Distributed IOweYou Credit Networks

Aniket Kate
Purdue University

DCCL 2016
Blockchains can change a lot of things

Source: http://startupmanagement.org/blog
Issues with Global Blockchains
Issues with Global Blockchains

✦ 51% attack (or The tyranny of power)
  ✦ If one controls more than half of the computation power in any permission-less cryptocurrency system, he/she can …
Issues with Global Blockchains

✦ 51% attack (or The tyranny of power)
  ✦ If one controls more than half of the computation power in any permission-less cryptocurrency system, he/she can …

✦ Money Generation (or Proof of Work)
  ✦ Bitcoin mining could consume as much electricity as Denmark by 2020!
Issues with Global Blockchains

- **51% attack (or The tyranny of power)**
  - If one controls more than half of the computation power in any permission-less cryptocurrency system, he/she can …

- **Money Generation (or Proof of Work)**
  - Bitcoin mining could consume as much electricity as Denmark by 2020!

- **Scalability and Speed**
  - Bitcoin’s 7 tps vs VISA’s 2000 tps
Issues with Global Blockchains

✦ 51% attack (or The tyranny of power)
  ✦ If one controls more than half of the computation power in any permission-less cryptocurrency system, he/she can …

✦ Money Generation (or Proof of Work)
  ✦ Bitcoin mining could consume as much electricity as Denmark by 2020!

✦ Scalability and Speed
  ✦ Bitcoin’s 7 tps vs VISA’s 2000 tps

✦ Money vs. Credit
  ✦ Money: medium of exchange
  ✦ Credit: exchange of present goods against future goods
Issues with Global Blockchains

- **51% attack (or The tyranny of power)**
  - If one controls more than half of the computation power in any permission-less cryptocurrency system, he/she can …

- **Money Generation (or Proof of Work)**
  - Bitcoin mining could consume as much electricity as Denmark by 2020!

- **Scalability and Speed**
  - Bitcoin’s 7 tps vs VISA’s 2000 tps

- **Money vs. Credit**
  - Money: medium of exchange
  - Credit: exchange of present goods against future goods

- **IOU (or Credit) Networks**
  - Combining credit and social trust (still not permissioned)
This talk aims at leaving you with *more questions* than answers regarding credit networks ;)}
Transactions in the real world

Bob → Alice: $100

Bob ← Alice: IOweYou 100
Transactions in the real world

Bob $100 Alice

Bob IOweYou 100 Alice

A credit network representation

Bob 100 Alice
Credit (or IOU settlement) Networks: Basics

Transactions in the real world

1. Bob gives Alice $100.
2. Bob owes Alice $100.

During a hike with Alice & Bob

1. Dave gives Carol $10.
2. Dave owes Carol $10.

A credit network representation

1. Bob owes Alice $100.
2. Alice owes Bob $100.
Credit (or IOU settlement) Networks: Basics

Transactions in the real world

Bob → Alice: $100
Bob ← Alice: IOweYou 100

During a hike with Alice & Bob

Dave → Carol: $10
Dave ← Carol: IOweYou 10
Transactions in the real world

Bob → Alice: $100
Bob → Alice: IOweYou 100

During a hike with Alice & Bob

Dave → Carol: $10
Dave → Carol: IOweYou 10

A credit network representation

Bob → Alice: 110
Dave → Bob: 10
Dave → Carol: 10
Alice → Carol: 10
Payment (or credit) Network: an Example

Bob → Eve: 10
Bob → Carol: 5
Eve → Carol: 15
Eve → Dave: 20
Eve → Alice: 115
Carol → Dave: 30
Payment (or credit) Network: an Example

Bob

Eve

Carol

Dave

Alice

Payment Amounts:

- Bob to Eve: 15
- Carol to Eve: 5
- Eve to Dave: 30
- Dave to Alice: 115
- Eve to Bob: 10
- Eve to Carol: 15
- Alice to Dave: 20
Payment (or credit) Network: an Example

