Bitcoin and Beyond

Darren Tapp
Director of Customer Service
SnoopWall LLC
darren.tapp@snoopwall.com tappdarren@gmail.com
http://www.snoopwall.com http://www.darrentapp.com
Co-host of Neocash Radio
http://www.neocashradio.com

Feburary 19, 2014

SnoopWall
RECLAIM YOUR PRIVACY ™

Thanks

I would like to extend a Thank you to:

SnoopWall

RECLAIM YOUR PRIVACY ™

Thanks

I would like to extend a Thank you to:

- Uli Walther – my advisor at Purdue
- Joseph Lipman – Who was kind enough to offer an elliptic curves course.
- The Boston chapter of the National Information Security Group

Outline of topics discussed

SnoopWall

RECLAIM YOUR PRIVACY ™

Outline of topics discussed

- Elliptic Curves

Outline of topics discussed

- Elliptic Curves
- hash functions `SHA256` and `RIPEMD160`

SnoopWall

RECLAIM YOUR PRIVACY ™

Outline of topics discussed

- Elliptic Curves
- hash functions `SHA256` and `RIPEMD160`
- Private and public keys, and wallets

Outline of topics discussed

- Elliptic Curves
- hash functions `SHA256` and `RIPEMD160`
- Private and public keys, and wallets
- Digital signatures and transactions

Outline of topics discussed

- Elliptic Curves
- hash functions `SHA256` and `RIPEMD160`
- Private and public keys, and wallets
- Digital signatures and transactions
- Known attacks on ECDSA

Outline of topics discussed

- Elliptic Curves
- hash functions `SHA256` and `RIPEMD160`
- Private and public keys, and wallets
- Digital signatures and transactions
- Known attacks on ECDSA
- Proof of work and mining

Outline of topics discussed

- Elliptic Curves
- hash functions `SHA256` and `RIPEMD160`
- Private and public keys, and wallets
- Digital signatures and transactions
- Known attacks on ECDSA
- Proof of work and mining
- What's after ECDSA?

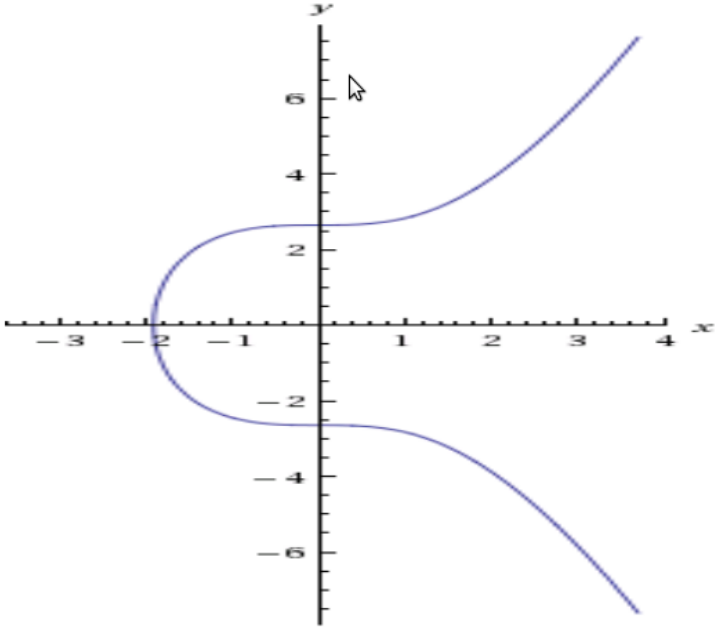
ELLIPTIC CURVES ARE BEAUTIFUL!

