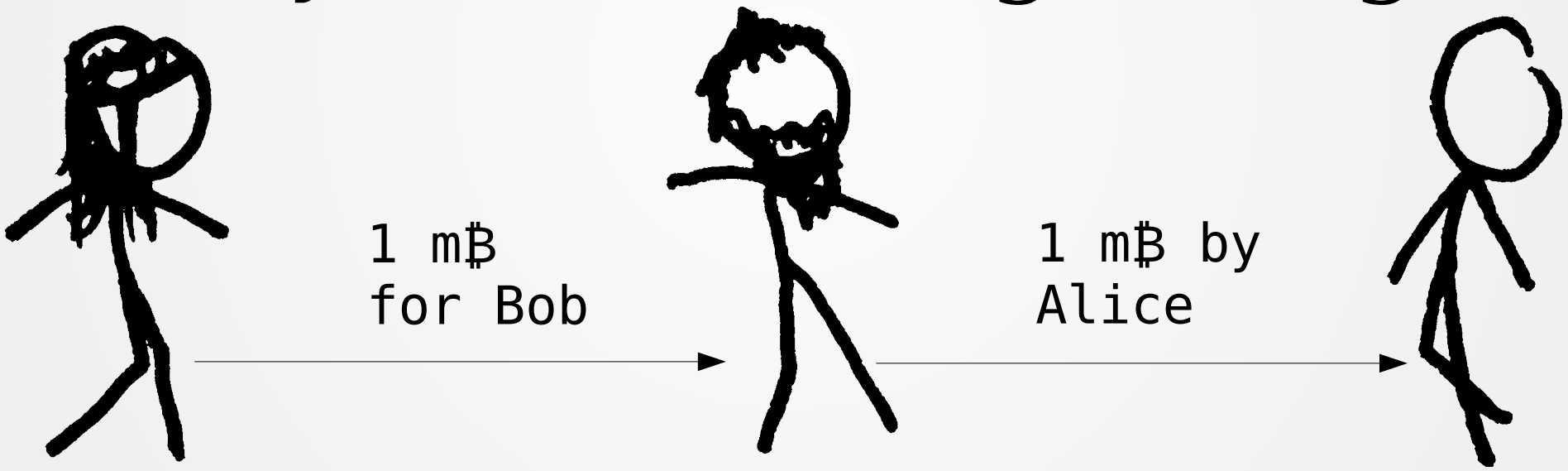
From Channels to Network: Off-chain Multi-hop Payments in Lightning



Orfeas Stefanos Thyfronitis Litos

University of Edinburgh

20,000 tx/s

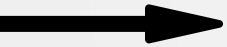
7 tx/s

Problem
All txs validated by all wallets

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

Part 0
Recap: 2-party channels

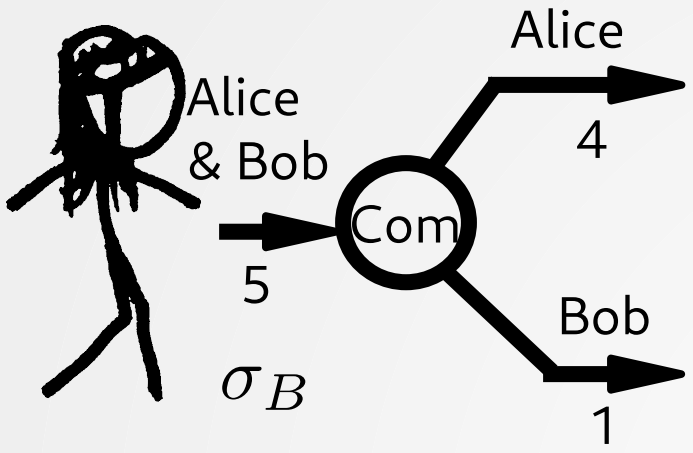
Alice

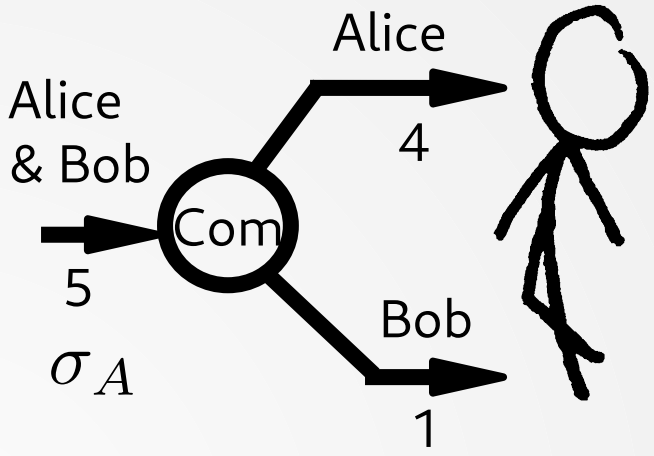


4

Bob

1


Alice
Alice
& Bob
Com
4
5
Bob
σ_B
1

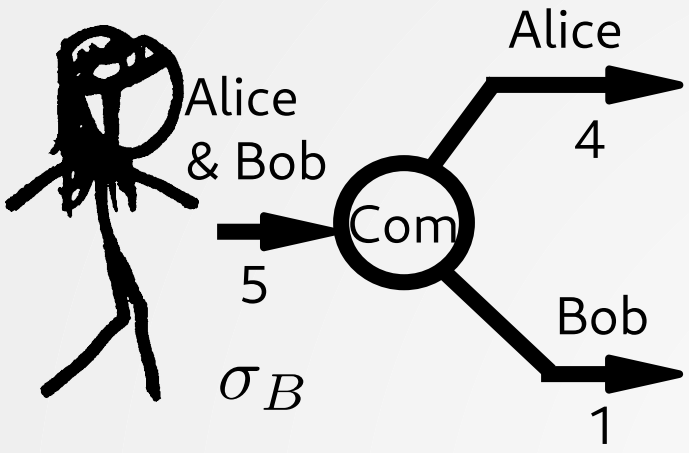

Alice
Alice
& Bob
4
Com
5
Bob
σ_A
1

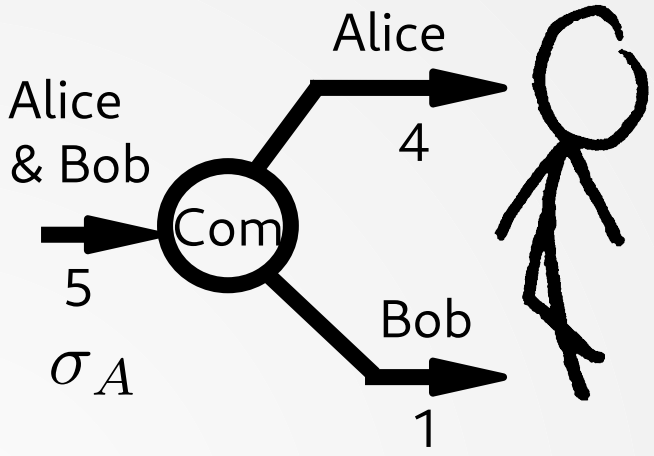
Alice

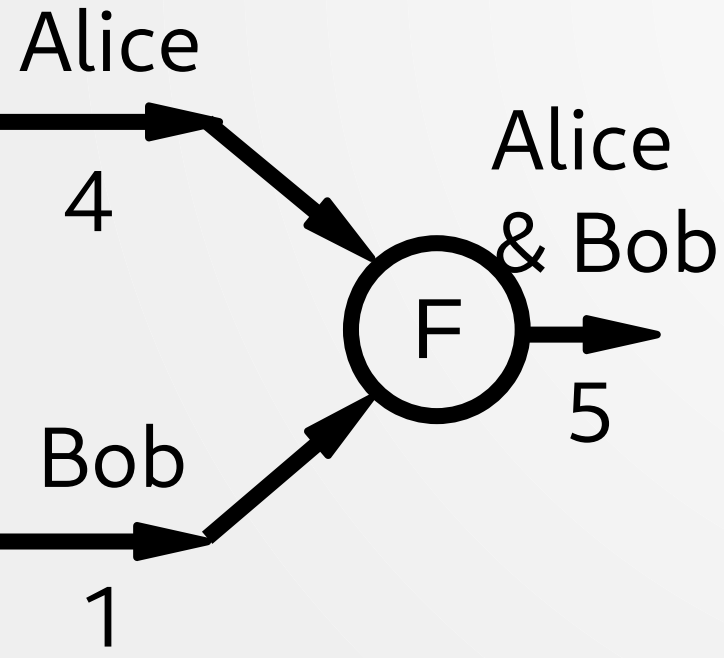
4

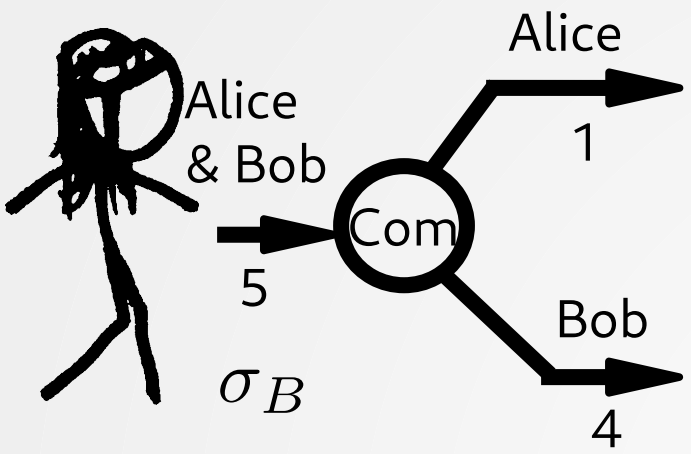
Bob

1

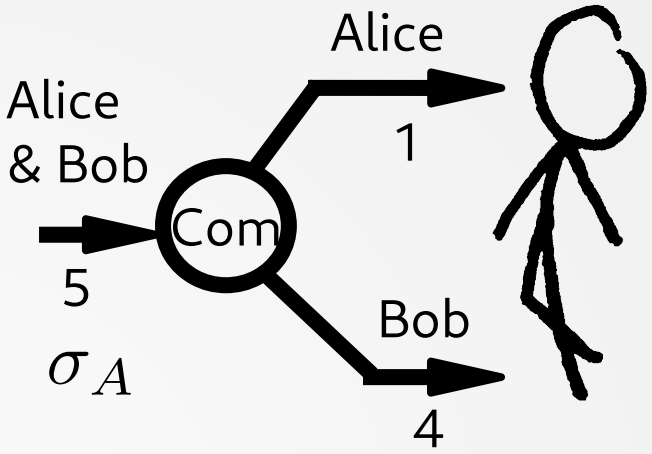

Alice
Alice
& Bob
Com
4
5
Bob
1
σ_B

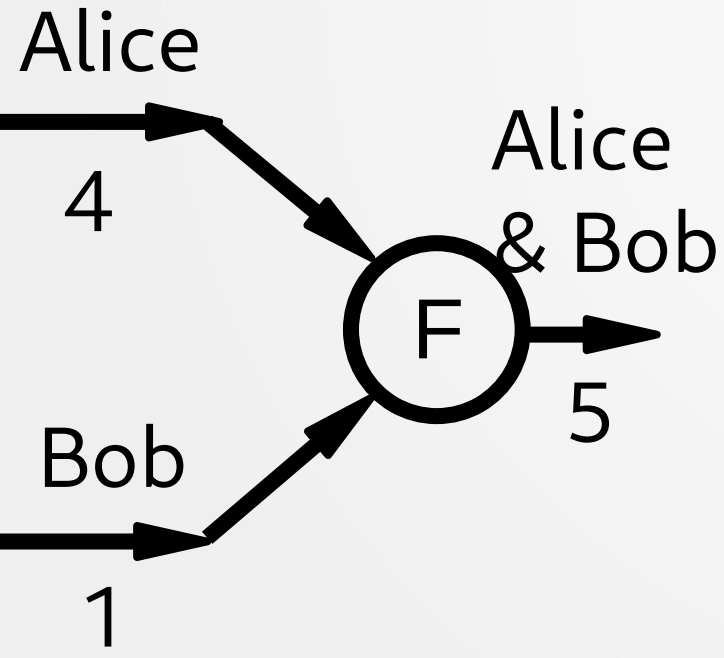

Alice
Alice
& Bob
Com
4
5
Bob
1
σ_A

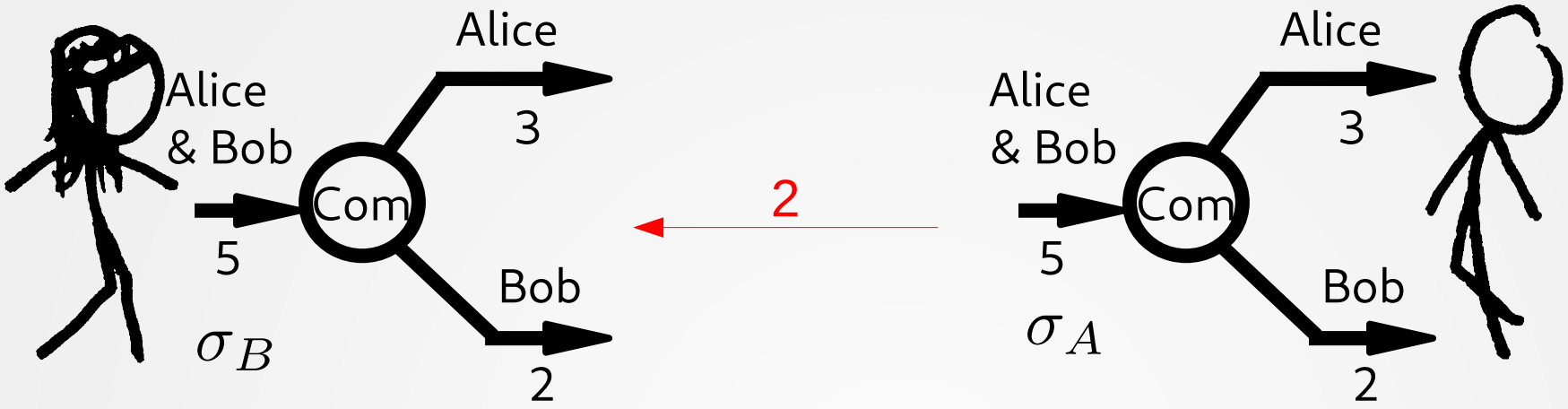

Alice
4
Alice
& Bob
F
5
Bob
1

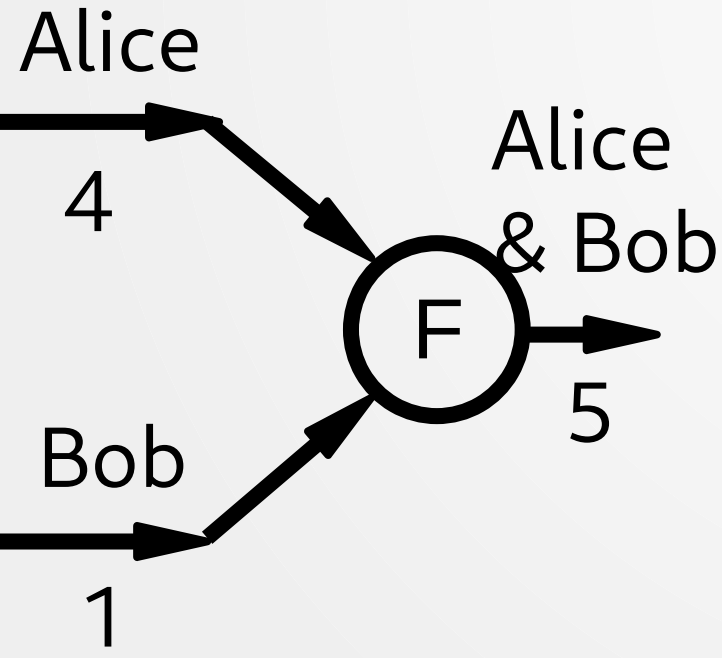

Alice
Alice
& Bob
Com
1
5
Bob
4
σ_B

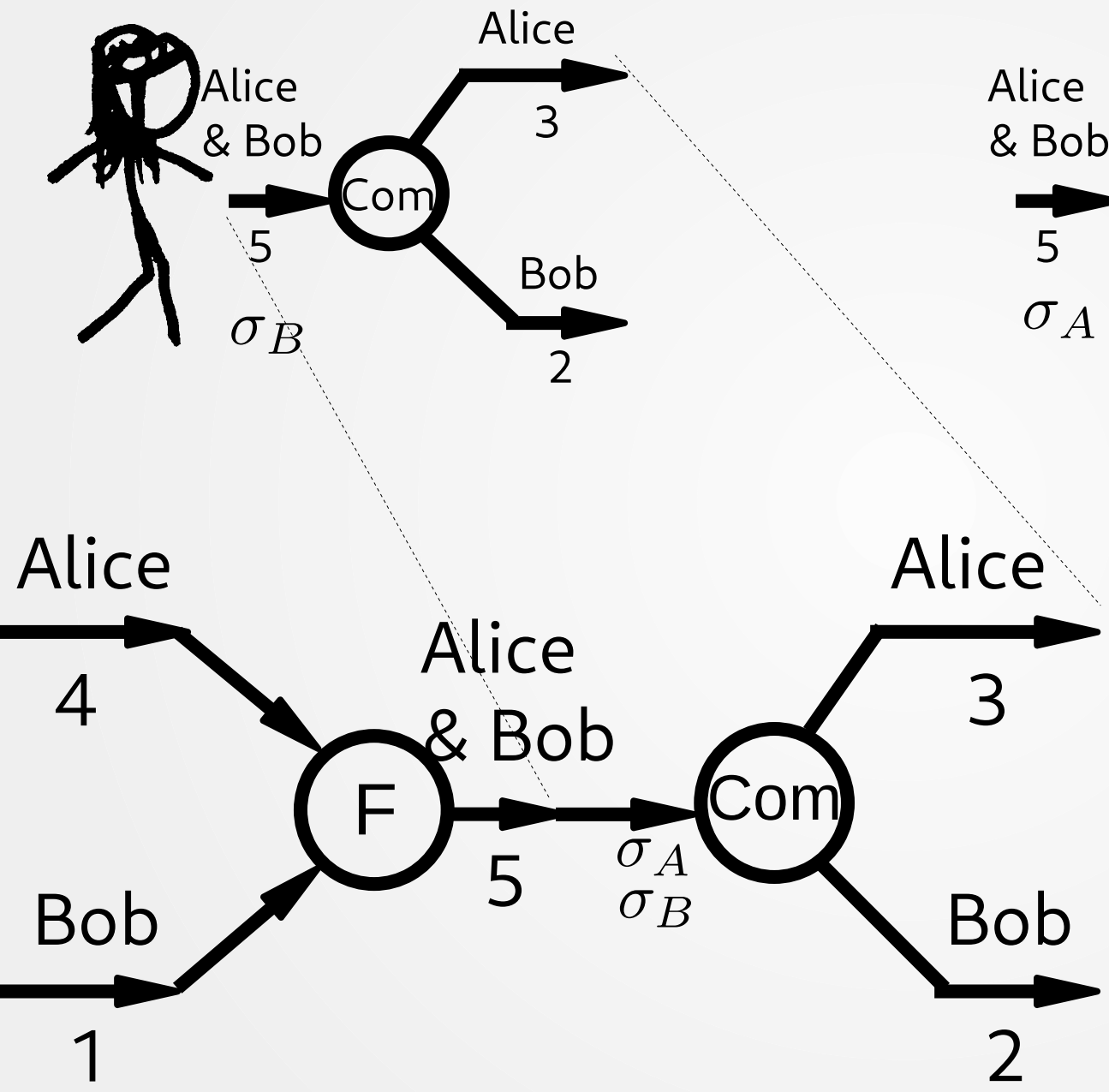
3

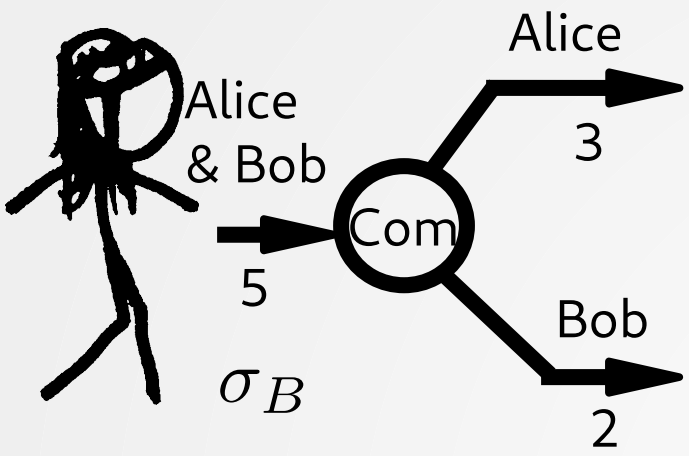

Alice
Alice
& Bob
Com
1
5
Bob
4
σ_A

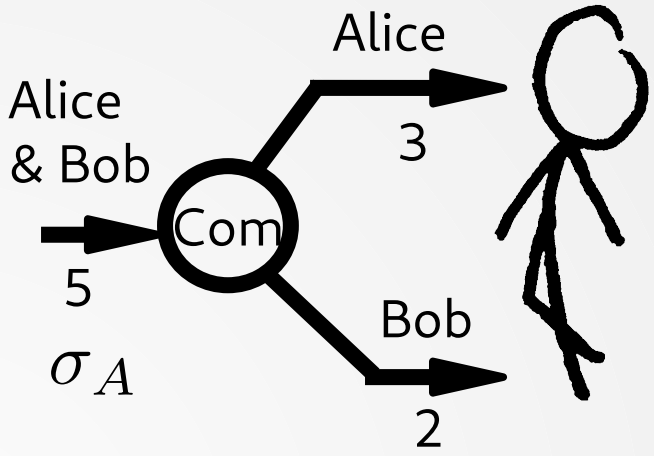

Alice
4
Alice
& Bob
F
5
Bob
1

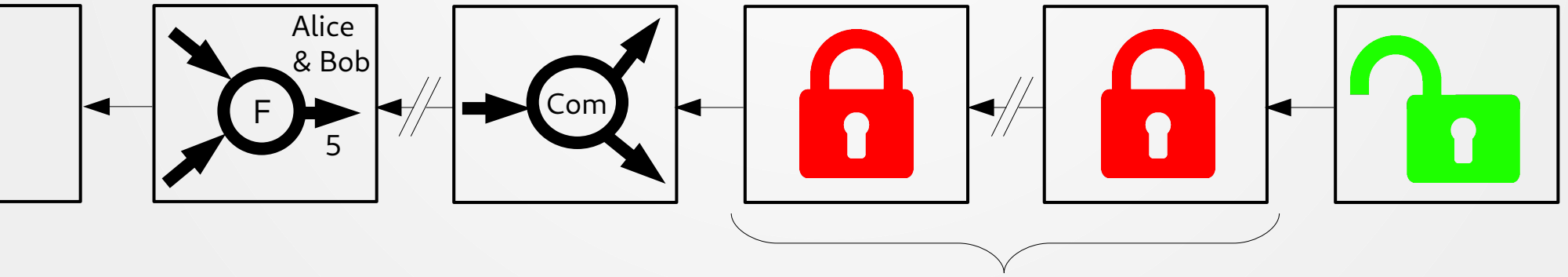

Alice
Alice
& Bob
Com
3
5
2
Bob
σ_B
Alice
Alice
& Bob
Com
3
5
2
Bob
σ_A


Alice
4
Alice
& Bob
F
5
Bob
1


Alice
Alice
& Bob
Com
3
5
Bob
2
σ_B
Alice
Alice
& Bob
Com
3
5
Bob
2
σ_A
Alice
4
Alice
& Bob
F
5
Com
Alice
3
σ_A
σ_B
Bob
Bob
1
2







