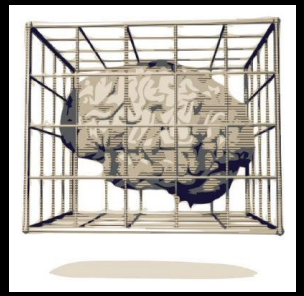
particl

A Private and Decentralized Marketplace

Kewde



kewde@particl.io
PGP: 09CF4376

- *Anonymous upon till now..*
- Bachelor's degree in business engineering (last year)
- 21 years old, but fascinated by computers at the age of 12

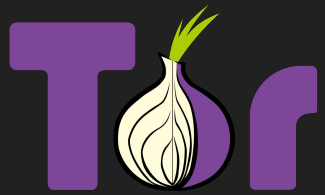
- **Interests**
  Programming, security, anonymity, applied cryptography, decentralized networks, cryptocurrencies, automation, electronics

- Dutch & French

particl
Research & Development

privacytools.io
Maintainer

Open Source software and funding



- Many Open Source projects are struggling to get by...
- Grants and user donations (e.g. Tor, Signal, ...)
- Traditional business model doesn’t apply here

Some privacy projects that were struggling to get by

GnuPG

- On the verge of abandonment
- Maintained by Werner Koch
- Raised $135K in grants and donations after a cry for help

CopperheadOS

`copperhead.co/android`

- Hardened Android OS
- Developed by Daniel Micay (strcat), James Donaldson (dnj)
- Open Source but not Free, adopted a more restrictive licence

There's one category that seems to have it somewhat figured out

Cryptocurrencies

- Economic incentive for developers, community members, ...
- Not relying on charity, grants etc.
- Interesting to see how they fund themselves

Privacy cryptocurrencies

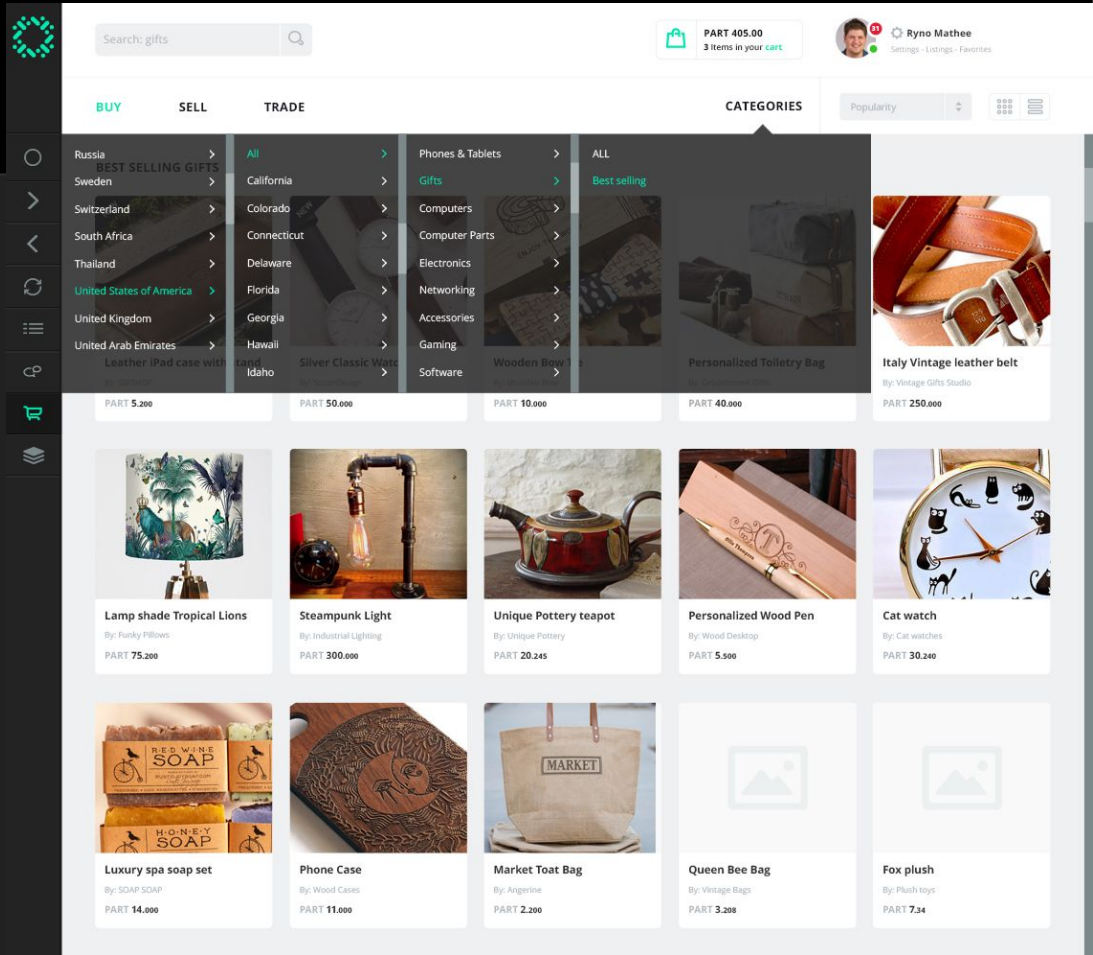
- Works for financial privacy
- Hopefully we'll see more privacy projects sustained by the cryptocurrency model

