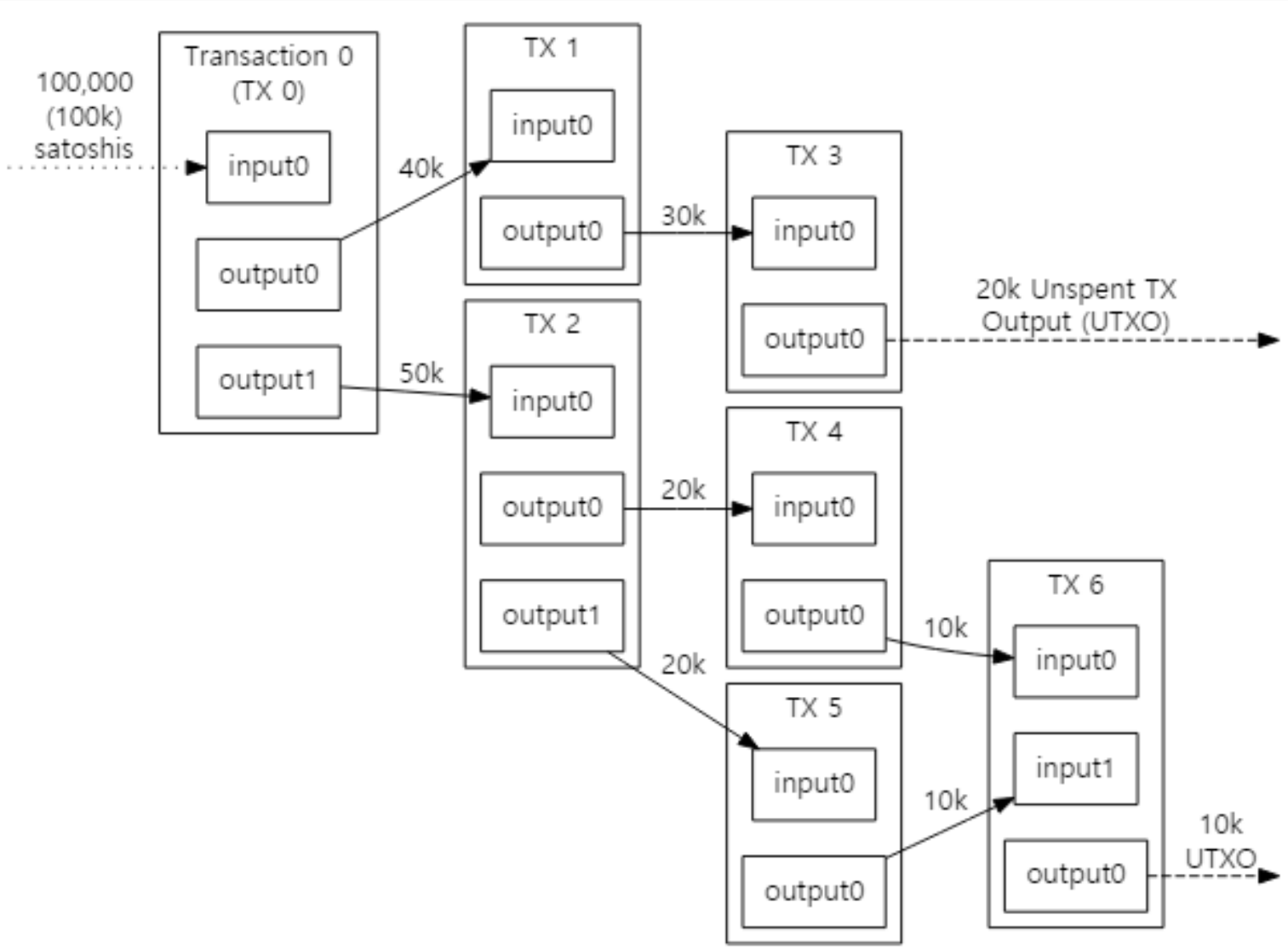
Privacy in blockchains and cryptographic techniques