Max-flow Computation

Bob → Eve: 10
Eve → Carol: 5
Eve → Dave: 15
Dave → Alice: 115
Bob → Eve: 15

Payment (or credit) Network: an Example

Max-flow Computation

Bob → Eve: $10$
Eve → Dave: $15$
Eve → Carol: $5$
Carol → Bob: $5$
Eve → Alice: $20$
Dave → Eve: $15$
Dave → Alice: $115$

Bob: $15$
Carol: $0$
Eve: $15$
Dave: $30$
Alice: $60$

Payment (or credit) Network: an Example

Max-flow Computation

Bob

Carol

Eve

Dave

Alice

Max-flow Computation

Bob

Carol

Eve

Dave

Alice

15

0

0

20

5

20

115
Payment (or credit) Network: an Example

Bob 0 Eve 20 Dave

Carol 0 5 115 Alice

Dave 20
Why credit networks matter?

- A flexible-yet-robust design for distributed (transitive) trust
  - through pairwise credit allocations
- Loss incurred due to misbehaving identities is bounded and (sometimes) localized
Why credit networks matter?

- A flexible-yet-robust design for distributed (transitive) trust
  - through pairwise credit allocations
- Loss incurred due to misbehaving identities is bounded and (sometimes) localized
Building trust with credit networks

✦ Sybil-resistant applications
Building trust with credit networks

✦ Sybil-resistant applications
Building trust with credit networks

- Sybil-resistant applications
Building trust with credit networks

- Sybil-resistant applications
Building trust with credit networks

✦ Sybil-resistant applications

![Diagram of credit network with well-behaved nodes and Sybil nodes. The diagram shows edge cut with numbers 30, 20, and 15 indicating connections.]
Building trust with credit networks

✦ Sybil-resistant applications

Introducing nodes is much easier than drawing trust from well-behaved nodes.
Building trust with credit networks

✦ Sybil-resistant applications

Introducing nodes is much easier than drawing trust from well-behaved nodes.

Well-behaved nodes

Sybil nodes

edge cut

✦ Several Systems
  ✦ Ostra: preventing e-mail spam [NSDI’08]
Building trust with credit networks

- Sybil-resistant applications

- Several Systems
  - Ostra: preventing e-mail spam [NSDI’08]
  - Bazaar: strengthening e-commerce [NSDI’11]

Introducing nodes is much easier than drawing trust from well-behaved nodes
Building trust with credit networks

- **Sybil-resistant applications**
  - Ostra: preventing e-mail spam [NSDI’08]
  - Bazaar: strengthening e-commerce [NSDI’11]
  - SumUp: Sybil-resilient content voting [NSDI’09]

Introducing nodes is much easier than drawing trust from well-behaved nodes.

Several Systems

- Introducing nodes is much easier than drawing trust from well-behaved nodes.

Well-behaved nodes

Sybil nodes

30

20

15

edge cut
Building trust with credit networks

- Sybil-resistant applications
  - Several Systems
    - Ostra: preventing e-mail spam [NSDI’08]
    - Bazaar: strengthening e-commerce [NSDI’11]
    - SumUp: Sybil-resilient content voting [NSDI’09]
    - Ripple: A real-life online settlement network

Introducing nodes is much easier than drawing trust from well-behaved nodes.

Well-behaved nodes

Sybil nodes

Introducing nodes is much easier than drawing trust from well-behaved nodes.
Ripple Credit (or Settlement) Network
Ripple Credit (or Settlement) Network
Ripple Credit (or Settlement) Network
Ripple Credit (or Settlement) Network
Ripple Credit (or Settlement) Network

- **CBW BANK**
- **fidor BANK**
- **Cross River Bank**
- **Santander**

- **$60** to **$40**
- **$100** to **$100**
- **£30** to **£45**
- **B10** to **B5**
- **€40** to **€10**
- **£280**
- **£70**

**Tx time**
- ~1 day
- ~5 seconds

**Worldwide, inter-currency tx**
- High fees
- Tiny fees

**Integrity**
- Bank only
- Public verifiability
We already have cryptocurrencies, then why do we need Ripple?
**Ripple vs Bitcoin**

