IT486 v3.0: Blockchains and Cryptocurrencies

Bitcoin privacy techniques: confidential transactions

Amount privacy: Motivation

- If people can see how many coins you have, they could charge you more or try to rob you, etc.

Amount privacy: Motivation

- If people can see how many coins you have, they could charge you more or try to rob you, etc.
- We can try to improve privacy by hiding amounts

Amount privacy: Motivation

- If people can see how many coins you have, they could charge you more or try to rob you, etc.
- We can try to improve privacy by hiding amounts
- But from whom?

Amount privacy: Motivation

- If people can see how many coins you have, they could charge you more or try to rob you, etc.
- We can try to improve privacy by hiding amounts
- But from whom?
  - People receiving payments should probably know how much they're receiving

Amount privacy: Motivation

- If people can see how many coins you have, they could charge you more or try to rob you, etc.
- We can try to improve privacy by hiding amounts
- But from whom?
  - People receiving payments should probably know how much they're receiving
  - People sending should also know how much they're sending

Hidden amount txn

- network view:

input 0 user A signature _ coins	output 0 address C _ coins
input 1 user B signature _ coins	output 1 address D _ coins

Hidden amount txn

- sender view:

input 0 user A signature 2 coins	output 0 address C 7 coins
input 1 user B signature 7 coins	output 1 address D 2 coins

Hidden amount txn

- receiver view:

input 0 user A signature _ coins	output 0 address C _ coins
input 1 user B signature _ coins	output 1 address D 2 coins

Hidden amount txn

- to the network we want to prove a sum:

$$w + x = y + z$$

without disclosing the amounts w, x, y, z

input 0 user A signature w coins	output 0 address C y coins
input 1 user B signature x coins	output 1 address D z coins

Recap: Commitments

- commit(value) $\rightarrow c$
- reveal value
- verify(c, value) $\rightarrow$ bool

Recap: Hash commitments

- Choose random r = b8bc7579
- hash(5, r) = 4dd8fa60
- to reveal, reveal both 5 and r

Recap: Hash commitments

- Choose random $r =$ b8bc7579
- hash(5, r) $=$ 4dd8fa60
- to reveal, reveal both 5 and r
- useful, but we want to be able to prove things about commitments

Homomorphic commitments

- commit(x) $\rightarrow a$
- commit(y) $\rightarrow b$
- reveal $z = x + y$
- verify(z, $a + b$) $\rightarrow$ true

Homomorphic commitments

- commit(x) $\rightarrow$ a
- commit(y) $\rightarrow$ b
- reveal $z = x + y$
- verify(z, $a + b$) $\rightarrow$ true
- This could be very useful: can prove a sum without revealing the constituent parts

Commitments on a curve

- commit(x) $\rightarrow xG$ $(= X)$
- commit(y) $\rightarrow yG$ $(= Y)$
- reveal $z = x + y$

Commitments on a curve

- commit(x) $\to xG$ ($= X$)
- commit(y) $\to yG$ ($= Y$)
- reveal $z = x + y$
- why won't this work?

Commitments on a curve

- commit(x) $\rightarrow xG$ $(= X)$
- commit(y) $\rightarrow yG$ $(= Y)$
- reveal $z = x + y$
- why won't this work?
- Not blinded: $X = 5G$, easy to guess 5

Commitments on a curve

- Try $X = (5 + r)G$; reveal 5 and r
- Why won't this work?

Commitments on a curve

- Try $X = (5 + r)G$; reveal 5 and r
- Why won't this work?
- Not binding
  - find $r' = (5 + r) - 6$
  - $6 + r' = 5 + r$ so X is the same
  - reveal 6, r'

Commitments on a curve

- Try $X = (5 + r)G$; reveal 5 and r
- Why won't this work?
- Not binding
  - find $r' = (5 + r) - 6$
  - $6 + r' = 5 + r$ so X is the same
  - reveal 6, r'
- use $hash(5, r)G$ ...?
- but then no longer homomorphic

Pedersen commitments

- introducing G's twin, H
- H is another generator point distinct from G

Pedersen commitments

- introducing G's twin, H
- H is another generator point distinct from G
- Verifier chooses a secret n such that $nG = H$ (pick a random point on the curve)

Pedersen commitments

- introducing G's twin, H
- H is another generator point distinct from G
- Verifier chooses a secret n such that $nG = H$ (pick a random point on the curve)
- To commit, sender calculates: $X = rG + vH$
    - v is the value committed
    - r is a blinding factor

Pedersen commitments

- $X = rG + vH$
- binding:

Pedersen commitments

- $X = rG + vH$
- binding:
  - Sender can't come up with another r, v that gets him to X

Pedersen commitments

- $X = rG + vH$
- binding:
  - Sender can't come up with another r, v that gets him to X
- hiding:
  - For the verifier, every value v is equally likely to be the value committed in X