ZCASH

particl

“A platform that integrates a variety of tools to take back privacy.”

- **Cryptocurrency**: Private transactions
- **SMSG**: end-to-end encrypted messaging
- **Private Marketplace** *(in progress)*



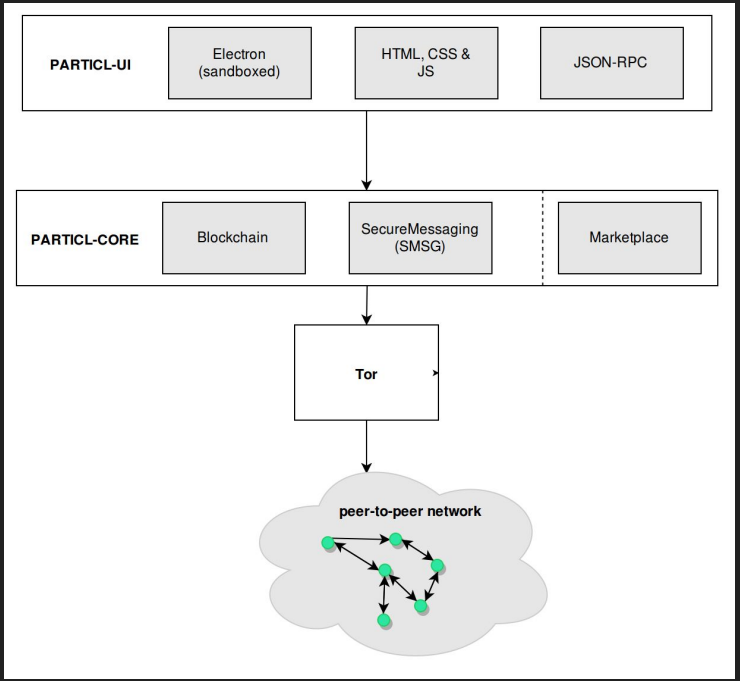
Particl — A Private and Decentralized Marketplace

particl

Why?

- Marketplaces these days require an absurd amount of personal information & trust
- Centralized marketplaces have a lot of power over sellers & buyers
- Businesses are generally more interested in privacy than the average person
- An open system allows competitors to easily identify your customers, best selling products, ...