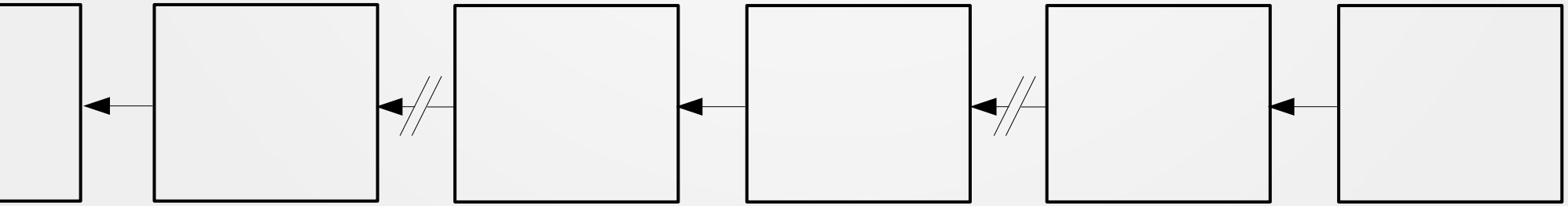
Dispute period

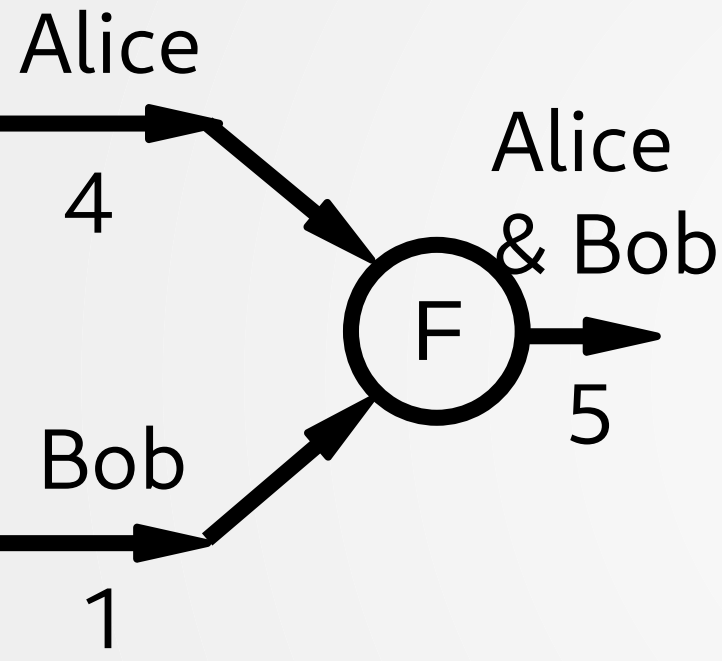
Alice

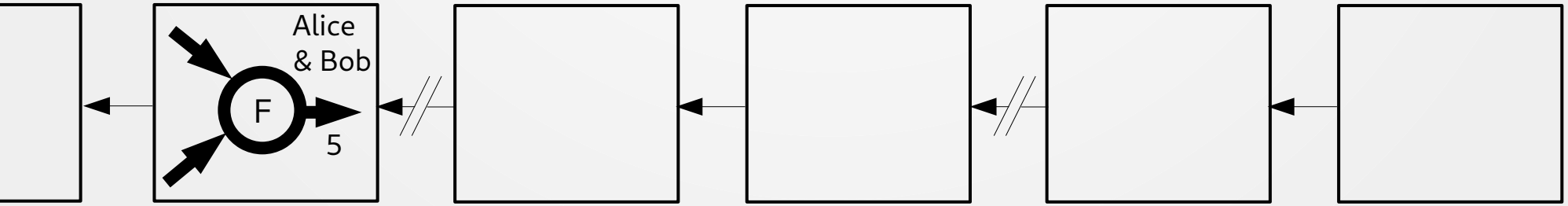
4

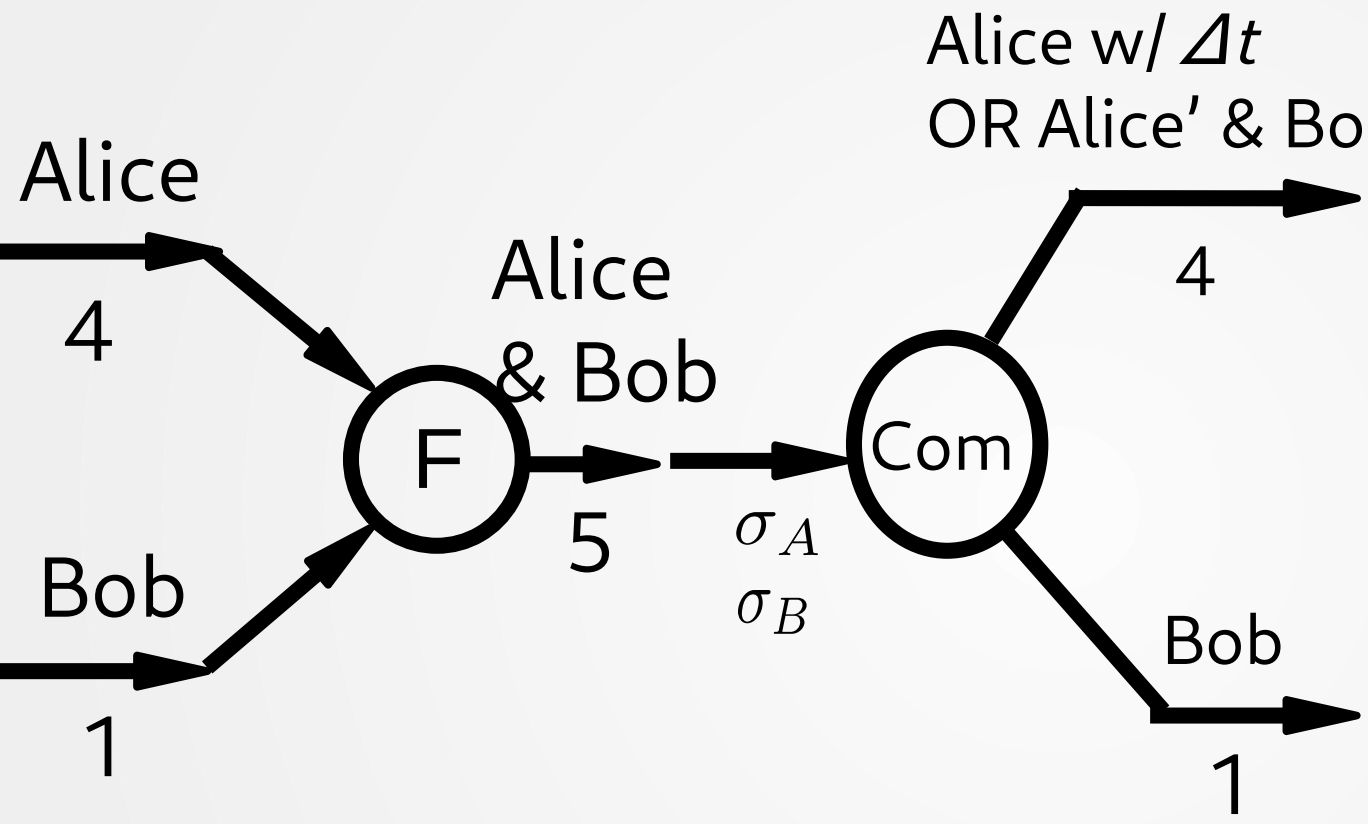
Bob

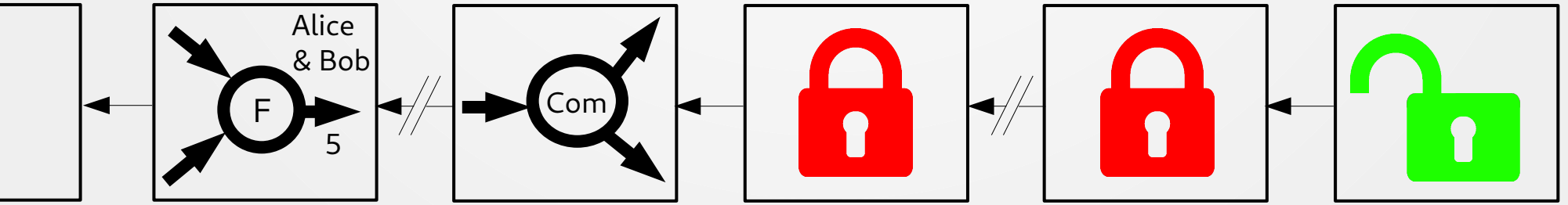
1



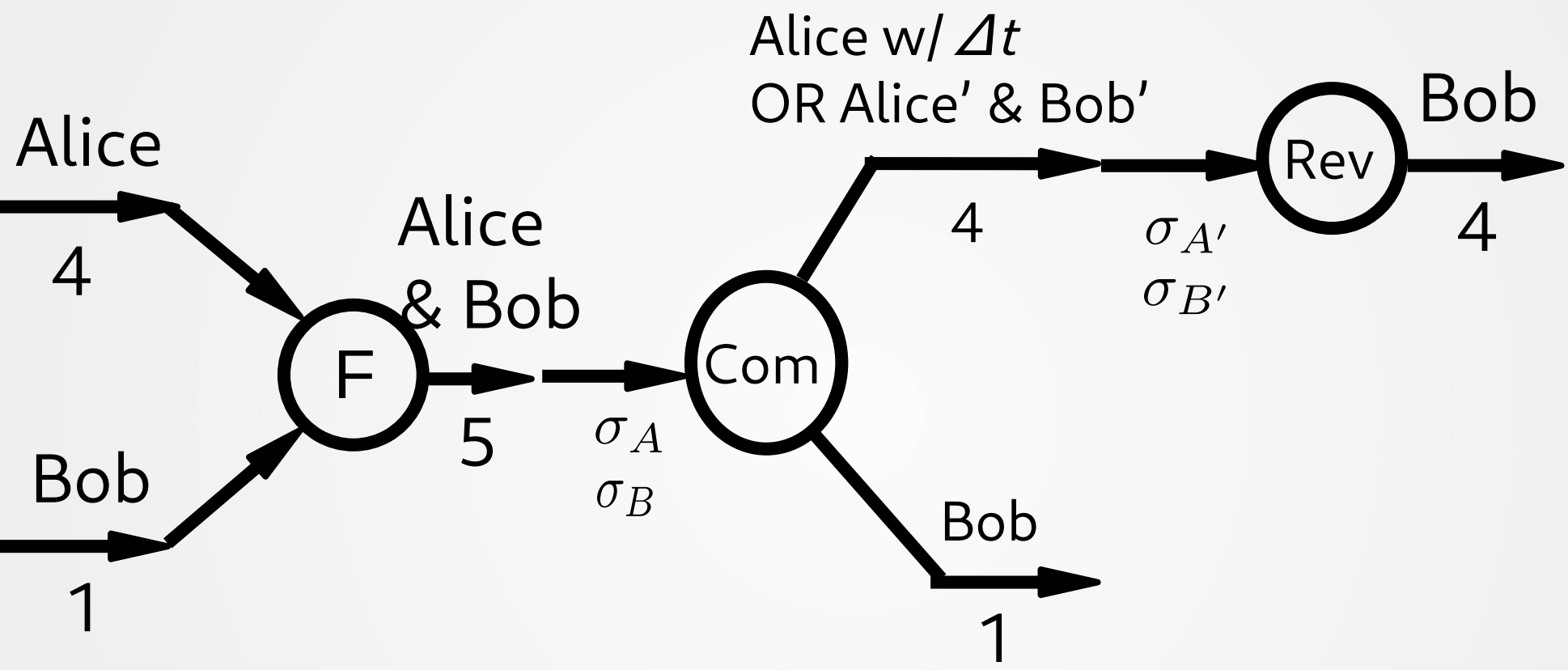

Alice
4
Bob
1
F
Alice
& Bob
5

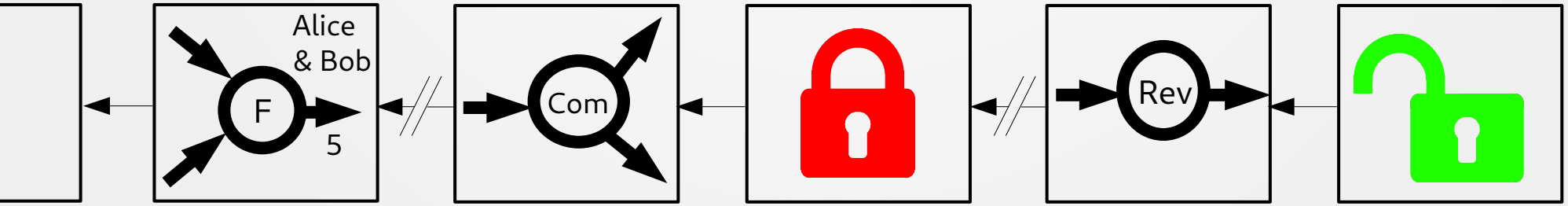

Alice
& Bob
F
5



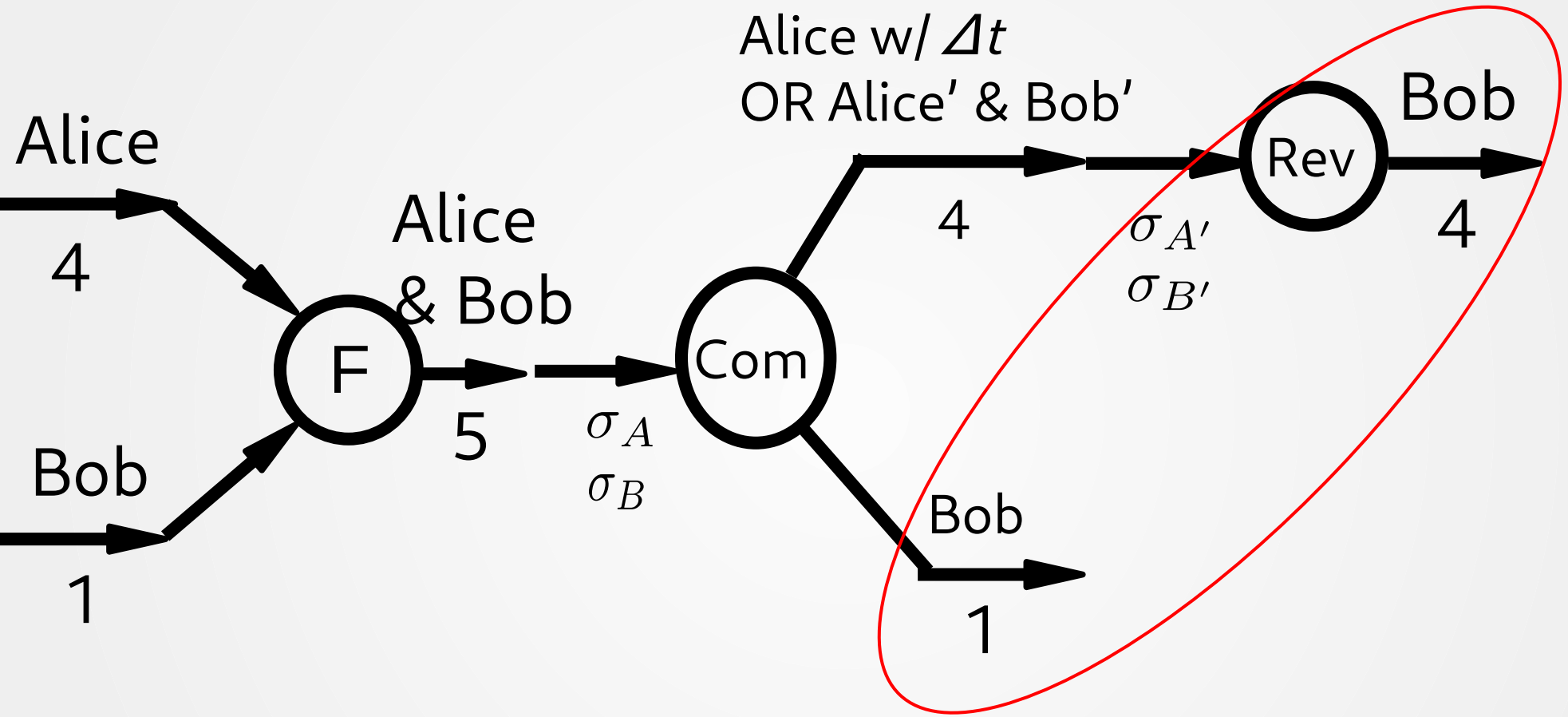


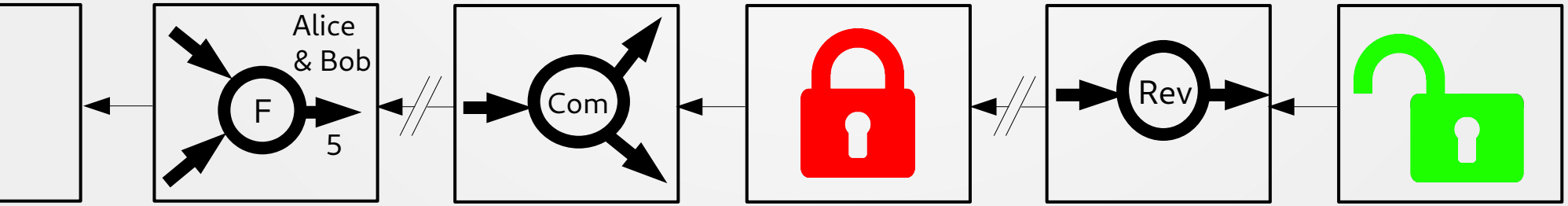
Dispute period *t*





Dispute period *t*





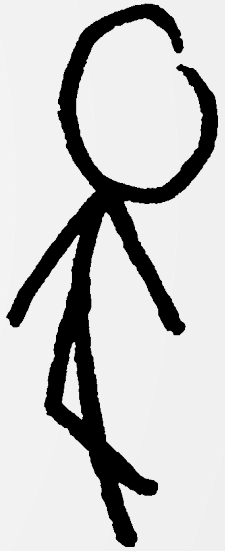
Dispute period *t*

Part 1
Multi-hop payments

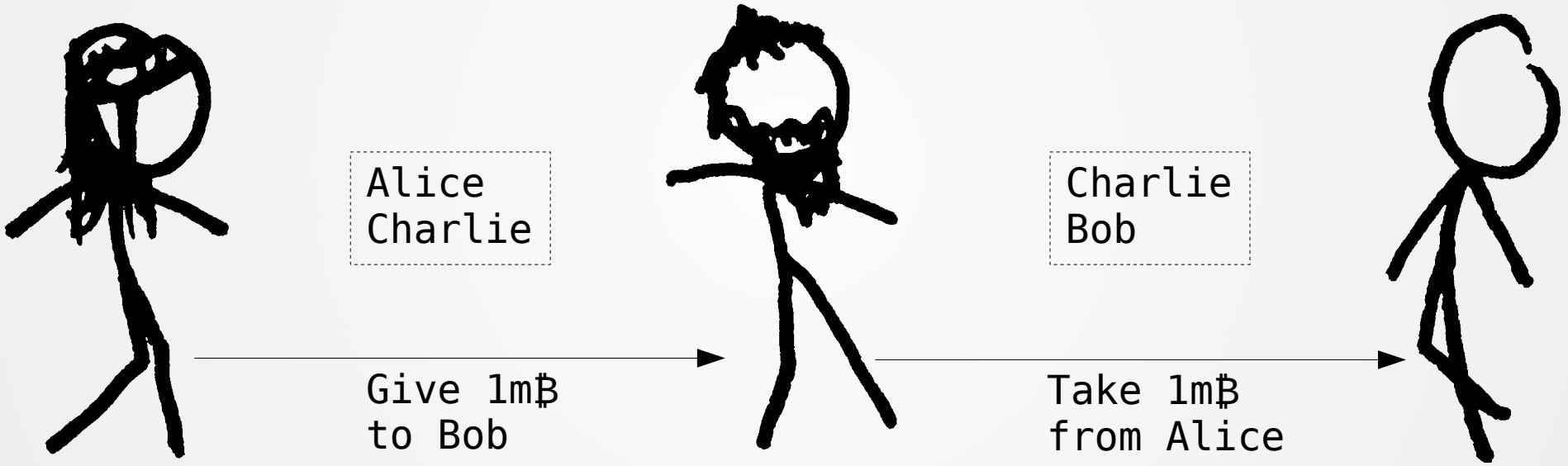
Multi-hop payments

Alice
Charlie

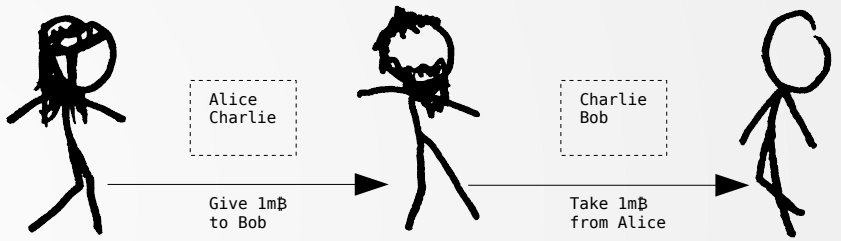
Charlie
Bob



Multi-hop payments

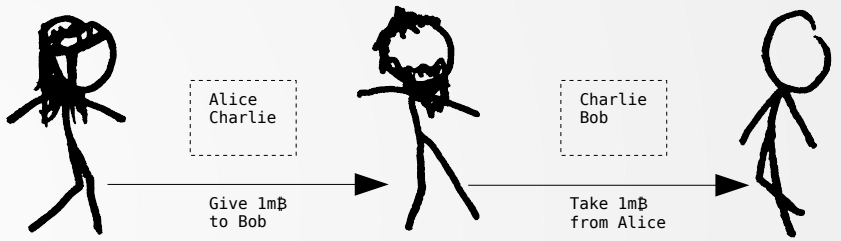


Why no one can cheat?



Why no one can cheat?

HTLC

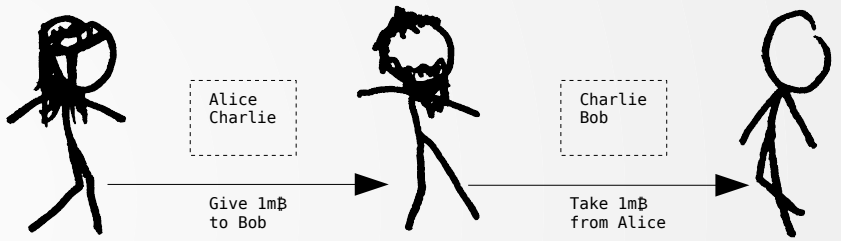


"I, Alice, promise to pay Charlie 1 coin if he reveals a preimage of **0xabcdef** within an hour"

Why no one can cheat?

HTLC

"I, Alice, promise to pay Charlie 1 coin if he reveals a preimage of `0xabcdef` within an hour"

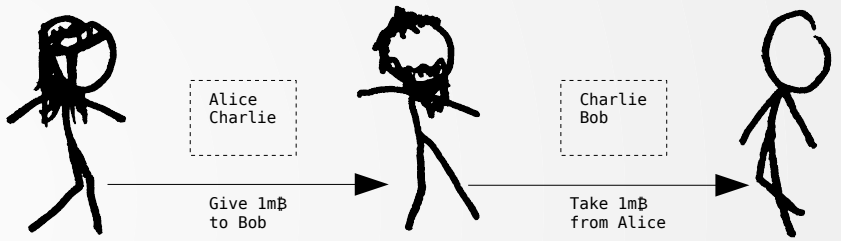


- Bob chooses random **R**, computes **h=H(R)**
- Bob sends **h** to Alice

Why no one can cheat?

HTLC

"I, Alice, promise to pay Charlie 1 coin if he reveals a preimage of `0xabcdef` within an hour"

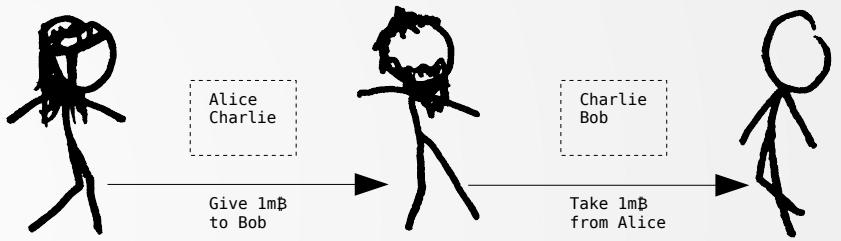


- Bob chooses random **R**, computes **h=H(R)**
- Bob sends **h** to Alice
- Alice signs an **h**-HTLC with Charlie

Why no one can cheat?

HTLC

"I, Alice, promise to pay Charlie 1 coin if he reveals a preimage of `0xabcdef` within an hour"

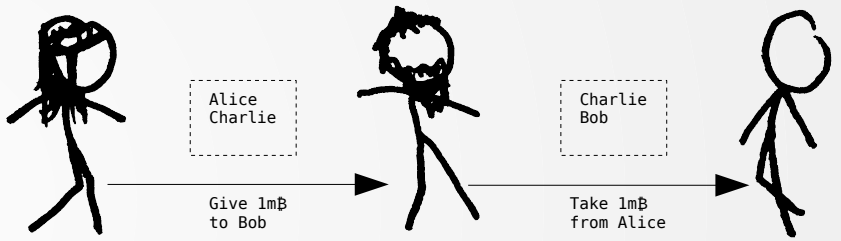


- Bob chooses random **R**, computes **h=H(R)**
- Bob sends **h** to Alice
- Alice signs an **h**-HTLC with Charlie
- Charlie signs an **h**-HTLC with Bob

Why no one can cheat?

HTLC

"I, Alice, promise to pay Charlie 1 coin if he reveals a preimage of `0xabcdef` within an hour"

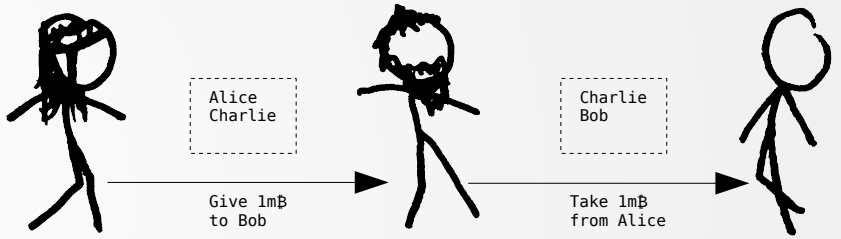


- Bob chooses random **R**, computes **h=H(R)**
- Bob sends **h** to Alice
- Alice signs an **h**-HTLC with Charlie
- Charlie signs an **h**-HTLC with Bob
- Bob reveals **R** to Charlie, gets 1 coin

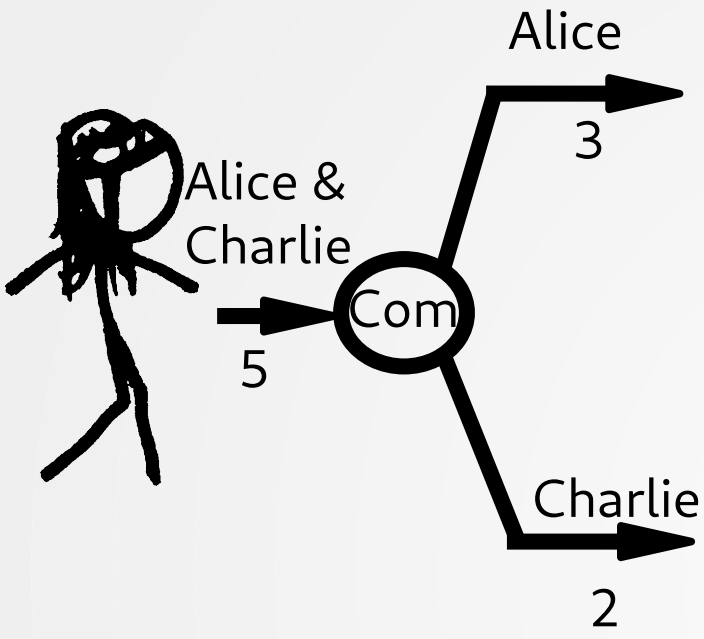
Why no one can cheat?

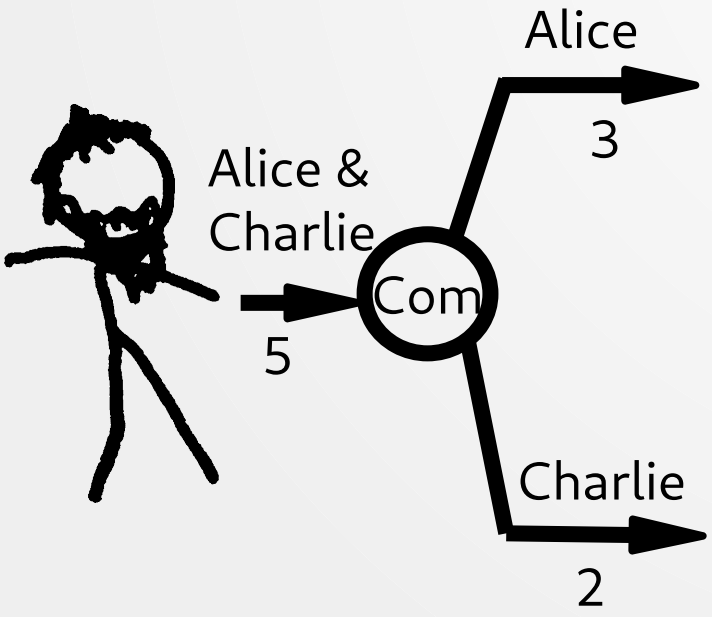
HTLC

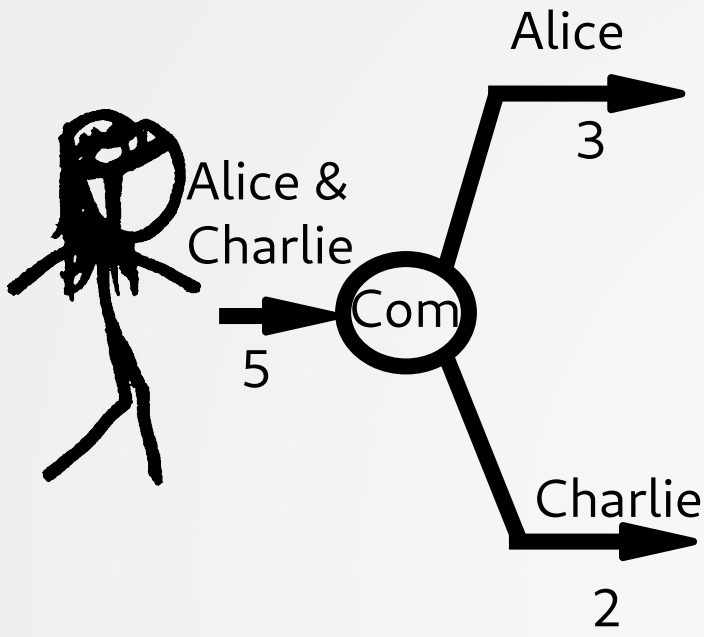
"I, Alice, promise to pay Charlie 1 coin if he reveals a preimage of `0xabcdef` within an hour"

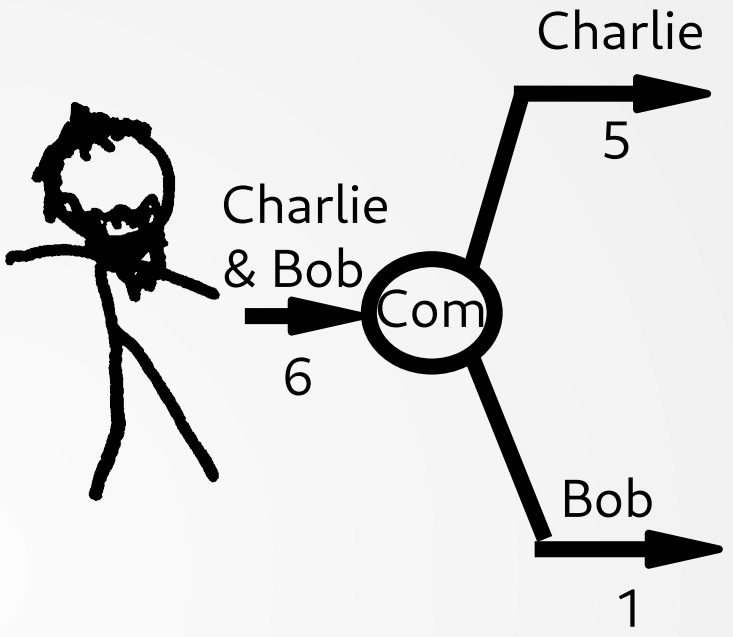


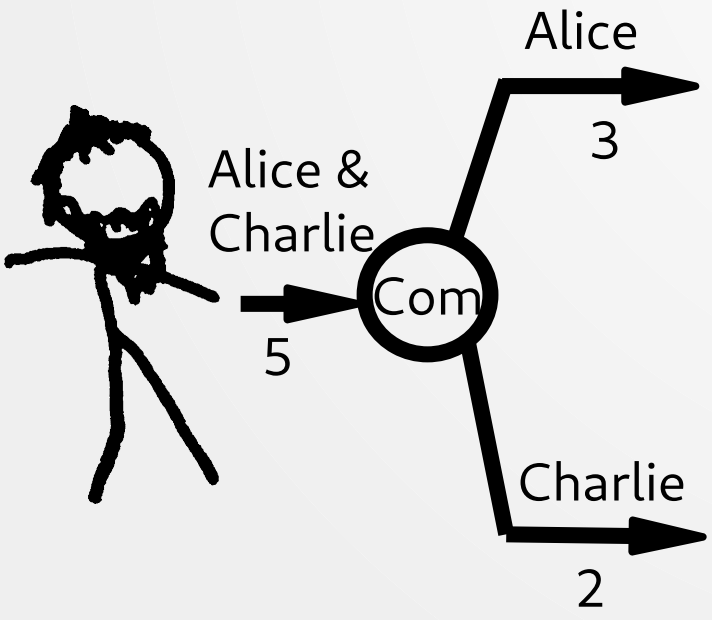
- Bob chooses random **R**, computes **h=H(R)**
- Bob sends **h** to Alice
- Alice signs an **h**-HTLC with Charlie
- Charlie signs an **h**-HTLC with Bob
- Bob reveals **R** to Charlie, gets 1 coin
- Charlie reveals **R** to Alice, gets 1 coin

