Decentralized Anonymous Micropayments

Alessandro Chiesa, Matthew Green, Jingcheng Liu, Peihan Miao, Ian Miers, Pratyush Mishra

http://eprint.iacr.org/2016/1033
Digital Payments

Customer → Merchant → Payment Network
Digital Payments

Customer → $ + $ → Merchant

Transaction fee
Transaction amount

Payment Network
Digital Payments

Customer → Merchant → Payment Network
Digital Payments

Customer → Merchant → Payment Network
Digital Payments

Supporting small payments is important for applications.
Digital Payments

Supporting small payments is important for applications.
Eg: payments instead of ads while browsing.
Digital Payments

Supporting small payments is important for applications. Eg: payments instead of ads while browsing.

Rich history of micropayment schemes constructions:
Digital Payments

Supporting small payments is important for applications. Eg: payments instead of ads while browsing.

Rich history of micropayment schemes constructions: [Whe96, Riv97, LO98, JY96, RS01, MR02]…
Digital Payments

Supporting small payments is important for applications. Eg: payments instead of ads while browsing.

Rich history of micropayment schemes constructions: [Whe96, Riv97, LO98, JY96, RS01, MR02]… … but no widespread deployments across multiple merchants.
Digital Payments

Supporting small payments is important for applications.
   Eg: payments instead of ads while browsing.

Rich history of micropayment schemes constructions:
   [Whe96, Riv97, LO98, JY96, RS01, MR02]…
   … but no widespread deployments across multiple merchants.

Potential reason: Prior systems required central mediator.
Digital Payments

Supporting small payments is important for applications. Eg: payments instead of ads while browsing.

Rich history of micropayment schemes constructions:
[Whe96, Riv97, LO98, JY96, RS01, MR02]…
… but no widespread deployments across multiple merchants.

Potential reason: Prior systems required central mediator. Why? Requires creating financial relations, meeting regulations, etc.
- Decentralized currency w/ quick adoption.
- No need to establish business relations between banks, merchants and regulators.
- To pay, just sign “from A to B: amt 4.3”.

Bitcoin
### Bitcoin

- Decentralized currency w/ quick adoption.
- No need to establish business relations between banks, merchants and regulators.
- To pay, just sign "from A to B: amt 4.3".

<table>
<thead>
<tr>
<th>From</th>
<th>To</th>
<th>Amt</th>
<th>Sign</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>M</td>
<td>10</td>
<td>σₐ</td>
</tr>
<tr>
<td>M</td>
<td>N</td>
<td>2.3</td>
<td>σₘ</td>
</tr>
</tbody>
</table>
Bitcoin

- Decentralized currency w/ quick adoption.
- No need to establish business relations between banks, merchants and regulators.
- To pay, just sign “from A to B: amt 4.3”.

<table>
<thead>
<tr>
<th>From</th>
<th>To</th>
<th>Amt</th>
<th>Sign</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>M</td>
<td>10</td>
<td>$\sigma_A$</td>
</tr>
<tr>
<td>M</td>
<td>N</td>
<td>2.3</td>
<td>$\sigma_M$</td>
</tr>
<tr>
<td>A</td>
<td>B</td>
<td>4.3</td>
<td>$\sigma_A$</td>
</tr>
</tbody>
</table>
Bitcoin

- Decentralized currency w/ quick adoption.
- No need to establish business relations between banks, merchants and regulators.
- To pay, just sign “from A to B: amt 4.3”.

Micropayments on Bitcoin?

<table>
<thead>
<tr>
<th>From</th>
<th>To</th>
<th>Amt</th>
<th>Sign</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>M</td>
<td>10</td>
<td>σₐ</td>
</tr>
<tr>
<td>M</td>
<td>N</td>
<td>2.3</td>
<td>σₘ</td>
</tr>
<tr>
<td>A</td>
<td>B</td>
<td>4.3</td>
<td>σₐ</td>
</tr>
</tbody>
</table>
Bitcoin

- Decentralized currency w/ quick adoption.
- No need to establish business relations between banks, merchants and regulators.
- To pay, just sign "from A to B: amt 4.3".

Micropayments on Bitcoin?

Problem 1: High Transaction fees
**LEDGER**

<table>
<thead>
<tr>
<th>From</th>
<th>To</th>
<th>Amt</th>
<th>Sign</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>M</td>
<td>10</td>
<td>$\sigma_A$</td>
</tr>
<tr>
<td>M</td>
<td>N</td>
<td>2.3</td>
<td>$\sigma_M$</td>
</tr>
<tr>
<td>A</td>
<td>B</td>
<td>4.3</td>
<td>$\sigma_A$</td>
</tr>
</tbody>
</table>

**Bitcoin**

- Decentralized currency with quick adoption.
- No need to establish business relations between banks, merchants, and regulators.
- To pay, just sign “from A to B: amt 4.3”.

**Micropayments on Bitcoin?**

**Problem 1: High Transaction fees**
- Projected to get higher.
Bitcoin

• Decentralized currency w/ quick adoption.
• No need to establish business relations between banks, merchants and regulators.
• To pay, just sign “from A to B: amt 4.3”.

Micropayments on Bitcoin?

Problem 1: High Transaction fees
• Projected to get higher.

Problem 2: Slow Confirmation time
Bitcoin

- Decentralized currency w/ quick adoption.
- No need to establish business relations between banks, merchants and regulators.
- To pay, just sign “from A to B: amt 4.3”.

Micropayments on Bitcoin?

Problem 1: High Transaction fees
- Projected to get higher.

Problem 2: Slow Confirmation time
- Bad for micropayment apps.
Bitcoin

- Decentralized currency w/ quick adoption.
- No need to establish business relations between banks, merchants and regulators.
- To pay, just sign “from A to B: amt 4.3”.

Micropayments on Bitcoin?

Problem 1: High Transaction fees
- Projected to get higher.

Problem 2: Slow Confirmation time
- Bad for micropayment apps.

