Decentralised location verification system

Conor Taylor

B.A.(Mod.) Computer Science
Final Year Project, April 2016
Supervisor: Stephen Barrett

Problem

A system that allows participants to verify a users claimed location.

Problem

A system that allows participants to verify a users claimed location.

Goals:

Problem

A system that allows participants to verify a users claimed location.

Goals:

- False location claims must be detectable.

Problem

A system that allows participants to verify a users claimed location.

Goals:

- False location claims must be detectable.
- Privacy protecting.

Problem

A system that allows participants to verify a users claimed location.

Goals:

- False location claims must be detectable.
- Privacy protecting.
- Cannot rely on any centralised resources.

Problem

A system that allows participants to verify a users claimed location.

Goals:

- False location claims must be detectable.
- Privacy protecting.
- Cannot rely on any centralised resources.
- Capable of running in the background on mobile devices.

Background

There are **no** known existing decentralised location proof systems.

Existing centralised solutions: hardware and/or software

Design

3 distinct entities:

- Mobile node 
- Miner node 

- Verifier node 

Design

Overview

Mobile node

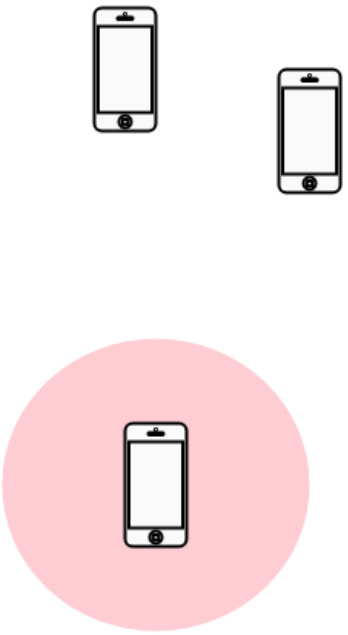
Design

Overview

