Blockchains & Cryptocurrencies

Anonymity

Image from cryptonomad.info

Instructors: Matt Green & Abhishek Jain
Johns Hopkins University - Spring 2023

*Some slides based on NBFMG

Today

Today

- New Thread: **Anonymity**

Some say Bitcoin provides anonymity

“ Bitcoin is a secure and anonymous digital currency ”

— WikiLeaks donations page

Others say it doesn’t

“ Bitcoin won't hide you from the NSA's prying eyes”

— Wired UK

What do we mean by anonymity?

Literally: anonymous = without a name

Bitcoin addresses are public key hashes rather than real identities

Computer scientists call this pseudonymity

Anonymity in computer science

Anonymity = pseudonymity + unlinkability

Different interactions of the same user with the system should not be linkable to each other

Pseudonymity vs anonymity in forums

Reddit: pick a long-term pseudonym

vs.

4Chan: make posts with no attribution at all

Why is unlinkability needed?

1. Many Bitcoin services require real identity

2. Linked profiles can be deanonymized by a variety of side channels

Defining unlinkability in Bitcoin

- Hard to link different addresses of the same user
- Hard to link different transactions of the same user
- Hard to link sender of a “payment” to its recipient

Quantifying anonymity

Anonymity set: Anonymity set of a transaction T is the set of transactions which an adversary cannot distinguish from T.

To calculate anonymity set:

- define adversary model
- reason carefully about: what the adversary knows, does not know, and cannot know

Why anonymous cryptocurrencies?

Block chain based currencies are totally, publicly, and permanently traceable

Without anonymity, privacy is much worse than traditional banking!

Anonymous e-cash: history

Introduced by David Chaum, 1982

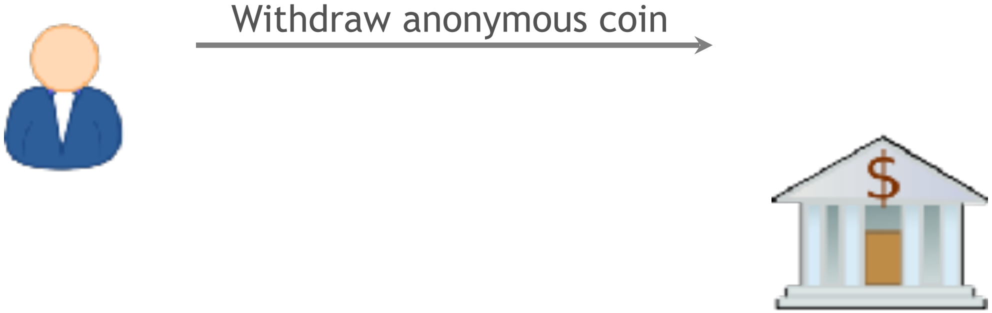
Blind signature: a two-party protocol to create digital signature without signer learning which message is being signed