Particl — A Private and Decentralized Marketplace



Currently done

- ✓ A private cryptocurrency (CT)
- ✓ SecureMessaging (SMSG)
  A decentralized network for messaging and storing the market listings

In progress

- ◻ Open Market Protocol (OMP)
- ◻ RingCT (testnet)
- ◻ Marketplace MVP

A recipe for a private marketplace

particl

Goals

- Minimize the information leakage to other nodes (searches, viewing listings, ...)
- Anonymize the communication between buyers and sellers (a message shouldn't reveal sender, receiver public keys or IP addresses)
- Unlinkability of transactions (purchases) and their respective market listings
- Obfuscate the exact origins of a transaction (RingCT)

A small overview

The presentation will follow the same workflow as a complete buyer & seller interaction

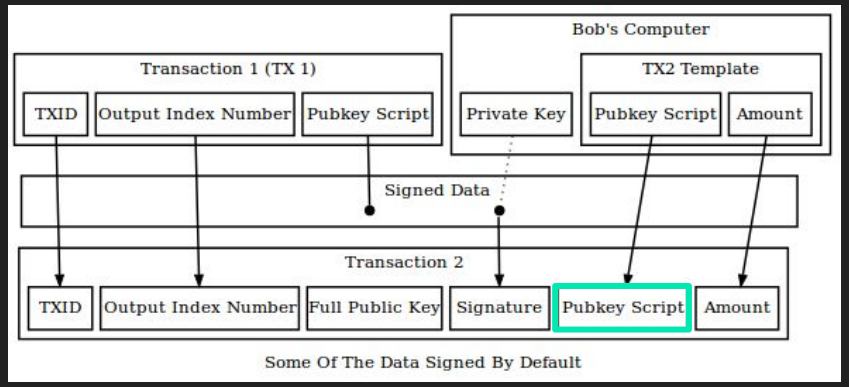
(From publishing the listing, down to the buyer paying for item)

- **Creating a market listing & broadcasting it**
- Open Market Protocol
- Private Transactions


```
{
  "version": "0.0.1.0",
  "item": {
    "information": {
      "title": "Title of the item",
      "short_description": "A short description / summary of item",
      "long_description": "A longer description of the item or service",
      "category": [ "Category", "Subcategory", "Subsubcategory" ]
    },
    "payment": {
      "type": "SALE",
      "escrow": {
        "type": "NOP"
      },
      "cryptocurrency": [
        {
```

1. Creating a market listing & broadcasting it

- A typical Bitcoin transaction



- Special Pubkey script for the output (next slide)
- Allows us to request a fee for registering the market listing

1. Creating a market listing & broadcasting it

- Creating an index of all market listings on the blockchain

OP_RETURN	VOP_REGLIST	item_pk	protocol_id	listing_hash	listing_id

- Creating an index of all the market listings on the blockchain
- **VOP_REGLIST**: “virtual” opcode that does not really exist in Bitcoin, indicates that market listing should be added
- **item_pk**: the input index from which the public key is retrieved (default: 0)
  - Item public key: allows for multiple listings (different listing_hashes) to link the same key, useful for per-item reputation
- **protocol_id**: specifies which protocol/network should be used to retrieve the content
  - Blockchain stores reference to the actual data to prevent bloat
- **listing_hash**: used to verify authenticity of the data returned (SHA256)
- **listing_id**: a unique identifier for retrieving the content (optional for some protocols)
  - e.g. `https://3g2upl4pq6kufc4m.onion/assets/logo_homepage.normal.v107.png`