<table>
<thead>
<tr>
<th>Definition</th>
<th>Currency</th>
<th>Transaction network</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transfer of funds</td>
<td>Direct transactions between any two wallets</td>
<td>Transactions only via a path with enough credit</td>
</tr>
<tr>
<td>Scalability</td>
<td>Limited transaction rate (&lt;100 tps)</td>
<td>Highly scalable</td>
</tr>
</tbody>
</table>

We already have cryptocurrencies, then why do we need Ripple?
## Ripple vs Bitcoin

We already have cryptocurrencies, then why do we need Ripple?

<table>
<thead>
<tr>
<th>Definition</th>
<th>Currency</th>
<th>Transaction network</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transfer of funds</td>
<td><strong>Direct</strong> transactions between any two wallets</td>
<td>Transactions <strong>only</strong> via a path with enough credit</td>
</tr>
<tr>
<td>Scalability</td>
<td><strong>Limited</strong> transaction rate (&lt;100 tps)</td>
<td>Highly scalable</td>
</tr>
</tbody>
</table>

**Public verifiability of transactions**
Attacks on privacy of Ripple links & transactions

Ripple provides **pseudonymity** to its users by employing public-key hashes as identities.
Attacks on privacy of Ripple links & transactions

Transaction Details

<table>
<thead>
<tr>
<th>Account</th>
<th>Destination</th>
<th>Amount</th>
</tr>
</thead>
<tbody>
<tr>
<td>rwvctTPLKZqK59f1fXpDkQ...</td>
<td>rMnVZ9maUWp5cAvmqBECZM...</td>
<td>380/XRP</td>
</tr>
<tr>
<td>rLSBpSquSHKbbfvcKt1c54...</td>
<td>rKoDt7VL83AKLYwLxVZEs...</td>
<td>75/XRP</td>
</tr>
<tr>
<td>r428G9fSSrDpQ6x4h16y...</td>
<td>rBeToNo4AwMaMbRX2n48NC...</td>
<td>0.8693482709148/CCK/rB...</td>
</tr>
<tr>
<td>rhD759dbJM2w33y2v9h5...</td>
<td>r42WJGvV9N9CfKj3Sn0...</td>
<td>380/XRP</td>
</tr>
<tr>
<td>rUnr1p7xkD6r83e1...</td>
<td>rw7UfGvzC93G...</td>
<td></td>
</tr>
<tr>
<td>rpWzntYw3...</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Credit Graph

It is possible to link multiple transactions and identities belonging to the same user.

Ripple provides **pseudonymity** to its users by employing public-key hashes as identities.
Is privacy a real problem in Ripple?

*Privacy Attacks: Innocent until Proven Guilty*
Is privacy a real problem in Ripple?

*Privacy Attacks: Innocent until Proven Guilty*

P. Moreno-Sanchez, M. B. Zafar, A. Kate:  
**Linking Wallets and Deanonymizing Transactions in the Ripple Network.**  
*Privacy Enhancing Technologies Symposium (PETS) 2016.*

Ripple Forum Discussion:  
Towards privacy-preserving transactions credit networks
Towards privacy-preserving transactions credit networks

P. Moreno-Sanchez, A. Kate, M. Maffei, and K. Pecina:
Privacy Preserving Payments in Credit Networks. 
NDSS 2015
Defining privacy for a credit network
Defining privacy for a credit network

Transaction value privacy

\[10 \approx 30\]
Defining privacy for a credit network

Transaction value privacy

\[ p_{\text{trans}} \approx 10^{30} \]

Transaction receiver privacy

\[ p_{\text{receiver}} \approx \]

Bob → Carol

Bob → Carol

Bob → Carol

Bob → Dave
Transaction sender privacy can be defined similarly.
Towards credit network privacy
Towards credit network privacy

- A decentralized or centralized architecture?
Towards credit network privacy

✧ A decentralized or centralized architecture?