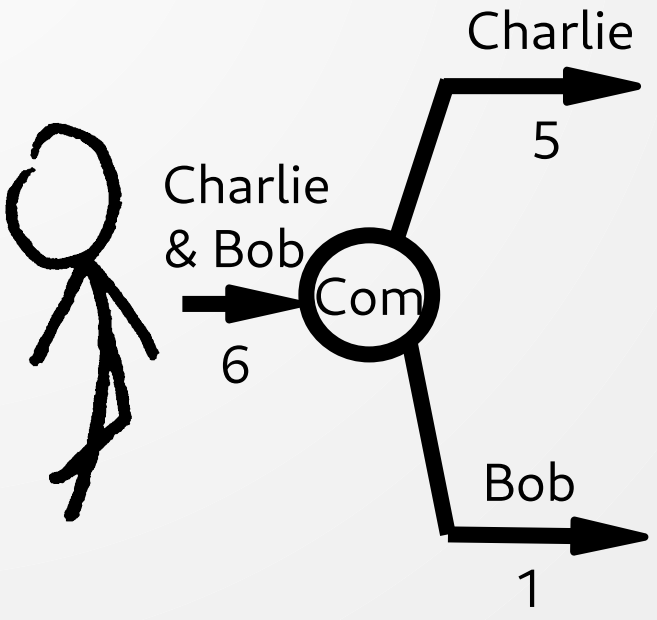

Alice
3
Alice &
Charlie
Com
5
Charlie
2

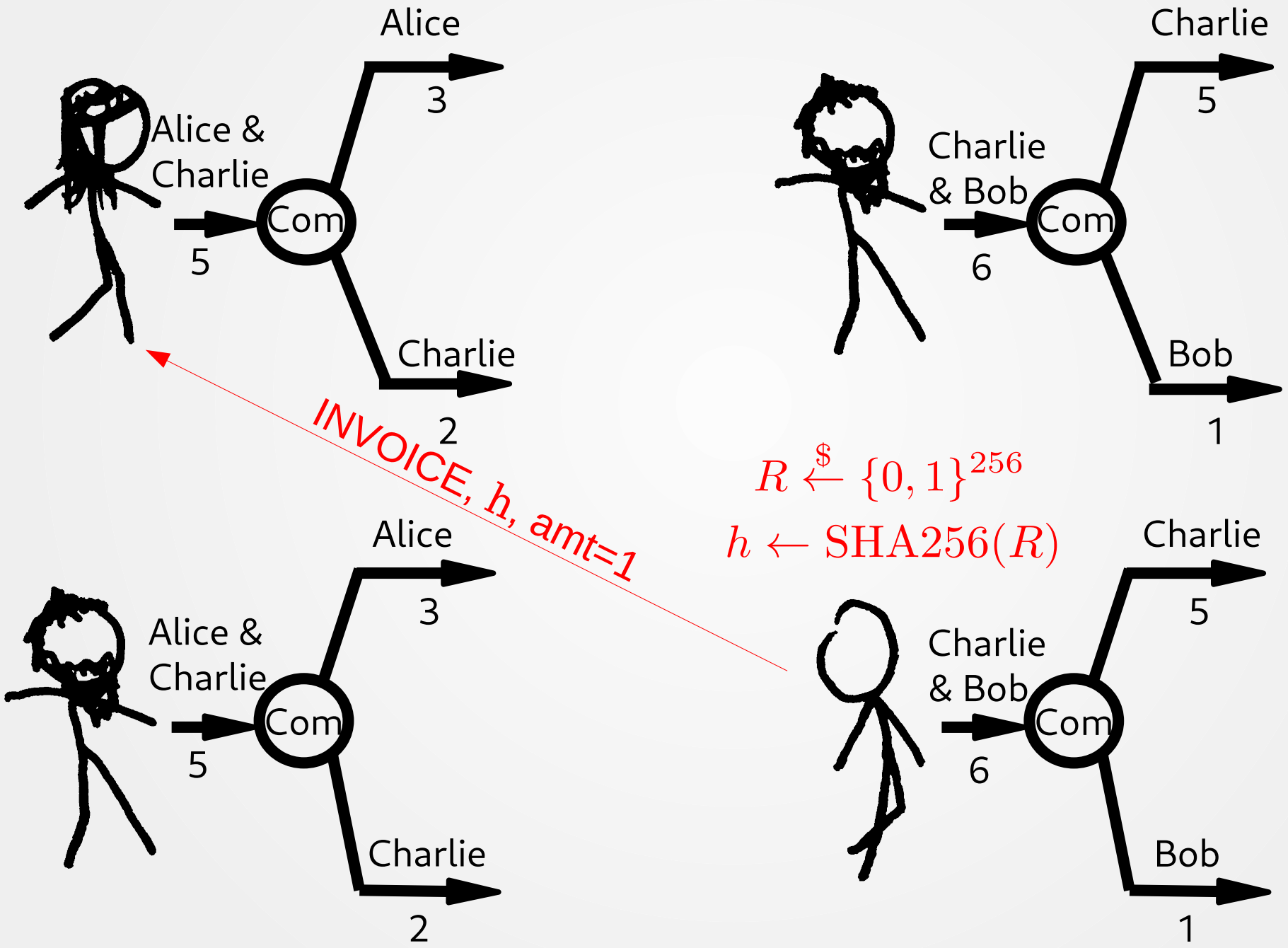

Alice
3
Alice &
Charlie
Com
5
Charlie
2

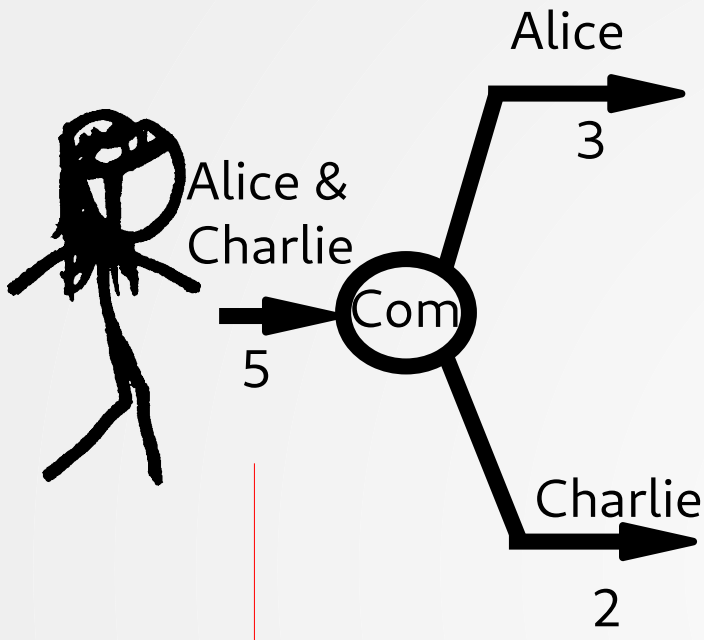

Alice
3
Alice &
Charlie
Com
5
Charlie
2

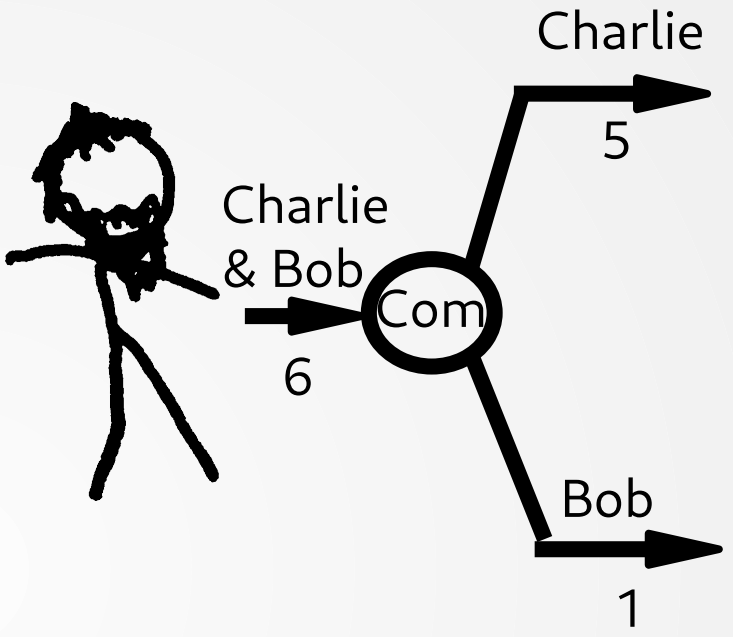

Charlie
5
Charlie
& Bob
Com
6
Bob
1


Alice
3
Alice &
Charlie
Com
5
Charlie
2

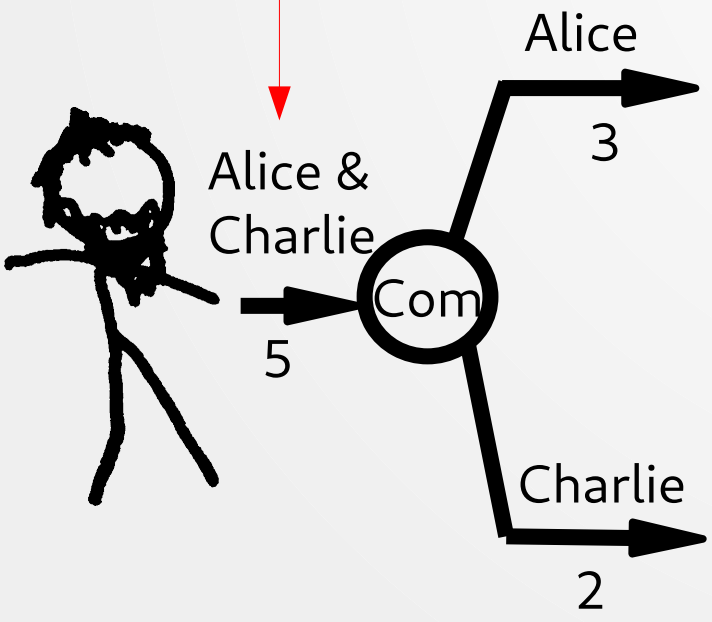

Charlie
5
Charlie
& Bob
Com
6
Bob
1

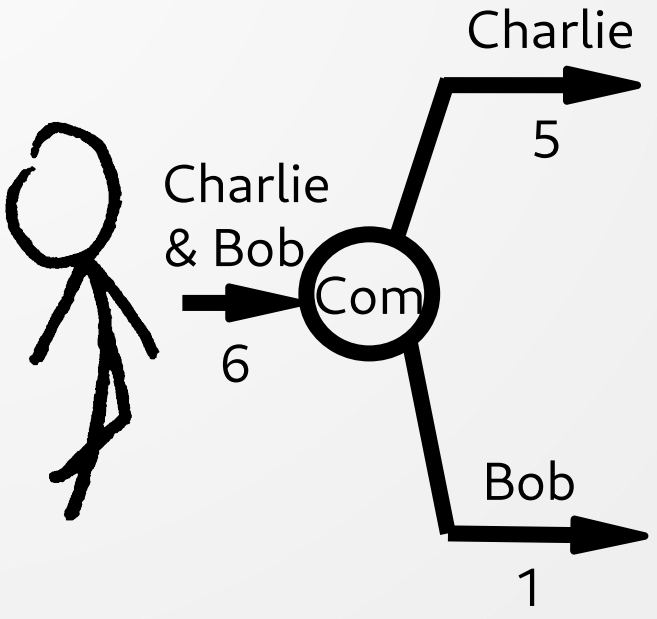

Alice
3
Alice &
Charlie
Com
5
Charlie
2
Charlie
5
Charlie
& Bob
Com
6
Bob
1
INVOICE, h, amt=1
$R \xleftarrow{\$} \{0,1\}^{256}$
$h \leftarrow \mathrm{SHA256}(R)$
Alice
3
Alice &
Charlie
Com
5
Charlie
2
Charlie
5
Charlie
& Bob
Com
6
Bob
1

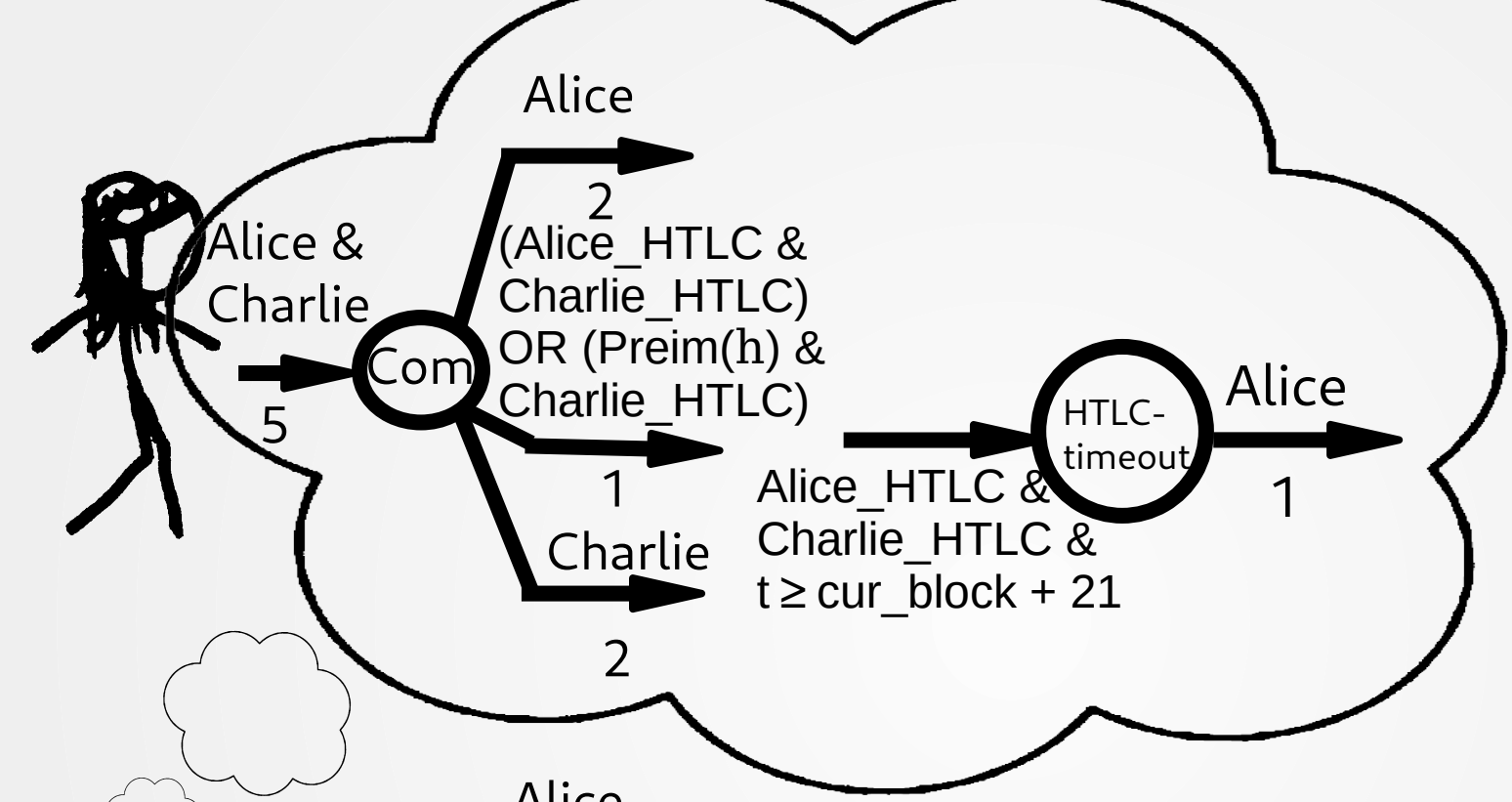


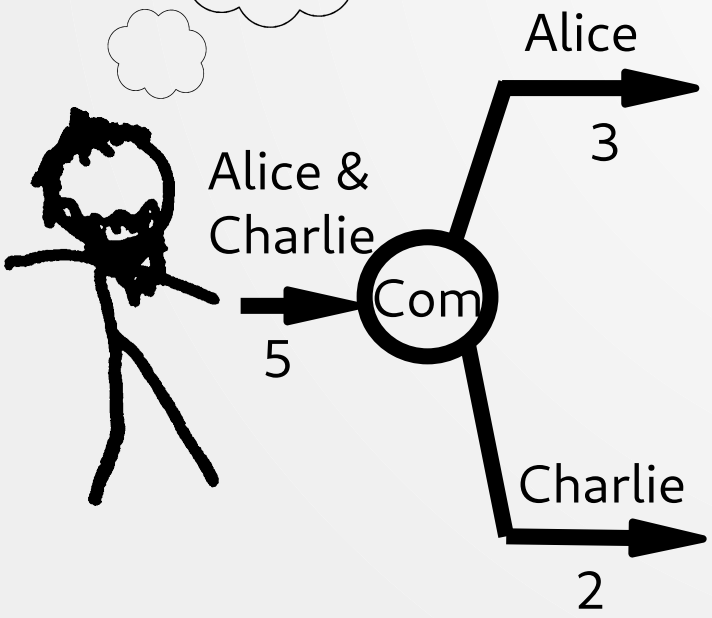


ADD_HTLC, h, amt=1, del=21



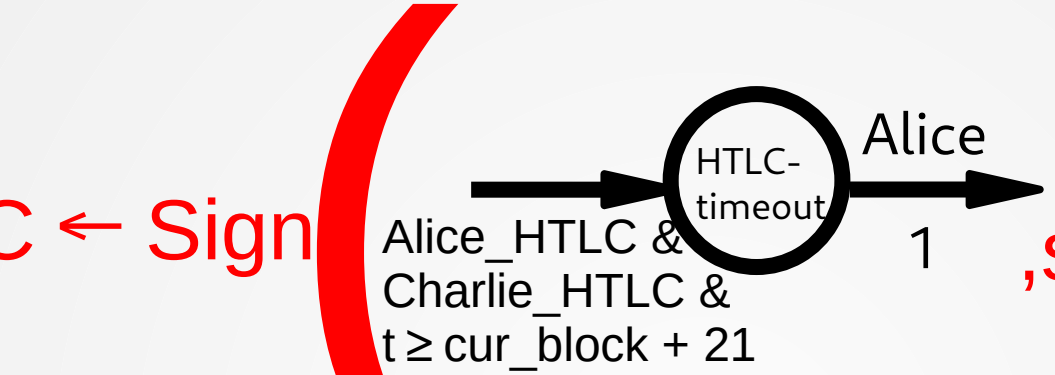






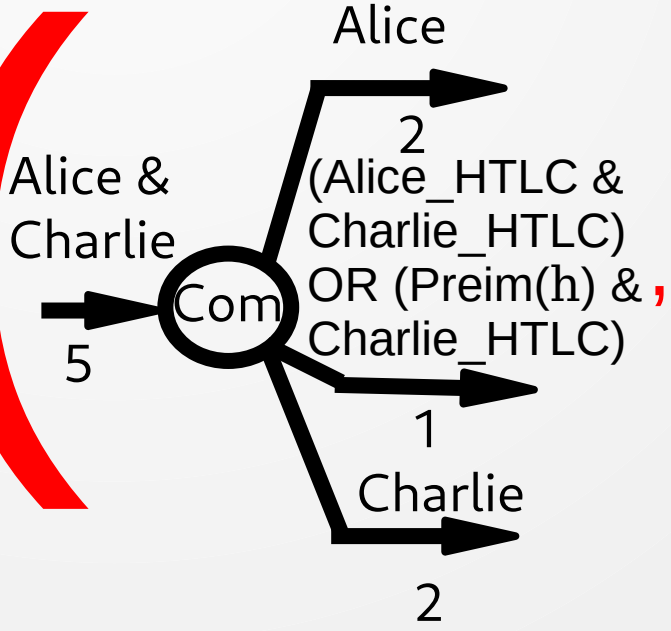
"I, Alice, promise to pay Charlie 1 coin if he reveals a preimage of `0xabcdef` within an hour"

sig_HTLC ← Sign

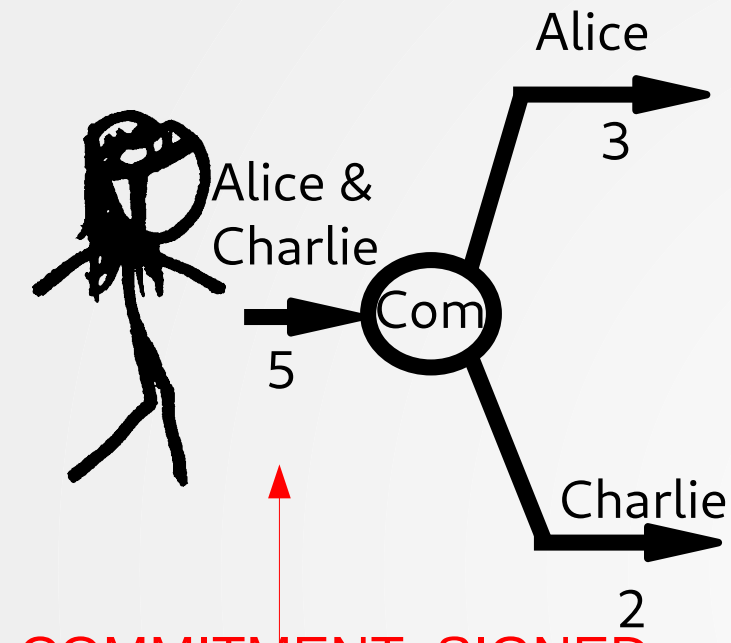


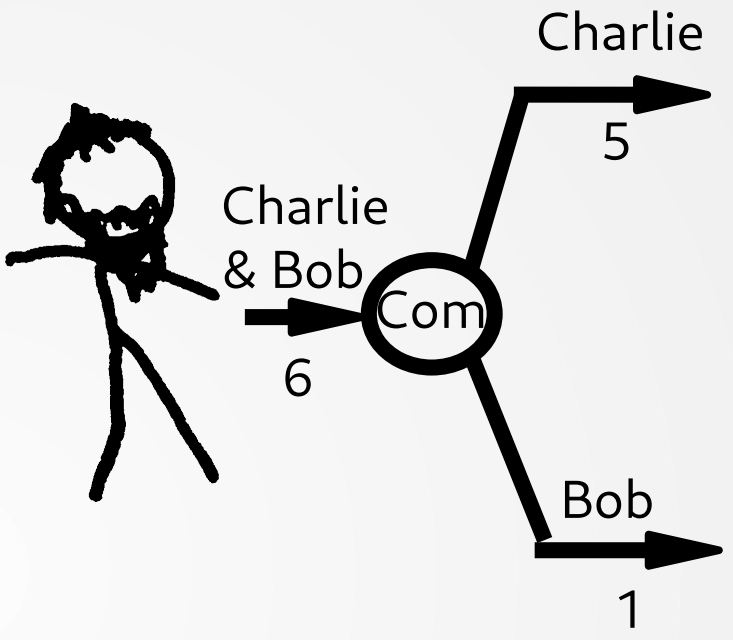
,sk-Charlie-HTLC)

sig_Com ← Sign

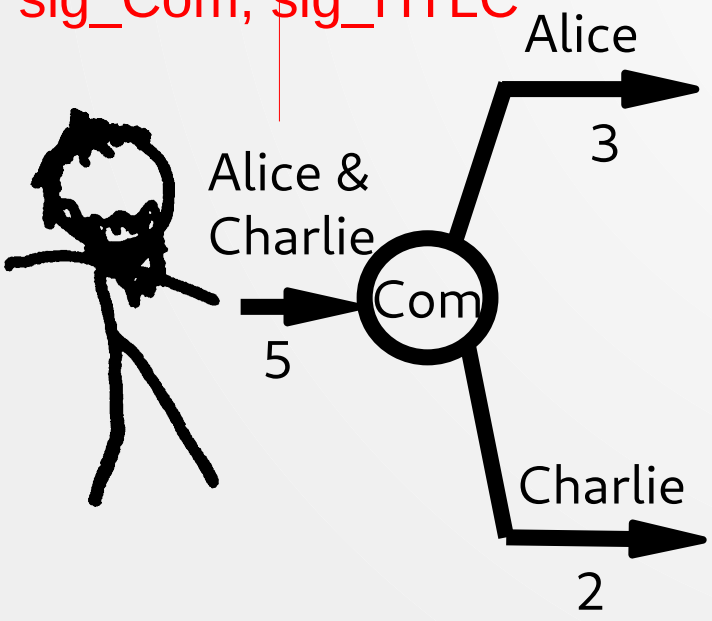


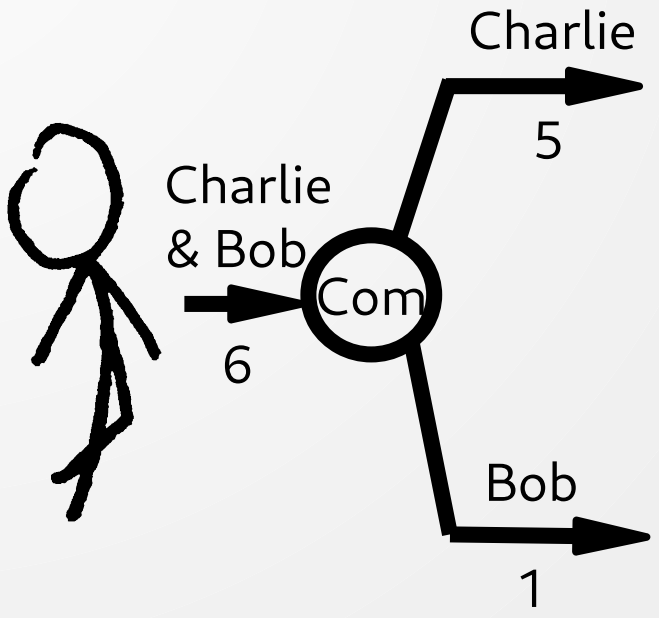
, sk-Charlie)

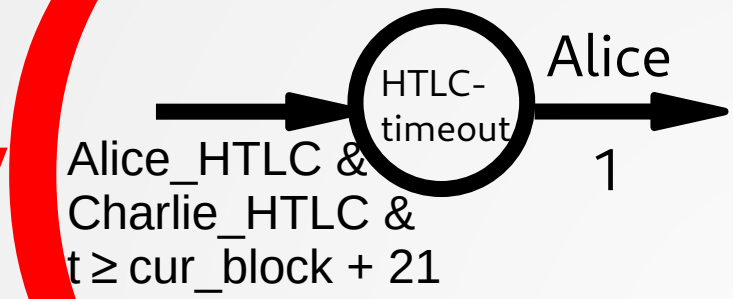




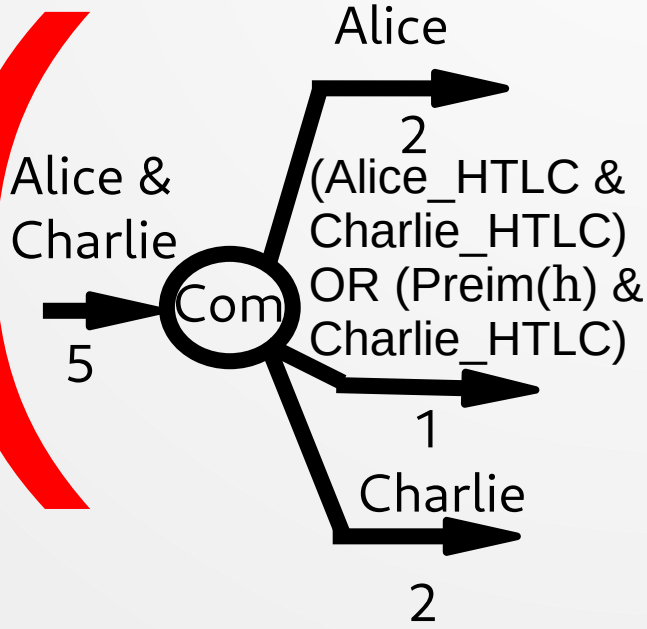
COMMITMENT_SIGNED,
sig_Com, sig_HTLC



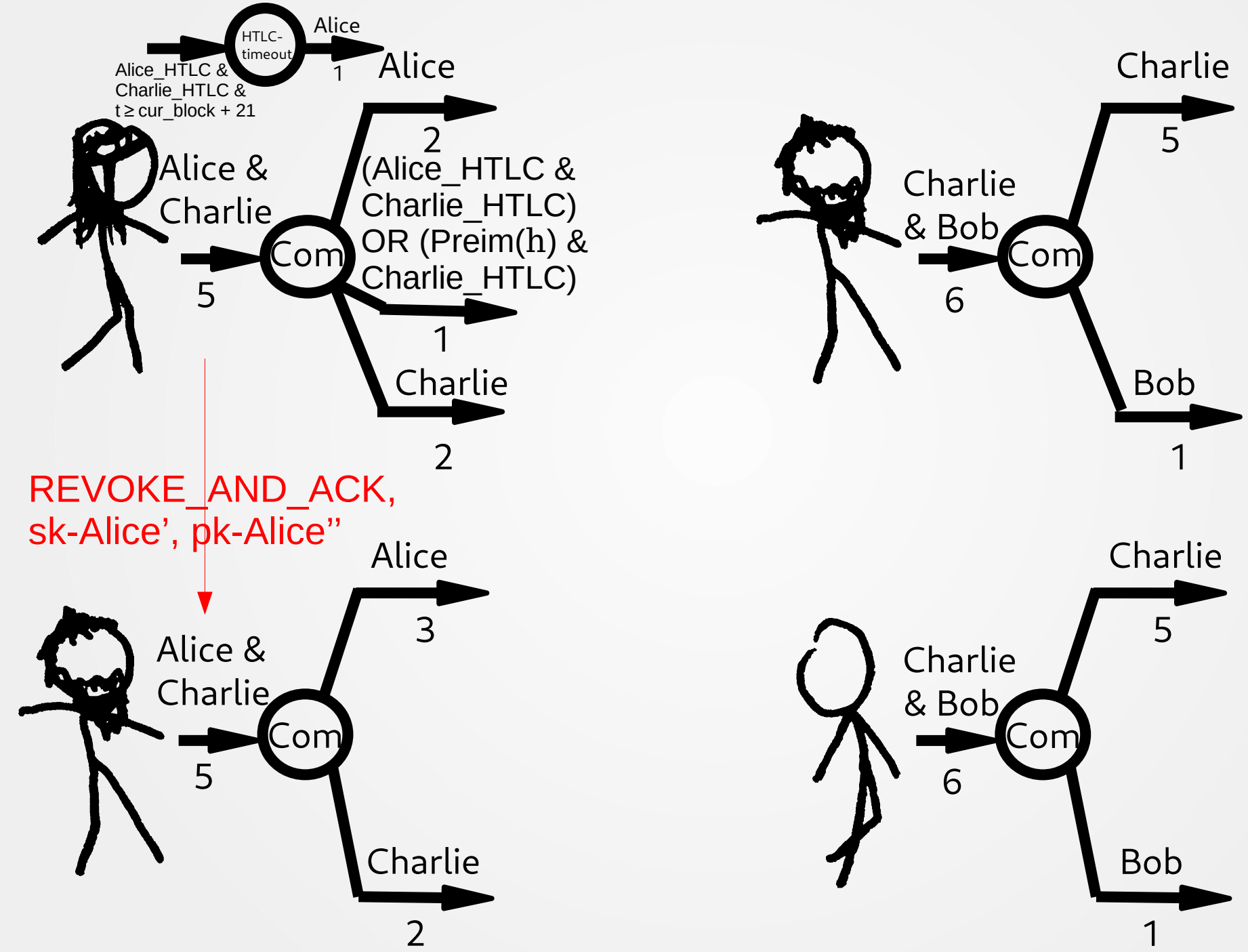


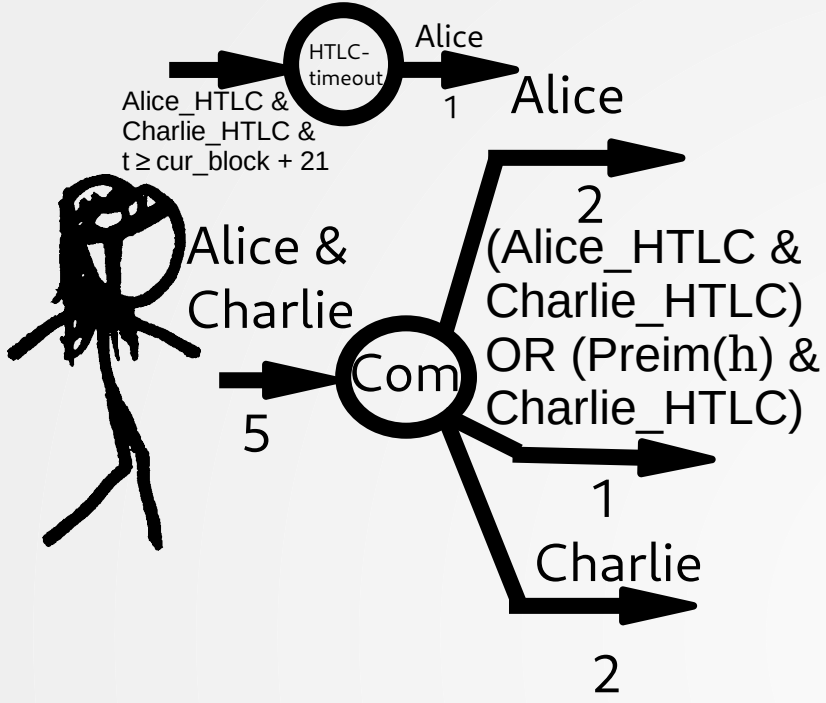
Verify 

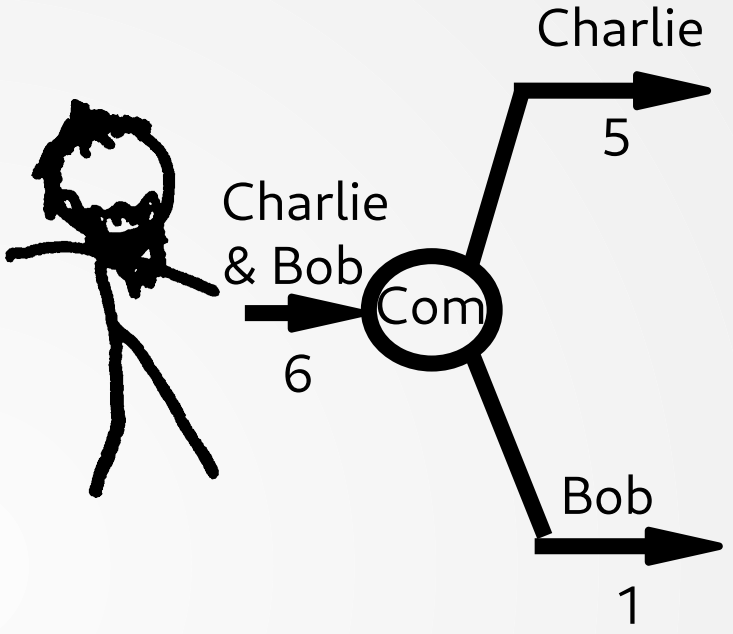
,pk-Charlie-HTLC, sig_HTLC)