SnoopWall

RECLAIM YOUR PRIVACY ™

ELLIPTIC CURVES ARE BEAUTIFUL!

$$y^2 = x^3 + 7$$



Rendered by
Wolframalpha.com

We will need a finite arithmetic

We will need a finite arithmetic

For example addition modulo 5

We will need a finite arithmetic

For example addition modulo 5

$+$	0	1	2	3	4
0	0	1	2	3	4
1	1	2	3	4	0
2	2	3	4	0	1
3	3	4	0	1	2
4	4	0	1	2	3

We will need a finite arithmetic

And multiplication modulo 5

We will need a finite arithmetic

And multiplication modulo 5

$\times$	0	1	2	3	4
0	0	0	0	0	0
1	0	1	2	3	4
2	0	2	4	1	3
3	0	3	1	4	2
4	0	4	3	2	1

We will need a finite arithmetic

And multiplication modulo 5

$\times$	0	1	2	3	4
0	0	0	0	0	0
1	0	1	2	3	4
2	0	2	4	1	3
3	0	3	1	4	2
4	0	4	3	2	1

It was important that our choice of 5 was prime.

The Elliptic curve used by Bitcoin

The specifications of the Elliptic curve used by Bitcoin

The Elliptic curve used by Bitcoin

The specifications of the Elliptic curve used by Bitcoin

- The curve is named `secp256k1`

The Elliptic curve used by Bitcoin

The specifications of the Elliptic curve used by Bitcoin

- The curve is named `secp256k1`
- It is defined by the equation $y^2 = x^3 + 7$

The Elliptic curve used by Bitcoin

The specifications of the Elliptic curve used by Bitcoin

- The curve is named `secp256k1`
- It is defined by the equation $y^2 = x^3 + 7$
- The prime used is $p = 2^{256} - 2^{32} - 2^9 - 2^8 - 2^7 - 2^6 - 2^4 - 1$

The Elliptic curve used by Bitcoin

The specifications of the Elliptic curve used by Bitcoin

- The curve is named `secp256k1`
- It is defined by the equation $y^2 = x^3 + 7$
- The prime used is $p = 2^{256} - 2^{32} - 2^9 - 2^8 - 2^7 - 2^6 - 2^4 - 1$ *instead of 5 above*

The Elliptic curve used by Bitcoin

The specifications of the Elliptic curve used by Bitcoin

- The curve is named `secp256k1`
- It is defined by the equation $y^2 = x^3 + 7$
- The prime used is $p = 2^{256} - 2^{32} - 2^9 - 2^8 - 2^7 - 2^6 - 2^4 - 1$ *instead of 5 above*
- There is a specific known solution of this curve.

The Elliptic curve used by Bitcoin

The specifications of the Elliptic curve used by Bitcoin

- The curve is named `secp256k1`
- It is defined by the equation $y^2 = x^3 + 7$
- The prime used is $p = 2^{256} - 2^{32} - 2^9 - 2^8 - 2^7 - 2^6 - 2^4 - 1$ *instead of 5 above*
- There is a specific known solution of this curve. We'll discuss this later.

What's the big deal about elliptic curves?

SnoopWall

RECLAIM YOUR PRIVACY ™

What's the big deal about elliptic curves?

Why elliptic curves?