Pedersen commitments

- $X = r_1 G + v_1 H,\ Y = r_2 G + v_2 H$
- Homomorphicity:

$$Z = X + Y = (r_1 + r_2)G + (v_1 + v_2)H$$

- We want to prove that $v = v_1 + v_2$ without revealing $v_1,\ v_2$

Pedersen commitments

- $X = r_1 G + v_1 H$, $Y = r_2 G + v_2 H$
- Homomorphicity:

$$Z = X + Y = (r_1 + r_2)G + (v_1 + v_2)H$$

- We want to prove that $v = v_1 + v_2$ without revealing v_1, v_2
- Reveal r, $v = r_1 + r_2$, $v_1 + v_2$

Pedersen commitments

- $X = r_1 G + v_1 H,\ Y = r_2 G + v_2 H$
- Homomorphicity:

$$Z = X + Y = (r_1 + r_2)G + (v_1 + v_2)H$$

- We want to prove that $v = v_1 + v_2$ without revealing $v_1,\ v_2$
- Reveal $r,\ v = r_1 + r_2,\ v_1 + v_2$
- Verifier can check if $rG + vH = Z$

Computationally binding

- A computationally bounded sender cannot open a commitment in two ways
- Claim: opening a Pedersen commitment in two ways is as hard as calculating a discrete log

Computationally binding

- A computationally bounded sender cannot open a commitment in two ways
- Claim: opening a Pedersen commitment in two ways is as hard as calculating a discrete log
- Proving this claim amounts to proving the following two results

Computationally binding

- A computationally bounded sender cannot open a commitment in two ways
- Claim: opening a Pedersen commitment in two ways is as hard as calculating a discrete log
- Proving this claim amounts to proving the following two results
  - **(lemma 1)** It is easy to violate binding if n is known to sender

Computationally binding

- A computationally bounded sender cannot open a commitment in two ways
- Claim: opening a Pedersen commitment in two ways is as hard as calculating a discrete log
- Proving this claim amounts to proving the following two results
  - **(lemma 1)** It is easy to violate binding if n is known to sender
  - **(lemma 2)** It is easy to determine n if commitment can be opened in two different ways

Proof of lemma 1

- We want to find r, r, v, v' such that

$$rG + vH = r'G + v'H$$

- This is easy when we know n

Proof of lemma 1

- We want to find r, r, v, v' such that

$$rG + vH = r'G + v'H$$

- This is easy when we know n
- Make all value except r random and set $r' = (v - v')n + r$

Proof of lemma 2

- We show that calculating n is easy when we can find r, r', v, v' such that

$$rG + vH = r'G + v'H$$

- Then

$$n = \frac{v - v'}{r - r'}$$

Perfectly hiding

- Even a computationally unbounded receiver cannot open the commited value
- Claim: For any r, v and any v' there is a r' such that

$$rG + vH = r'G + v'H$$

- Proof: choose $r' = (v - v')n + r$

Pedersen commitments

- binding, hiding, homomorphic
- great! We can prove sums

Pedersen amount txn

- network can verify that inputs = outputs
- by checking $W + X = Y + Z$
- just add up all the points on each side and make sure they are equal

input 0 user A signature $W = r_1 G + wH$ coins	output 0 address C $Y = r_3 G + yH$ coins
input 1 user B signature $X = r_2 G + xH$ coins	output 1 address D $Z = r_4 G + zH$ coins

Pedersen amount txn

- receiver learns own v, r
- sender privately reveals them to the receiver

input 0 user A signature $W = r_1 G + wH$ coins	output 0 address C $Y = r_3 G + yH$ coins
input 1 user B signature $X = r_2 G + xH$ coins	output 1 address D $Z = r_4 G + 2H$ coins

Blinding factors

- we want $r_1 + r_2 = r_3 + r_4$
- when making outputs, make all r's but the last random
- compute last r

Pedersen txn

- everyone can verify that inputs = outputs
- just add up all the points on both sides and make sure they're equal

Pedersen txn

- everyone can verify that inputs = outputs
- just add up all the points on both sides and make sure they're equal
- reveal output r, v to person receiving the coins
- don't forget own r (why?)

What about transaction fees?

- The transaction fee f cannot be deduced from inputs and outputs
- Therefore it is explicitly published
- everyone can verify that inputs - outputs $= fH$ (blinding factor $= 0$)

Pedersen amount tx

- can make invalid outputs
- just take points with no known r, v
- but no receiver will accept

input 0 user A signature $W = r_1 G + wH$ coins	output 0 address C $Y = r_3 G + yH$ coins
input 1 user B signature $X = r_2 G + xH$ coins	output 1 address D $Z = W + X - Y$

Pedersen amount tx

- can make invalid outputs using negative amounts!
- Ex: 2+7 = -99 + 108
- that negative output will be hidden

input 0 user A signature $W = r_1 G + 2H$ coins	output 0 address C $Y = r_3 G + -99H$ coins
input 1 user B signature $X = r_2 G + 7H$ coins	output 1 address D $Z = r_4 G + 108H$ coins

Confidential txs

- we need more than the proof the sums are equal
- we also need a proof that they're non-negative