Mobile node

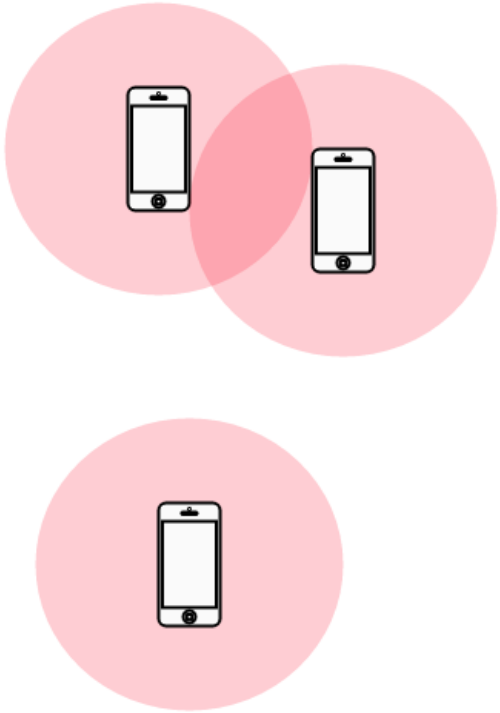
Design

Overview



Design

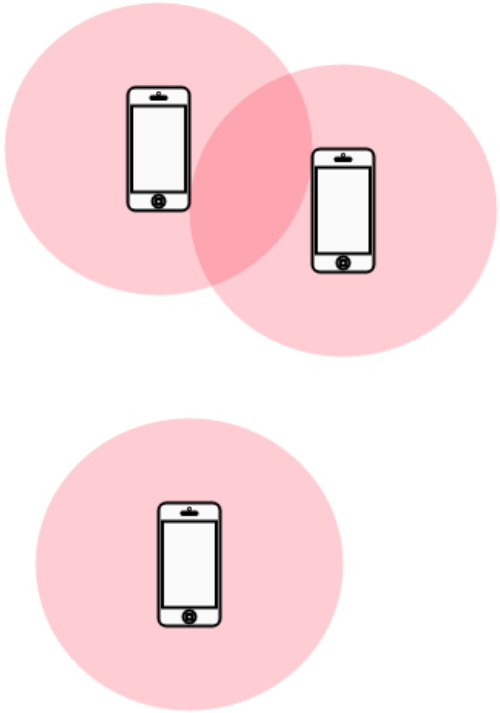
Overview



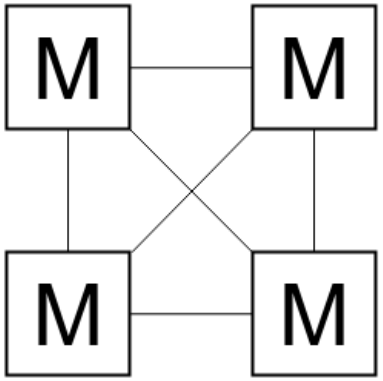
Mobile nodes

Design

Overview



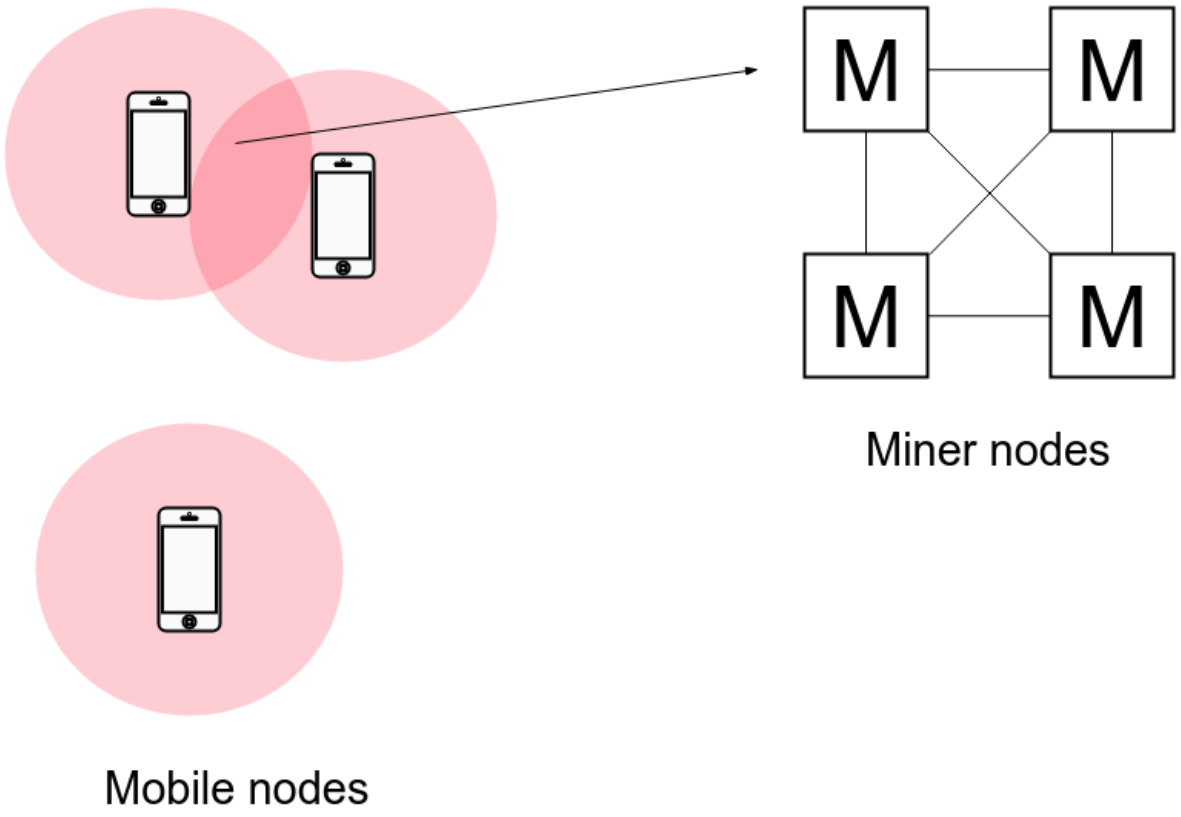
Mobile nodes



Miner nodes

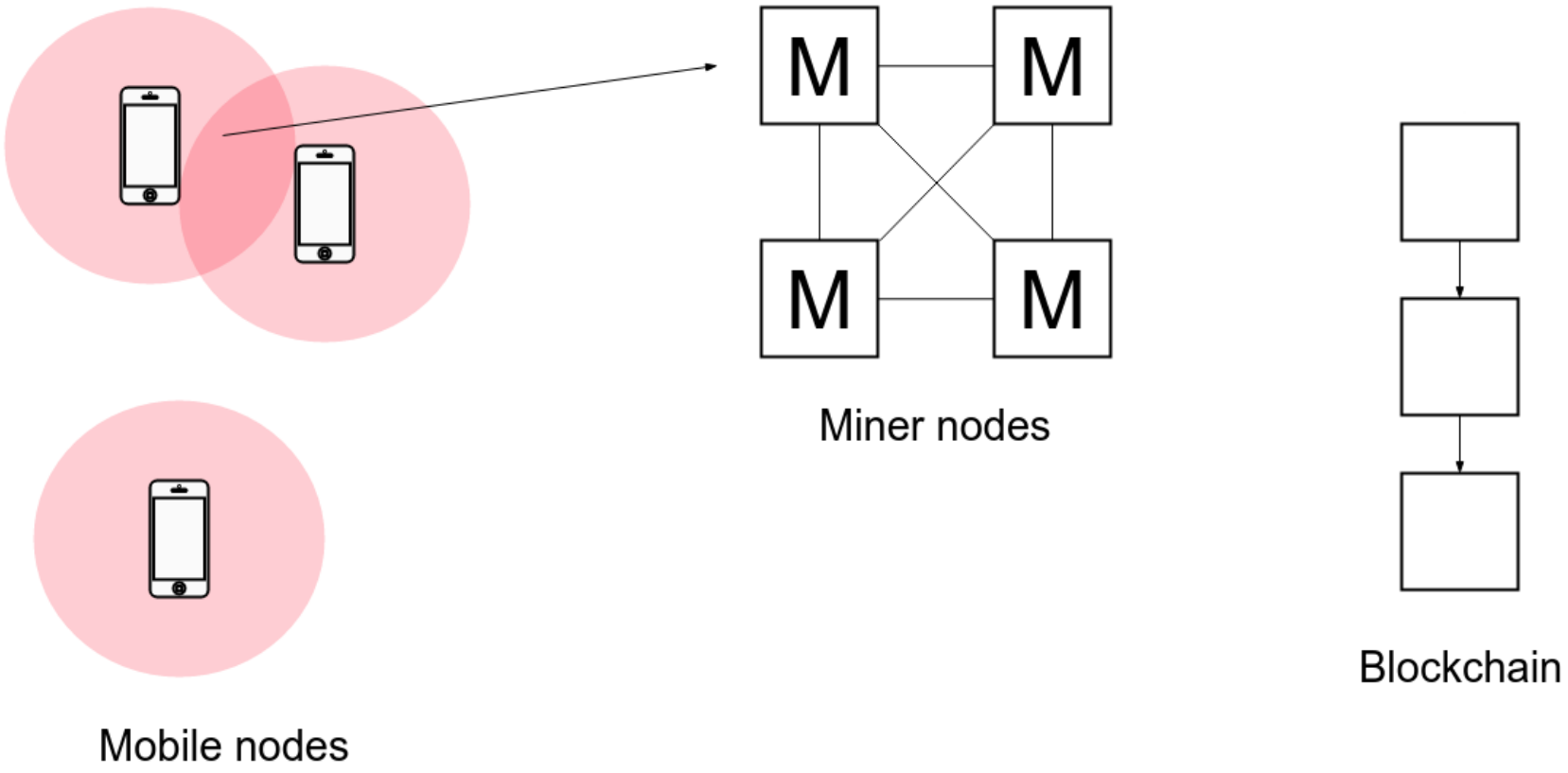
Design

Overview



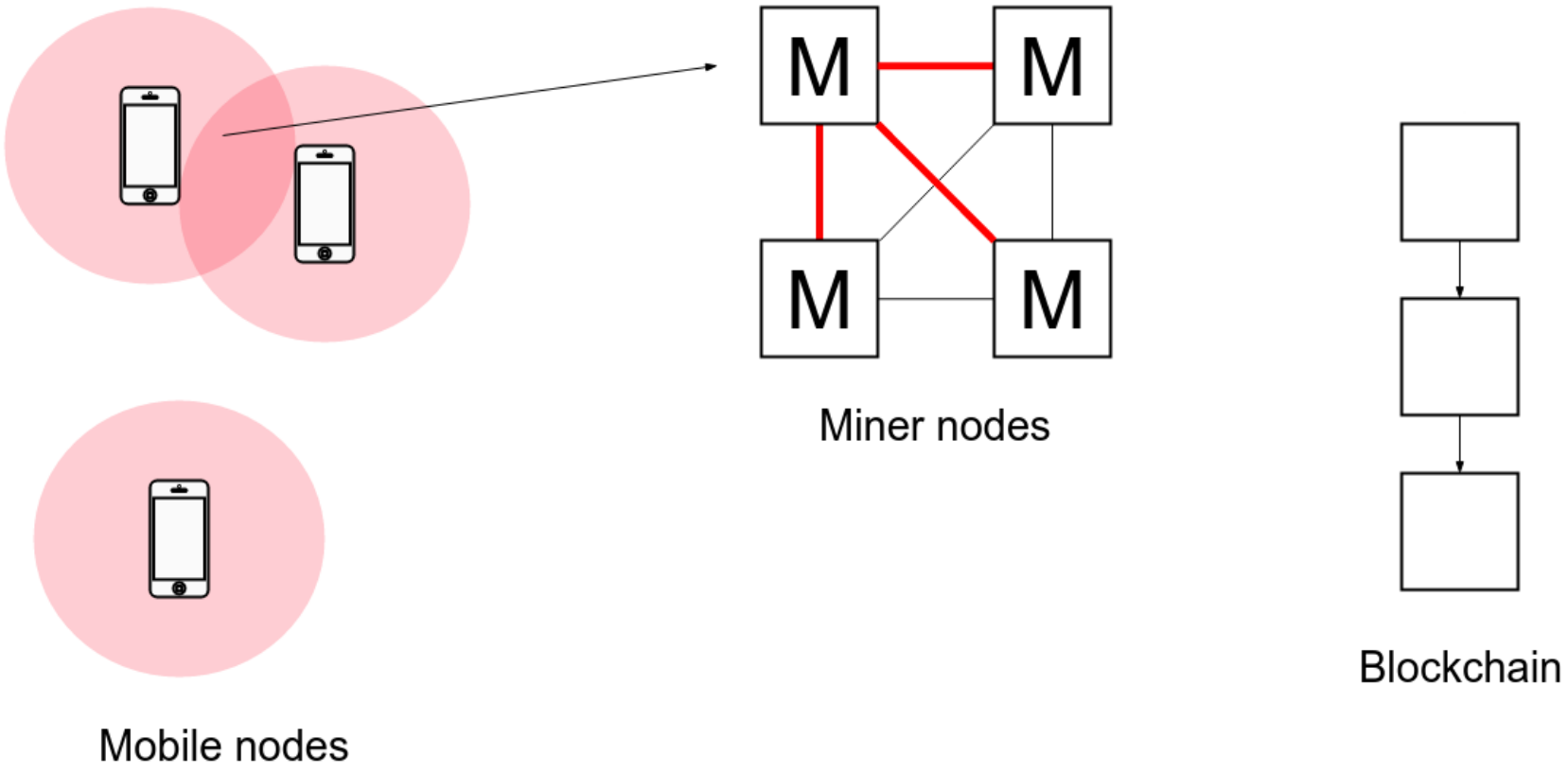
Design

Overview



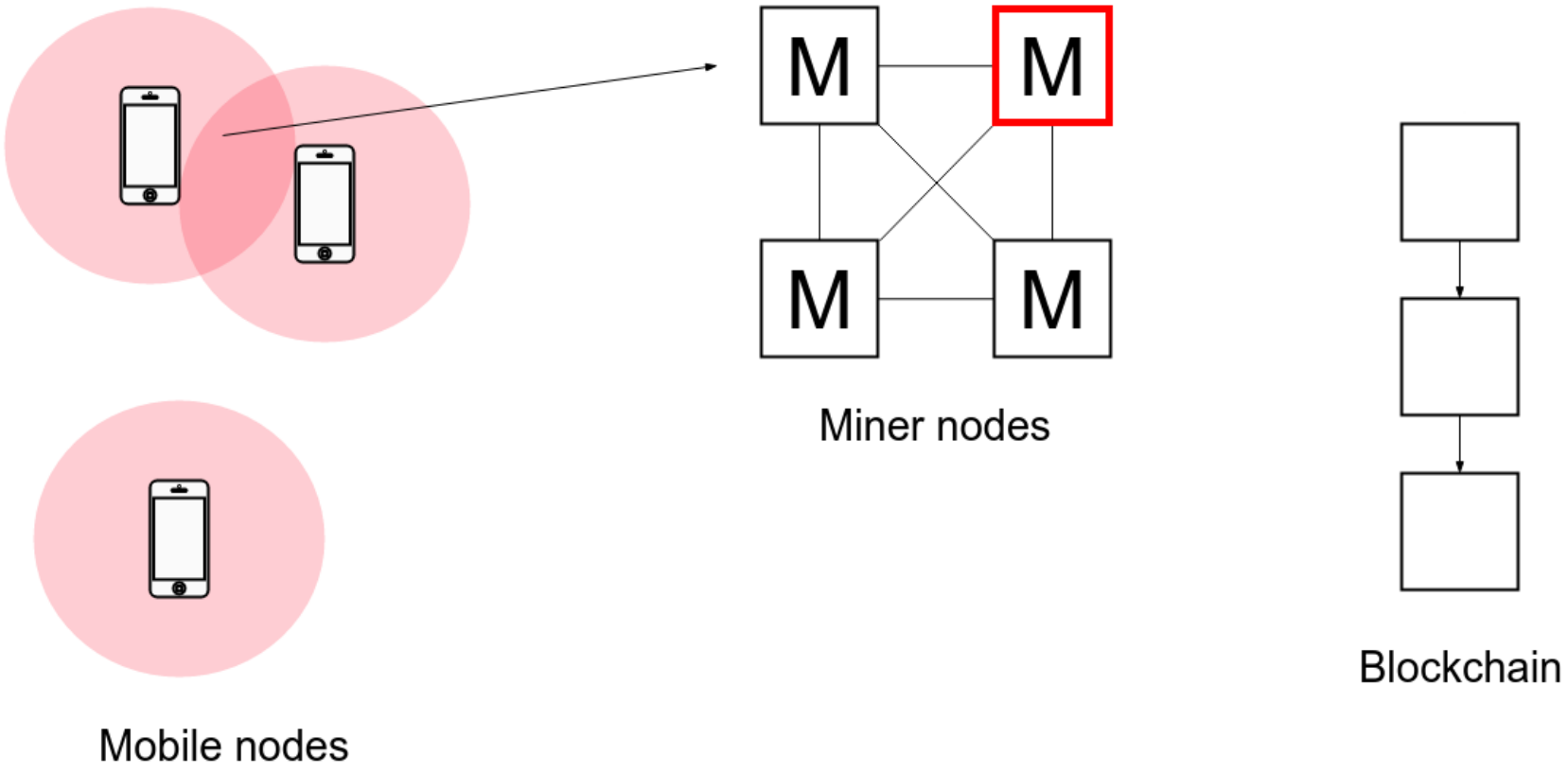
Design