1. Creating a market listing & broadcasting it

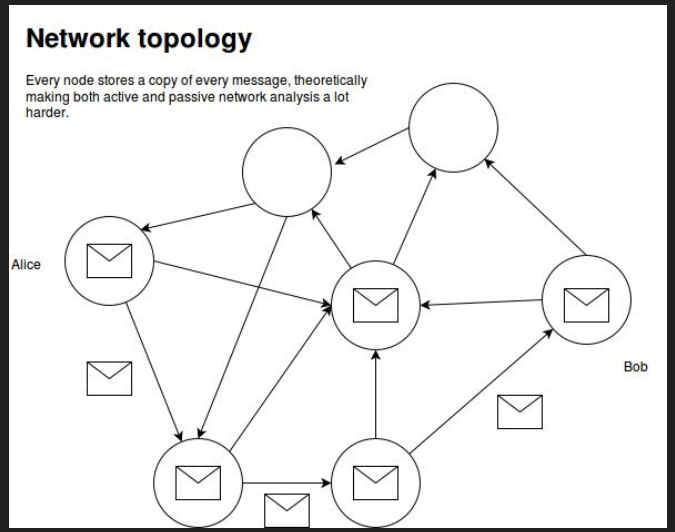
protocol_id	listing_hash	listing_id

- **Data Storage Network (DSN) reference**
- Abstraction layer: allows for different protocols and networks in the future
- Currently we only have a few protocol IDs
  - URL
  - SMSG
- Future: IPFS ZeroNet

SecureMessaging — SMSG

"A decentralized network where every node store all encrypted messages for 48 hours."

- A variant of BitMessage (Python, OpenSSL)
- Operates over the same networking stack as `particl-core` (C++, libsecp256k1)



SecureMessaging — SMSG

particl

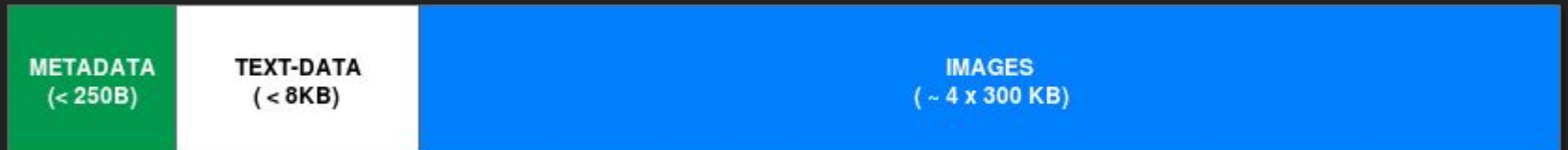
Benefits of local storage

- No information leakage
  - **DHT lookups:** other nodes know what content you're accessing
    Improved DHT lookup are described in academic literature, but not many implementations
  - **SMSG:** no other node knows what you're searching or accessing
- Low latency in the UI

Drawbacks of local storage

- Doesn't scale, especially images take up a big load

IPFS



SecureMessaging — SMSG

Future improvements

- More anonymous message broadcasting (Dandelion)
  - Currently just flooding, use with Tor for now
- Perfect forward secrecy (PFS)

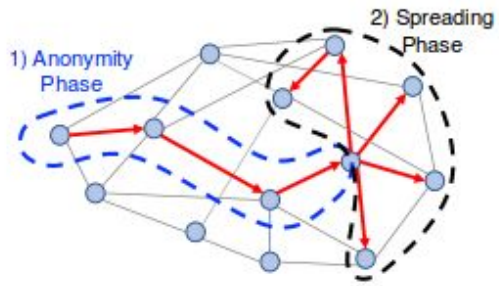


Figure 4: Dandelion spreading forwards a message in a line over the graph, then broadcasts it using diffusion. Here both phases occur over the same graph, i.e., $H = G$.

A small overview

particl

Creating a market listing & broadcasting it

Open Market Protocol

Private Transactions


```
{
  "version": "0.0.1.0",
  "item": {
    "information": {
      "title": "Title of the item",
      "short_description": "A short description / summary of item",
      "long_description": "A longer description of the item or service",
      "category": [ "Category", "Subcategory", "Subsubcategory" ]
    },
    "payment": {
      "type": "SALE",
      "escrow": {
        "type": "NOP"
      },
      "cryptocurrency": [
        {
          "currency": "BITCOIN",
          "base_price": 100000000
        }
      ]
```

2. Open Market Protocol (OMP)

particl

"An open protocol for marketplaces:
standardize the interactions between buyers and sellers into a single format."

- **Public listing format**
  Contains all the data about an item/service (description, images, ...)
- **Private message format (WIP)**
  Communication between buyers and sellers (address, transaction, ...)