Marc Beunardeau – Nomadic Labs

October 8, 2019

Privacy issues with Bitcoin

UTXO model



Triple-Entry Bookkeeping (Transaction-To-Transaction Payments) As Used By Bitcoin

Privacy properties of Bitcoin

- Unintuitive privacy model, thought to be safe
- Unknown user, fresh addresses
- While graph of transaction in known (and timestamped)
- Any information can be propagated
- Everything breaks if users are identified with a set of addresses

User profile indistinguishability

- Let $Sim, (X, Y)$ be a similarity function ranging in $[0, 1]$
- We compute $Sim(Estimated, Truth) - Sim(Random, Truth)$

Shadows addresses heuristic

- Transaction with two outputs
- One the output existed before
- $\Rightarrow$ The second one is the change and belong to the same user

Multi-input addresses heuristic

- $\Rightarrow$ All addresses participating in a multi-input transaction belong to the same user

Applying the two heuristics

- Estimated 60 000 users
- 1 600 000 unique addresses
- 1 000 000 users with multi-input heuristic
- 7 000 000 users with shadow addresses heuristic
- 58 % addresses grouped with an average of 11.5 addresses by identified users

Machine learning to extrapolate

- Using the first grouping, try clustering to get 60 000 users
- The first grouping is sent to a Hierarchical Agglomerative Clustering which send his output to a K-means
- We assume users do not do two transactions at the same time