Problem 3: Lack of Anonymity
**Bitcoin**

- Decentralized currency w/ quick adoption.
- No need to establish business relations between banks, merchants and regulators.
- To pay, just sign “from A to B: amt 4.3”.

**LEDGER**

<table>
<thead>
<tr>
<th>From</th>
<th>To</th>
<th>Amt</th>
<th>Sign</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>M</td>
<td>10</td>
<td>σ_A</td>
</tr>
<tr>
<td>M</td>
<td>N</td>
<td>2.3</td>
<td>σ_M</td>
</tr>
<tr>
<td>A</td>
<td>B</td>
<td>4.3</td>
<td>σ_A</td>
</tr>
</tbody>
</table>

**Micropayments on Bitcoin?**

**Problem 1: High Transaction fees**
- Projected to get higher.

**Problem 2: Slow Confirmation time**
- Bad for micropayment apps.

**Problem 3: Lack of Anonymity**
- Sender, receiver, amount are all public.
Bitcoin

- Decentralized currency w/ quick adoption.
- No need to establish business relations between banks, merchants and regulators.
- To pay, just sign “from A to B: amt 4.3”.

Micropayments on Bitcoin?

Problem 1: High Transaction fees
- Projected to get higher.

Problem 2: Slow Confirmation time
- Bad for micropayment apps.

Problem 3: Lack of Anonymity
- Sender, receiver, amount are all public.

Consequences:
Bitcoin

- Decentralized currency w/ quick adoption.
- No need to establish business relations between banks, merchants and regulators.
- To pay, just sign “from A to B: amt 4.3”.

Micropayments on Bitcoin?

**Problem 1: High Transaction fees**
- Projected to get higher.

**Problem 2: Slow Confirmation time**
- Bad for micropayment apps.

**Problem 3: Lack of Anonymity**
- Sender, receiver, amount are all public.

Consequences:
- No fungibility.
**Bitcoin**

- Decentralized currency w/ quick adoption.
- No need to establish business relations between banks, merchants and regulators.
- To pay, just sign “from A to B: amt 4.3”.

---

<table>
<thead>
<tr>
<th>From</th>
<th>To</th>
<th>Amt</th>
<th>Sign</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>M</td>
<td>10</td>
<td>$\sigma_A$</td>
</tr>
<tr>
<td>M</td>
<td>N</td>
<td>2.3</td>
<td>$\sigma_M$</td>
</tr>
<tr>
<td>A</td>
<td>B</td>
<td>4.3</td>
<td>$\sigma_A$</td>
</tr>
</tbody>
</table>

---

**Micropayments on Bitcoin?**

**Problem 1: High Transaction fees**
- Projected to get higher.

**Problem 2: Slow Confirmation time**
- Bad for micropayment apps.

**Problem 3: Lack of Anonymity**
- Sender, receiver, amount are all public.

Consequences:
- No fungibility.
- No privacy. (especially bad for micropayment apps)
Bitcoin

- Decentralized currency w/ quick adoption.
- No need to establish business relations between banks, merchants and regulators.
- To pay, just sign “from A to B: amt 4.3”.

Micropayments on Bitcoin?

Problem 3: Lack of Anonymity
- Sender, receiver, amount are all public.

Consequences:
- No fungibility.
- No privacy. (especially bad for micropayment apps)
### Bitcoin

- Decentralized currency w/ quick adoption.
- No need to establish business relations between banks, merchants and regulators.
- To pay, just sign “from A to B: amt 4.3”.

<table>
<thead>
<tr>
<th>From</th>
<th>To</th>
<th>Amt</th>
<th>Sign</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>M</td>
<td>10</td>
<td>$\sigma_A$</td>
</tr>
<tr>
<td>M</td>
<td>N</td>
<td>2.3</td>
<td>$\sigma_M$</td>
</tr>
<tr>
<td>A</td>
<td>B</td>
<td>4.3</td>
<td>$\sigma_A$</td>
</tr>
</tbody>
</table>

### Micropayments on Bitcoin?

**Pass-Shelat** *(CCS 2015)*

**Problem 3: Lack of Anonymity**

- Sender, receiver, amount are all public.

**Consequences:**

- No fungibility.
- No privacy. (especially bad for micropayment apps)
Bitcoin

- Decentralized currency w/ quick adoption.
- No need to establish business relations between banks, merchants and regulators.
- To pay, just sign “from A to B: amt 4.3”.

Micropayments on Bitcoin?

Pass-Shelat (CCS 2015)

- Probabilistic payments for Bitcoin.

Problem 3: Lack of Anonymity

- Sender, receiver, amount are all public.

Consequences:
- No fungibility.
- No privacy. (especially bad for micropayment apps)
Bitcoin

• Decentralized currency w/ quick adoption.
• No need to establish business relations between banks, merchants and regulators.
• To pay, just sign “from A to B: amt 4.3”.

Micropayments on Bitcoin?

Pass-Shelat (CCS 2015)

• Probabilistic payments for Bitcoin.
• Solves problem 1: Amortized tx fee.

Problem 3: Lack of Anonymity

• Sender, receiver, amount are all public.

Consequences:
• No fungibility.
• No privacy. (especially bad for micropayment apps)
Bitcoin

- Decentralized currency w/ quick adoption.
- No need to establish business relations between banks, merchants and regulators.
- To pay, just sign “from A to B: amt 4.3”.

Micropayments on Bitcoin?

Pass-Shelat (CCS 2015)

- Probabilistic payments for Bitcoin.
- Solves problem 1: Amortized tx fee.
- Solves problem 2: Quick confirmation.

Problem 3: Lack of Anonymity

- Sender, receiver, amount are all public.

Consequences:
- No fungibility.
- No privacy. (especially bad for micropayment apps)
**Bitcoin**

- Decentralized currency w/ quick adoption.
- No need to establish business relations between banks, merchants and regulators.
- To pay, just sign “from A to B: amt 4.3”.

