provide you with some secret key like: Password, PIN, OTP, which you can use to verify your identity
**problem**: No trusted Banks
**Solution:** use Digital signature with secret key of RSA :)

# Problem 3: How to know your balance?

# Problem 3: How to know your balance?

How does Banks do it?

# Problem 3: How to know your balance?

How does Banks do it?
They maintain each and every account and every transaction.

# Problem 3: How to know your balance?

How does Banks do it?
They maintain each and every account and every transaction.
Basically a ledger

# Problem 3: How to know your balance?

How does Banks do it?
They maintain each and every account and every transaction.
Basically a ledger and a ledger for each and every account

# Problem 3: How to know your balance?

How does Banks do it?
They maintain each and every account and every transaction.
Basically a ledger and a ledger for each and every account
what could we do?

# Problem 3: How to know your balance?

How does Banks do it?
They maintain each and every account and every transaction.
Basically a ledger and a ledger for each and every account
what could we do? Maintain a ledger over distributed nodes

# Problem 3: How to know your balance?

How does Banks do it?
They maintain each and every account and every transaction.
Basically a ledger and a ledger for each and every account
what could we do? Maintain a ledger over distributed nodes
Problem solved?

# Problem 3: How to know your balance?

How does Banks do it?
They maintain each and every account and every transaction.
Basically a ledger and a ledger for each and every account
what could we do? Maintain a ledger over distributed nodes
Problem solved? No!

# Problem 3: How to update ledger

Solution 1

- allow any node to write it

# Problem 3: How to update ledger

Solution 1

- allow any node to write it
- Problem:

# Problem 3: How to update ledger

Solution 1

- allow any node to write it
- Problem: how do you know that node is trustworthy?

# Problem 3: How to update ledger

Solution 1

- allow any node to write it
- Problem: how do you know that node is trustworthy?
- Solution:

# Problem 3: How to update ledger

Solution 1

- allow any node to write it
- Problem: how do you know that node is trustworthy?
- Solution: verify it by majority

# Problem 3: How to update ledger

Solution 1

- allow any node to write it
- Problem: how do you know that node is trustworthy?
- Solution: verify it by majority
- cost for not trusting central bank

# Problem 3: How to update ledger

Solution 2

- allow any node to write it

# Problem 3: How to update ledger

Solution 2

- allow any node to write it and verify it by majority

# Problem 3: How to update ledger

Solution 2

- allow any node to write it and verify it by majority
- Problem:

# Problem 3: How to update ledger

Solution 2

- allow any node to write it and verify it by majority
- Problem:Sybil Attack

# Problem 3: How to update ledger

Solution 2

- allow any node to write it and verify it by majority
- Problem:Sybil Attack
- Sybil attack: Where one node have more than one identities

# Problem 3: How to update ledger

Solution 2

- allow any node to write it and verify it by majority
- Problem:Sybil Attack
- Sybil attack: Where one node have more than one identities
- Solution:

# Problem 3: How to update ledger

Solution 2

- allow any node to write it and verify it by majority
- Problem:Sybil Attack
- Sybil attack: Where one node have more than one identities
- Solution: Ask them to show proof of some limited resources

# Problem 3: How to update ledger

Solution 2

- allow any node to write it and verify it by majority
- Problem:Sybil Attack
- Sybil attack: Where one node have more than one identities
- Solution: Ask them to show proof of some limited resources
- What are the resources available?

# Proof of Work

- Limited Resource is your computation power.

# Proof of Work

- Limited Resource is your computation power. you cannot fake it :)

# Proof of Work

- Limited Resource is your computation power. you cannot fake it :)
- How to implement it?

# Proof of Work

- Limited Resource is your computation power. you cannot fake it :)
- How to implement it?
- Ask them to solve some very difficult problems

# Proof of Work

- Limited Resource is your computation power. you cannot fake it :)
- How to implement it?
- Ask them to solve some very difficult problems
- What could be a difficult problem ?

# Proof of Work

- Limited Resource is your computation power. you cannot fake it :)
- How to implement it?
- Ask them to solve some very difficult problems
- What could be a difficult problem ?
- We have seen one such problem before.

