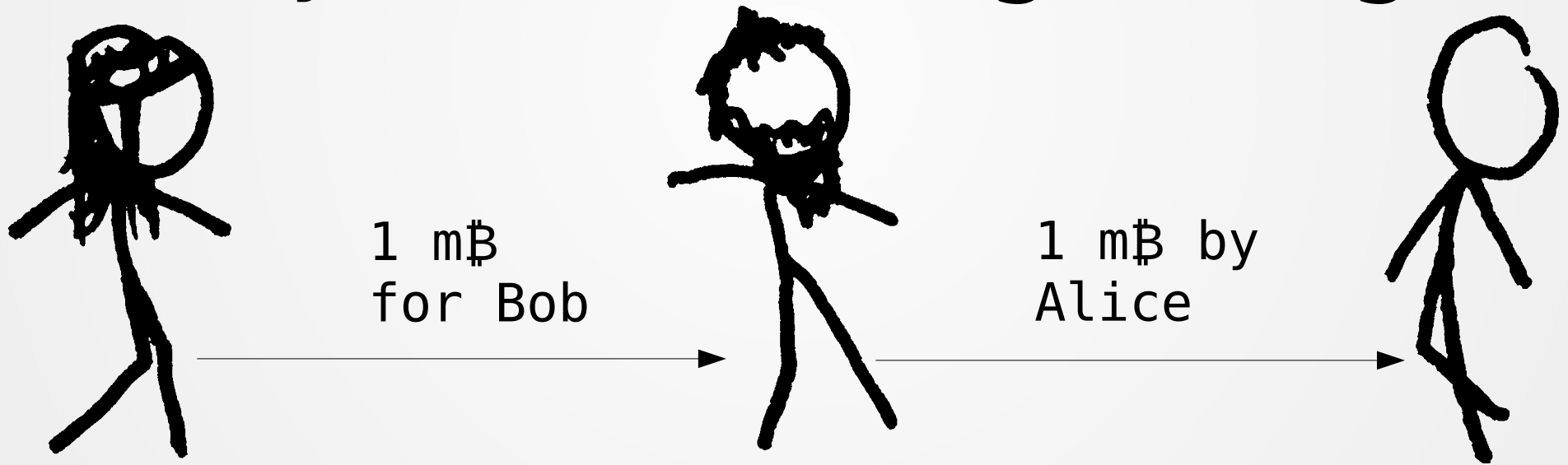


# From Channels to Network: Off-chain Multi-hop Payments in Lightning



Orfeas Stefanos  
Thyfronitis Litos

University of Edinburgh

**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

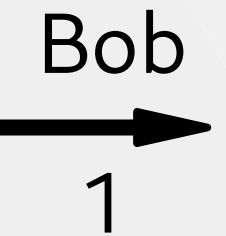
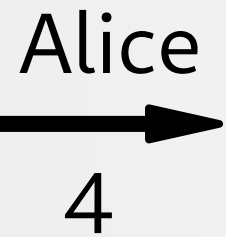
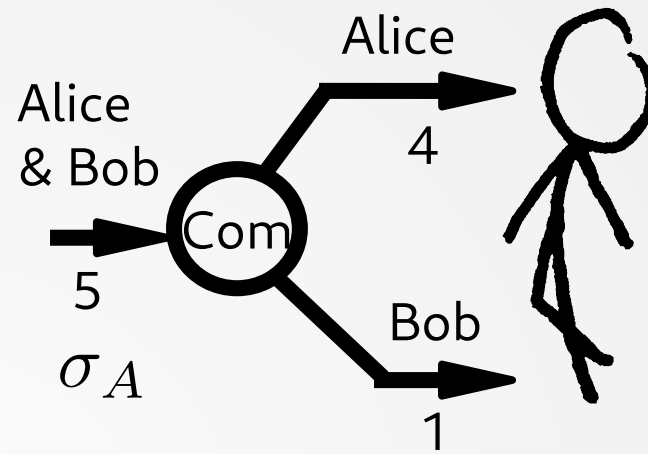
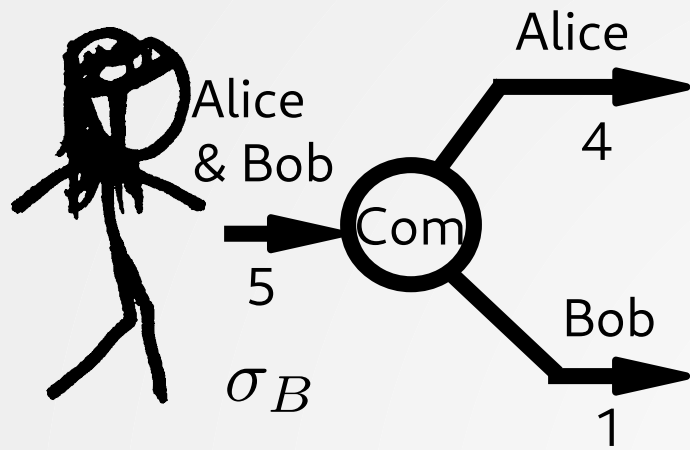
# Part 0

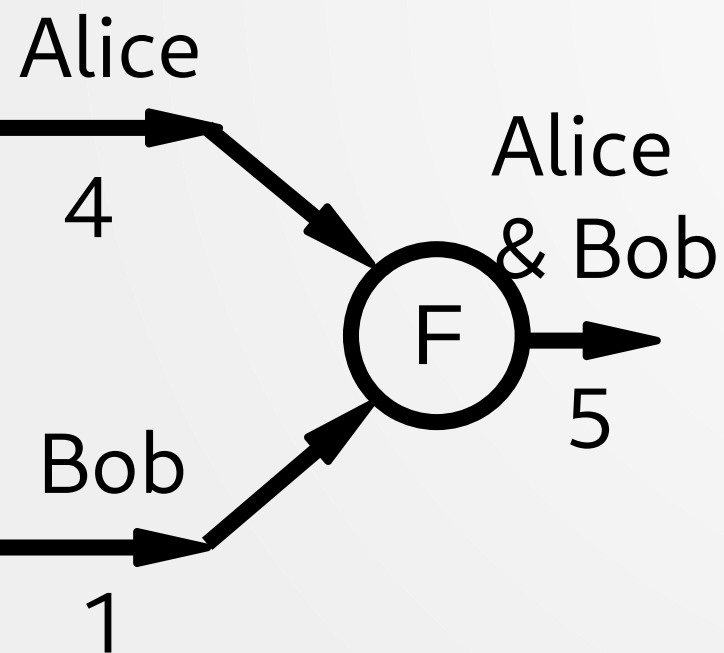
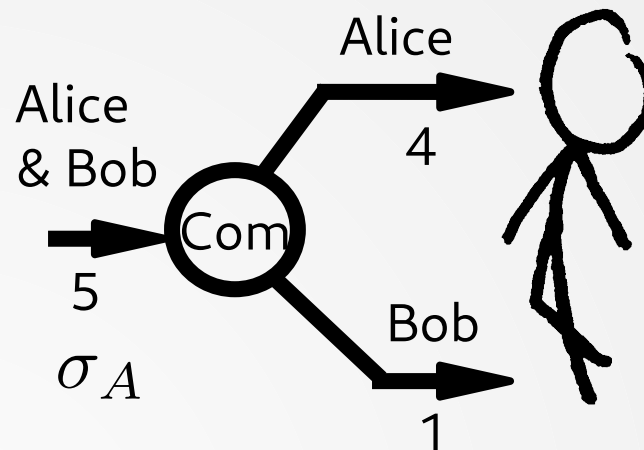
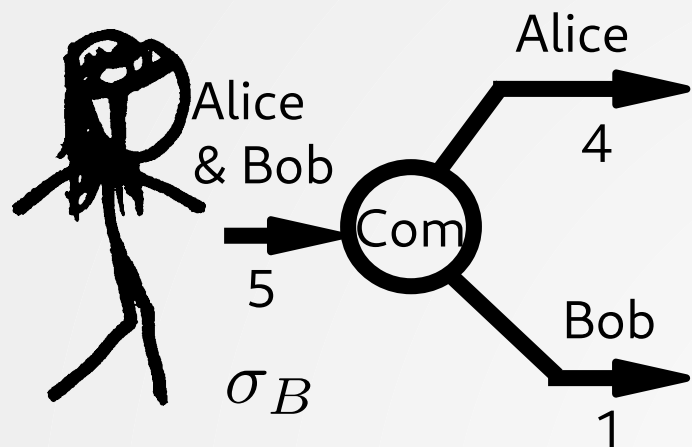
## Recap: 2-party channels

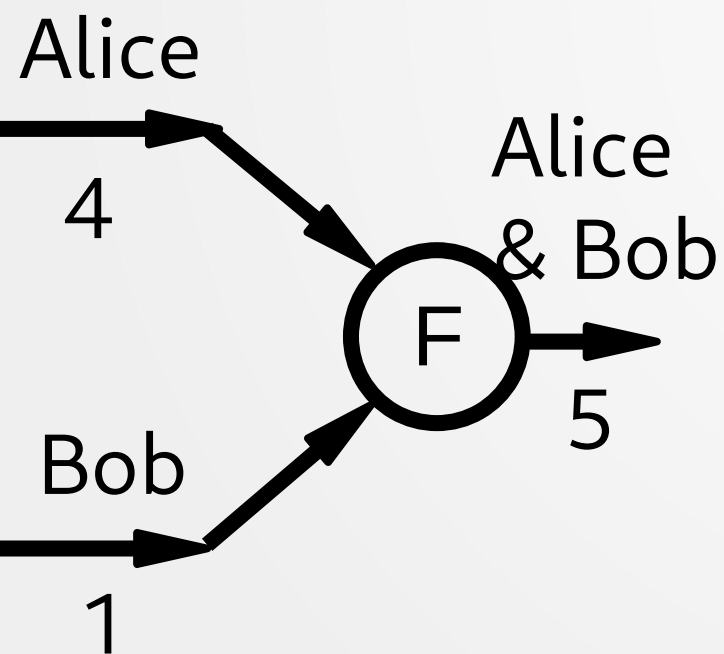
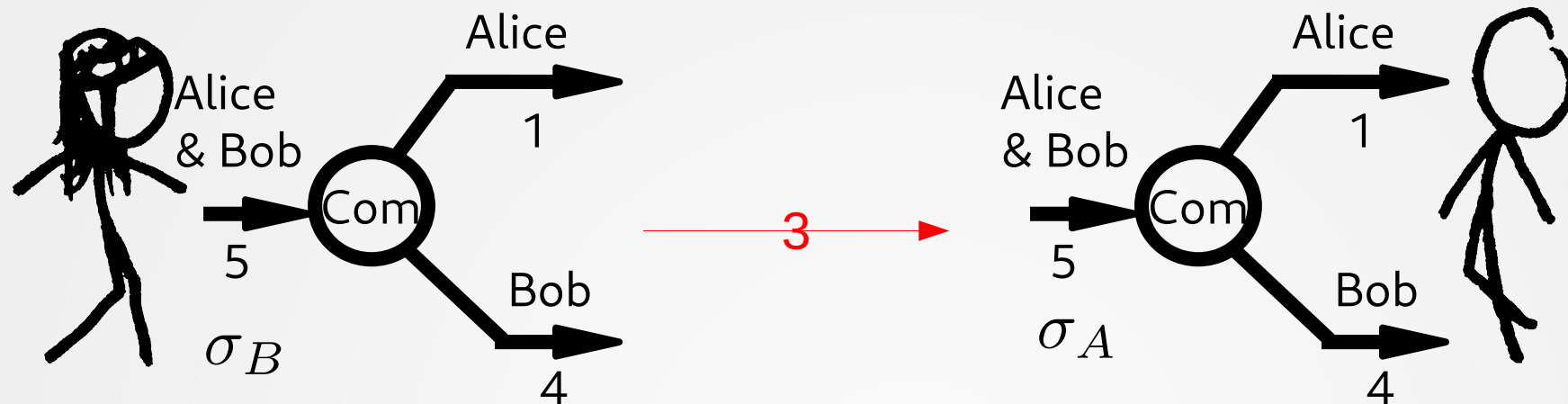


Alice  
→  
4

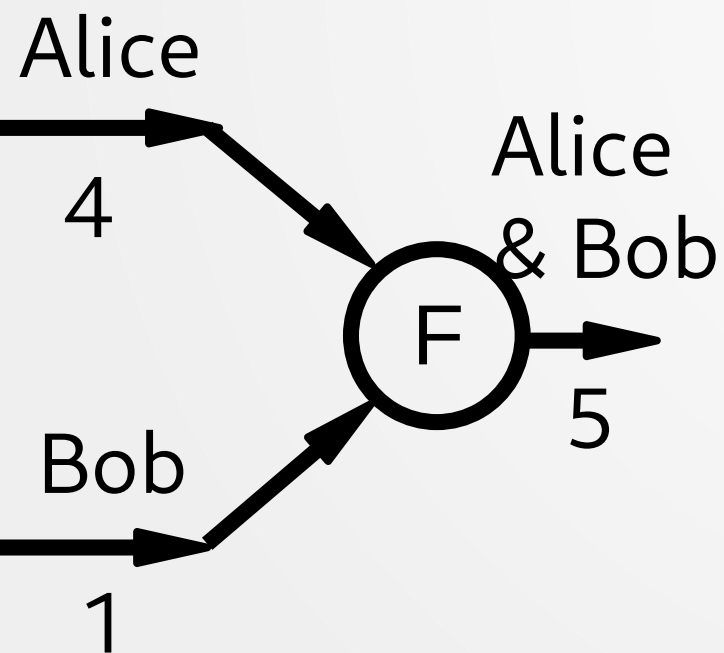
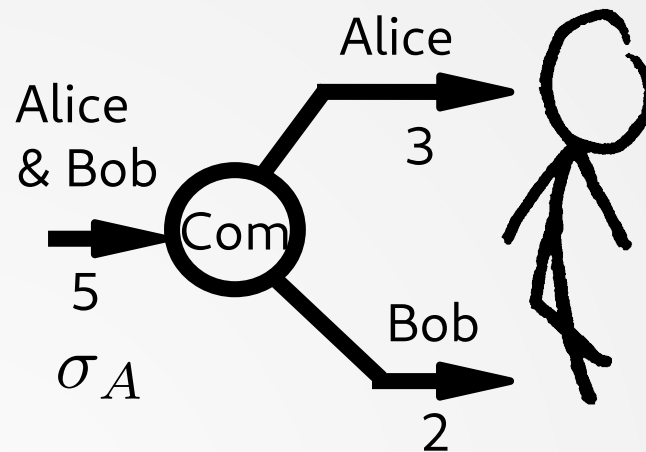
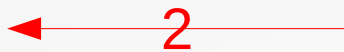
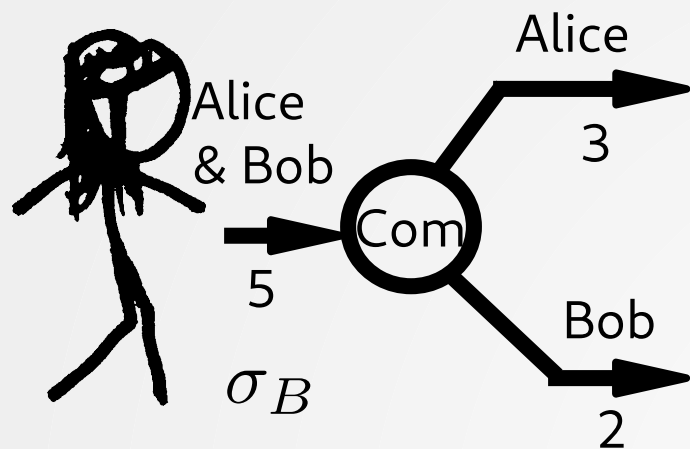
Bob  
→  
1

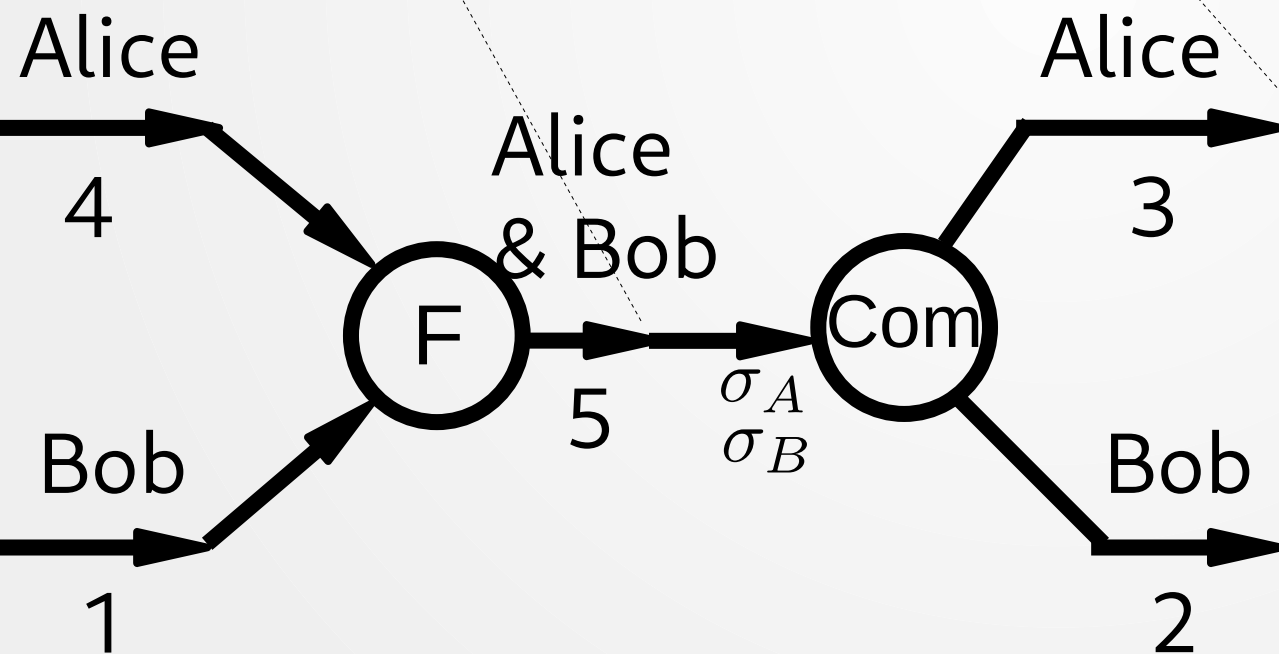
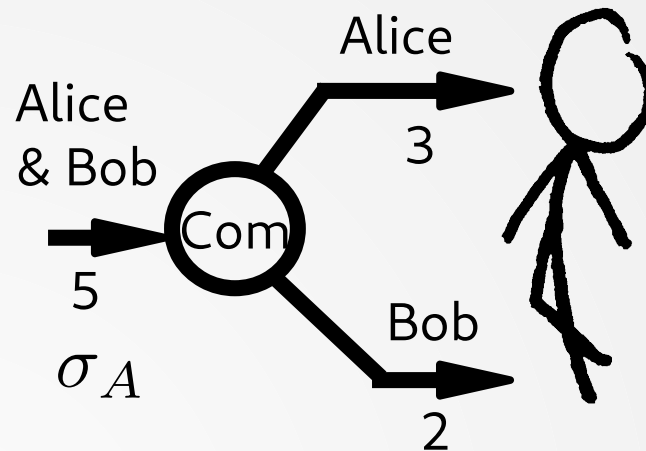
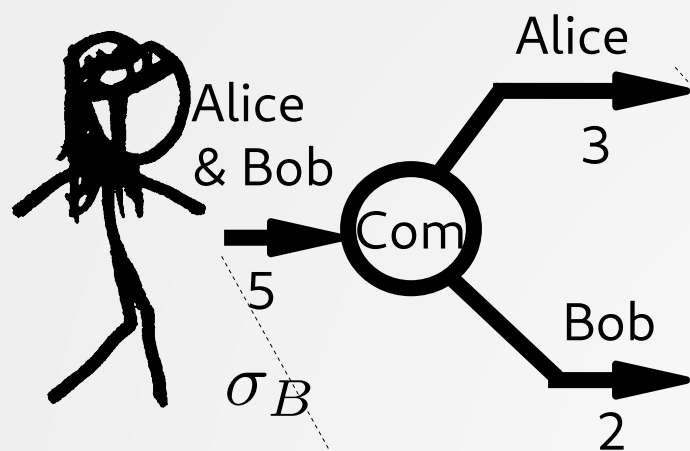


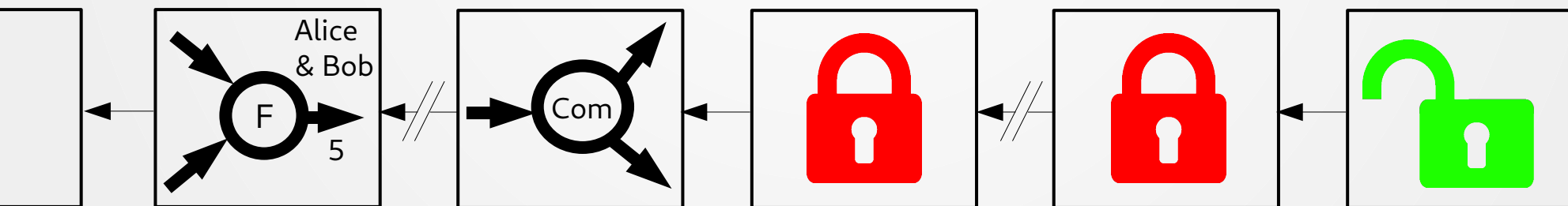
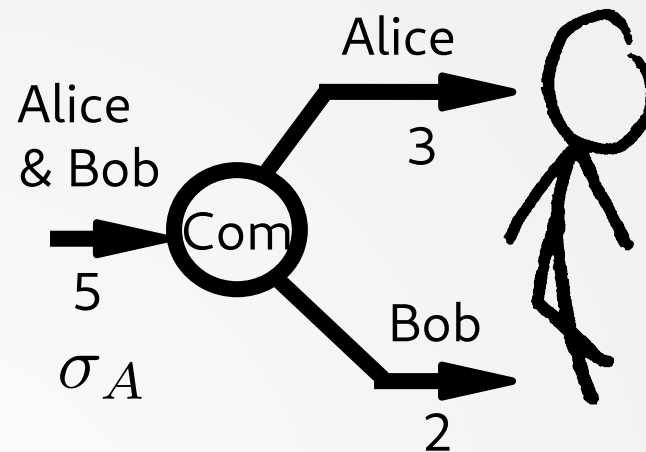
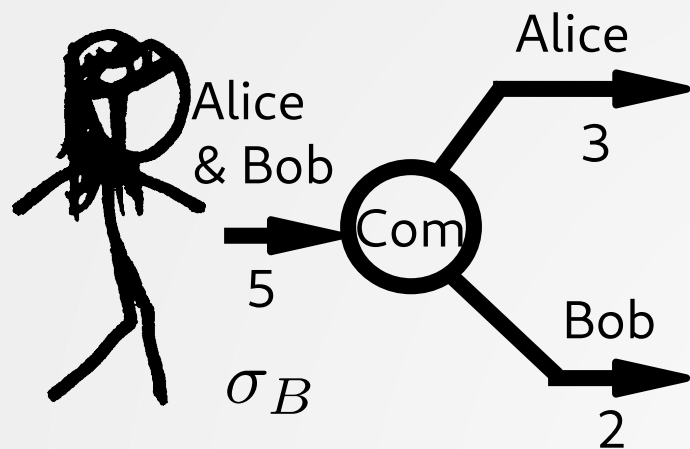






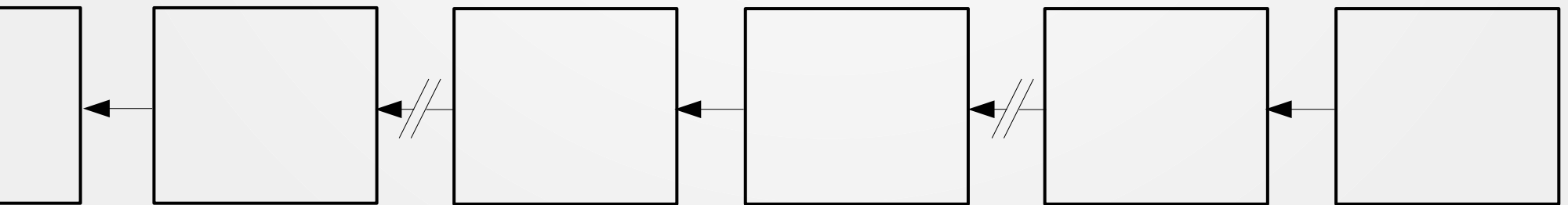


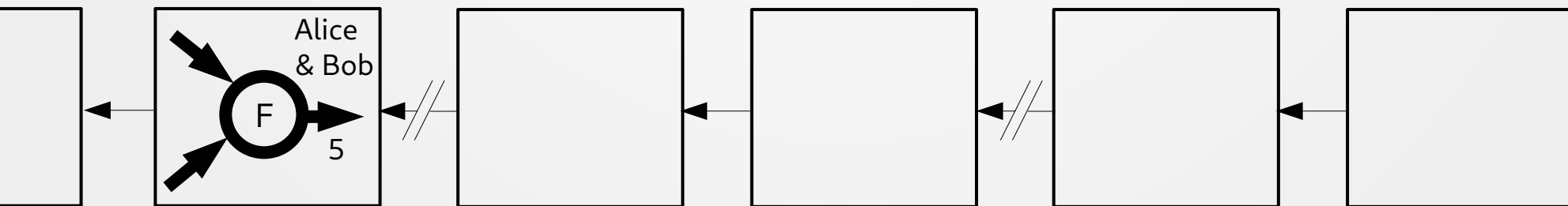
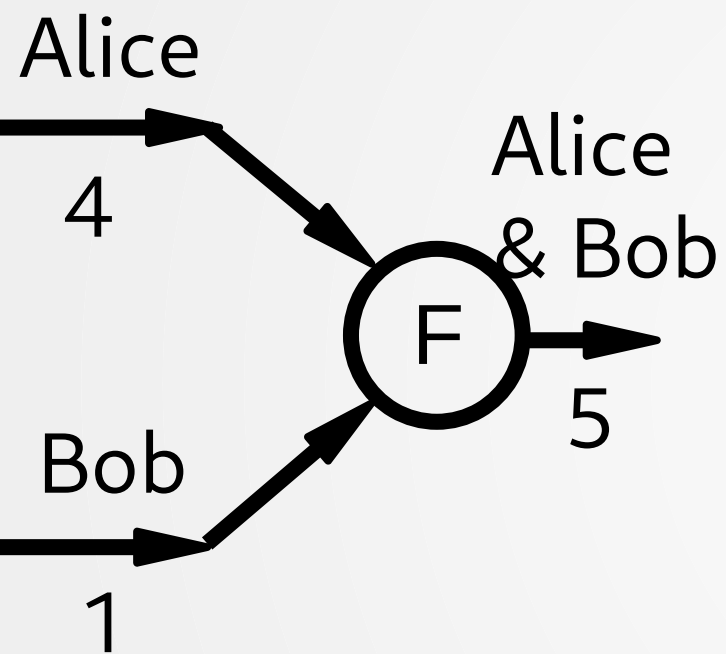


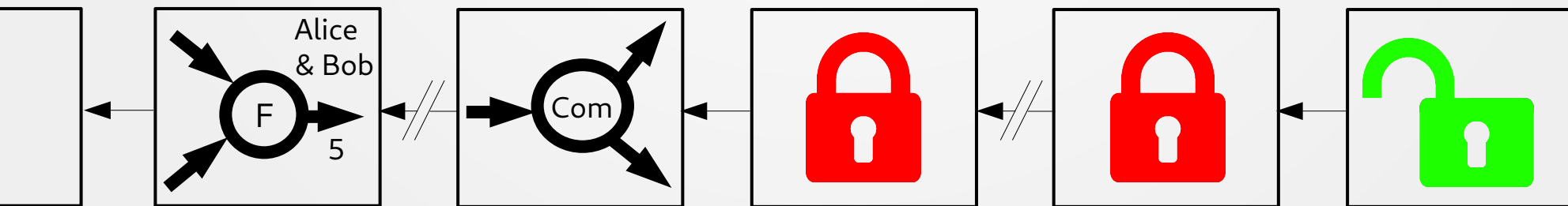
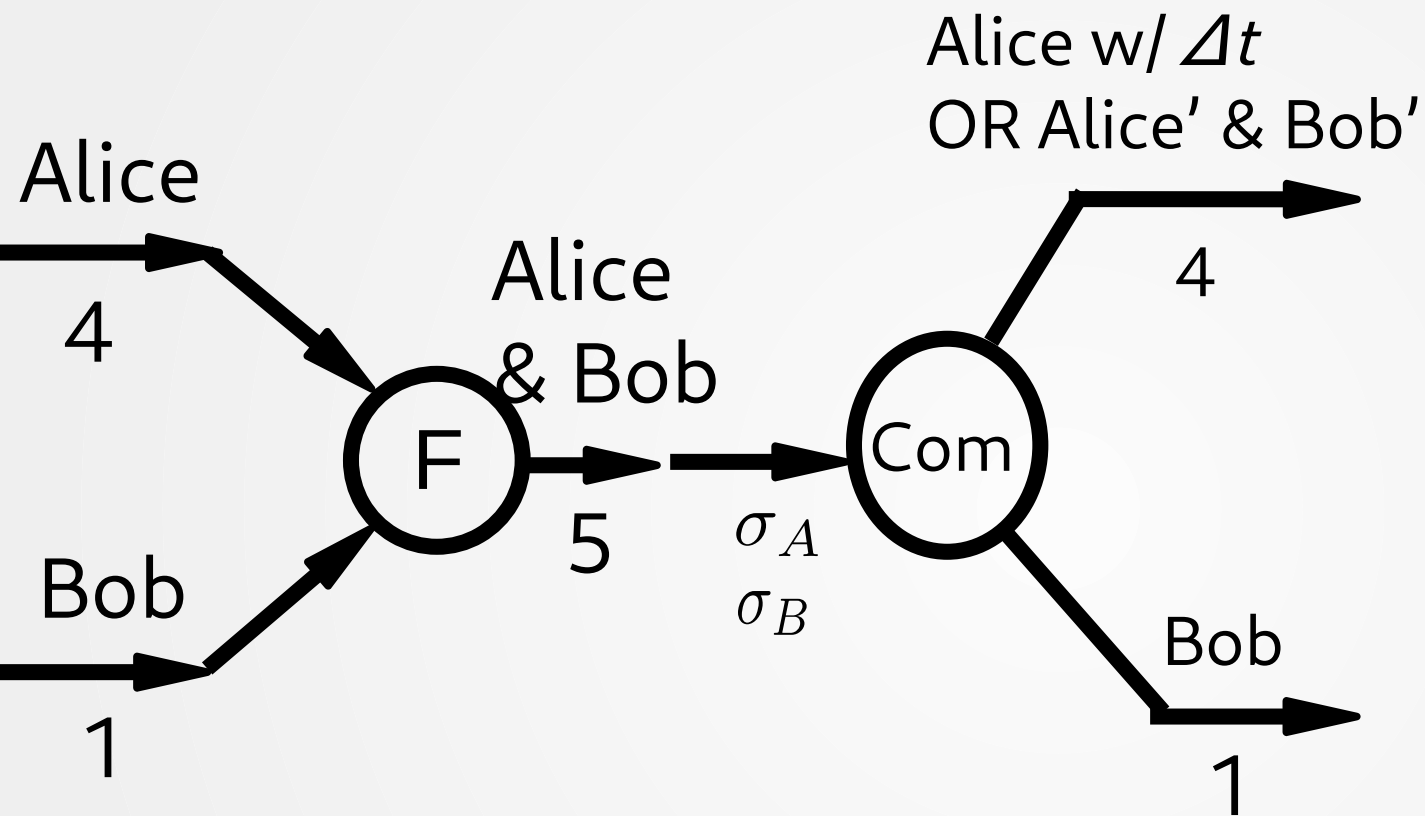