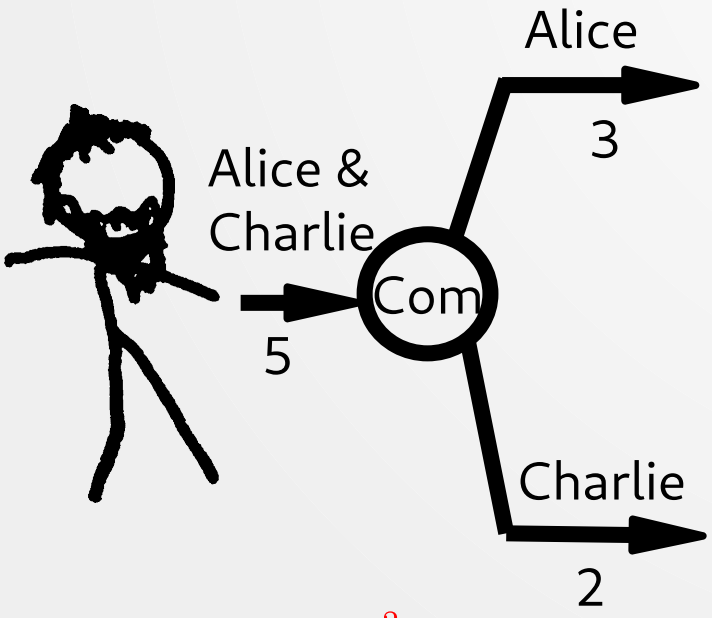
$\stackrel{?}{=}$ Verify (

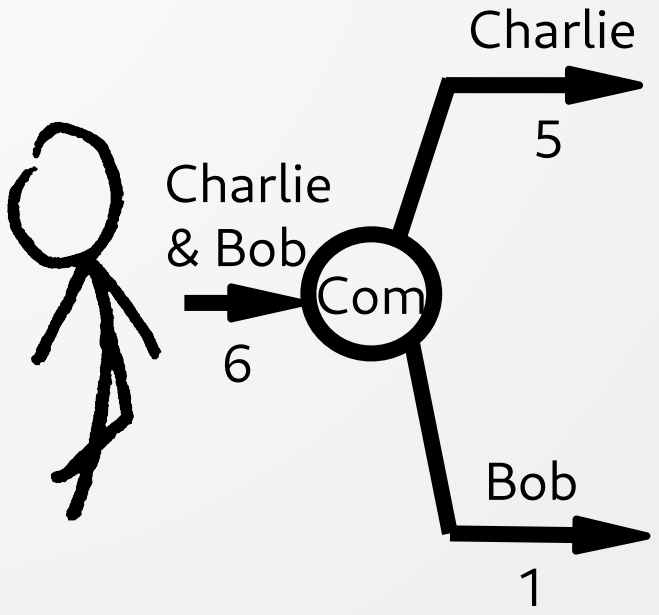
, pk-Charlie, sig_Com) $\stackrel{?}{=} 1$


HTLC-timeout
Alice
1
Alice_HTLC &
Charlie_HTLC &
t ≥ cur_block + 21
Alice
2
Alice &
Charlie
(Alice_HTLC &
Charlie_HTLC)
OR (Preim(h) &
Charlie_HTLC)
Com
5
1
Charlie
2
Charlie
5
Charlie
& Bob
Com
6
Bob
1
REVOKE_AND_ACK,
sk-Alice', pk-Alice''
Alice
3
Alice &
Charlie
Com
5
Charlie
2
Charlie
5
Charlie
& Bob
Com
6
Bob
1

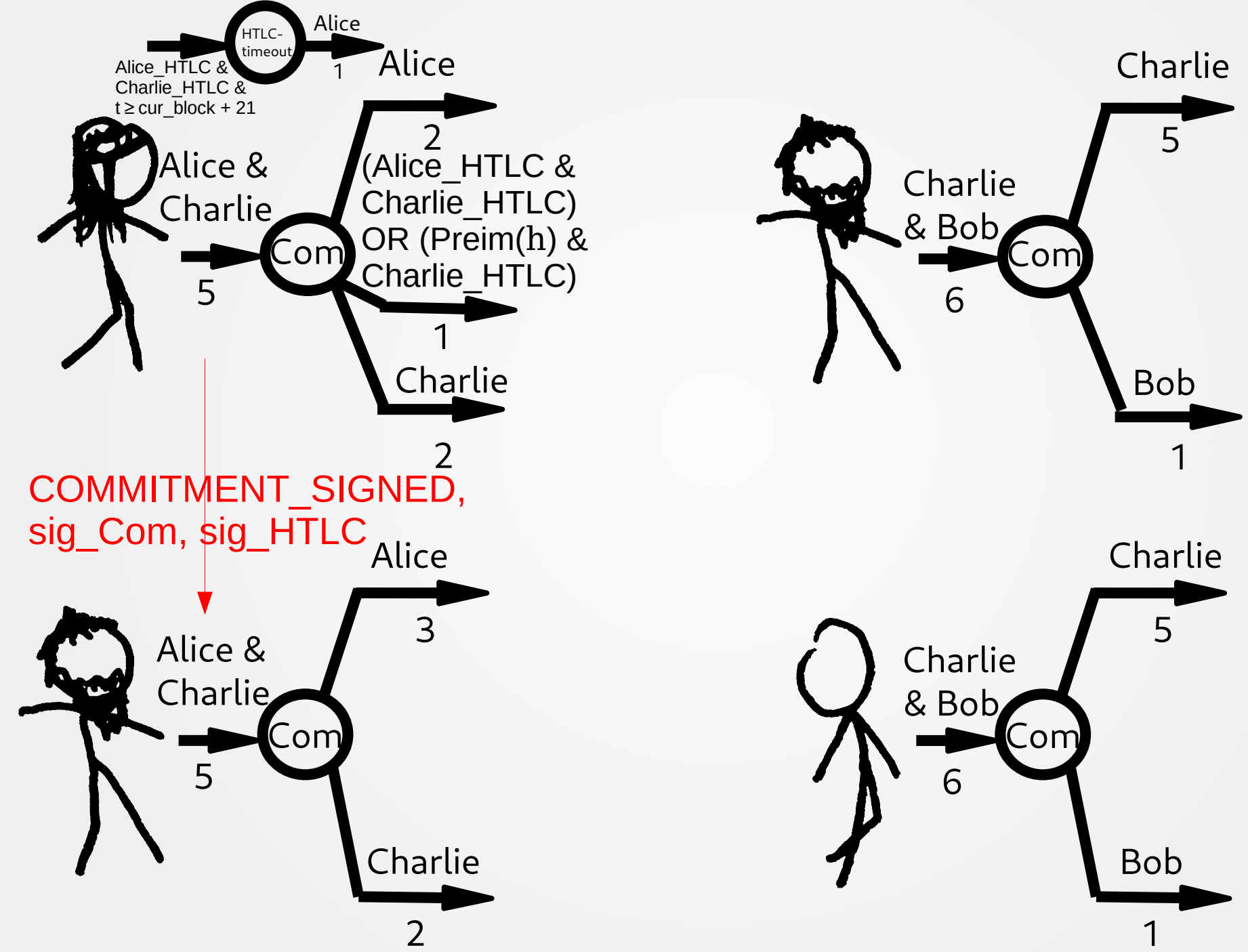


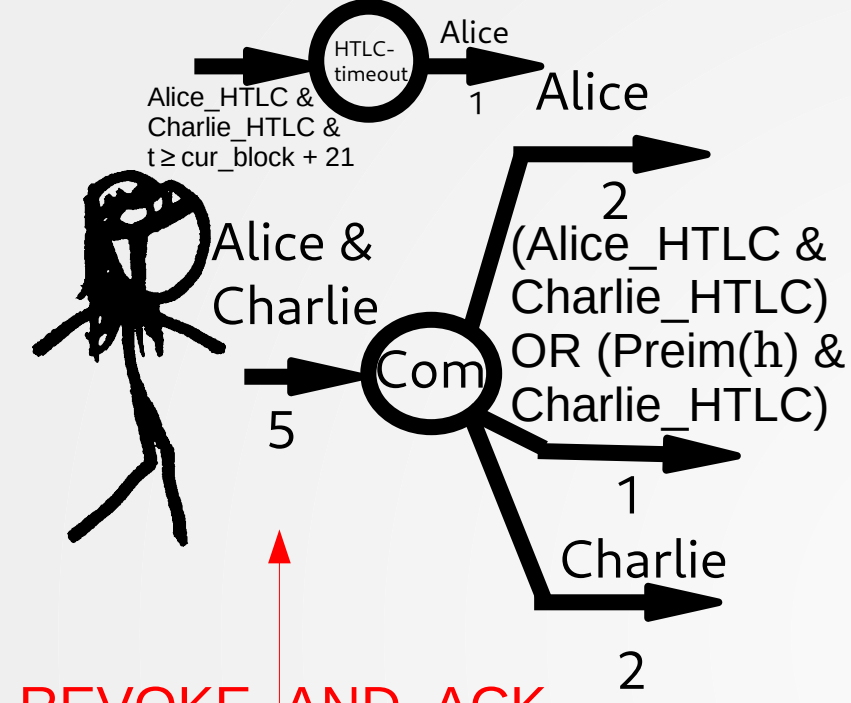


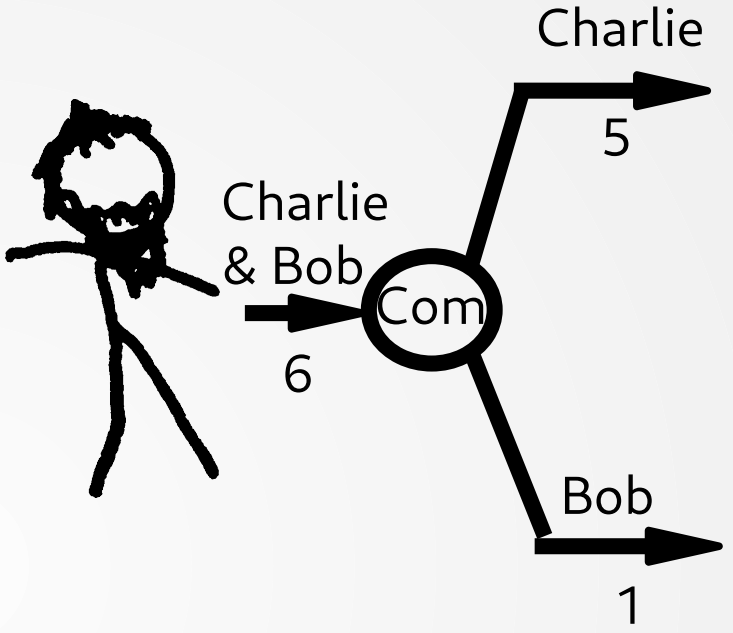




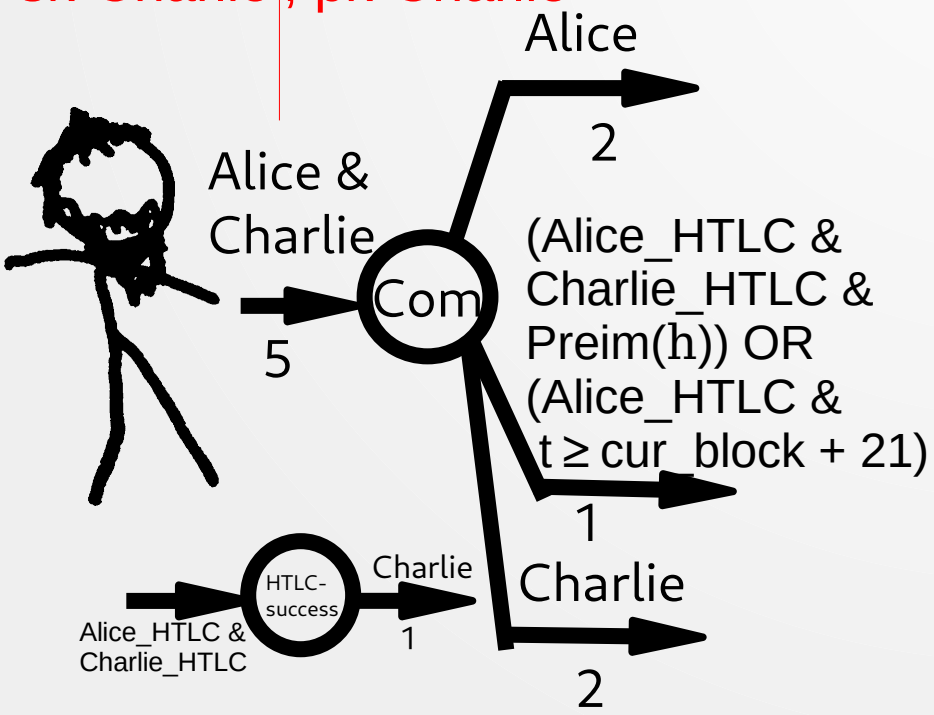
pk(sk-Alice') $\stackrel{?}{=}$ pk-Alice'

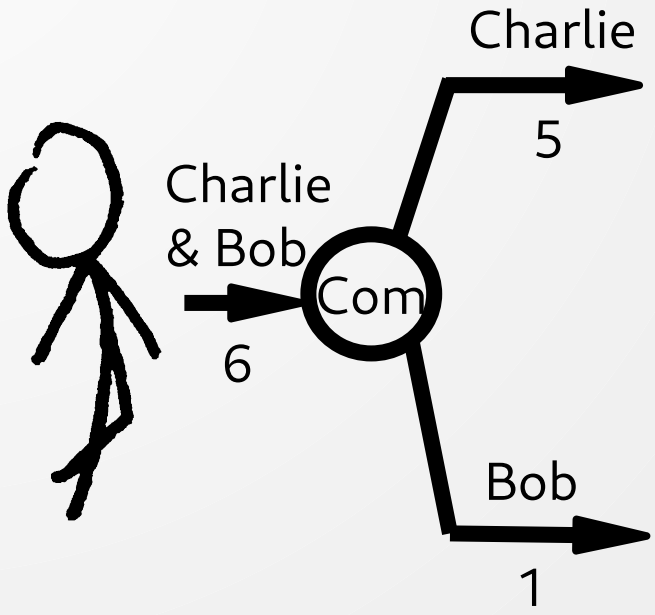

HTLC-timeout
Alice
Alice_HTLC &
Charlie_HTLC &
t ≥ cur_block + 21
1
Alice
2
Alice &
Charlie
(Alice_HTLC &
Charlie_HTLC)
OR (Preim(h) &
Charlie_HTLC)
Com
5
1
Charlie
2
COMMITMENT_SIGNED,
sig_Com, sig_HTLC
Alice
3
Alice &
Charlie
Com
5
Charlie
2
Charlie
5
Charlie
& Bob
Com
6
Bob
1
Charlie
5
Charlie
& Bob
Com
6
Bob
1

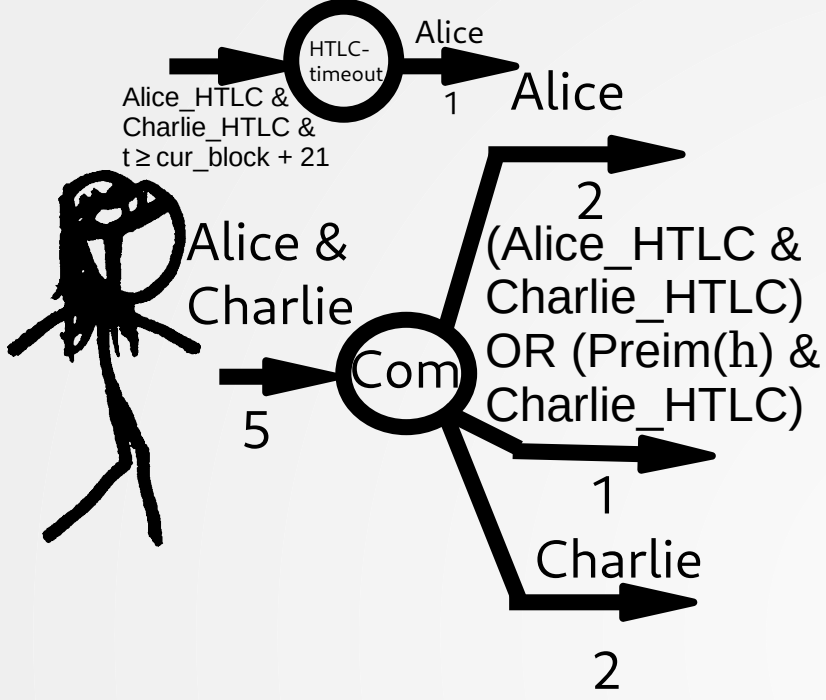


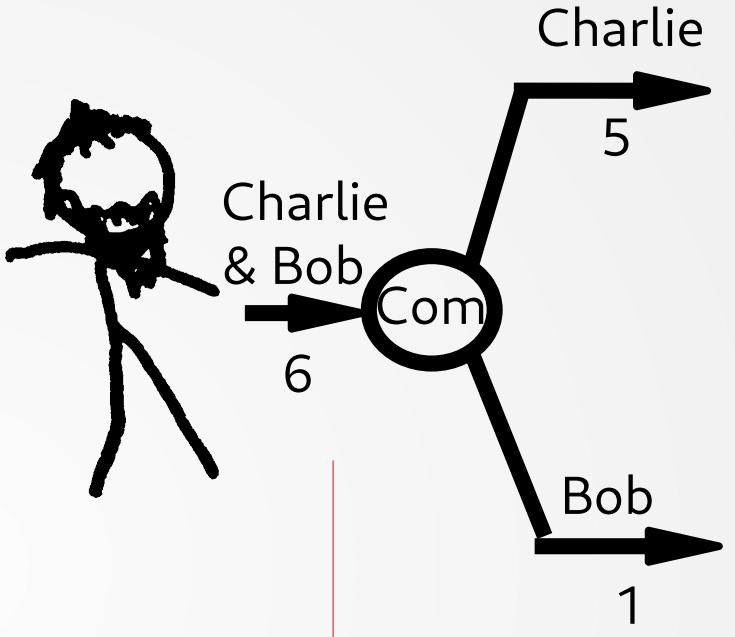


REVOKE_AND_ACK,
sk-Charlie', pk-Charlie''

