A Composable Security Treatment of the Lightning Network

Aggelos Kiayias
Orfeas Stefanos Thyfronitis Litos
University of Edinburgh

24/6/2020

Part 1
The Lightning Network

VISA

20,000 tx/s

bitcoin

7 tx/s

Problem

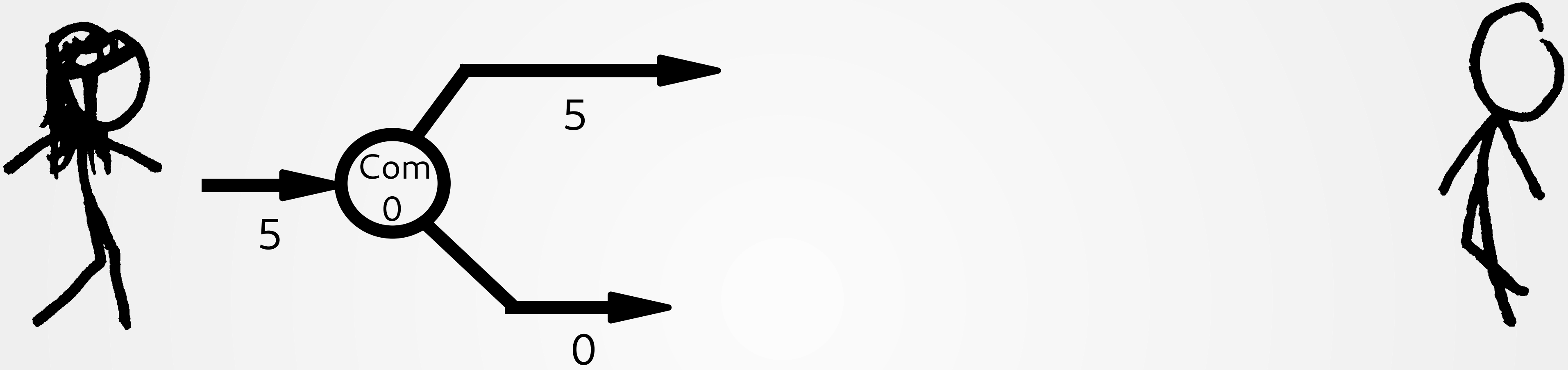
All txs validated by all wallets

Solution

- Move most txs off-chain
- Resolve disputes on-chain

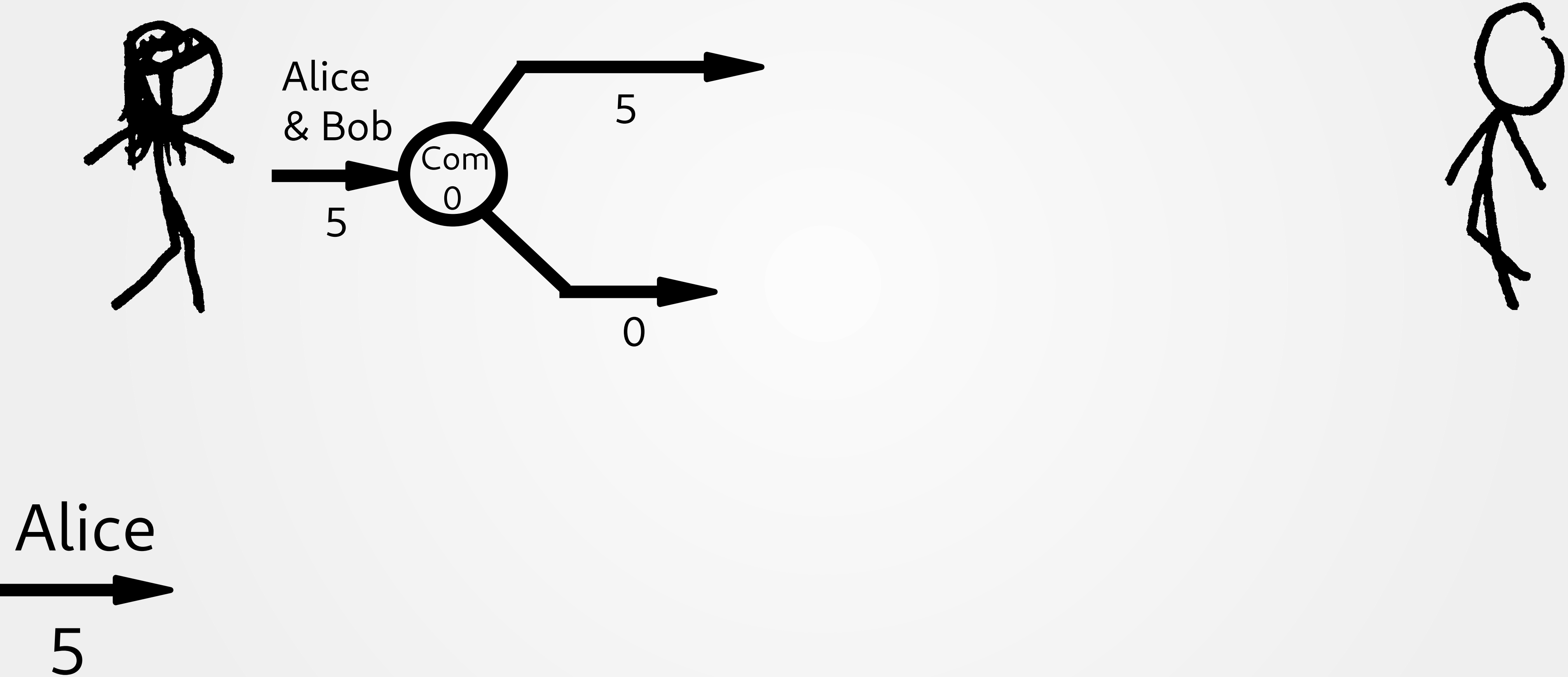
Alice

5