# Proof of Work

- Limited Resource is your computation power. you cannot fake it :)
- How to implement it?
- Ask them to solve some very difficult problems
- What could be a difficult problem ?
- We have seen one such problem before.
- **Factoring**

# Proof of Work

- Limited Resource is your computation power. you cannot fake it :)
- How to implement it?
- Ask them to solve some very difficult problems
- What could be a difficult problem ?
- We have seen one such problem before.
- **Factoring**
- problem: how would you find such difficult problems again and again

# Proof of Work

- Limited Resource is your computation power. you cannot fake it :)
- How to implement it?
- Ask them to solve some very difficult problems
- What could be a difficult problem ?

# Proof of Work

- Limited Resource is your computation power. you cannot fake it :)
- How to implement it?
- Ask them to solve some very difficult problems
- What could be a difficult problem ?
- We know hash is uniformly distributed

# Proof of Work

- Limited Resource is your computation power. you cannot fake it :)
- How to implement it?
- Ask them to solve some very difficult problems
- What could be a difficult problem ?
- We know hash is uniformly distributed
- Ask them to find a hash of number, such that it has certain number of leading zeros

# Proof of Work

- Limited Resource is your computation power. you cannot fake it :)
- How to implement it?
- Ask them to solve some very difficult problems
- What could be a difficult problem ?
- We know hash is uniformly distributed
- Ask them to find a hash of number, such that it has certain number of leading zeros
- Advantage: you can control the difficulty level of the problem

# Proof of Work

- Limited Resource is your computation power. you cannot fake it :)
- How to implement it?
- Ask them to solve some very difficult problems
- What could be a difficult problem ?
- We know hash is uniformly distributed
- Ask them to find a hash of number, such that it has certain number of leading zeros
- Advantage: you can control the difficulty level of the problem

# SHA: Secure Hash Functions

An Ideal Hash function $f(x) : \{0,1\}^m \rightarrow \{0,1\}^n$ is one which has following properties

- given $f(x)$ it is impossible to guess $x$

# SHA: Secure Hash Functions

An Ideal Hash function $f(x) : \{0,1\}^m \rightarrow \{0,1\}^n$ is one which has following properties

- given $f(x)$ it is impossible to guess $x$
- given $x_1$ its is impossible to find $x_2$ such that $f(x_1) = f(x_2)$

# SHA: Secure Hash Functions

An Ideal Hash function $f(x) : \{0,1\}^m \rightarrow \{0,1\}^n$ is one which has following properties

- given $f(x)$ it is impossible to guess $x$
- given $x_1$ its is impossible to find $x_2$ such that $f(x_1) = f(x_2)$
- it is impossible to find $x_1, x_2$, such that $x_1 \neq x_2$ and $f(x_1) = f(x_2)$

# SHA: Secure Hash Functions

An Ideal Hash function $f(x) : \{0,1\}^m \rightarrow \{0,1\}^n$ is one which has following properties

- given $f(x)$ it is impossible to guess $x$
- given $x_1$ its is impossible to find $x_2$ such that $f(x_1) = f(x_2)$
- it is impossible to find $x_1, x_2$, such that $x_1 \neq x_2$ and $f(x_1) = f(x_2)$

# Problem 3: update the ledger

- Ask every node to collect the transactions

# Problem 3: update the ledger

- Ask every node to collect the transactions verify them

# Problem 3: update the ledger

- Ask every node to collect the transactions verify them
- consolidate them into a block and then

# Problem 3: update the ledger

- Ask every node to collect the transactions verify them
- consolidate them into a block and then guess a random number

# Problem 3: update the ledger

- Ask every node to collect the transactions verify them
- consolidate them into a block and then guess a random number
- such that hash of the block has certain leading zeros

# Problem 3: update the ledger

- Ask every node to collect the transactions verify them
- consolidate them into a block and then guess a random number
- such that hash of the block has certain leading zeros
- then the successful node propagates the block to every other node, and they too verify it

# Problem 3: update the ledger

- Ask every node to collect the transactions verify them
- consolidate them into a block and then guess a random number
- such that hash of the block has certain leading zeros
- then the successful node propagates the block to every other node, and they too verify it
- Gossip Protocol

# Problem 3: update the ledger

- Ask every node to collect the transactions verify them
- consolidate them into a block and then guess a random number
- such that hash of the block has certain leading zeros
- then the successful node propagates the block to every other node, and they too verify it
- Gossip Protocol
- once verified they add it to their ledger

# Problem 3: update the ledger

- Ask every node to collect the transactions verify them
- consolidate them into a block and then guess a random number
- such that hash of the block has certain leading zeros
- then the successful node propagates the block to every other node, and they too verify it
- Gossip Protocol
- once verified they add it to their ledger