`https://kewde.gitbooks.io/protocol/`

2. Open Market Protocol (OMP)

particl

Why?

- No other specifications, everybody just *"does their thing"*
- Give back power to the sellers: portability of item data

amazon → ebay

- Designed to work in a decentralized network
- Designed for usage with cryptocurrencies
  - Protocol allows for any cryptocurrency to be used

bitcoin

2. Open Market Protocol (OMP)

particl

Private Message Format

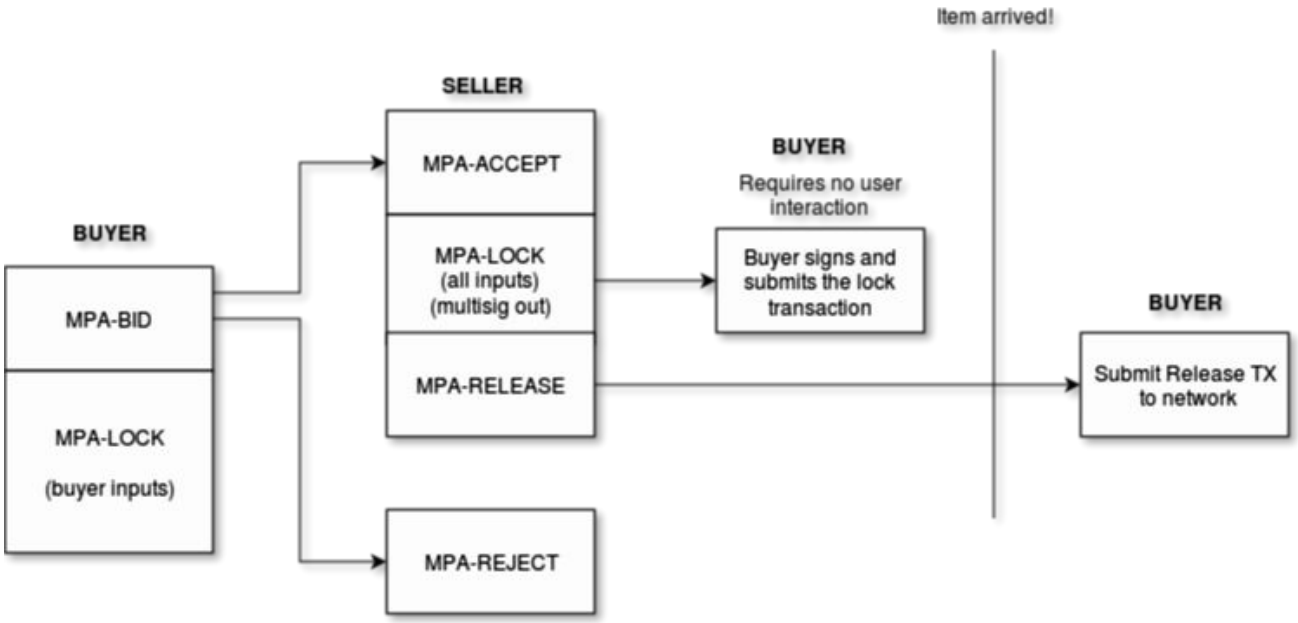
- The buyer found an item he wishes to purchase, he contacts the seller using this format
- The message should be encrypted (contains sensitive information)
  - Address of the buyer
  - Raw transactions
- Transactions are sent over as raw transactions
  - Robust but needs decoding
  - Sanity checks!

2. Open Market Protocol (OMP)

particl

Private Message Format

GENERAL WORKFLOW



A small overview

particl

- Creating a market listing & broadcasting it
- Open Market Protocol
- **Private Transactions**


```
{
  "version": "0.0.1.0",
  "item": {
    "information": {
      "title": "Title of the item",
      "short_description": "A short description / summary of item",
      "long_description": "A longer description of the item or service",
      "category": [ "Category", "Subcategory", "Subsubcategory" ]
    },
    "payment": {
      "type": "SALE",
      "escrow": {
        "type": "NOP"
      },
      "cryptocurrency": [
        {
          "currency": "BITCOIN",
          "base_price": 100000000
        }
      ]
```


3. Confidential Transactions — background

"Blinds the amount being transacted from a passive observer."