Alice

5



Alice
& Bob

5

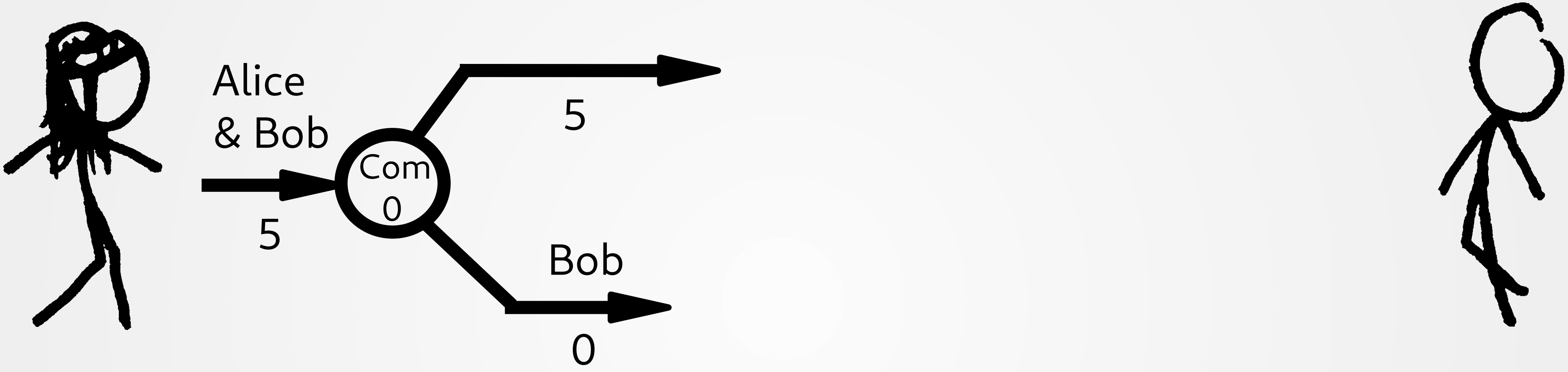
Com
0

5

0

Alice

5



Alice & Bob

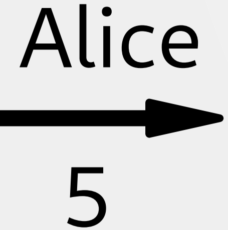
5

Com 0

5

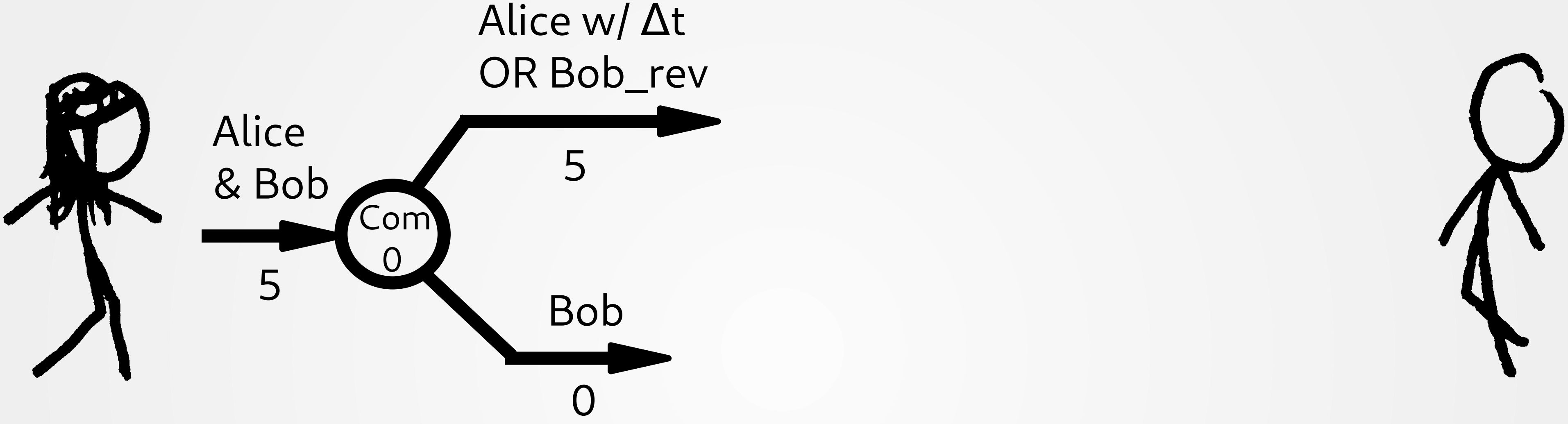
Bob

0



Alice

5



Alice
& Bob

5

Com
0

Alice w/ Δt
OR Bob_rev

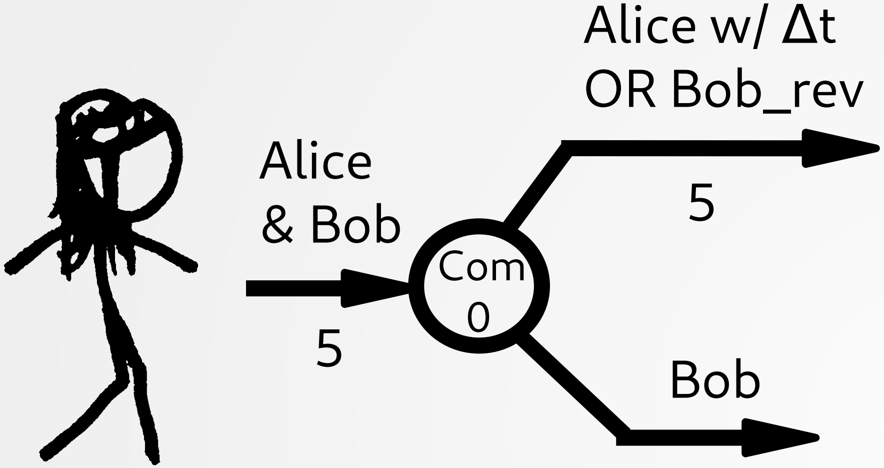
5

Bob

0

Alice

5



Alice w/ Δt
OR Bob_rev

5

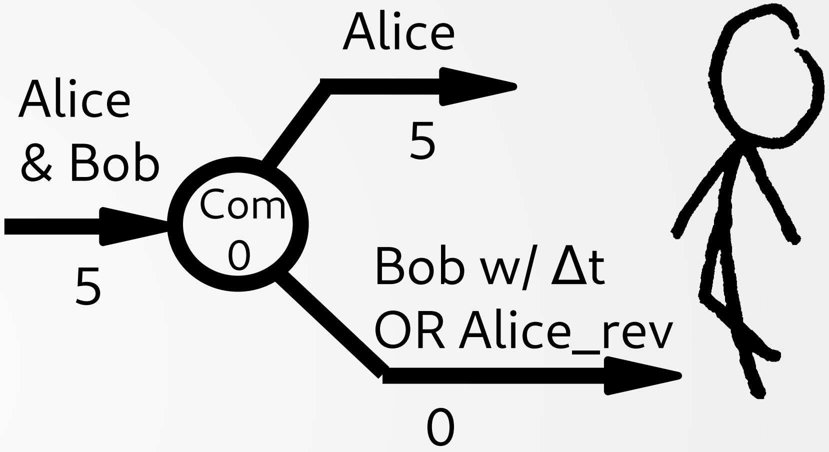
Alice