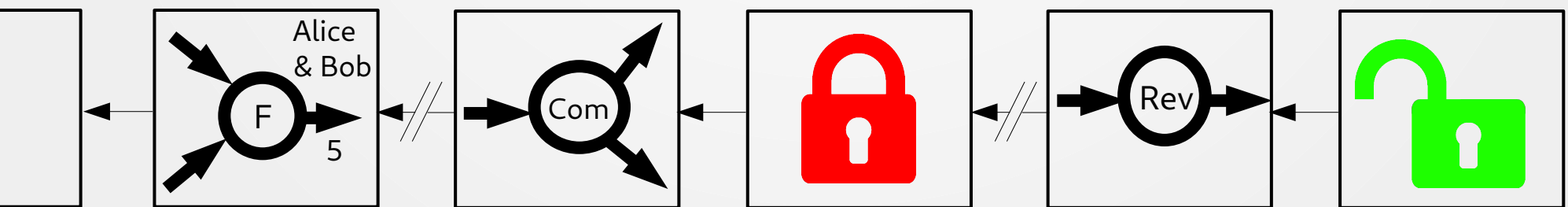
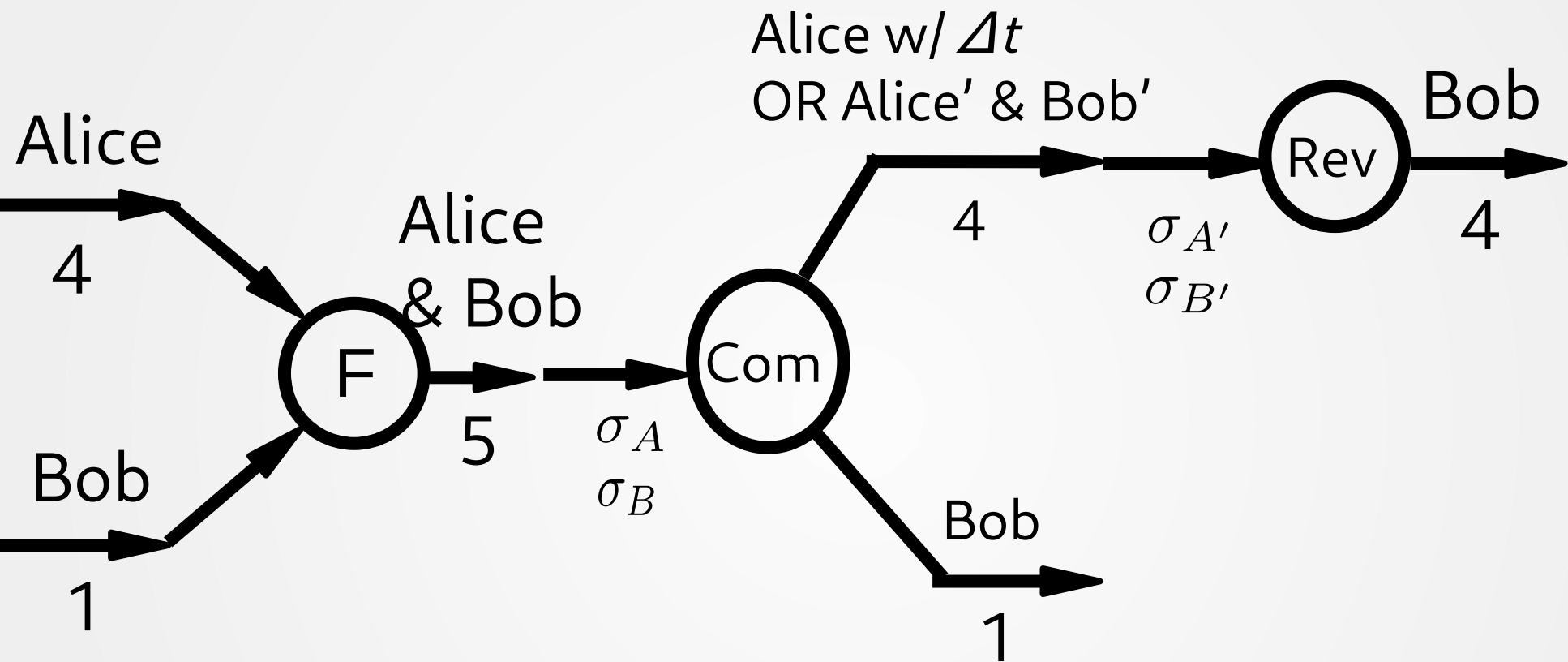
Alice  
→  
4

Bob  
→  
1

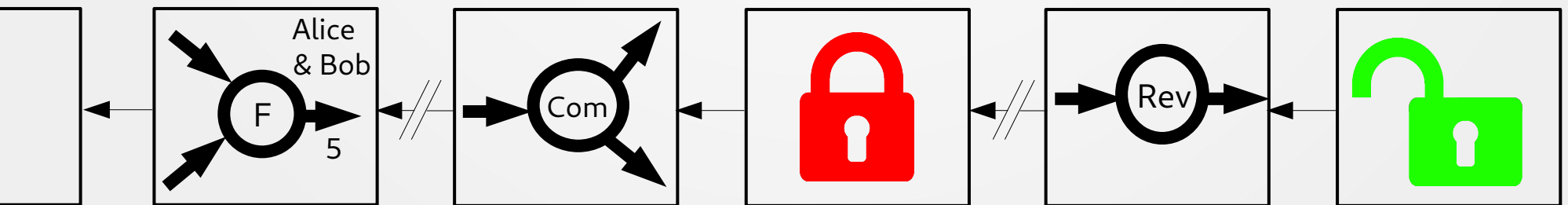
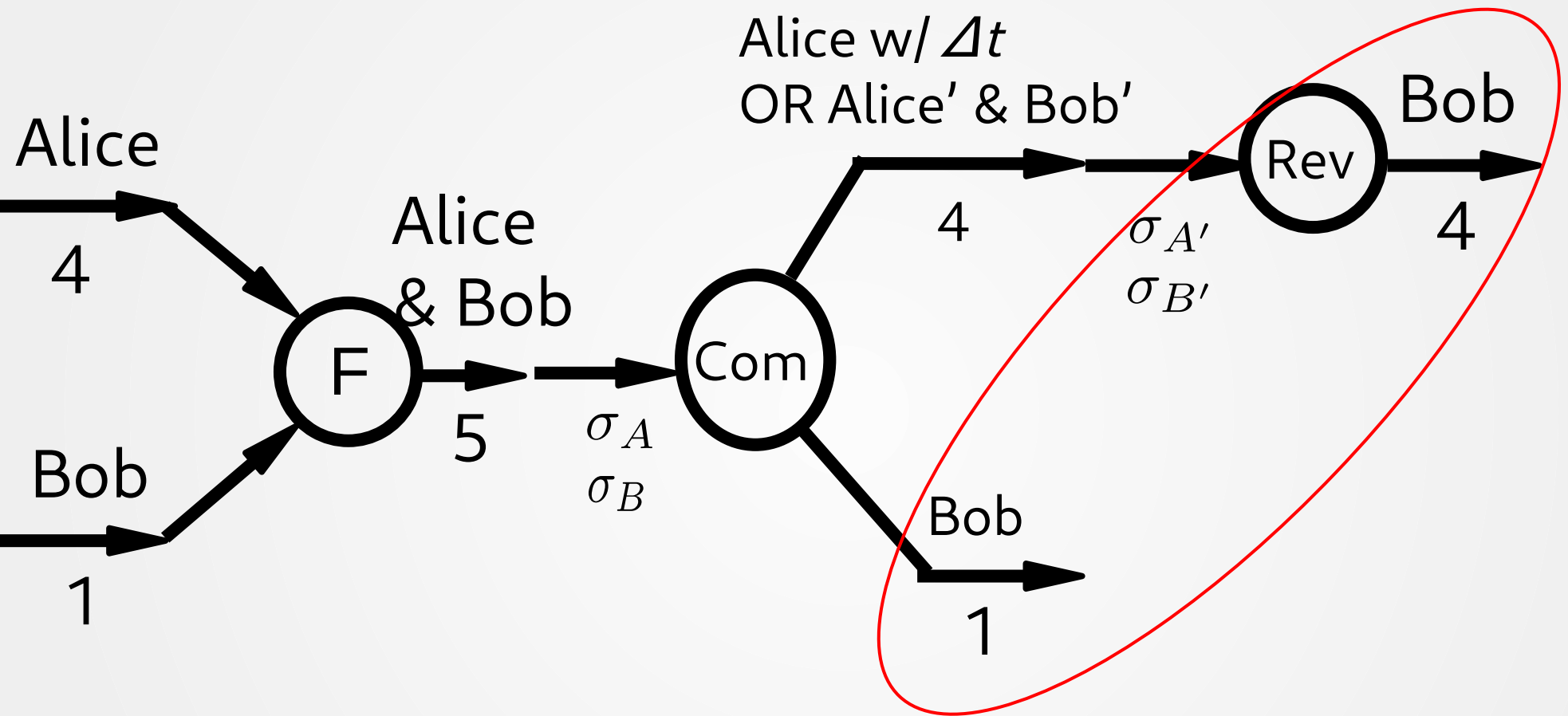








Dispute period  $t$



Dispute period  $t$



# Part 1

## Multi-hop payments

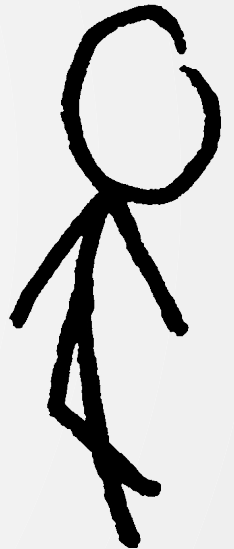
# Multi-hop payments



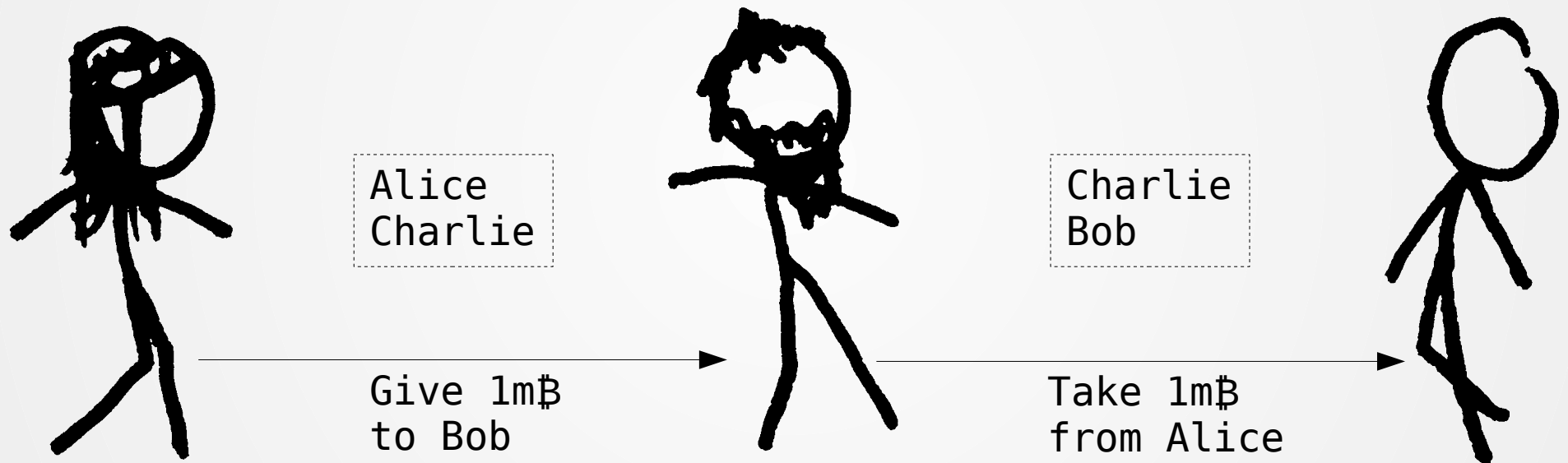
Alice  
Charlie



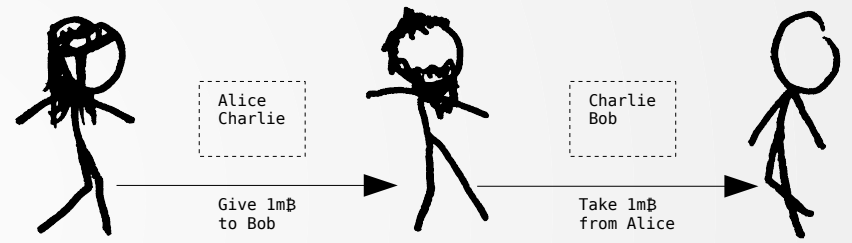
Charlie  
Bob



# Multi-hop payments

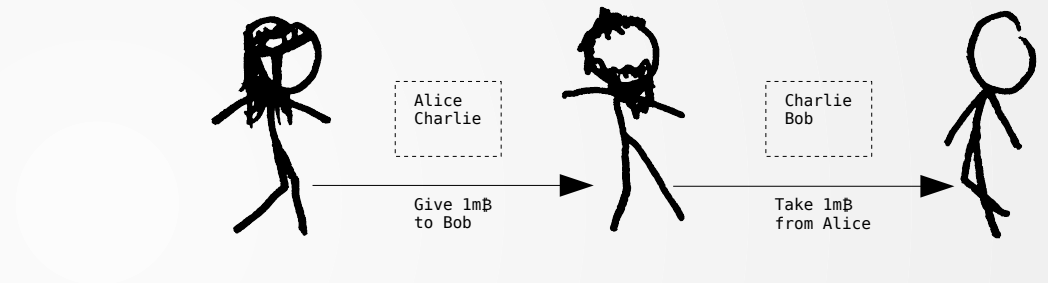


# Why no one can cheat?



# Why no one can cheat?

## HTLC

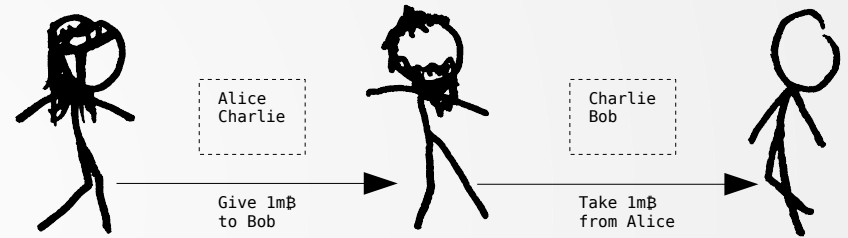


"I, Alice, promise to pay Charlie 1 coin if he reveals a preimage of **0xabcdef** within an hour"

# Why no one can cheat?

## HTLC

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

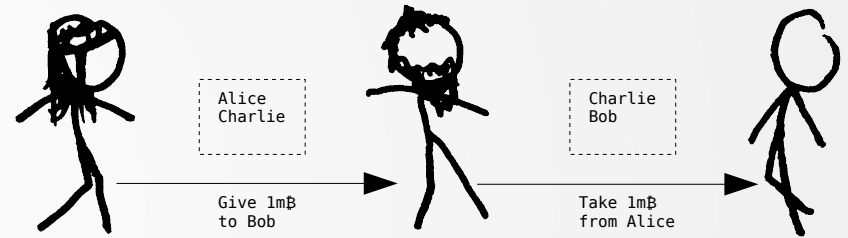


- Bob chooses random **R**, computes  **$h=H(R)$**
- Bob sends **h** to Alice

# Why no one can cheat?

## HTLC

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

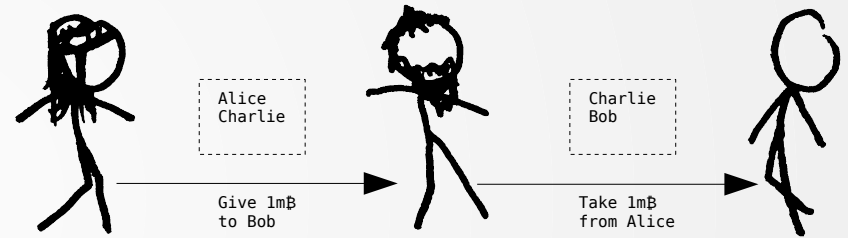


- Bob chooses random **R**, computes  **$h=H(R)$**
- Bob sends **h** to Alice
- Alice signs an **h**-HTLC with Charlie

# Why no one can cheat?

## HTLC

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



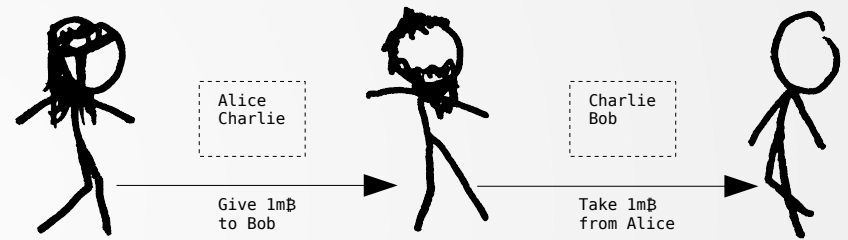
- Bob chooses random **R**, computes  **$h=H(R)$**
- Bob sends **h** to Alice
- Alice signs an **h**-HTLC with Charlie
- Charlie signs an **h**-HTLC with Bob



# Why no one can cheat?

## HTLC

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

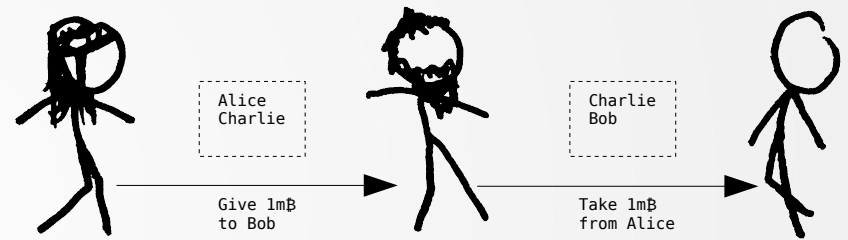


- Bob chooses random **R**, computes  **$h=H(R)$**
- Bob sends **h** to Alice
- Alice signs an **h**-HTLC with Charlie
- Charlie signs an **h**-HTLC with Bob
- Bob reveals **R** to Charlie, gets 1 coin

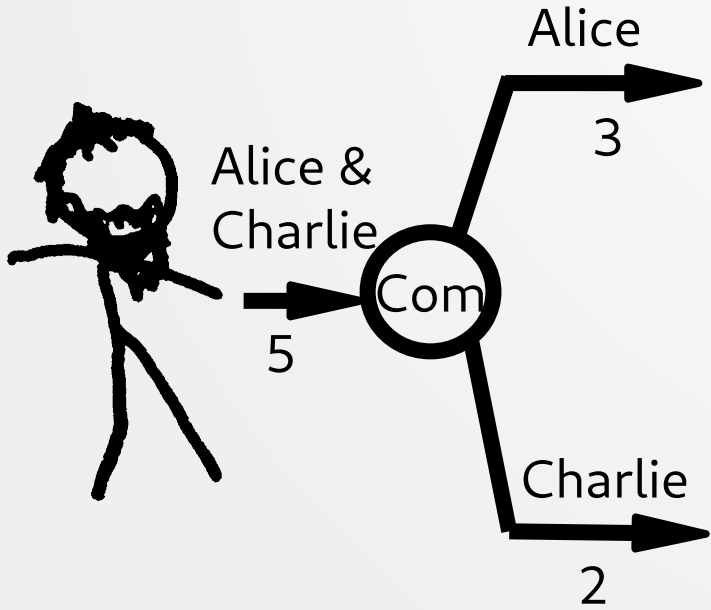
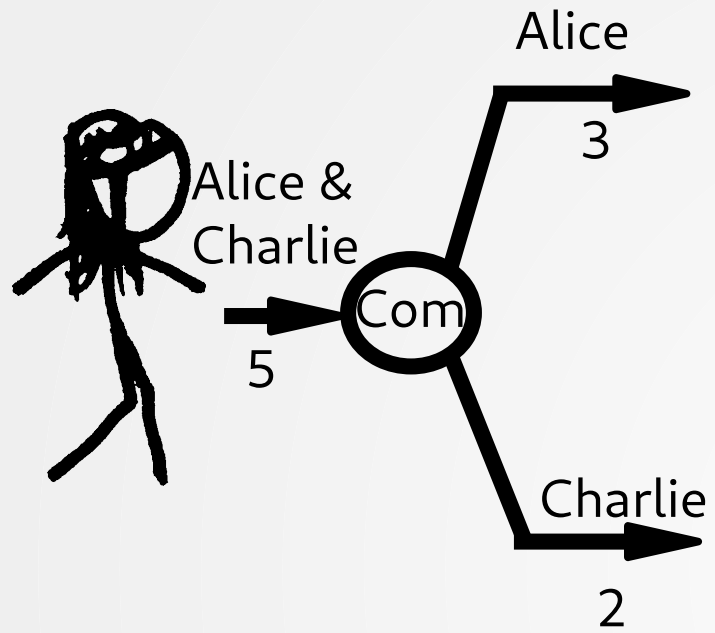
# Why no one can cheat?

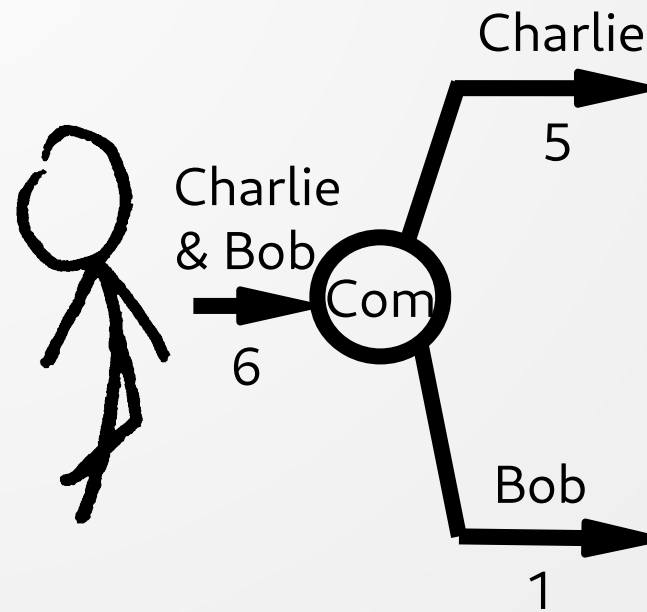
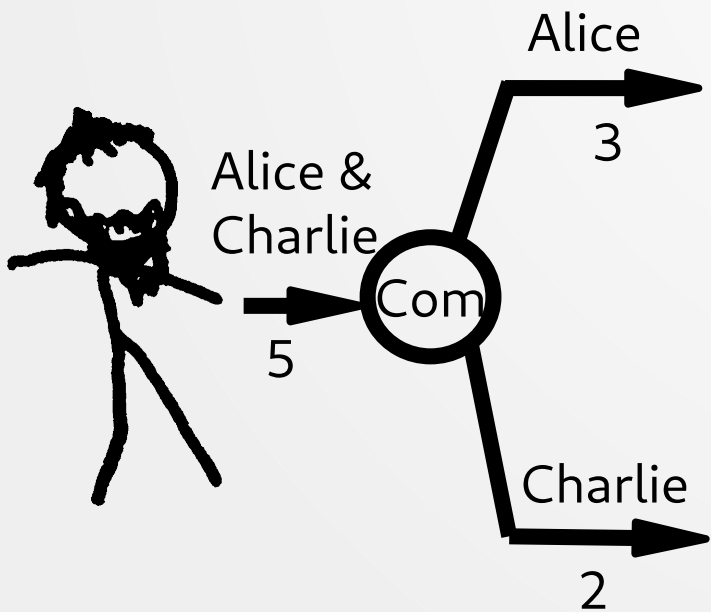
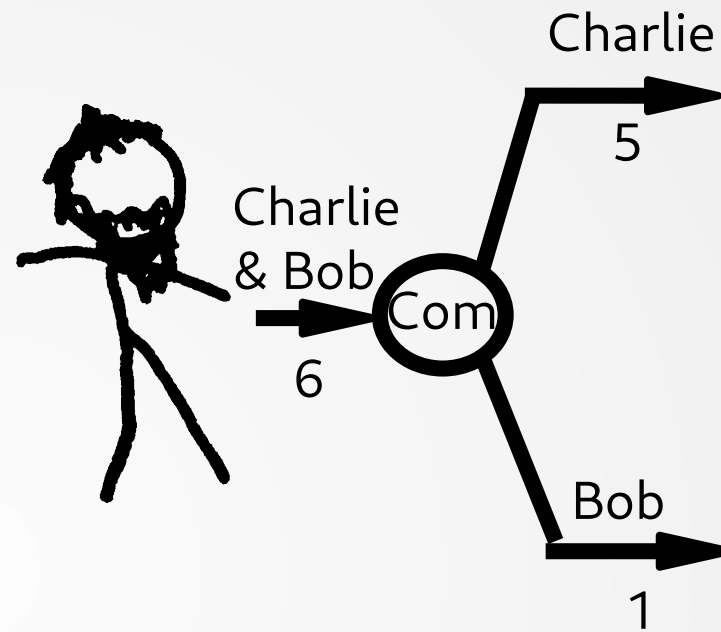
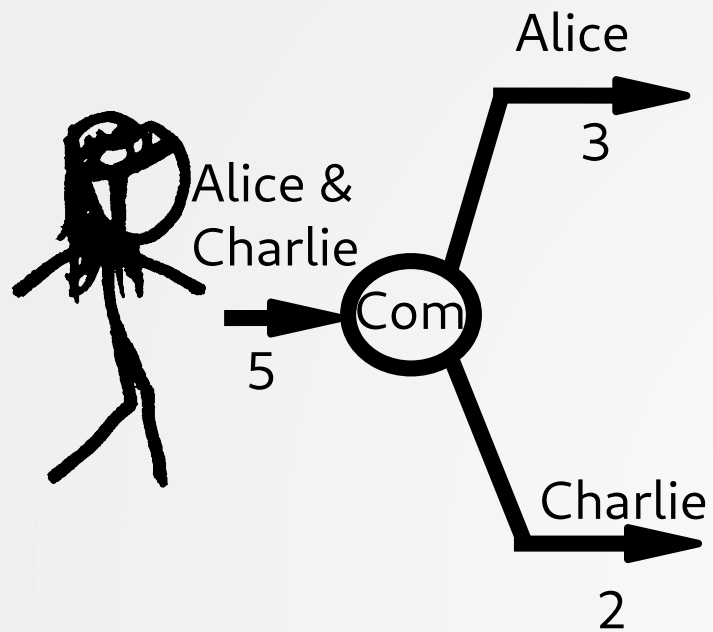
## HTLC