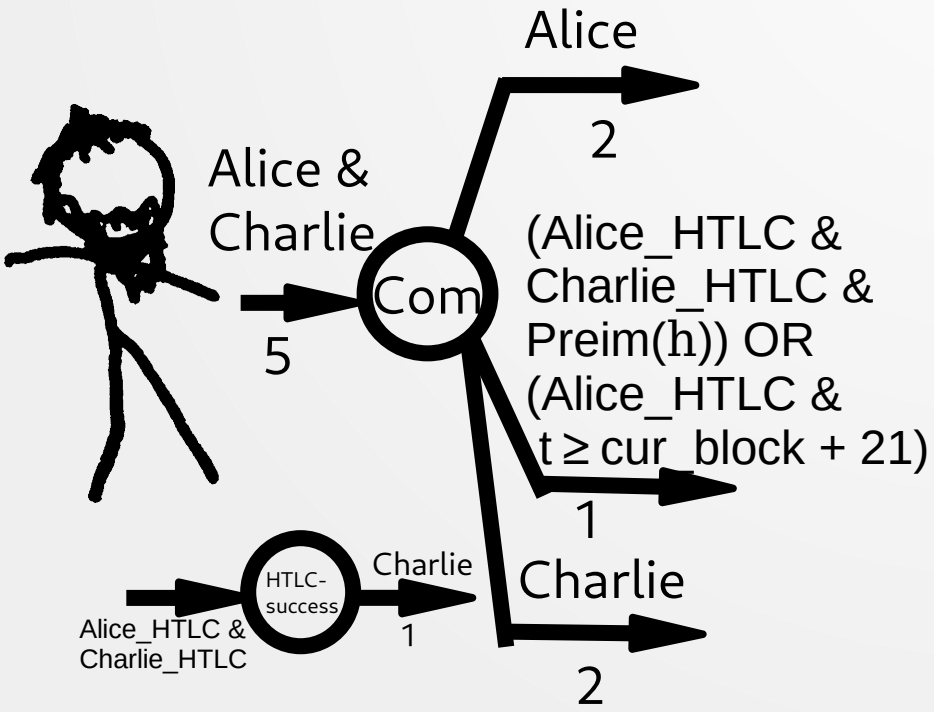


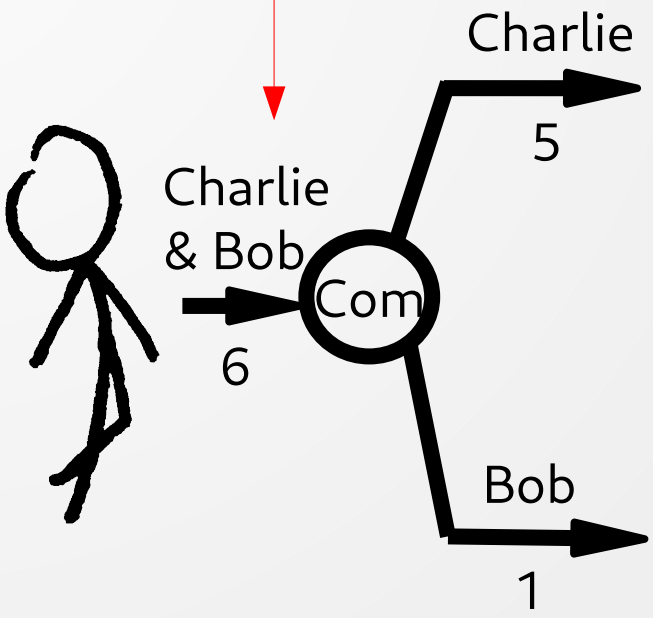


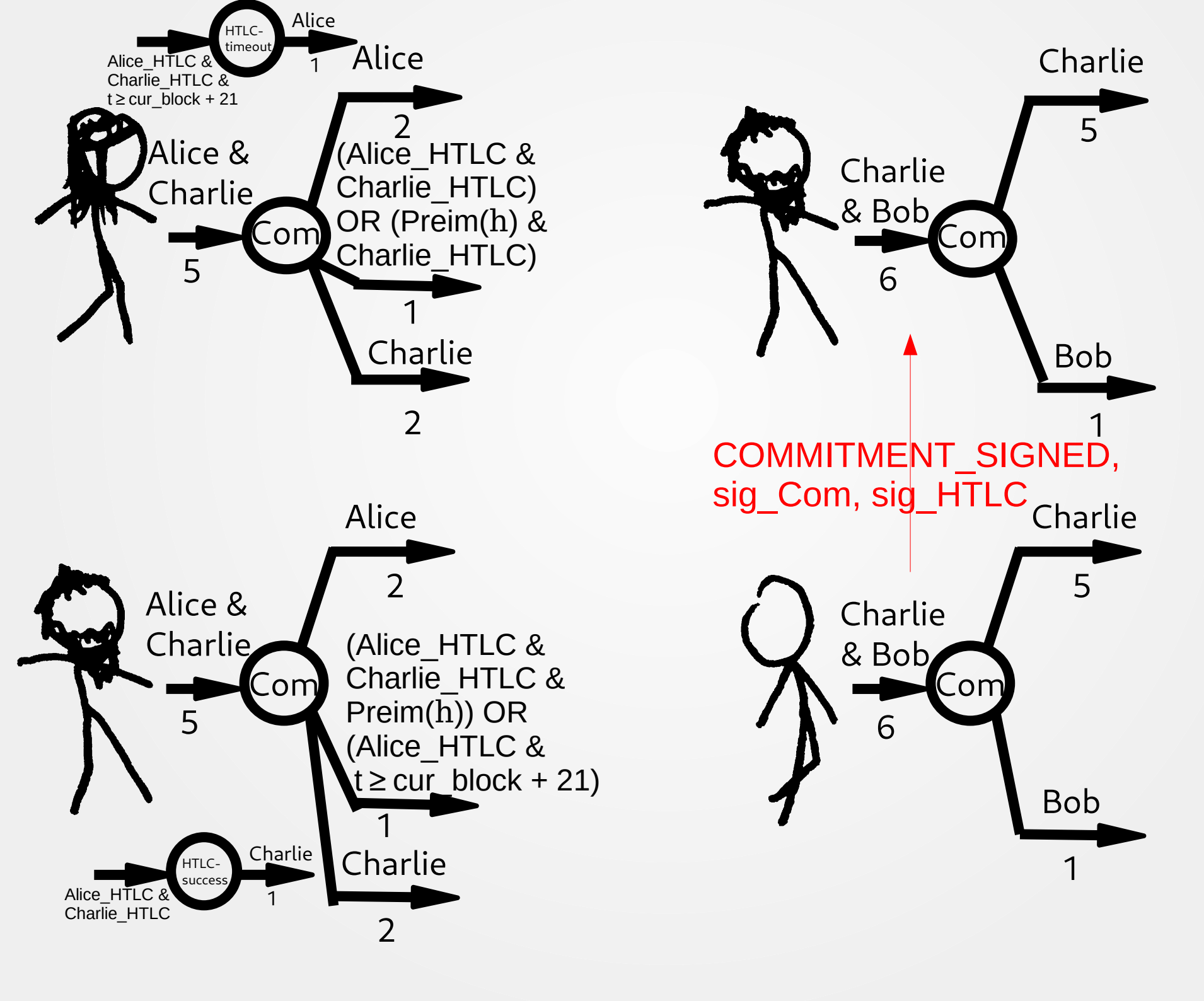


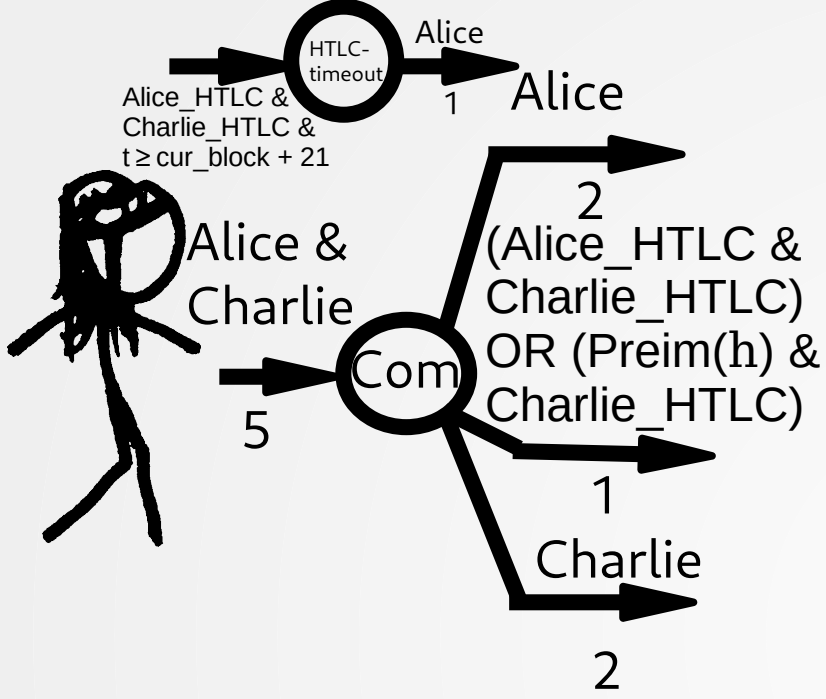


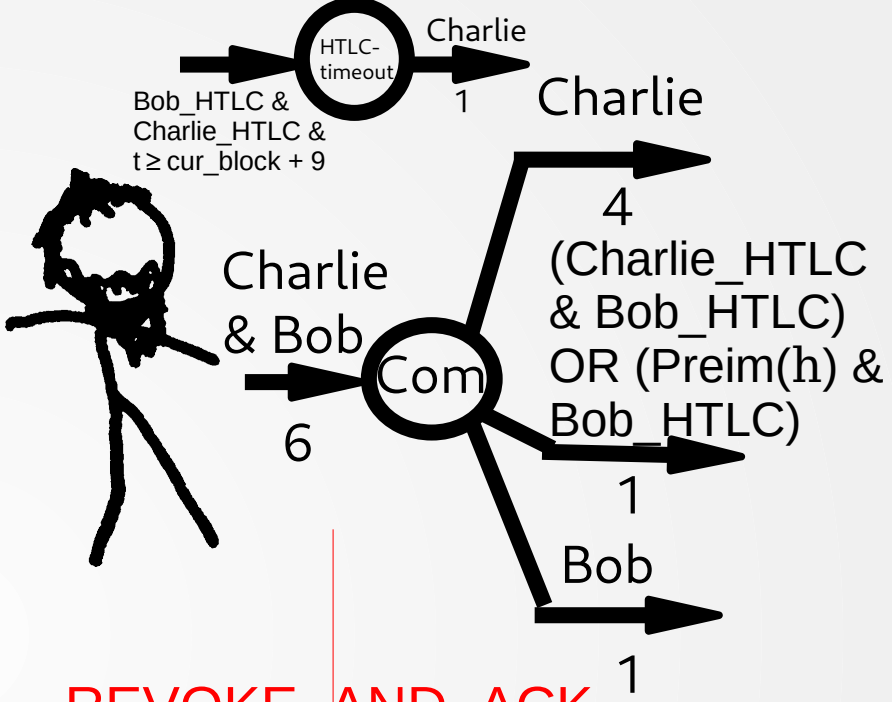
ADD_HTLC, h, amt=1, del=9



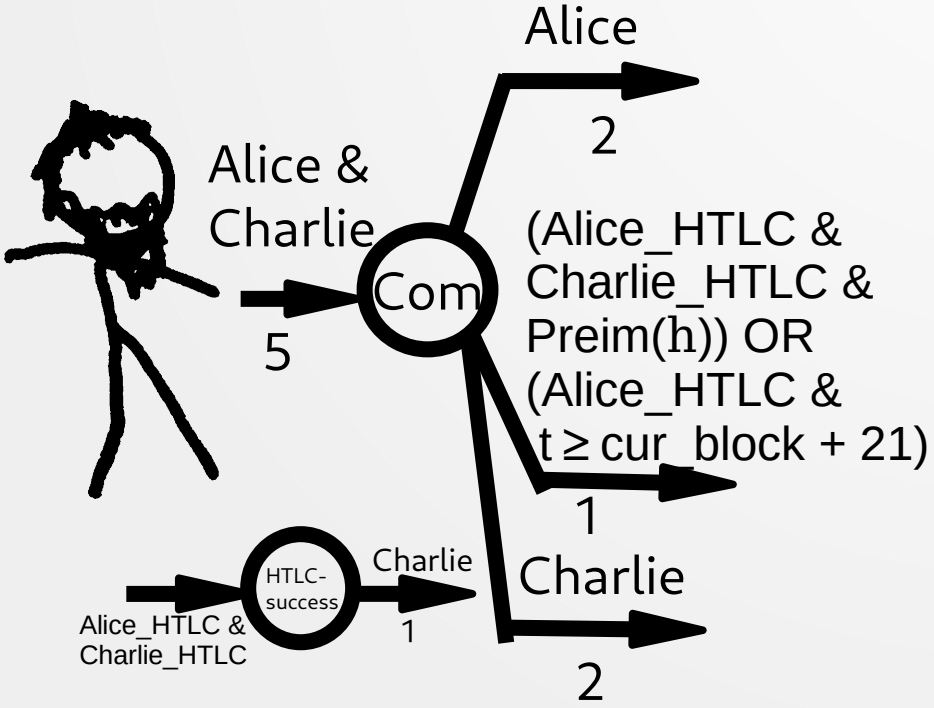


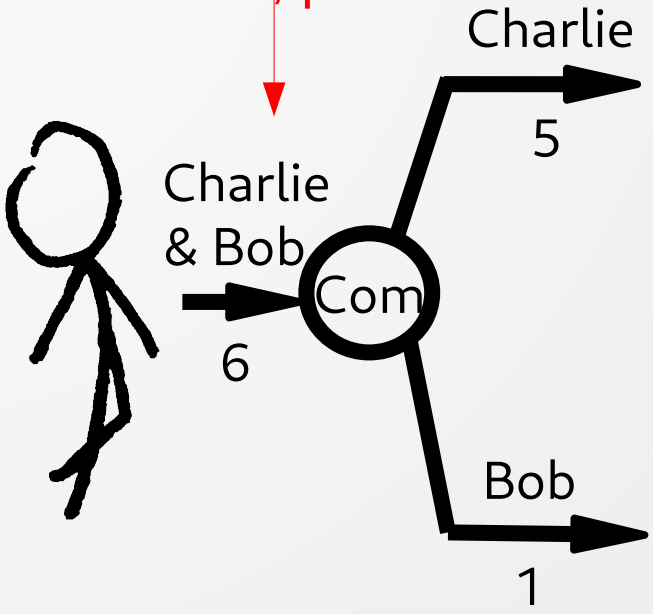

HTLC-timeout
Alice
Alice_HTLC &
Charlie_HTLC &
t ≥ cur_block + 21
1
Alice
2
Alice &
Charlie
Com
(Alice_HTLC &
Charlie_HTLC)
OR (Preim(h) &
Charlie_HTLC)
5
1
Charlie
2
Charlie
5
Charlie
& Bob
Com
6
Bob
1
COMMITMENT_SIGNED,
sig_Com, sig_HTLC
Alice
2
Alice &
Charlie
Com
(Alice_HTLC &
Charlie_HTLC &
Preim(h)) OR
(Alice_HTLC &
t ≥ cur_block + 21)
5
1
HTLC-success
Charlie
Alice_HTLC &
Charlie_HTLC
1
Charlie
2
Charlie
5
Charlie
& Bob
Com
6
Bob
1

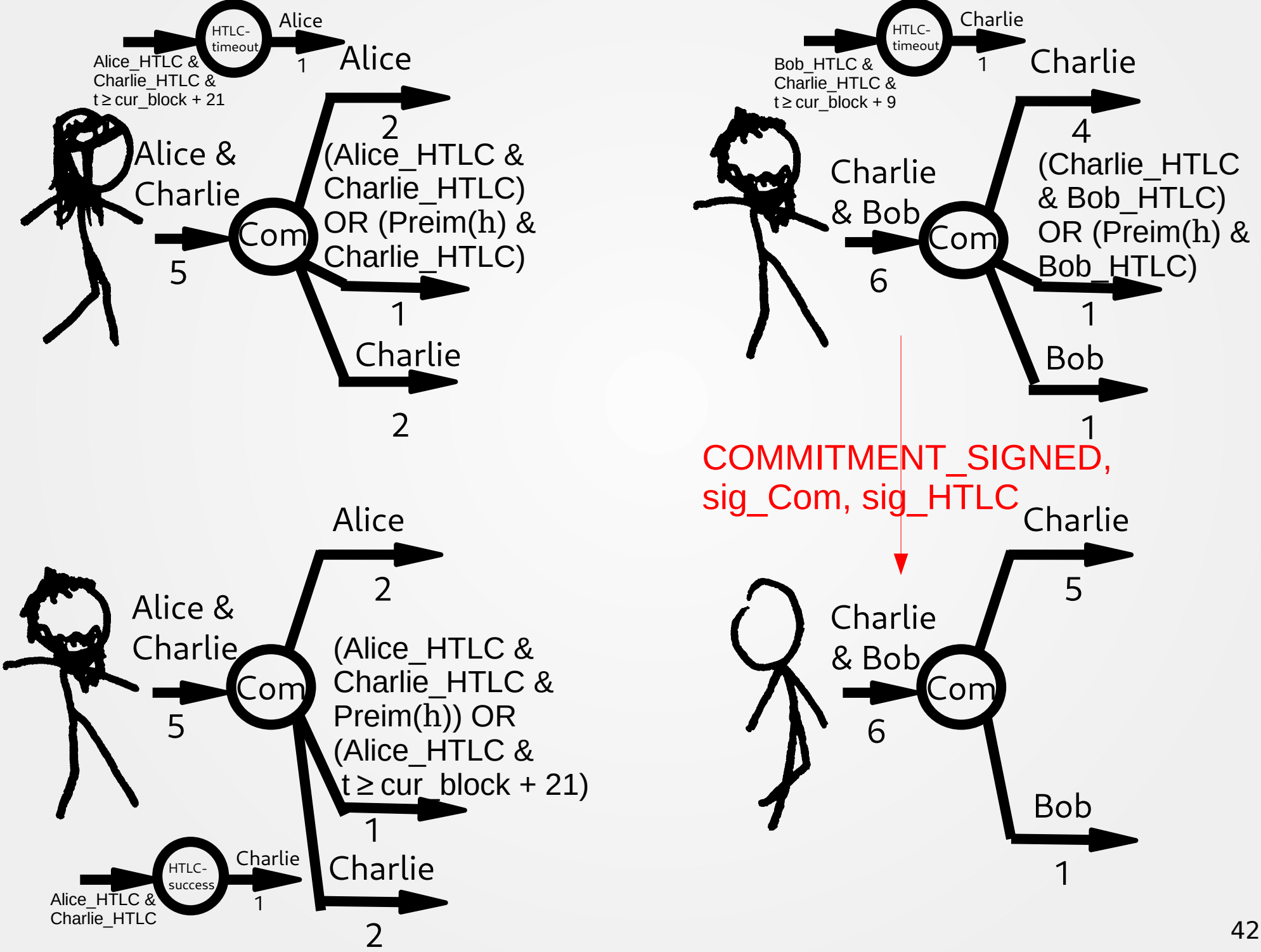


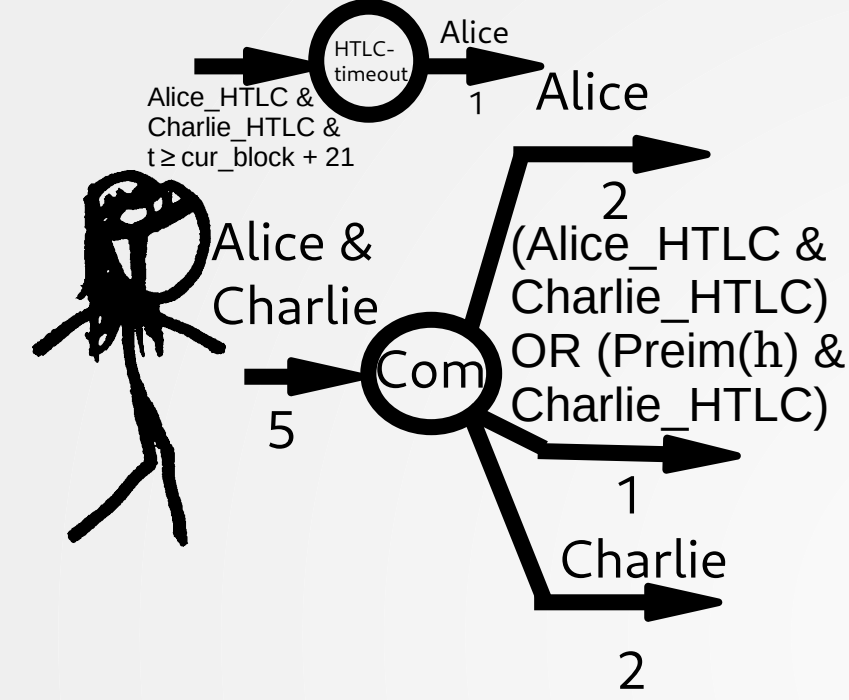


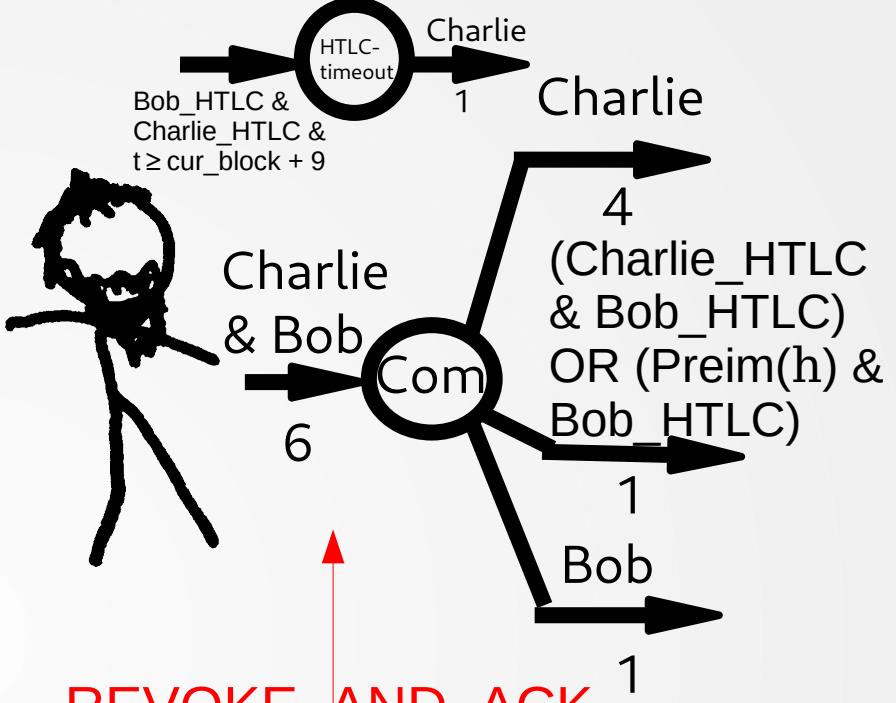
REVOKE_AND_ACK,
sk-Charlie', pk-Charlie''



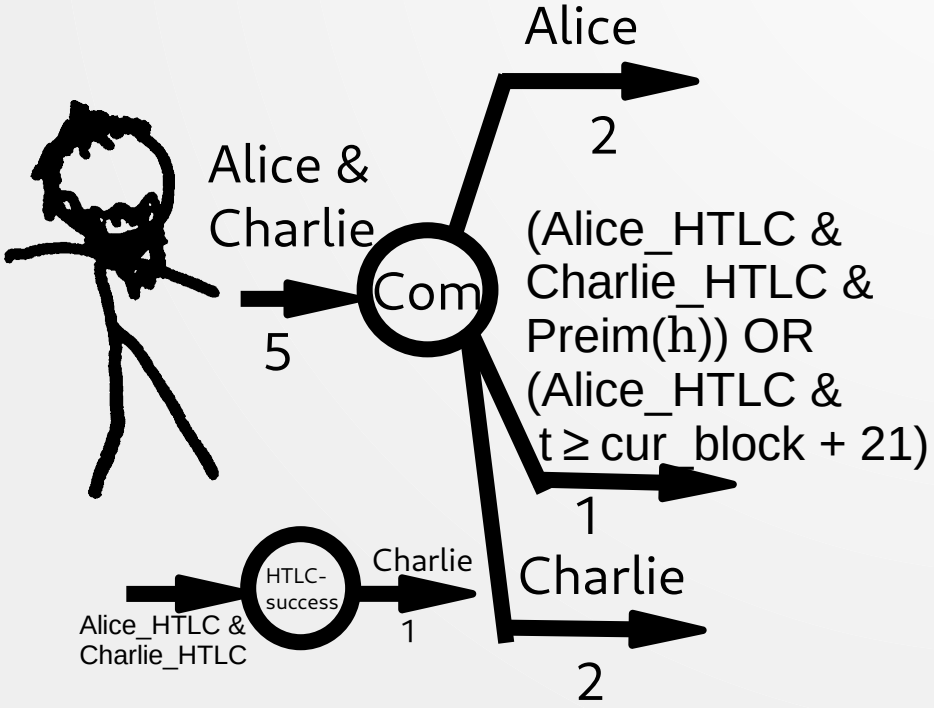


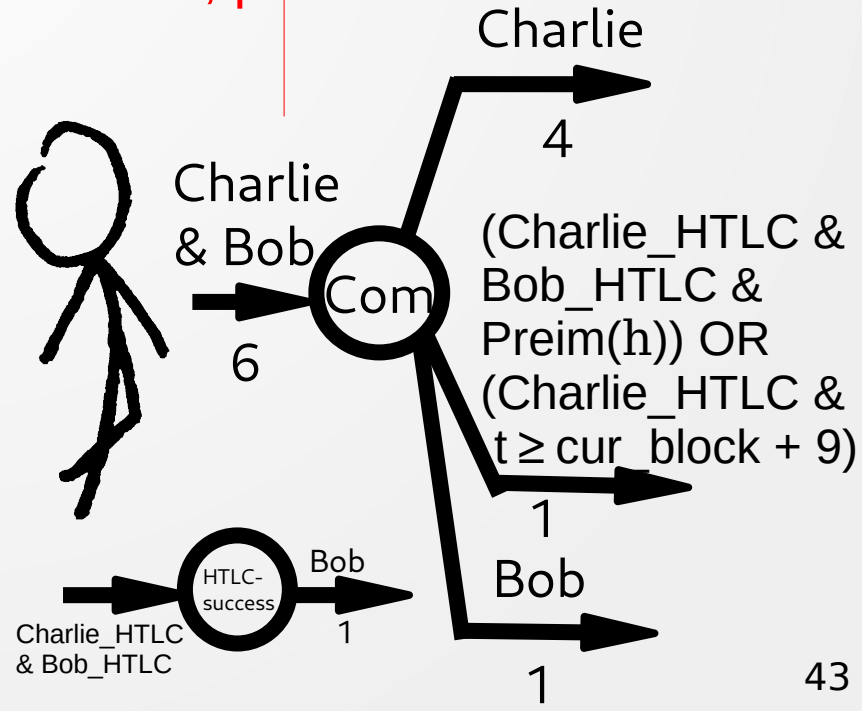

HTLC-timeout
Alice
Alice_HTLC &
Charlie_HTLC &
t ≥ cur_block + 21
1
Alice
2
Alice &
Charlie
Com
(Alice_HTLC &
Charlie_HTLC)
OR (Preim(h) &
Charlie_HTLC)
5
1
Charlie
2
HTLC-timeout
Charlie
Bob_HTLC &
Charlie_HTLC &
t ≥ cur_block + 9
1
Charlie
4
Charlie
& Bob
Com
(Charlie_HTLC
& Bob_HTLC)
OR (Preim(h) &
Bob_HTLC)
6
1
Bob
1
COMMITMENT_SIGNED,
sig_Com, sig_HTLC
Alice
2
Alice &
Charlie
Com
(Alice_HTLC &
Charlie_HTLC &
Preim(h)) OR
(Alice_HTLC &
t ≥ cur_block + 21)
5
1
HTLC-success
Charlie
Alice_HTLC &
Charlie_HTLC
1
Charlie
2
Charlie
5
Charlie
& Bob
Com
6
Bob
1

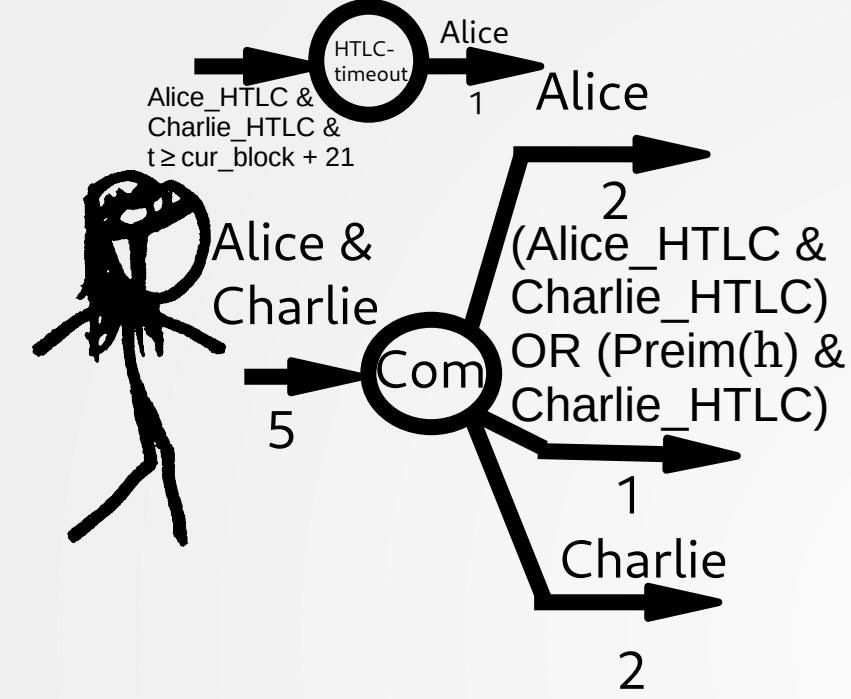


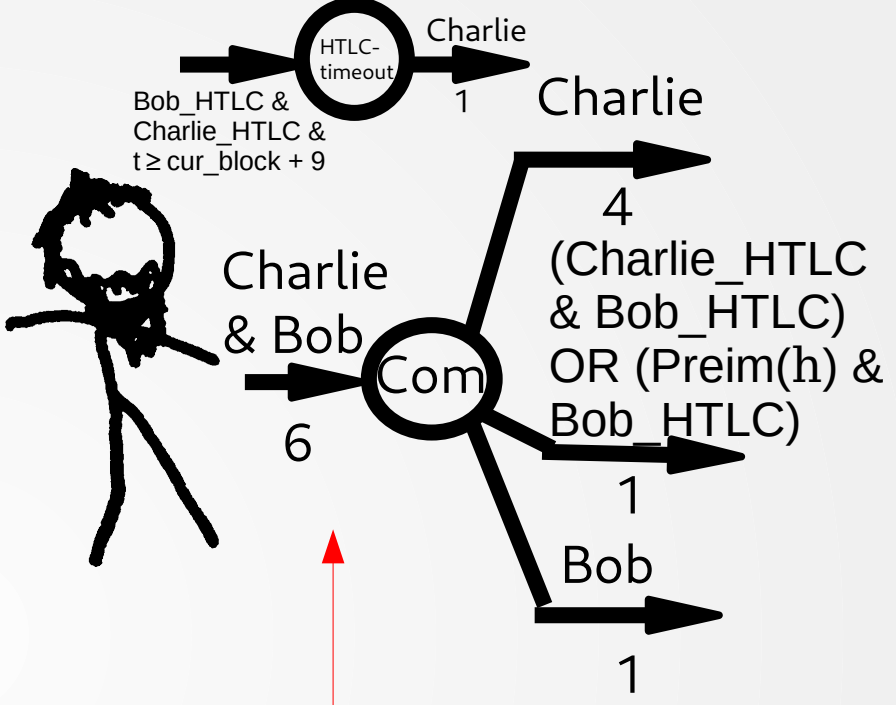


REVOKE_AND_ACK,
sk-Bob', pk-Bob''