& Bob

Com
0

5

Bob

0



Alice

5

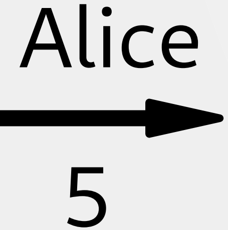
Alice
& Bob

Com
0

5

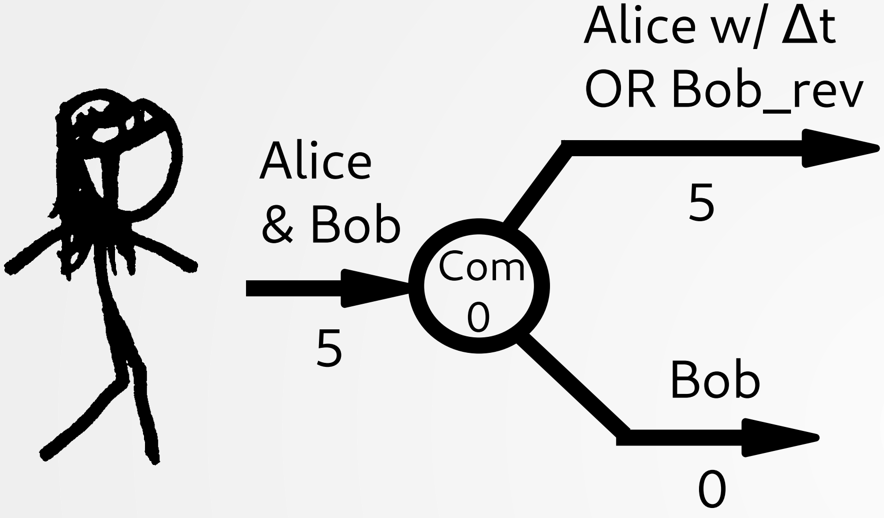
Bob w/ Δt
OR Alice_rev

0



Alice

5



Alice w/ Δt
OR Bob_rev

5

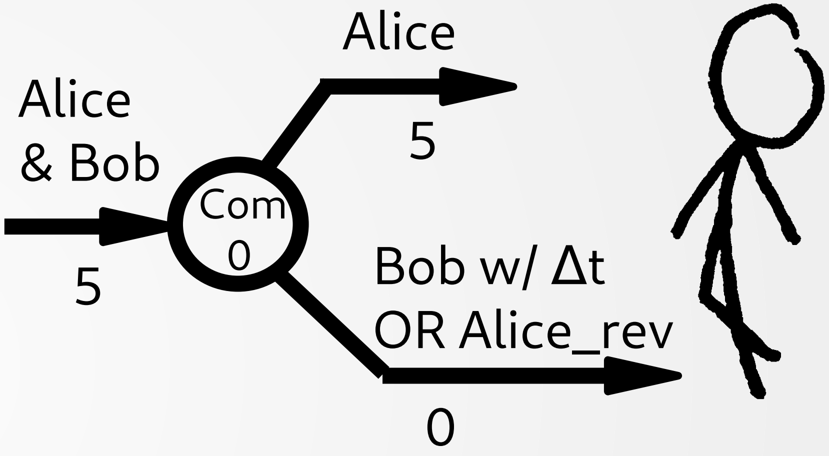
Alice
& Bob

5

Com
0

Bob

0



Alice

5

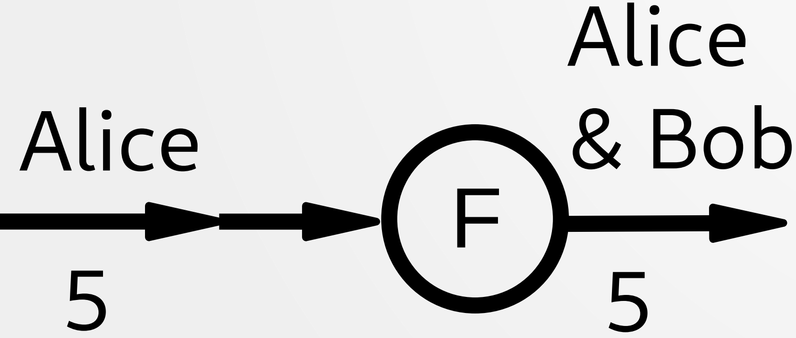
Alice
& Bob

5

Com
0

Bob w/ Δt
OR Alice_rev

0



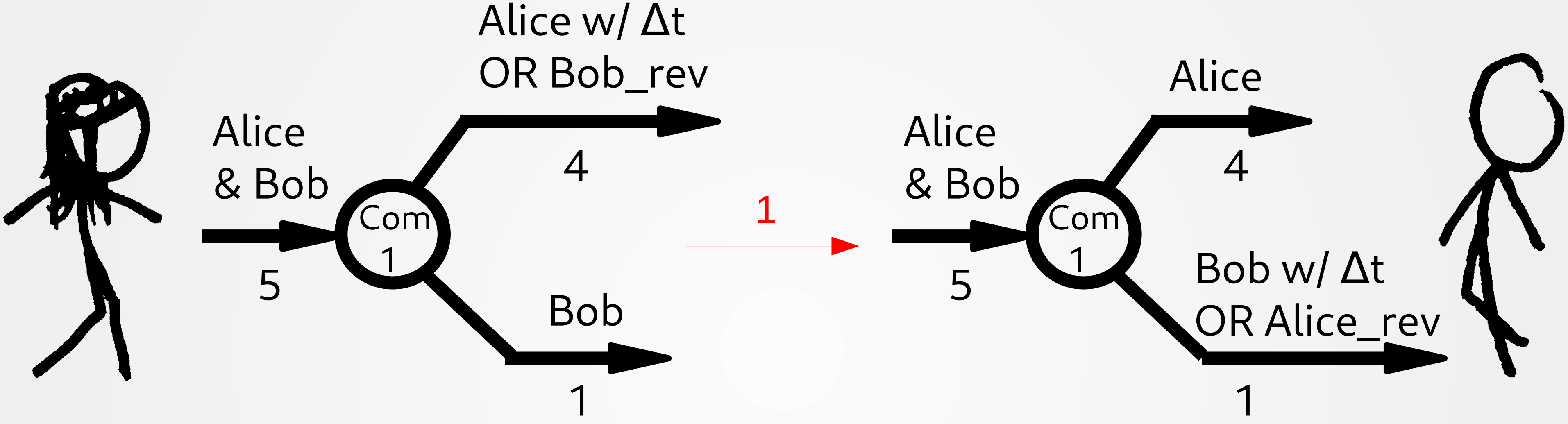