✧ **Centralized setting:** the network is maintained by a server
  - The service provider can trivially break the privacy
    - The routing computation can be performed privately, but any modifications to the edges not
    - Use of pseudonyms and anonymous channels (e.g., Tor) is not sufficient
  - In our NDSS’15 paper, we resolve this issue using minimally trusted hardware and oblivious algorithms
Towards credit network privacy

- A decentralized or centralized architecture?

- **Centralized setting:** the network is maintained by a server
  - The service provider can trivially break the privacy
    - The routing computation can be performed privately, but any modifications to the edges not
    - Use of pseudonyms and anonymous channels (e.g., Tor) is not sufficient
  - In our NDSS’15 paper, we resolve this issue using minimally trusted hardware and oblivious algorithms

- **Decentralized setting:** edges are maintained locally
  - A transaction passing through a node requires its active involvement
  - We will focus on this solution in the rest of the talk
A Distributed Credit Network

- Each user maintains her own credit links
A Distributed Credit Network

- Each user maintains her own credit links
A Distributed Credit Network

- Each user maintains her own credit links
Local Knowledge is Sufficient!

- Credit links of a user determine his credit in the network
Local Knowledge is Sufficient!

✦ Credit links of a user determine his credit in the network

In-flow = 450
Out-flow = 40
Net-flow = 410
Local Knowledge is Sufficient!

- Credit links of a user determine his credit in the network

```
In-flow = 450
Out-flow = 40
Net-flow = 410
```

- A user checks net-flow does not change
Local Knowledge is Sufficient!

✦ Credit links of a user determine his credit in the network

✦ A user checks net-flow does not change

In-flow = 450
Out-flow = 40
Net-flow = 410
Local Knowledge is Sufficient!

✧ Credit links of a user determine his credit in the network

In-flow = 450
Out-flow = 40
Net-flow = 410

✧ A user checks net-flow does not change

In-flow = 450
Out-flow = 40
Net-flow = 410
Local Knowledge is Sufficient!

- Credit links of a user determine his credit in the network.

  ![Network Diagram]

  In-flow = 450
  Out-flow = 40
  Net-flow = 410

- A user checks net-flow does not change.

  ![Network Diagram]

  In-flow = 450
  Out-flow = 40
  Net-flow = 410
Local Knowledge is Sufficient!

- Credit links of a user determine his credit in the network

In-flow = 450
Out-flow = 40
Net-flow = 410

- A user checks net-flow does not change

In-flow = 450
Out-flow = 40
Net-flow = 410
Local Knowledge is Sufficient!

✦ Credit links of a user determine his credit in the network

- In-flow = 450
- Out-flow = 40
- Net-flow = 410

✦ A user checks net-flow does not change

- In-flow = 450
- Out-flow = 40
- Net-flow = 410
Challenges

✦ How to find paths between a sender and a receiver?

✦ How to find the IOU credit available in the path?

✦ How to ensure credit links form a path?

✦ And maintaining strong privacy and accountability guarantees…
Challenges

✦ How to find paths between a sender and a receiver?
✦ How to find the IOU credit available in the path?
✦ How to ensure credit links form a path?
✦ And maintaining strong privacy and accountability guarantees...

A. Kate, M. Maffei, G. Malavolta, and P. Moreno-Sanchez: 
Whispers: A Distributed Architecture for Enforcing Privacy in Credit Networks.
Routing: max-flow computation

- **Routing challenge:**
  Known max-flow algorithms are not scalable: $O(V^3)$ or $O(V^2 \log(E))$

- We employ **landmark routing:**
  Calculate only a subset of all possible routes through intermediary nodes called **landmarks**

[Tsuchiya SigComm’88] [Vishanath et al. Eurosys’12]
Routing: max-flow computation

- **Routing challenge:**
  Known max-flow algorithms are not scalable: $O(V^3)$ or $O(V^2 \log(E))$

- **We employ landmark routing:**
  Calculate only a subset of all possible routes through intermediary nodes called **landmarks**