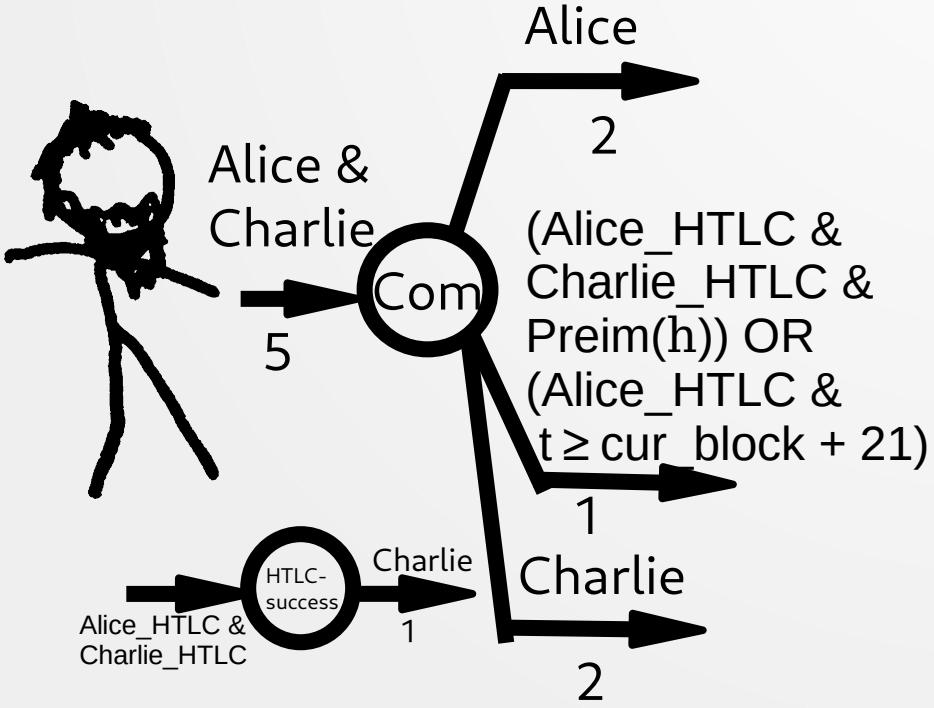


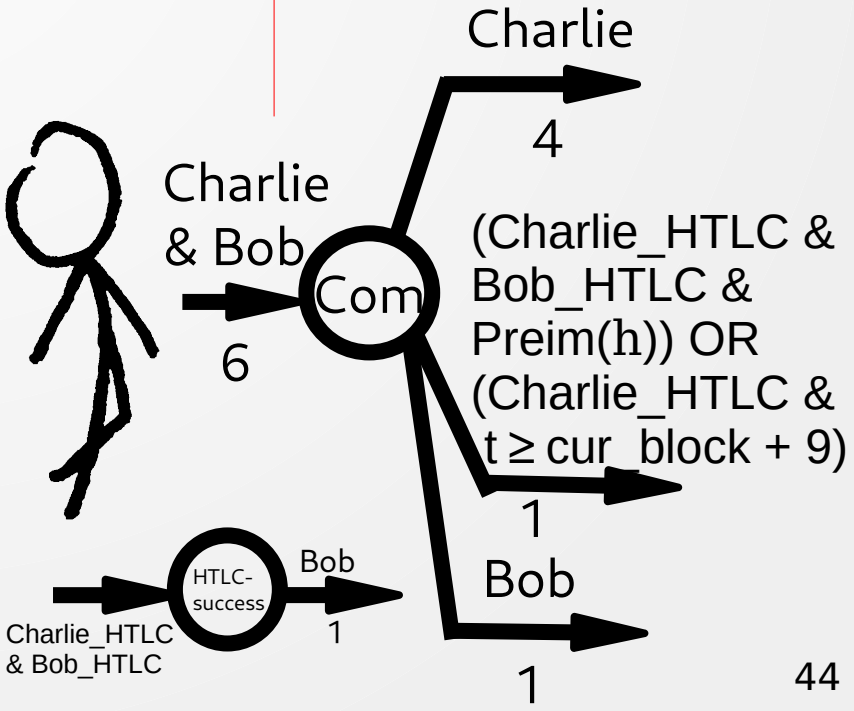


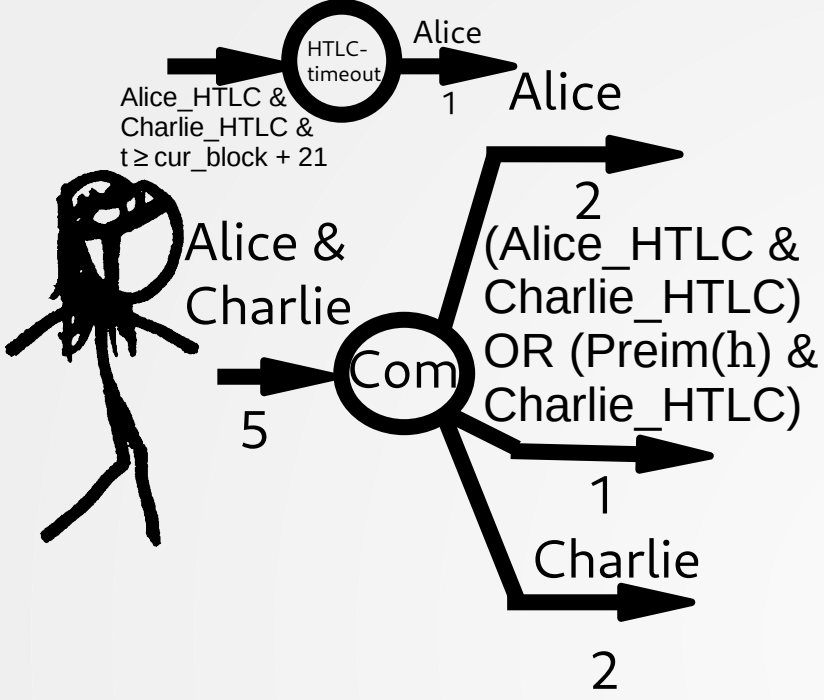


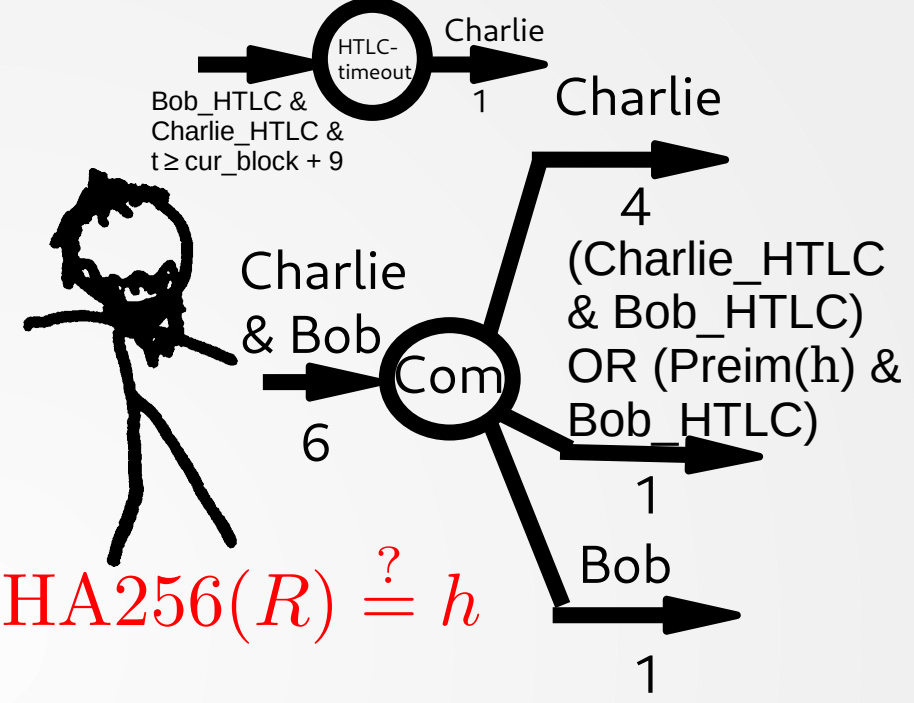


FULFILL_HTLC, R

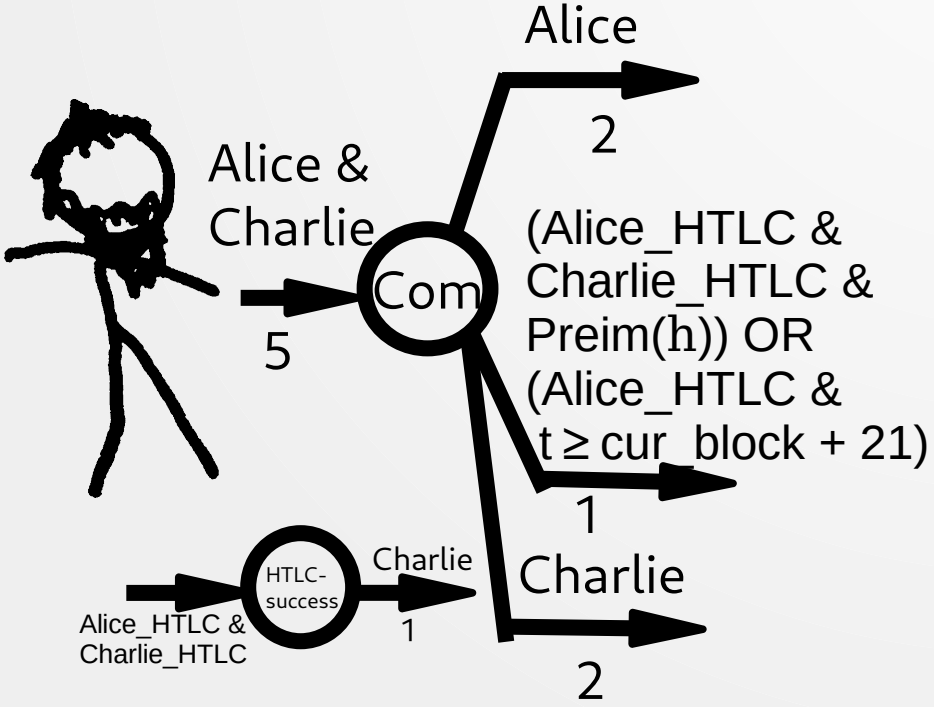


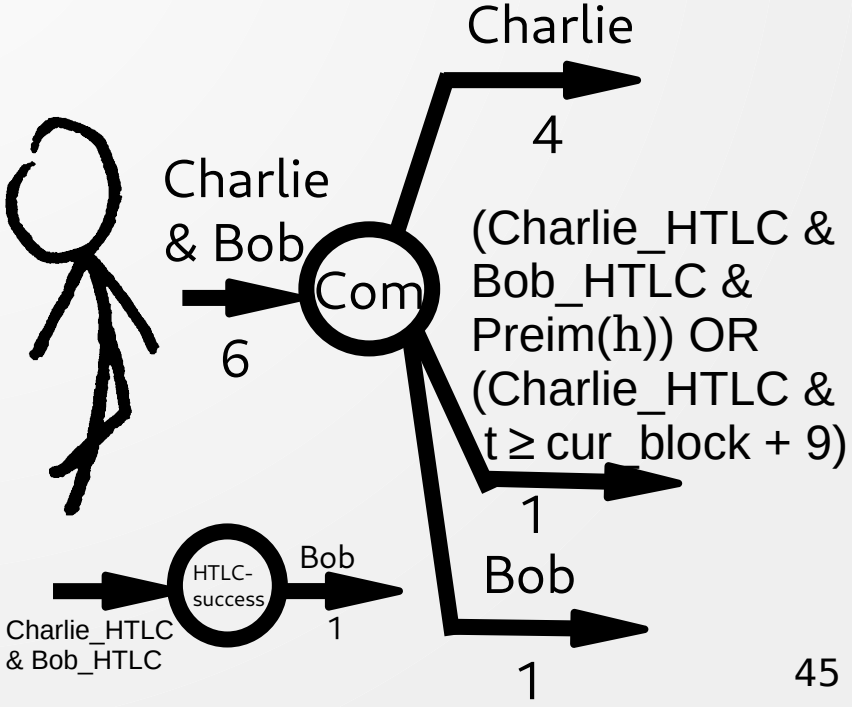




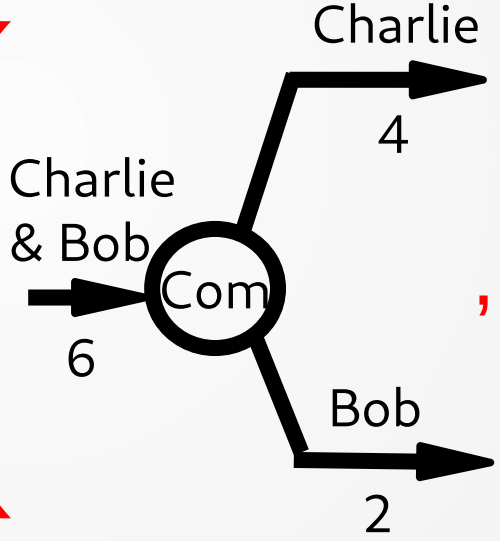


$$\mathrm{SHA256}(R) \stackrel{?}{=} h$$

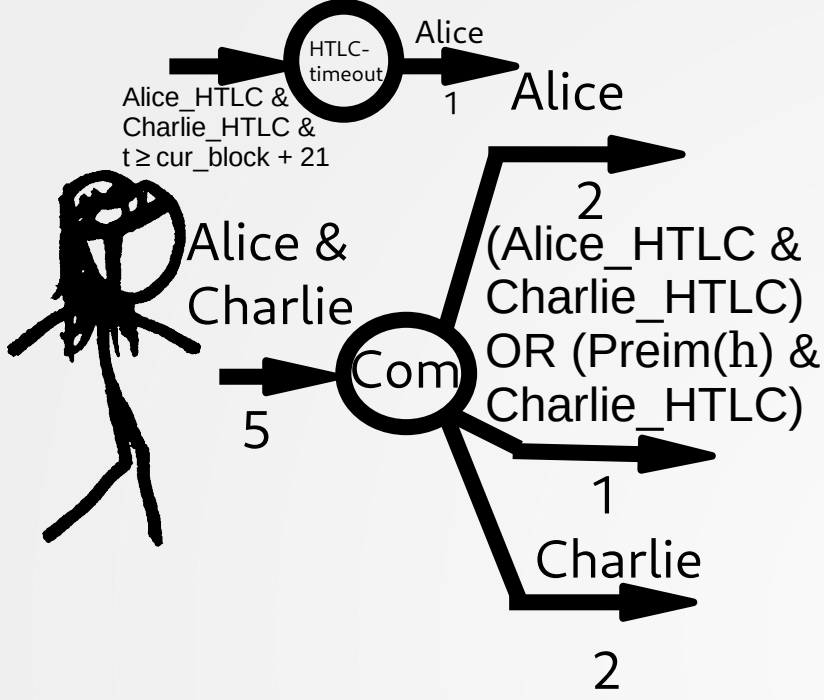


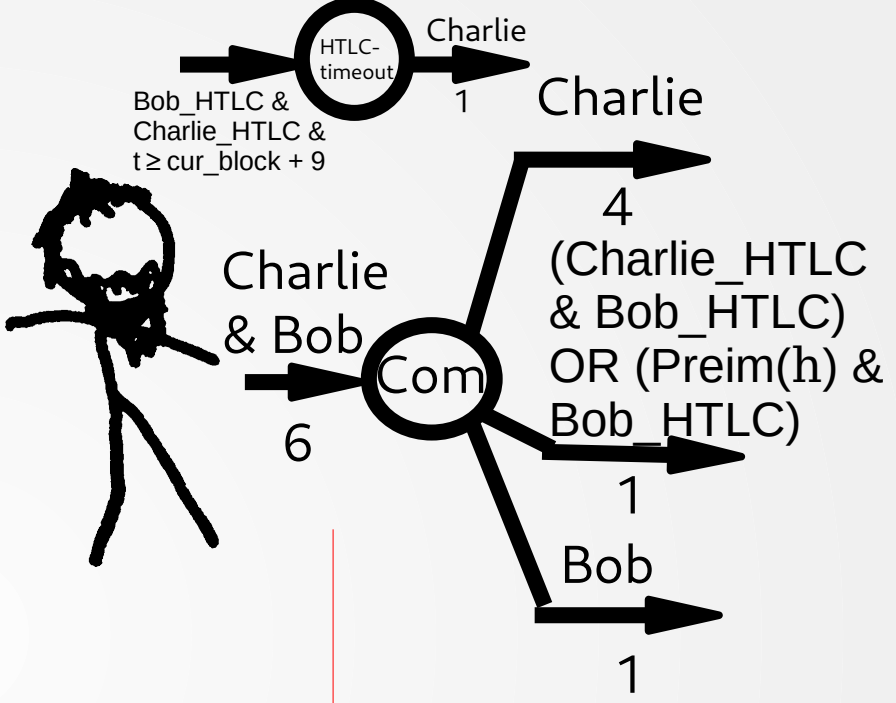


sig_Com ← Sign(

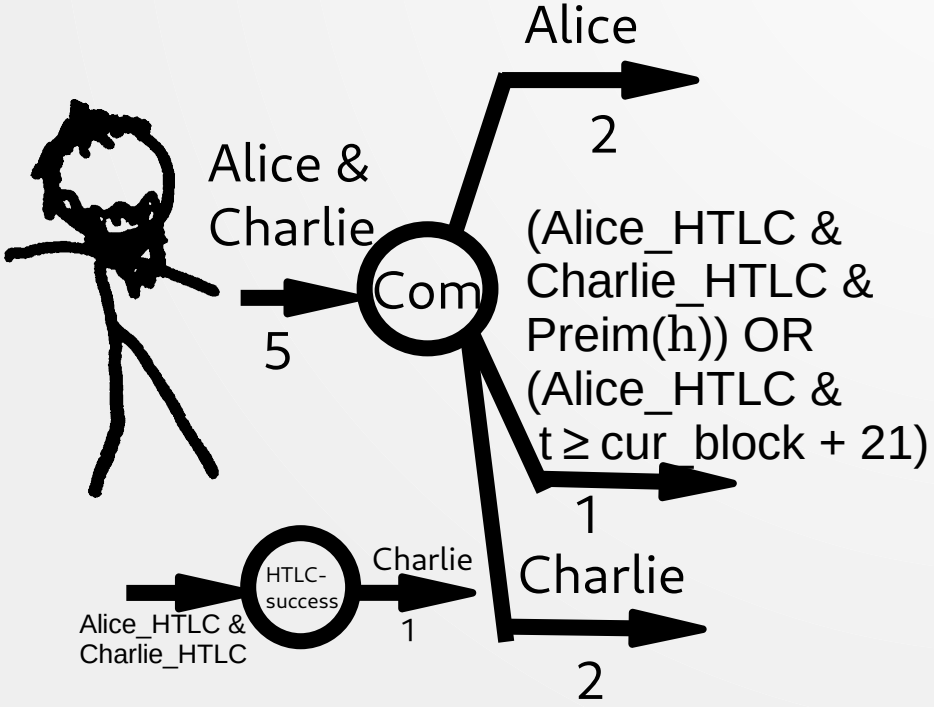


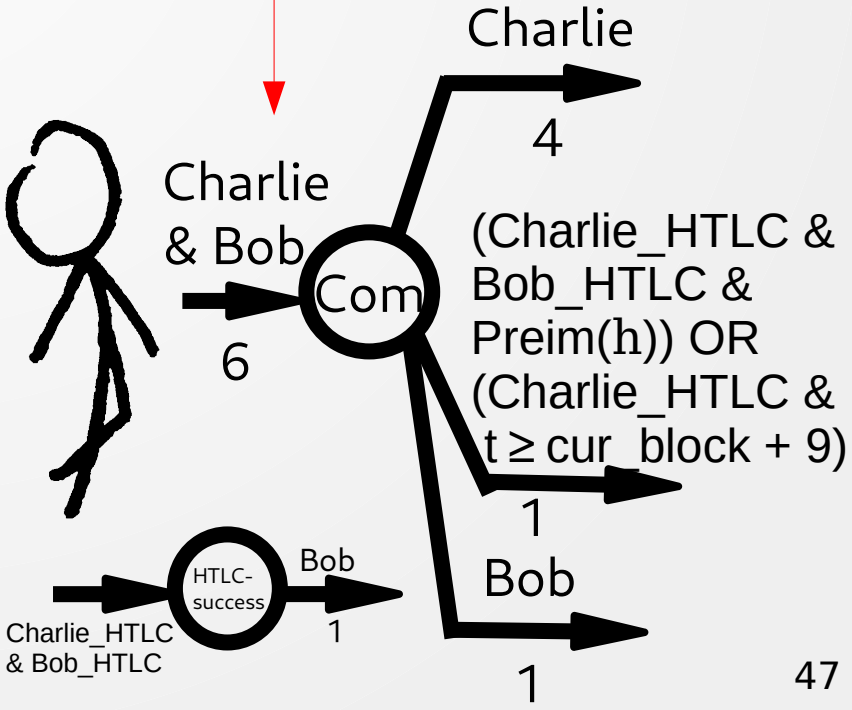
, sk-Charlie)

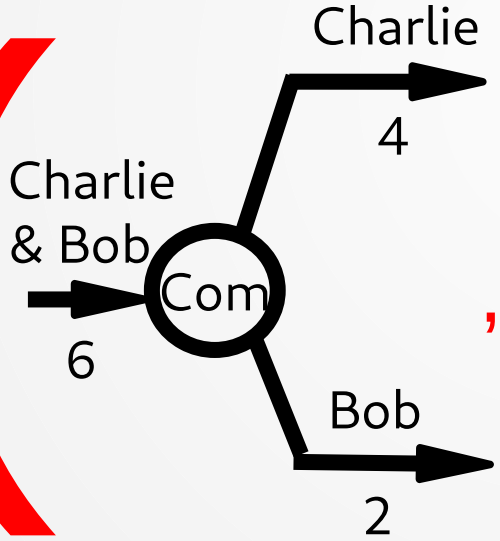




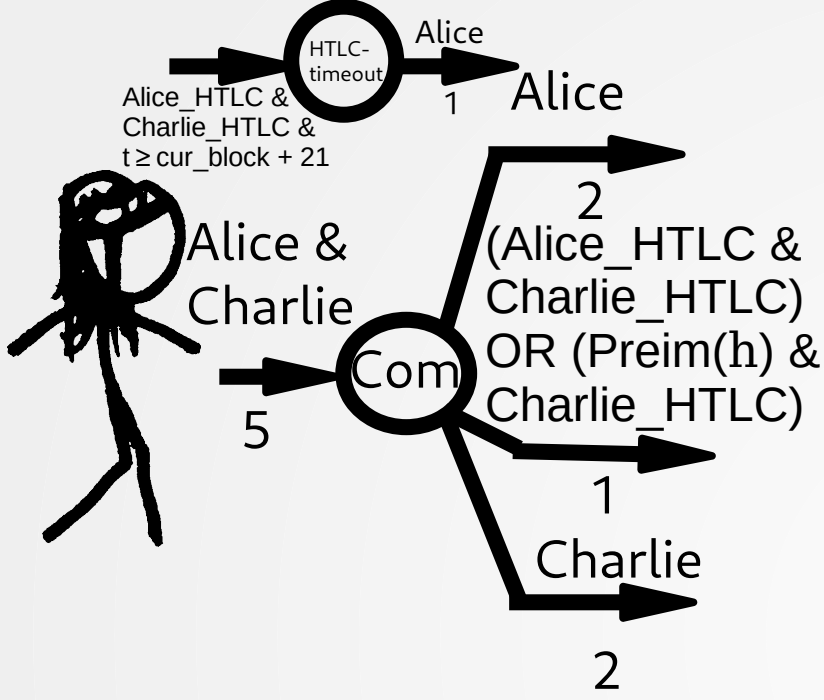
COMMITMENT_SIGNED, sig_Com

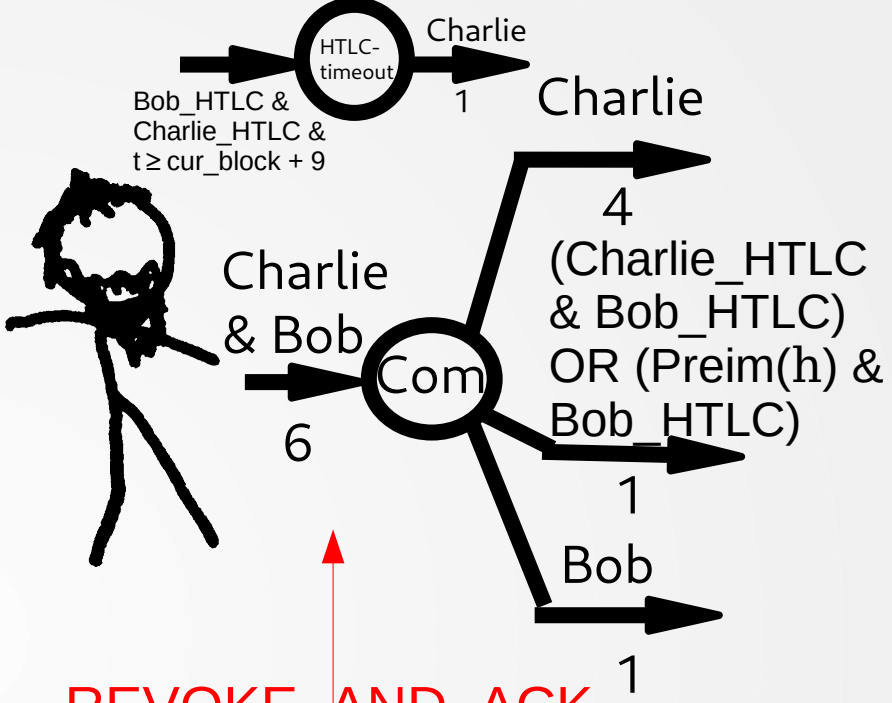




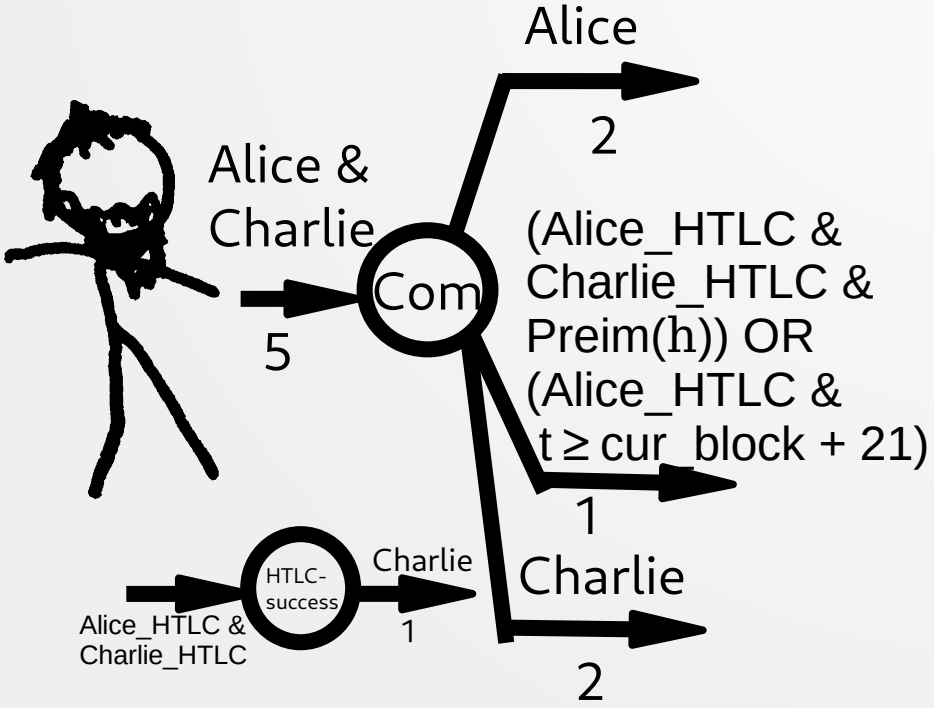
Verify 

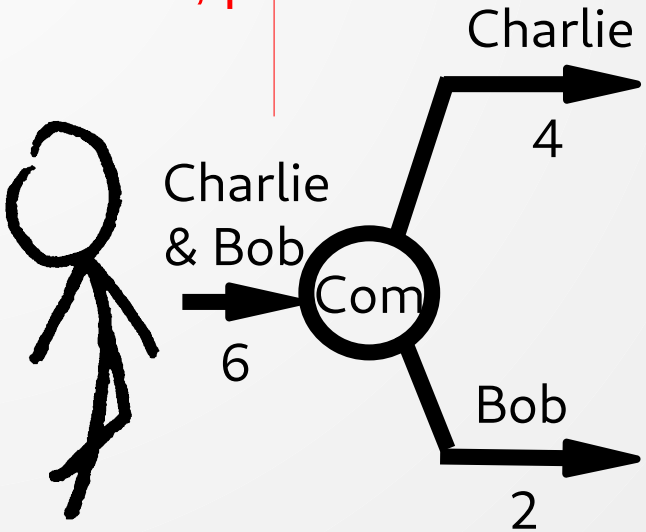
, pk-Charlie, sig_Com $\stackrel{?}{=} 1$

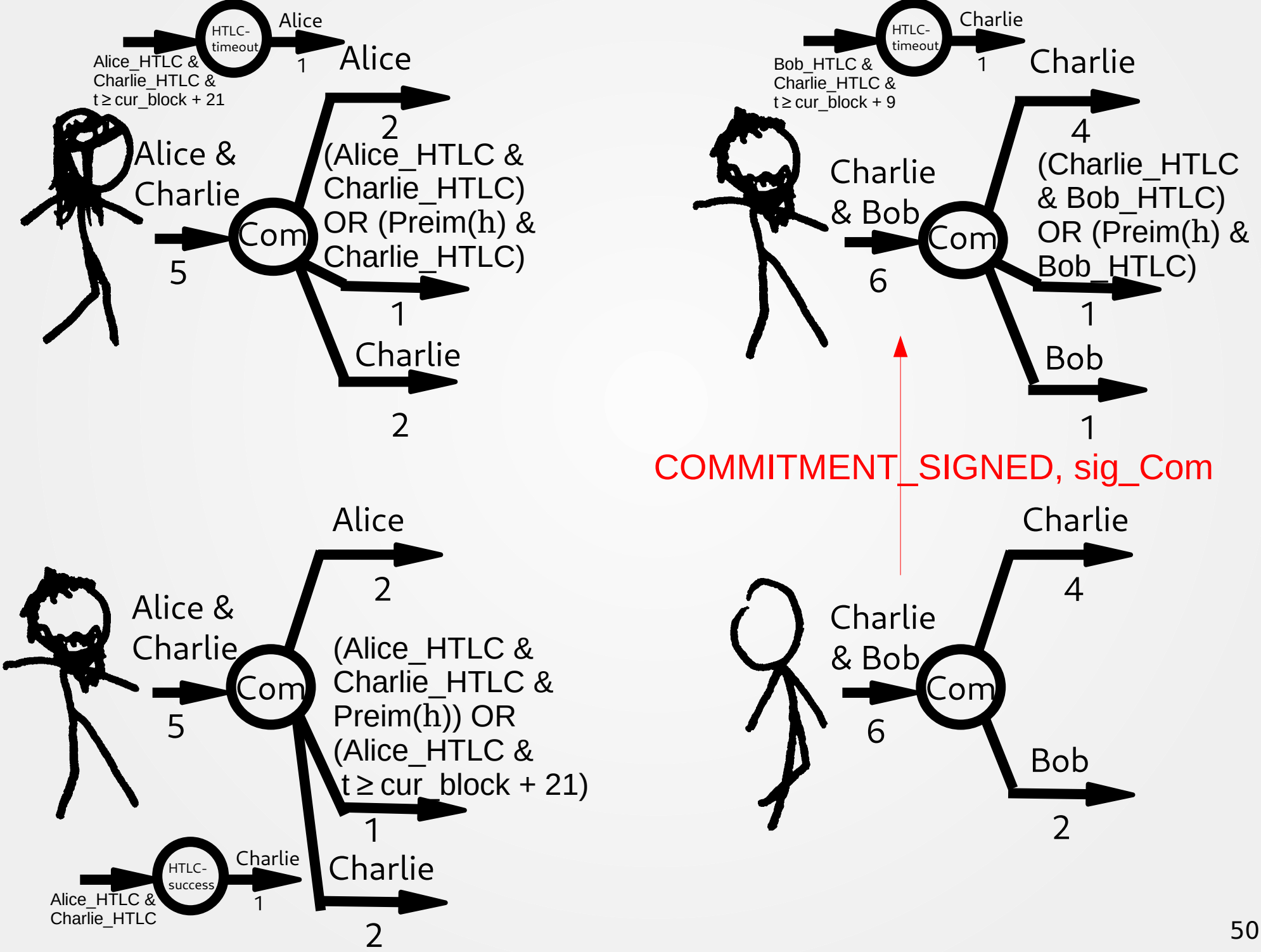


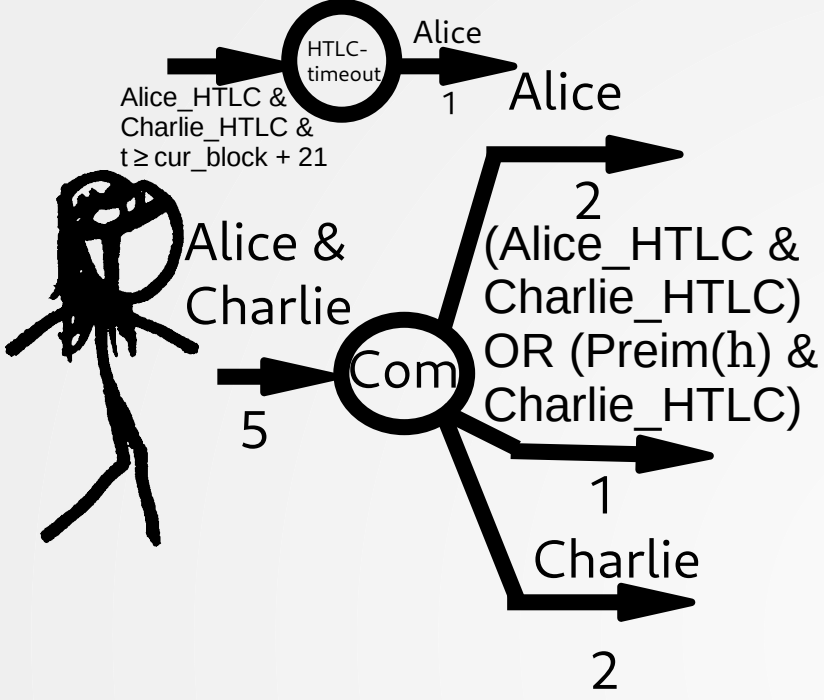


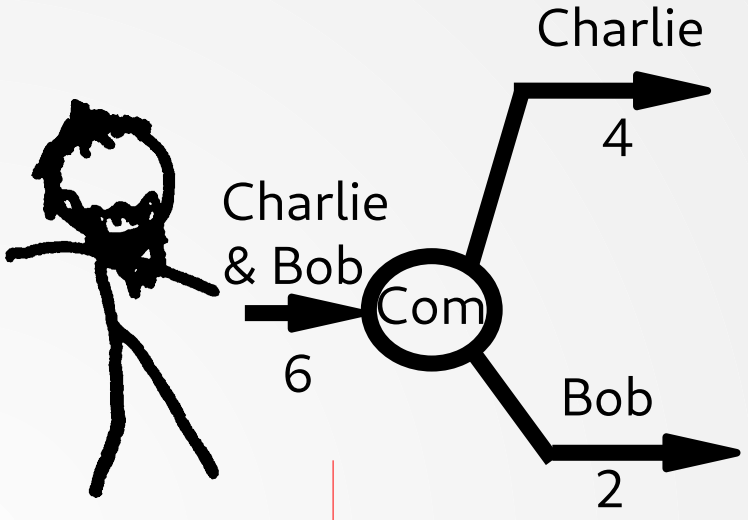
REVOKE_AND_ACK,
sk-Bob', pk-Bob''