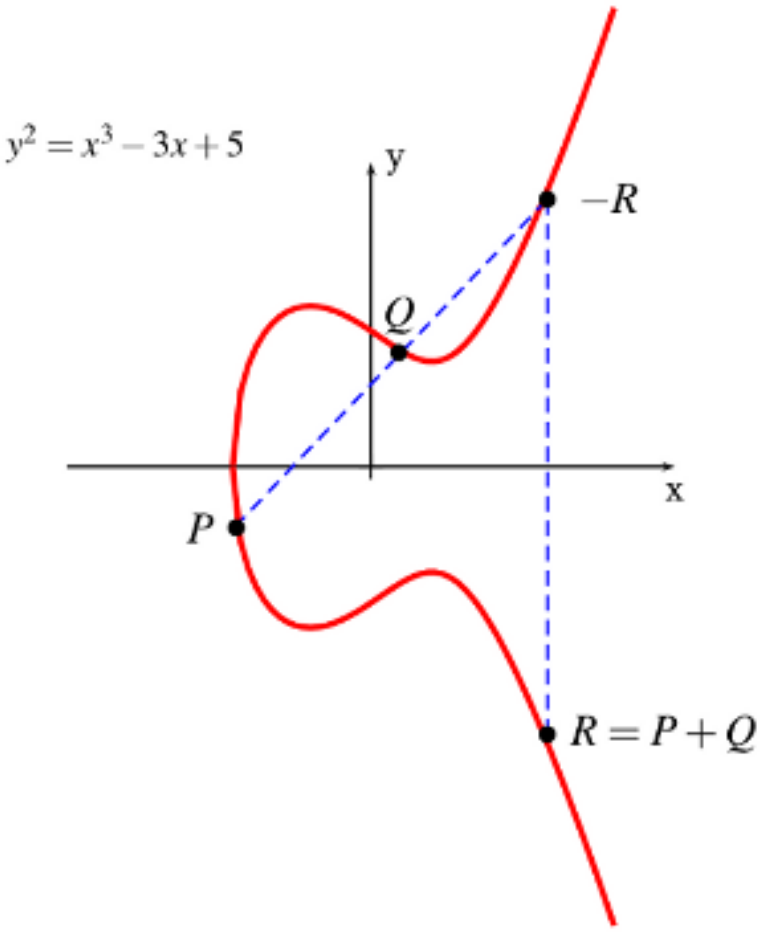
SnoopWall

RECLAIM YOUR PRIVACY ™

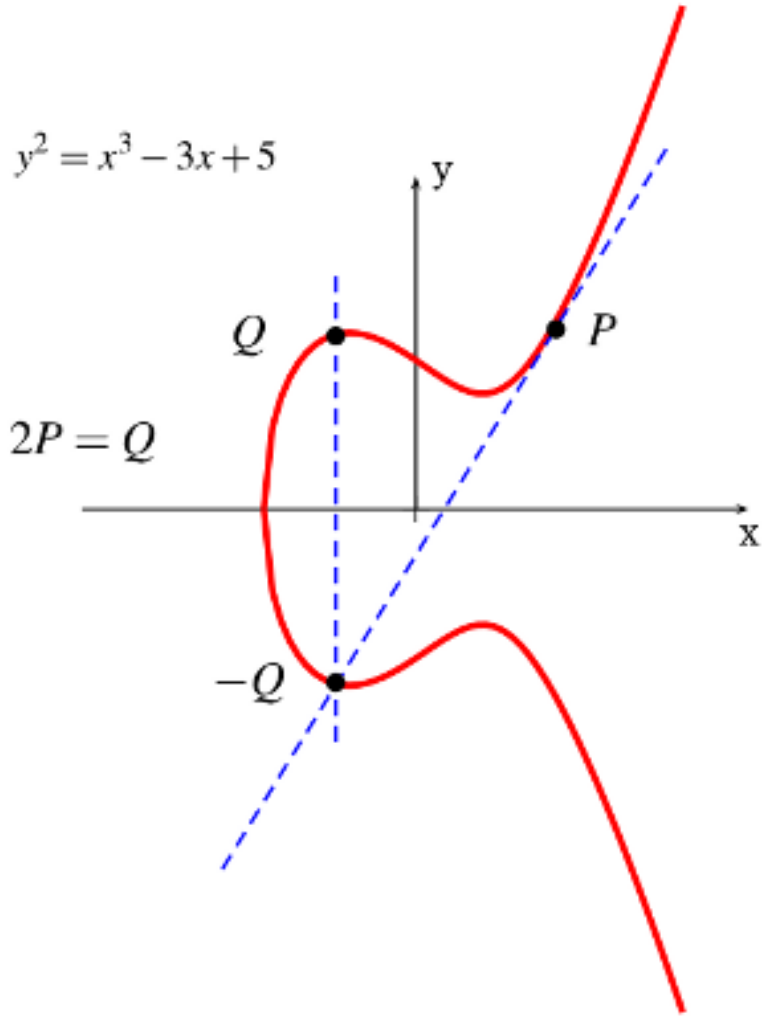
What's the big deal about elliptic curves?

Why elliptic curves?
There is a way to "add" two solutions of an elliptic curve together to get another solution.