[Tsuchiya SigComm'88] [Vishanath et al. Eurosys’12]
Routing: max-flow computation

- **Routing challenge:**
  Known max-flow algorithms are not scalable: $O(V^3)$ or $O(V^2 \log(E))$

- **We employ landmark routing:**
  Calculate only a subset of all possible routes through intermediary nodes called **landmarks**

[Tsuchiya SigComm’88] [Vishanath et al. Eurosys’12]
Routing: max-flow computation

✦ Routing challenge:
   Known max-flow algorithms are not scalable: \( O(V^3) \) or \( O(V^2 \log(E)) \)

✦ We employ landmark routing:
   Calculate only a subset of all possible routes through intermediary nodes called landmarks

[Tsuchiya SigComm’88] [Vishanath et al. Eurosys’12]
Routing: max-flow computation

✦ Routing challenge:
Known max-flow algorithms are not scalable: $O(V^3)$ or $O(V^2 \log(E))$

✦ We employ landmark routing:
Calculate only a subset of all possible routes through intermediary nodes called landmarks

[Tsuchiya SigComm’88] [Vishanath et al. Eurosys’12]
Routing: max-flow computation

✦ Routing challenge:
   Known max-flow algorithms are not scalable: $O(V^3)$ or $O(V^2 \log(E))$

✦ We employ landmark routing:
   Calculate only a subset of all possible routes through intermediary nodes called landmarks

[Tsuchiya SigComm’88] [Vishanath et al. Eurosys’12]
Routing challenge:
Known max-flow algorithms are not scalable: $O(V^3)$ or $O(V^2 \log(E))$

We employ landmark routing:
Calculate only a subset of all possible routes through intermediary nodes called landmarks

[Tsuchiya SigComm’88] [Vishanath et al. Eurosys’12]
Credit in a Path

[x]: Secret sharing of x
Credit in a Path

[x]: Secret sharing of x
Credit in a Path

[x]: Secret sharing of x
Credit in a Path

[x]: Secret sharing of x

✧ Given [x] it is not possible to know x
Credit in a Path

Given [x] it is not possible to know x
How to ensure that [x] comes from a user in a path?

[x]: Secret sharing of x
Credit in a Path

[x]: Secret sharing of x

✧ Given [x] it is not possible to know x
✧ How to ensure that [x] comes from a user in a path?
Proof of Credit Links in a Path
Proof of Credit Links in a Path
Proof of Credit Links in a Path

\[ \sigma_1 := \text{Sig}(sk_1, ([30], vk_1, vk_2)) \]
\[ \sigma_2 := \text{Sig}(sk_2, ([30], vk_1, vk_2)) \]
Proof of Credit Links in a Path

\[ \sigma_1 := \text{Sig}(sk_1, ([30], vk_1, vk_2)) \]
\[ \sigma_2 := \text{Sig}(sk_2, ([30], vk_1, vk_2)) \]
Proof of Credit Links in a Path

\[ \sigma_1 := \text{Sig}(sk_1, ([30], vk_1, vk_2)) \]
\[ \sigma_2 := \text{Sig}(sk_2, ([30], vk_1, vk_2)) \]

Correct proof for a path

\((vk_1, vk_2), (vk_2, vk_3), (vk_3, vk_4), \ldots\)
Proof of Credit Links in a Path

\[ \sigma_1 := \text{Sig}(sk_1, ([30], vk_1, vk_2)) \]
\[ \sigma_2 := \text{Sig}(sk_2, ([30], vk_1, vk_2)) \]

Correct proof for a path

\[ (vk_1, vk_2), (vk_2, vk_3), (vk_3, vk_4), \ldots \]
Proof of Credit Links in a Path

\[
\sigma_1 := \text{Sig}(sk_1, ([30], vk_1, vk_2))
\]
\[
\sigma_2 := \text{Sig}(sk_2, ([30], vk_1, vk_2))
\]