Anonymous e-cash via blind signatures

User	Balance
…	…
	10
…	…
	5

Spent coins
…

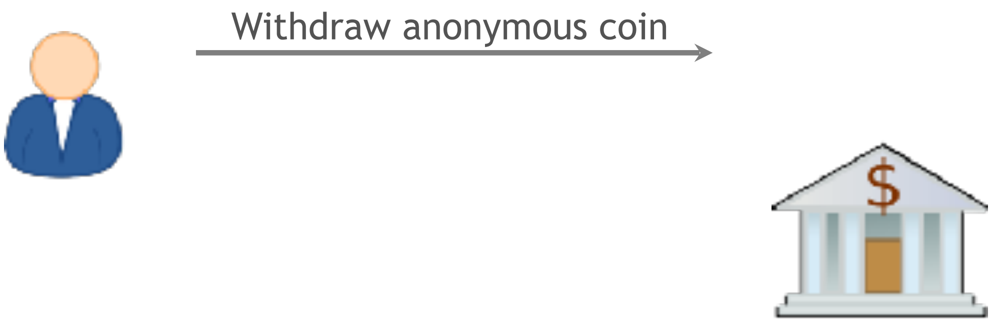
Anonymous e-cash via blind signatures



User	Balance
…	…
	10
…	…
	5

Spent coins
…

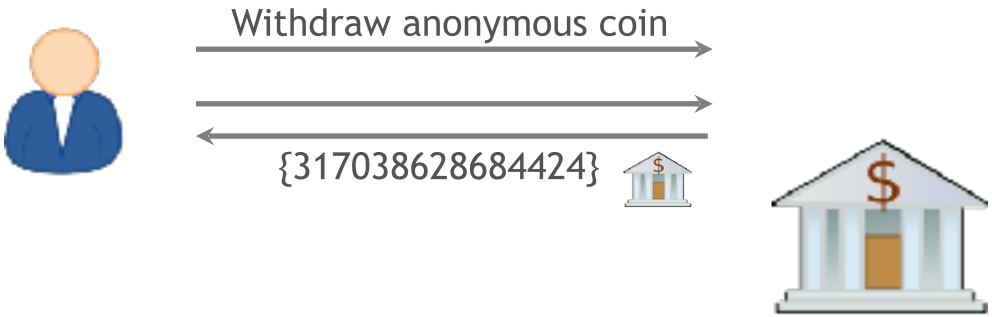
Anonymous e-cash via blind signatures



User	Balance
…	…
🔵	9
…	…
🔴	5

Spent coins
…

Anonymous e-cash via blind signatures



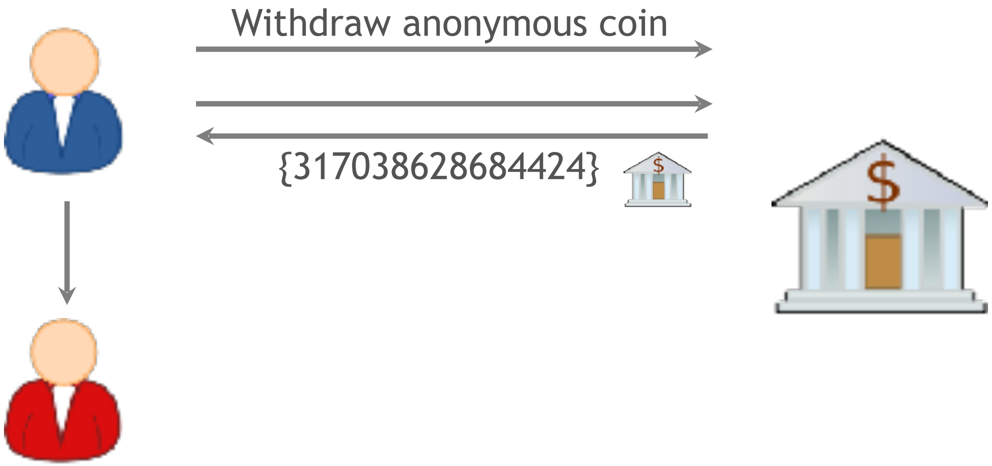
Withdraw anonymous coin

{317038628684424}

User	Balance
…	…
	9
…	…
	5

Spent coins
…

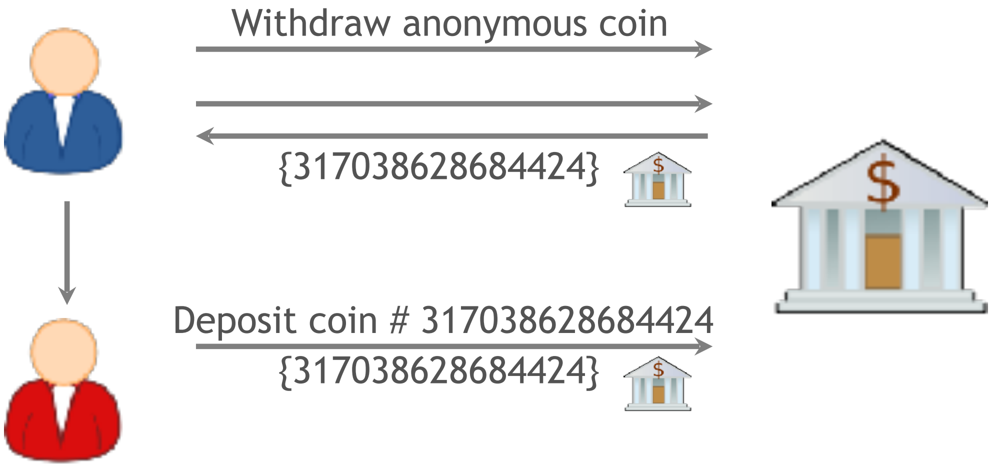
Anonymous e-cash via blind signatures



User	Balance
…	…
	9
…	…
	5

Spent coins
…

Anonymous e-cash via blind signatures



Withdraw anonymous coin

{317038628684424}

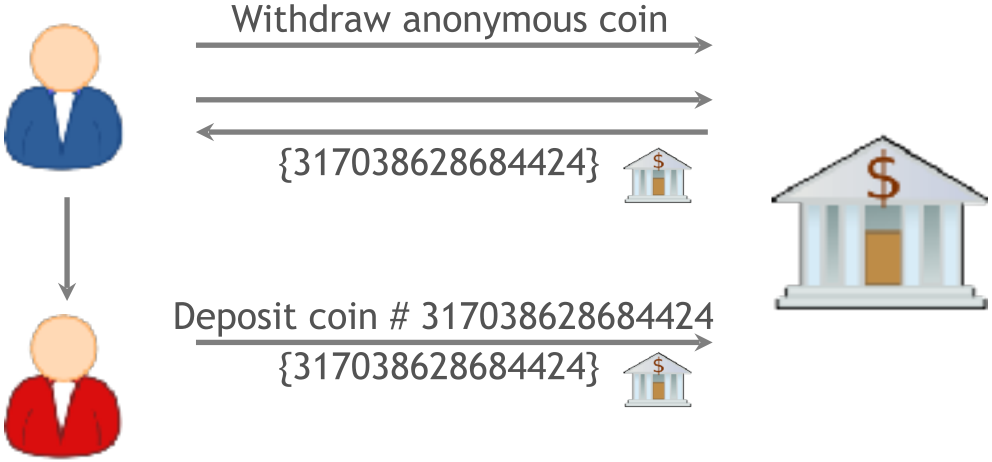
Deposit coin # 317038628684424

{317038628684424}

User	Balance
…	…
🟦	9
…	…
🟥	5

Spent coins
…

Anonymous e-cash via blind signatures