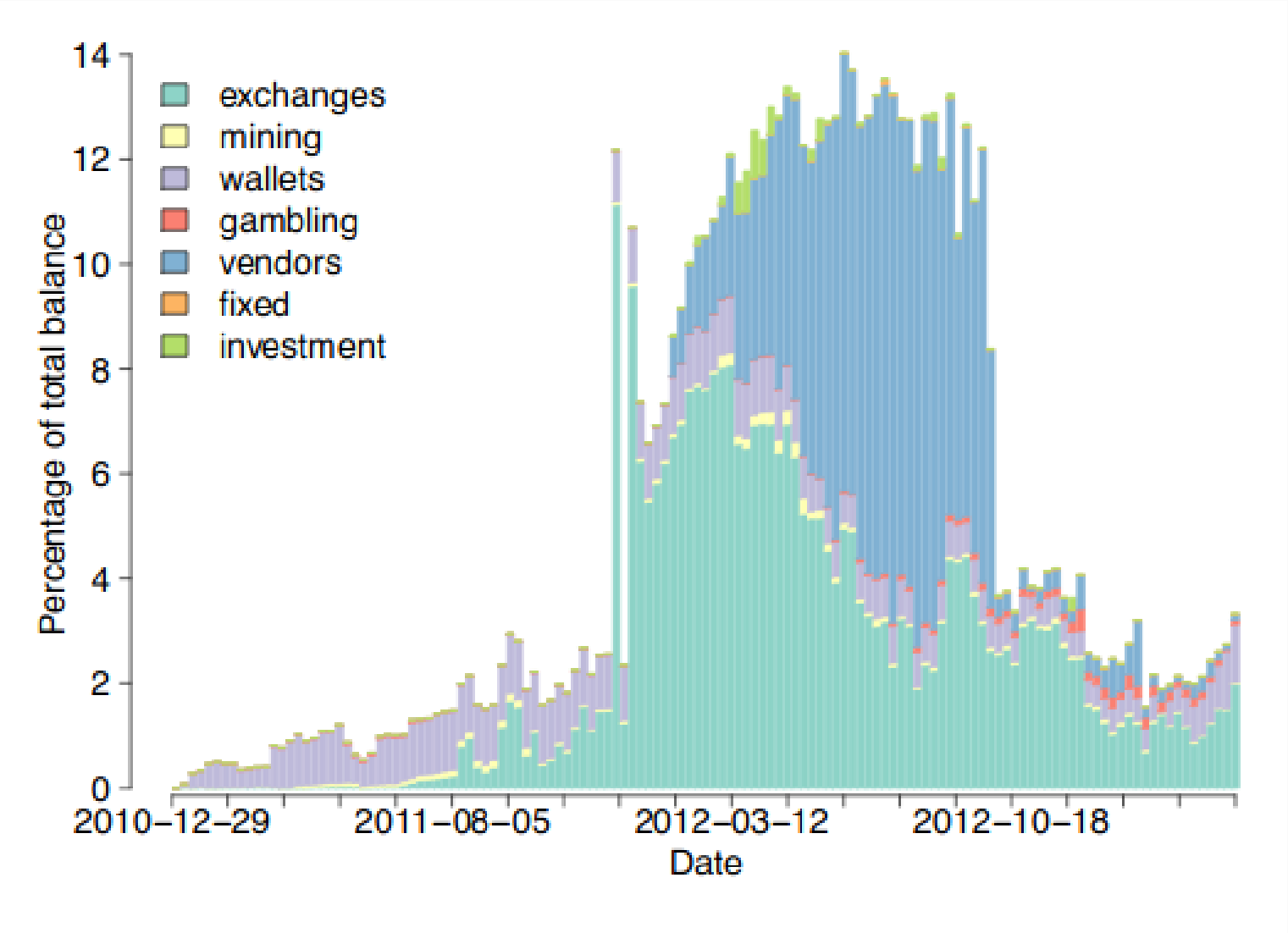
Simulated Results in a university

- Create random simulated users with different profiles (Prof, student, shops etc...)
- Train the model to match the real world
- Create the attacker
- $Sim(Estimated, Truth) - Sim(Random, Truth)$ range from 0.6 to 0.8

State of the Bitcoin

Larger or equal to	Smaller than	Number of entities
1	2	2,214,186
2	10	234,015
10	100	12,026
100	500	499
500	1,000	35
1,000	5,000	41
5,000	10,000	5
10,000	50,000	5
50,000	100,000	1
100,000		1

Adding data


exchanges
mining
wallets
gambling
vendors
fixed
investment
Percentage of total balance
0
2
4
6
8
10
12
14
2010–12–29
2011–08–05
2012–03–12
2012–10–18
Date

Monero

Reminder: Commitment scheme

A commitment scheme is composed of two algorithms $Commit(x, r) = c$ and $Reveal(c, x, r) = boolean$ satisfying two properties :

- Hiding : From c no reasonable algorithm can find x (nor any info about x)
- Binding : Given $c = Commit(x, \cdot)$ no reasonable algorithm can find x', r'such that $Reval(c, x', r') = T$

Homomorphic commitment

Same as commitment but there exists public operation $\circ$ such that

$$Commit(x, r) \circ Commit(x', r') = Commit(x + x', r + r')$$

Zero-knowledge Range proofs

Let $c = Commit(x, r)$
A range proofs has two algorithms $Prove(c, r, a, b) = \pi$,
$Verify(c, a, b, \pi) = boolean$

- Completeness: $Verify(c, a, b, Prove(c, r, a, b)) = a \leq x \leq b$
- Soundness: There are no reasonable algorithm A such that $Verify(c, a, b, A(c, a, b, x, r)) = T$ and not $a \leq x \leq b$
- Zero-knowledge: π does not give any information on x (nor r)

A range proof can turned into an equality proof by setting $a = b$

Ring signature syntax

A ring signature scheme comprise of three algorithm:

- $KeyGen() = (pk, sk)$
- $Sign(m, sk_i, \{pk_1, \cdots, pk_n\}) = \sigma$
- $Verify(m, \{pk_1, \cdots, pk_n\}, \sigma) = boolean$