---

**Micropayments on Bitcoin?**

**Pass-Shelat (CCS 2015)**

- Probabilistic payments for Bitcoin.
- **Solves problem 1**: Amortized tx fee.
- **Solves problem 2**: Quick confirmation.

**Zerocash (Oakland 2014)**

- Anonymous Bitcoin-like currency.
- **Solves problem 3**: Hides sender, receiver and amount.
Goal
Goal

**micropayments** that are:
Goal

**micropayments** that are:

**decentralized** (for ease of deployment),
Goal

**micropayments** that are:

decentralized (for ease of deployment),
anonymous  (for fungibility, etc.), and
Goal

**micropayments** that are:

**decentralized** (for ease of deployment),

**anonymous** (for fungibility, etc.), and

**offline** (for fast response).
Goal

micropayments that are:
decentralized (for ease of deployment),
anonymous (for fungibility, etc.), and
offline (for fast response).

Contributions
Goal

micropayments that are:
decentralized (for ease of deployment),
anonymous (for fungibility, etc.), and
offline (for fast response).

Contributions

1. Definition of cryptographic primitive via ideal functionality.
Goal

micropayments that are:
decentralized (for ease of deployment),
anonymous (for fungibility, etc.), and
offline (for fast response).

Contributions

1. Definition of cryptographic primitive via ideal functionality.
2. Construction under standard crypto assumptions.
Goal

**micropayments** that are:

- **decentralized** (for ease of deployment),
- **anonymous** (for fungibility, etc.), and
- **offline** (for fast response).

Contributions

1. Definition of **cryptographic primitive** via **ideal functionality**.
2. **Construction** under **standard crypto assumptions**.
3. Techniques: we use two tools:
Goal

micropayments that are:
decentralized (for ease of deployment),
anonymous (for fungibility, etc.), and
offline (for fast response).

Contributions

1. Definition of cryptographic primitive via ideal functionality.

2. Construction under standard crypto assumptions.

3. Techniques: we use two tools:
   - translucent crypto: new fractional message transfer protocol (probabilistic)
Goal

**micropayments** that are:
- **decentralized** (for ease of deployment),
- **anonymous** (for fungibility, etc.), and
- **offline** (for fast response).

Contributions

1. Definition of **cryptographic primitive** via **ideal functionality**.
2. **Construction** under **standard crypto assumptions**.
3. Techniques: we use two tools:
   - **translucent crypto**: new **fractional message transfer** protocol.
     (probabilistic)
   - **game theory**: characterization of double-spending.
Probabilistic Payments
Probabilistic Payments

Alice "pays" Bob $0.01
Probabilistic Payments

Alice "pays" Bob $0.01

$1
Probabilistic Payments

Alice "pays" Bob $0.01

$1
Probabilistic Payments

Alice "pays" Bob $0.01
Probabilistic Payments

Alice "pays" Bob $0.01

w.p. 99/100

$1
Probabilistic Payments

Alice "pays" Bob $0.01 with probability 99/100, or $1 with probability 1/100.
Probabilistic Payments

Alice "pays" Bob $0.01 with a probability of 99/100.

nullpayment (Alice wins)
Probabilistic Payments

Alice "pays" Bob $0.01 w.p. 1/100

nullpayment (Alice wins)
Probabilistic Payments

Alice "pays" Bob $0.01

$1

w.p. 99/100

$1

w.p. 1/100

nullpayment
(Alice wins)
Probabilistic Payments

Alice "pays" Bob $0.01

$1

w.p. 99/100

Alice wins

$1

w.p. 1/100

Bob wins

macropayment

nullpayment

$1

$1
Probabilistic Payments

Alice "pays" Bob $0.01

$1

w.p. 99/100

nullpayment
(Alice wins)

$1

w.p. 1/100

macropayment
(Bob wins)

$1

Probabilistic payments imply micropayments:
Probabilistic Payments

Alice "pays" Bob $0.01

$1

w.p. 99/100

nullpayment (Alice wins)

$1

w.p. 1/100

macropayment (Bob wins)

$1

Probabilistic payments imply micropayments:

Transaction fee is amortized over many payments.
Probabilistic Payments

Probabilistic payments imply micropayments:

Transaction fee is amortized over many payments.

Nullpayments are offline and do not require interaction with payment network.
Building Blocks

Pass-Shelat

Zerocash
Building Blocks

Pass-Shelat
coin-flipping + Bitcoin

Zerocash
Building Blocks

Pass-Shelat
coin-flipping + Bitcoin

Zerocash

<table>
<thead>
<tr>
<th>Ledger</th>
</tr>
</thead>
<tbody>
<tr>
<td>From</td>
</tr>
<tr>
<td>⋮</td>
</tr>
<tr>
<td>M</td>
</tr>
<tr>
<td>A</td>
</tr>
</tbody>
</table>
Pass-Shelat

coin-flipping + Bitcoin

1. Alice escrows $v$.

<table>
<thead>
<tr>
<th>From</th>
<th>To</th>
<th>Amt</th>
<th>Sign</th>
</tr>
</thead>
<tbody>
<tr>
<td>M</td>
<td>N</td>
<td>2.3</td>
<td>$\sigma_M$</td>
</tr>
<tr>
<td>A</td>
<td>M</td>
<td>10</td>
<td>$\sigma_A$</td>
</tr>
<tr>
<td>A</td>
<td>E</td>
<td>4.3</td>
<td>$\sigma_A$</td>
</tr>
</tbody>
</table>

Zerocash

coin-flipping + Bitcoin
Building Blocks

Pass-Shelat
coin-flipping + Bitcoin

1. Alice escrows $v$.
2. Alice and Bob engage in coin-flip.

<p>| Ledger          |
|-----------------|-----------------|</p>
<table>
<thead>
<tr>
<th><strong>From</strong></th>
<th><strong>To</strong></th>
<th><strong>Amt</strong></th>
<th><strong>Sign</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>M</td>
<td>N</td>
<td>2.3</td>
<td>$\sigma_M$</td>
</tr>
<tr>
<td>A</td>
<td>M</td>
<td>10</td>
<td>$\sigma_A$</td>
</tr>
<tr>
<td>A</td>
<td>E</td>
<td>4.3</td>
<td>$\sigma_A$</td>
</tr>
</tbody>
</table>