Withdraw anonymous coin

{317038628684424}

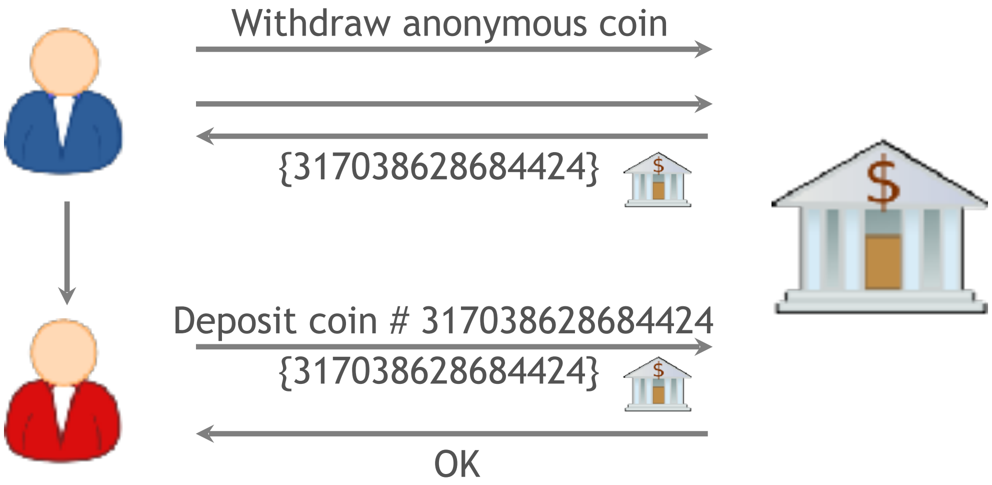
Deposit coin # 317038628684424

{317038628684424}

User	Balance
…	…
👤	9
…	…
👤	6

Spent coins
…
31703862…

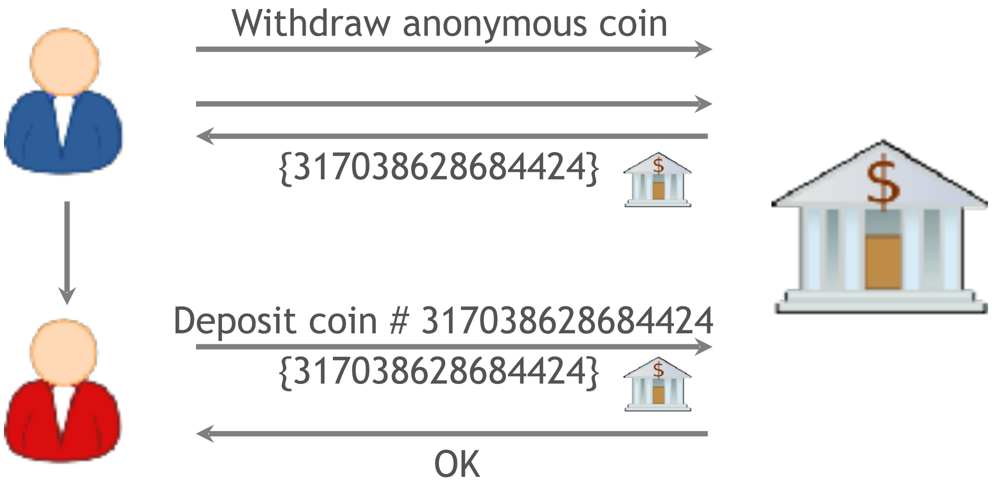
Anonymous e-cash via blind signatures



User	Balance
…	…
👤	9
…	…
👤	6

Spent coins
…
31703862…

Anonymous e-cash via blind signatures



User	Balance
…	…
🔵	9
…	…
🔴	6

Spent coins
…
31703862…

Bank cannot link the two users

Anonymity & decentralization: in conflict

- Interactive cryptographic protocols with bank are hard to decentralize
  - Later: Zerocoin and Zerocash overcome this challenge by using non-interactive cryptographic techniques
- Decentralization often achieved via public traceability to enforce security

How to de-anonymize Bitcoin

Trivial to create new addresses in Bitcoin

Best practice: always receive at fresh address

So, unlinkable?