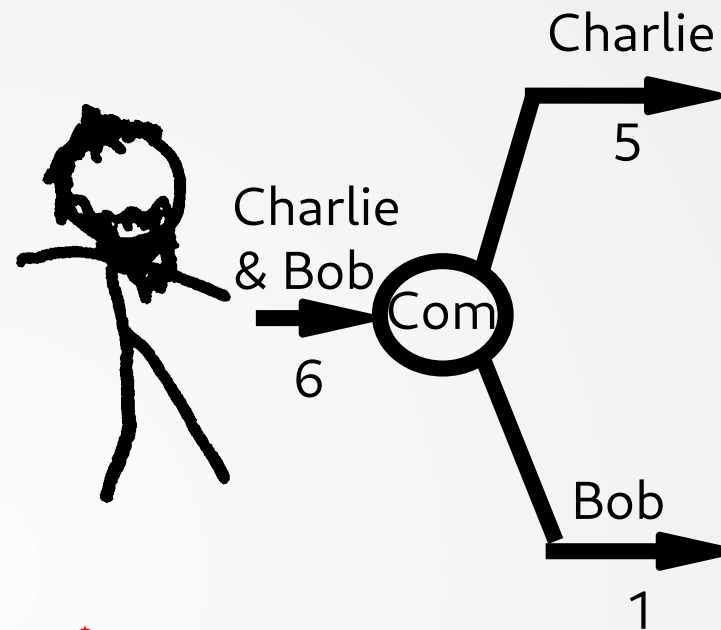
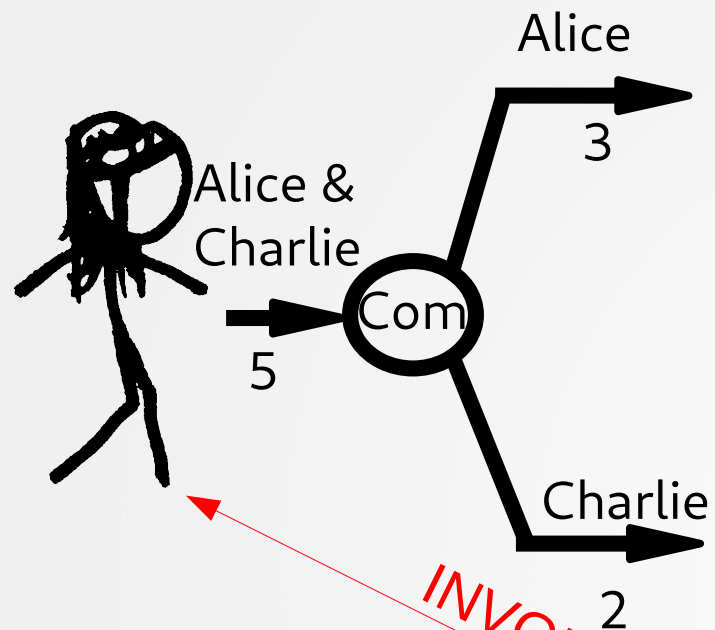
"I, Alice, promise to pay Charlie 1 coin if he reveals a preimage of `0xabcdef` within an hour"



- 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



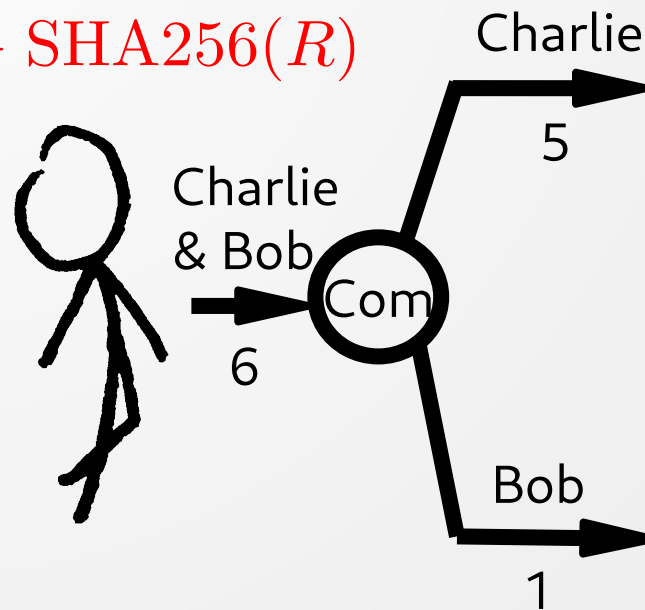
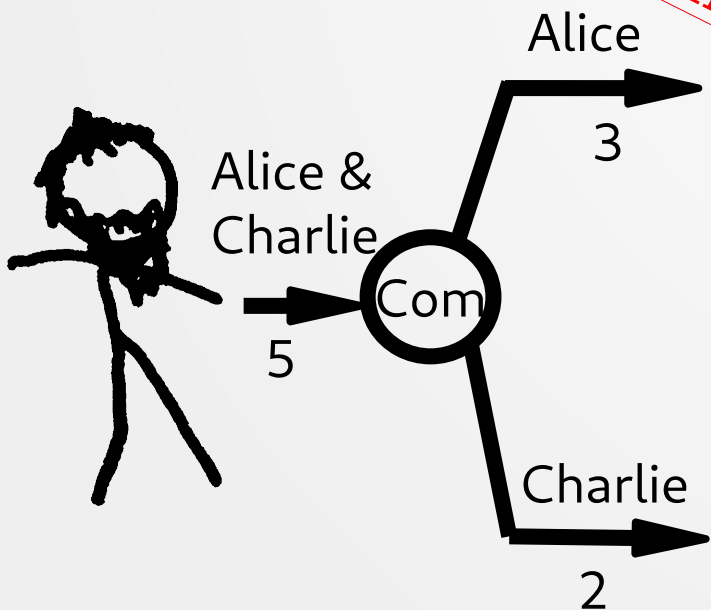


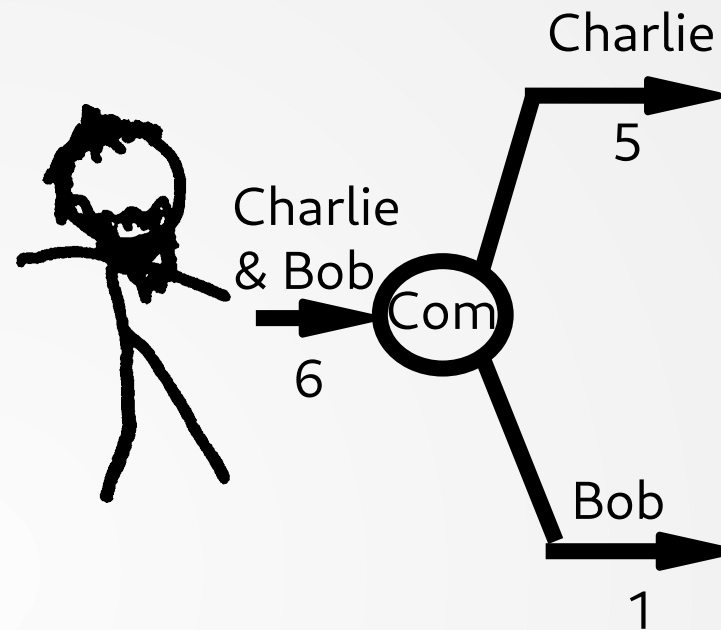
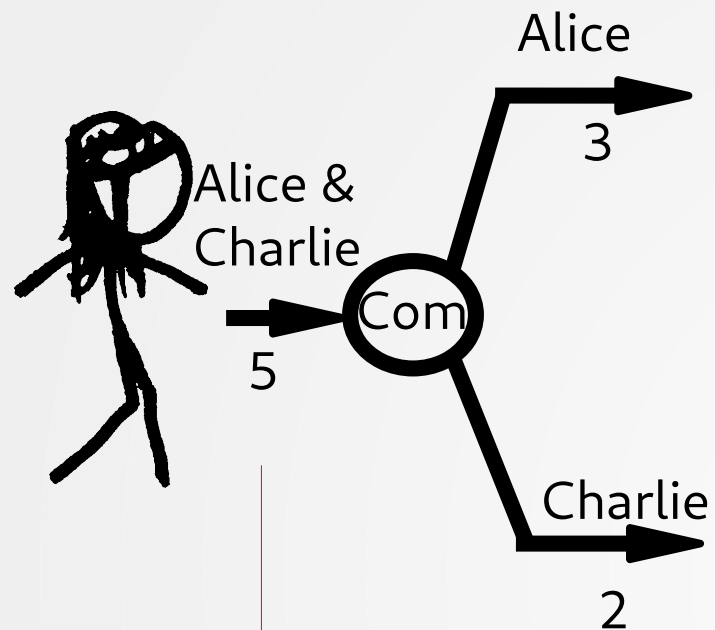


$$R \xleftarrow{\$} \{0, 1\}^{256}$$

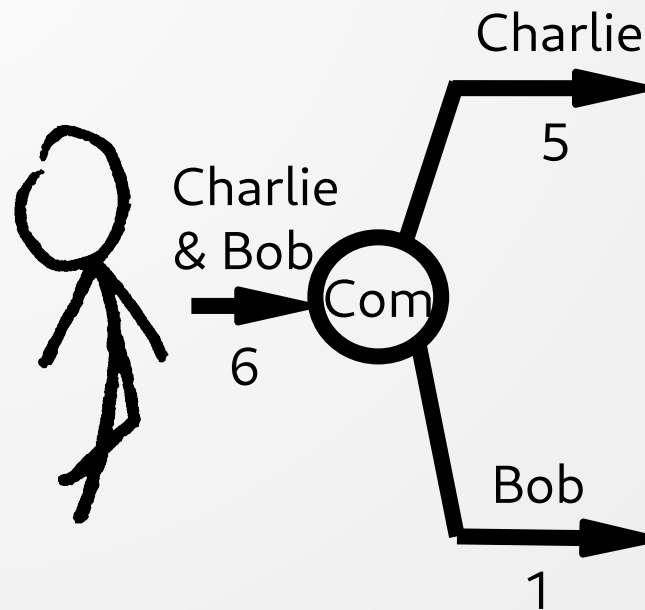
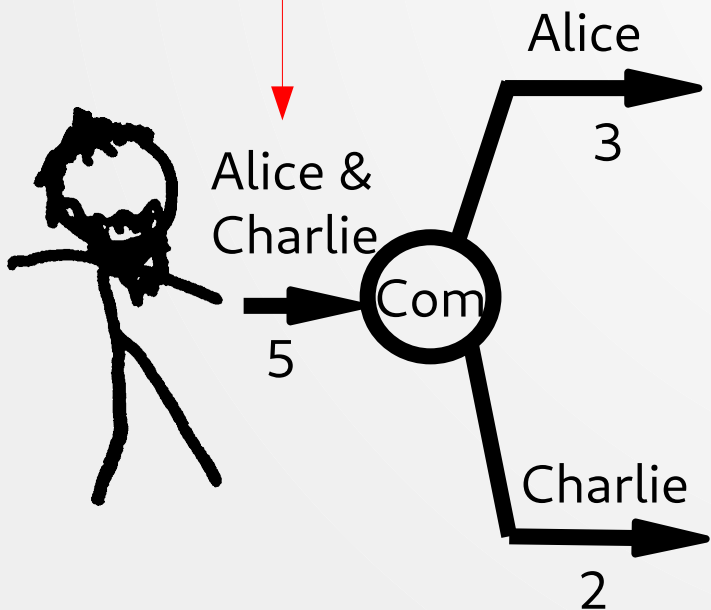
$$h \leftarrow \text{SHA256}(R)$$

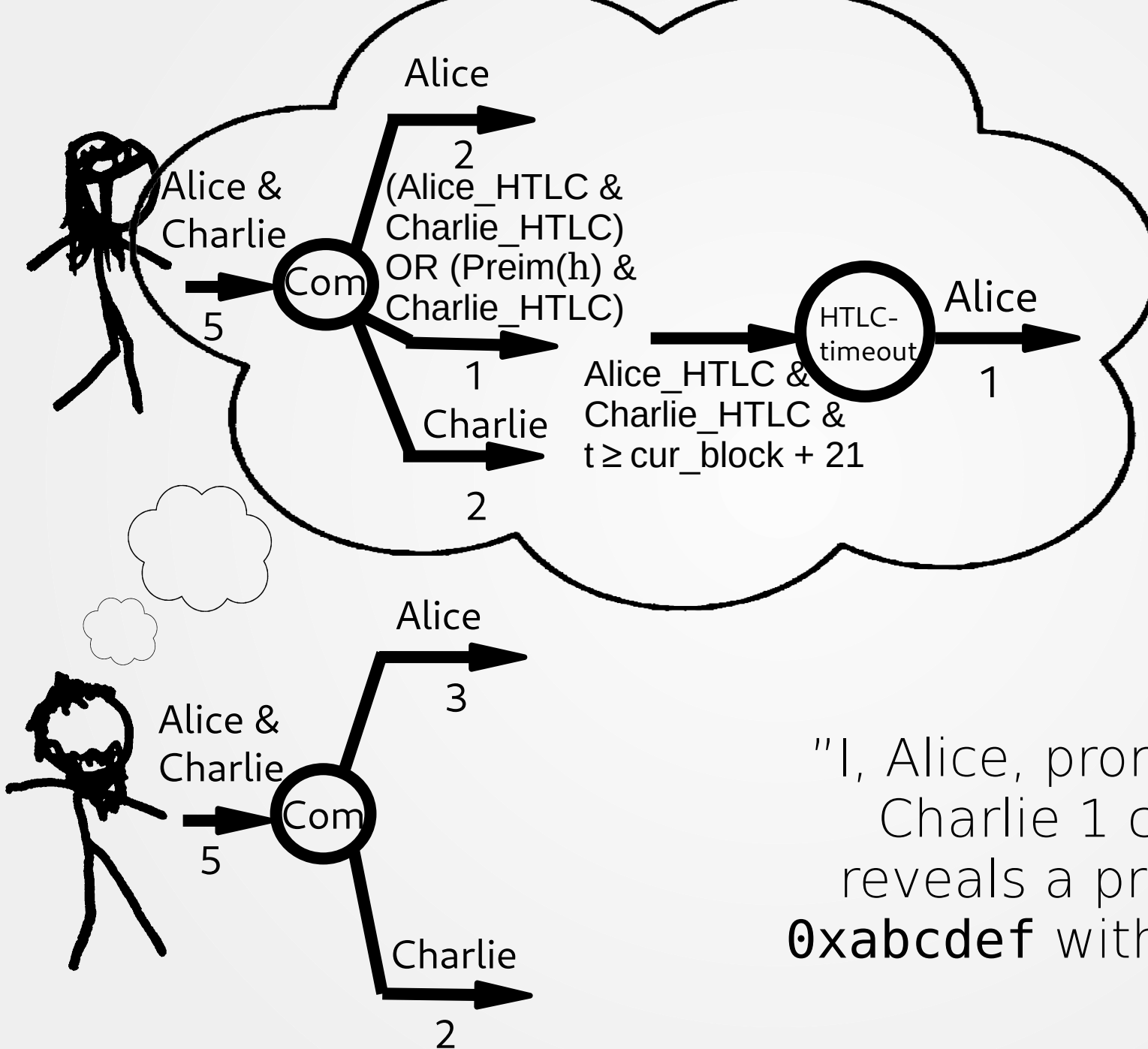
*INVOICE, h, amt=1*



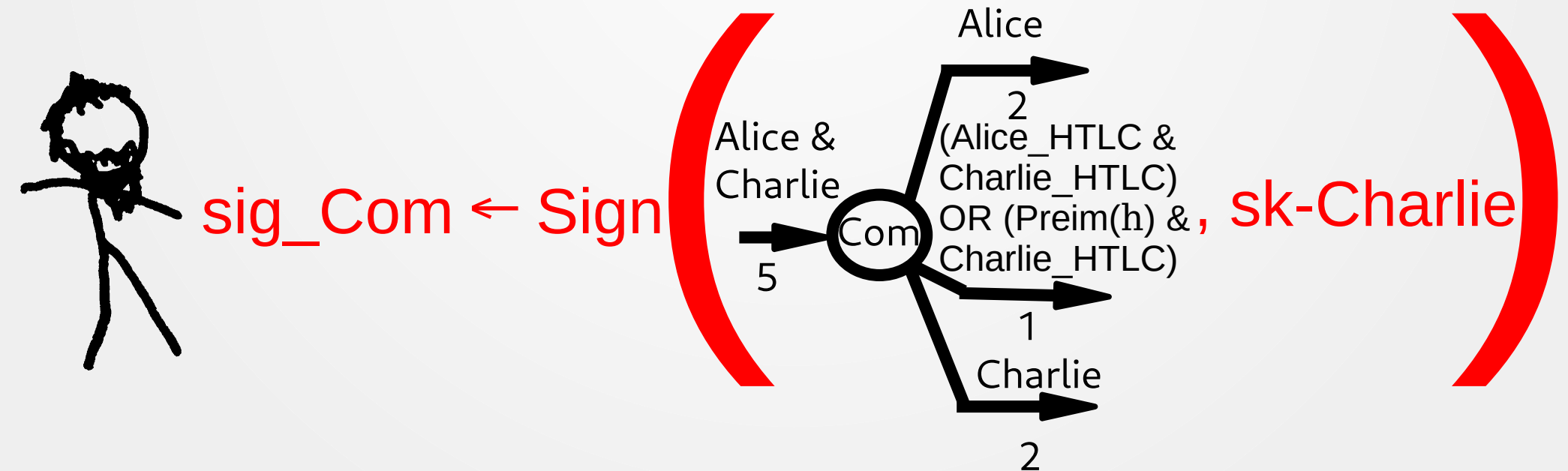
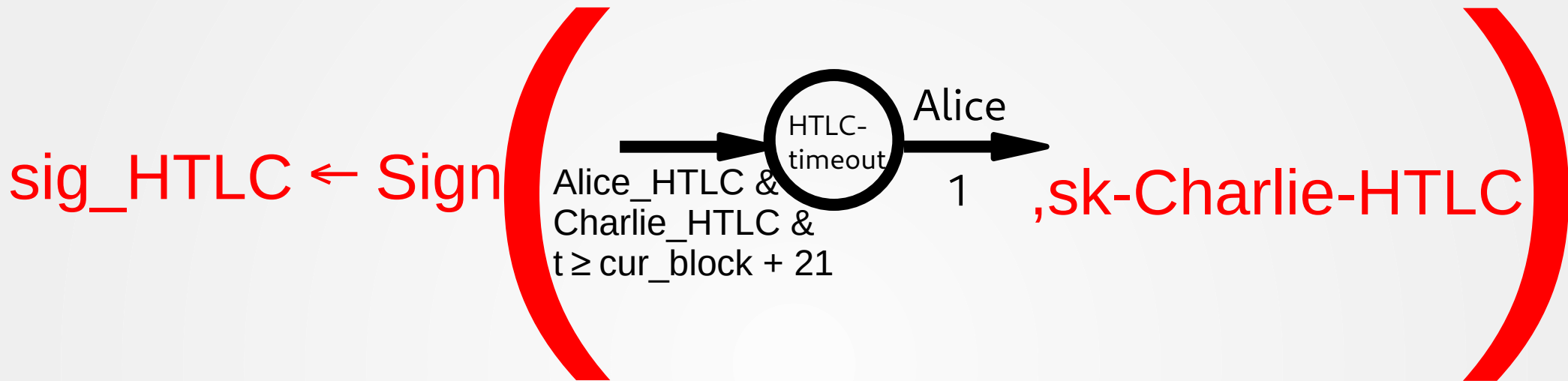


ADD\_HTLC, h, amt=1, del=21

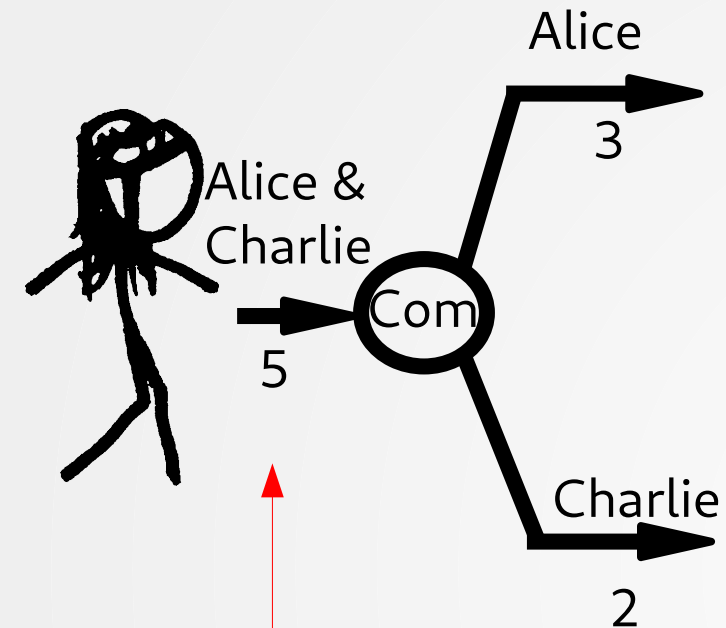




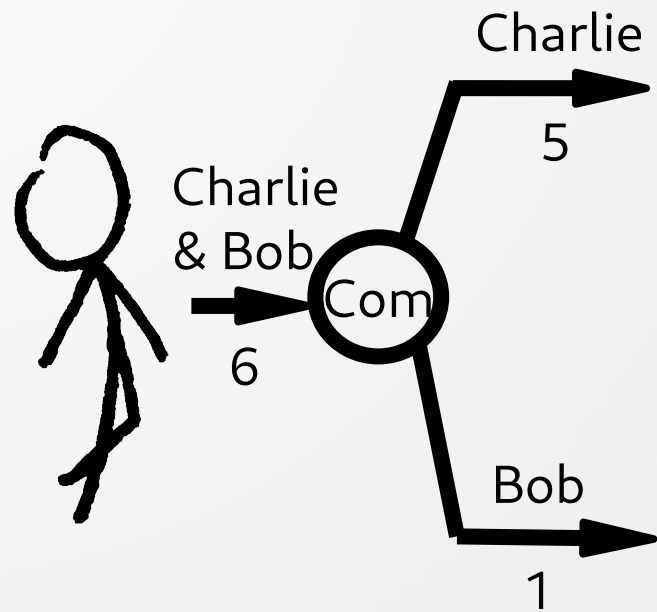
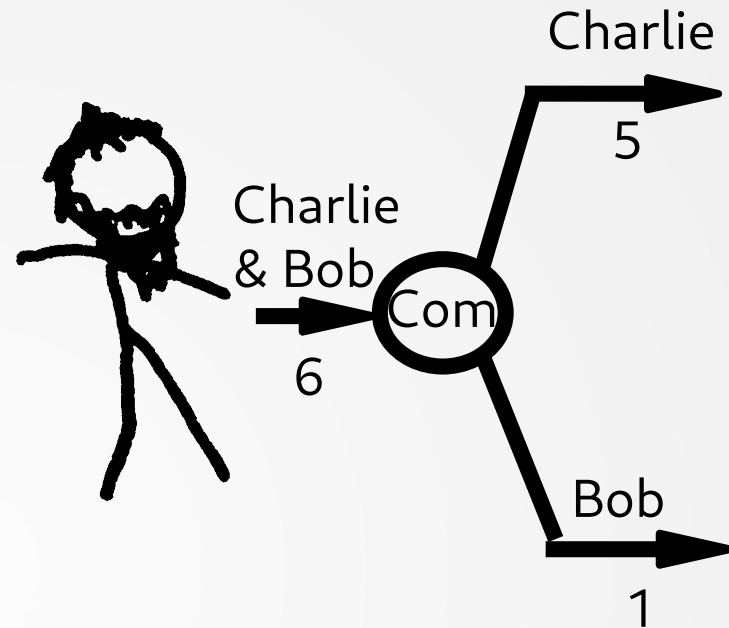
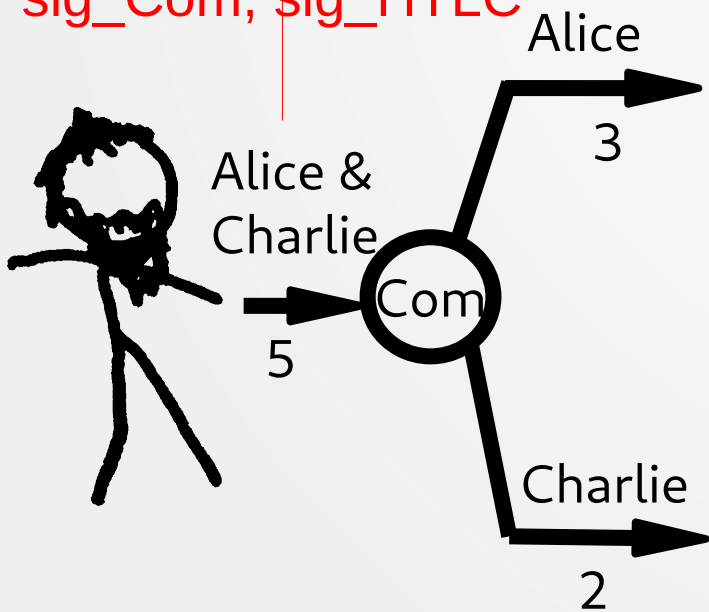
"I, Alice, promise to pay Charlie 1 coin if he reveals a preimage of **0xabcdef** within an hour"

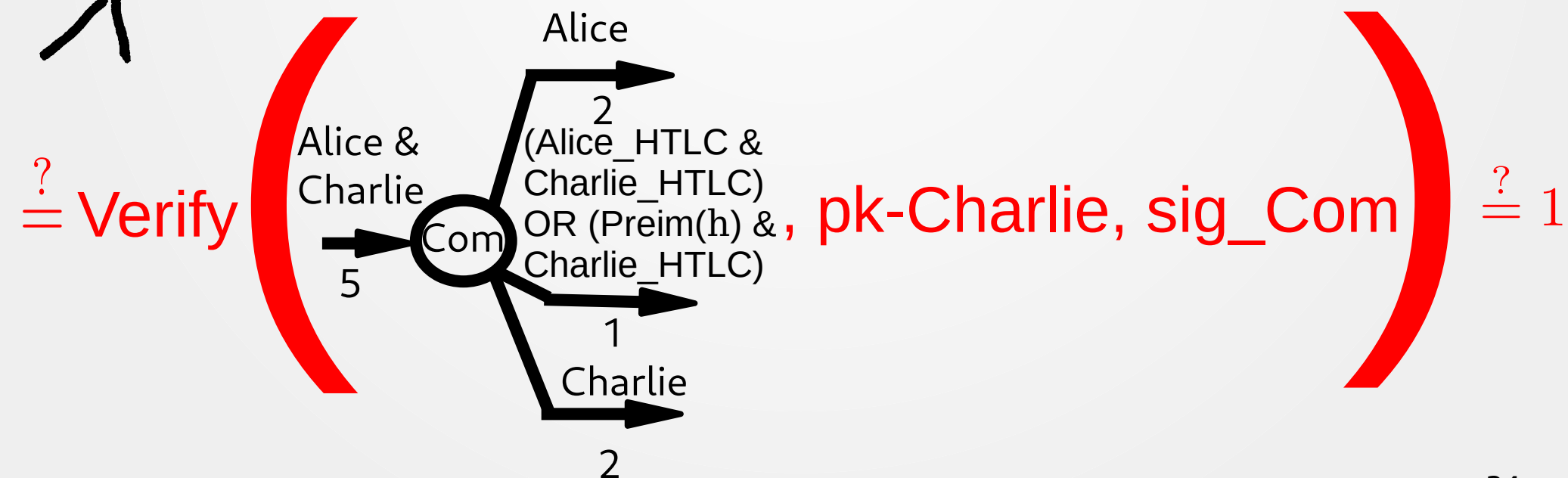
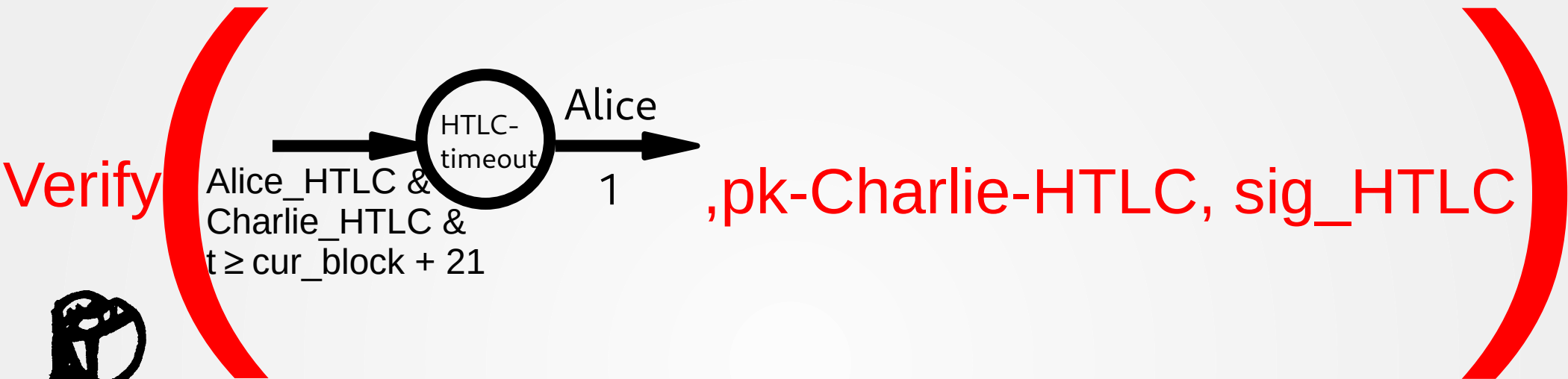