Correct proof for a path

(\text{vk}_1, \text{vk}_2), (\text{vk}_2, \text{vk}_3), (\text{vk}_3, \text{vk}_4), \ldots
Proof of Credit Links in a Path

\[ \sigma_1 := \text{Sig}(sk_1, ([30], vk_1, vk_2)) \]
\[ \sigma_2 := \text{Sig}(sk_2, ([30], vk_1, vk_2)) \]

Correct proof for a path

\((vk_1, vk_2), (vk_2, vk_3), (vk_3, vk_4), \ldots\)

Fresh keys per transaction
Privacy-preserving Credit in a Path
Privacy-preserving Credit in a Path

\[ \text{30}, \text{vk}_{1,2}, \sigma_{1,2} \]

\[ \text{15} \]

\[ \text{25} \]

\[ \text{10} \]
Privacy-preserving Credit in a Path

[Diagram showing a network with nodes and arrows labeled with numerical values and symbols representing transactions and credits.]

- From node 30, value [30, vk(1,2), σ(2)]
- From node 30, value [30, vk(1,2), σ(2)]
- From node 15, value [15, vk(2,3), σ(2,3)]
- From node 25, value [25, vk(2,3), σ(2,3)]
- From node 10, value [10]
Privacy-preserving Credit in a Path
Privacy-preserving Credit in a Path
Landmarks perform SMPC min computation over the shared link values
Landmarks perform SMPC min computation over the shared link values.
Landmarks perform SMPC min computation over the shared link values
Given enough “copies” of [x] it is possible to recover x for Alice
Transaction Execution
Transaction Execution

- Sequential friend-to-friend communication
Transaction Execution

- Sequential friend-to-friend communication
- Two-step transaction: on hold (or block) and settle
Transaction Execution

- Sequential friend-to-friend communication
- Two-step transaction: on hold (or block) and settle
- Example:

1. A person transfers 5 to another person.
2. The transaction goes through a bank, which holds it for 15 units of time.
3. The bank releases the transaction, and it settles with the other person in 20 units of time.
Transaction Execution

- Sequential friend-to-friend communication
- Two-step transaction: on hold (or block) and settle
- Example:
Transaction Execution

- Sequential friend-to-friend communication
- Two-step transaction: on hold (or block) and settle
- Example:

```
5

(5)
```

```
Incentive
```

```
10
```

```
20
```
Transaction Execution

- Sequential friend-to-friend communication
- Two-step transaction: on hold (or block) and settle
- Example:

![Example Diagram]

- 5
- (5)
- 10
- (5)
- 25
- Incentive
Transaction Execution

- Sequential friend-to-friend communication
- Two-step transaction: on hold (or block) and settle
- Example:

  ![Diagram showing a two-step transaction process with incentives and communication between users.](image)
Transaction Execution

- Sequential friend-to-friend communication
- Two-step transaction: on hold (or block) and settle
- Example:

![Diagram of transaction execution with arrows and actors]
Transaction Execution

- Sequential friend-to-friend communication
- Two-step transaction: on hold (or block) and settle
- Example:

```
5

Incentive

Ok, received!
```
Whispers: Characteristics/Limitations

- Distributed credit network transactions are possible without requiring:
  - a blockchain ledger
  - a proof-of-work

- Whispers can be modified by using landmarks as distributed stores
  [more details in the techreport]

- In case of disputes, this leaves task of proving links to the users

- It is blocking solution, and deadlocks are possibles

- In the near future
  - design non-blocking solutions in the asynchronous communication setting
    - distributed max-flow computation and atomic broadcast
In the Future
In the Future

✦ Payment Channels and lighting network
   https://lightning.network
   ✦ Designing distributed solutions for lighting network
In the Future

✦ Payment Channels and lighting network
   https://lightning.network
   ✦ Designing distributed solutions for lighting network

✦ The Interledger Protocol
   https://www.w3.org/community/interledger
   ✦ Several distributed/decentralized/centralized ledger solutions are coming up
   ✦ Performing transactions across different ledgers
Take home message
Take home message

- Credit networks have interesting properties and can be used in multiple scenarios

Why Credit Networks?