Adding points on an Elliptic Curve



Adding points on an Elliptic Curve



Hash functions: SHA256

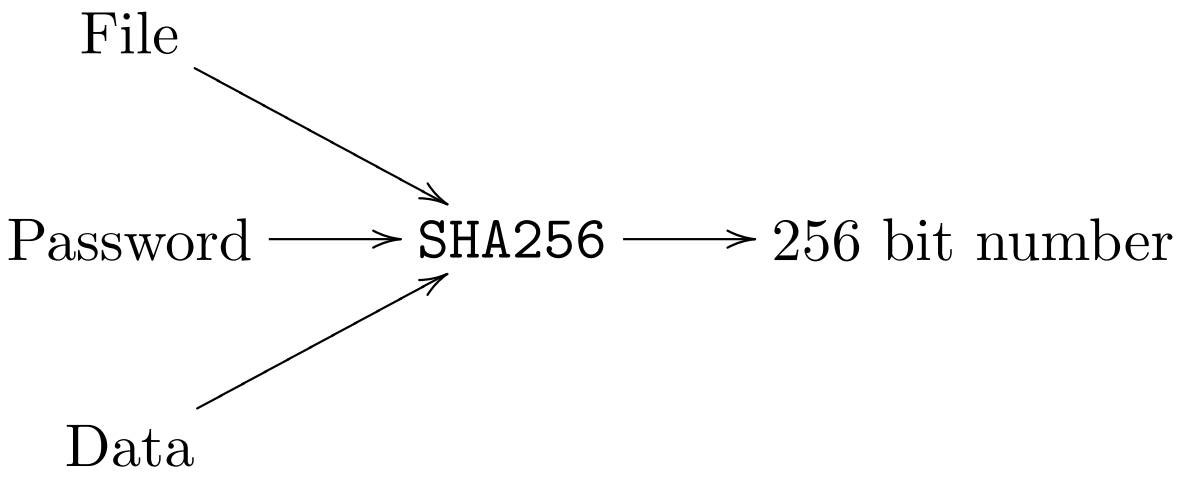
SHA256 is a hash function which takes an input,

SnoopWall

RECLAIM YOUR PRIVACY ™

Hash functions: SHA256

SHA256 is a hash function which takes an input,



and has a 256 bit output.

Hash functions

You probably are familiar with hash functions because they're used in password protection.

Hash functions

You probably are familiar with hash functions because they're used in password protection.
A good hash function has these properties:

Hash functions

You probably are familiar with hash functions because they're used in password protection.
A good hash function has these properties:

- Is difficult/impossible to undo

Hash functions

You probably are familiar with hash functions because they're used in password protection.
A good hash function has these properties:

- Is difficult/impossible to undo
- A small change in the input produces a wild change in the output

Hash functions

You probably are familiar with hash functions because they're used in password protection.
A good hash function has these properties:

- Is difficult/impossible to undo
- A small change in the input produces a wild change in the output
- The number of outputs must be large enough that a table of the outputs can't be constructed.