Overview



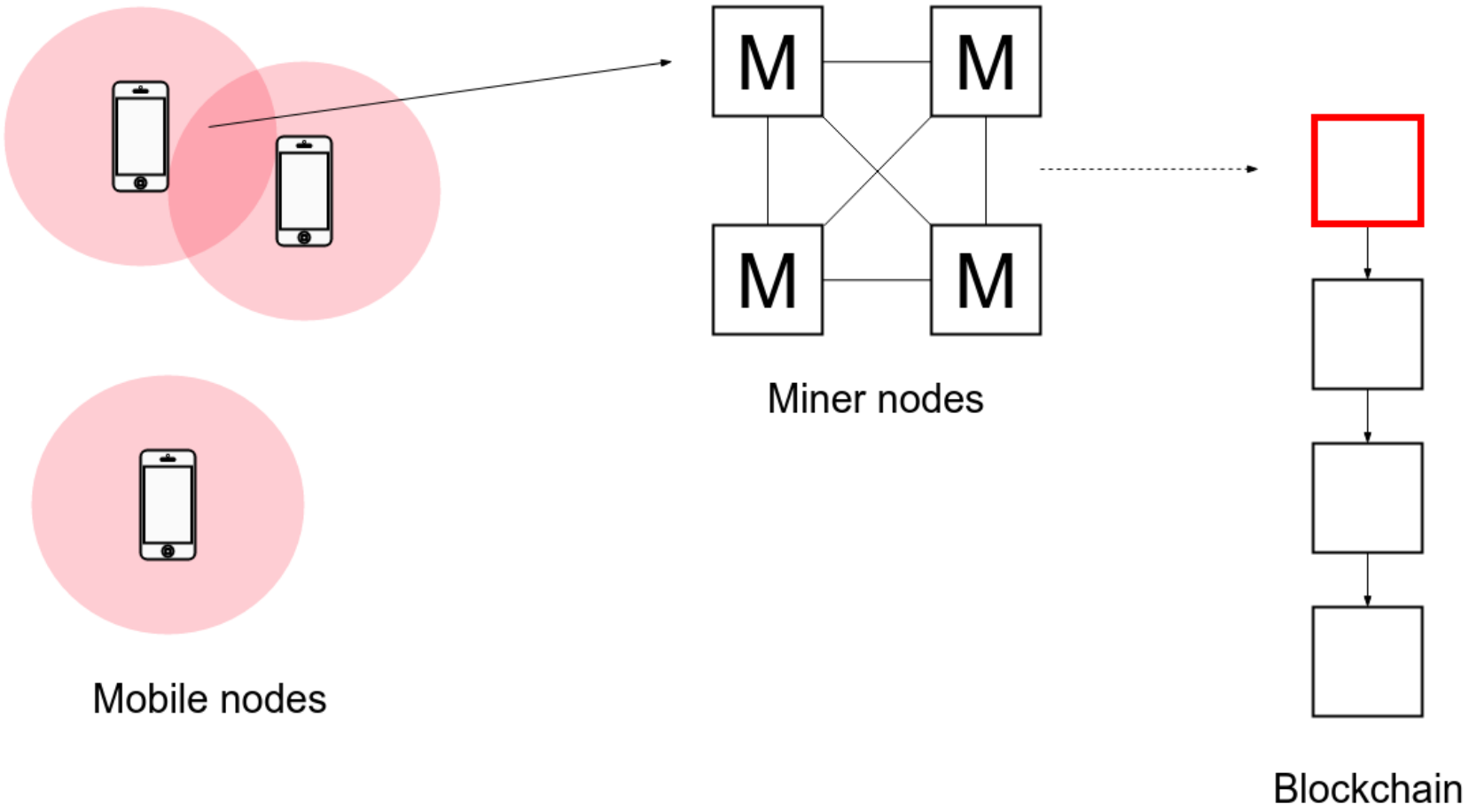
Design

Overview



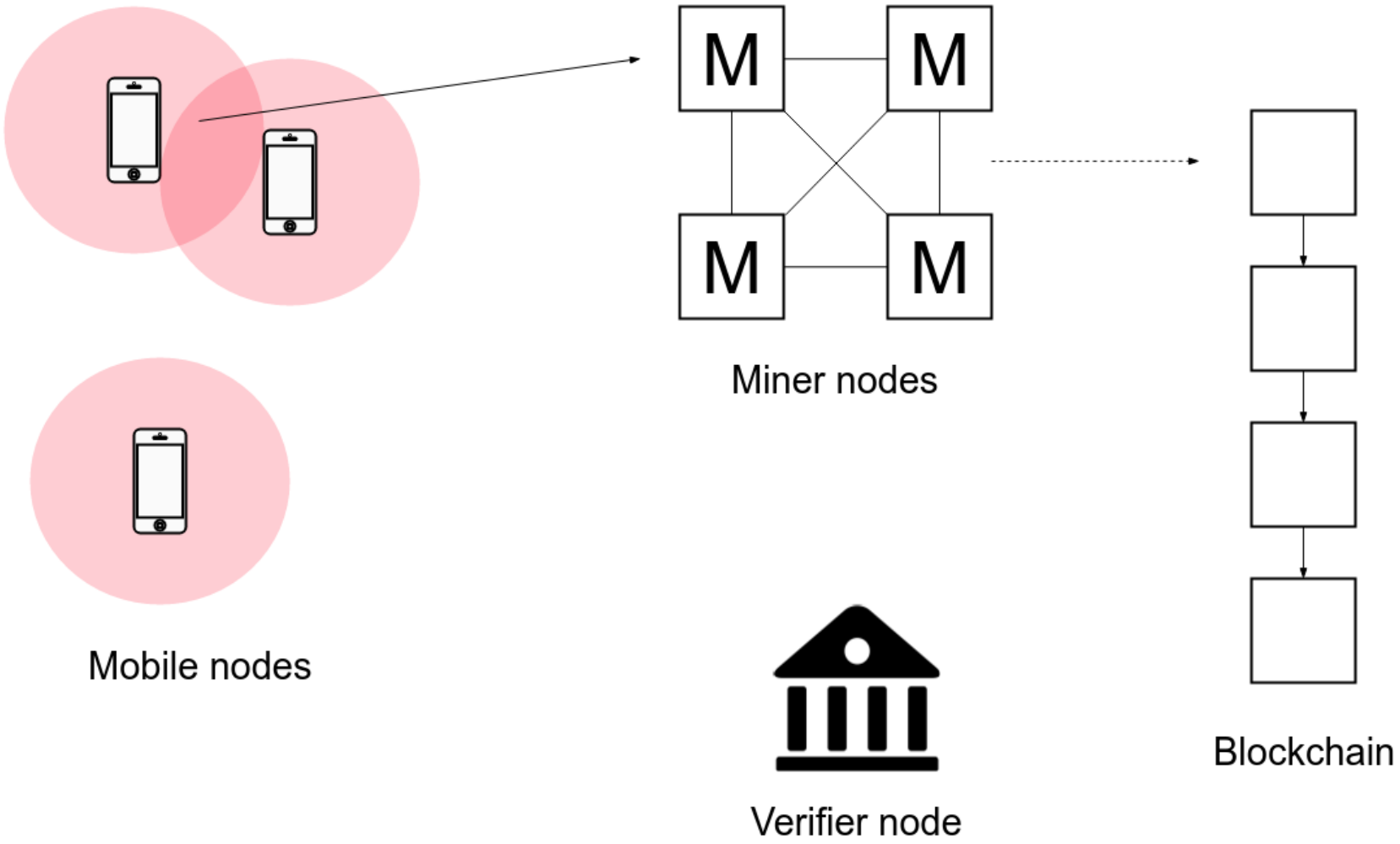
Design

Overview



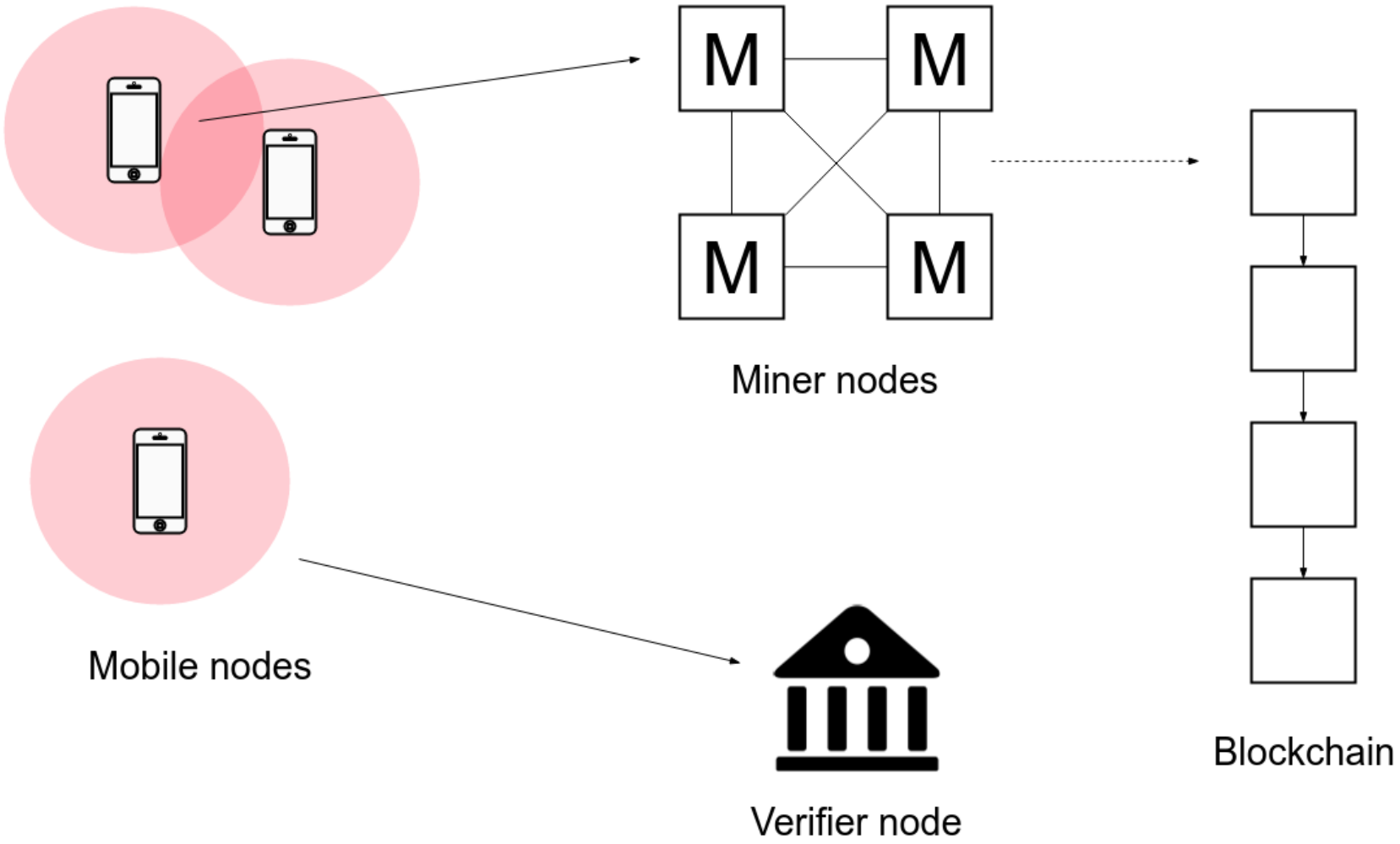
Design

Overview



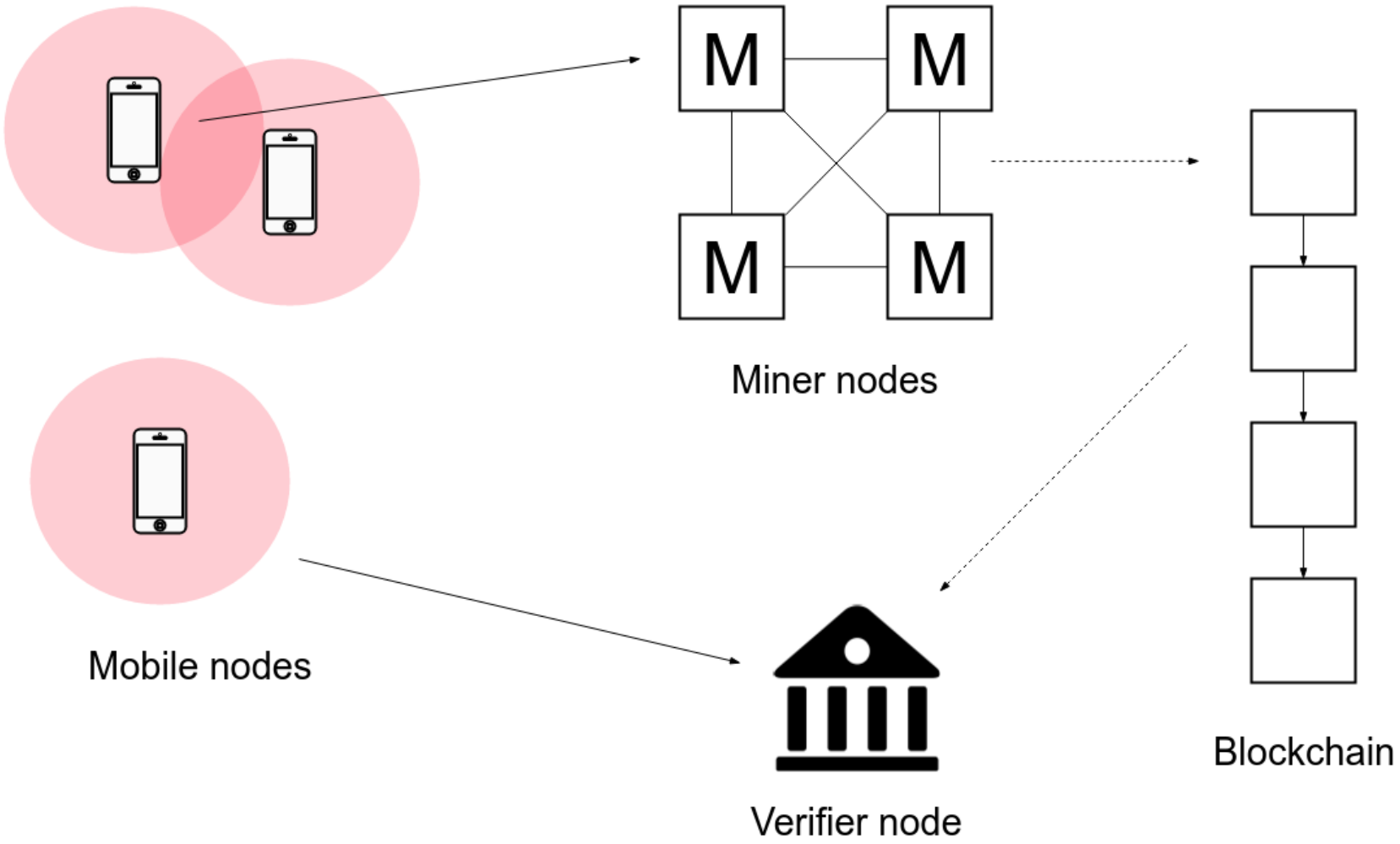
Design

Overview



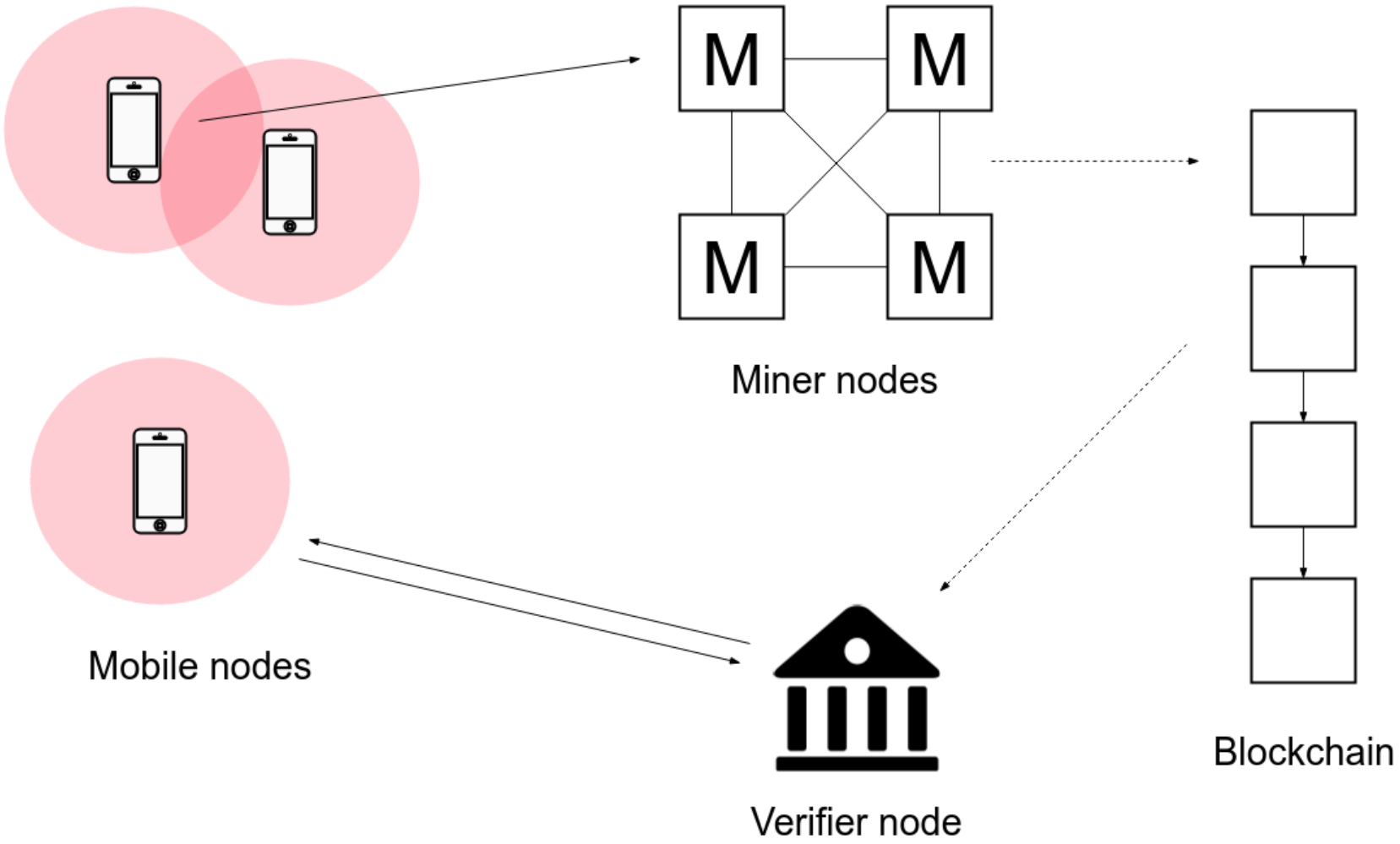
Design

Overview



Design

Overview



Design

Identities

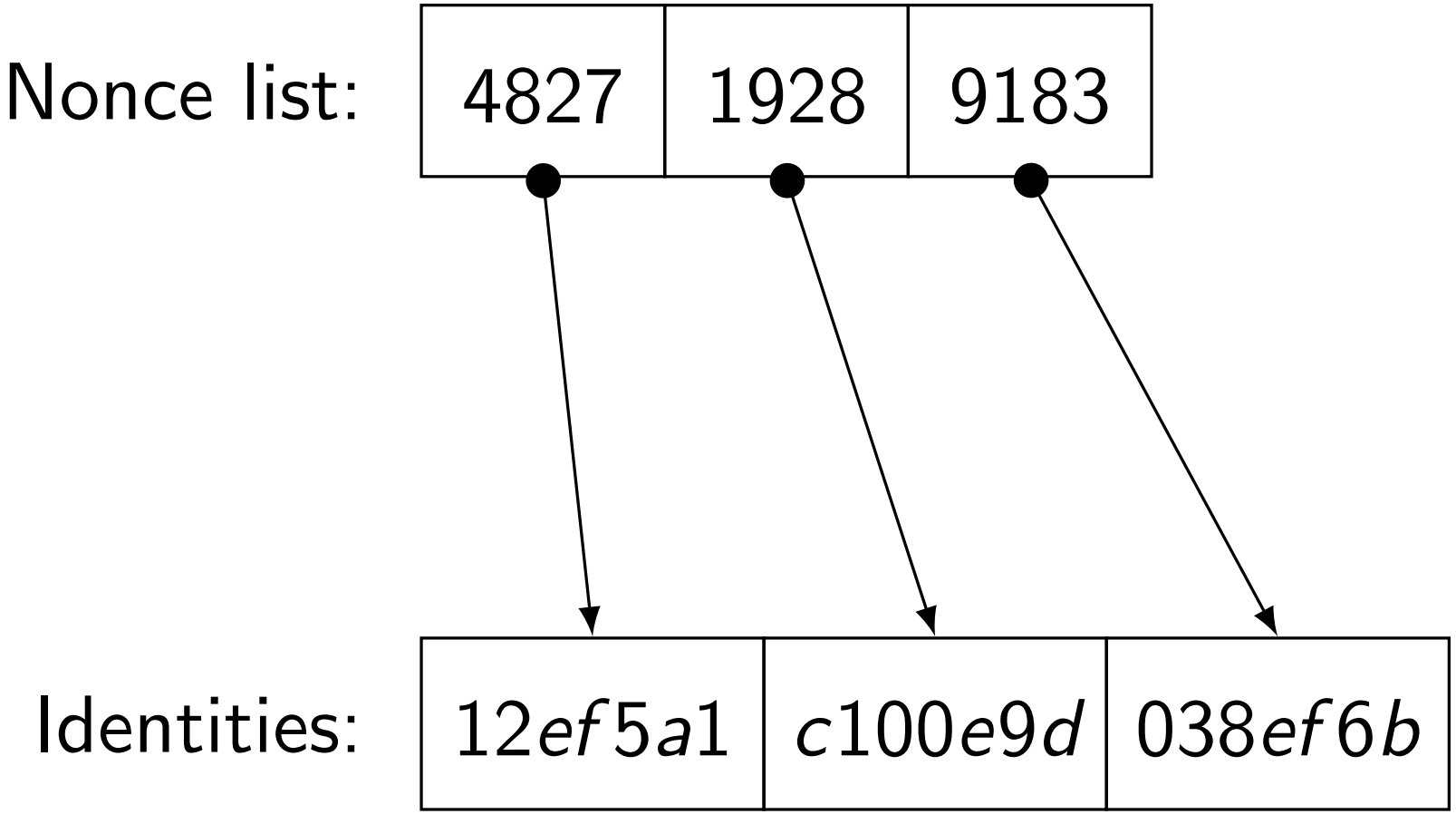
Used to anonymously identify a node in a transaction.