Alice

5

F

Alice
& Bob

5



Alice w/ Δt
OR Bob_rev

Alice
& Bob

Com
1

4

5

Bob

1

1

Alice
& Bob

Com
1

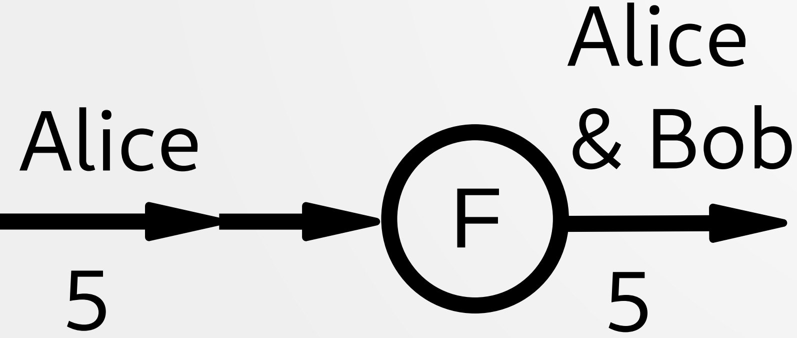
Alice

4

5

Bob w/ Δt
OR Alice_rev

1



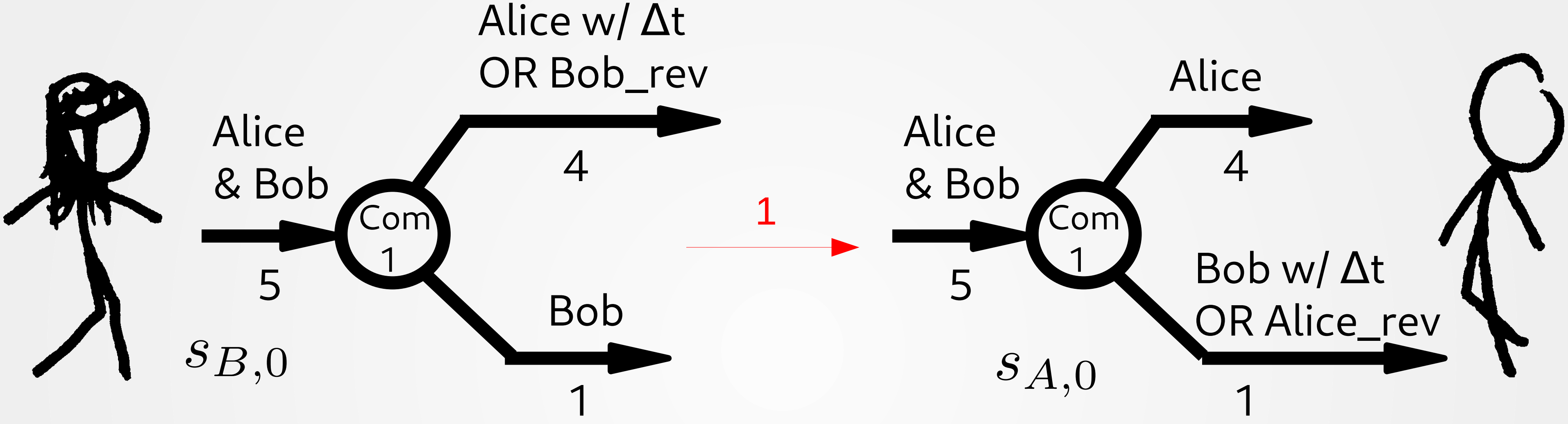
Alice

5

F

Alice
& Bob

5



Alice w/ Δt
OR Bob_rev

Alice
& Bob

Com
1

5

4

Bob

1

$s_{B,0}$

1

Alice
& Bob

Com
1

5

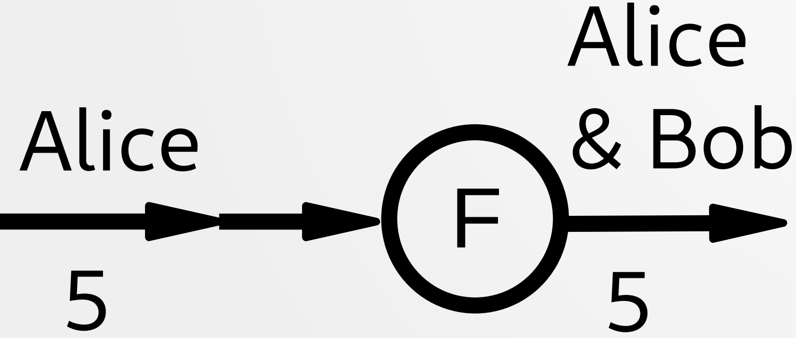