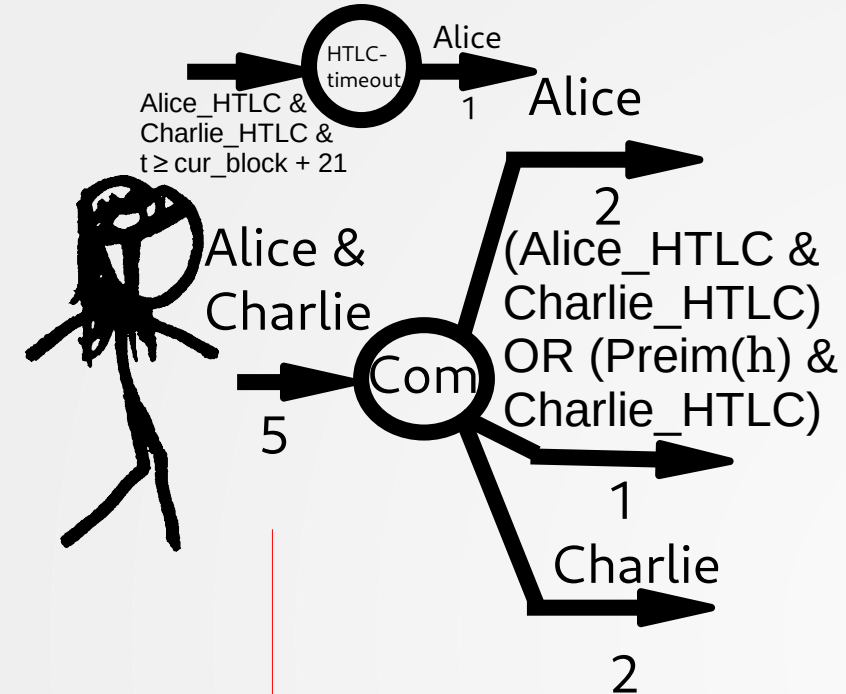




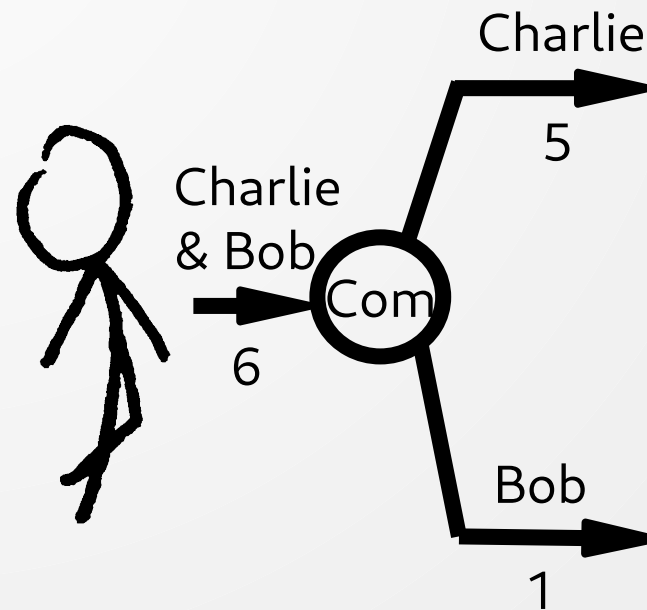
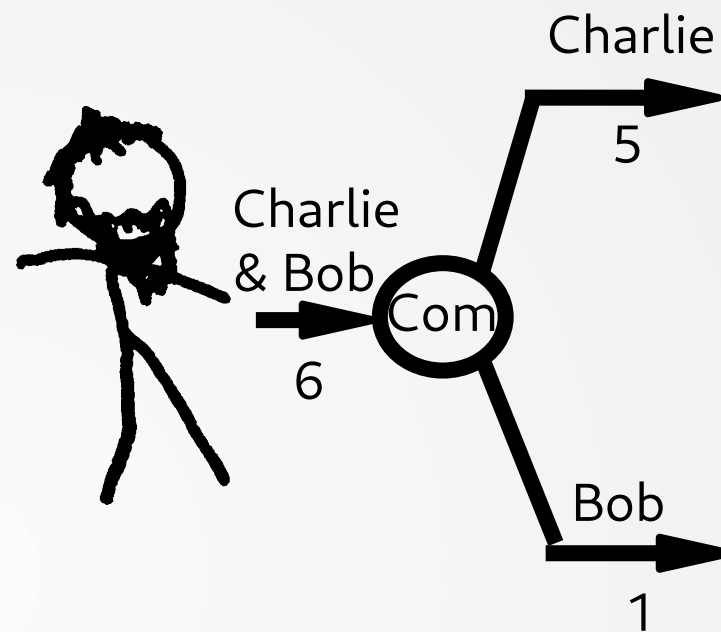
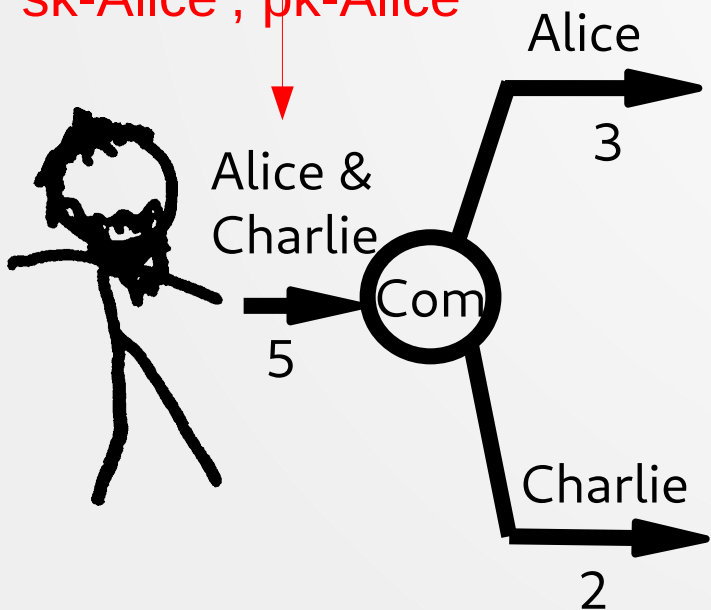
COMMITMENT\_SIGNED,  
sig\_Com, sig\_HTLC

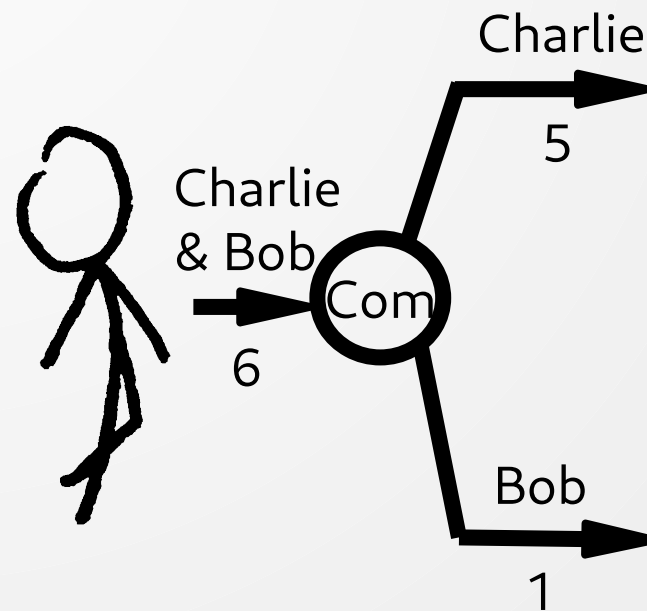
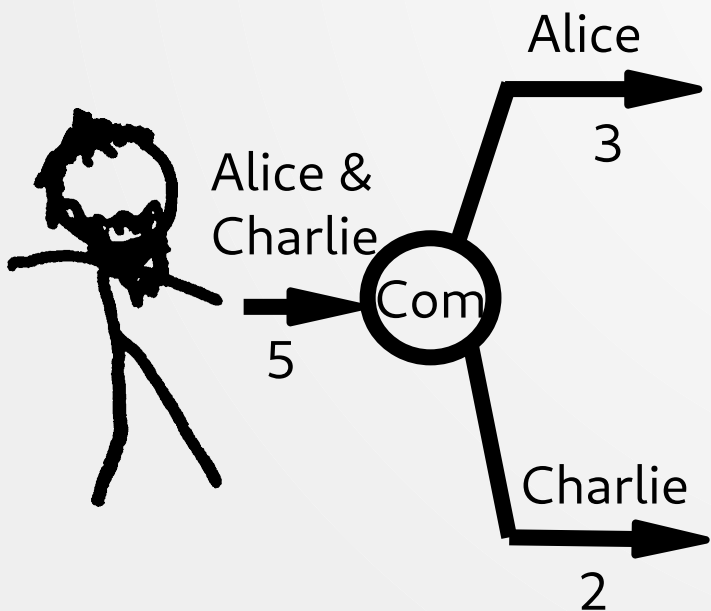
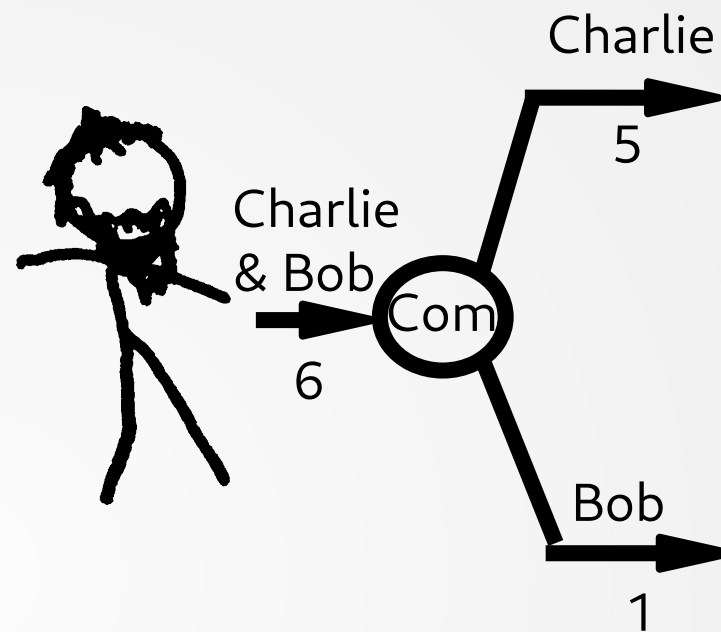
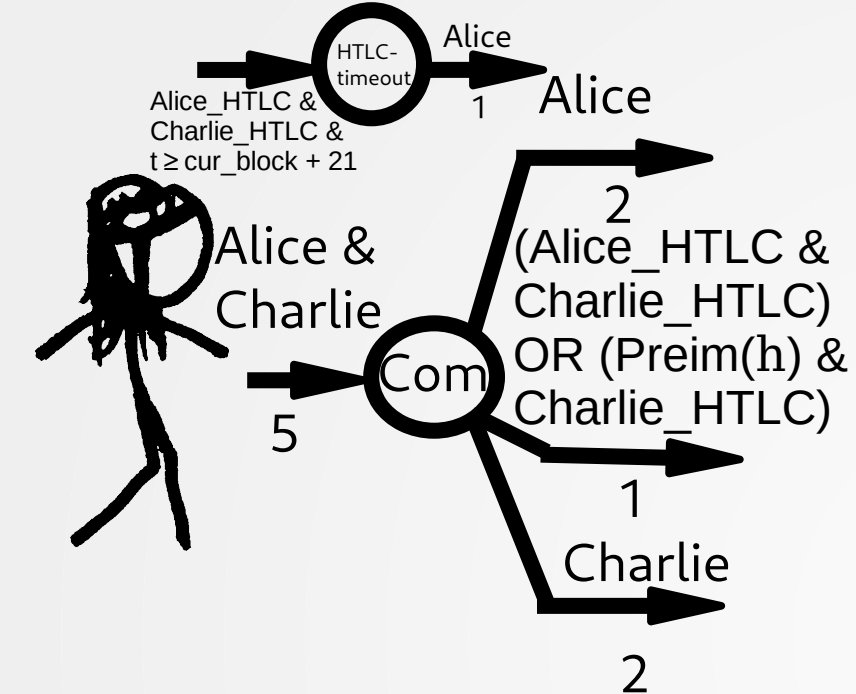




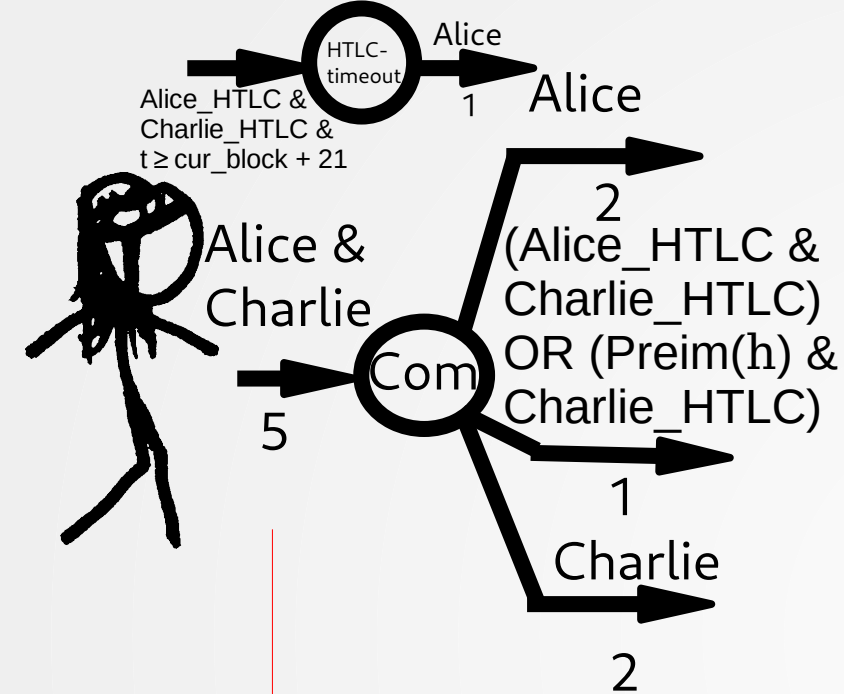


REVOKE\_AND\_ACK,  
sk-Alice', pk-Alice''

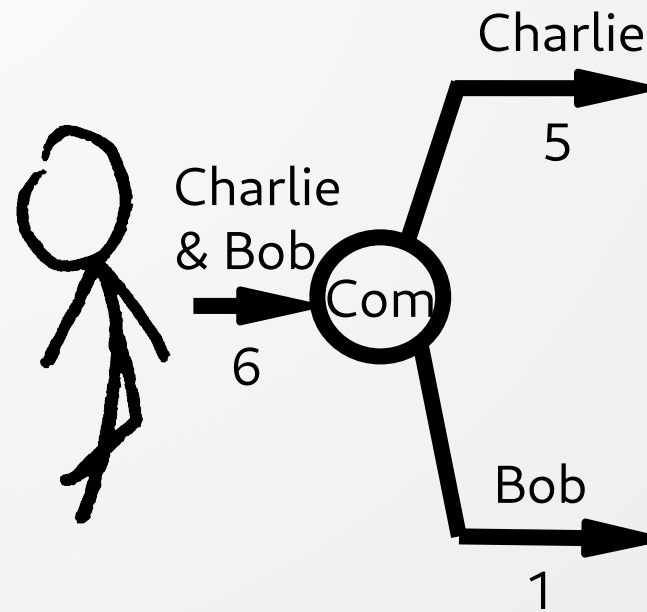
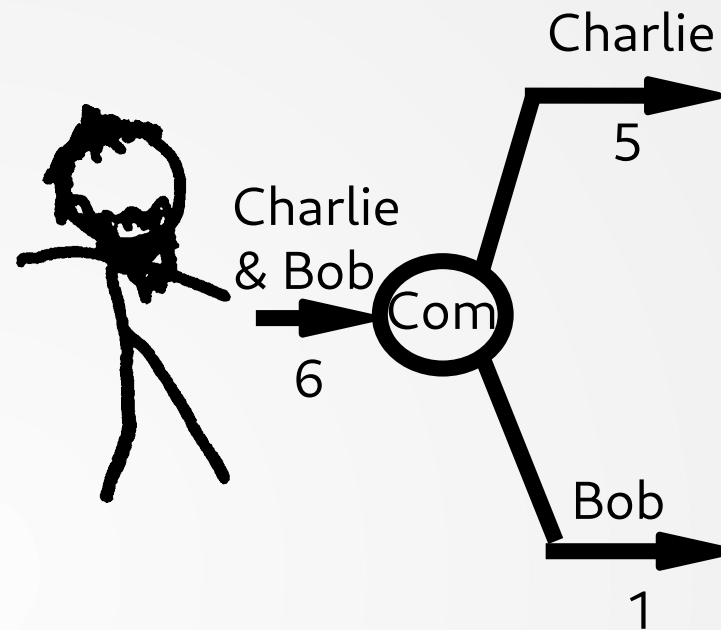
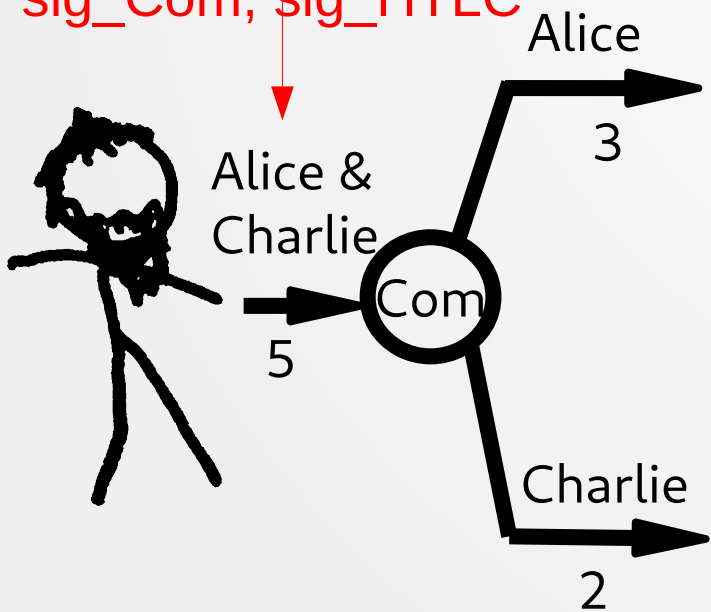


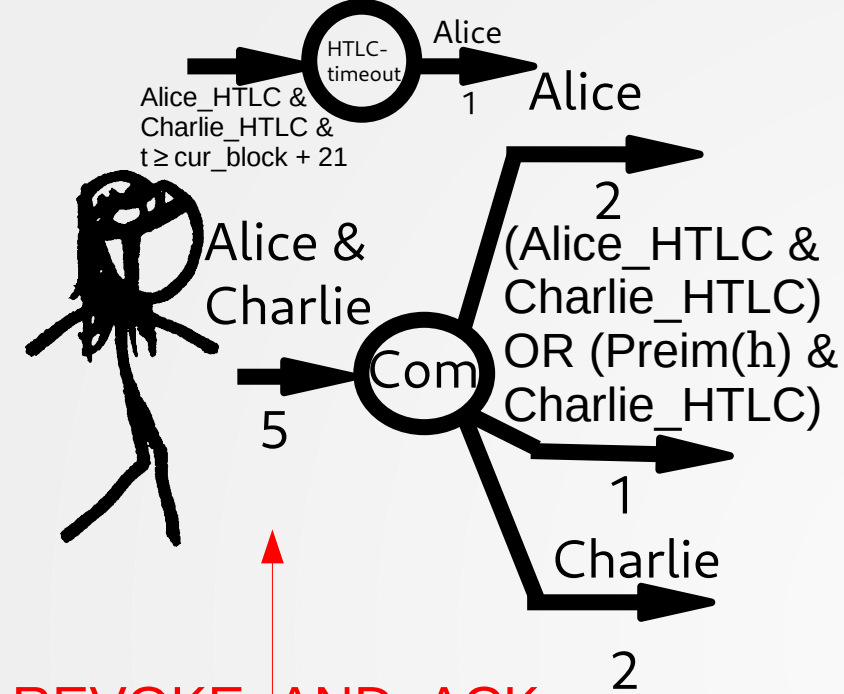


$\text{pk}(\text{sk-Alice}') \stackrel{?}{=} \text{pk-Alice}'$

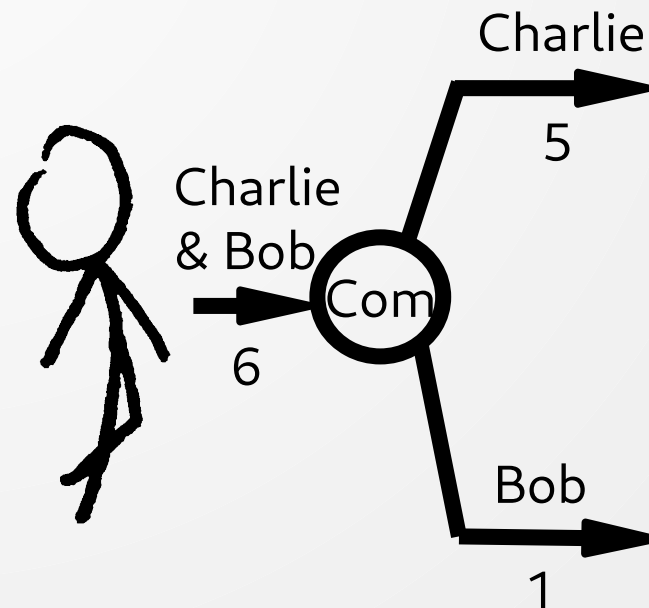
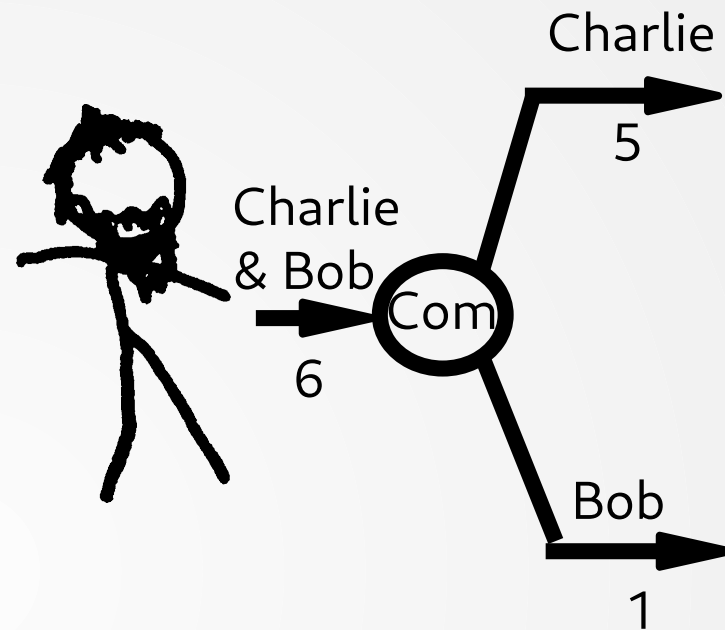
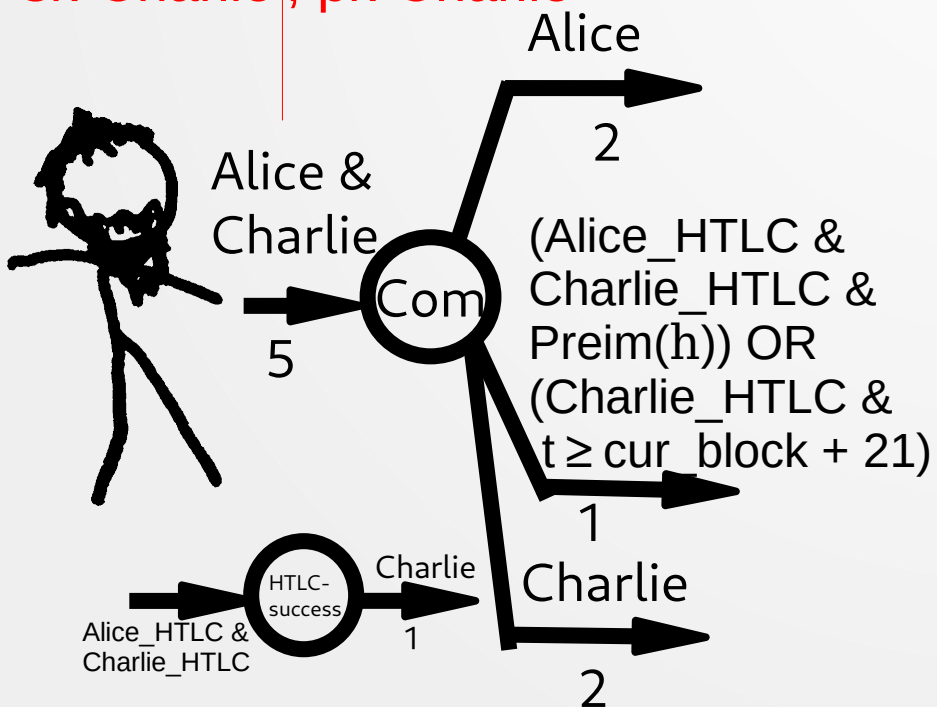


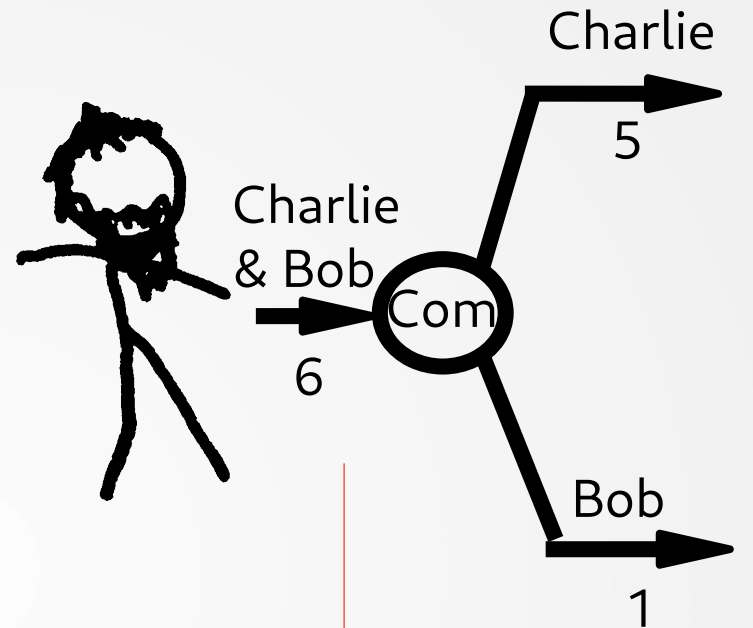
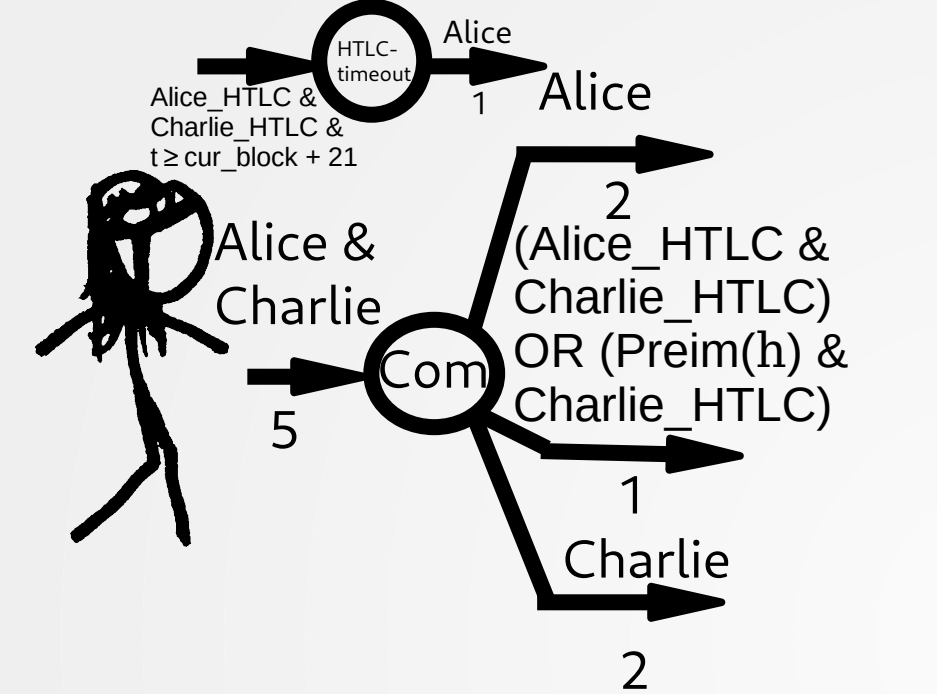
COMMITMENT\_SIGNED,  
sig\_Com, sig\_HTLC



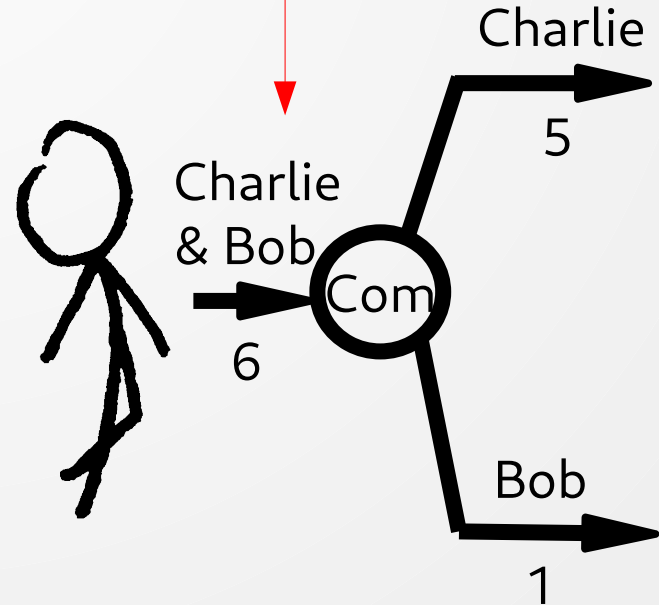
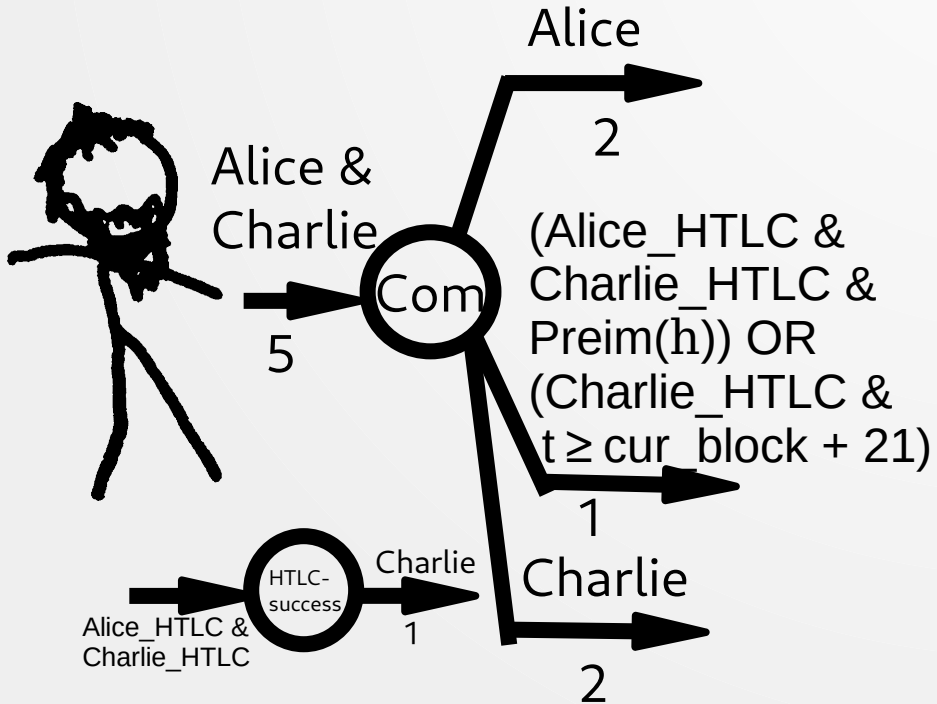