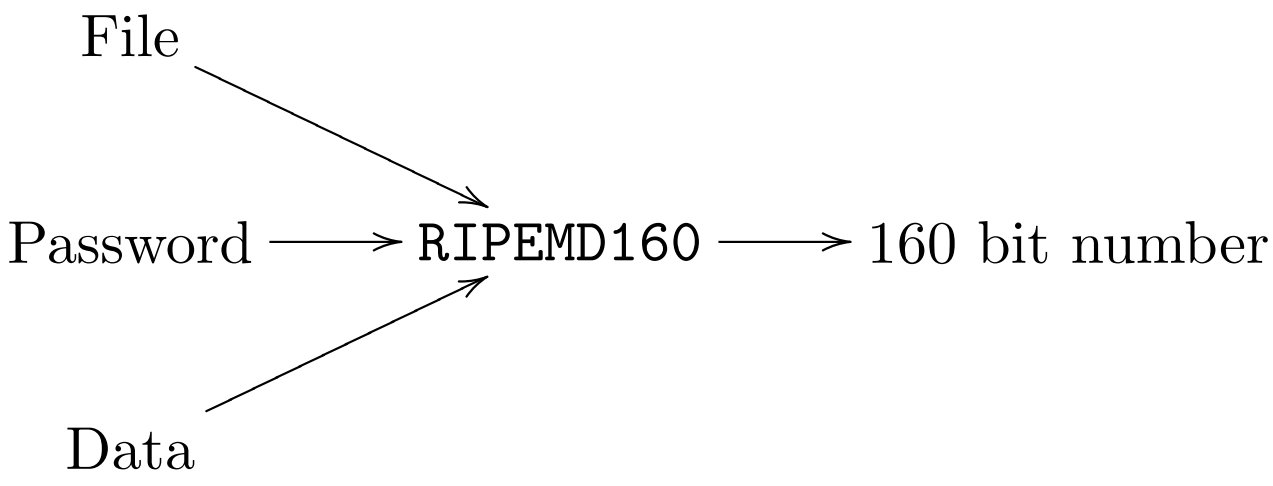
Hash functions

You probably are familiar with hash functions because they're used in password protection.
A good hash function has these properties:

- Is difficult/impossible to undo
- A small change in the input produces a wild change in the output
- The number of outputs must be large enough that a table of the outputs can't be constructed.

In the case of `SHA256` there are 2^{256} outputs.

Hash functions: RIPEMD160



Public, private keys, and addresses

SnoopWall

RECLAIM YOUR PRIVACY ™

Public, private keys, and addresses

- Remember that the elliptic curve protocol chooses a special point of the curve. I'll call this point E.

Public, private keys, and addresses

- Remember that the elliptic curve protocol chooses a special point of the curve. I'll call this point E.
- A private key is a *number*.

Public, private keys, and addresses

- Remember that the elliptic curve protocol chooses a special point of the curve. I'll call this point E.
- A private key is a *number*.
- The public key associated to a private key is E added to itself that number of times.

Public, private keys, and addresses

- Remember that the elliptic curve protocol chooses a special point of the curve. I'll call this point E.
- A private key is a *number*.
- The public key associated to a private key is E added to itself that number of times.
- The address associated to the key is the `RIPEMD160` hash of the `SHA256` hash of the public key.

Public, private keys, and addresses

- Remember that the elliptic curve protocol chooses a special point of the curve. I'll call this point E.
- A private key is a *number*.
- The public key associated to a private key is E added to itself that number of times.
- The address associated to the key is the `RIPEMD160` hash of the `SHA256` hash of the public key.
- An address, `1L5rFEcJUix2b9q6u2n7yqyLEzSxYhB4a1`, is a base 58 encoding of that hash. Always with a 1 at the beginning.

Public, private keys, and addresses

In math speak:

Public, private keys, and addresses

In math speak:

- Private key: a number, n in the 0 to 2^{256} range

Public, private keys, and addresses

In math speak:

- Private key: a number, n in the 0 to 2^{256} range
- Public key: E added to itself n times:

$$P = \underbrace{E + E + \cdots + E}_{n \text{ times}}$$

Public, private keys, and addresses

In math speak:

- Private key: a number, n in the 0 to 2^{256} range
- Public key: E added to itself n times:

$$P = \underbrace{E + E + \cdots + E}_{n \text{ times}}$$

- Address 1 and then $\texttt{RIPEMD160}(\texttt{SHA256}(P))$ encoded in base 58.

Why public and private keys?

SnoopWall

RECLAIM YOUR PRIVACY ™

Why public and private keys?

Given the public key it is difficult/impossible to compute the private key.

SnoopWall

RECLAIM YOUR PRIVACY ™

Why public and private keys?

Given the public key it is difficult/impossible to compute the private key.
This problem is called a discrete logarithm problem.

Why public and private keys?

Given the public key it is difficult/impossible to compute the private key.
This problem is called a discrete logarithm problem.
It is possible to prove you know the private key associated to a public key, without divulging the private key.

Wallets

A bitcoin wallet is:

SnoopWall

RECLAIM YOUR PRIVACY ™

Wallets

A bitcoin wallet is:
A file that contains private, public keys and addresses. It usually will contain many keys. It may also contain labeling information.

Digital Signatures

SnoopWall

RECLAIM YOUR PRIVACY ™

Digital Signatures

We can use our elliptic curve to construct a Digital signature algorithm.

Digital Signatures

We can use our elliptic curve to construct a Digital signature algorithm.

- A random number is needed to construct a signature.

Digital Signatures

We can use our elliptic curve to construct a Digital signature algorithm.