Alice

4

Bob w/ Δt
OR Alice_rev

1

$s_{A,0}$



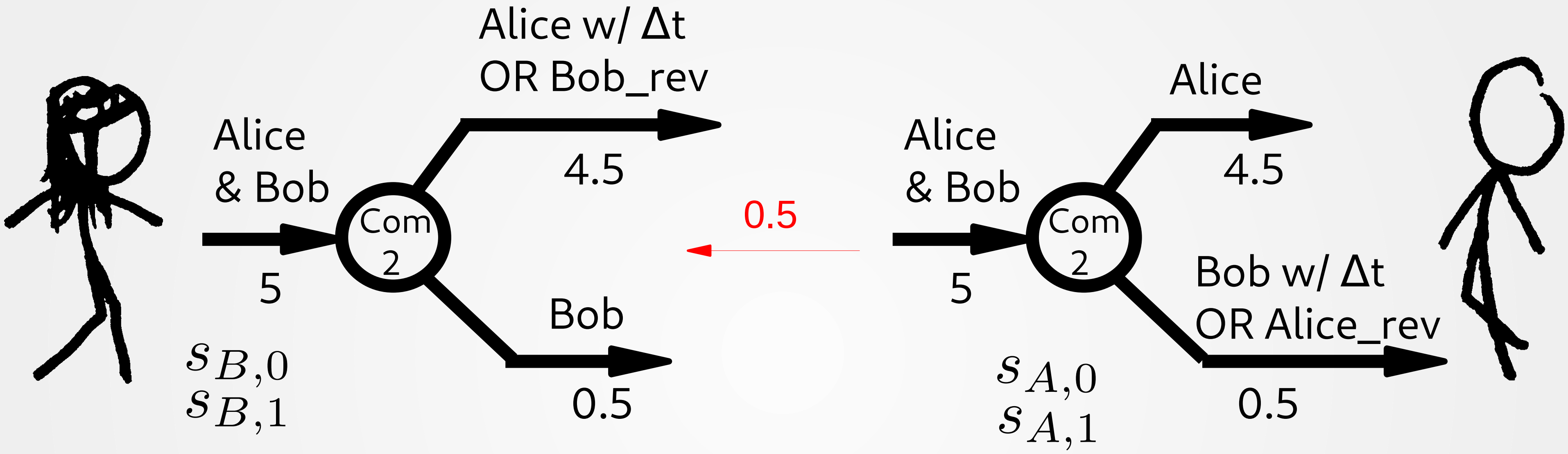
Alice

5

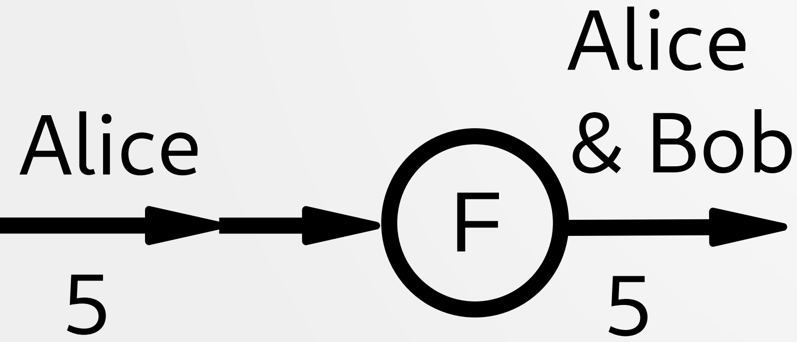
F

Alice
& Bob

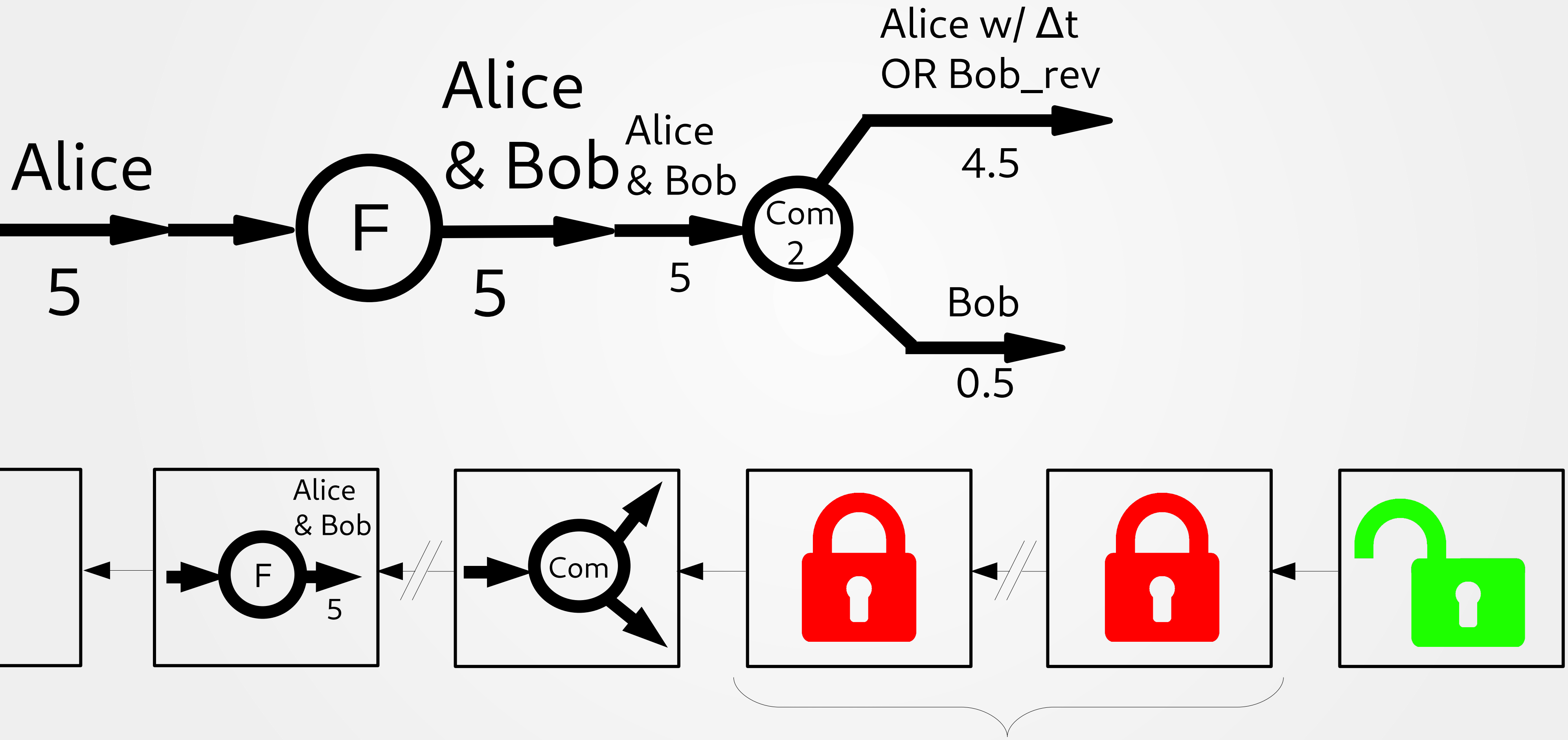
5



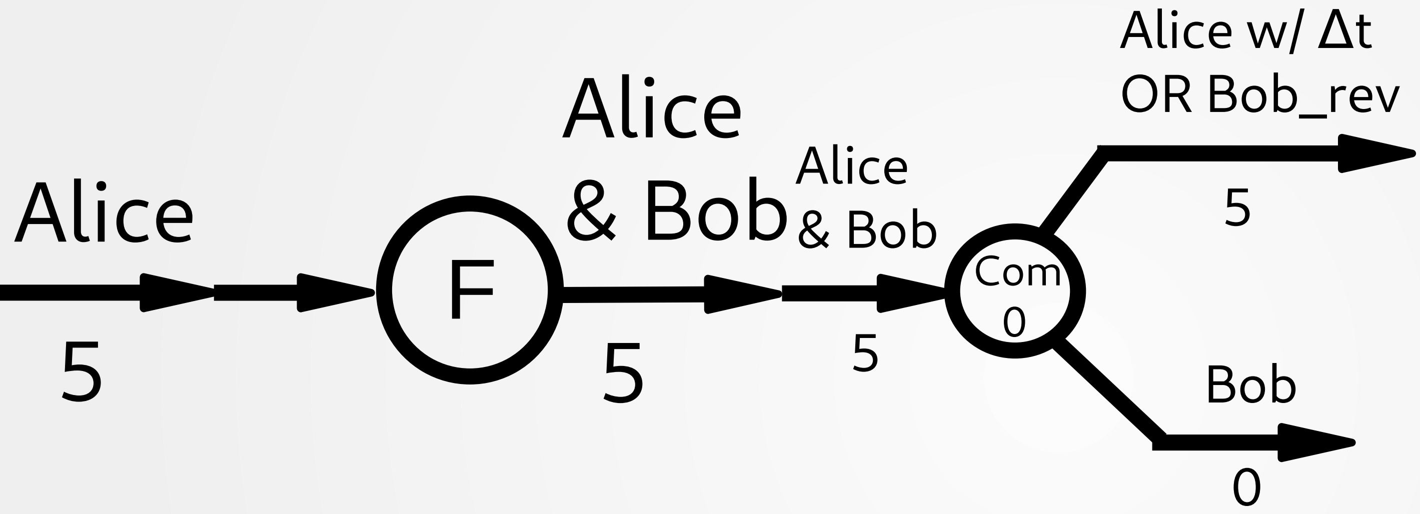
Alice w/ Δt
OR Bob_rev
Alice
& Bob
Com
2
4.5
5
Bob
0.5
$s_{B,0}$
$s_{B,1}$
0.5
Alice
& Bob
Com
2
Alice
4.5
5
Bob w/ Δt
OR Alice_rev
0.5
$s_{A,0}$
$s_{A,1}$