Zerocash

coin-flipping + Bitcoin
Building Blocks

Pass-Shelat
coin-flipping + Bitcoin

1. Alice escrows $v$.
2. Alice and Bob engage in coin-flip.
3. If Alice wins: she can reuse escrow.

<table>
<thead>
<tr>
<th>From</th>
<th>To</th>
<th>Amt</th>
<th>Sign</th>
</tr>
</thead>
<tbody>
<tr>
<td>M</td>
<td>N</td>
<td>2.3</td>
<td>$\sigma_M$</td>
</tr>
<tr>
<td>A</td>
<td>M</td>
<td>10</td>
<td>$\sigma_A$</td>
</tr>
<tr>
<td>A</td>
<td>E</td>
<td>4.3</td>
<td>$\sigma_A$</td>
</tr>
</tbody>
</table>

Zerocash

coin-flipping + Bitcoin
Building Blocks

Pass-Shelat

coin-flipping + Bitcoin

1. Alice escrows $v$.
2. Alice and Bob engage in coin-flip.
3. If Alice wins: she can reuse escrow.
4. If Bob wins: he gets $v$.

<table>
<thead>
<tr>
<th>From</th>
<th>To</th>
<th>Amt</th>
<th>Sign</th>
</tr>
</thead>
<tbody>
<tr>
<td>M</td>
<td>N</td>
<td>2.3</td>
<td>$\sigma_M$</td>
</tr>
<tr>
<td>A</td>
<td>M</td>
<td>10</td>
<td>$\sigma_A$</td>
</tr>
<tr>
<td>A</td>
<td>E</td>
<td>4.3</td>
<td>$\sigma_A$</td>
</tr>
<tr>
<td>E</td>
<td>B</td>
<td>4.3</td>
<td>$\sigma_E$</td>
</tr>
</tbody>
</table>

Zerocash

coin-flipping + Bitcoin
Pass-Shelat
coin-flipping + Bitcoin

1. Alice escrows \( v \).
2. Alice and Bob engage in coin-flip.
3. If Alice wins: she can reuse escrow.
4. If Bob wins: he gets \( v \).

<table>
<thead>
<tr>
<th>From</th>
<th>To</th>
<th>Amt</th>
<th>Sign</th>
</tr>
</thead>
<tbody>
<tr>
<td>M</td>
<td>N</td>
<td>2.3</td>
<td>( \sigma_M )</td>
</tr>
<tr>
<td>A</td>
<td>M</td>
<td>10</td>
<td>( \sigma_A )</td>
</tr>
<tr>
<td>A</td>
<td>E</td>
<td>4.3</td>
<td>( \sigma_A )</td>
</tr>
<tr>
<td>E</td>
<td>B</td>
<td>4.3</td>
<td>( \sigma_E )</td>
</tr>
</tbody>
</table>

Zerocash
zero knowledge proofs + Bitcoin

Building Blocks
Building Blocks

Pass-Shelat
coin-flipping + Bitcoin

1. Alice escrows $v$.
2. Alice and Bob engage in coin-flip.
3. If Alice wins: she can reuse escrow.
4. If Bob wins: he gets $v$.

<table>
<thead>
<tr>
<th>Ledger</th>
<th>From</th>
<th>To</th>
<th>Amt</th>
<th>Sign</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>M</td>
<td>N</td>
<td>2.3</td>
<td>$\sigma_M$</td>
</tr>
<tr>
<td></td>
<td>A</td>
<td>M</td>
<td>10</td>
<td>$\sigma_A$</td>
</tr>
<tr>
<td></td>
<td>A</td>
<td>E</td>
<td>4.3</td>
<td>$\sigma_A$</td>
</tr>
<tr>
<td></td>
<td>E</td>
<td>B</td>
<td>4.3</td>
<td>$\sigma_E$</td>
</tr>
</tbody>
</table>

ZeroCash
zero knowledge proofs + Bitcoin

<table>
<thead>
<tr>
<th>Ledger</th>
<th>Old</th>
<th>New</th>
<th>Proof</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>8436378</td>
<td>cm$_1$</td>
<td>$\pi_1$</td>
</tr>
<tr>
<td></td>
<td>6327690</td>
<td>cm$_2$</td>
<td>$\pi_2$</td>
</tr>
</tbody>
</table>

pk$_A$, sk$_A$

pk$_B$, sk$_B$
Building Blocks

Pass-Shelat
coin-flipping + Bitcoin

1. Alice escrows $v$.
2. Alice and Bob engage in coin-flip.
3. If Alice wins: she can reuse escrow.
4. If Bob wins: he gets $v$.

<table>
<thead>
<tr>
<th>Ledger</th>
</tr>
</thead>
<tbody>
<tr>
<td>From</td>
</tr>
<tr>
<td>-------</td>
</tr>
<tr>
<td>M</td>
</tr>
<tr>
<td>A</td>
</tr>
<tr>
<td>A</td>
</tr>
<tr>
<td>E</td>
</tr>
</tbody>
</table>

Zerocash
zero knowledge proofs + Bitcoin

1. Alice owns coin $c_1$ with comm $cm_1$.

<table>
<thead>
<tr>
<th>Ledger</th>
</tr>
</thead>
<tbody>
<tr>
<td>Old</td>
</tr>
<tr>
<td>------</td>
</tr>
<tr>
<td>8436378</td>
</tr>
<tr>
<td>6327690</td>
</tr>
</tbody>
</table>

$pk_A, sk_A$

$pk_B, sk_B$
Building Blocks

Pass-Shelat

1. Alice escrows \( v \).
2. Alice and Bob engage in coin-flip.
3. If Alice wins: she can reuse escrow.
4. If Bob wins: he gets \( v \).