- A random number is needed to construct a signature.
- In bitcoin a transaction, to spend from an address requires a signature by the corresponding private key.

Digital Signatures

We can use our elliptic curve to construct a Digital signature algorithm.

- A random number is needed to construct a signature.
- In bitcoin a transaction, to spend from an address requires a signature by the corresponding private key.
- A transaction can draw from one or more unspent inputs. If it draws from one or more addresses, then multiple signatures are needed.

Digital Signatures

We can use our elliptic curve to construct a Digital signature algorithm.

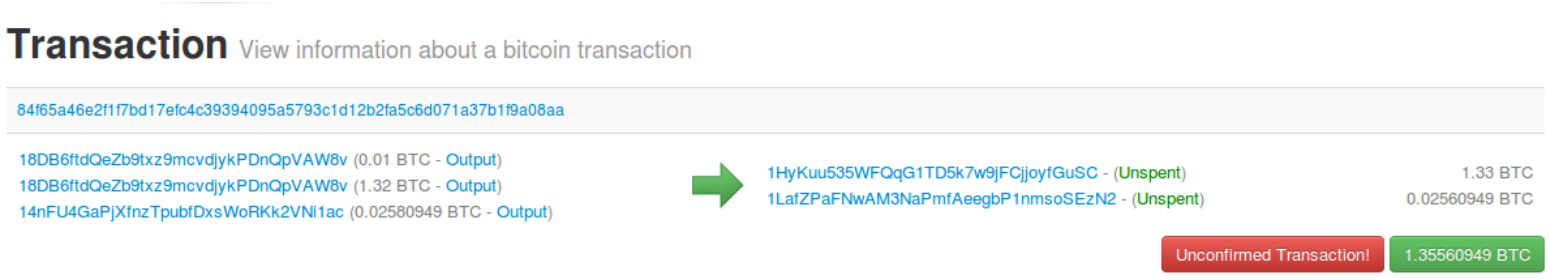
- A random number is needed to construct a signature.
- In bitcoin a transaction, to spend from an address requires a signature by the corresponding private key.
- A transaction can draw from one or more unspent inputs. If it draws from one or more addresses, then multiple signatures are needed.
- A transaction can have one or more outputs.

Digital Signatures

We can use our elliptic curve to construct a Digital signature algorithm.

- A random number is needed to construct a signature.
- In bitcoin a transaction, to spend from an address requires a signature by the corresponding private key.
- A transaction can draw from one or more unspent inputs. If it draws from one or more addresses, then multiple signatures are needed.
- A transaction can have one or more outputs.
- An input must be spent in full. Any amount that is too much can be sent back to an address owned by the spender.

A transaction



Transactions

- A public key is not published until after a transaction is made.

Transactions

- A public key is not published until after a transaction is made.
- A transaction could include a fee by making the outputs less than the inputs.

Transactions

- A public key is not published until after a transaction is made.
- A transaction could include a fee by making the outputs less than the inputs.
- Some transactions that are non-standard and will not be relayed by the network.

Transactions

- A public key is not published until after a transaction is made.
- A transaction could include a fee by making the outputs less than the inputs.
- Some transactions that are non-standard and will not be relayed by the network. This helps prevent a DDOS.

Known attacks on ECDSA

If only a public key is known:

SnoopWall

RECLAIM YOUR PRIVACY ™

Known attacks on ECDSA

If only a public key is known:

This is a discrete logarithm problem.

Known attacks on ECDSA

If only a public key is known:

This is a discrete logarithm problem.

There has been some work done in small characteristic. Since the characteristic that Bitcoin uses is so large $\sim 2^{256}$, this won't work.

Known attacks on ECDSA

If a signature is known:

SnoopWall

RECLAIM YOUR PRIVACY ™

Known attacks on ECDSA

If a signature is known:

If the random number picked for a signature is used twice then we can solve for the private key.

Known attacks on ECDSA

If a signature is known:

If the random number picked for a signature is used twice then we can solve for the private key. (Android)

Known attacks on ECDSA

If a signature is known:

If the random number picked for a signature is used twice then we can solve for the private key. (Android)

if the random number or private key is too big or too small then a lattice decent algorithm *may* provide the private key.