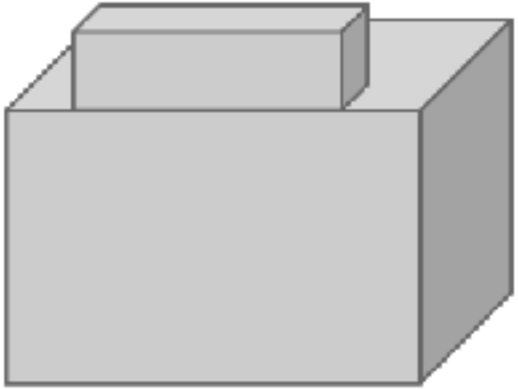
Alice buys a teapot at Big box store

5

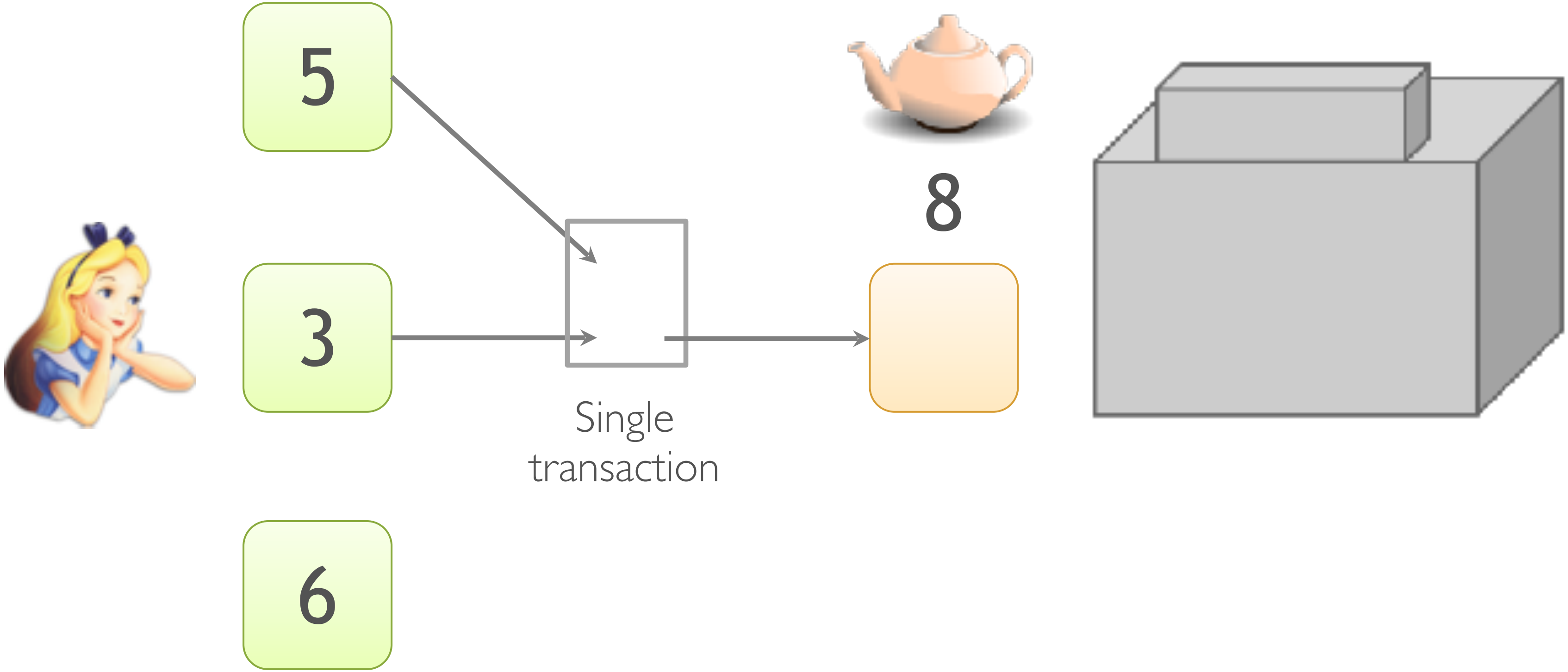
3

6

8

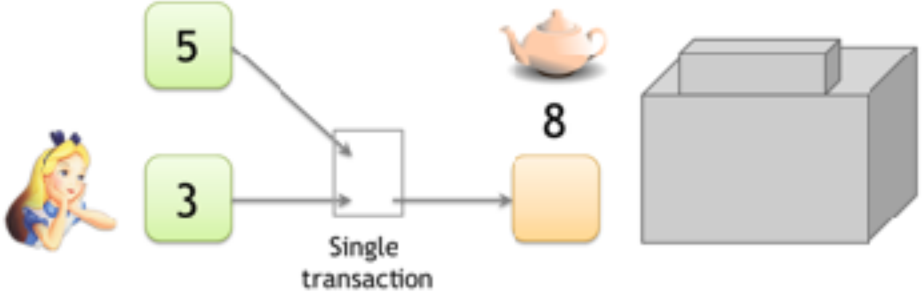


Alice buys a teapot at Big box store



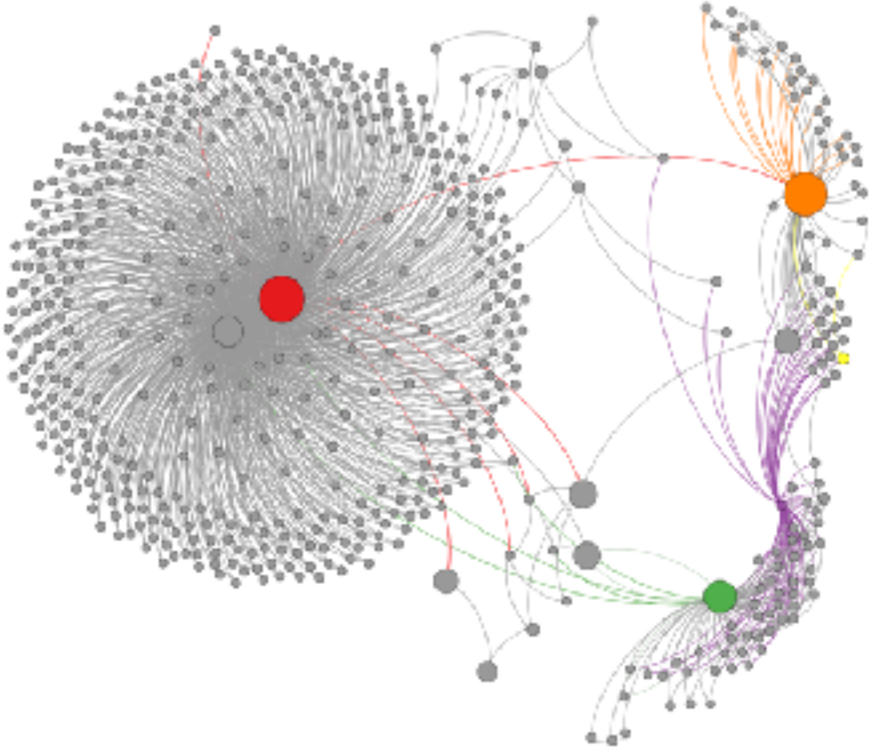
Linking addresses

Shared spending is evidence of joint cont



Addresses can be linked transitively

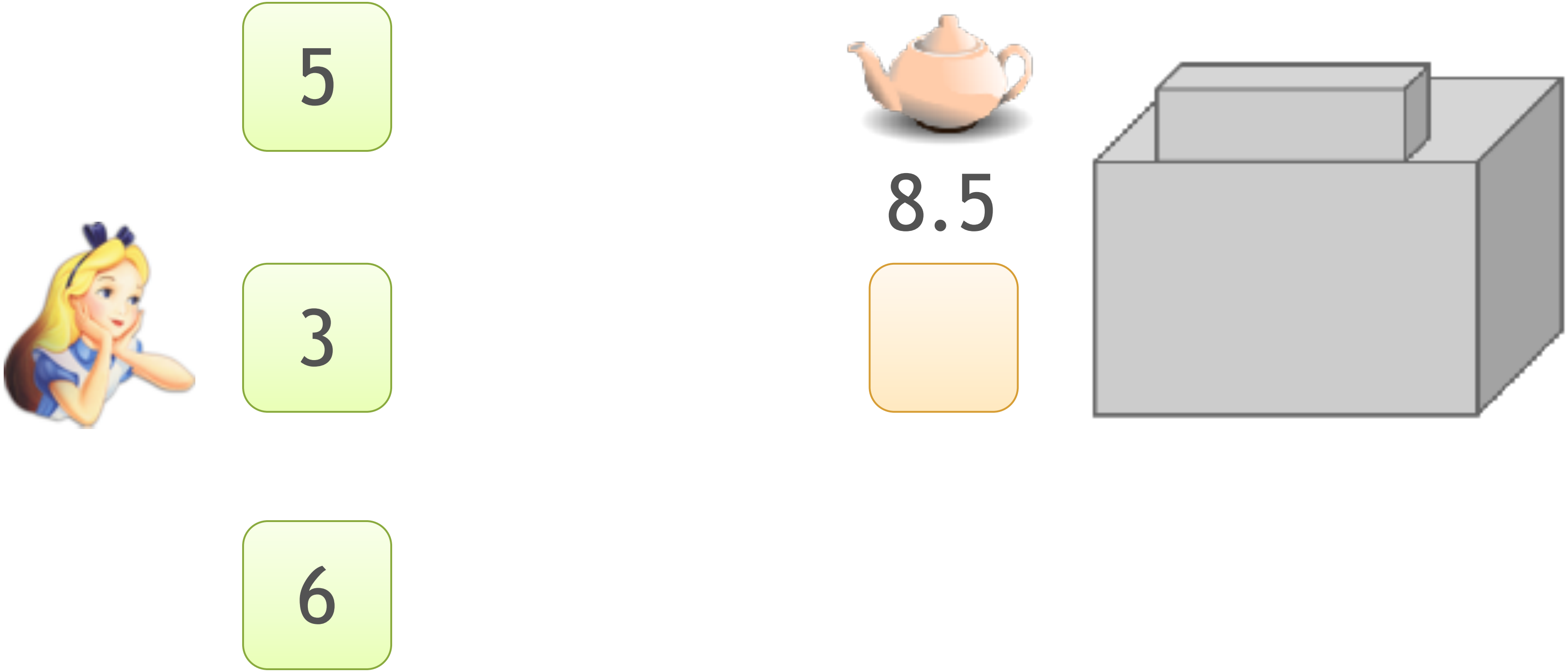
Clustering of addresses



An Analysis of Anonymity in the Bitcoin System

F. Reid and M. Harrigan
PASSAT 2011

Change addresses



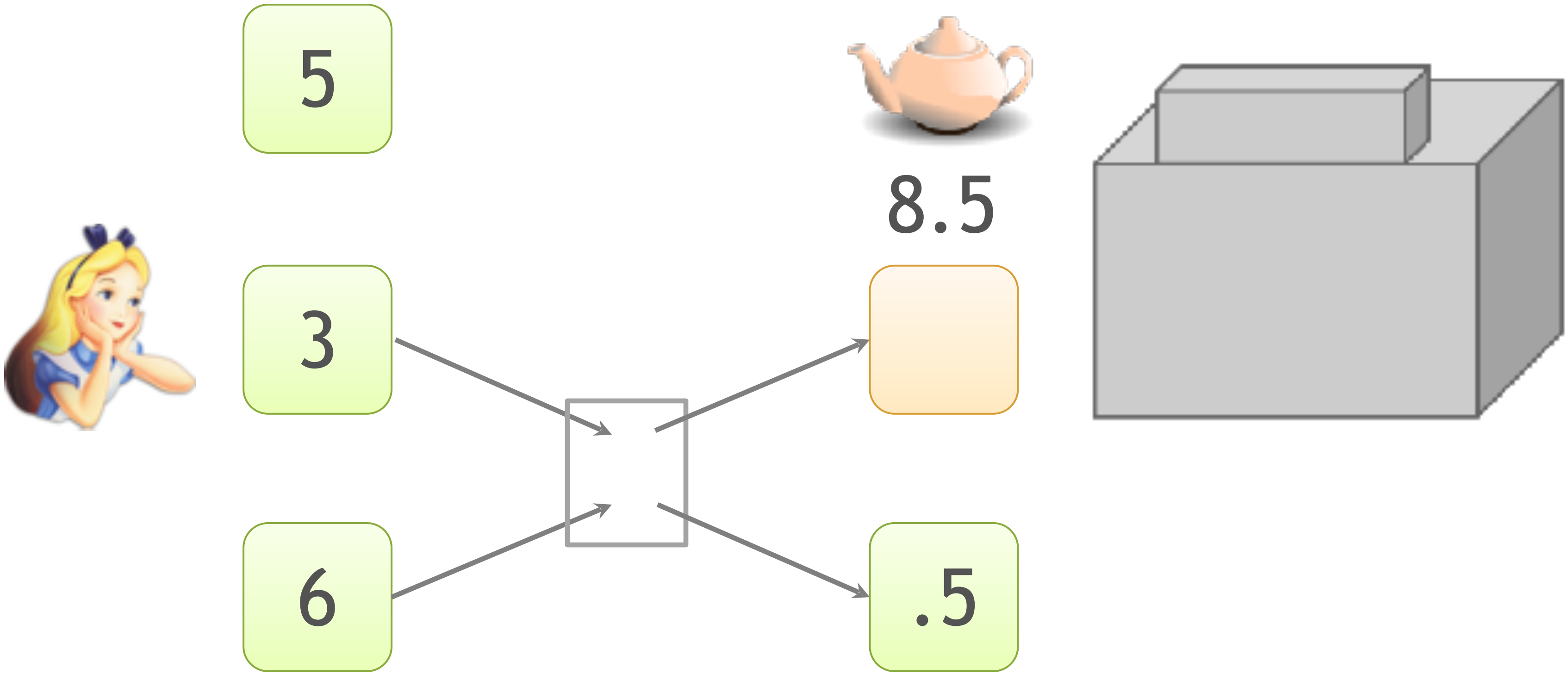
5

3

6

8.5

Change addresses



5

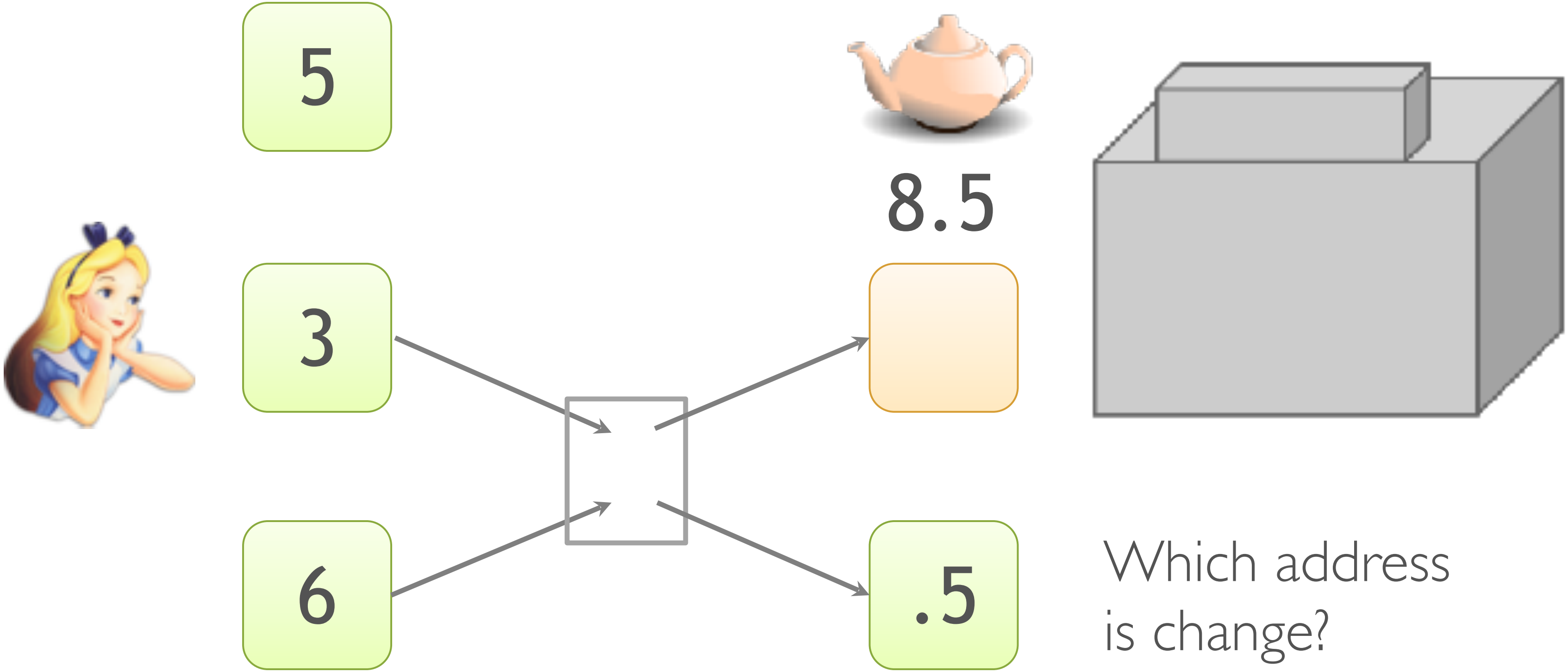
3

6

8.5

.5

Change addresses



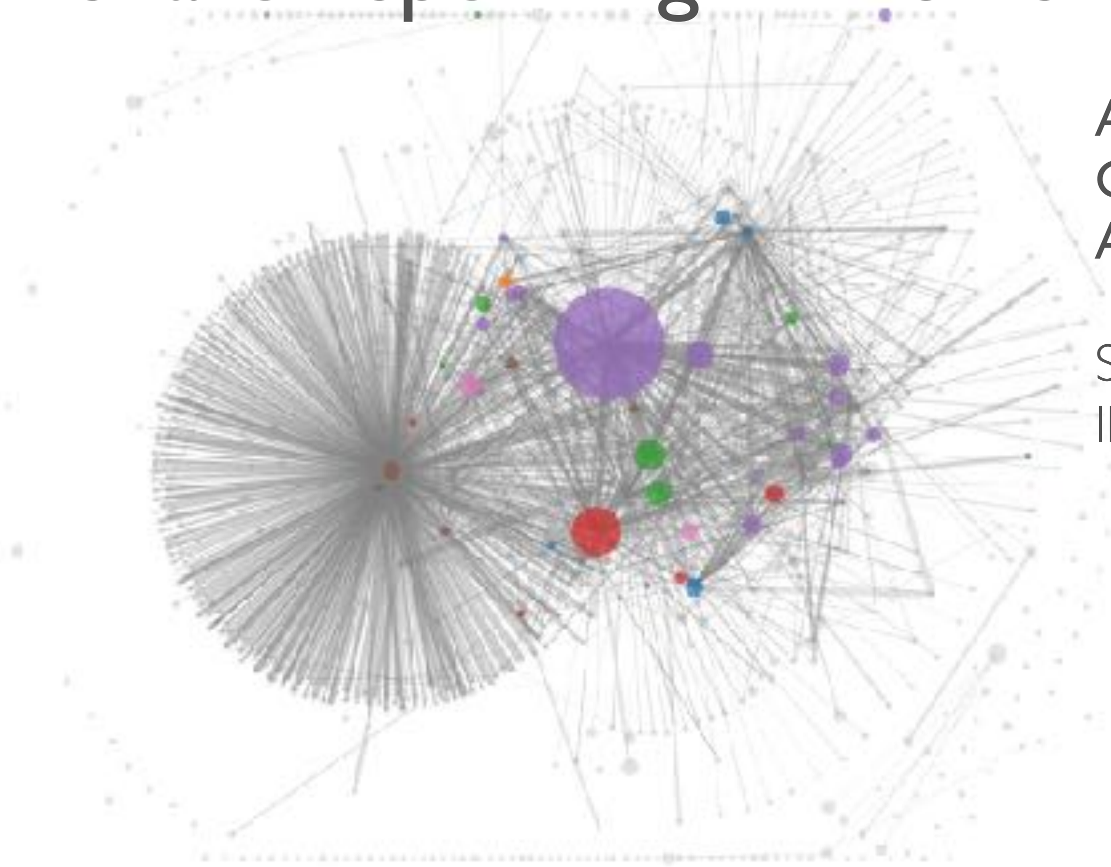