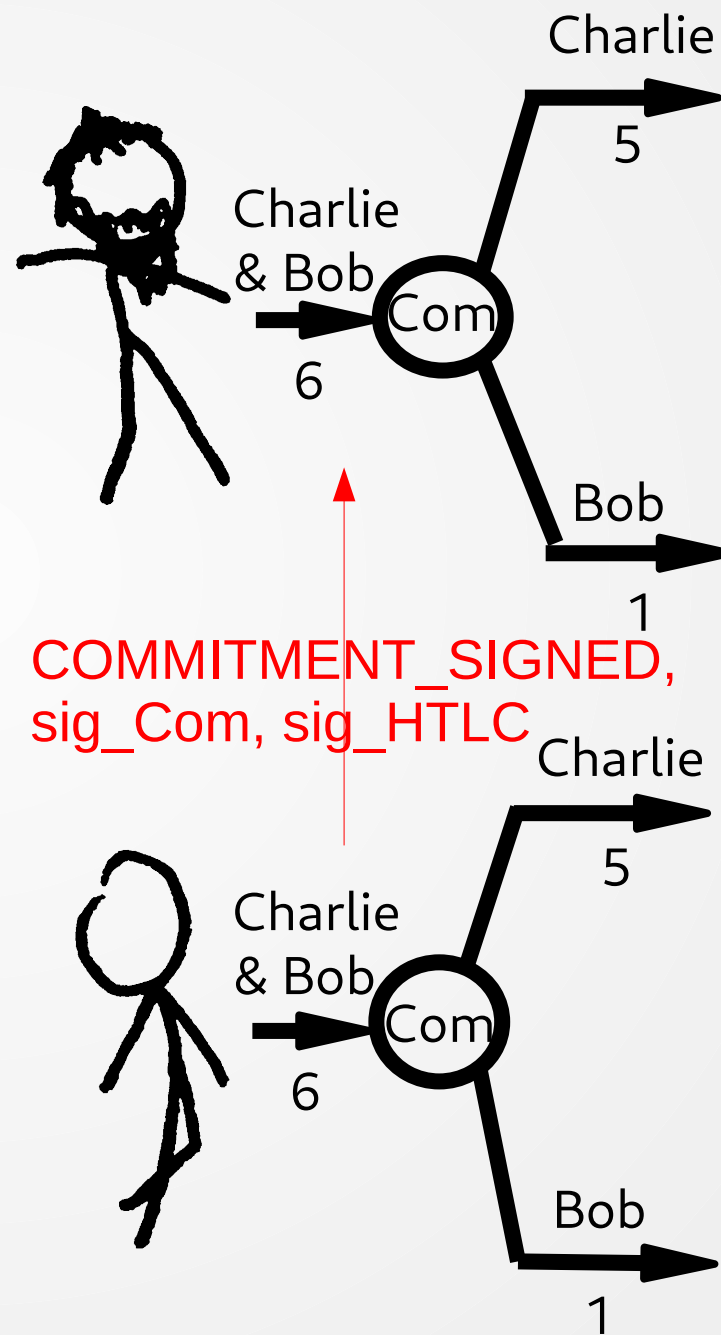
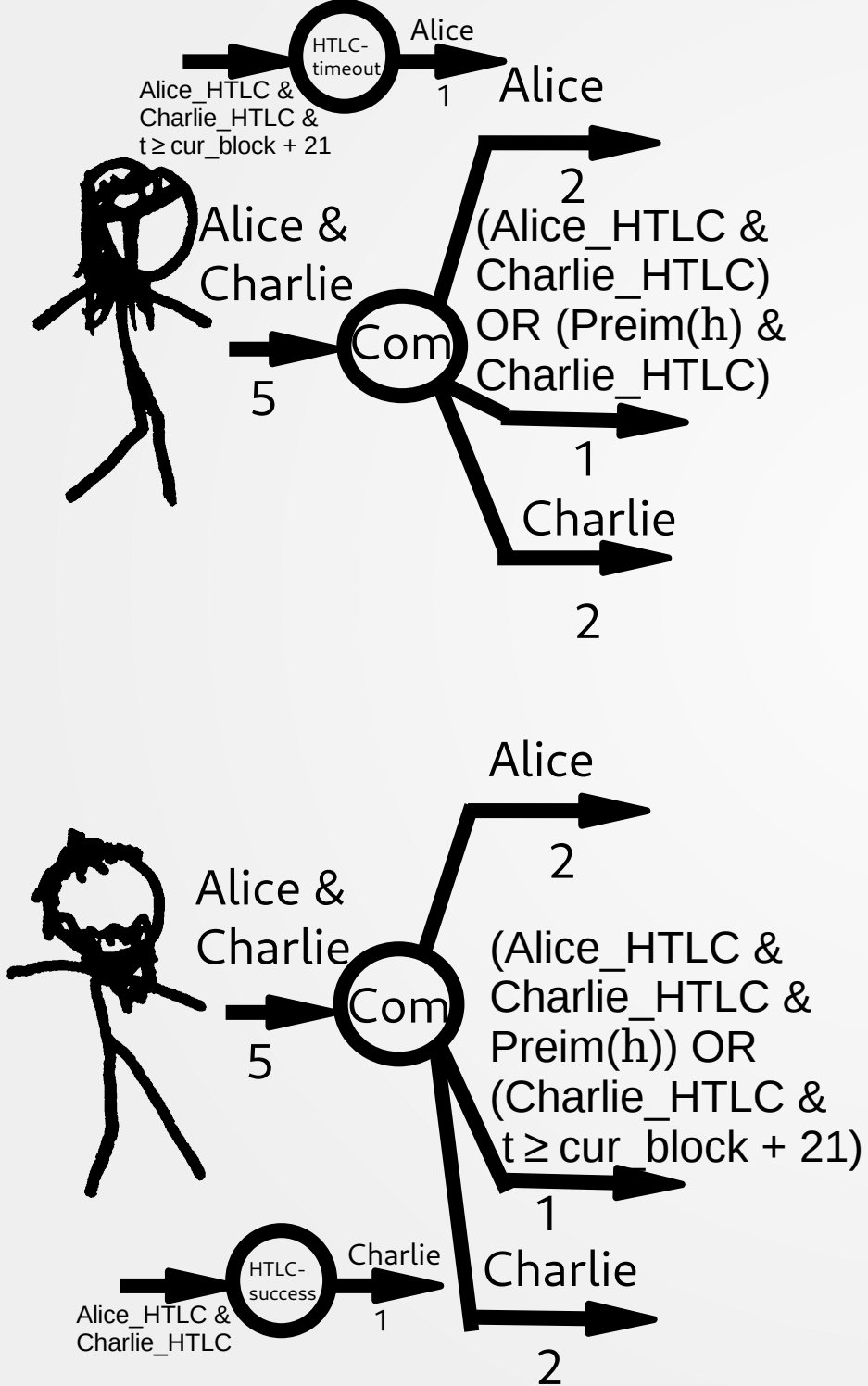
REVOKE\_AND\_ACK,  
sk-Charlie, pk-Charlie



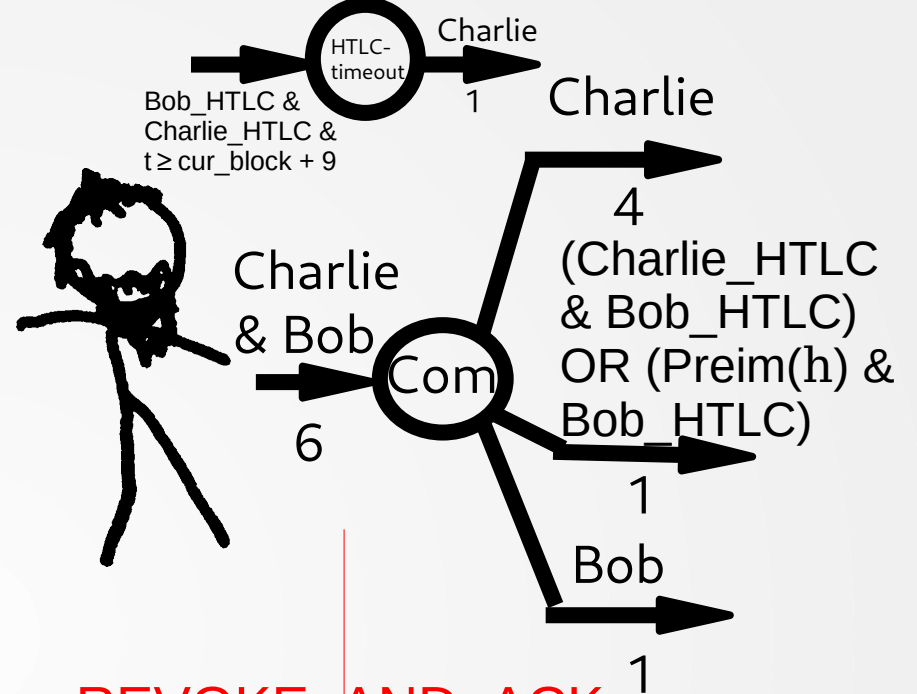
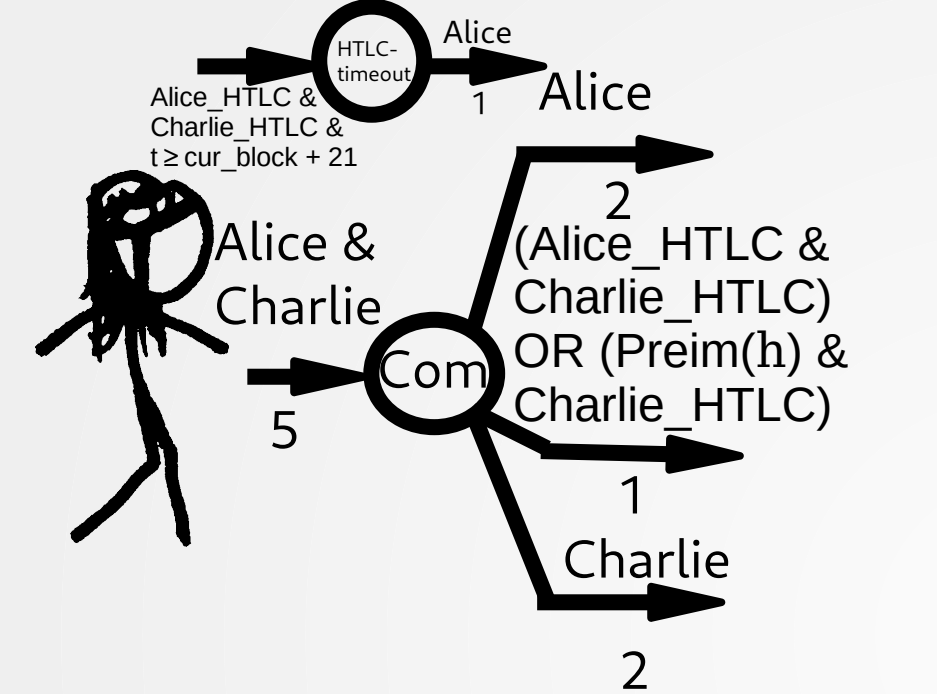


ADD\_HTLC, h, amt=1, del=9

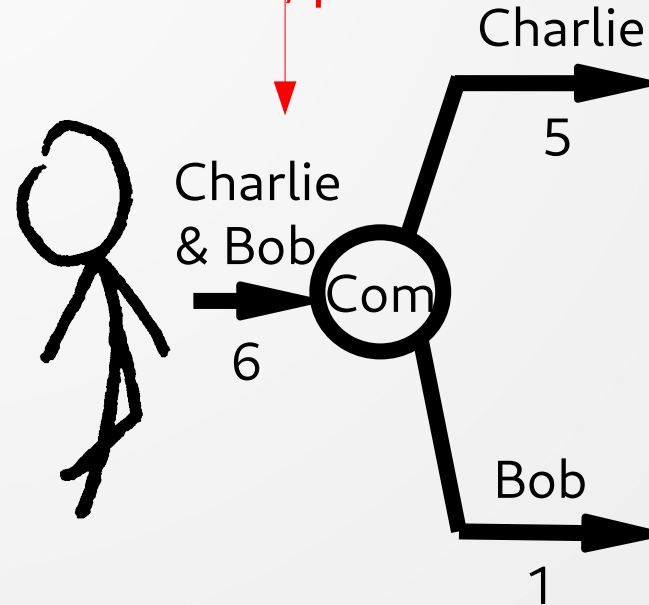
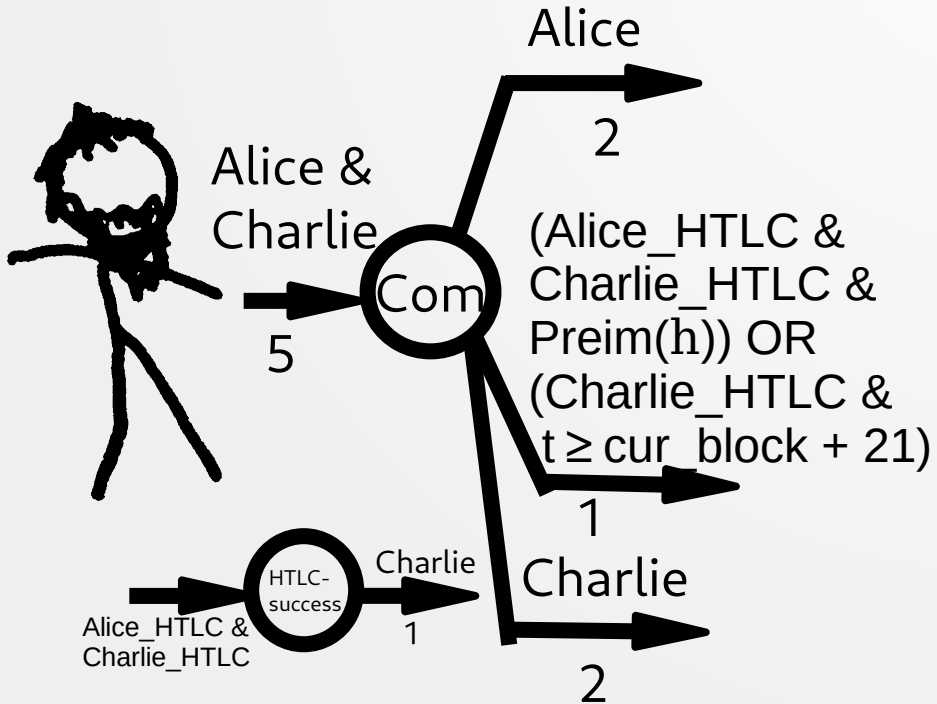


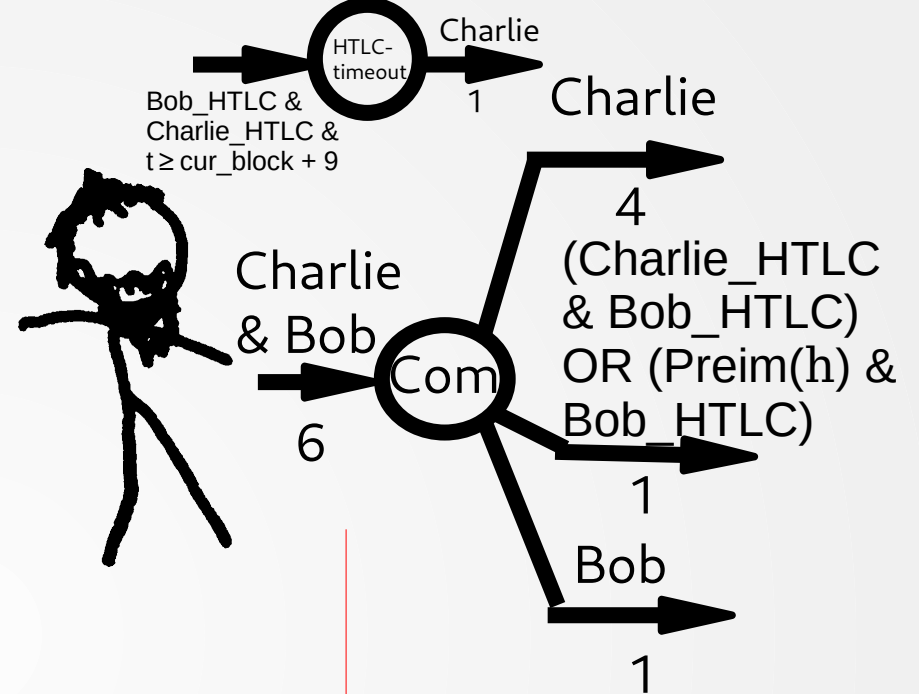
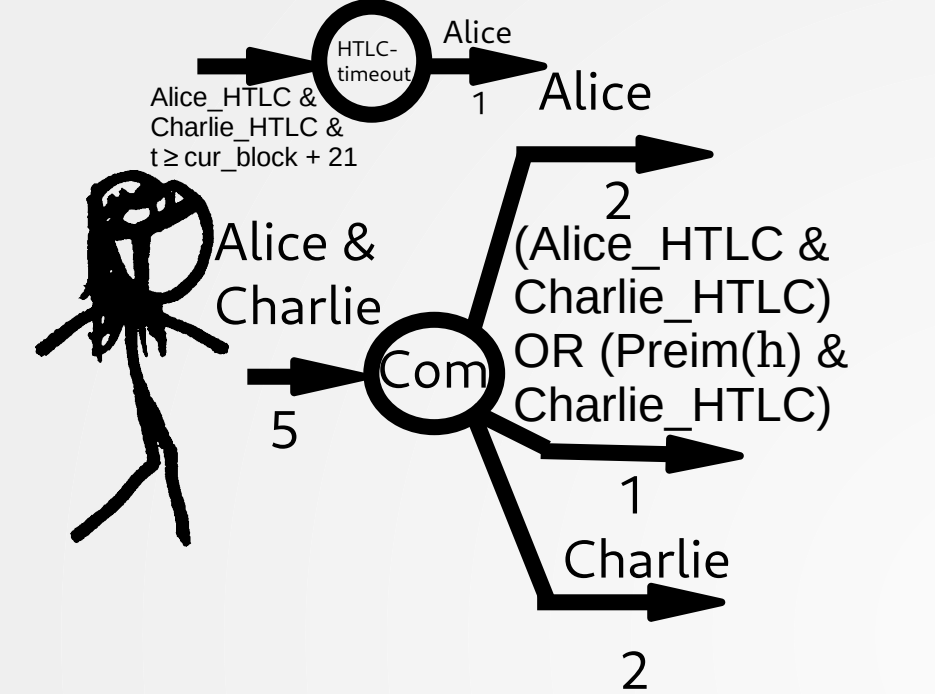




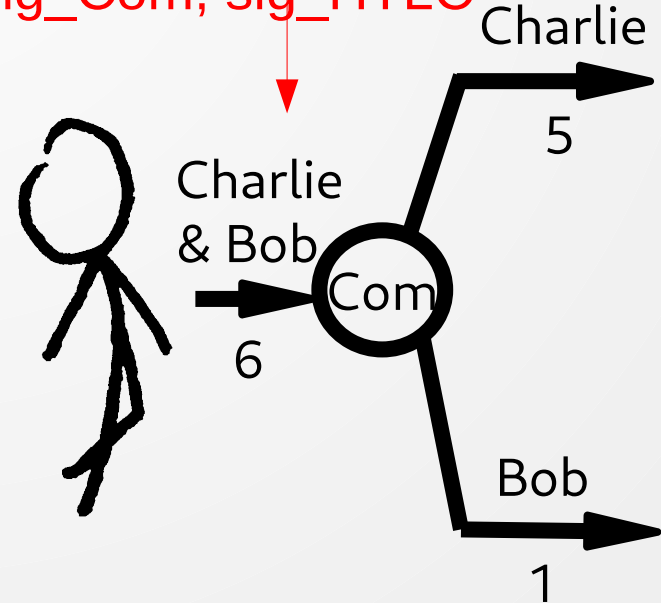
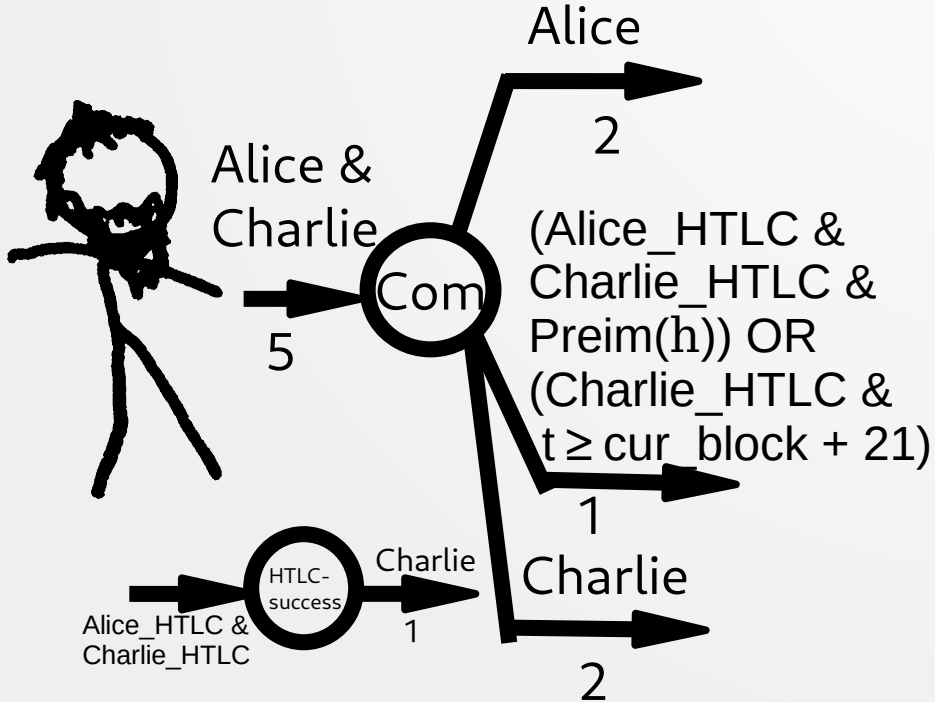


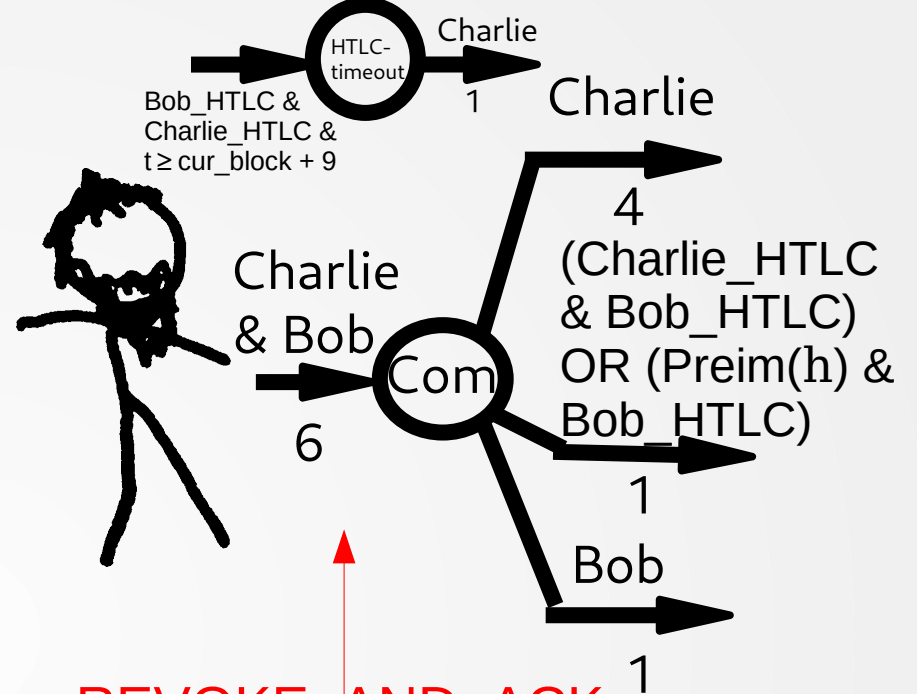
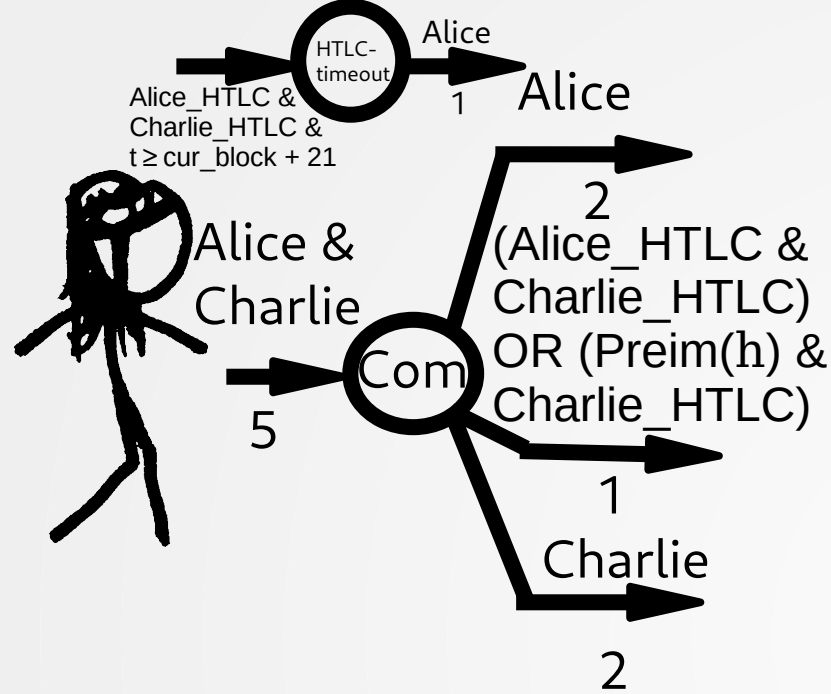
REVOKE\_AND\_ACK,  
sk-Charlie', pk-Charlie"



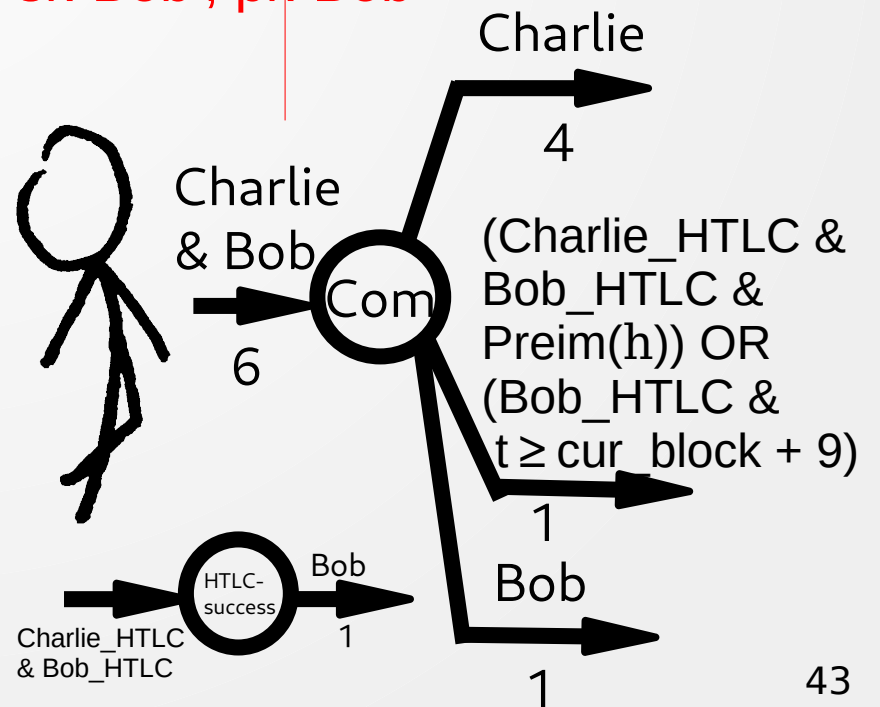
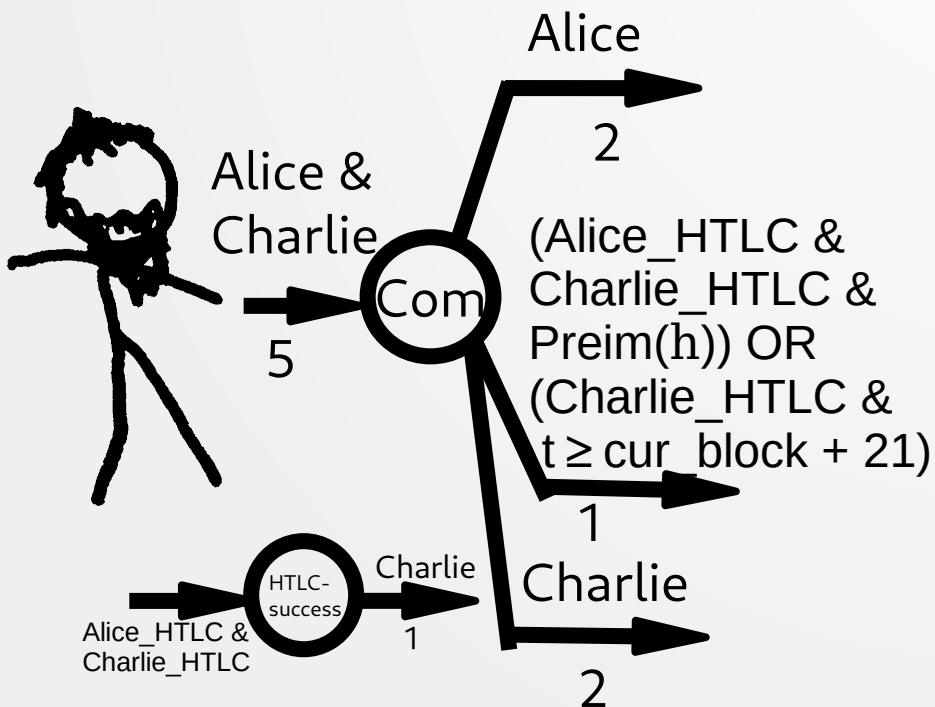