Alice
5
F
Alice
& Bob
5



Dispute period *t*



Alice

5

F

Alice & Bob

5

Alice & Bob

5

Com

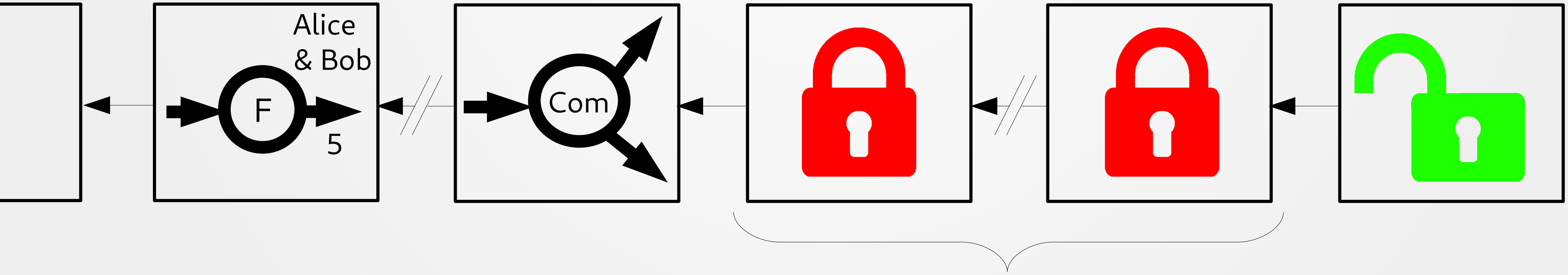
0

Alice w/ Δt OR Bob_rev

5

Bob

0



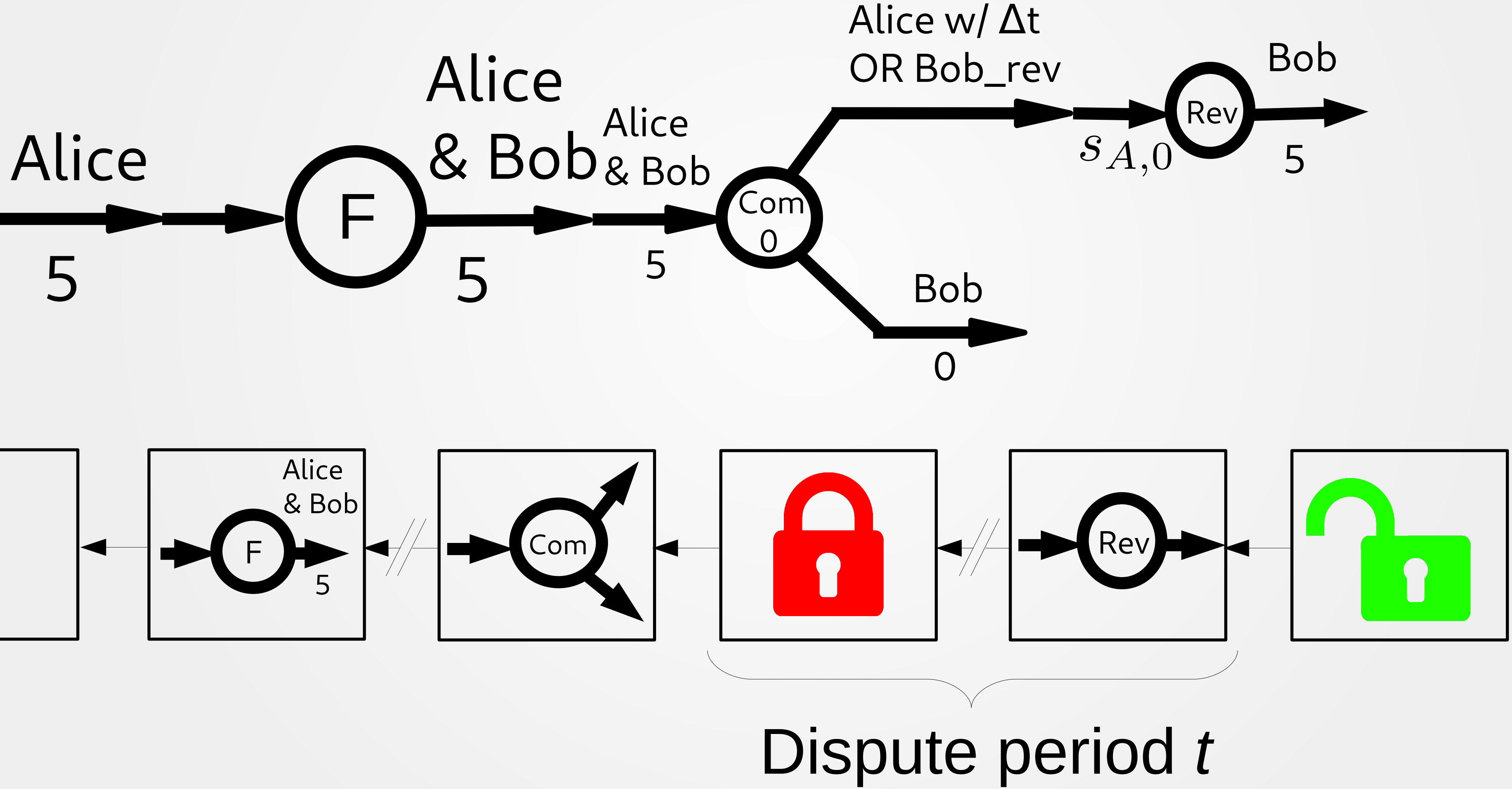
Alice & Bob

F

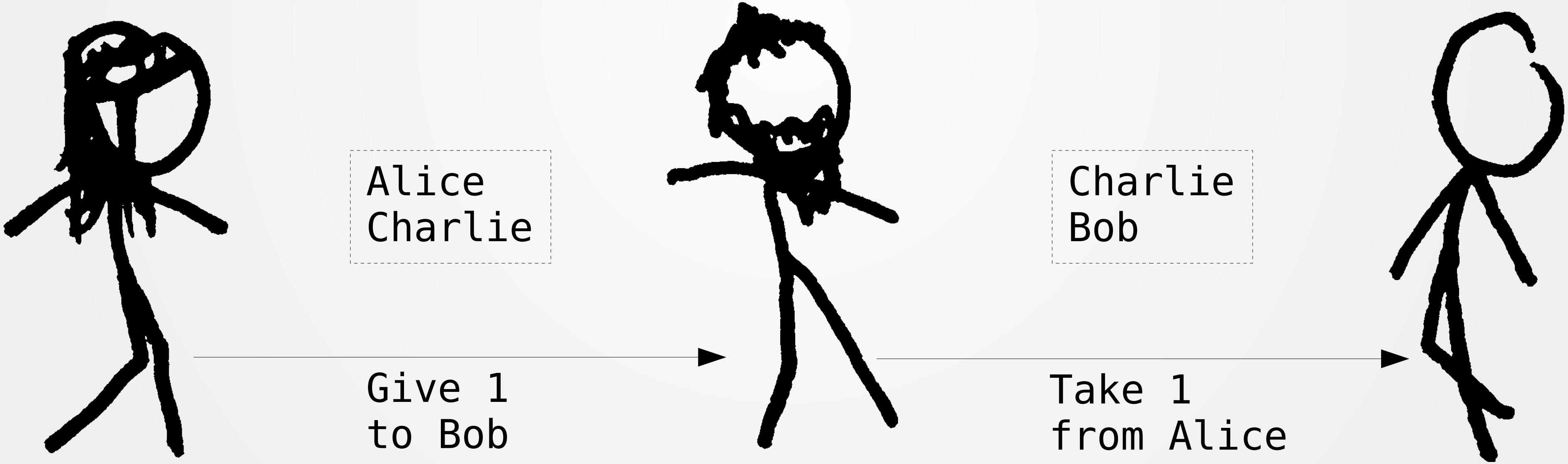
5

Com

Dispute period *t*


Alice
5
F
Alice
& Bob
5
Alice
& Bob
5
Com
0
Alice w/ Δt
OR Bob_rev
$s_{A,0}$
Rev
Bob
5
Bob
0
Alice
& Bob
F
5
Com
Rev
Dispute period *t*