Balancing goals:

- False location claims must be detectable.
- Privacy protecting.

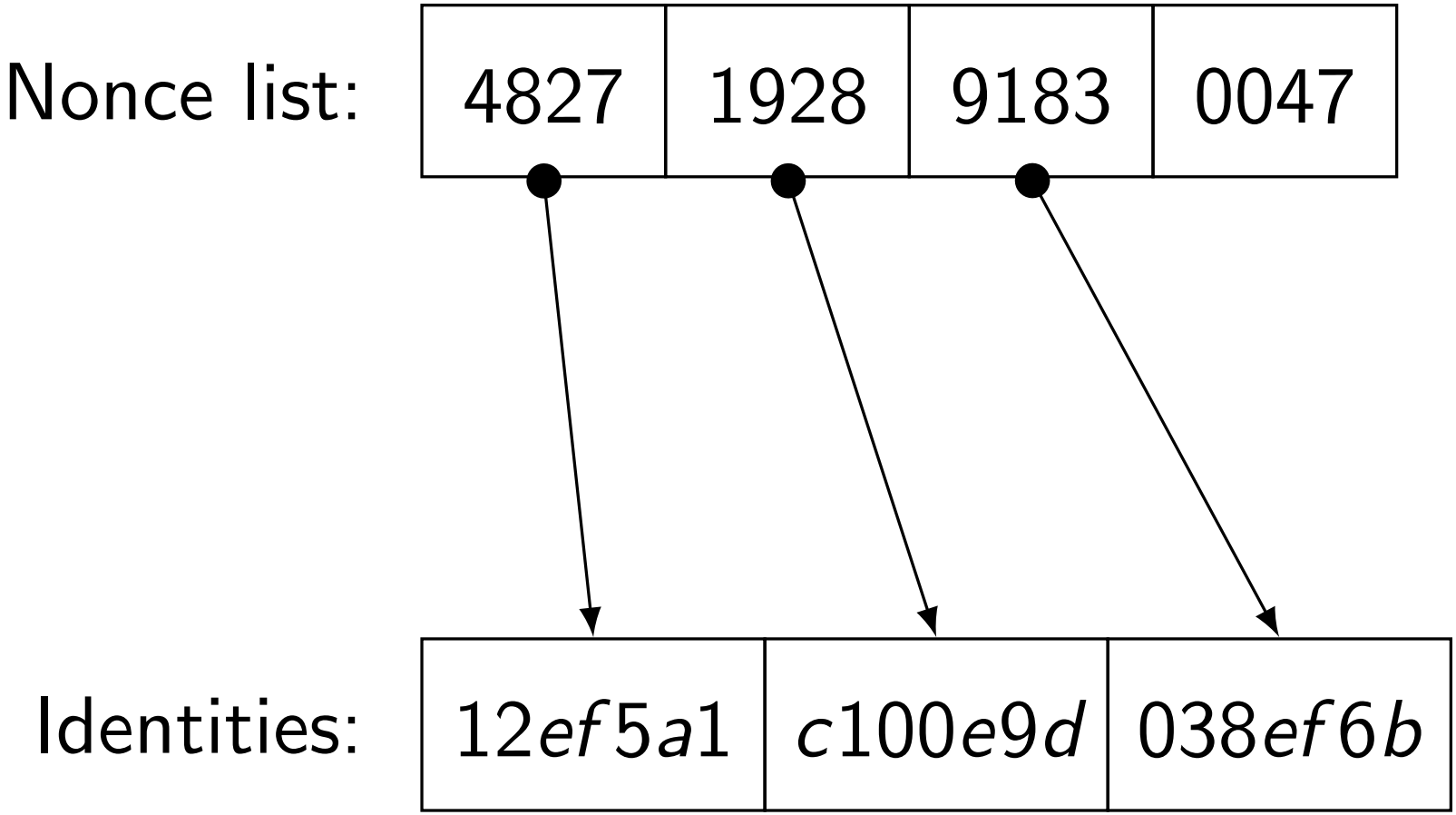
Design

Identities: Nonce Lists



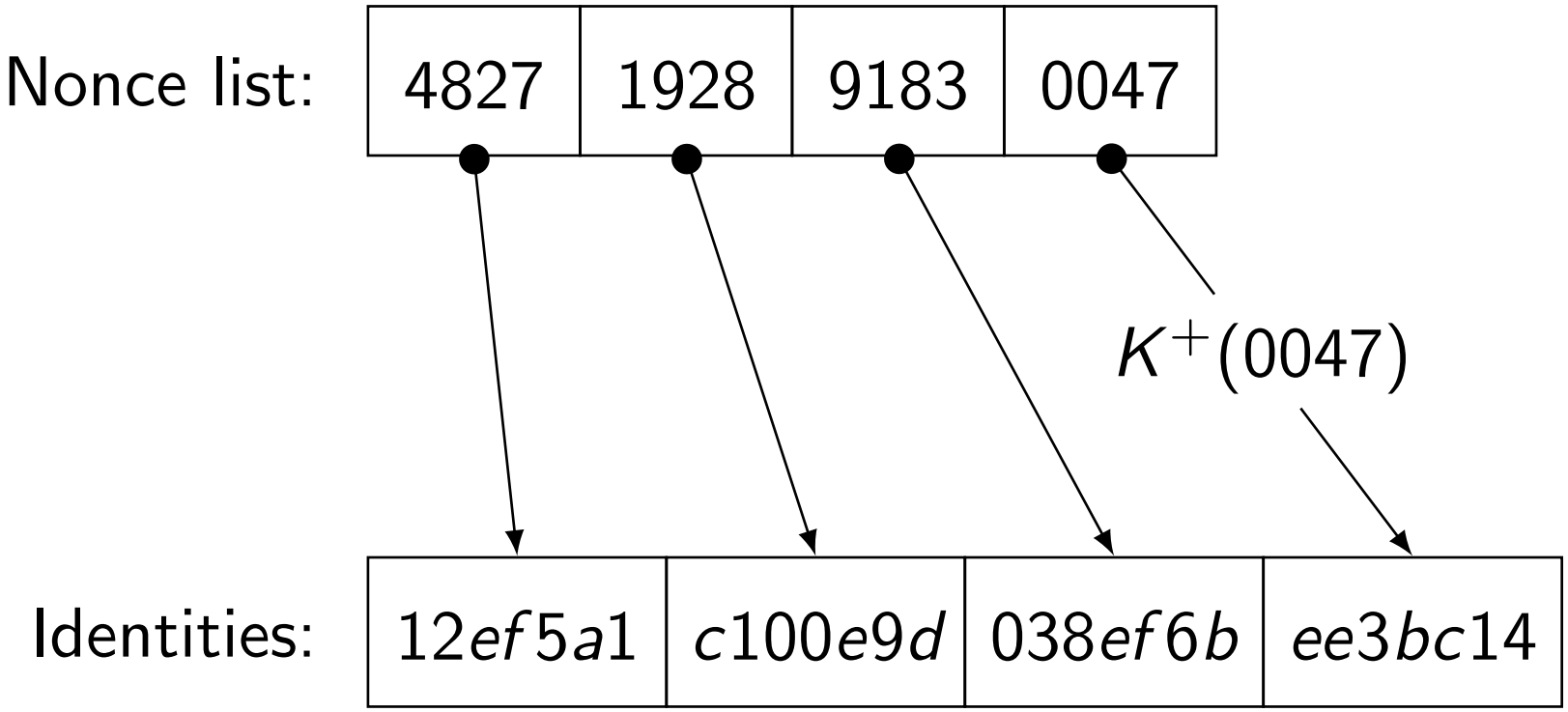
Design

Identities: Nonce Lists



Design

Identities: Nonce Lists



Design

Identities: Duplication

Identity duplication is unavoidable in a decentralised system.

Design

Identities: Duplication

Identity duplication is unavoidable in a decentralised system.

ID	**Contents**
...	
ffa0	
ffa1	
ffa2	T_{A4}
ffa3	
ffa4	T_{B87}
...	

Design

Identities: Duplication

Identity duplication is unavoidable in a decentralised system.