# Merkel tree

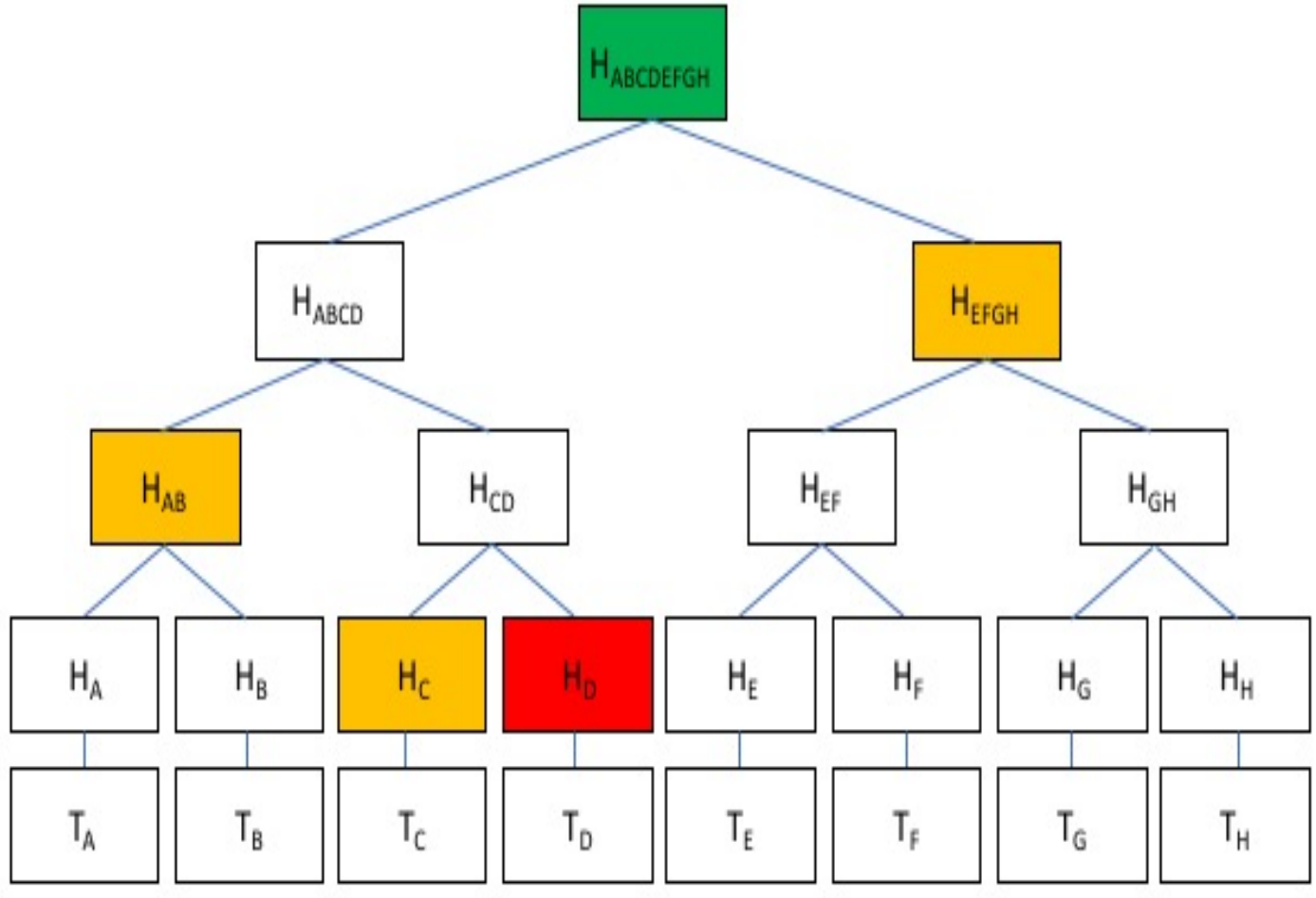



Figure: Merkle Tree

# Problem 3: update the ledger

- What if two node are successful?
- accept the one which has better solution

# Problem 3: update the ledger

- What if two node are successful?
- accept the one which has better solution

Problem solved?

# Problem 3: update the ledger

- What if two node are successful?
- accept the one which has better solution

Problem solved? No!