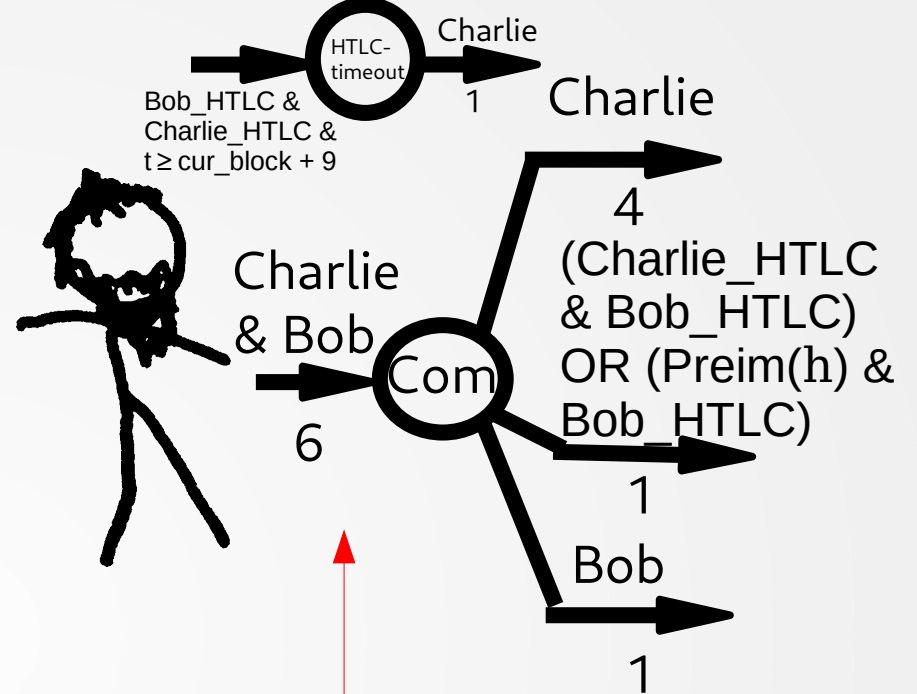
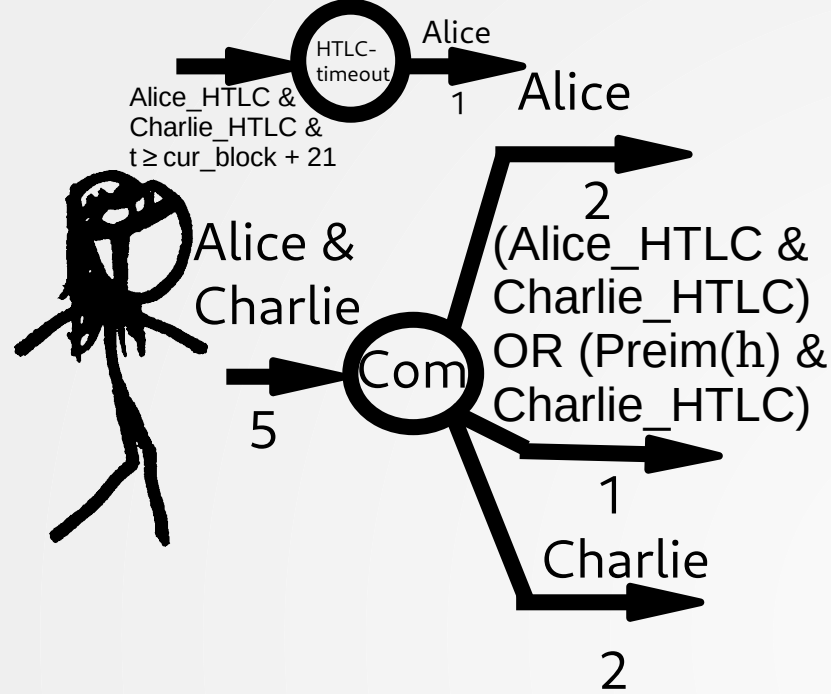
COMMITMENT\_SIGNED,  
sig\_Com, sig\_HTLC



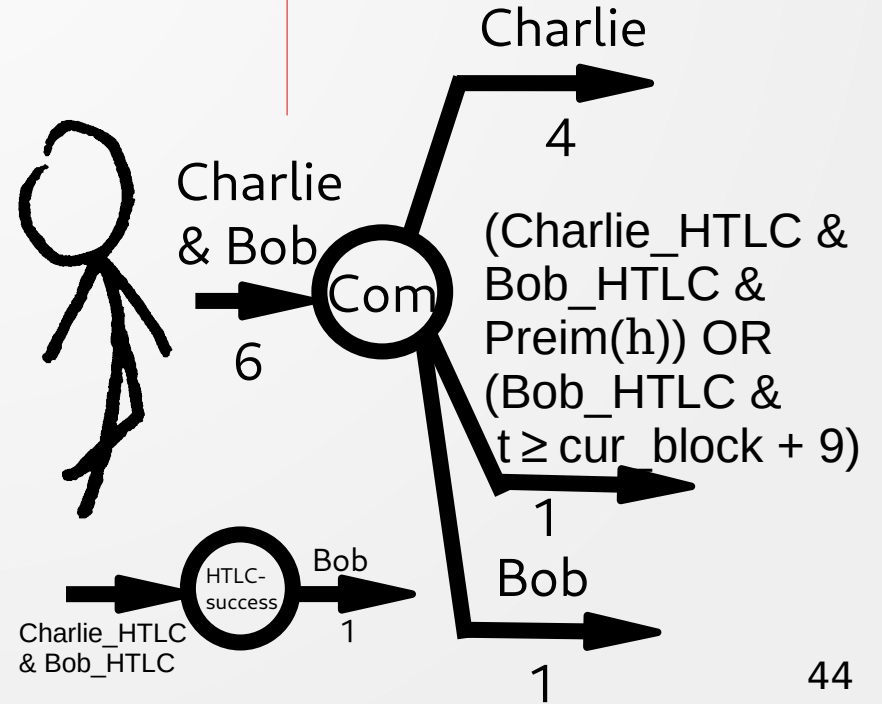
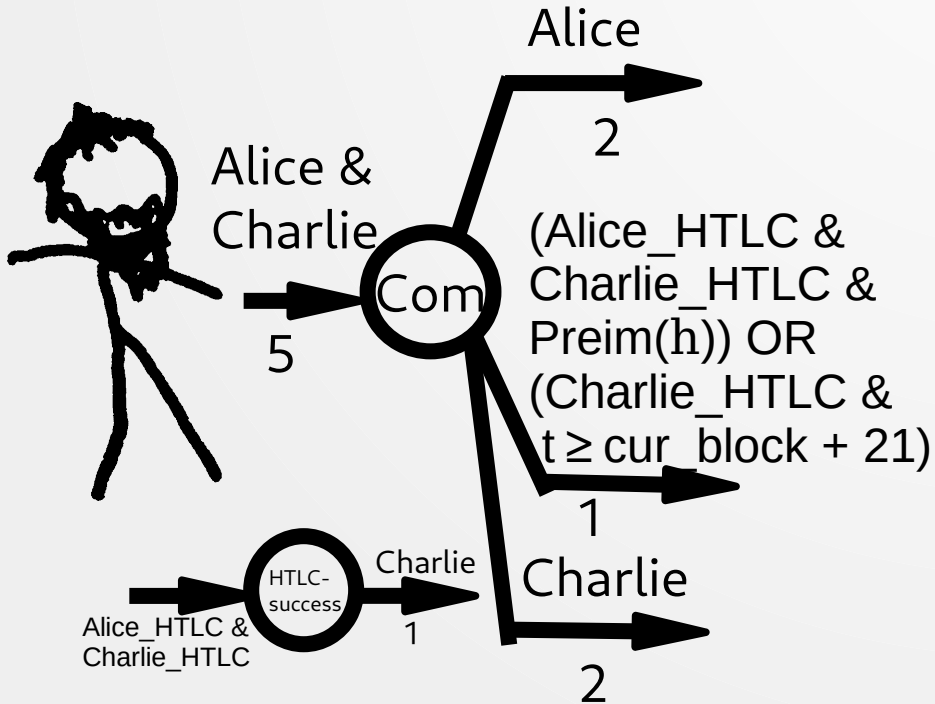


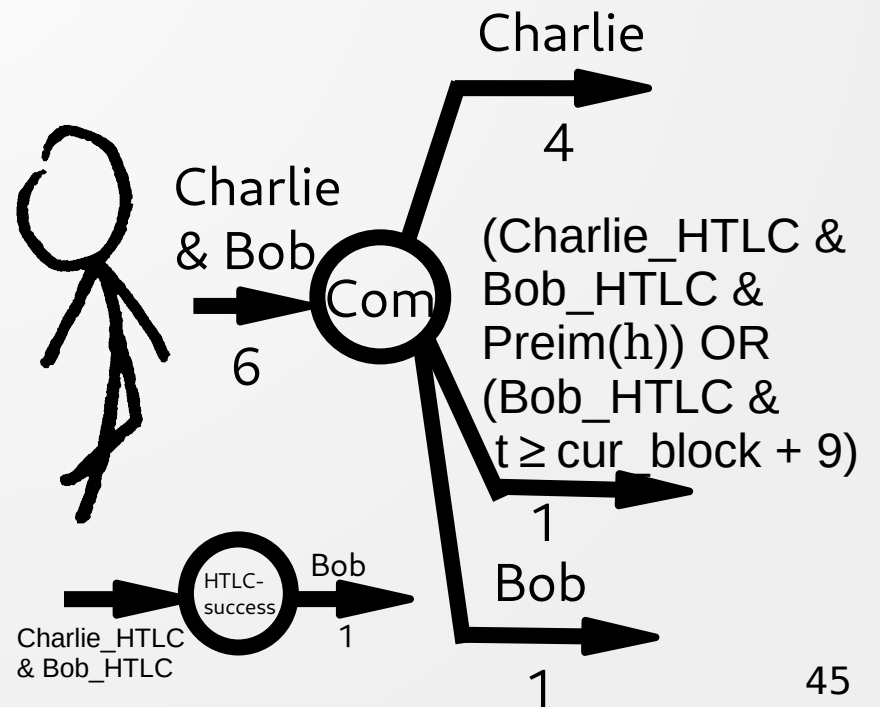
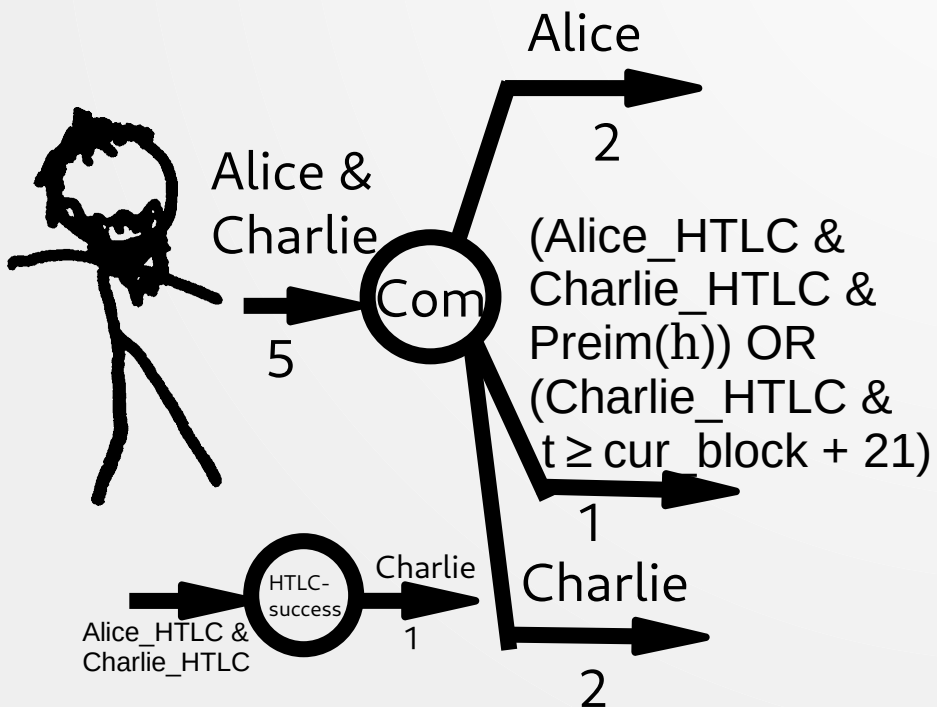
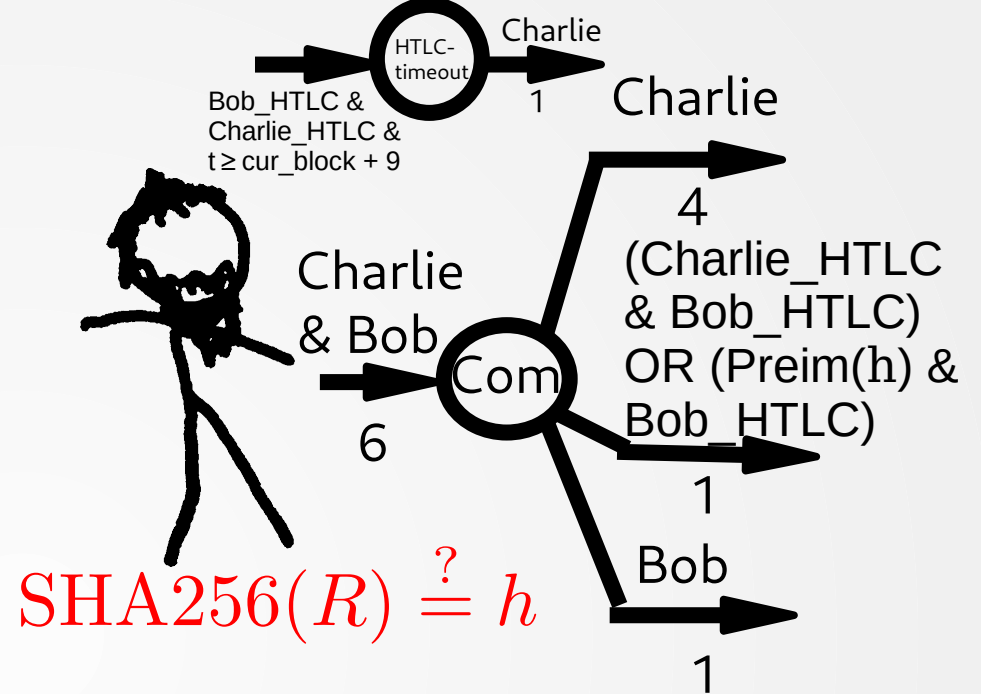
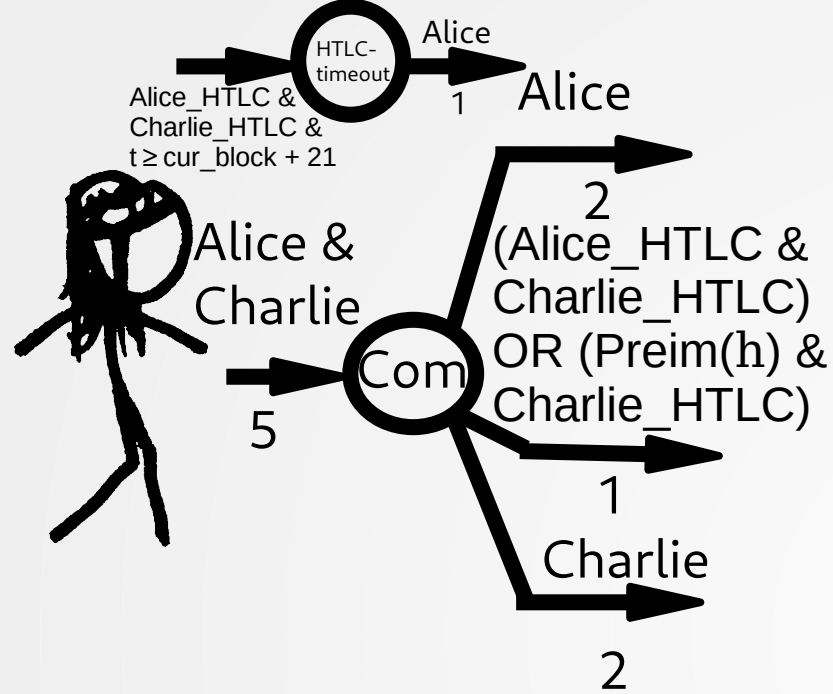
REVOKE\_AND\_ACK,  
sk-Bob', pk-Bob''





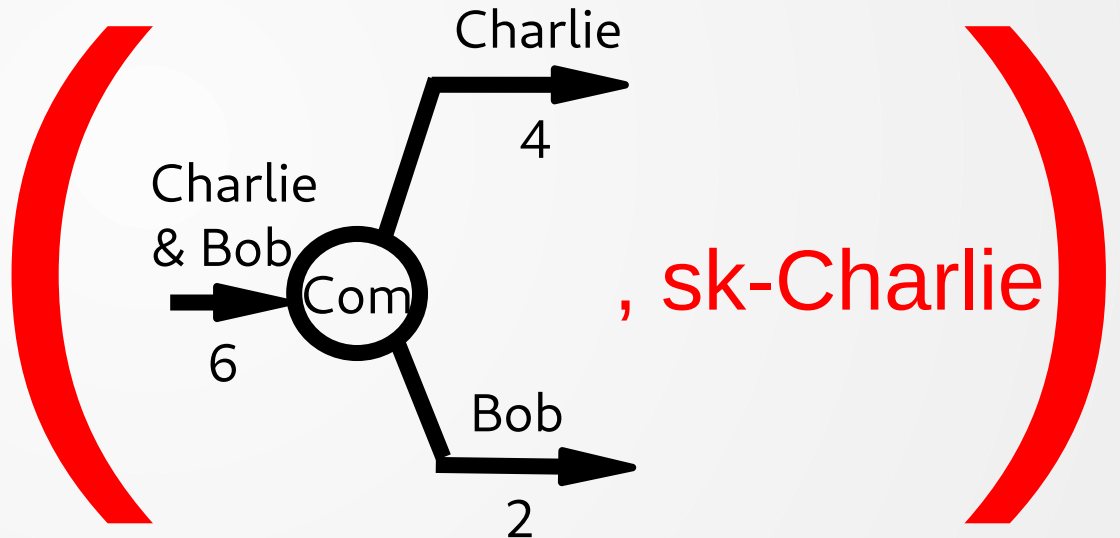
FULFILL\_HTLC, R

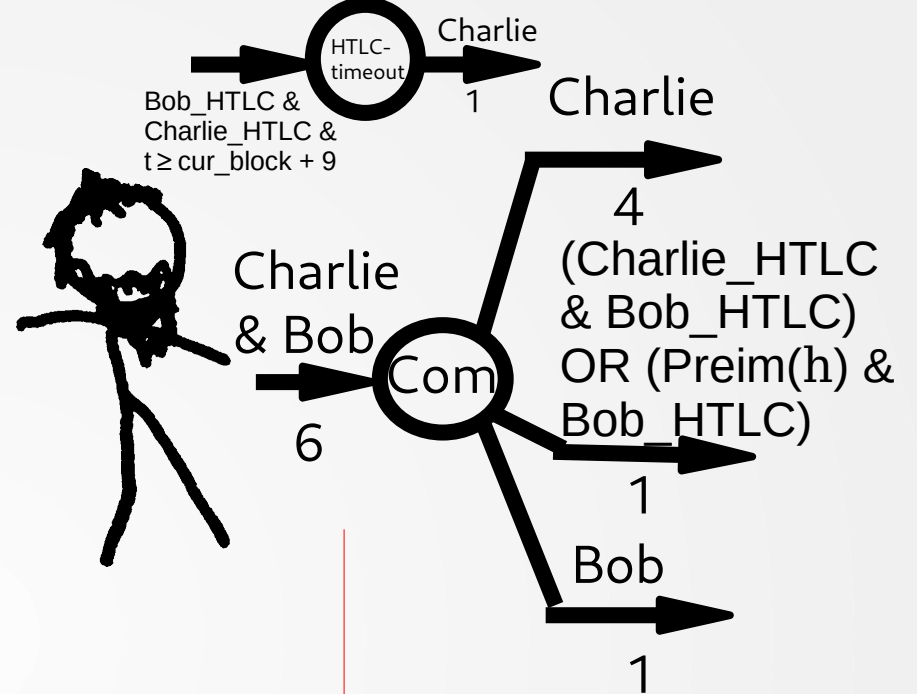
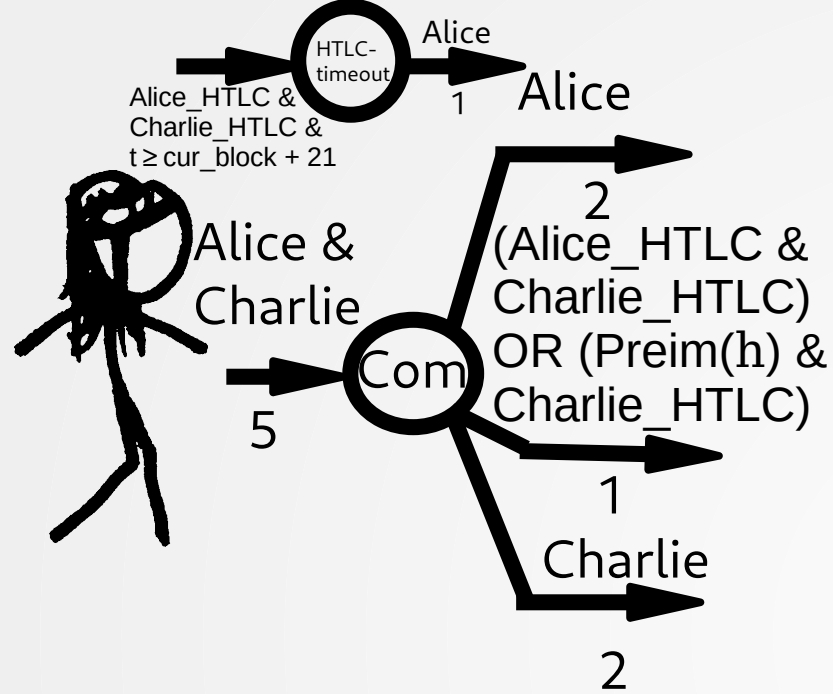




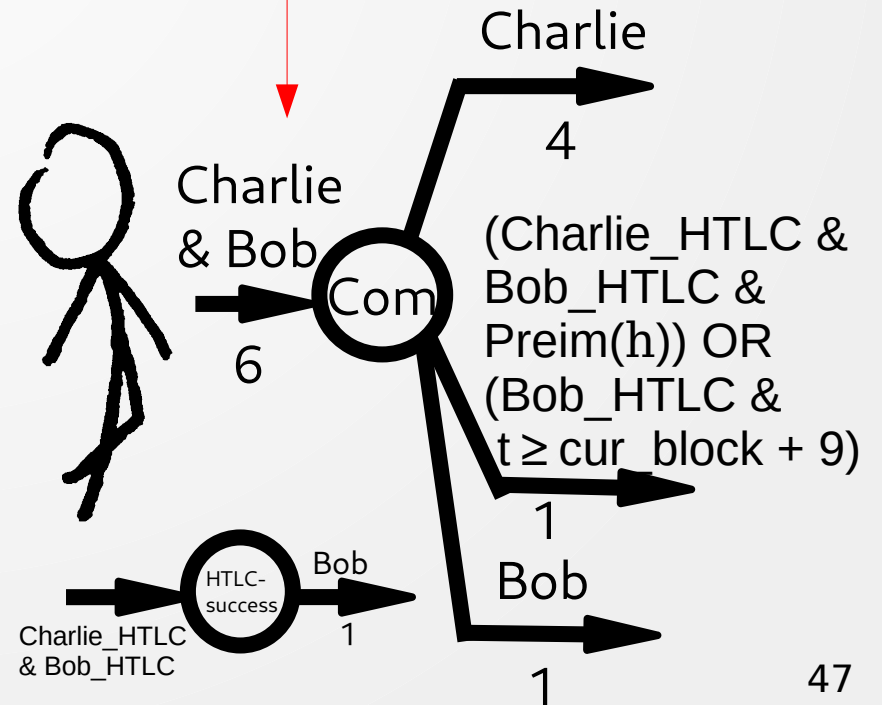
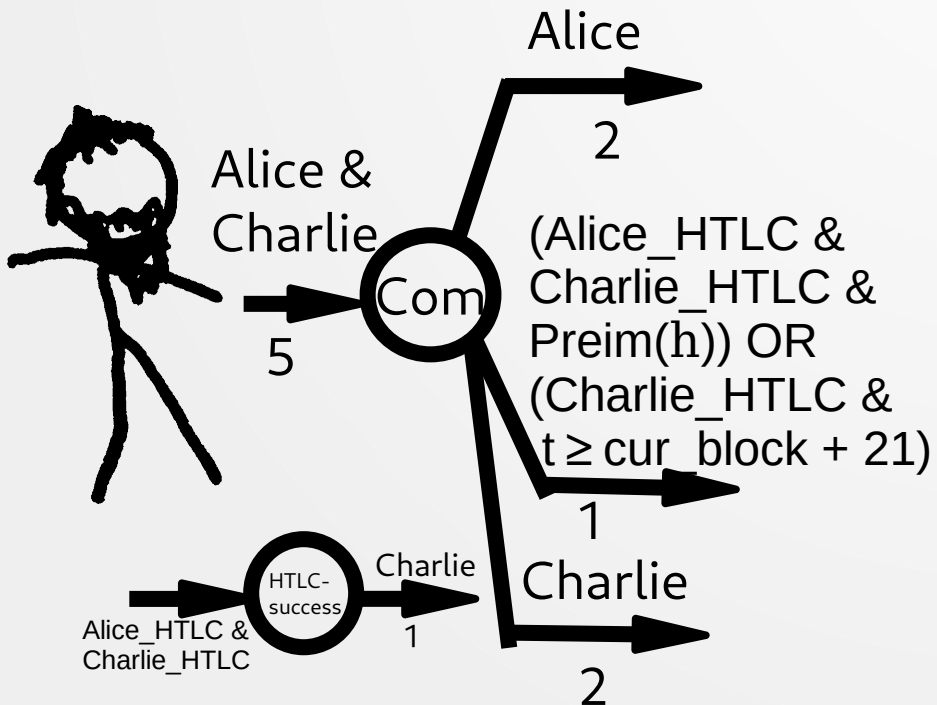


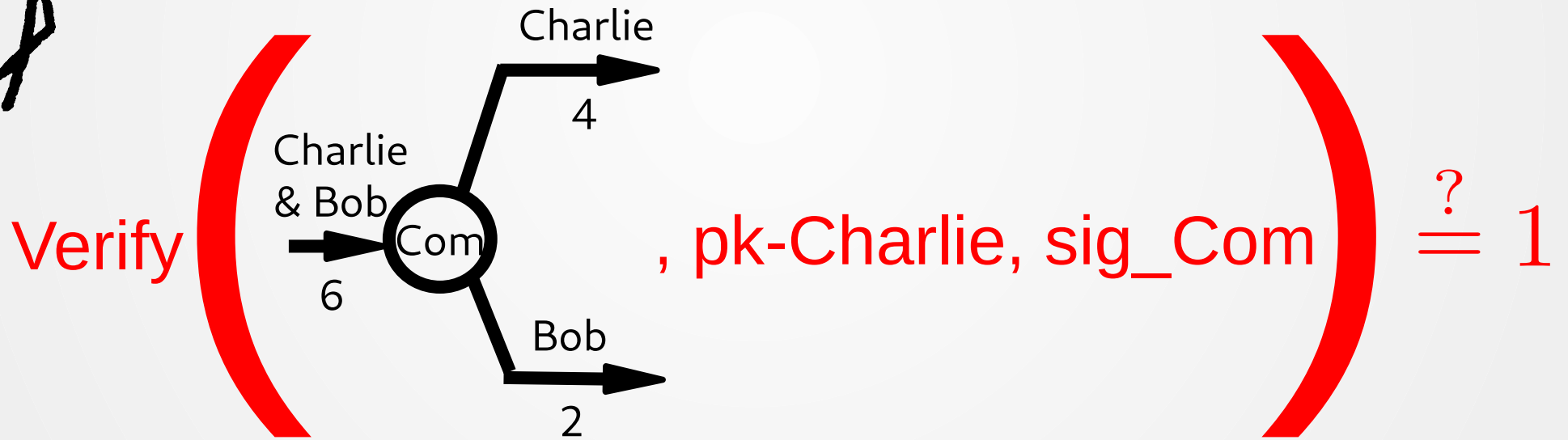
$\text{sig\_Com} \leftarrow \text{Sign}$