Ring signature correctness

- For all $(pk, sk) = KeyGen()$, m, $\{pk_1, \cdots pk_n\}$
- with $\sigma = Sign(m, sk_i, \{pk_1, \cdots, pk_n\})$ and $pk_i \in \{pk_1, \cdots, pk_n\}$
- Then $Verify(m, pk_1, \cdots, pk_n, \sigma) = T$

Ring signature security

- For all reasonable algorithm that can asks for some $\{pk_1, \cdots pk_n\}$
  and some $\sigma_i = Sign(m_i, sk_i, E)$ where $E \subset \{pk_1, \cdots, pk_n\}$
- It is impossible to produce σ, $m \neq m_i$, $F \subset \{pk_1, \cdots, pk_n\}$ st. $Verify(m, F, \sigma) = T$

Monero's protocol

- Similar to Bitcoin (forked)
- Hide the amounts in homomorphic commitments
- Pick a few random public keys an ring sign transaction with a set
- Input a commitment of 0 for the others inputs
- input a commitment of v for yours
- Output a commitment of v for the receiver
- Give a range proof that the sum of outputs minus inputs equal 0

Anonymity set size

- The public keys participating in a transaction is its anonymity set
- The size, quality and diversity of the anonymity set is key to make it work

Real life state

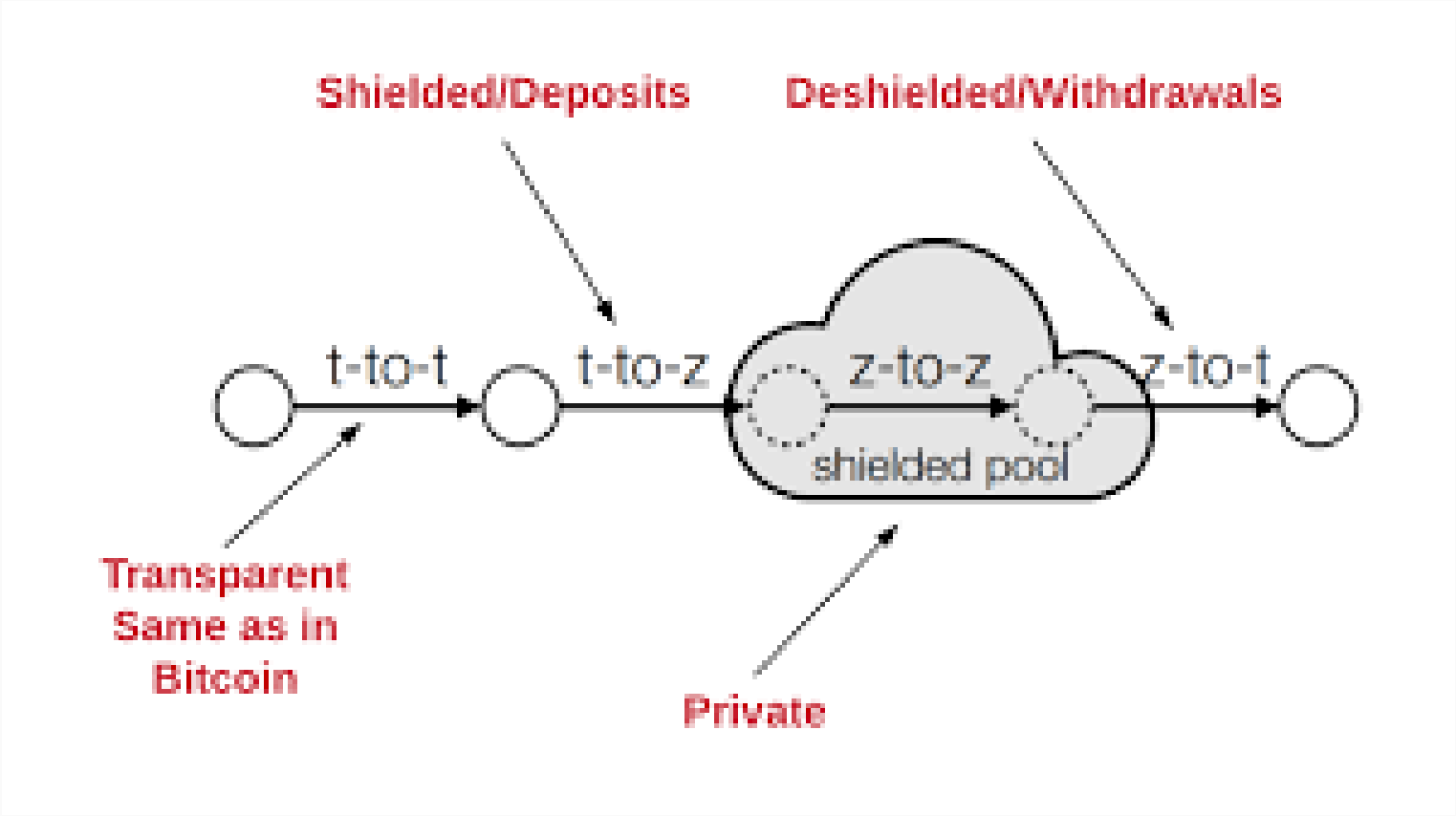
- 65 % of inputs have anonymity set of size 1
- 22 % of inputs are contaminated as a result
- Preceding bitcoin heuristics (timing, multi-input)