- Invented by Gregory Maxwell
- Built on `libsecp256k1` with additional modules (by The Elements Project)

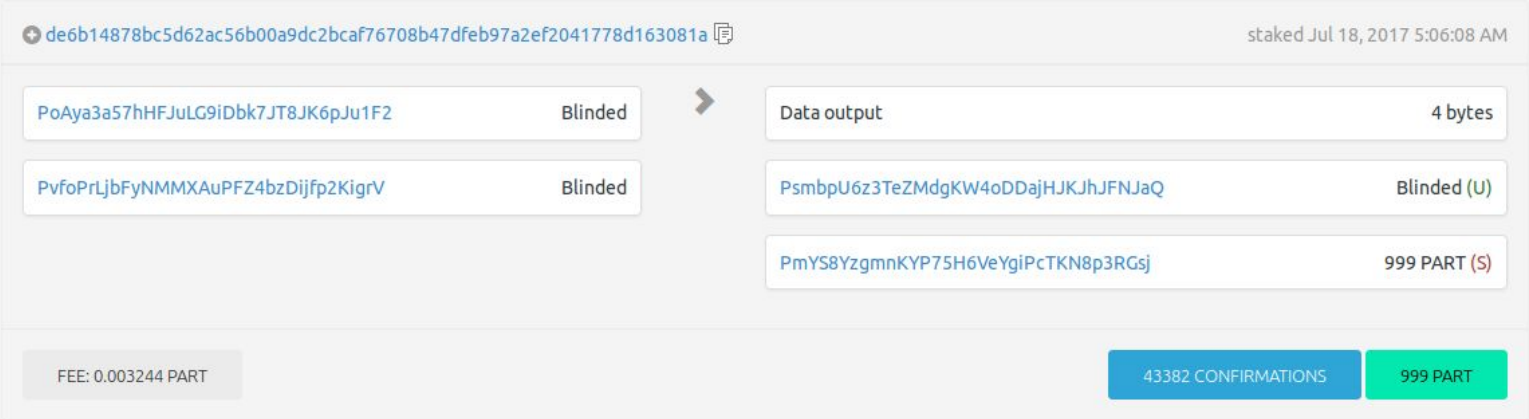
- Tecnovert implemented in on Bitcoin Core 0.15
- Allows for hidden amounts in multisignature addresses
- Required to prevent amount linkability, a flaw in marketplaces using Bitcoin

Disconnecting market listings and transactions

- Protect against passive observers (blockchain analysis firms)
- Unlinkability between transactions (purchases) and the items/services
- Fatal flaw in marketplaces using Bitcoin: amount linkability
  - A coffee machine selling for 0.00444036 BTC → can be linked to the purchase transaction
  - Amount is a potentially unique identifier

Disconnecting market listings and transactions

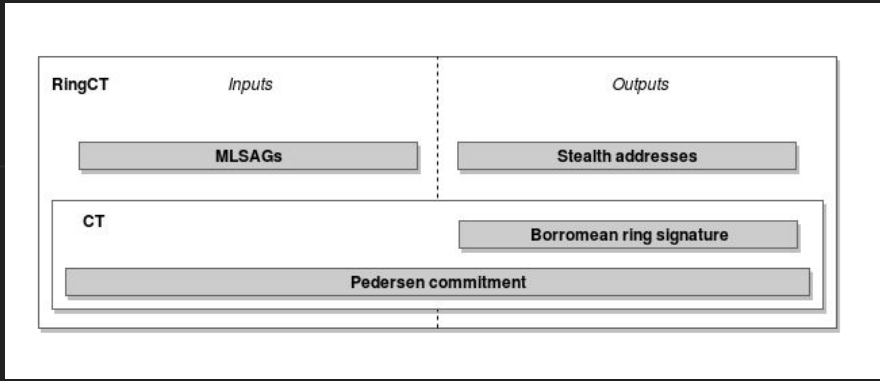
- Confidential Transactions: blinds the amounts being transferred
- Improves unlinkability between transactions (purchases) and the actual item/service