Multi-hop payments



Alice
Charlie

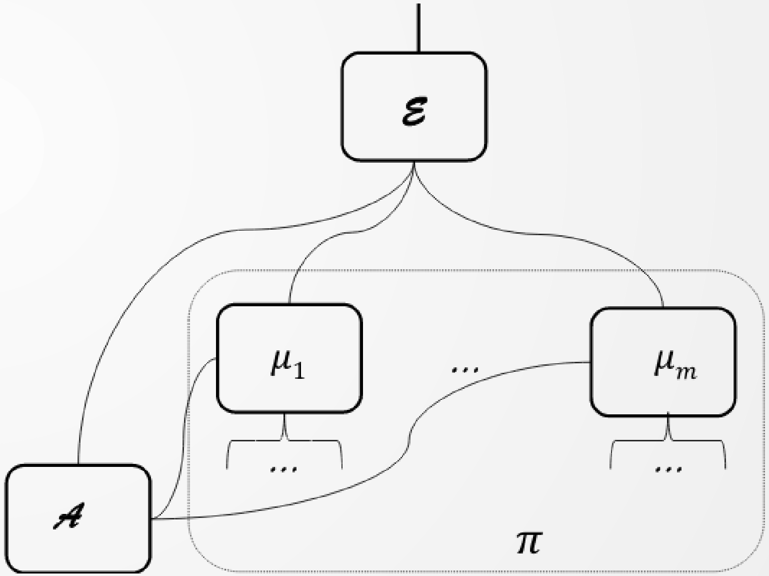
Give 1
to Bob

Charlie
Bob

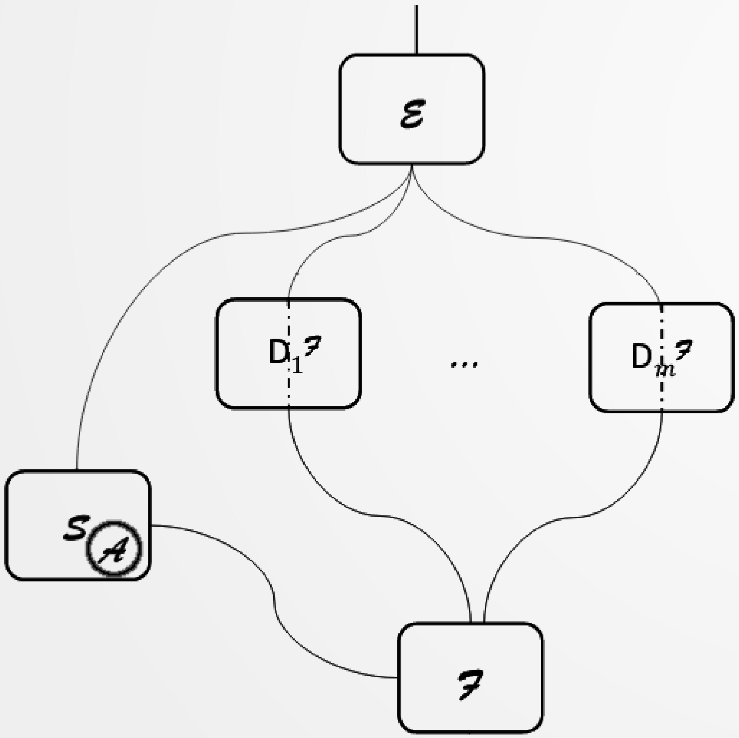
Take 1
from Alice

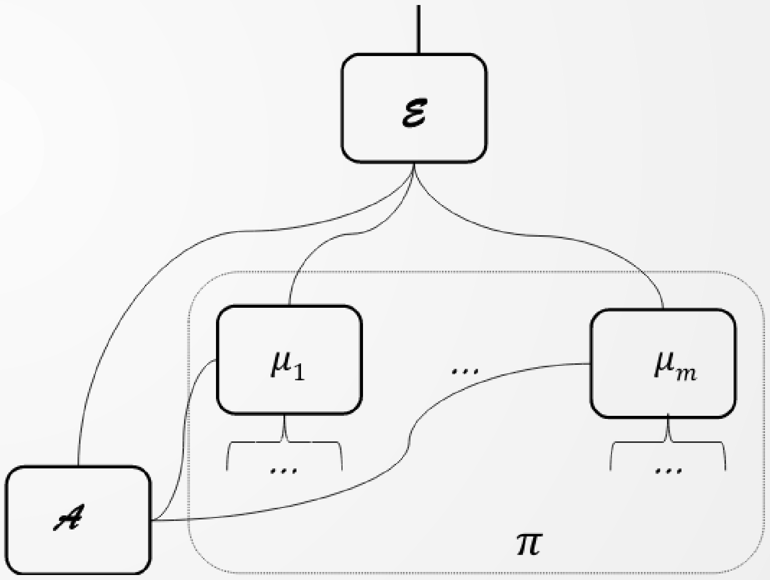
Part 2
Our contribution

Simulation-based Security



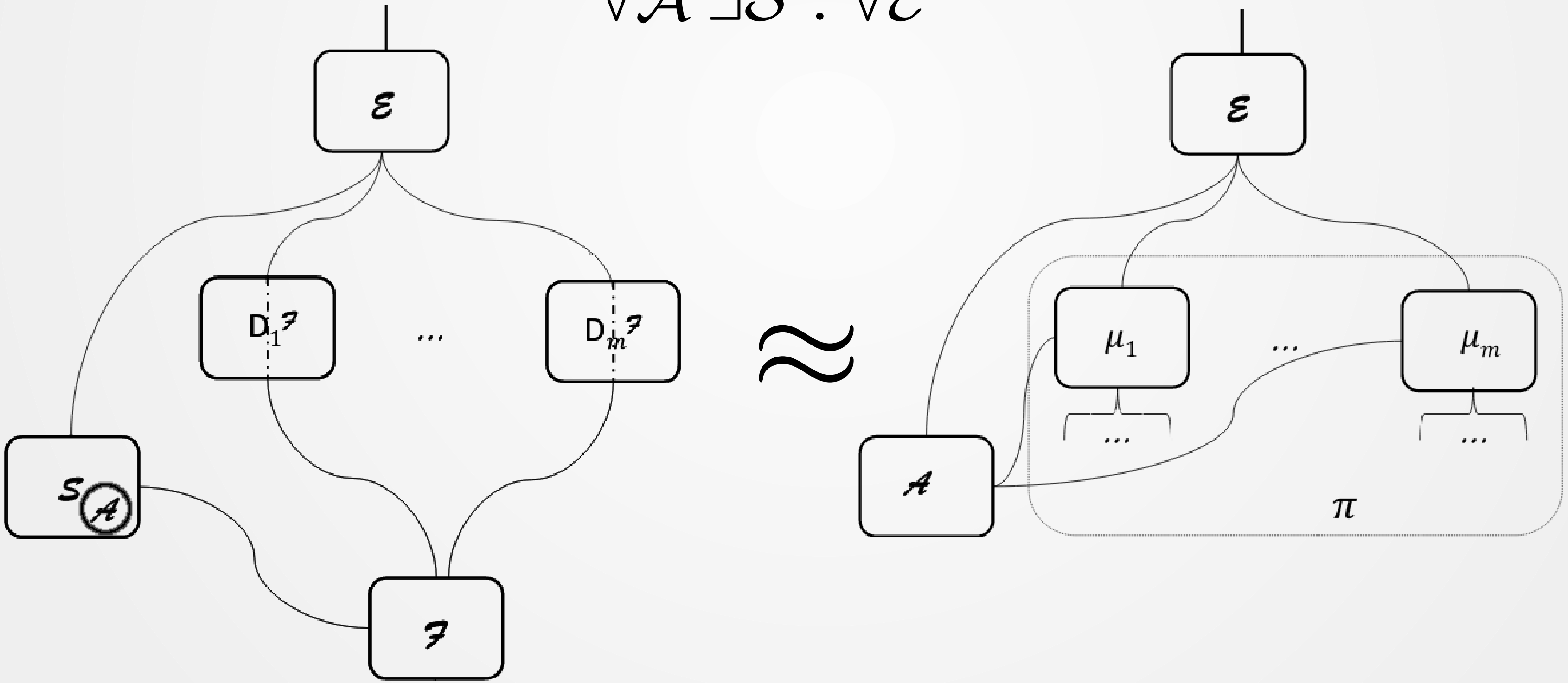
Simulation-based Security





Simulation-based Security

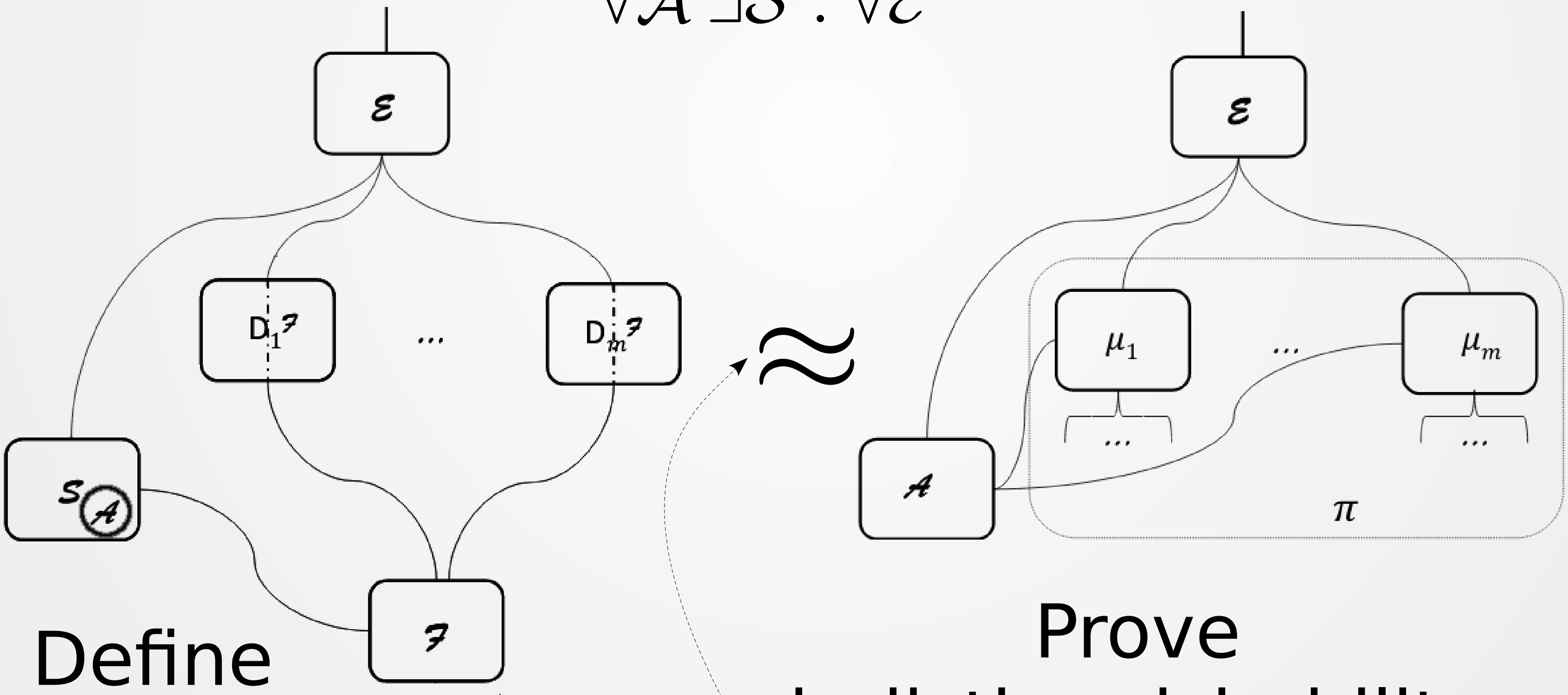
$$\forall \mathcal{A}\ \exists \mathcal{S} : \forall \mathcal{E}$$



Credits: “Universally Composable Security”, Ran Canetti
https://eprint.iacr.org/2000/067

Our paper

$\forall \mathcal{A}\, \exists \mathcal{S} : \forall \mathcal{E}$



Define functionality

Prove indistinguishability

Universal Composition

If functionality F is UC-realized by protocol P, then a protocol R that internally uses F is indistinguishable from R’ that is like R but internally uses P

Universal Composition

$\mathcal{G}_{\mathrm{ledger}}$ [BMTZ’17, BGKRZ’18]

Functionality $\mathcal{F}_{\text{PayNet}}$ – interface

- from $\mathcal{E}$:
  - (register, delay, relayDelay)
  - (toppedUp)
  - (openChannel, $Alice$, Bob, x, tid)
  - (checkForNew, $Alice$, Bob, tid)
  - (pay, Bob, x, $\overrightarrow{\texttt{path}}$, $\texttt{receipt}$)
  - (closeChannel, $\texttt{receipt}$, $pchid$)
  - (forceCloseChannel, $\texttt{receipt}$, $pchid$)
  - (poll)
  - (pushFulfill, $pchid$)
  - (pushAdd, $pchid$)
  - (commit, $pchid$)
  - (fulfillOnChain)