Known attacks on ECDSA

If a signature is known:

If the random number picked for a signature is used twice then we can solve for the private key. (Android)

if the random number or private key is too big or too small then a lattice decent algorithm *may* provide the private key.

Poulakis, Dimitrios, *Some Lattice Attacks on DSA and ECDSA*, Applicable Algebra in Engineering Communication and Computing, 2011

Known attacks on ECDSA

If a signature is known:

If the random number picked for a signature is used twice then we can solve for the private key. (Android)

if the random number or private key is too big or too small then a lattice decent algorithm *may* provide the private key.

Poulakis, Dimitrios, *Some Lattice Attacks on DSA and ECDSA*, Applicable Algebra in Engineering Communication and Computing, 2011

If two signatures are known this method becomes more robust.

For safety sake

Don't use the same address twice.

SnoopWall

RECLAIM YOUR PRIVACY ™

Mining

What is bitcoin mining?

SnoopWall

RECLAIM YOUR PRIVACY ™

Mining

What is bitcoin mining?

- Bitcoin mining is the process which allows disjoint actors to agree on a particular transaction ledger.

Mining

What is bitcoin mining?

- Bitcoin mining is the process which allows disjoint actors to agree on a particular transaction ledger.
- Transactions are bunched together called a block.

Mining

What is bitcoin mining?

- Bitcoin mining is the process which allows disjoint actors to agree on a particular transaction ledger.
- Transactions are bunched together called a block.
- A miner approves a block (hard)

Mining

What is bitcoin mining?

- Bitcoin mining is the process which allows disjoint actors to agree on a particular transaction ledger.
- Transactions are bunched together called a block.
- A miner approves a block (hard)
- The block is broadcast to other nodes which verify the block as valid.

Mining

What is bitcoin mining?

- Bitcoin mining is the process which allows disjoint actors to agree on a particular transaction ledger.
- Transactions are bunched together called a block.
- A miner approves a block (hard)
- The block is broadcast to other nodes which verify the block as valid.
- The bitcoins out of a block is 25 more than what goes in.

Mining

What is bitcoin mining?

- Bitcoin mining is the process which allows disjoint actors to agree on a particular transaction ledger.
- Transactions are bunched together called a block.
- A miner approves a block (hard)
- The block is broadcast to other nodes which verify the block as valid.
- The bitcoins out of a block is 25 more than what goes in. (This has changed and will change.)

Mining

What is bitcoin mining?

SnoopWall

RECLAIM YOUR PRIVACY ™

Mining

What is bitcoin mining?

- Each next block must have the hash of the previous block.

Mining

What is bitcoin mining?

- Each next block must have the hash of the previous block. It's a chain

Mining

What is bitcoin mining?

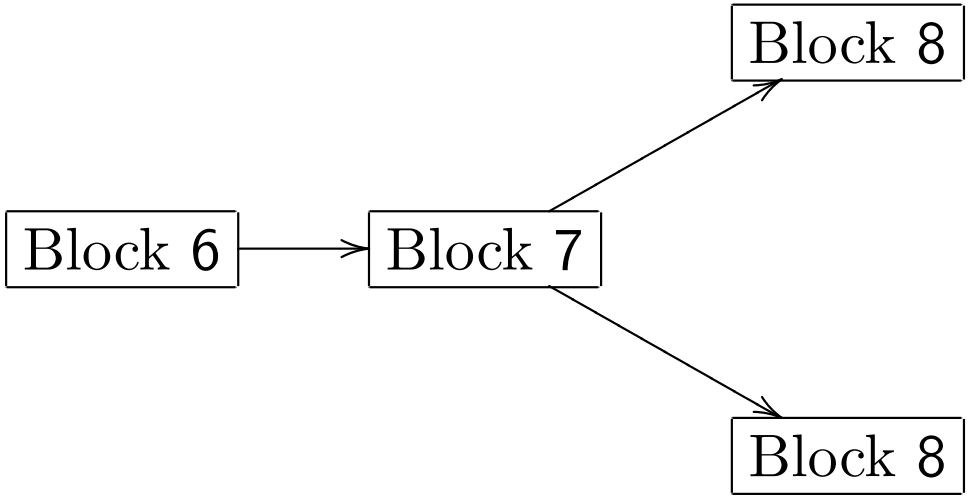
- Each next block must have the hash of the previous block. It's a chain
- The longest chain is always considered the valid one.