- Sybil-resistant applications
  - Introducing nodes is much easier than drawing trust from well-behaved nodes
  - Several applications:
    - Ostra: preventing e-mail spam [NSDI'08]
    - Bazaar: strengthening e-commerce [NSDI'11]
    - SumUp: Sybil-resilient content voting [NSDI'09]
    - Ripple: A real-life online settlement network
Credit networks have interesting properties and can be used in multiple scenarios.

Why Credit Networks?
- Sybil-resistant applications
- Introducing nodes is much easier than drawing trust from well-behaved nodes
- Several applications:
  - Ostra: preventing e-mail spam [NSDI’08]
  - Bazaar: strengthening e-commerce [NSDI’11]
  - SumUp: Sybil-resilient content voting [NSDI’09]
  - Ripple: A real-life online settlement network

Ledgers although provide accountability, it makes privacy a real problem in credit networks.

The tale of two Public Logs
<table>
<thead>
<tr>
<th>Bitcoin</th>
<th>Ripple</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input</td>
<td>Output</td>
</tr>
<tr>
<td>Alice-Bitcoin: 6 BTC</td>
<td>DR-Bitcoin: 6 BTC</td>
</tr>
</tbody>
</table>

Alice-Bitcoin

How to link these two events?
Take home message

- Credit networks have **interesting properties** and can be used in multiple scenarios

**Why Credit Networks?**

- Sybil-resistant applications
  - Well-behaved nodes
  - Sybil nodes

Introducing nodes is much easier than drawing trust from well-behaved nodes

- Several applications:
  - Ostra: preventing e-mail spam [NSDI'08]
  - Bazaar: strengthening e-commerce [NSDI'11]
  - SumUp: Sybil-resilient content voting [NSDI'09]
  - Ripple: A real-life online settlement network

Ledgers although provide accountability, it makes privacy a **real problem** in credit networks

**Privacy-preserving Credit in a Path**

- Landmarks perform SMPC min computation over the shared link values
- Given enough “copies” of [x] it is possible to recover x for Alice

**The tale of two Public Logs**

<table>
<thead>
<tr>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice-Bitcoin</td>
<td>DR-Bitcoin: 6 BTC</td>
</tr>
<tr>
<td>Alice</td>
<td>Alice-Ripple</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Sender</th>
<th>Receiver</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice-Bitcoin</td>
<td>Alice-Ripple</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Value</th>
<th>Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>6 BTC 10U</td>
<td>Bob → Alice</td>
</tr>
</tbody>
</table>

- How to link these two events?
Take home message

- Credit networks have interesting properties and can be used in multiple scenarios

Why Credit Networks?

- Sybil-resistant applications
  - Introducing nodes is much easier than drawing trust from well-behaved nodes
  - Several applications:
    - Ostra: preventing e-mail spam [NSDI’08]
    - Bazaar: strengthening e-commerce [NSDI’11]
    - SumUp: Sybil-resilient content voting [NSDI’09]
    - Ripple: A real-life online settlement network

Ledgers although provide accountability, it makes privacy a real problem in credit networks

Privacy-preserving Credit in a Path

- Landmarks perform SMPC min computation over the shared link values
- Given enough “copies” of [x] it is possible to recover x for Alice

Several questions remain unanswered leaving lots of open problems

**The tale of two Public Logs**

<table>
<thead>
<tr>
<th></th>
<th>Bitcoin</th>
<th>Ripple</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input</strong></td>
<td>Alice-Bitcoin: 6 BTC</td>
<td>Alice-Bitcoin: 6 BTC</td>
</tr>
<tr>
<td><strong>Output</strong></td>
<td>Alice-Ripple</td>
<td>DR-Ripple</td>
</tr>
</tbody>
</table>

**In the Future**

- Payment Channels and lighting network
  - https://lightning.network
- Designing distributed solutions for lighting network
- The Interledger Protocol
  - https://www.w3.org/community/interledger
- Several distributed/decentralized/centralized ledger solutions are coming up
- Performing transactions across different ledgers

- How to link these two events?