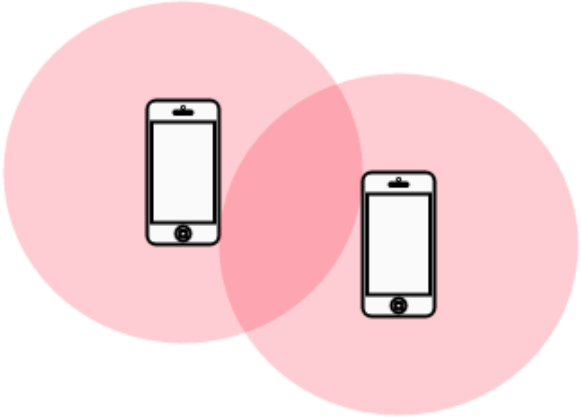
ID	**Contents**
...	
ffa0	
ffa1	
ffa2	T_{A4} , T_{C102}
ffa3	
ffa4	T_{B87}
...	

Design

Transactions

Transactions are created when two mobile nodes physically meet.

- Ad-hoc bluetooth connection between the nodes.



Design

Transactions II

Node A will create the following transaction after meeting node B:

$$T_{An} = K_A(ts_A|loc_A|ID_{An}|KP_{Bm})$$

Design

Transactions II

Node A will create the following transaction after meeting node B:

$$T_{An} = K_A(ts_A|loc_A|ID_{An}|KP_{Bm})$$

Design

Transactions: Key Packets

Key Packets provide a Verifier with a means of examining a mobile node's transactions.

Design

Transactions: Key Packets

Key Packets provide a Verifier with a means of examining a mobile node's transactions.

Two main properties:

- Allow a Verifier to build a tree of a mobile node's activity.
- Allow a mobile node to preserve control its own privacy.

Design

Transactions: Key Packets

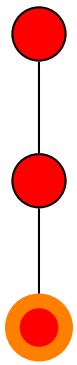
Key Packets provide a Verifier with a means of examining a mobile node's transactions.

Two main properties:

- Allow a Verifier to build a graph of a mobile node's activity.
- Allow a mobile node to preserve control its own privacy.

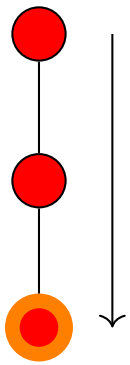
Design

Transactions: Key Packets - Verification



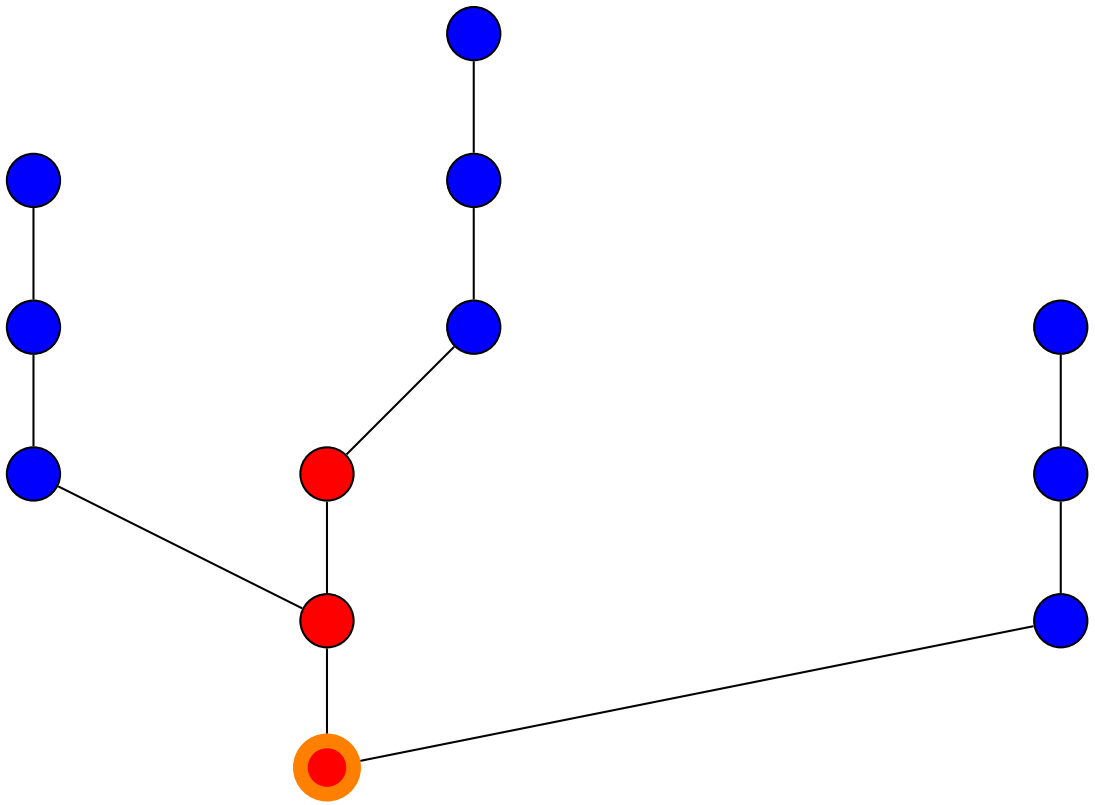
Design

Transactions: Key Packets - Verification



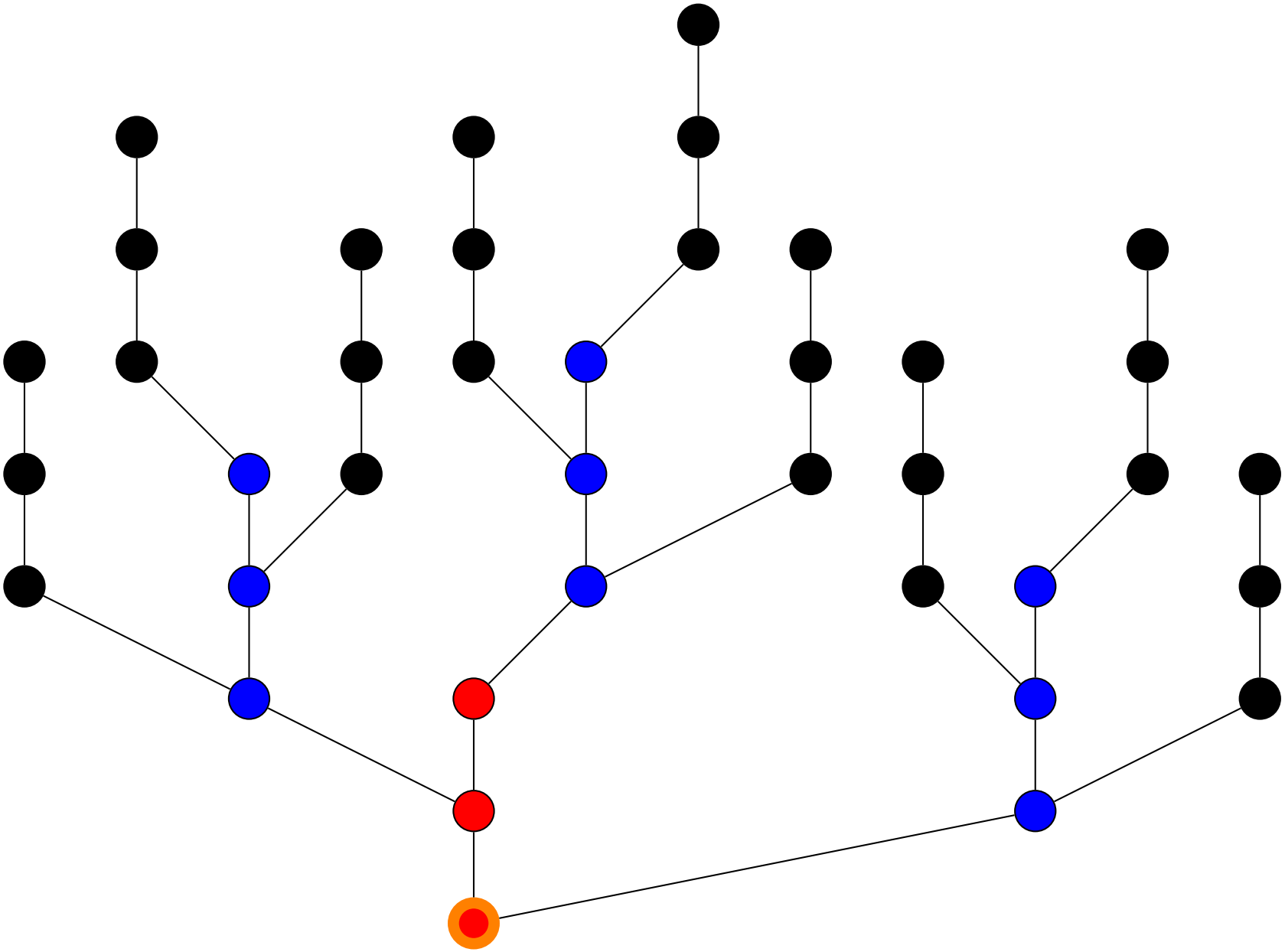
Design

Transactions: Key Packets - Verification



Design

Transactions: Key Packets - Verification



Design

Transactions: Key Packets

Key Packets provide a Verifier with a means of examining a mobile node's transactions.