What if two blocks don't agree?

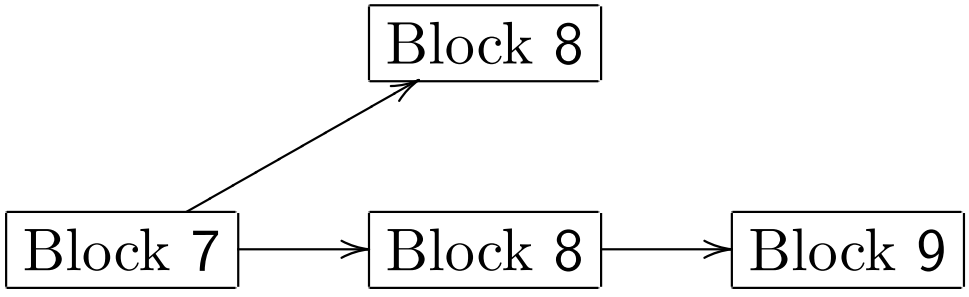
SnoopWall

RECLAIM YOUR PRIVACY ™

What if two blocks don't agree?



What if two blocks don't agree?



How does a miner “approve” a block?

SnoopWall

RECLAIM YOUR PRIVACY ™

How does a miner “approve” a block?

- A miner approves a block by hashing the block and a random number, called a nonce, together.

How does a miner “approve” a block?

- A miner approves a block by hashing the block and a random number, called a nonce, together.
- If the resulting hash is small enough the block is accepted by the network.

How does a miner “approve” a block?

- A miner approves a block by hashing the block and a random number, called a nonce, together.
- If the resulting hash is small enough the block is accepted by the network.
- The network will adjust the difficulty in order to target one block every 10 minutes.

How does a miner “approve” a block?

- A miner approves a block by hashing the block and a random number, called a nonce, together.
- If the resulting hash is small enough the block is accepted by the network.
- The network will adjust the difficulty in order to target one block every 10 minutes.
- The “small enough” threshold gets smaller when more computers join the network.

How does a miner "approve" a block?

- A miner approves a block by hashing the block and a random number, called a nonce, together.
- If the resulting hash is small enough the block is accepted by the network.
- The network will adjust the difficulty in order to target one block every 10 minutes.
- The "small enough" threshold gets smaller when more computers join the network. This makes the network more secure as it is harder to approve a block.

How does a miner "approve" a block?

- A miner approves a block by hashing the block and a random number, called a nonce, together.
- If the resulting hash is small enough the block is accepted by the network.
- The network will adjust the difficulty in order to target one block every 10 minutes.
- The "small enough" threshold gets smaller when more computers join the network. This makes the network more secure as it is harder to approve a block.
- Finding a good nonce is called finding a block.

Upgrading ECDSA

Possible ways to upgrade ECDSA of bitcoin are Lamport Signatures and larger fields for the curves.

Upgrading ECDSA

Possible ways to upgrade ECDSA of bitcoin are Lamport Signatures and larger fields for the curves. Each has drawbacks.

Upgrading ECDSA

Possible ways to upgrade ECDSA of bitcoin are Lamport Signatures and larger fields for the curves. Each has drawbacks.

- Lamport public and private keys are too big. Bandwidth will be an issue.

Upgrading ECDSA

Possible ways to upgrade ECDSA of bitcoin are Lamport Signatures and larger fields for the curves. Each has drawbacks.

- Lamport public and private keys are too big. Bandwidth will be an issue.
- What if there is a quantum attack on ECDSA. Larger fields won't be much help.

Upgrading ECDSA

Possible ways to upgrade ECDSA of bitcoin are Lamport Signatures and larger fields for the curves. Each has drawbacks.

- Lamport public and private keys are too big. Bandwidth will be an issue.
- What if there is a quantum attack on ECDSA. Larger fields won't be much help.
- The current protocol is resistant to an ECDSA attack because of the use of hash functions.