RingCT — background

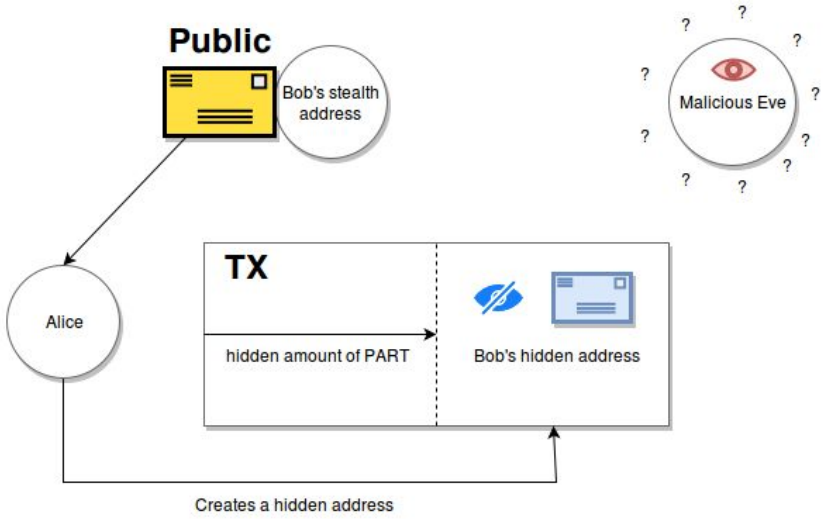
"Obfuscates the origin and receiver of a transaction, also hides the amount being transacted from a passive observer."

- Invented by Shen Noether (Monero)
- Builds on inventions of Bitcoin Core developers
  - Stealth Addresses (Peter Todd)
  - Efficient LSAGs (Adam Back)
  - Confidential Transactions (Gregory Maxwell)



- Tecnovert ported RingCT to Bitcoin Core 0.15
- Currently active on testnet
- Next few slides are a basic introduction

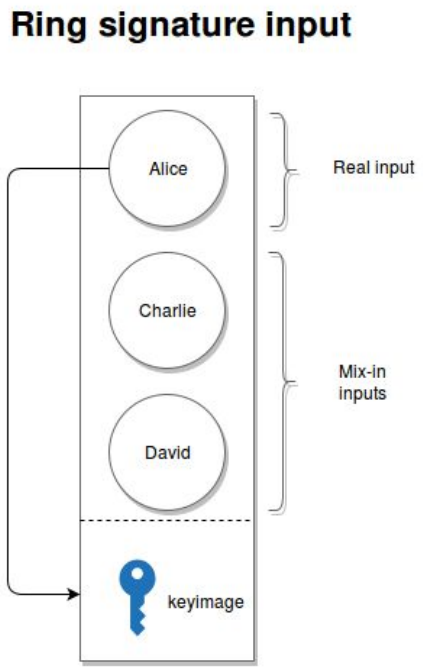
RingCT — Understanding Stealth Addresses



- **Alice** can use **Bob's** Stealth Address and create a hidden address
- **Malicious Eve** can not link the hidden address to the corresponding stealth address
- **Bob** can re-create the hidden address using metadata in the TX
- Only **Bob** can spend the coins
- Metadata in `OP_RETURN`: one-time public key

RingCT — Understanding unique ring signatures

- Special type of input that does not reveal the real spender
  - Mix-in inputs: a set of decoy inputs
    One of the inputs is the real spender, but we don't know which one
- Double spend prevention through KeyImages (KI)
  - Record all KI of all transactions
  - Reject TX with duplicate KI
- Every input must only generate one valid KI



particl

Thanks!

Any questions?

https://particl.io