# Problem 4: how to know your balance

Assume that there is a public ledger

- It is not practical to maintain each and every account

# Problem 4: how to know your balance

Assume that there is a public ledger

- It is not practical to maintain each and every account
- Solution: modify transaction recording to get the balance

# Problem 4: how to know your balance

Assume that there is a public ledger

- It is not practical to maintain each and every account
- Solution: modify transaction recording to get the balance
- Suppose: $A$ have to give $B$ Rs. $x$. Initial balance of $A$ was $a$ and $B$ was $b$
- we record transaction as

# Problem 4: how to know your balance

Assume that there is a public ledger

- It is not practical to maintain each and every account
- Solution: modify transaction recording to get the balance
- Suppose: $A$ have to give $B$ Rs. $x$. Initial balance of $A$ was $a$ and $B$ was $b$
- we record transaction as $A \rightarrow B$ Rs $x$
  $A \rightarrow A$ Rs $a - x$

# Problem 4: how to know your balance

Assume that there is a public ledger

- It is not practical to maintain each and every account
- Solution: modify transaction recording to get the balance
- Suppose: $A$ have to give $B$ Rs. $x$. Initial balance of $A$ was $a$ and $B$ was $b$
- we record transaction as $A \rightarrow B$ Rs $x$
  $A \rightarrow A$ Rs $a - x$
- Gain: you once you get to this transaction, you would never need to see previous transactions to get balance of $A$