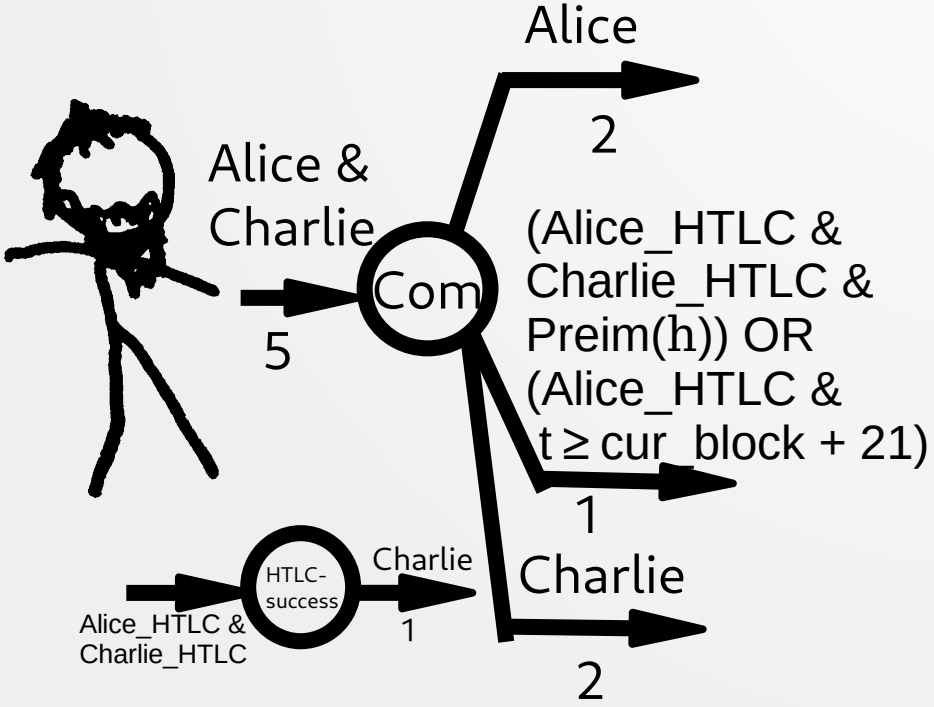


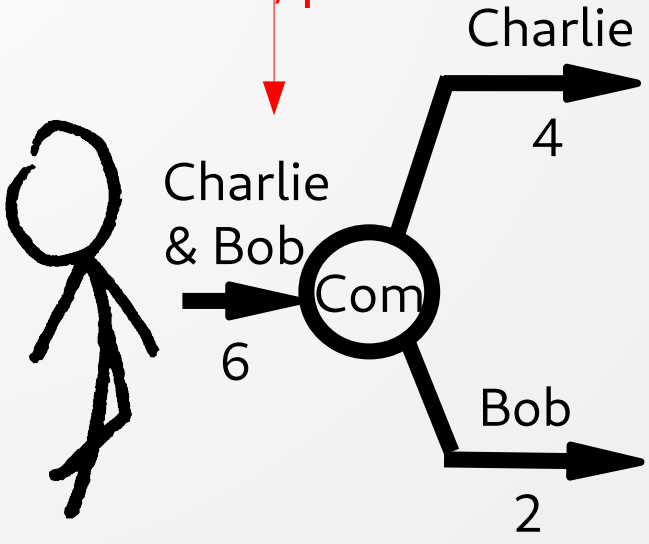

HTLC-timeout
Alice
Alice_HTLC &
Charlie_HTLC &
t ≥ cur_block + 21
1
Alice
2
Alice &
Charlie
Com
(Alice_HTLC &
Charlie_HTLC)
OR (Preim(h) &
Charlie_HTLC)
5
1
Charlie
2
HTLC-timeout
Charlie
Bob_HTLC &
Charlie_HTLC &
t ≥ cur_block + 9
1
Charlie
4
Charlie
& Bob
Com
(Charlie_HTLC
& Bob_HTLC)
OR (Preim(h) &
Bob_HTLC)
6
1
Bob
1
COMMITMENT_SIGNED, sig_Com
Alice
2
Alice &
Charlie
Com
(Alice_HTLC &
Charlie_HTLC &
Preim(h)) OR
(Alice_HTLC &
t ≥ cur_block + 21)
5
1
HTLC-success
Charlie
Alice_HTLC &
Charlie_HTLC
1
Charlie
2
Charlie
4
Charlie
& Bob
Com
6
Bob
2

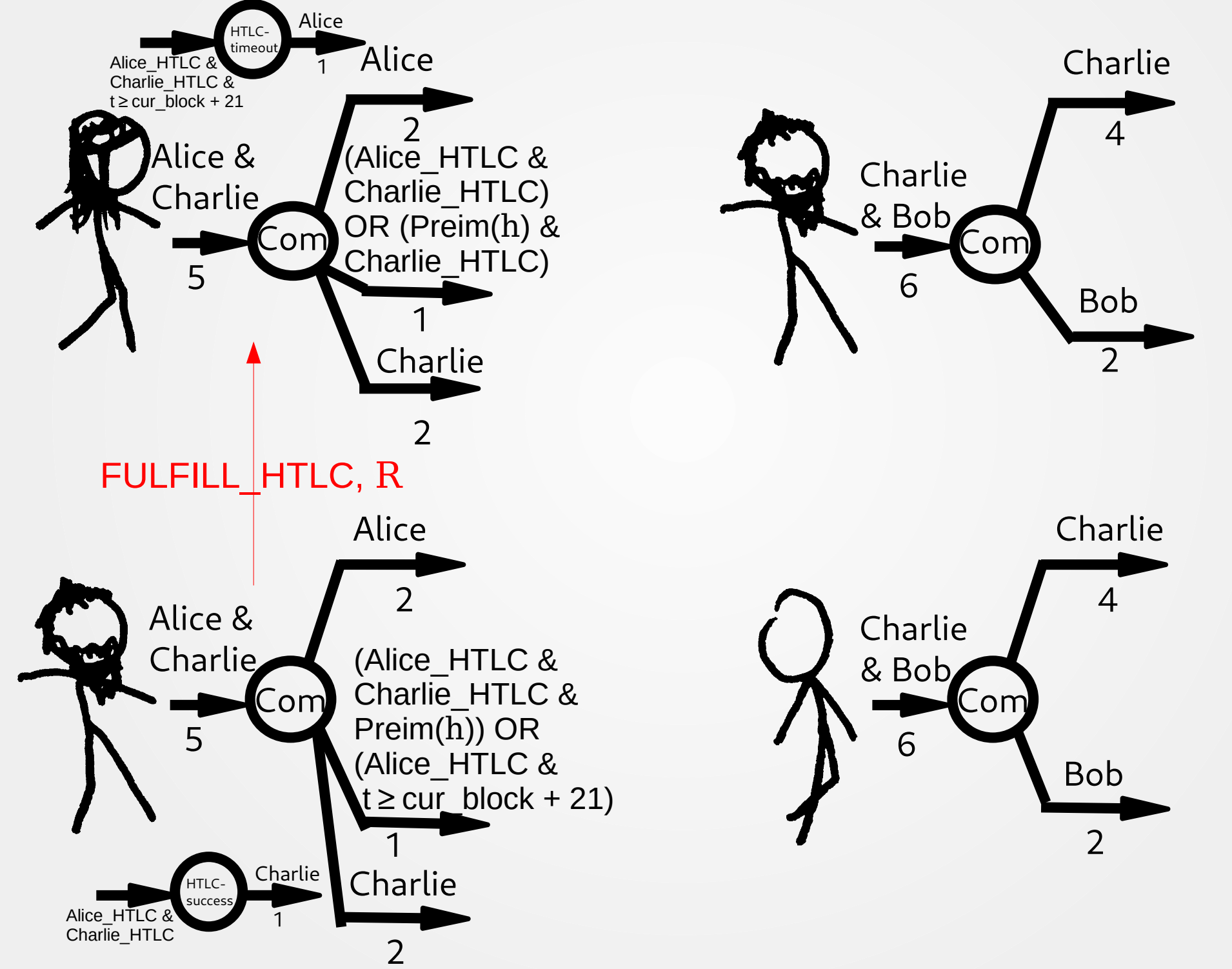


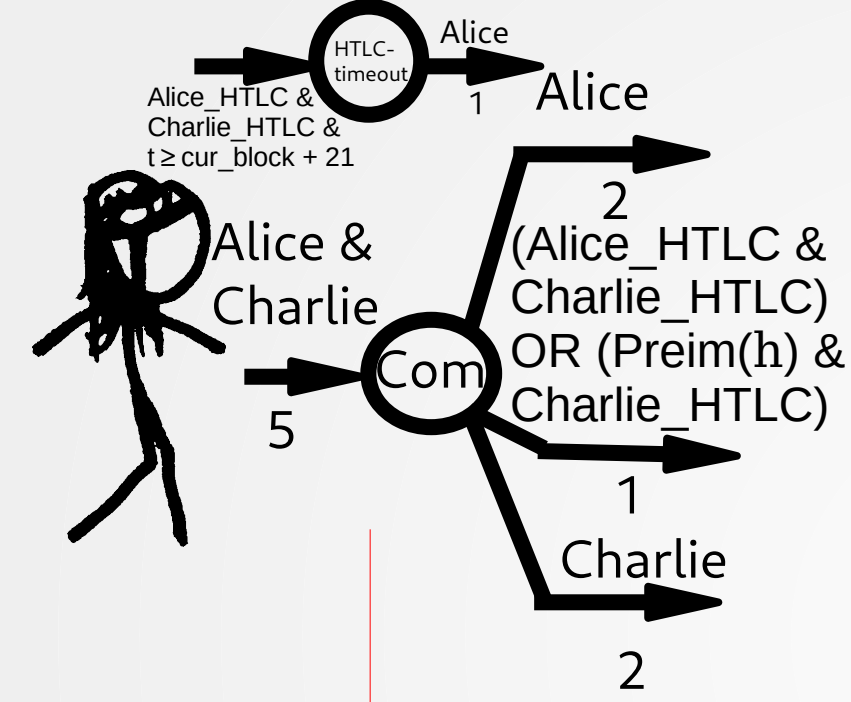


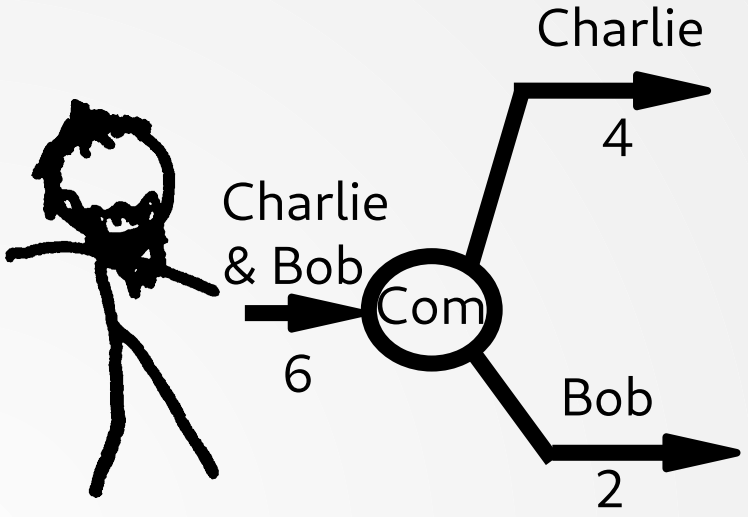
REVOKE_AND_ACK,
sk-Charlie', pk-Charlie''



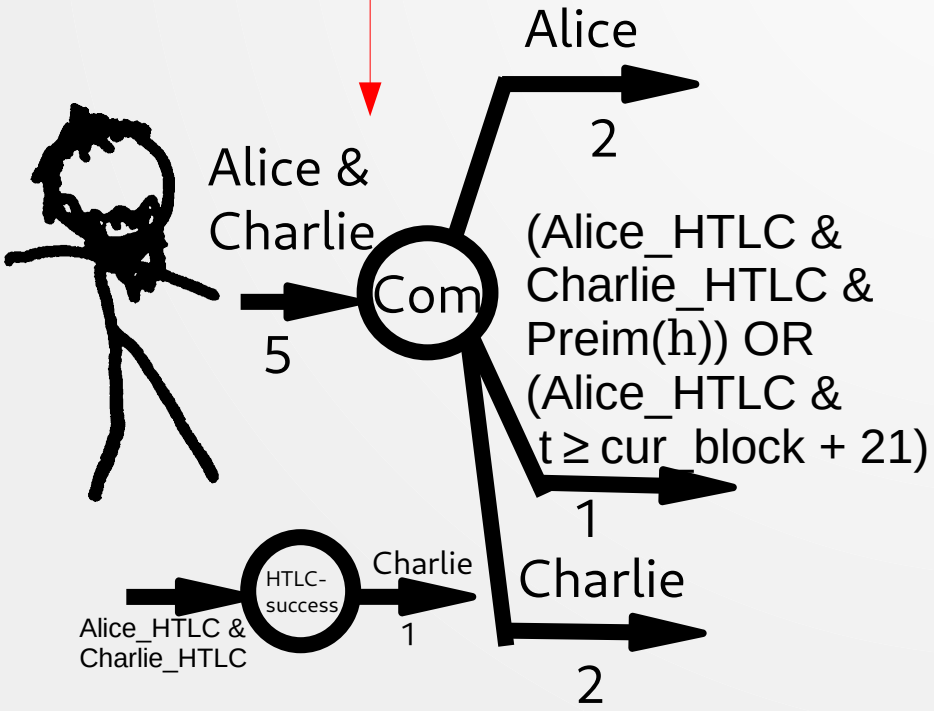


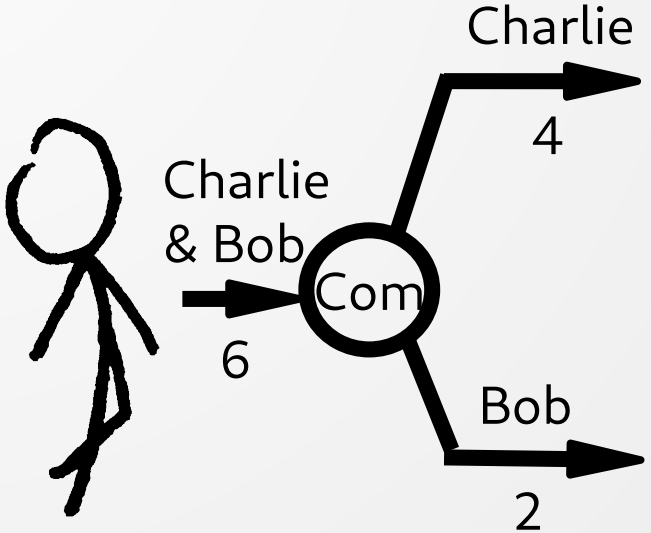

HTLC-timeout
Alice
Alice_HTLC &
Charlie_HTLC &
t ≥ cur_block + 21
1
Alice
2
Alice &
Charlie
Com
(Alice_HTLC &
Charlie_HTLC)
OR (Preim(h) &
Charlie_HTLC)
5
1
Charlie
2
Charlie
4
Charlie
& Bob
Com
6
Bob
2
FULFILL_HTLC, R
Alice
2
Alice &
Charlie
Com
(Alice_HTLC &
Charlie_HTLC &
Preim(h)) OR
(Alice_HTLC &
t ≥ cur_block + 21)
5
1
HTLC-success
Charlie
Alice_HTLC &
Charlie_HTLC
1
Charlie
2
Charlie
4
Charlie
& Bob
Com
6
Bob
2

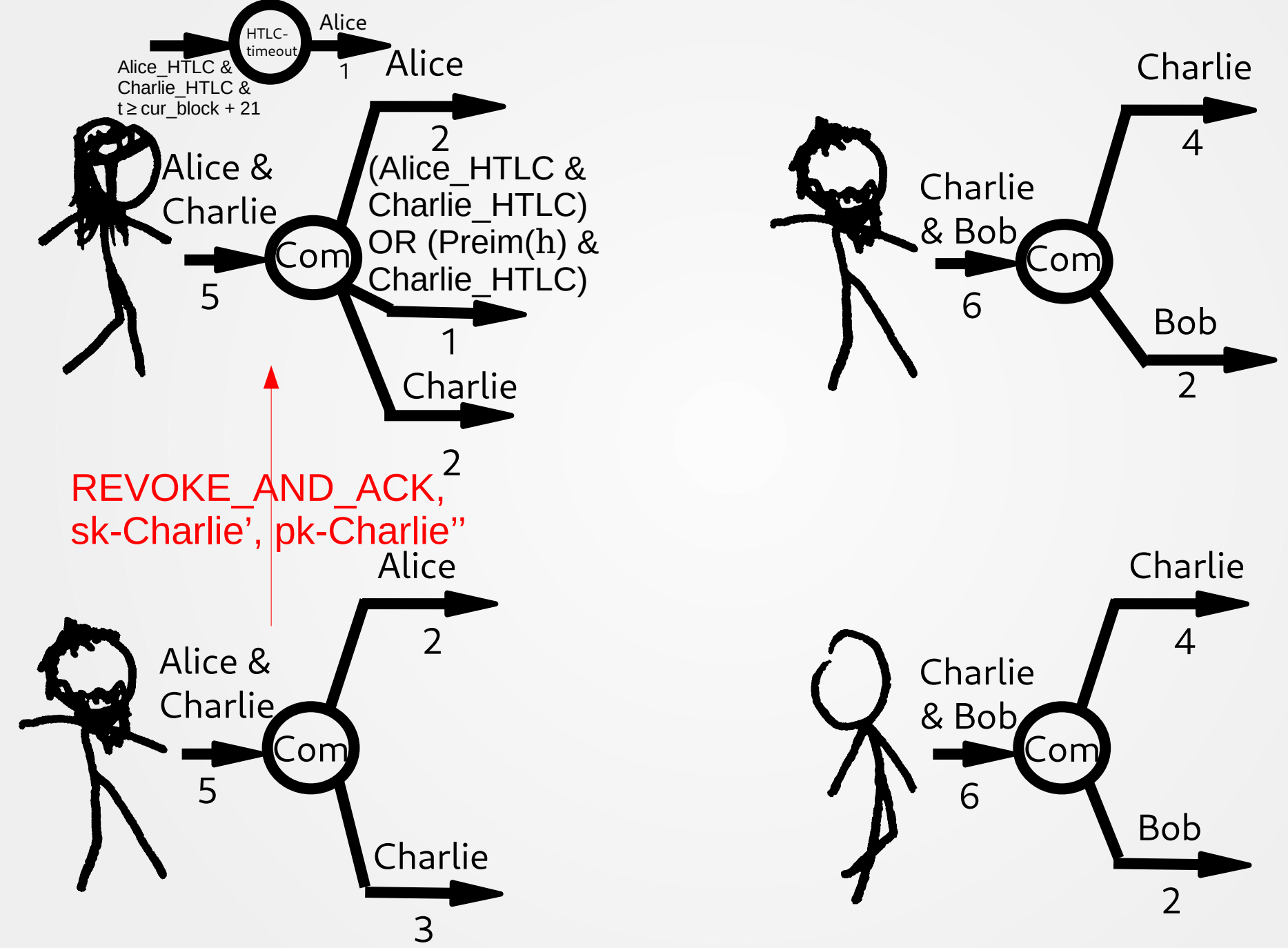


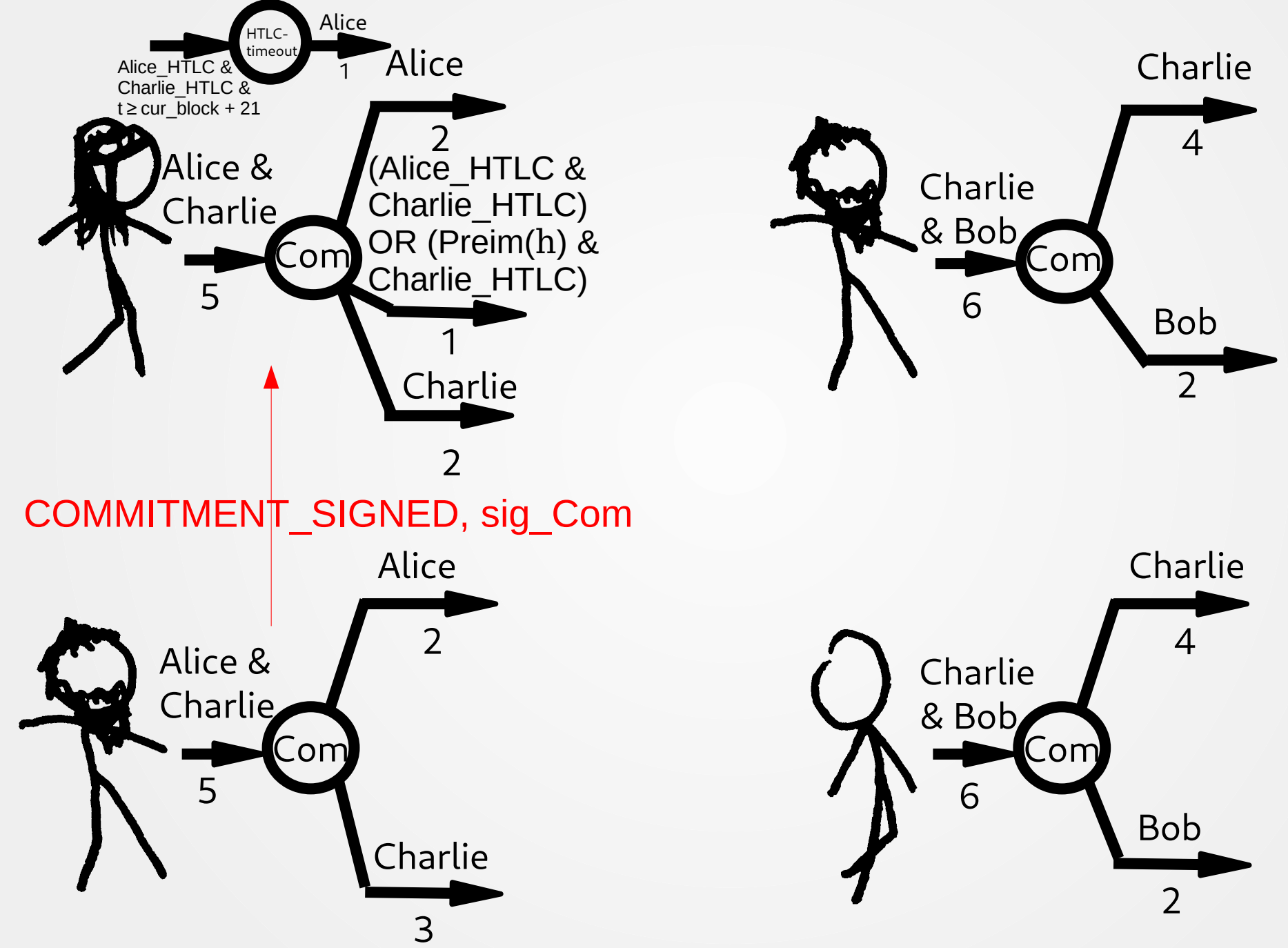


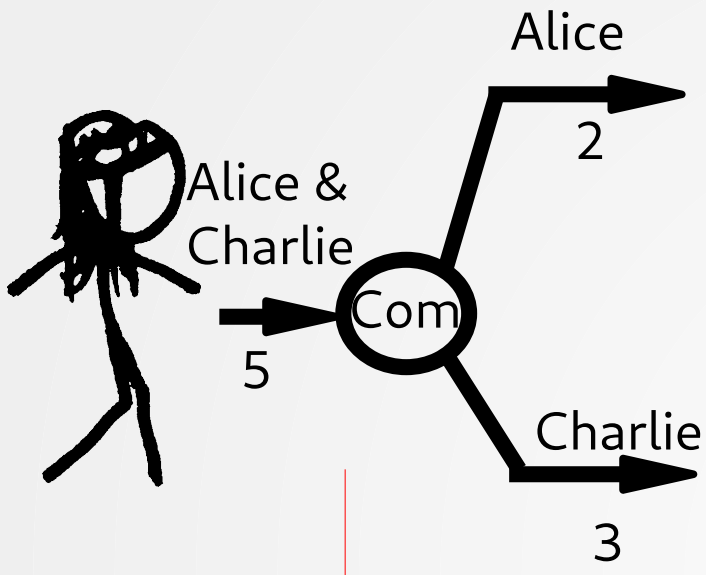
COMMITMENT_SIGNED, sig_Com

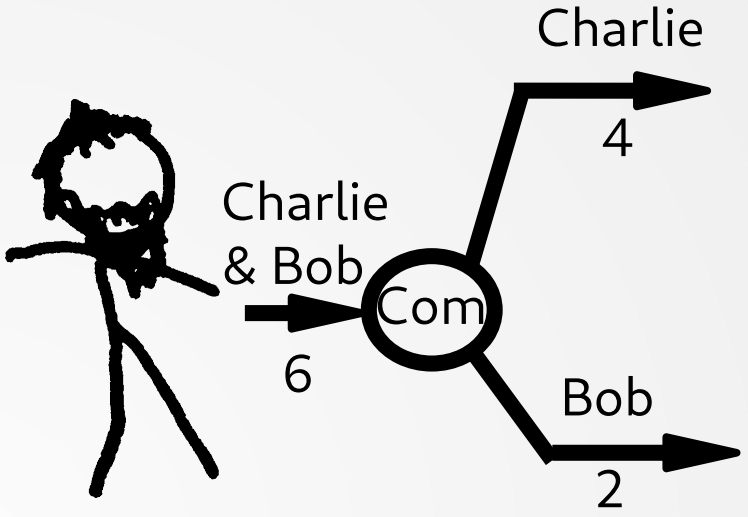




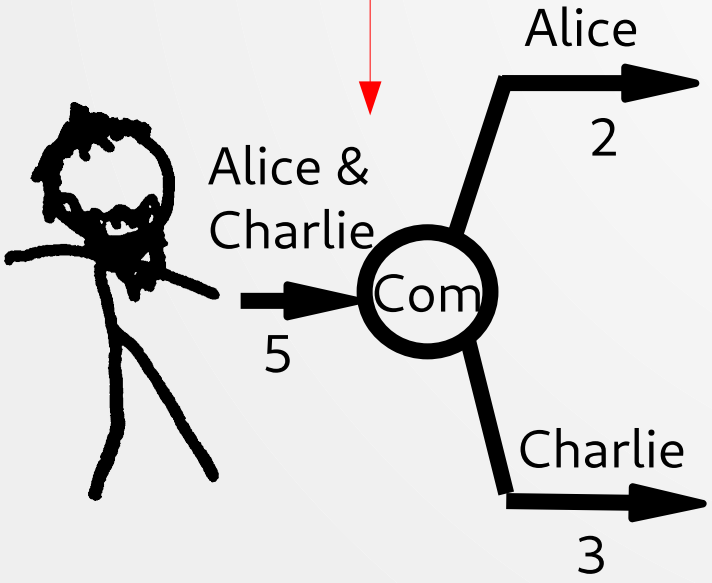

HTLC-timeout
Alice
1
Alice
Alice_HTLC &
Charlie_HTLC &
t ≥ cur_block + 21
2
Alice &
Charlie
(Alice_HTLC &
Charlie_HTLC)
OR (Preim(h) &
Charlie_HTLC)
Com
5
1
Charlie
2
REVOKE_AND_ACK,
sk-Charlie', pk-Charlie''
Alice
2
Alice &
Charlie
Com
5
Charlie
3
Charlie
4
Charlie
& Bob
Com
6
Bob
2
Charlie
4
Charlie
& Bob
Com
6
Bob
2

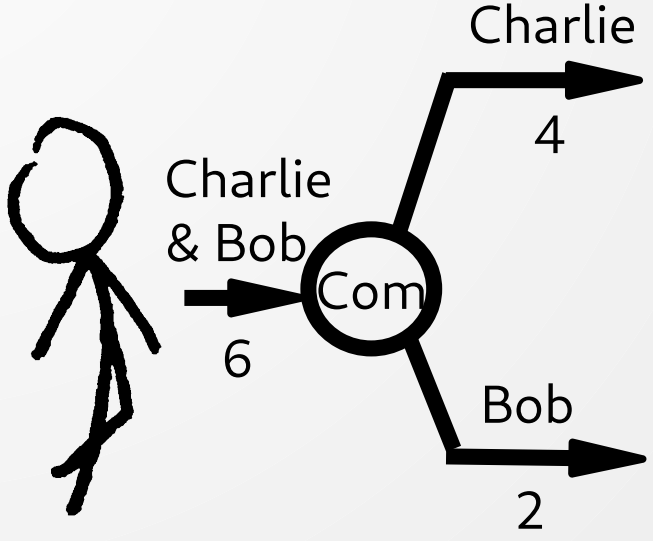

HTLC-timeout
Alice
1
Alice_HTLC &
Charlie_HTLC &
t ≥ cur_block + 21
Alice
2
Alice &
Charlie
Com
(Alice_HTLC &
Charlie_HTLC)
OR (Preim(h) &
Charlie_HTLC)
5
1
Charlie
2
Charlie
4
Charlie
& Bob
Com
6
Bob
2
COMMITMENT_SIGNED, sig_Com
Alice
2
Alice &
Charlie
Com
5
Charlie
3
Charlie
4
Charlie
& Bob
Com
6
Bob
2

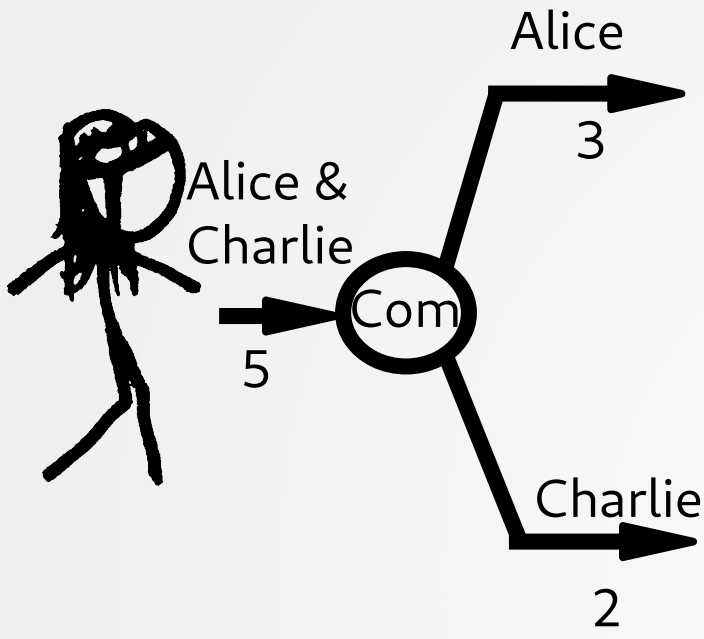


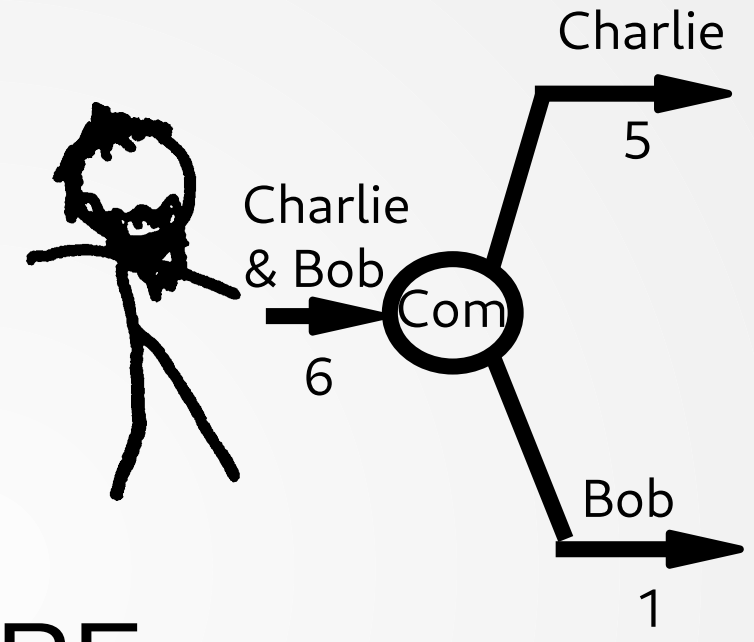


REVOKE_AND_ACK,
sk-Alice', pk-Alice''