Which address is change?

“Idioms of use”

Idiosyncratic features of wallet software

e.g., each address used only once as change

Shared spending + idioms of use



A Fistful of Bitcoins: Characterizing Payments Among Men with No Names

S. Meiklejohn et al.
IMC 2013

To tag service providers: transact!

A Fistful of Bitcoins: Characterizing Payments Among Men with No Names

S. Meiklejohn et al.

344 transactions

- Mining pools
- Wallet services
- Exchanges
- Vendors
- Gambling sites

From services to users

1. High centralization in service providers

   Most flows pass through one of these — in a traceable way

2. Address — identity links in forums

Achieving Anonymity

Approaches

- **Mixing:** Pool in multiple transactions (ideally same value), and then create new transactions
  - Centralized: E.g., online wallets
  - Decentralized: E.g., CoinJoin
  - Untrusted intermediary using crypto: Tumblebit
- **New cryptocurrencies:**
  - Using Zero-knowledge proofs: Zerocoin and Zerocash
  - Using Ring signatures: Monero

Early solutions

- **Mixes**
  - Create a centralized server, many people send coins
  - Mixer shuffles those together, sends the right amount back to each user (less a fee), thus unlinking the sources of transactions
    - Risk 1: Mixer can "exit" and steal your cash
    - Risk 2: Mixer keeps track of the sources/destinations
    - Risk 3: Low volume of mixing can make tracing easy