# Problem 5: Double spending

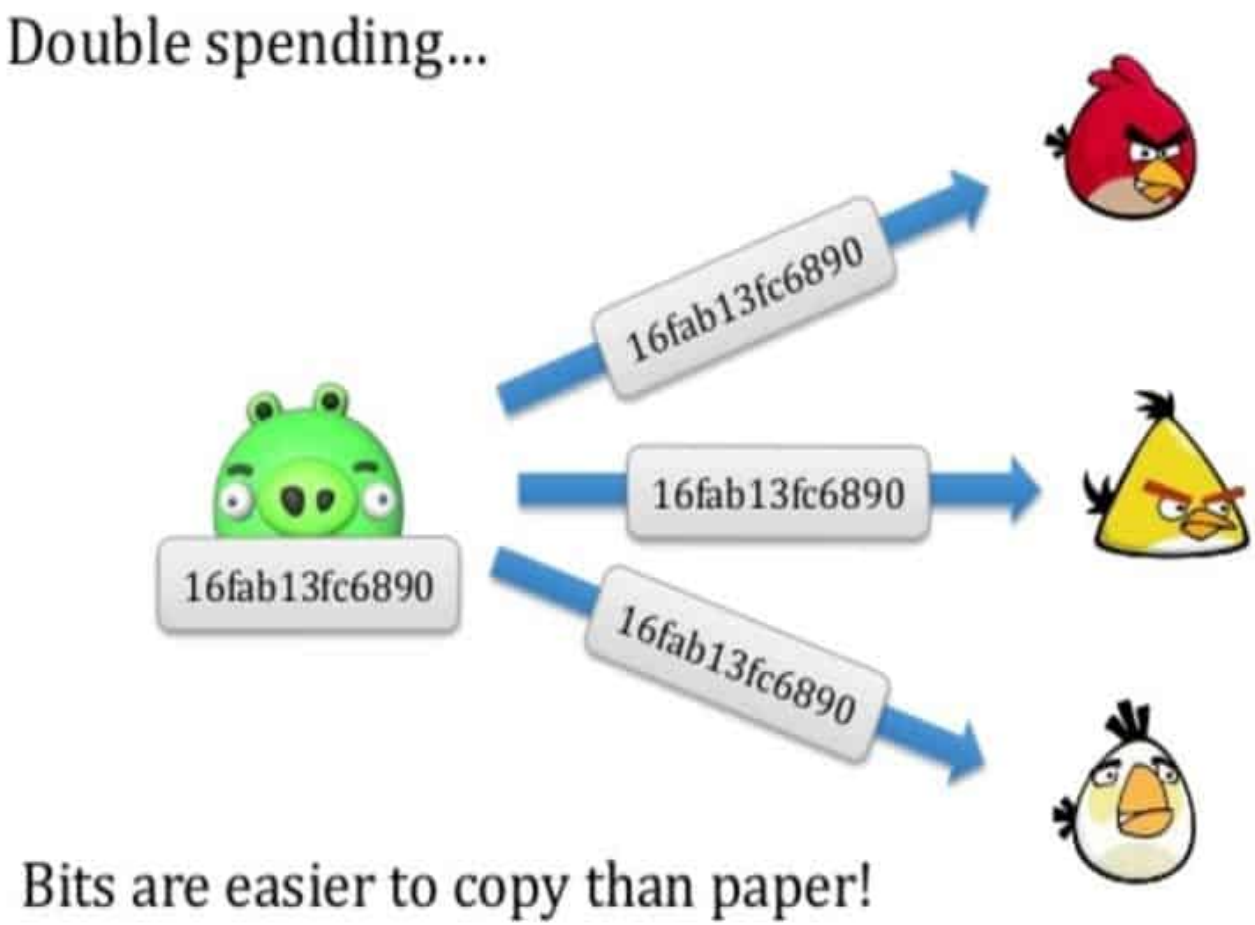



Figure: Double Spending

# Double spending

- We have created blocks and verified them

# Double spending

- We have created blocks and verified them
- integrity of each block is ensured

# Double spending

- We have created blocks and verified them
- integrity of each block is ensured
- we need to have integrity of all the block

# Double spending

- We have created blocks and verified them
- integrity of each block is ensured
- we need to have integrity of all the block