<table>
<thead>
<tr>
<th>Ledger</th>
</tr>
</thead>
<tbody>
<tr>
<td>From</td>
</tr>
<tr>
<td>M</td>
</tr>
<tr>
<td>A</td>
</tr>
<tr>
<td>A</td>
</tr>
<tr>
<td>E</td>
</tr>
</tbody>
</table>

Zerocash

to knowledge proofs + Bitcoin

1. Alice owns coin \( c_1 \) with comm \( cm_1 \).
2. To pay Bob, Alice:

<table>
<thead>
<tr>
<th>Ledger</th>
</tr>
</thead>
<tbody>
<tr>
<td>Old</td>
</tr>
<tr>
<td>8436378</td>
</tr>
<tr>
<td>6327690</td>
</tr>
</tbody>
</table>
Building Blocks

Pass-Shelat
coin-flipping + Bitcoin

1. Alice escrows $v$.
2. Alice and Bob engage in coin-flip.
3. If Alice wins: she can reuse escrow.
4. If Bob wins: he gets $v$.

<table>
<thead>
<tr>
<th>From</th>
<th>To</th>
<th>Amt</th>
<th>Sign</th>
</tr>
</thead>
<tbody>
<tr>
<td>M</td>
<td>N</td>
<td>2.3</td>
<td>$\sigma_M$</td>
</tr>
<tr>
<td>A</td>
<td>M</td>
<td>10</td>
<td>$\sigma_A$</td>
</tr>
<tr>
<td>A</td>
<td>E</td>
<td>4.3</td>
<td>$\sigma_A$</td>
</tr>
<tr>
<td>E</td>
<td>B</td>
<td>4.3</td>
<td>$\sigma_E$</td>
</tr>
</tbody>
</table>

Zerocash
zero knowledge proofs + Bitcoin

1. Alice owns coin $c_1$ with comm $cm_1$.
2. To pay Bob, Alice:
   a) derives $sn_1$ from $c_1$ and $sk_A$.

<table>
<thead>
<tr>
<th>Old</th>
<th>New</th>
<th>Proof</th>
</tr>
</thead>
<tbody>
<tr>
<td>8436378</td>
<td>$cm_1$</td>
<td>$\pi_1$</td>
</tr>
<tr>
<td>6327690</td>
<td>$cm_2$</td>
<td>$\pi_2$</td>
</tr>
</tbody>
</table>
Building Blocks

Pass-Shelat

coin-flipping + Bitcoin

1. Alice escrows $v$.
2. Alice and Bob engage in coin-flip.
3. If Alice wins: she can reuse escrow.
4. If Bob wins: he gets $v$.

<table>
<thead>
<tr>
<th>Ledger</th>
</tr>
</thead>
<tbody>
<tr>
<td>From</td>
</tr>
<tr>
<td>-------</td>
</tr>
<tr>
<td>$\cdots$</td>
</tr>
<tr>
<td>M</td>
</tr>
<tr>
<td>A</td>
</tr>
<tr>
<td>A</td>
</tr>
<tr>
<td>E</td>
</tr>
</tbody>
</table>

Zerocash

zero knowledge proofs + Bitcoin

1. Alice owns coin $c_1$ with comm $cm_1$.
2. To pay Bob, Alice:
   a) derives $sn_1$ from $c_1$ and $sk_A$.
   b) creates new coin $c_3$ with comm $cm_3$.

<table>
<thead>
<tr>
<th>Ledger</th>
</tr>
</thead>
<tbody>
<tr>
<td>Old</td>
</tr>
<tr>
<td>-------</td>
</tr>
<tr>
<td>$\cdots$</td>
</tr>
<tr>
<td>$8436378$</td>
</tr>
<tr>
<td>$6327690$</td>
</tr>
</tbody>
</table>

$pk_A$, $sk_A$, $sn_1$, $pk_B$, $sk_B$
Pass-Shelat
coin-flipping + Bitcoin

1. Alice escrows $v$.
2. Alice and Bob engage in coin-flip.
3. If Alice wins: she can reuse escrow.
4. If Bob wins: he gets $v$.

<table>
<thead>
<tr>
<th>Ledger</th>
<th>From</th>
<th>To</th>
<th>Amt</th>
<th>Sign</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>M</td>
<td>N</td>
<td>2.3</td>
<td>$\sigma_M$</td>
</tr>
<tr>
<td></td>
<td>A</td>
<td>M</td>
<td>10</td>
<td>$\sigma_A$</td>
</tr>
<tr>
<td></td>
<td>A</td>
<td>E</td>
<td>4.3</td>
<td>$\sigma_A$</td>
</tr>
<tr>
<td></td>
<td>E</td>
<td>B</td>
<td>4.3</td>
<td>$\sigma_E$</td>
</tr>
</tbody>
</table>

Zerocash
zero knowledge proofs + Bitcoin

1. Alice owns coin $c_1$ with comm $cm_1$.
2. To pay Bob, Alice:
   a) derives $s_{n_1}$ from $c_1$ and $sk_A$.
   b) creates new coin $c_3$ with comm $cm_3$.
   c) creates ZK proof $\pi_3$ for above.

<table>
<thead>
<tr>
<th>Ledger</th>
<th>Old</th>
<th>New</th>
<th>Proof</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>8436378</td>
<td>cm_1</td>
<td>$\pi_1$</td>
</tr>
<tr>
<td></td>
<td>6327690</td>
<td>cm_2</td>
<td>$\pi_2$</td>
</tr>
</tbody>
</table>
Building Blocks

Pass-Shelat
coin-flipping + Bitcoin

1. Alice escrows $v$.
2. Alice and Bob engage in coin-flip.
3. If Alice wins: she can reuse escrow.
4. If Bob wins: he gets $v$.