  - (getNews)
- to $\mathcal{E}$:
  - (register, $Alice$, $\texttt{delay}(Alice)$, $\texttt{relayDelay}(Alice)$, pubKey)
  - (registered)
  - (news, newChannels, closedChannels, updatesToReport)
- from $\mathcal{S}$:
  - (registerDone, $Alice$, pubKey)
  - (corrupted, $Alice$)
  - (channelAnnounced, $Alice$, $p_{Alice,F}$, $p_{Bob,F}$, $fchid$, $pchid$, tid)
  - (update, $\texttt{receipt}$, $Alice$)
  - (closedChannel, $\texttt{channel}$, $Alice$)
  - (resolvePays, $payid$, $\texttt{charged}$)
- to $\mathcal{S}$:
  - (register, $Alice$, delay, relayDelay)
  - (openChannel, $Alice$, Bob, x, $fchid$, tid)
  - (channelOpened, $Alice$, $fchid$)
  - (pay, $Alice$, Bob, x, $\overrightarrow{\texttt{path}}$, $\texttt{receipt}$, $payid$)
  - (continue)
  - (closeChannel, $fchid$, $Alice$)
  - (forceCloseChannel, $fchid$, $Alice$)
  - (poll, Σ_{Alice}, $Alice$)
  - (pushFulfill, $pchid$, $Alice$)
  - (pushAdd, $pchid$, $Alice$)
  - (commit, $pchid$, $Alice$)
  - (fulfillOnChain, t, $Alice$)

Our contributions

- Prove Lightning Network security in UC framework
- Derive exact time bounds for how often parties need to check the chain
- Prove that a ledger with instant finality is unrealizable

Our contributions

- Prove Lightning Network security in UC framework
- Derive exact time bounds for how often parties need to check the chain
- Prove that a ledger with instant finality is unrealizable

Thank you!