Two main properties:

- Allow a Verifier to build a graph of a mobile node's activity.
- Allow a mobile node to preserve control its own privacy.

Design

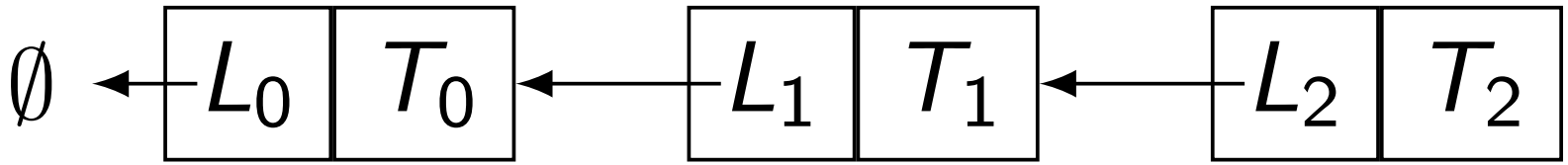
Transactions: Key Packets - Privacy

Published transactions split into two parts: Link and Transaction

Design

Transactions: Key Packets - Privacy

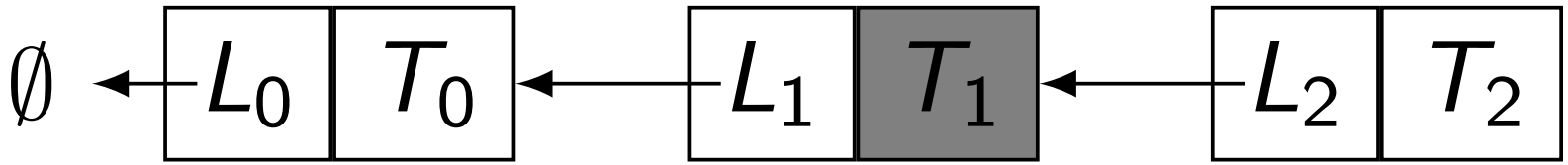
Published transactions split into two parts: Link and Transaction



Design

Transactions: Key Packets - Privacy

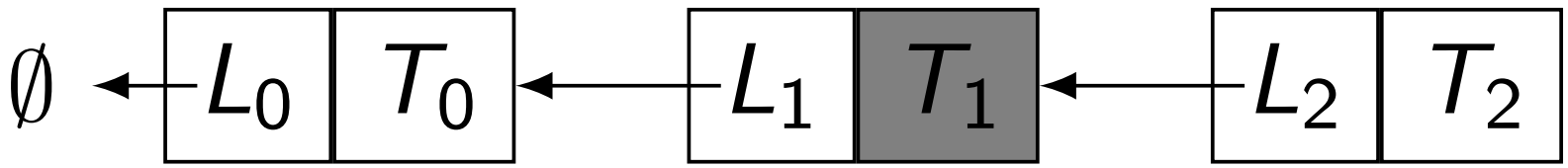
Published transactions split into two parts: Link and Transaction



Design

Transactions: Key Packets - Privacy

Published transactions split into two parts: Link and Transaction

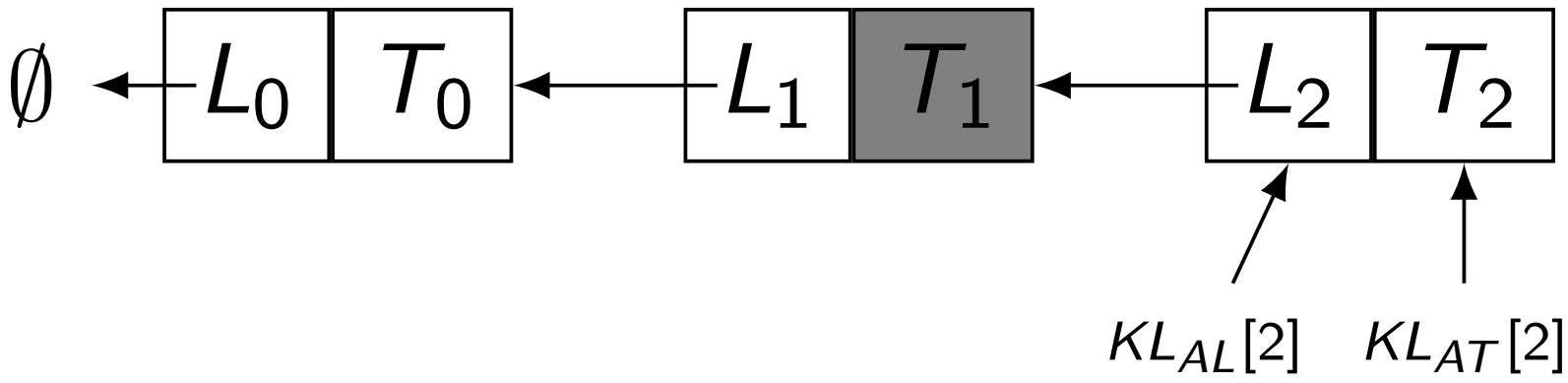


Two *Key Lists*: KL_{AT} and KL_{AL}.

Design

Transactions: Key Packets - Privacy

Published transactions split into two parts: Link and Transaction

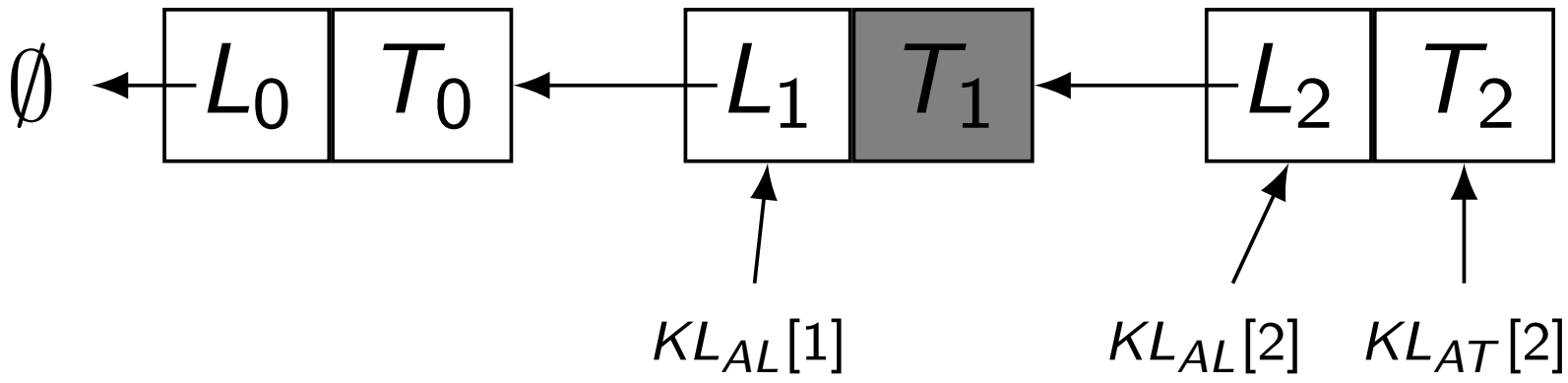


Two *Key Lists*: KL_{AT} and KL_{AL}.

Design

Transactions: Key Packets - Privacy

Published transactions split into two parts: Link and Transaction



Two *Key Lists*: KL_{AT} and KL_{AL}.

Design

Transactions III

Node A will create the following transaction after meeting node B:

$$T_{An} = K_A(ts_A|loc_A|ID_{An}|KP_{Bm})$$

Design

Transactions III

Node A will create the following transaction after meeting node B:

$$T_{An} = KL_{AT}[n](ts_A|loc_A|ID_{An}|KP_{Bm})$$

Design

Transactions III

Node A will create the following transaction after meeting node B:

$$T_{An} = KL_{AT}[n](ts_A|loc_A|ID_{An}|KP_{Bm})$$

Node A will then publish the following to the blockchain:

$$P_{An} = ID_{An}|KL_{AL}[n](ID_{An-1}|ts_A)|T_{An}$$

Design

Verification

Mobile node needs to provide Verifier node with:

- ID of most recent transaction.
- Key Packet for n most recent transactions.
- Nonce list for n most recent IDs.
- Public key.