<table>
<thead>
<tr>
<th>From</th>
<th>To</th>
<th>Amt</th>
<th>Sign</th>
</tr>
</thead>
<tbody>
<tr>
<td>M</td>
<td>N</td>
<td>2.3</td>
<td>$\sigma_M$</td>
</tr>
<tr>
<td>A</td>
<td>M</td>
<td>10</td>
<td>$\sigma_A$</td>
</tr>
<tr>
<td>A</td>
<td>E</td>
<td>4.3</td>
<td>$\sigma_A$</td>
</tr>
<tr>
<td>E</td>
<td>B</td>
<td>4.3</td>
<td>$\sigma_E$</td>
</tr>
</tbody>
</table>

Zerocash
zero knowledge proofs + Bitcoin

1. Alice owns coin $c_1$ with comm $cm_1$.
2. To pay Bob, Alice:
   a) derives $sn_1$ from $c_1$ and $sk_A$.
   b) creates new coin $c_3$ with comm $cm_3$.
   c) creates ZK proof $\pi_3$ for above.
   d) appends $tx = (sn_1, cm_3, \pi_3)$.

<table>
<thead>
<tr>
<th>Old</th>
<th>New</th>
<th>Proof</th>
</tr>
</thead>
<tbody>
<tr>
<td>8436378</td>
<td>$cm_1$</td>
<td>$\pi_1$</td>
</tr>
<tr>
<td>6327690</td>
<td>$cm_2$</td>
<td>$\pi_2$</td>
</tr>
<tr>
<td>$sn_1$</td>
<td>$cm_3$</td>
<td>$\pi_3$</td>
</tr>
</tbody>
</table>
Building Blocks

Pass-Shelat

coin-flipping + Bitcoin

1. Alice escrows $v$.
2. Alice and Bob engage in coin-flip.
3. If Alice wins: she can reuse escrow.
4. If Bob wins: he gets $v$.

<table>
<thead>
<tr>
<th>From</th>
<th>To</th>
<th>Amt</th>
<th>Sign</th>
</tr>
</thead>
<tbody>
<tr>
<td>M</td>
<td>N</td>
<td>2.3</td>
<td>$\sigma_M$</td>
</tr>
<tr>
<td>A</td>
<td>M</td>
<td>10</td>
<td>$\sigma_A$</td>
</tr>
<tr>
<td>A</td>
<td>E</td>
<td>4.3</td>
<td>$\sigma_A$</td>
</tr>
<tr>
<td>E</td>
<td>B</td>
<td>4.3</td>
<td>$\sigma_E$</td>
</tr>
</tbody>
</table>

Zerocash

zero knowledge proofs + Bitcoin

1. Alice owns coin $c_1$ with comm $cm_1$.
2. To pay Bob, Alice:
   a) derives $sn_1$ from $c_1$ and $sk_A$.
   b) creates new coin $c_3$ with comm $cm_3$.
   c) creates ZK proof $\pi_3$ for above.
   d) appends $tx = (sn_1, cm_3, \pi_3)$.

Cannot link $sn_1$ with $cm_1$ without $sk_A$.
Naive Attempt: PS + Zerocash
Naive Attempt: PS + Zerocash

<table>
<thead>
<tr>
<th>Ledger</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Old</td>
<td>New</td>
<td>Proof</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8436378 cm</td>
<td>CM₁</td>
<td>π₁</td>
<td></td>
</tr>
<tr>
<td>6327690 cm</td>
<td>CM₂</td>
<td>π₂</td>
<td></td>
</tr>
</tbody>
</table>
1. Alice escrows $v$ in a Zerocash transaction.

<table>
<thead>
<tr>
<th>Ledger</th>
</tr>
</thead>
<tbody>
<tr>
<td>Old</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>8436378</td>
</tr>
<tr>
<td>6327690</td>
</tr>
<tr>
<td>$SN_1$</td>
</tr>
</tbody>
</table>
Naive Attempt: PS + Zerocash

1. Alice escrows $v$ in a Zerocash transaction.
2. Alice and Bob engage in coin-flip.

<table>
<thead>
<tr>
<th>Ledger</th>
<th>Old</th>
<th>New</th>
<th>Proof</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$8436378$</td>
<td>$CM_1$</td>
<td>$\pi_1$</td>
</tr>
<tr>
<td></td>
<td>$6327690$</td>
<td>$CM_2$</td>
<td>$\pi_2$</td>
</tr>
<tr>
<td></td>
<td>$SN_1$</td>
<td>$CM_3$</td>
<td>$\pi_3$</td>
</tr>
</tbody>
</table>
Naive Attempt: PS + Zerocash

1. Alice escrows $v$ in a Zerocash transaction.
2. Alice and Bob engage in coin-flip.
3. If Alice wins: she can reuse escrow.

<table>
<thead>
<tr>
<th>Ledger</th>
<th>Old</th>
<th>New</th>
<th>Proof</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>⋮</td>
<td></td>
<td>⋮</td>
</tr>
<tr>
<td></td>
<td>8436378</td>
<td>CM₁</td>
<td>$\pi₁$</td>
</tr>
<tr>
<td></td>
<td>6327690</td>
<td>CM₂</td>
<td>$\pi₂$</td>
</tr>
<tr>
<td></td>
<td>SN₁</td>
<td>CM₃</td>
<td>$\pi₃$</td>
</tr>
</tbody>
</table>
Naive Attempt: PS + Zerocash

1. Alice escrows $v$ in a Zerocash transaction.
2. Alice and Bob engage in coin-flip.
3. If Alice wins: she can reuse escrow.
4. If Bob wins: he gets $v$.  