# A Chain of blocks

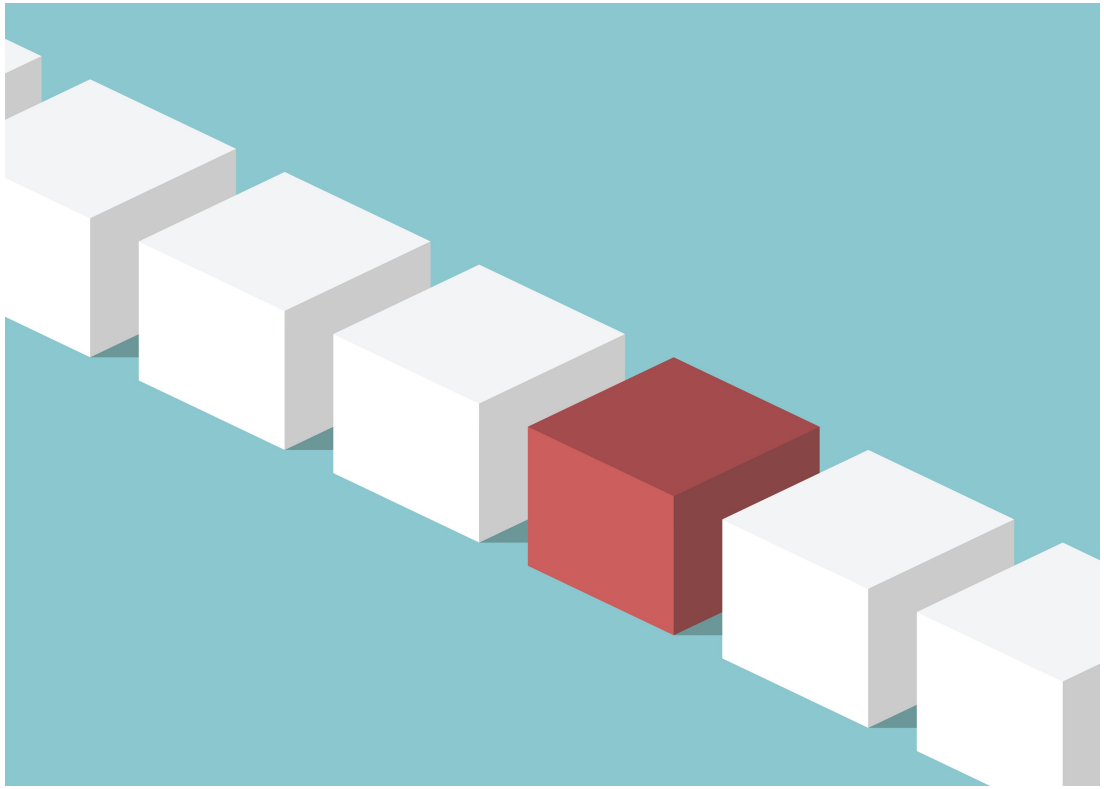


Figure: A chain of block

# A Chain of blocks

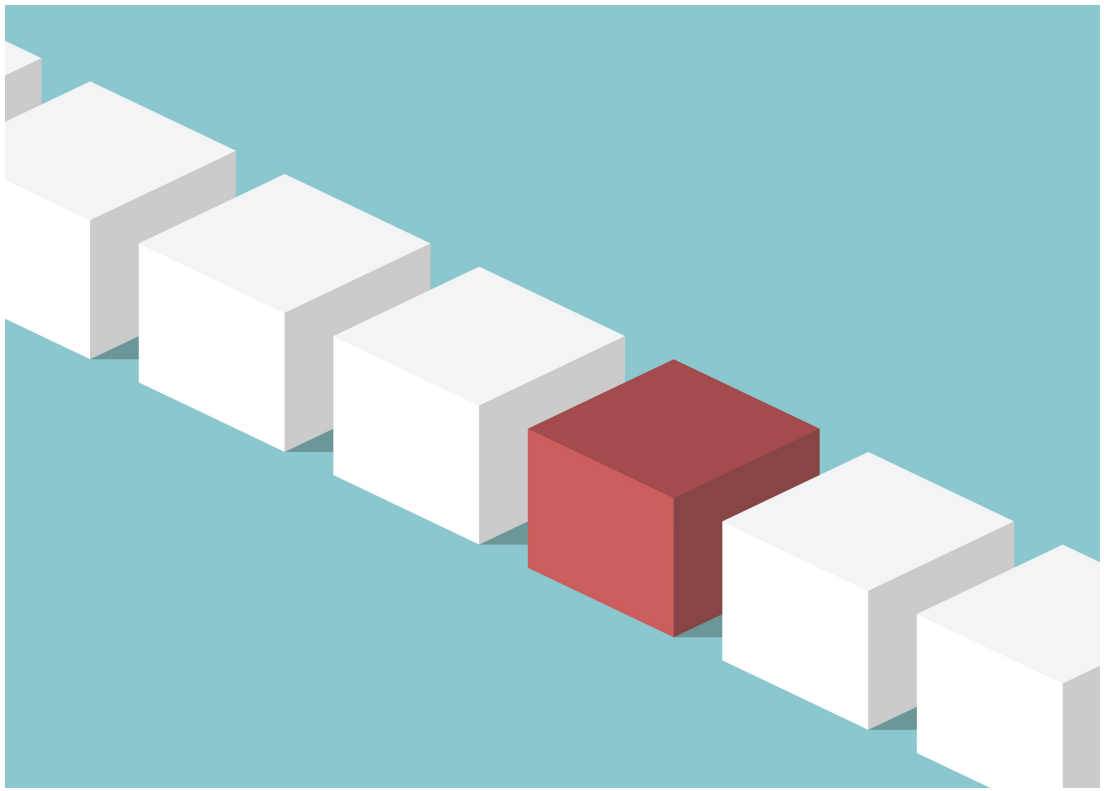


Figure: A chain of block

**Problem: How to organize them?**

# A Chain of blocks

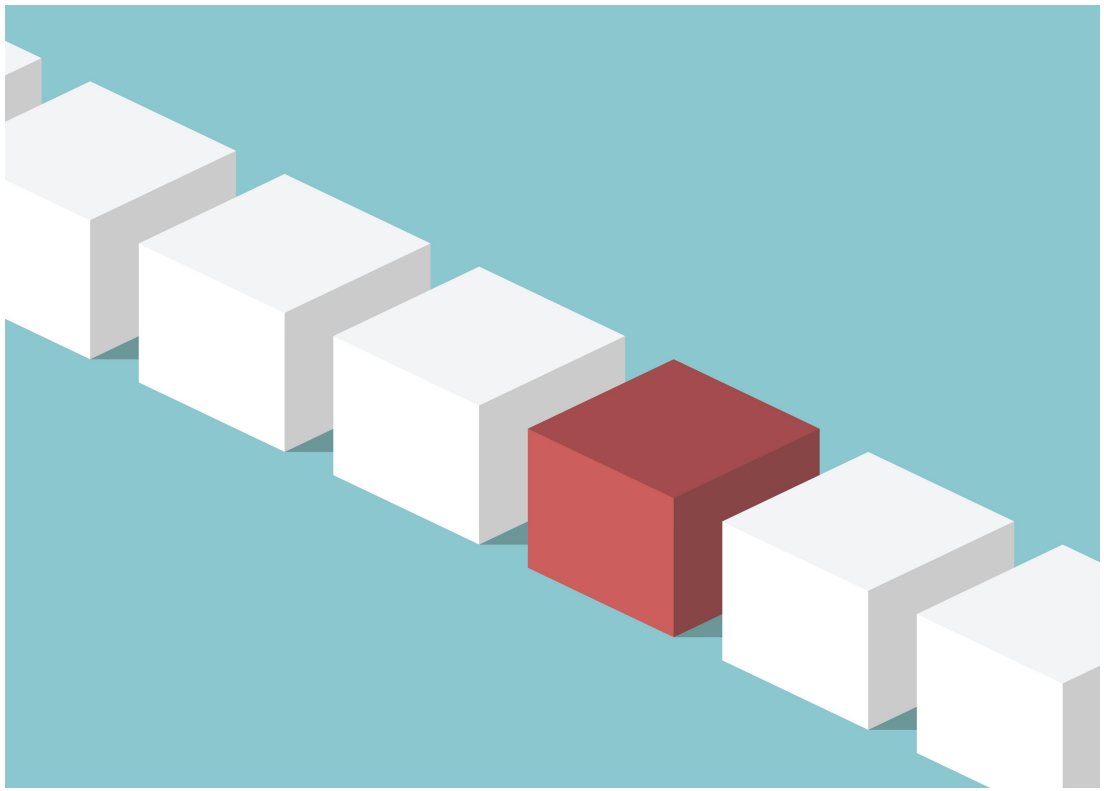