Early solutions

- **CoinJoin**
  - Proposed by Maxwell; variants even earlier by “killerstorm” on BitcoinTalk*
    - Solves the “scamming mixer” problem
  - Idea: each transaction has multiple inputs and outputs
    - Have a mixer author <u>one single transaction</u> that consumes
      N equal-value inputs, produces N outputs

CryptoNote & RingCT

- 2012: CryptoNote (“Nicolas van Saberhagen”)
    - Originally launched as part of the ByteCoin currency
    - Anonymous creator, did a pre-mine
    - Was forked multiple times into many different currencies, including bitmonero -> Monero
    - Protocol ideas later improved into RingCT, which hid amounts as well as inputs (used in Monero today)

CryptoNote idea

- I want to make a transaction with (e.g.,) one input
  - But I don’t want to reveal which transaction is my input
  - Standard Bitcoin transactions do reveal this, and it leads to privacy problems
  - I could mix with other people (e.g., CoinJoin) but they would have to participate with me online, and that’s annoying

Ingredient: **Ring Signatures**

- Normal signature: sign with *sk*, verify with *pk*
- **Ring signature**:
  - Sign with my secret key + N-1 other people's public keys (Signer does not have to know the other secret keys!)
  - Verifier verifies with all *N* public keys (she must know them)
  - **Privacy**: verifier does not learn which signer actually made the signature! (It could be any of the key owners!)

CryptoNote idea

- Make all transactions the same value (e.g., 1 ByteCoin)

- Make all addresses single-use (auto-generated)

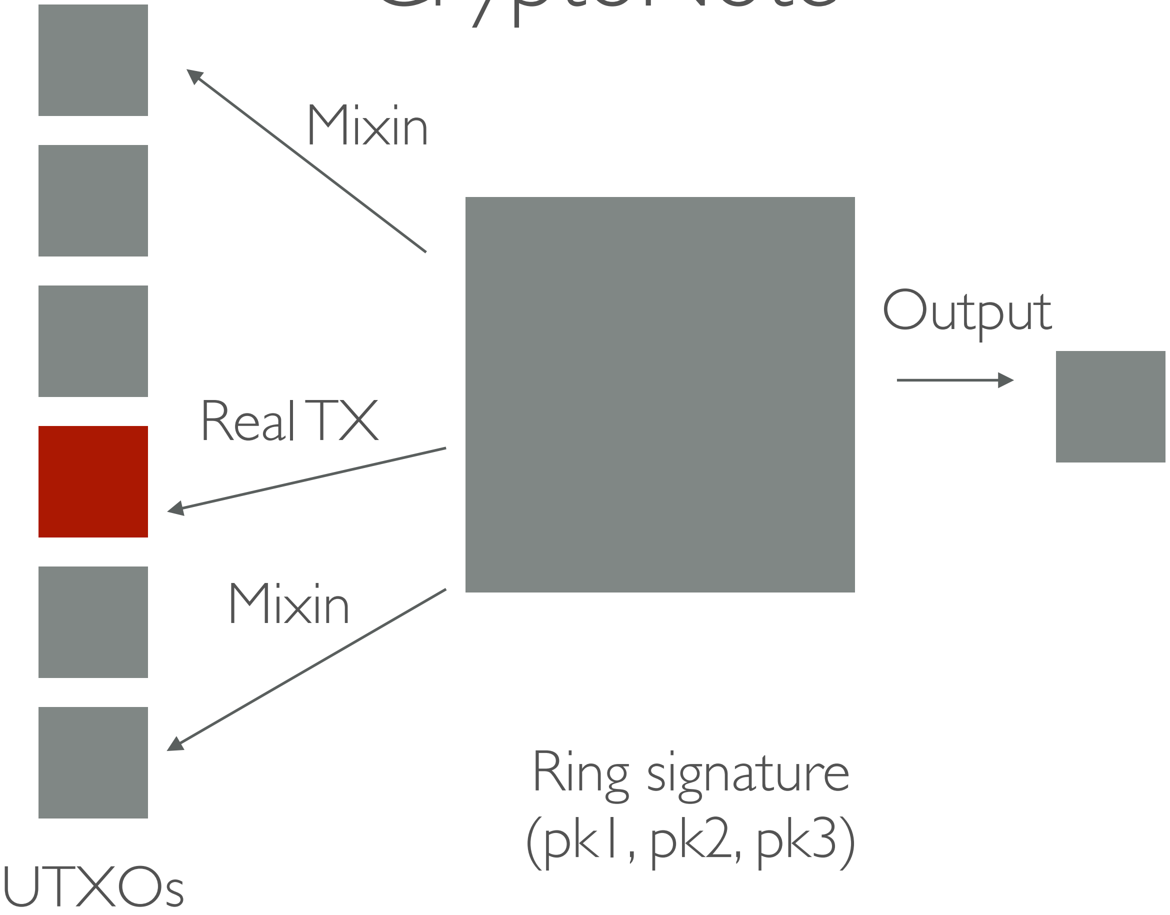
- Assume (for simplicity) that spender has one "real" input

1. Identify N-1 unrelated "cover" transactions from the UTXO set, get those public keys ("mixins")

2. Make a ring signature on her transaction, using her secret key and the N-1 public keys for the mixing

3. Post signature plus a "key image" (function of the real secret key) to prevent the real transaction being spent twice

CryptoNote



Ring signature
(pk1, pk2, pk3)

CryptoNote Limitations

- CryptoNote ring signatures grow as O(N) where N is number of inputs (including Mixins)
  - Ditto signing time and verification time
  - In practice this limits Mixin number to something modestly small (1-7)
  - Original CryptoNote required all input transactions be the same value, requiring multiple "denominations" (RingCT fixes this)
  - Surprisingly advanced crypto, surprisingly advanced code

Confidential Transactions [Maxwell]

Confidential Transactions [Maxwell]

- “Hide” transaction value using commitments

Confidential Transactions [Maxwell]

- "Hide" transaction value using commitments
  - Think: Why would this be beneficial?

Confidential Transactions [Maxwell]

- "Hide" transaction value using commitments
  - Think: Why would this be beneficial?
- What if we want to support multiple inputs and outputs?

Confidential Transactions [Maxwell]

- "Hide" transaction value using commitments
  - Think: Why would this be beneficial?
- What if we want to support multiple inputs and outputs?
  - Need to establish that "total" input >= "total" output.