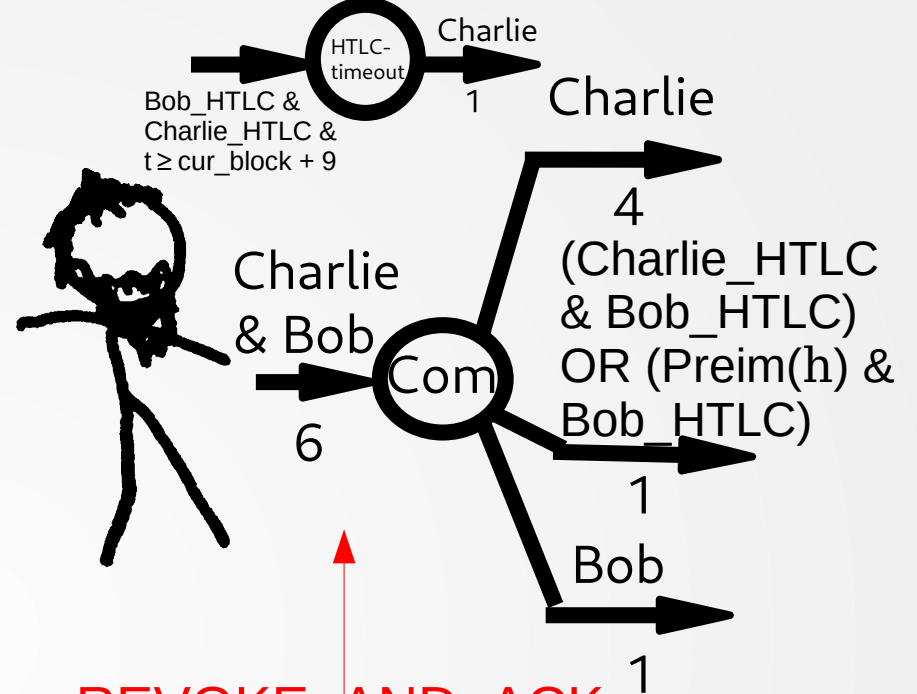
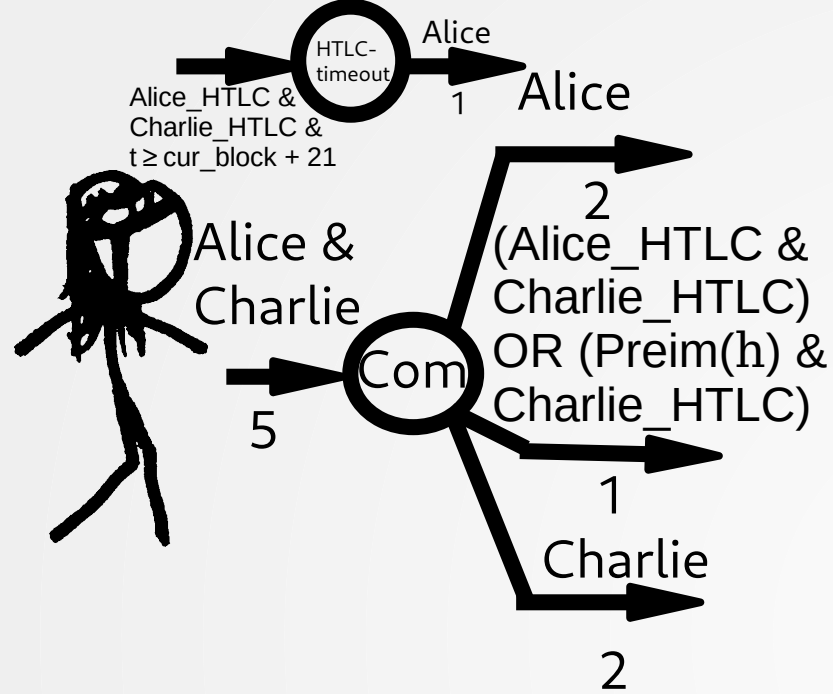


COMMITMENT\_SIGNED, sig\_Com

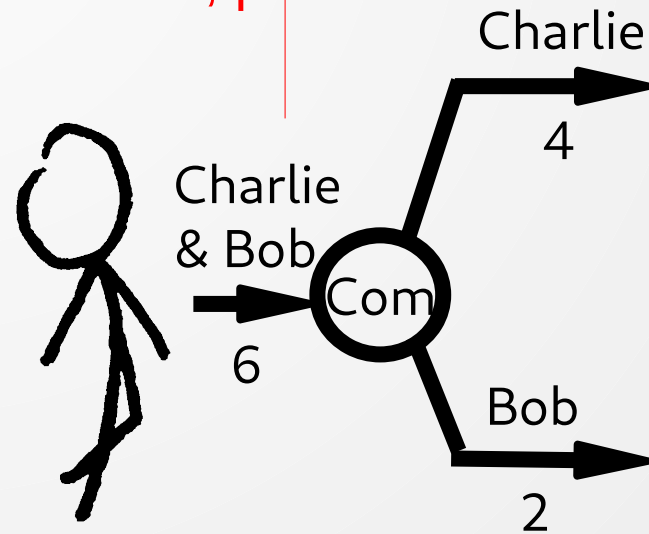
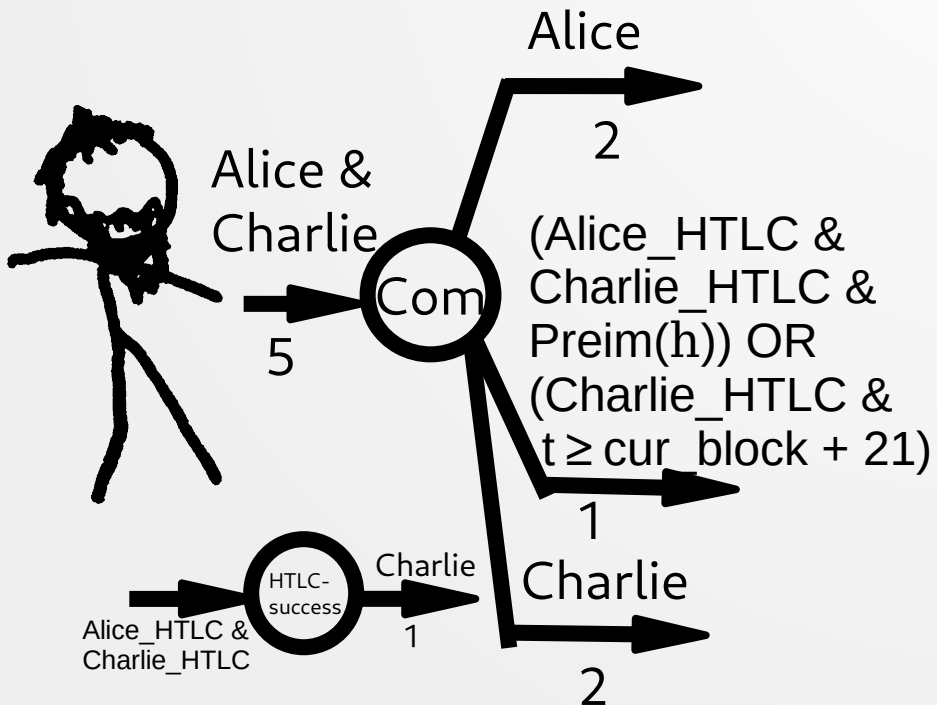


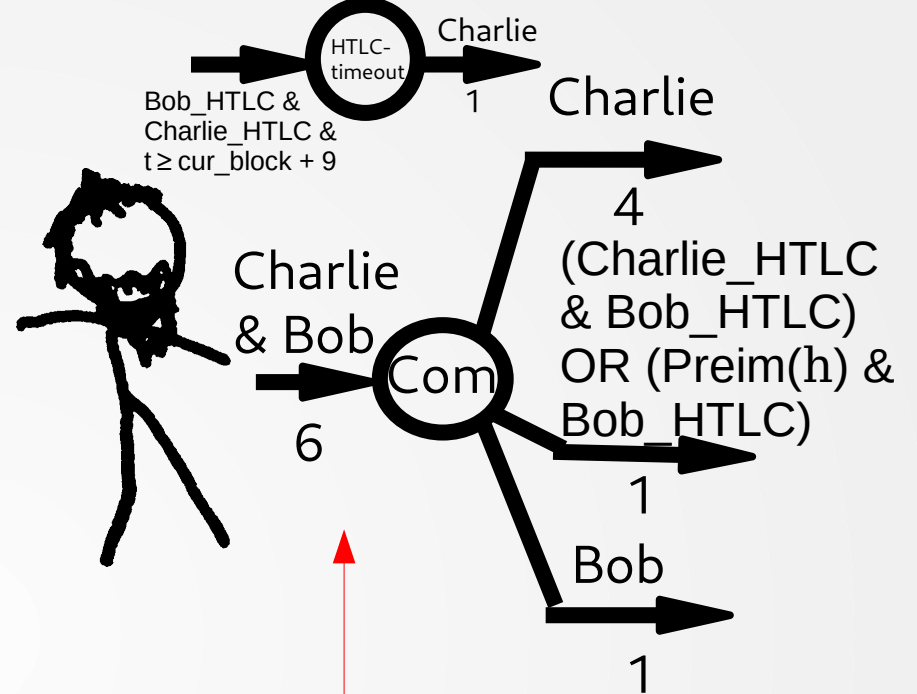
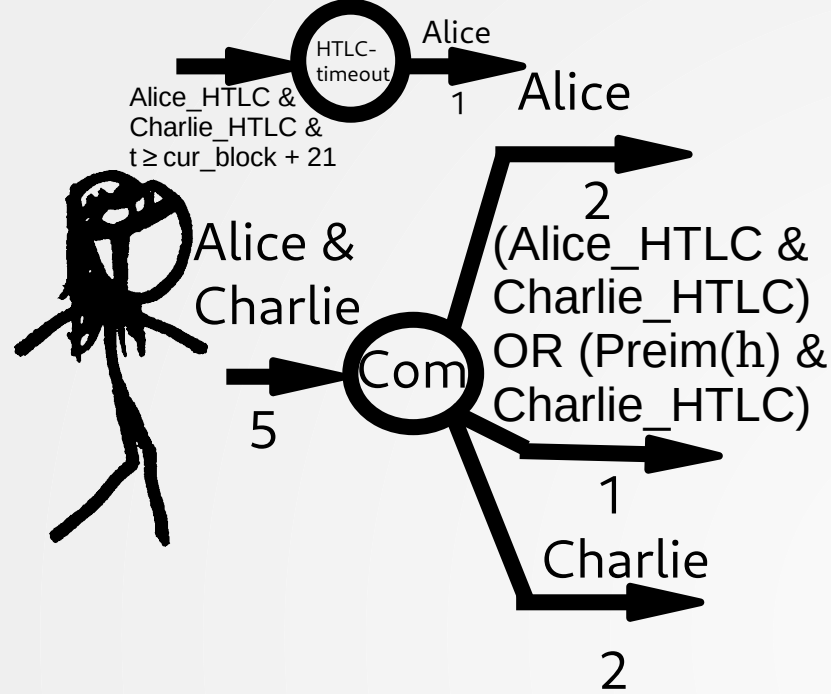




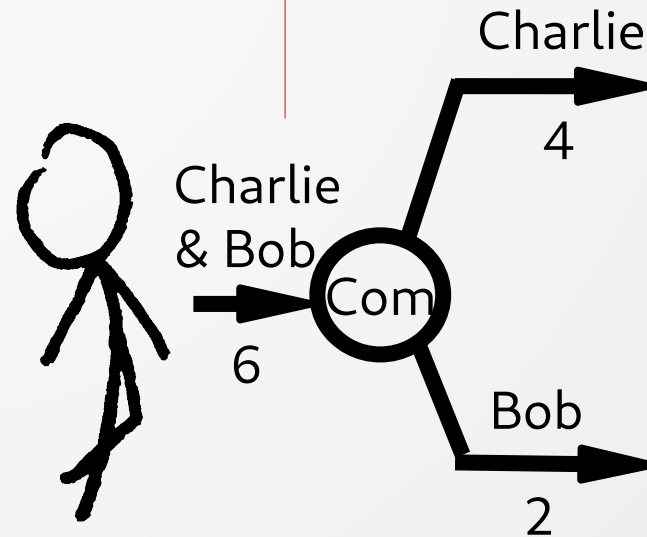
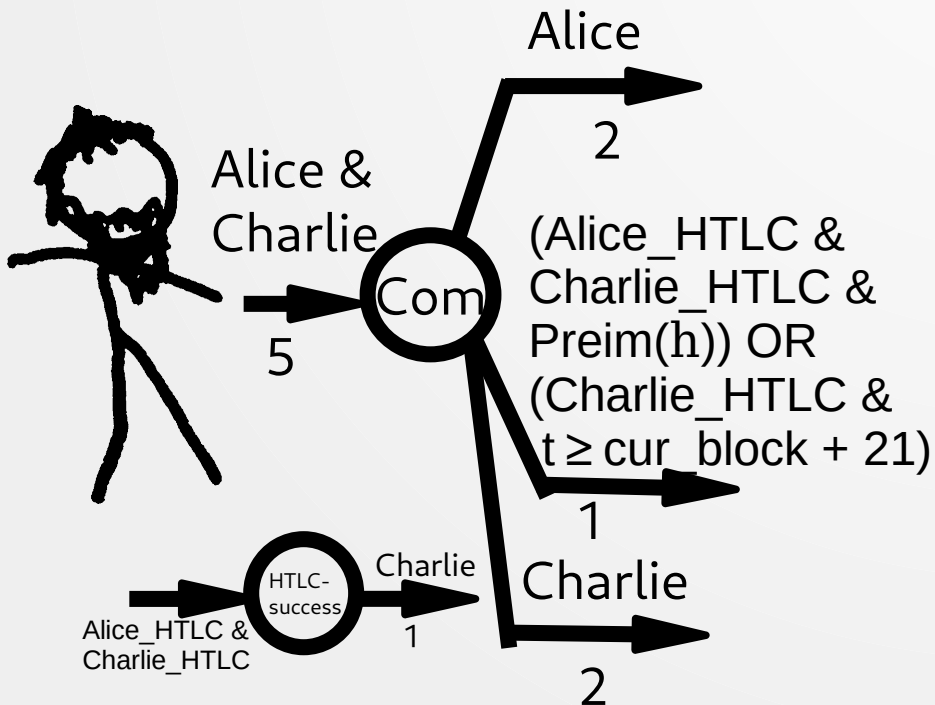


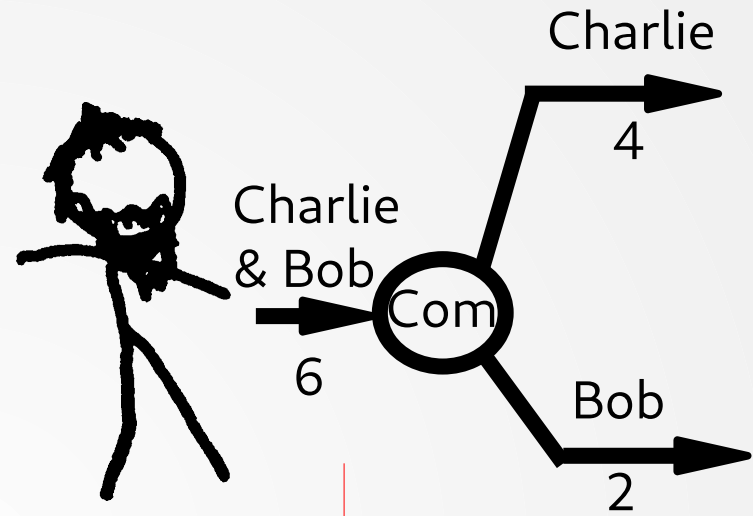
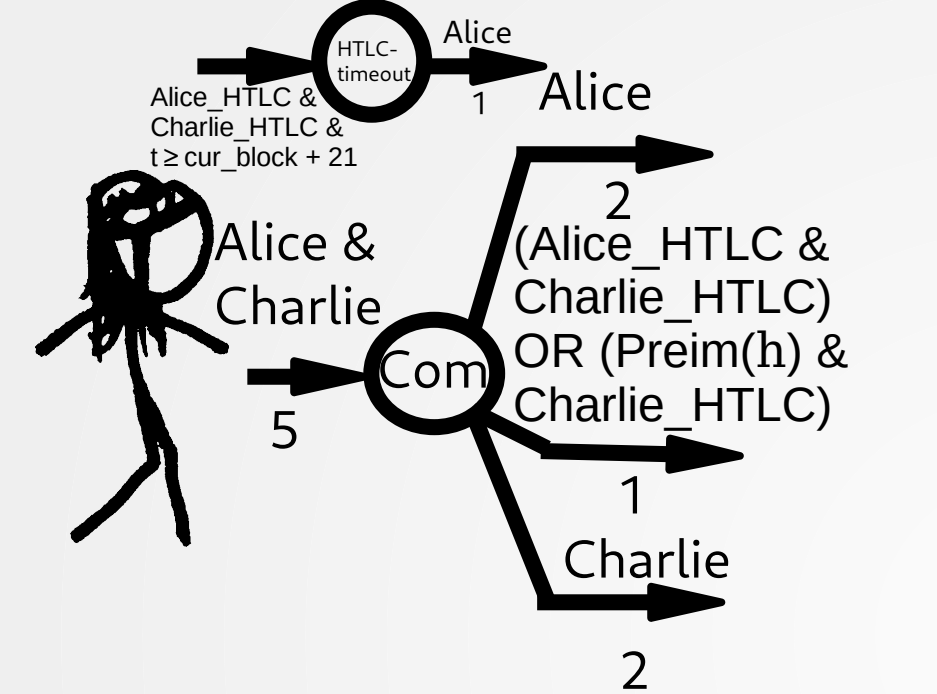
REVOKE\_AND\_ACK,  
sk-Bob', pk-Bob''



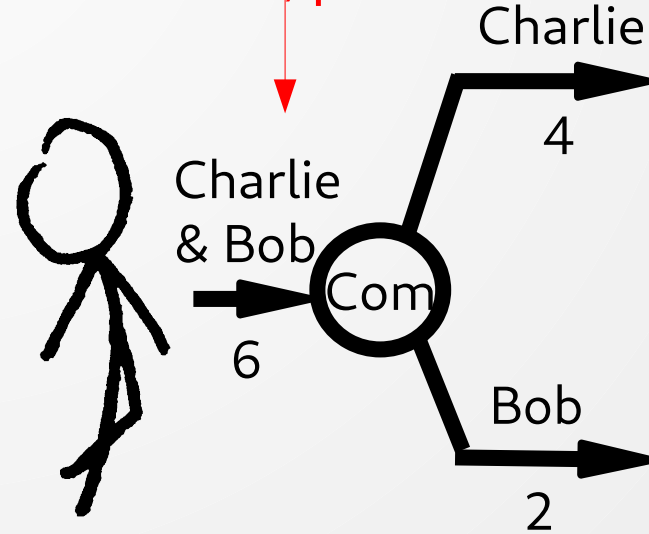
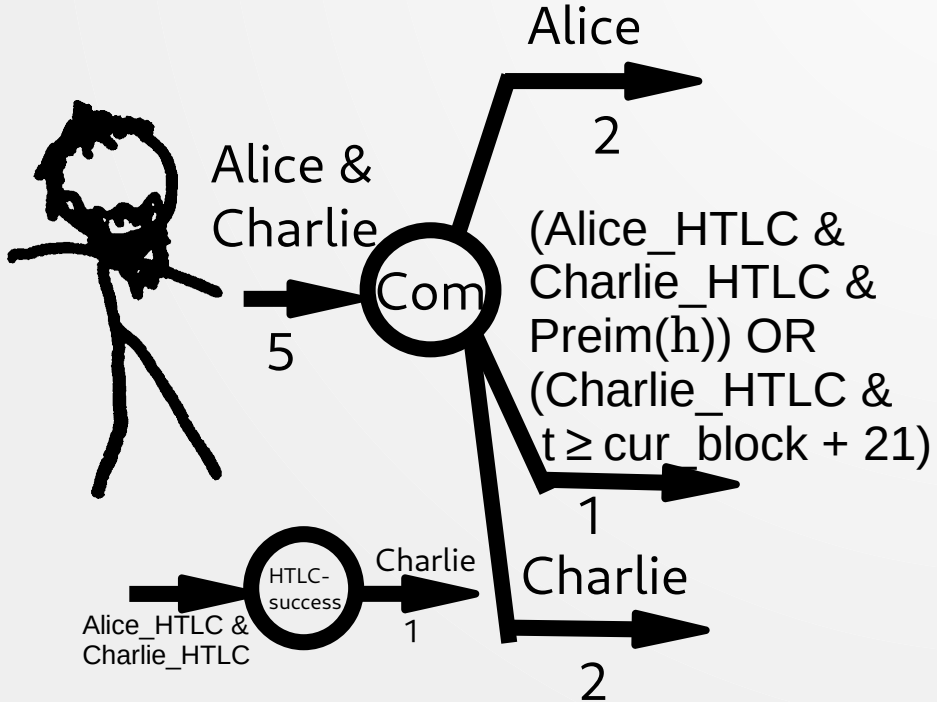


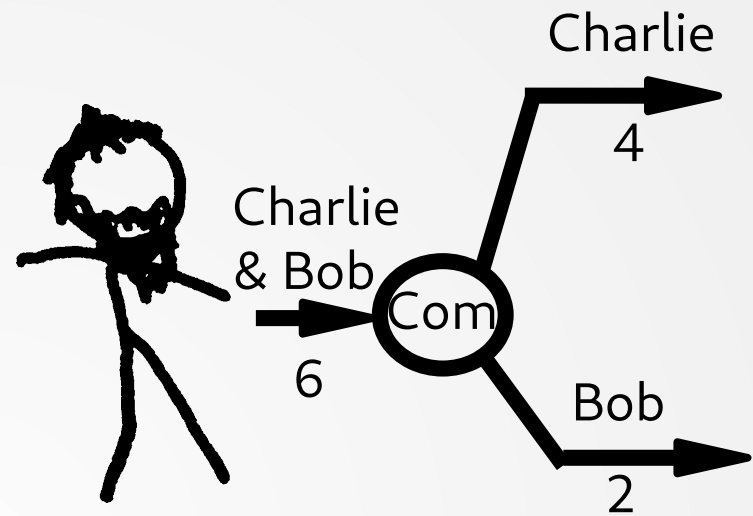
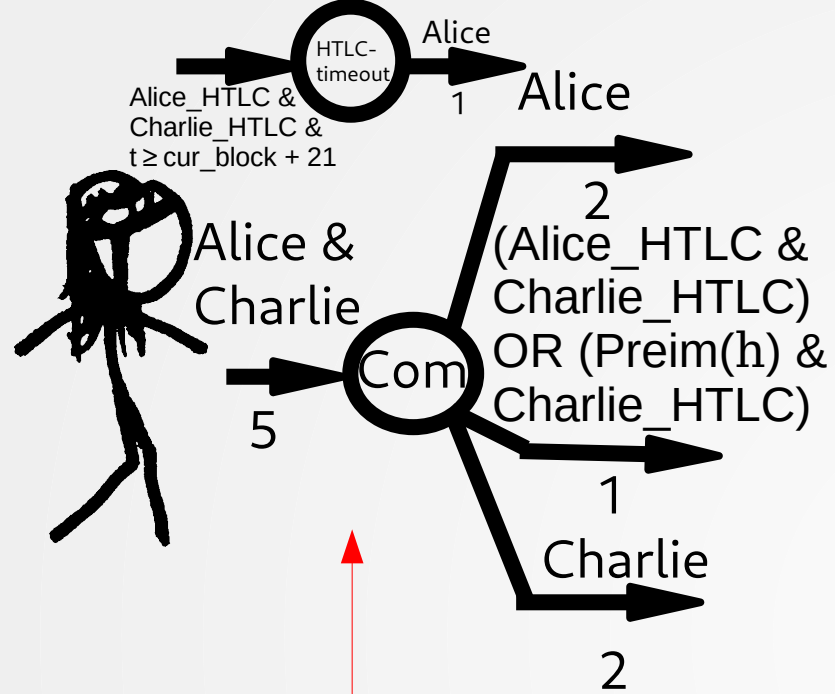
COMMITMENT\_SIGNED, sig\_Com



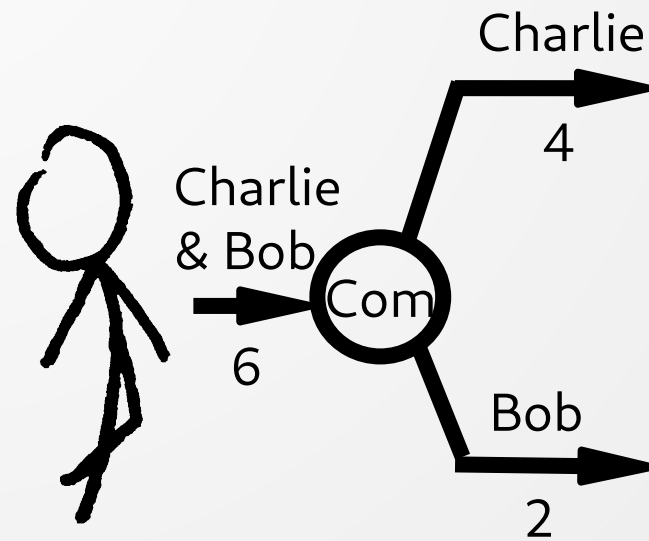
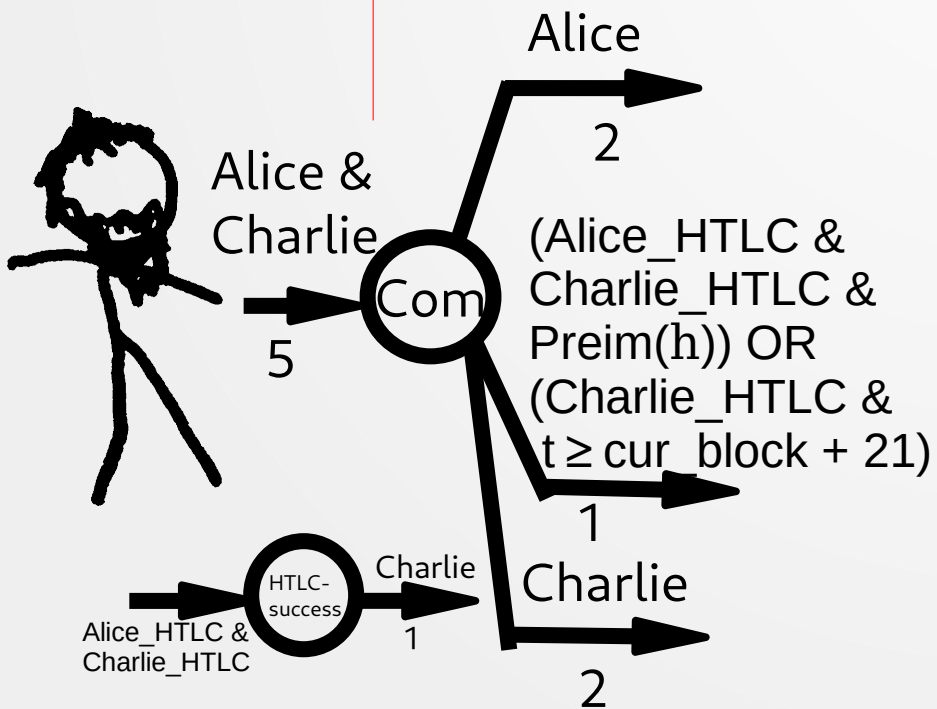


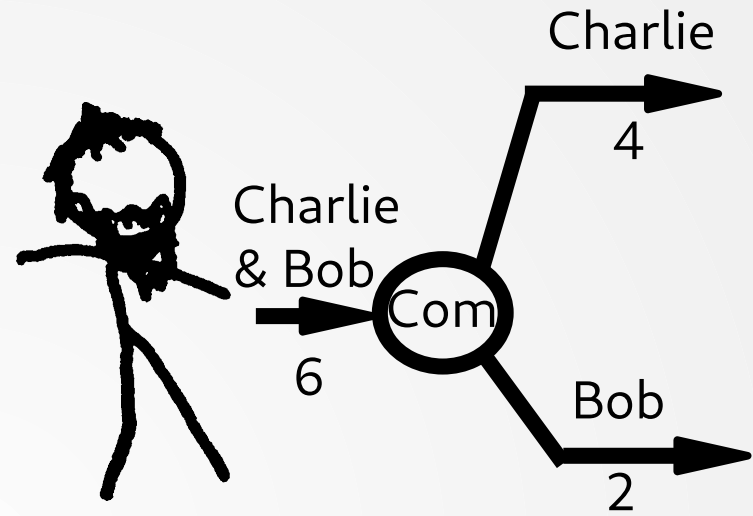
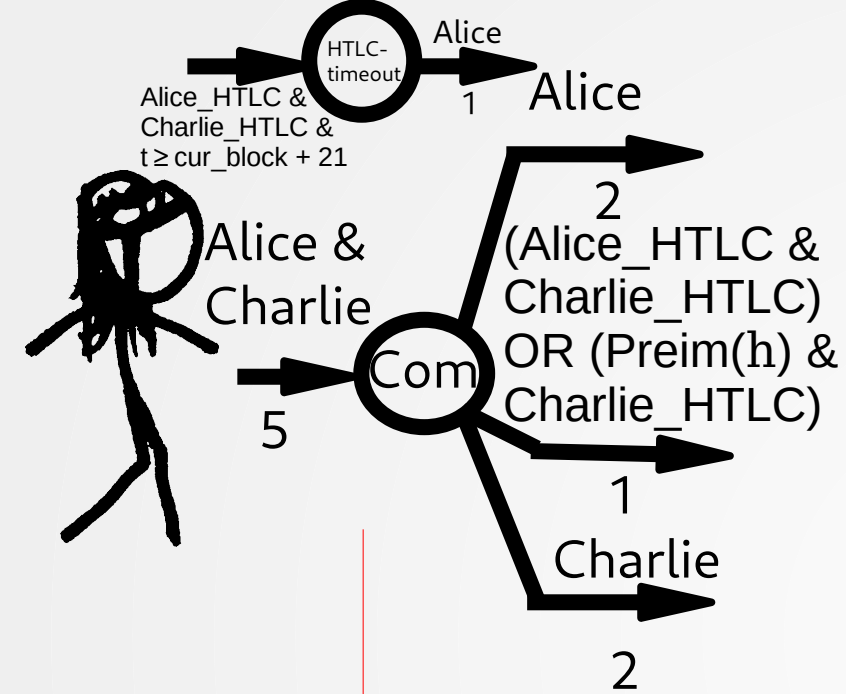
REVOKE\_AND\_ACK,  
sk-Charlie', pk-Charlie"