Z-cash

Shielded pool

- Implemented in Z-cash
- Pool of coins, in which you can make unlikable transactions
- Amounts, senders and receivers are hidden
- Timing, network information remains
- You can mint a coin: destroy a public coin and create a shielded one
- You can burn a coin: destroy a shielded one and create a public one

Shielded pool



Reminder Merkle tree

Given a set S, and the root of its Merkle tree r there exists two algorithm *Prove* and *Verify* such that for all $x \in S$

- $\mathit{Verify}(\mathit{Prove}(x, S), r) = T$
- There are no reasonable algorithm A, such that given $x' \notin S$, $\mathit{Verify}(A(x, S), r) = T$
- *Verify* runs in $O(\log |S|)$

Zero-knowledge proofs syntax

Given a program $P(x, w) = \textit{boolean}$ there exists two algorithm *Prove* and *Verify*

- $\textit{Prove}(x, w) = \pi$
- $\textit{Verify}(x, \pi) = \textit{boolean}$

ZKP properties

- Completeness: $\mathit{Verify}(x, \mathit{Prove}(x, w)) = P(x, w)$
- Soundness: There are no reasonable A such that $\mathit{Verify}(x, A(x, w)) = T$ with $P(x, w) = F$
- Zero-knowledge: π gives no information on x