Confidential Transactions [Maxwell]

- "Hide" transaction value using commitments
  - <u>Think</u>: Why would this be beneficial?
- What if we want to support multiple inputs and outputs?
  - Need to establish that "total" input >= "total" output.
- **<u>Main Challenge:</u>** How to verify that a transaction is valid when the values are hidden?

Confidential Transactions [Maxwell]

- Two Ideas:

Confidential Transactions [Maxwell]

- Two Ideas:
  - **(Additively) Homomorphic commitments**: There is an operation that can be performed on commitments that will result in addition of underlying values

Confidential Transactions [Maxwell]

- Two Ideas:
  - **(Additively) Homomorphic commitments**: There is an operation that can be performed on commitments that will result in addition of underlying values
    - Now, need to establish that (Sum of inputs) - (Sum of outputs) is non-negative.

Confidential Transactions [Maxwell]

- Two Ideas:
  - **(Additively) Homomorphic commitments**: There is an operation that can be performed on commitments that will result in addition of underlying values
    - Now, need to establish that (Sum of inputs) - (Sum of outputs) is non-negative.
  - **Zero-Knowledge Proofs**: Prove something about committed values **without revealing the values**!

Recall: Commitments

- Like a digital "envelope": allows you to commit to a message value, without revealing what it is
  - C = Commit(message; randomness)
  - **Hiding**: given a commitment, can't see what message it is, until I "open" the commitment and reveal it to you
  - **Binding**: giving you a commitment "binds" me to a specific message/value. I can't change my mind when I open it.

Recall: Hash commitments

- **Commit Procedure**:
  - Pick some random “salt” (e.g., 256 bits) **r**
  - Compute C = Hash(message || r)
- **Open Procedure**: Reveal (message, r), verifier checks hash
- **Additive Homomorphism**: Not known for general hash functions :-(

Pedersen Commitments

- Let $G = \langle g \rangle$ be a "cyclic" group where it is hard to find *x* given (g, g^x) — AKA the **discrete log problem** (DLP) is hard
  - E.g., *G* can be a subgroup of a finite field $\{1, \ldots, p-1\}$ where exponentiation/multiplication are modulo *p*
  - We also need two public **generators**: g, h *such that nobody knows the discrete log of h w.r.t. g*
- Commitment to message *m*: Pick random $r \in \{0, \ldots, groupOrder - 1\}$, compute: $C = g^m \cdot h^r$
- To open the commitment, simply reveal (m, r)

Pedersen Commitments

- Why is this secure?

    - **Hiding:** If g, h are generators, then h^r is a random element of the group, so. $C = g^m \cdot h^r$ is too

Pedersen Commitments

- Why is this secure?
  - **Hiding:** If g, h are generators, then h^r is a random element of the group, so $C = g^m \cdot h^r$ is too
  - **Binding:** Let q be the group order. Let $h = g^x$ for some unknown *x*. Assume an attacker can find (m, r) != (m', r') such that .Then it holds that: $g^m h^r = g^{m'} h^{r'}$

$$g^m g^{xr} = g^{m'} g^{xr'}$$

and thus,

$$m + xr = m' + xr' \ mod \ q$$

We can solve for *x*, which means solving DLP, which is contradiction!

Pederson Commitments

- Pedersen commitments are <u>additively homomorphic:</u>
  - Commit to "m1": $C_1 = g^{m_1} h^{r_1}$
    Commit to "m2": $C_2 = g^{m_2} h^{r_2}$
  - Now multiply the two commitments together:

$$\begin{aligned} C_3 &= C_1 \cdot C_2 \\ &= g^{m_1} h^{r_1} \cdot g^{m_2} h^{r_2} \\ &= g^{m_1+m_2} h^{r_1+r_2} \end{aligned}$$

Notice that C3 is a commitment to the <u>sum</u> m1+m2 (under randomness r1+r2)

Zero-Knowledge (ZK) Proofs

Zero-Knowledge (ZK) Proofs

- Invented by Goldwasser, Micali, Rackoff in 1980s

Zero-Knowledge (ZK) Proofs

- Invented by Goldwasser, Micali, Rackoff in 1980s
  - Prove a statement without revealing any other information

Zero-Knowledge (ZK) Proofs

- Invented by Goldwasser, Micali, Rackoff in 1980s
  - Prove a statement <u>without revealing any other information</u>
  - What does this mean?

Zero-Knowledge (ZK) Proofs

- Invented by Goldwasser, Micali, Rackoff in 1980s
    - Prove a statement <u>without revealing any other information</u>
    - What does this mean?
- Powerful Theorem by Goldreich-Micali-Wigderson from 1980s:
  Anything in NP can be proven in zero-knowledge

Zero-Knowledge (ZK) Proofs

- Invented by Goldwasser, Micali, Rackoff in 1980s
  - Prove a statement without revealing any other information
  - What does this mean?
- Powerful Theorem by Goldreich-Micali-Wigderson from 1980s:
  Anything in NP can be proven in zero-knowledge
- What is NP?

Zero-Knowledge (ZK) Proofs

- Invented by Goldwasser, Micali, Rackoff in 1980s
  - Prove a statement <u>without revealing any other information</u>
  - What does this mean?
- Powerful Theorem by Goldreich-Micali-Wigderson from 1980s:

  Anything in NP can be proven in zero-knowledge
- What is NP?
  - Class of languages where membership can be efficiently verified using a "witness" (a.k.a certificate of validity)

Zero-Knowledge (ZK) Proofs

- Invented by Goldwasser, Micali, Rackoff in 1980s
  - Prove a statement <u>without revealing any other information</u>
  - What does this mean?
- Powerful Theorem by Goldreich-Micali-Wigderson from 1980s:
  Anything in NP can be proven in zero-knowledge
- What is NP?
  - Class of languages where membership can be efficiently verified using a "witness" (a.k.a certificate of validity)

RingCT Extension to CryptoNote

- Uses these tools to achieve variable-value, hidden transactions
- Builds on ideas from Maxwell's Confidential Transactions
- Proofs of transaction validity used in RingCT are special-purpose, not general-purpose (we will later discuss how using general-purpose proofs can simplify design)