<table>
<thead>
<tr>
<th>Ledger</th>
<th>8436378</th>
<th>$\text{CM}_1$</th>
<th>$\pi_1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Old</td>
<td>6327690</td>
<td>$\text{CM}_2$</td>
<td>$\pi_2$</td>
</tr>
<tr>
<td>New</td>
<td>$S\text{N}_1$</td>
<td>$\text{CM}_3$</td>
<td>$\pi_3$</td>
</tr>
<tr>
<td>Proof</td>
<td>$S\text{N}_3$</td>
<td>$\text{CM}_4$</td>
<td>$\pi_4$</td>
</tr>
</tbody>
</table>
Naive Attempt: PS + Zerocash

1. Alice escrows $v$ in a Zerocash transaction.
2. Alice and Bob engage in coin-flip.
3. If Alice wins: she can reuse escrow.
4. If Bob wins: he gets $v$.

**Major Issues:**

- **Linkability**
- **Double Spending**
Problem 1: Linkability
Problem 1: Linkability

- To amortize transaction fees, Alice has to reuse escrow.
- Bob *always* learns serial number of escrowed coin.
Problem 1: Linkability

- To amortize transaction fees, Alice has to reuse escrow.
- Bob **always** learns serial number of escrowed coin.
  - Can track Alice when she spends coin w/ others.
Problem 1: Linkability

- To amortize transaction fees, Alice has to reuse escrow.
- Bob **always** learns serial number of escrowed coin.
  - Can track Alice when she spends coin w/ others.
Problem 1: Linkability

- To amortize transaction fees, Alice has to reuse escrow.
- Bob **always** learns serial number of escrowed coin.
  - Can track Alice when she spends coin w/ others.
Problem 1: Linkability

- To amortize transaction fees, Alice has to reuse escrow.
- Bob *always* learns serial number of escrowed coin.
  - Can track Alice when she spends coin w/ others.
- Further attacks lead to loss of most privacy.
Solution: Make \textit{sn} translucent
Solution: Make sn translucent
Solution: Make \textit{sn} translucent

1. Creates tx, but doesn't append to ledger. Instead, commits to it and generates ZK proof of correctness.

\begin{itemize}
  \item \texttt{tx1}
  \item \texttt{tx2}
\end{itemize}

\texttt{c = COMM(tx_3)}
Solution: Make sn translucent

1. Creates tx, but doesn't append to ledger. Instead, commits to it and generates ZK proof of correctness.

\[ c = \text{COMM}(tx_3) \]

2. Sends commitment & proof to Bob.

\[ c, \pi \]
Solution: Make **sn** translucent

1. Creates tx, but doesn't append to ledger. Instead, commits to it and generates ZK proof of correctness.

   \[ c = \text{COMM}(tx_3) \]

2. Sends commitment & proof to Bob.

3. Alice and Bob attempt to open the commitment probabilistically.

**Ledger**

\[ \vdots \]

\[ tx_1 \]

\[ tx_2 \]
Solution: Make sn translucent

1. Creates tx, but doesn't append to ledger. Instead, commits to it and generates ZK proof of correctness.

2. Sends commitment & proof to Bob.

\[ c = \text{COMM}(tx_3) \]

3. Alice and Bob attempt to open the commitment probabilistically.

Nullpayment: Alice can spend coin again, but Bob learns nothing about the coin!
Solution: Make $sn$ translucent

1. Creates tx, but doesn't append to ledger. Instead, commits to it and generates ZK proof of correctness.

$c = \text{COMM}(tx_3)$

2. Sends commitment & proof to Bob.

$c, \pi$

prob. opening

1-$p$

3. Alice and Bob attempt to open the commitment probabilistically.

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$

$\pi$
Solution: Make *sn* translucent

1. Creates tx, but doesn’t append to ledger. Instead, commits to it and generates ZK proof of correctness.

2. Sends commitment & proof to Bob.

```latex
\text{c = COMM(tx_3)}
```

3. Alice and Bob attempt to open the commitment probabilistically.

**Nullpayment:** Alice can spend coin again, but Bob learns nothing about the coin!

**Macropayment:** Bob gets tx and learns serial number.

Ledger:
- \( \vdots \)
- \( tx_1 \)
- \( tx_2 \)
- \( tx_3 \)
Solution: Make sn translucent

Fractional Message Transfer

**Fractional hiding:** w.p. $1-p$, Bob learns nothing about message.

**Fractional binding:** Bob can always open with probability $p$.

1. Creates tx, but doesn’t append to ledger. Instead, commits to it and generates ZK proof of correctness.

Nullpayment: Alice can spend coin again, but Bob learns nothing about the coin!

Macropayment: Bob gets tx and learns serial number.

2. Sends commitment & proof to Bob.

3. Alice and Bob attempt to open the commitment probabilistically.
Solution: Make sn translucent

**Fractional Message Transfer**

**Fractional hiding**: w.p. 1-\(p\), Bob learns nothing about message.

**Fractional binding**: Bob can always open with probability \(p\).

1. Creates tx, but doesn't append to ledger. Instead, commits to it and generates ZK proof of correctness.

Wants fractional hiding

Nullpayment: Alice can spend coin again, but Bob learns nothing about the coin!

Macropayment: Bob gets tx and learns serial number.

3. Alice and Bob attempt to open the commitment probabilistically.
Solution: Make sn translucent

Fractional Message Transfer

Fractional binding: Bob can always open with probability p.

Wants fractional hiding

Nullpayment: Alice can spend coin again, but Bob learns nothing about the coin!

Wants fractional binding

Macropayment: Bob gets tx and learns serial number.

1. Creates tx, but doesn’t append to ledger. Instead, commits to it and generates ZK proof of correctness.
Problem 2: Double-Spending
Problem 2: Double-Spending

Malice can use the same coin in multiple payments in parallel.
Problem 2: Double-Spending

Malice can use the same coin in multiple payments *in parallel.*
Problem 2: Double-Spending

Malice can use the same coin in multiple payments \textit{in parallel}.
Problem 2: Double-Spending

Malice can use the same coin in multiple payments in parallel.
Problem 2: Double-Spending

Malice can use the same coin in multiple payments in parallel.
Problem 2: Double-Spending

Malice can use the same coin in multiple payments in parallel.
Problem 2: Double-Spending

Malice can use the same coin in multiple payments in parallel.