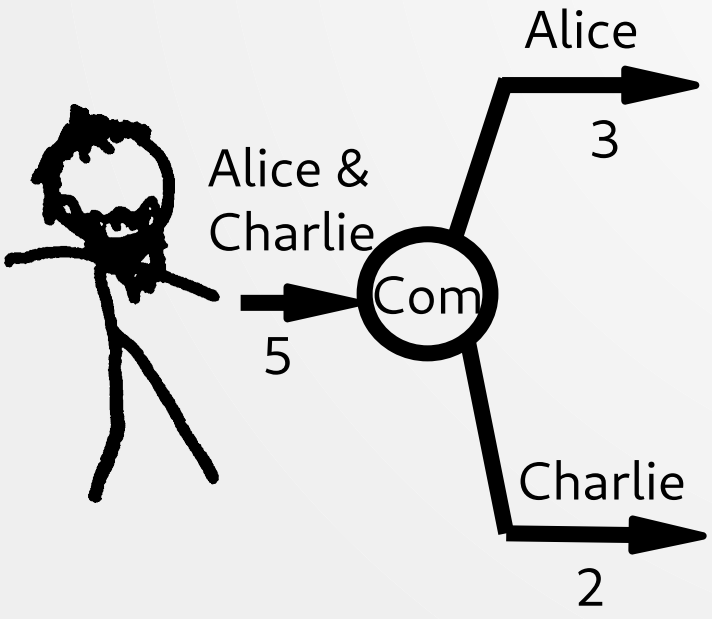


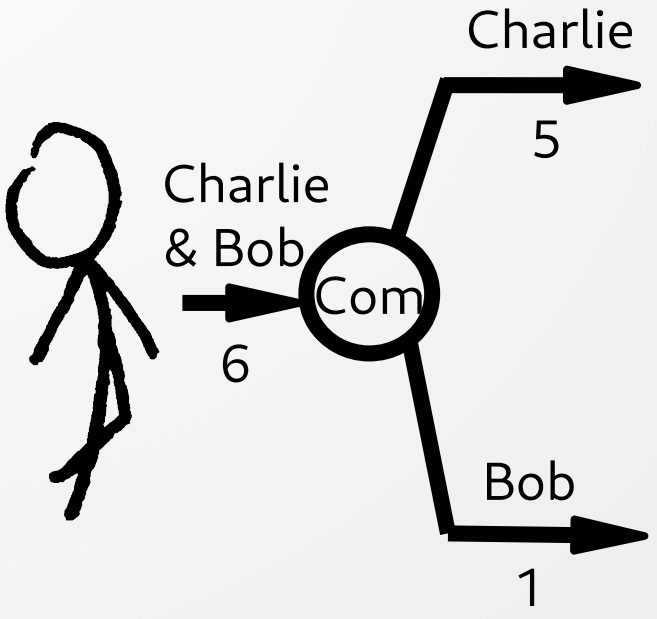


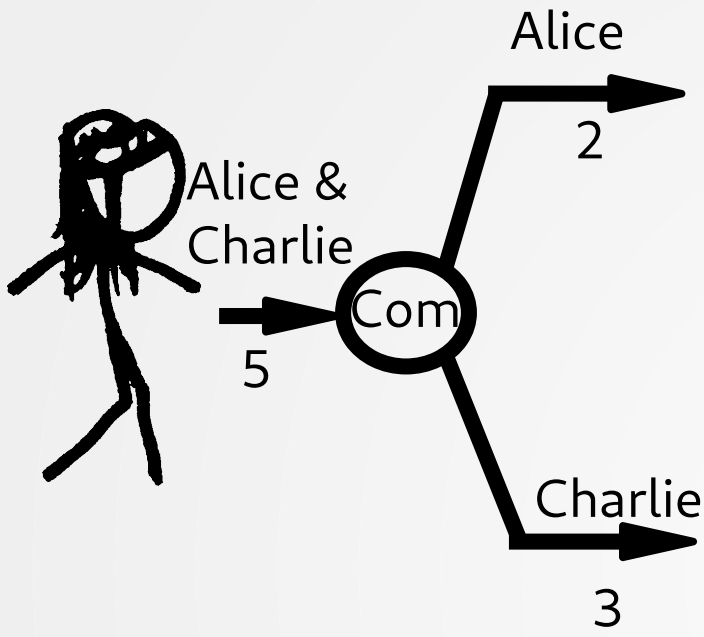

Alice
3
Alice &
Charlie
Com
5
Charlie
2

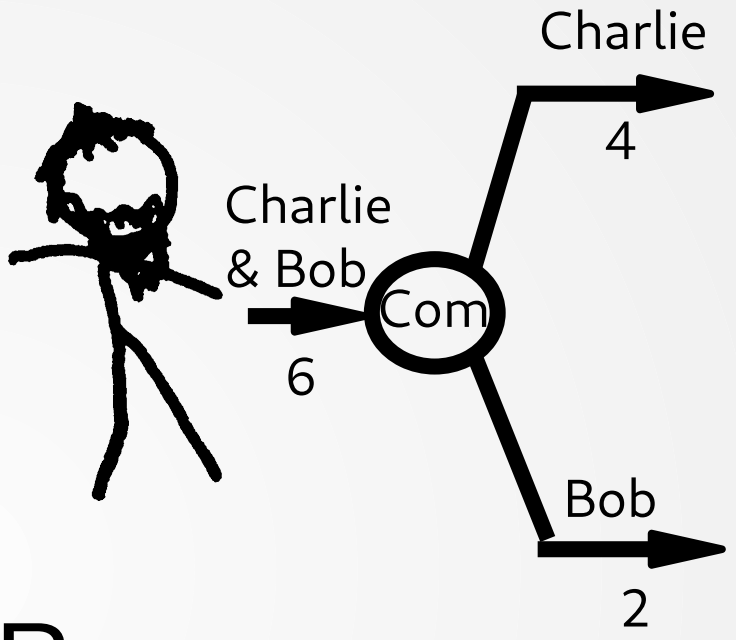

Charlie
5
Charlie
& Bob
Com
6
Bob
1

BEFORE

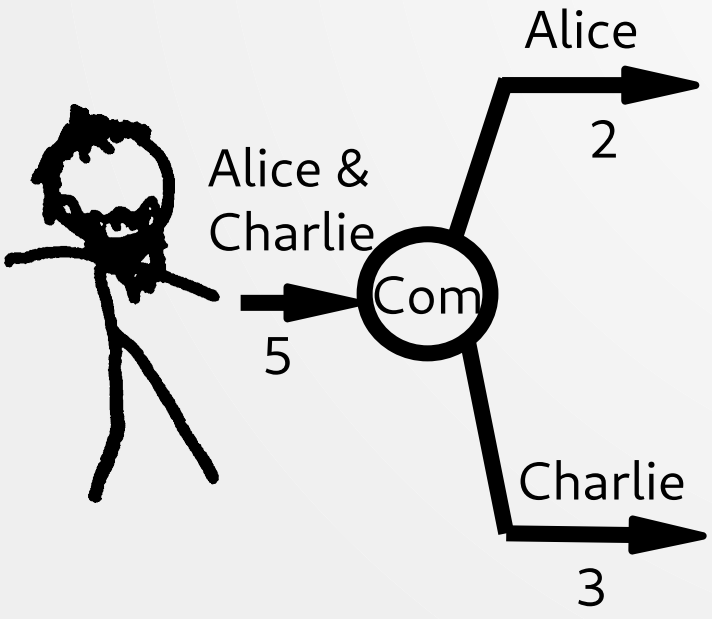

Alice
3
Alice &
Charlie
Com
5
Charlie
2

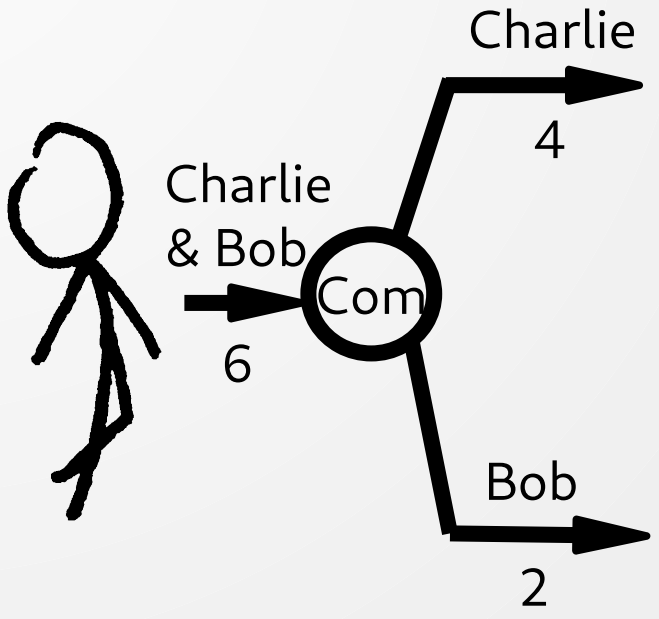

Charlie
5
Charlie
& Bob
Com
6
Bob
1


Alice
2
Alice &
Charlie
Com
5
Charlie
3


Charlie
4
Charlie
& Bob
Com
6
Bob
2

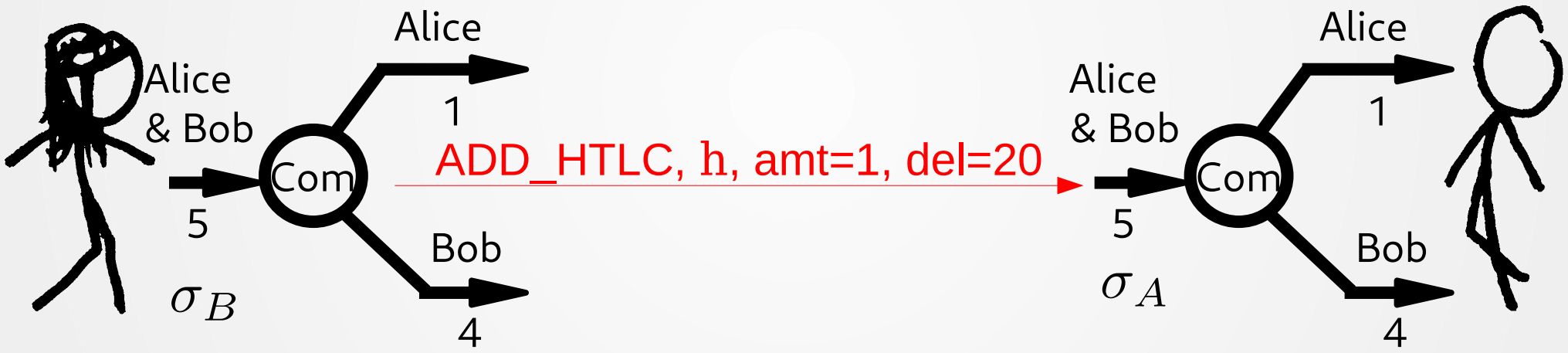
AFTER

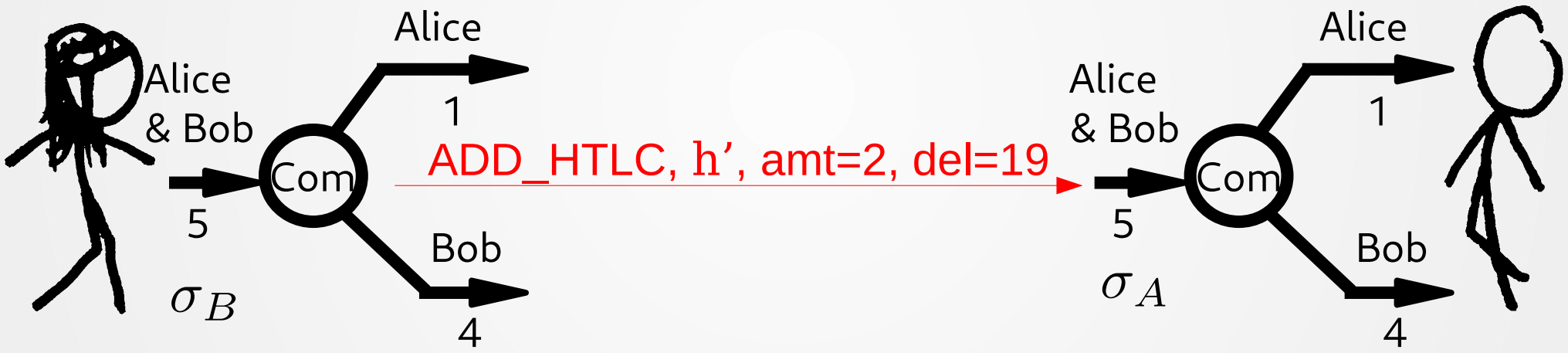

Alice
2
Alice &
Charlie
Com
5
Charlie
3

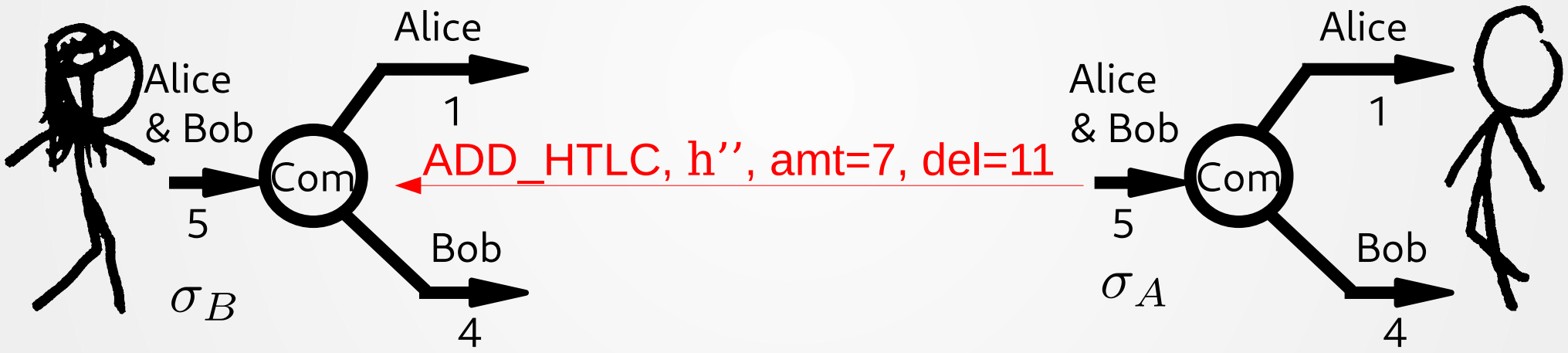

Charlie
4
Charlie
& Bob
Com
6
Bob
2

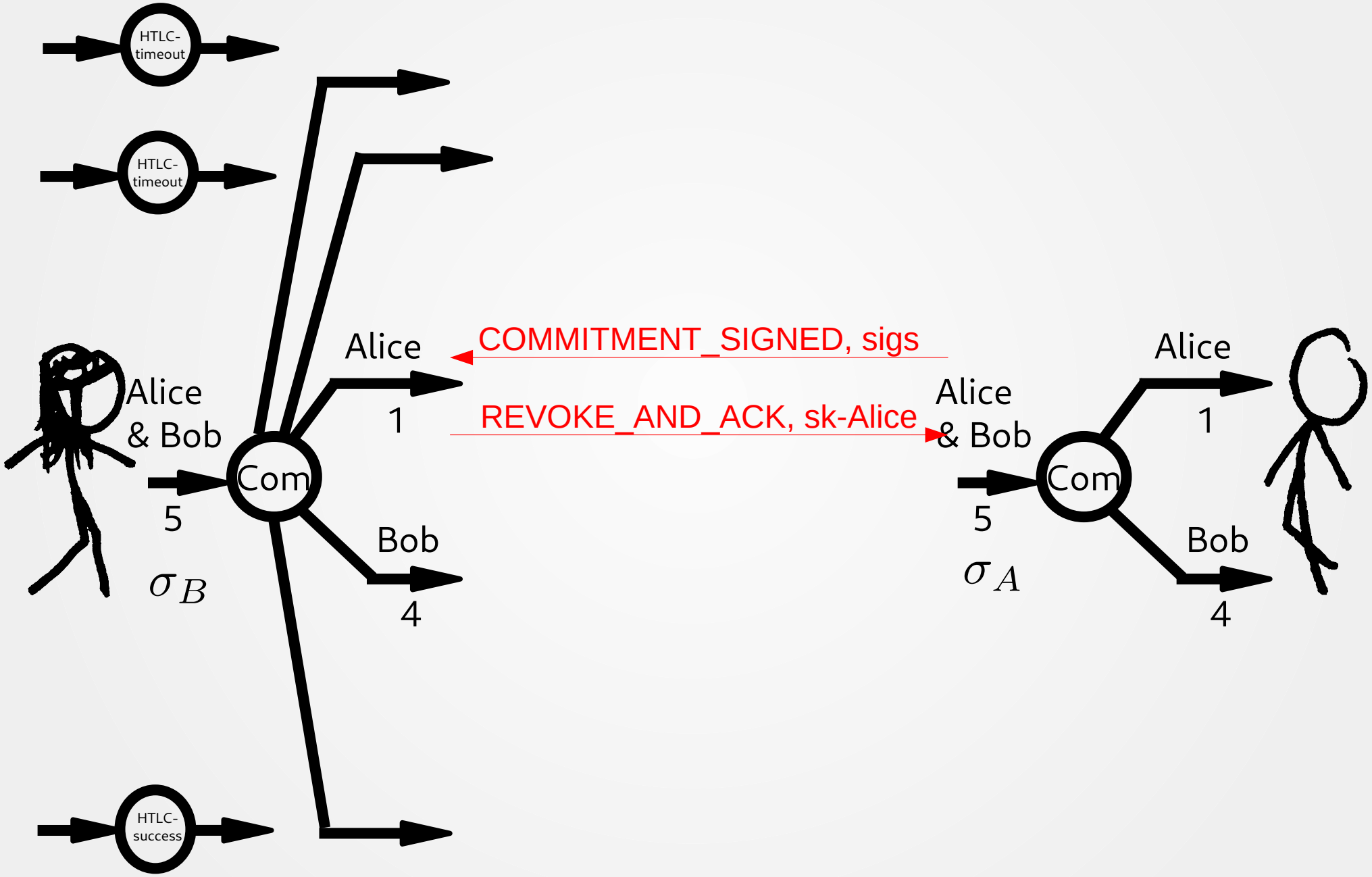
Part 3
Optimisations, Privacy, Fees

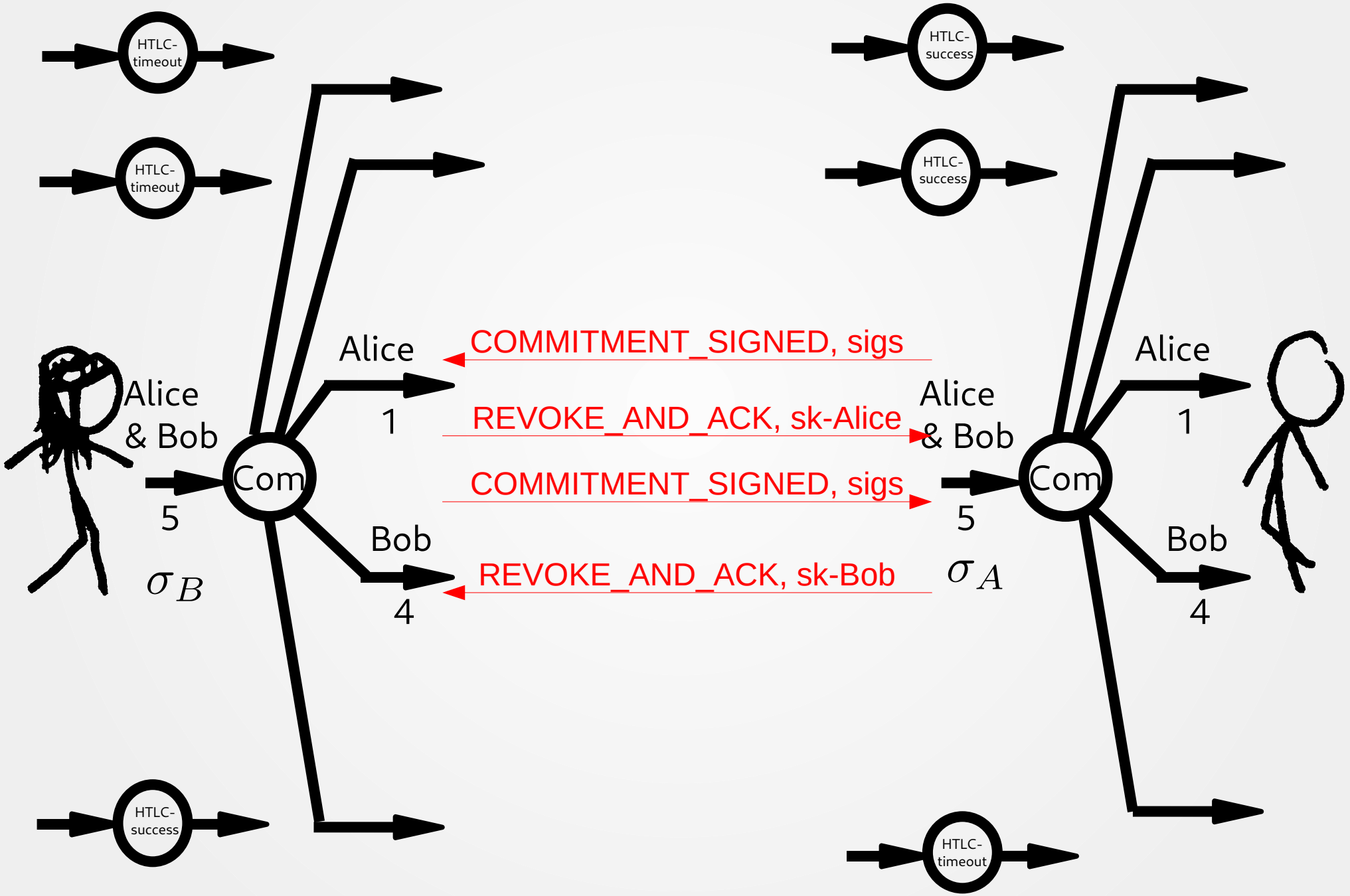
HTLCs can be batched


Alice
& Bob
5
σ_B
Com
Alice
1
Bob
4
ADD_HTLC, h, amt=1, del=20
Alice
& Bob
5
σ_A
Com
Alice
1
Bob
4


Alice
& Bob
5
σ_B
Com
Alice
1
Bob
4
ADD_HTLC, h', amt=2, del=19
Alice
& Bob
5
σ_A
Com
Alice
1
Bob
4


Alice
& Bob
5
σ_B
Com
Alice
1
Bob
4
ADD_HTLC, h'', amt=7, del=11
Alice
& Bob
5
σ_A
Com
Alice
1
Bob
4


HTLC-timeout
HTLC-timeout
Alice
COMMITMENT_SIGNED, sigs
Alice & Bob
1
REVOKE_AND_ACK, sk-Alice
Com
5
σ_B
Bob
4
HTLC-success
Alice
Alice & Bob
1
Com
5
σ_A
Bob
4


HTLC-timeout
HTLC-timeout
HTLC-success
HTLC-success
Alice
& Bob
Com
Alice
1
Bob
4
5
σ_B
COMMITMENT_SIGNED, sigs
REVOKE_AND_ACK, sk-Alice
COMMITMENT_SIGNED, sigs
REVOKE_AND_ACK, sk-Bob
Alice
& Bob
Com
Alice
1
Bob
4
5
σ_A
HTLC-success
HTLC-timeout

Basepoints & per-commitment points

4 keys per update

- Revocation
- Payment
- Delayed payment
- HTLC

OPEN_CHANNEL, basepoint_rev,
basepoint_payment,
basepoint_delayed_payment,
basepoint_htlc,
first_per_commitment_point, ...

...

REVOKE_AND_ACK,
next_per_commitment_point, ...

```
Alice_pubkey_# = basepoint_# +
SHA256(per_commitment_point || basepoint_#) * G
```

(except for `_rev`)

Revocation *keys* instead of multisigs

```
revocationpubkey =
revocation_basepoint *
SHA256(revocation_basepoint || per_commitment_point)
+ per_commitment_point *
SHA256(per_commitment_point || revocation_basepoint)
```

”*This construction ensures that neither the node providing the basepoint nor the node providing the per_commitment_point can know the private key without the other node's secret.*”

```
revocationpubkey =
revocation_basepoint *
SHA256(revocation_basepoint || per_commitment_point)
+ per_commitment_point *
SHA256(per_commitment_point || revocation_basepoint)
```

”This construction ensures that neither the node providing the basepoint nor the node providing the per_commitment_point can know the private key without the other node's secret.”

```
revocationprivkey =
revocation_basepoint_secret *
SHA256(revocation_basepoint || per_commitment_point)
+ per_commitment_secret *
SHA256(per_commitment_point || revocation_basepoint)
```

Cooperative closing:
No timelocks!

- Complete HTLCs (don’t start new)
- Create and sign TX that spends F and gives parties their due amounts
- Send signature
- Counterparty should also sign and broadcast
- If not, close unilaterally

Watchtowers:
Watching the chain for you
with loads of privacy

Sphinx[1] Onions for multi-hop privacy

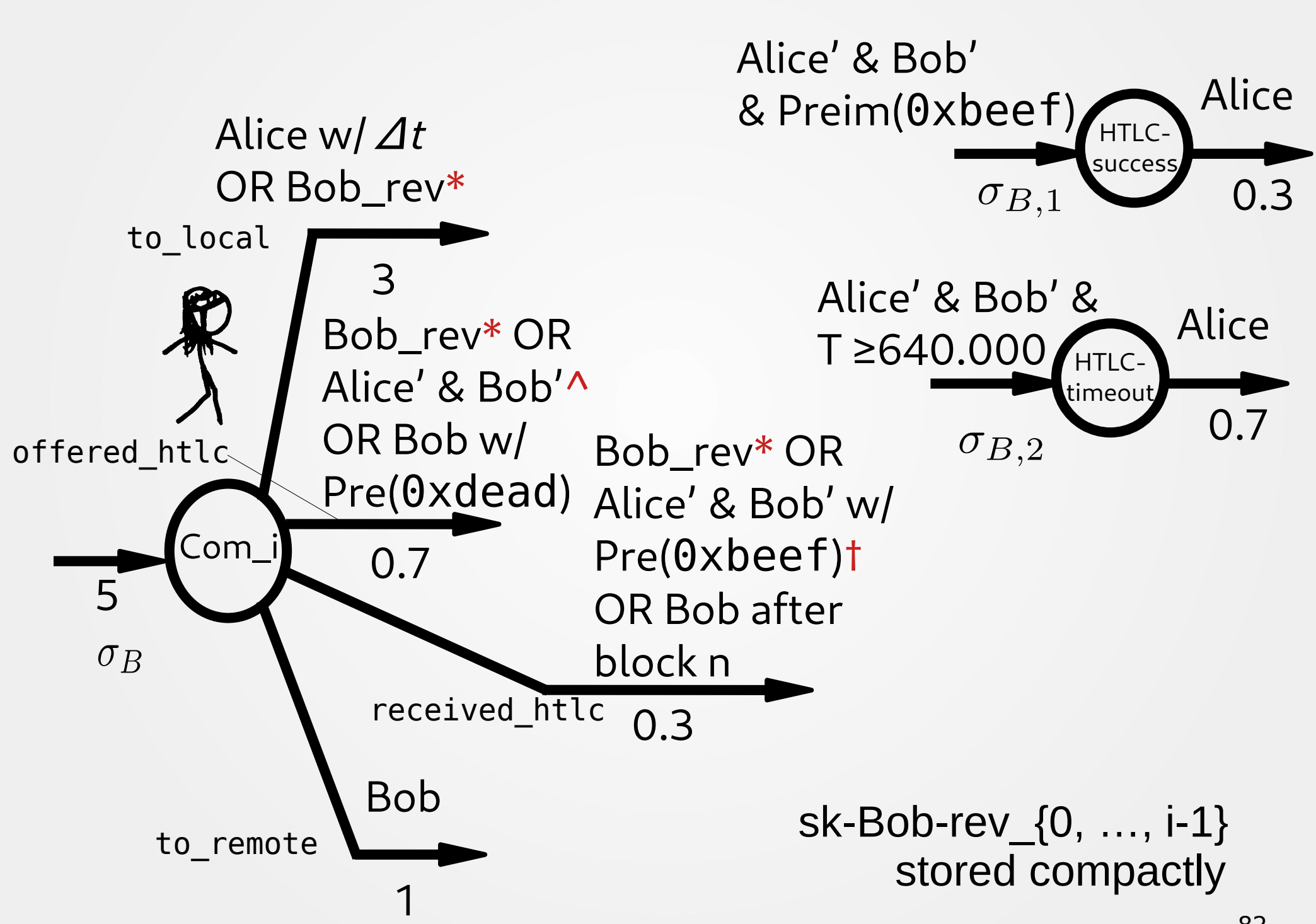
[1] Danezis G., Goldberg I., *Sphinx: A Compact and Provably Secure Mix Format*, 30th IEEE Symposium on Security and Privacy (S&P 2009), 17-20 May 2009, Oakland, California, USA https://www.cypherpunks.ca/~iang/pubs/Sphinx_Oakland09.pdf

Dust outputs:
Trim them and stay `standard()`

Fees:
On- and off-chain

- Off-chain
  - Nodes declare their fee
  - Payers pay fees for all hops
- On-chain
  - Fees hardcoded in Commitment TX
  - Nodes negotiate when opening
  - May renegotiate in an update
  - May renegotiate when closing cooperatively
  - Should overpay fees to ensure they can close (attacks galore!)

Routing:
Achilles’ heel for decentralization



*Revocation ^To HTLC-timeout TX †To HTLC-success TX

https://github.com/lightningnetwork/lightning-rfc/

Thank you!