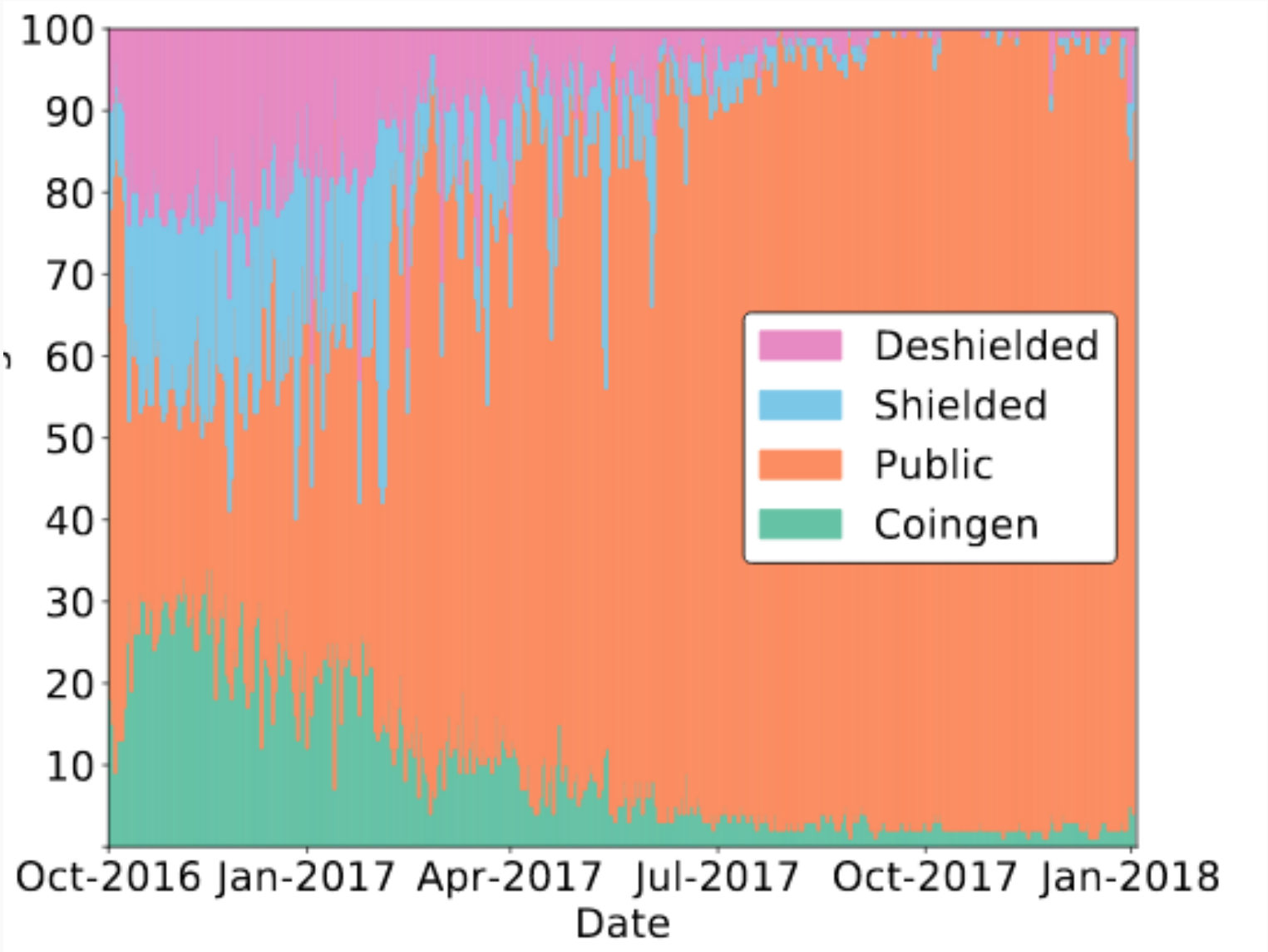
Simplified Z-cash's protocol

- A coin is a commitment c of a random x and a pk, the key is r the opening of the commitment and sk associated to pk with $H(sk) = pk$
- A nullifier associated to a coin is $H(x)$ where H is a hash function
- The blockchain stores the Merkle tree *Tree* of all coins and all the nullfiers of spent coins

Sending money in simplified Z-cash

- To spend give of zk proof of $P(c, Tree; x, r, sk)$ and its nullifier $nf = H(x)$ which verfies that
  - I know the opening r of a commitment c such that
  - c is in the tree
  - $Reveal(c, x|pk, r) = T$
  - $nf = H(x)$
  - $H(sk) = pk$
- Create a new coin and send out-of-band its randomness
- The consensus checks the proof, and that nf wasn't stored before, stores nf and adds the new coin in the tree

Real life state



Conclusion

Conclusion

- Cryptocurrencies are not private
- Cryptographic techniques can/could help
- Bad user behaviour prevents privacy
- Privacy aware user are deanonymised by non privacy aware user
- Big identifiable users makes things worst
- Things might change in the future by increasing the cryptograhpy usage, set to default good behaviours, having a more widespread usage