Offline setting ⇒ such attacks cannot be prevented.
Solution: deposits + rationality
Solution: deposits + rationality

Ledger

⋮

\[ \text{tx}_1 \]
\[ \text{tx}_2 \]
Solution: deposits + rationality

1. Before any probabilistic payments, Alice creates a deposit coin.
Solution: deposits + rationality

1. Before any probabilistic payments, Alice creates a deposit coin.

2. Commitment also contains secret share of the deposit $s_n$.

$$c = \text{COMM}(t_{\text{mp}}, s_{1})$$
Solution: deposits + rationality

1. Before any probabilistic payments, Alice creates a deposit coin.

2. Commitment also contains secret share of the deposit sn.

3. Also proves deposit is valid & secret share is correct.

\[ c = \text{COMM}(tx_{mp}, ss_1) \]
Solution: deposits + rationality

1. Before any probabilistic payments, Alice creates a deposit coin.

2. Commitment also contains secret share of the deposit sn.

3. Also proves deposit is valid & secret share is correct.

\[ c = \text{COMM}(t_{\text{mp}}, ss_1) \]
Solution: deposits + rationality

1. Before any probabilistic payments, Alice creates a deposit coin.

2. Commitment also contains secret share of the deposit sn.

3. Also proves deposit is valid & secret share is correct.

4. If Bob wins, he gets \((tx_{mp}, ss_1)\) and he posts this to the ledger.
Solution: deposits + rationality

1. Before any probabilistic payments, Alice creates a deposit coin.

2. Commitment also contains secret share of the deposit sn.

3. Also proves deposit is valid & secret share is correct.

4. If Bob wins, he gets \((t\text{x}_{mp}, s\text{s}_1)\) and he posts this to the ledger.
Why does this work?
Why does this work?
Why does this work?

Ledger

⋮

\( tx_1 \)

\( tx_2 \)
Why does this work?

Ledger

⋮

\(\text{tx}_1\)

\(\text{tx}_2\)

\[\text{prob. payment}\]
Why does this work?

1. Macropayment

Ledger:

- \( \_ \_ \)
- \( tx_1 \)
- \( tx_2 \)
- \( tx_{mp}, SS_1 \)

\( tx_{mp}, SS_1 \)
Why does this work?

1. Macropayment

Ledger
- ...
- tx₁
- tx₂
- tx_{mp, SS₁}

prob. payment

tx_{mp, SS₁}
Why does this work?

1. Macropayment

Ledger:

- \( \cdots \)
- \( \text{tx}_1 \)
- \( \text{tx}_2 \)
- \( \text{tx}_{mp, SS_1} \)
Why does this work?

1. Macropayment

2. Macropayment again!
Why does this work?

1. Macropayment

2. Macropayment again!

\[ \text{sn}_{\text{dep}} = \text{ss}_{1} + \text{ss}_{2} \]
Why does this work?

1. Macropayment

2. Macropayment again!

-∞ utility!

SS₁ + SS₂ = S_{dep}
So far
So far

Probabilistic opening:
So far

Probabilistic opening:
Deposits:
So far

Probabilistic opening: prevents linkability.
Deposits:
So far

Probabilistic opening: prevents linkability.
Deposits: prevent double-spending.
So far

Probabilistic opening: prevents linkability.
Deposits: prevent double-spending.

Are we done?
So far

Probabilistic opening: prevents linkability.
Deposits: prevent double-spending.

Are we done?

Functionality:
So far

Probabilistic opening: prevents linkability.
Deposits: prevent double-spending.

Are we done?

---

**Functionality:**
*Feature:* Customers should be able to withdraw deposits.
So far

Probabilistic opening: prevents linkability.
Deposits: prevent double-spending.

Are we done?

Functionality:
Feature: Customers should be able to withdraw deposits.
Problem: Customer can withdraw before revocation.
So far

Probabilistic opening: prevents linkability.
Deposits: prevent double-spending.

Are we done?

**Functionality:**

*Feature:* Customers should be able to withdraw deposits.
*Problem:* Customer can withdraw before revocation.
*Problem:* What if merchant refuses to reply?
So far

Probabilistic opening: prevents linkability.
Deposits: prevent double-spending.

Are we done?

**Functionality:**
*Feature:* Customers should be able to withdraw deposits.
*Problem:* Customer can withdraw before revocation.
*Problem:* What if merchant refuses to reply?

**Economic analysis:** How to set deposit value?
So far

Probabilistic opening: prevents linkability.
Deposits: prevent double-spending.

Are we done?

**Functionality:**
*Feature:* Customers should be able to withdraw deposits.
*Problem:* Customer can withdraw before revocation.
*Problem:* What if merchant refuses to reply?

**Economic analysis:** How to set deposit value?

See paper for solutions!
Takeaways
Takeaways

Used translucent crypto + game theory to construct
Takeaways

Used translucent crypto + game theory to construct
  Decentralized
  Anonymous
  Micropayments
Takeaways

Used translucent crypto + game theory to construct

Decentralized
Anonymous
Micropayments

Game-theoretic analysis more broadly applicable:
   Eg: Pass-Shelat do not specify value of deposit.
   Eg: Probabilistic smart contracts.
Takeaways

Used translucent crypto + game theory to construct
- Decentralized
- Anonymous
- Micropayments

Game-theoretic analysis more broadly applicable:
- Eg: Pass-Shelat do not specify value of deposit.
- Eg: Probabilistic smart contracts.

We also discovered pain points in Zerocash interface.
Resulted in a more “programmable” interface.
Takeaways

Used translucent crypto + game theory to construct
Decentralized
Anonymous
Micropayments

Game-theoretic analysis more broadly applicable:
Eg: Pass-Shelat do not specify value of deposit.
Eg: Probabilistic smart contracts.

We also discovered pain points in Zerocash interface.
Resulted in a more “programmable” interface.

Thanks!

http://eprint.iacr.org/2016/1033