Figure: A chain of block

**Problem: How to organize them?**

**Solution: Include hash of previous block**

# A Chain of blocks

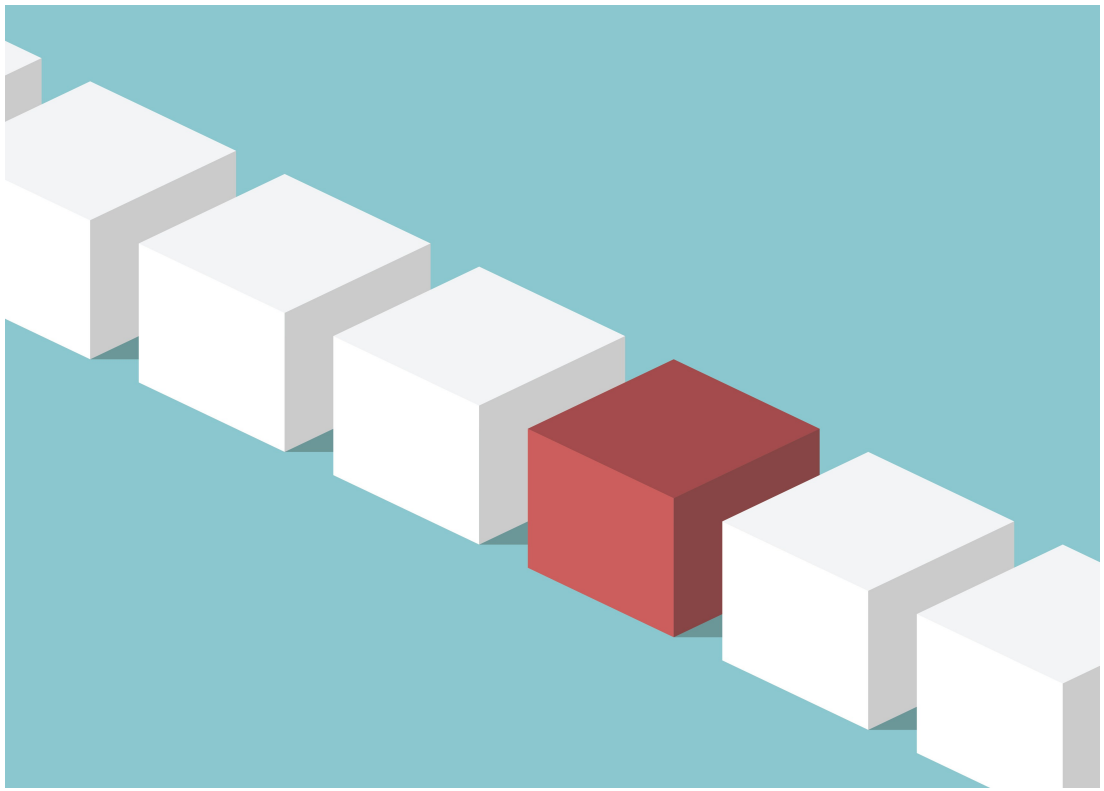


Figure: A chain of block

**Problem: How to organize them?**

**Solution: Include hash of previous block**