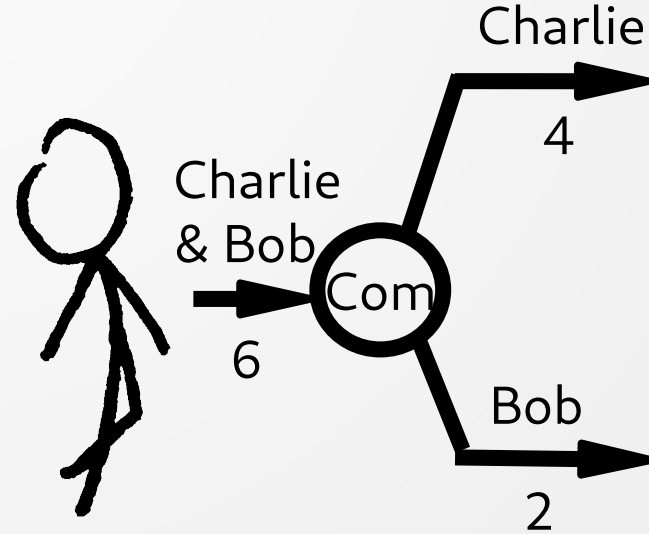
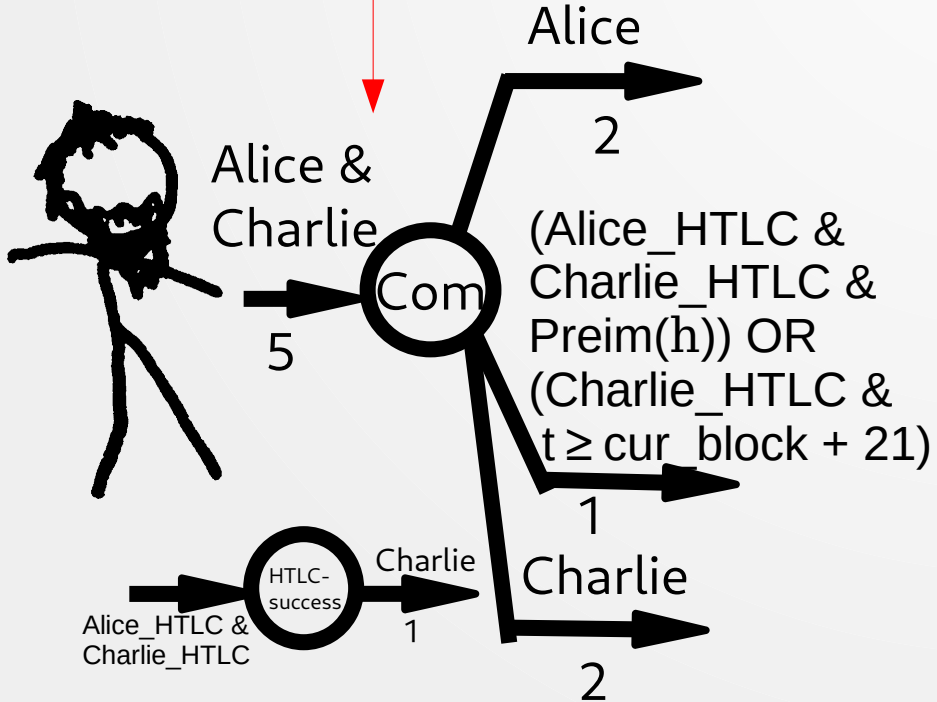


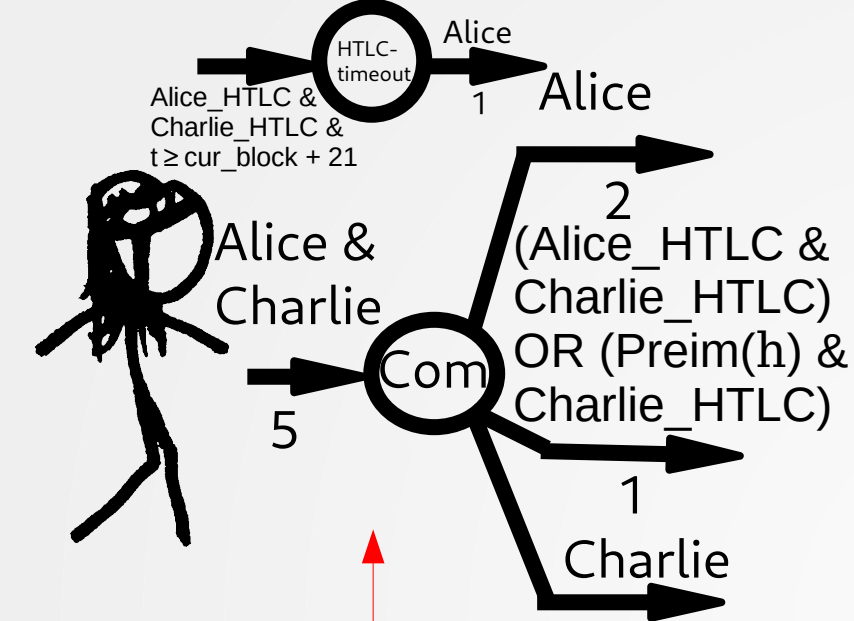
FULFILL\_HTLC, R



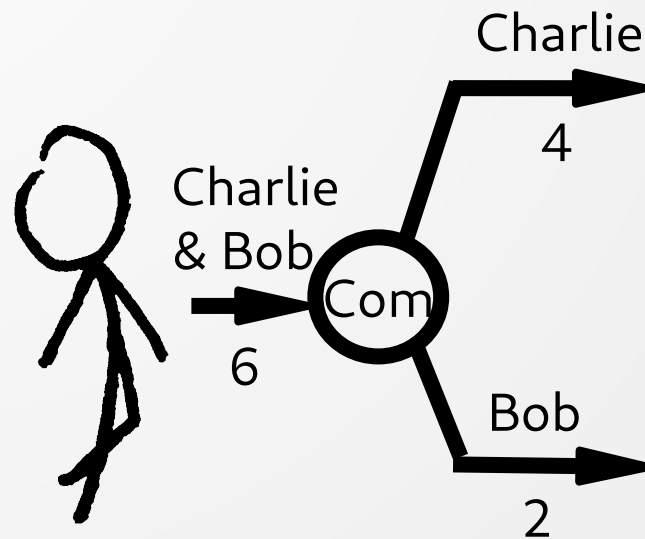
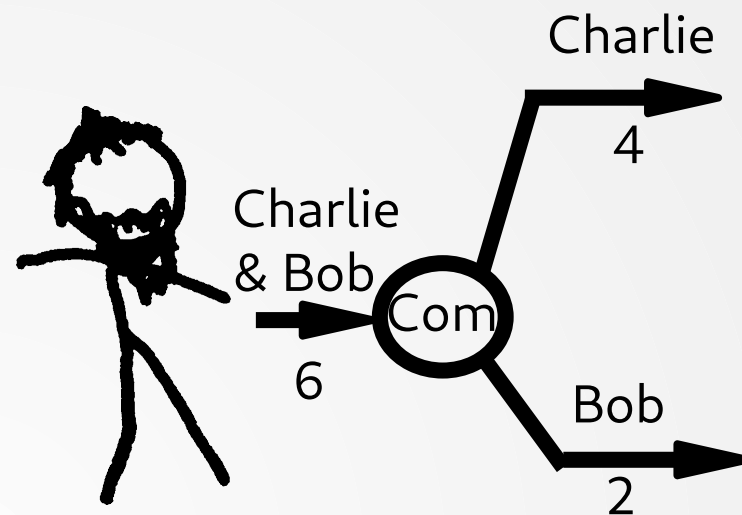
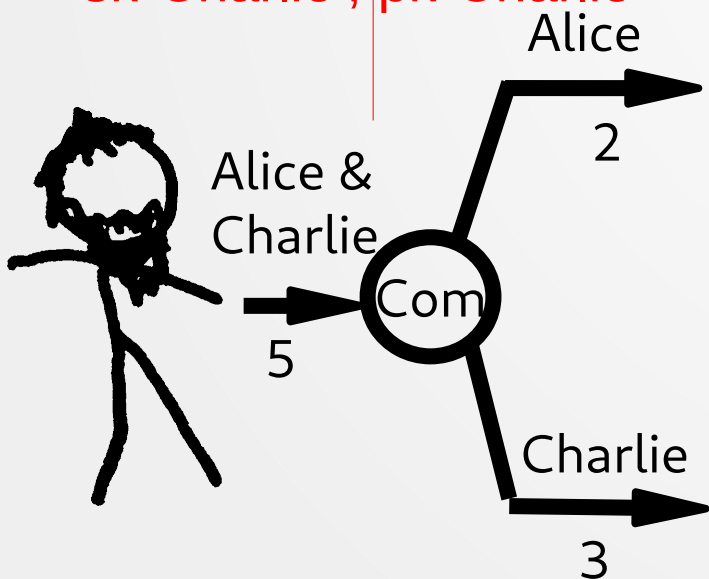


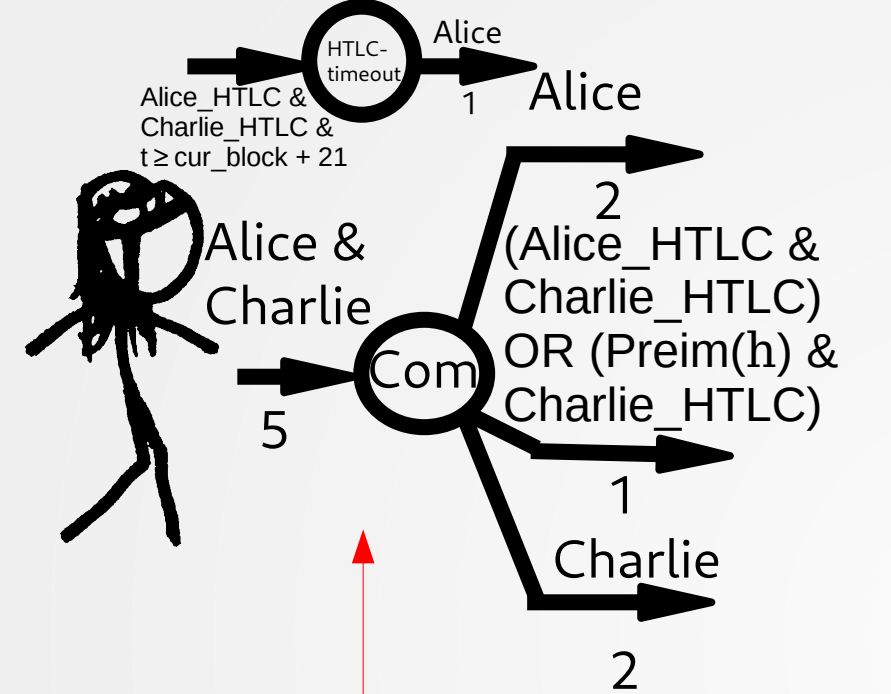
COMMITMENT\_SIGNED, sig\_Com



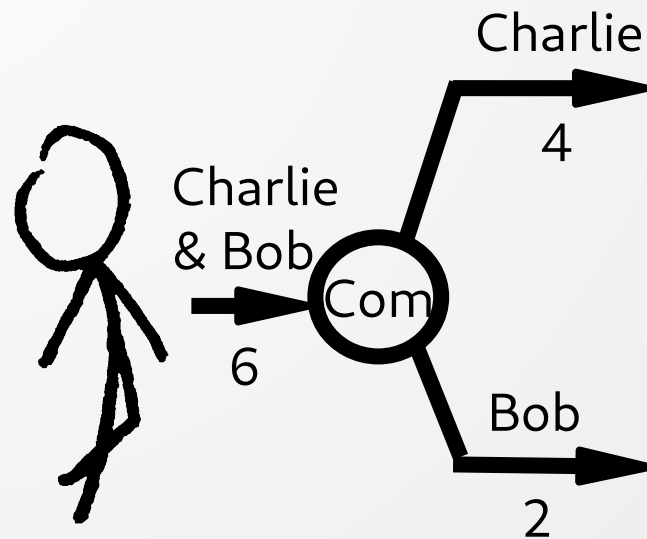
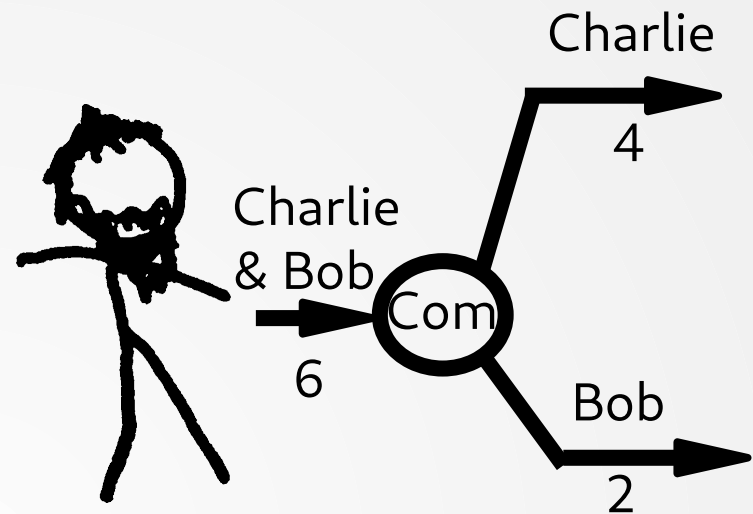
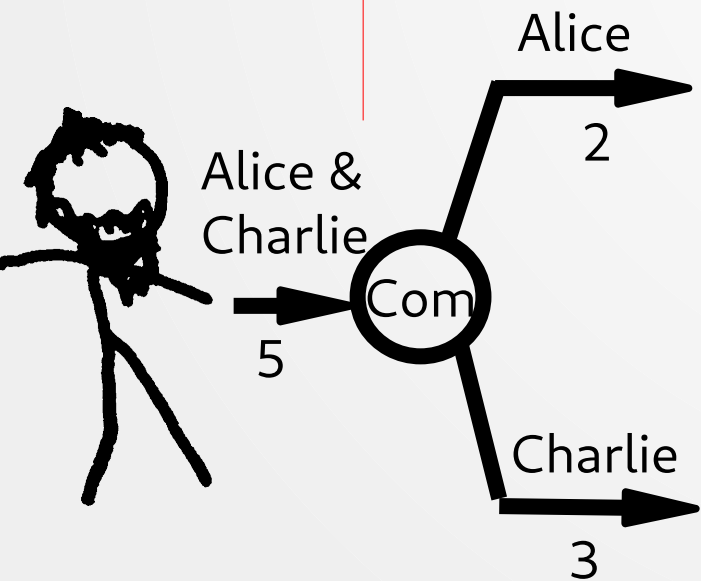


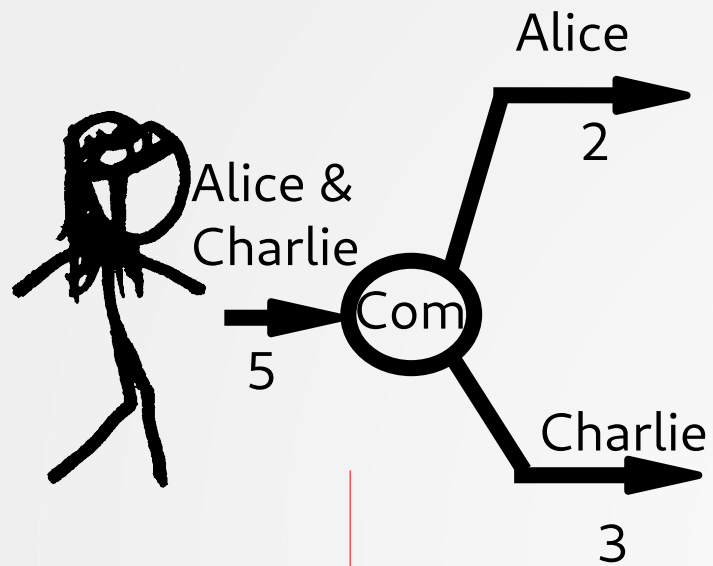
REVOKE\_AND\_ACK,  
sk-Charlie', pk-Charlie"



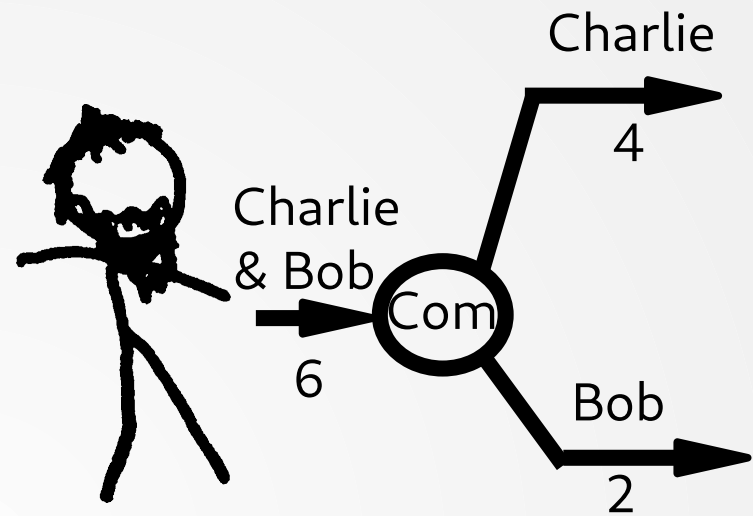
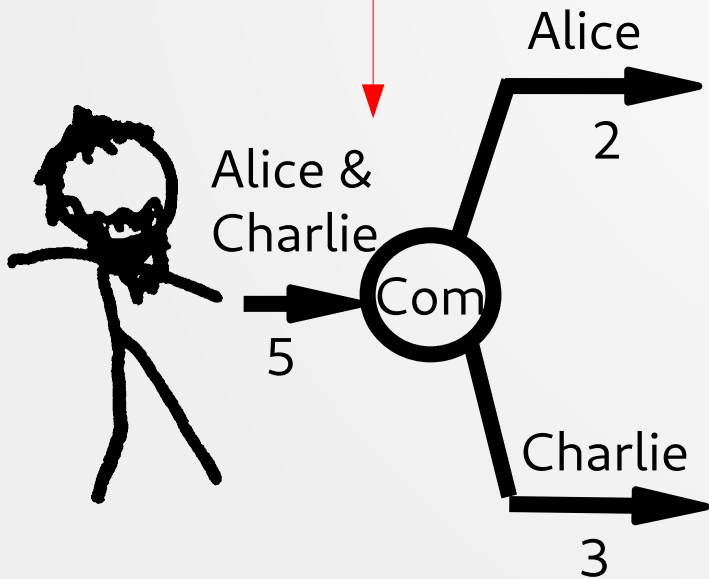


COMMITMENT\_SIGNED, sig\_Com

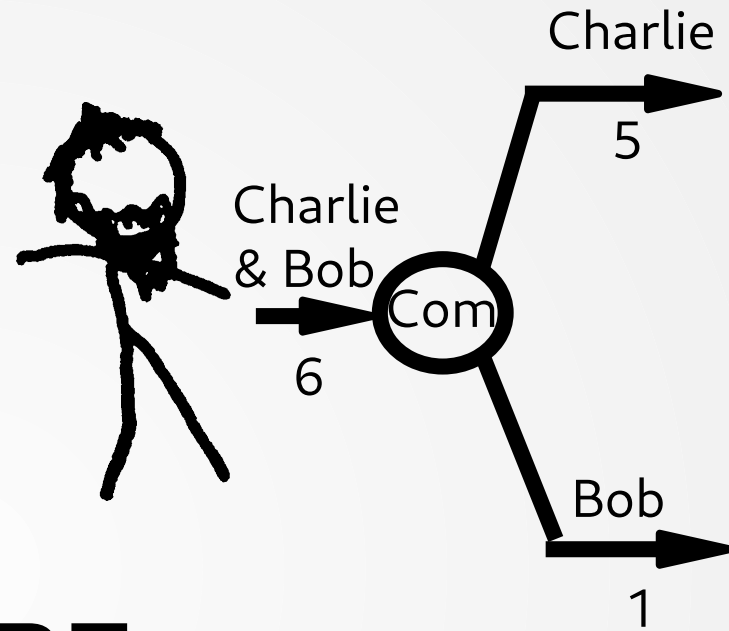
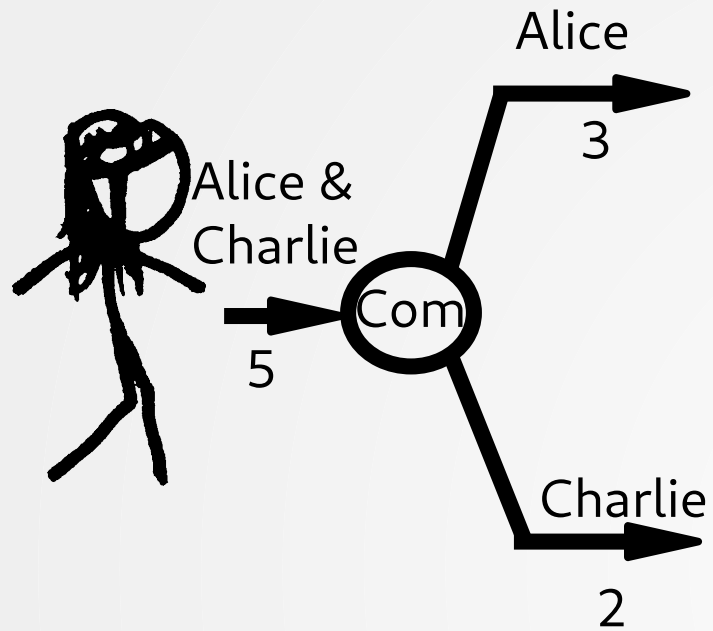




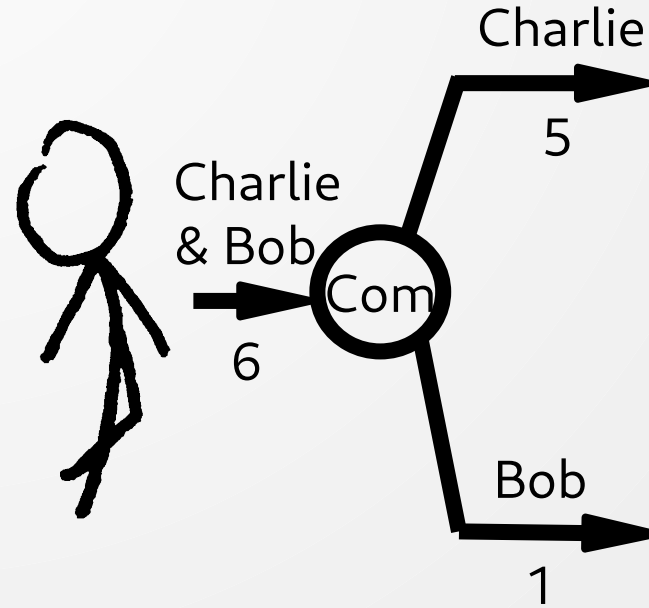
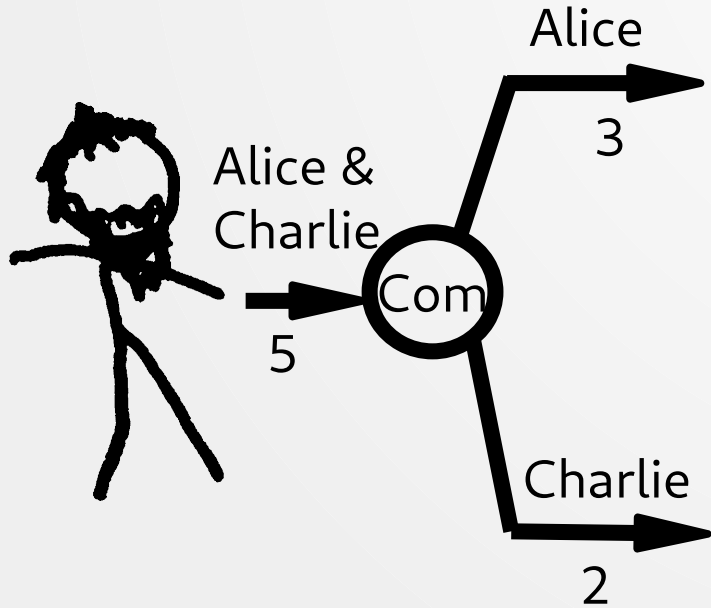
REVOKE\_AND\_ACK,  
sk-Alice', pk-Alice''

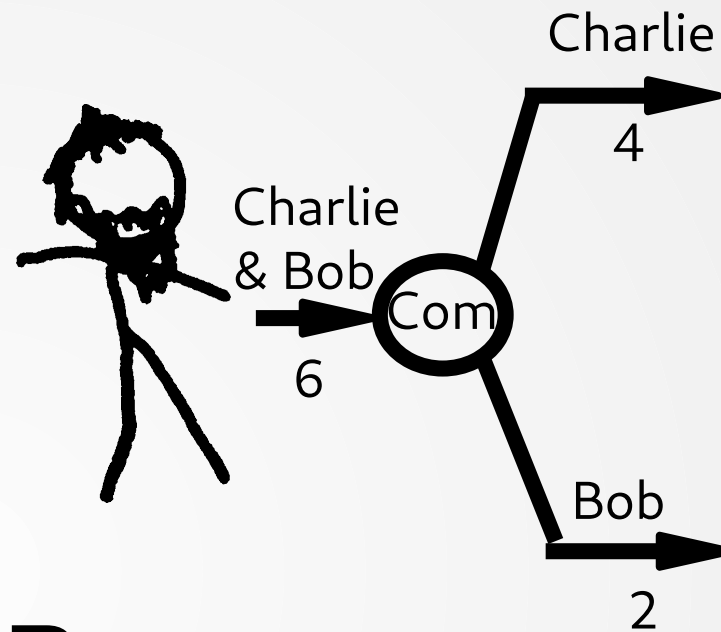
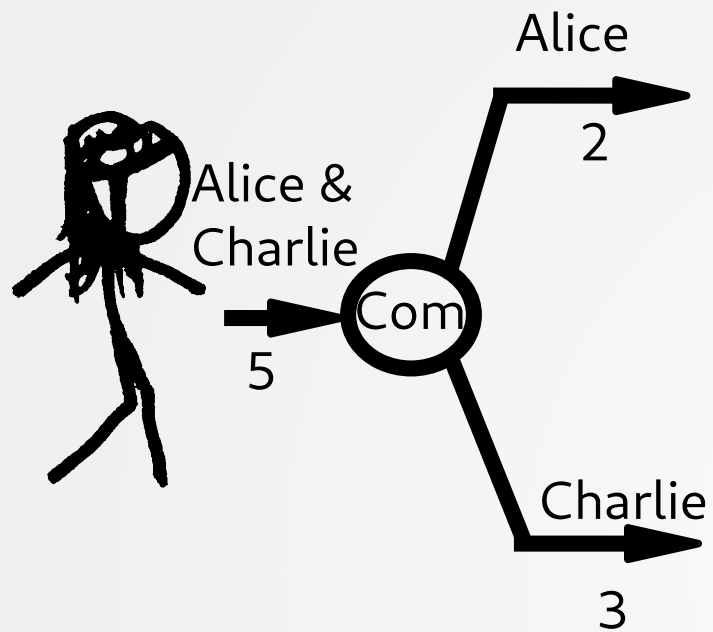




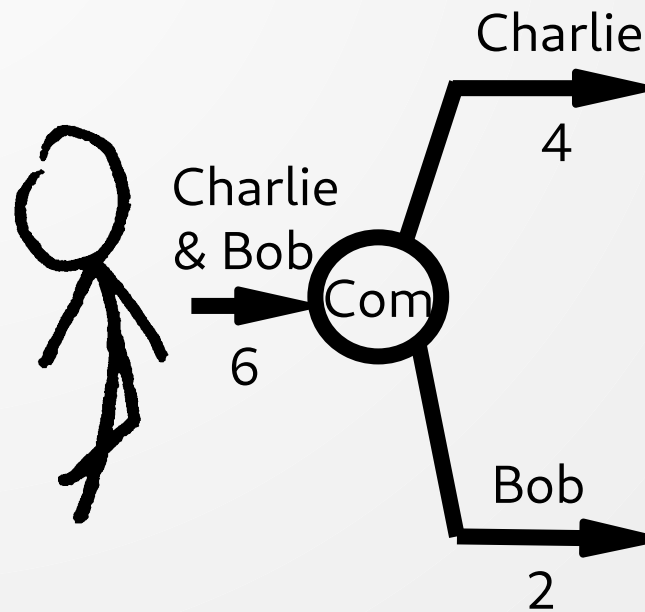
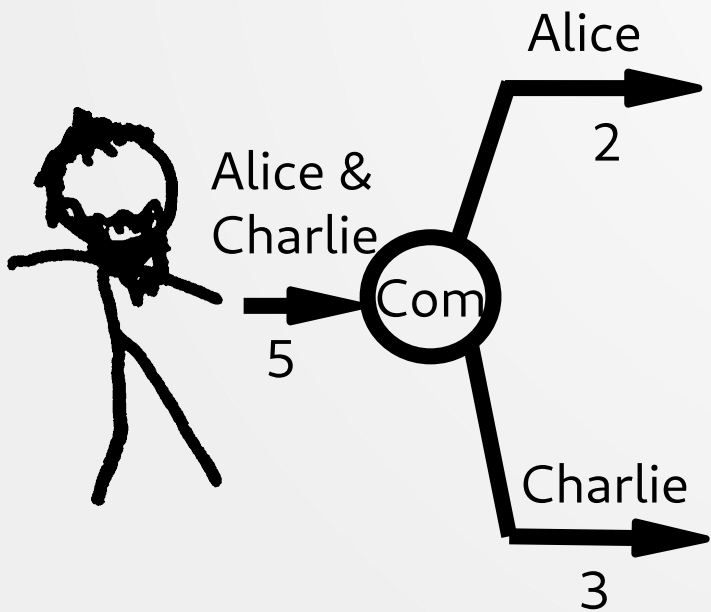


BEFORE





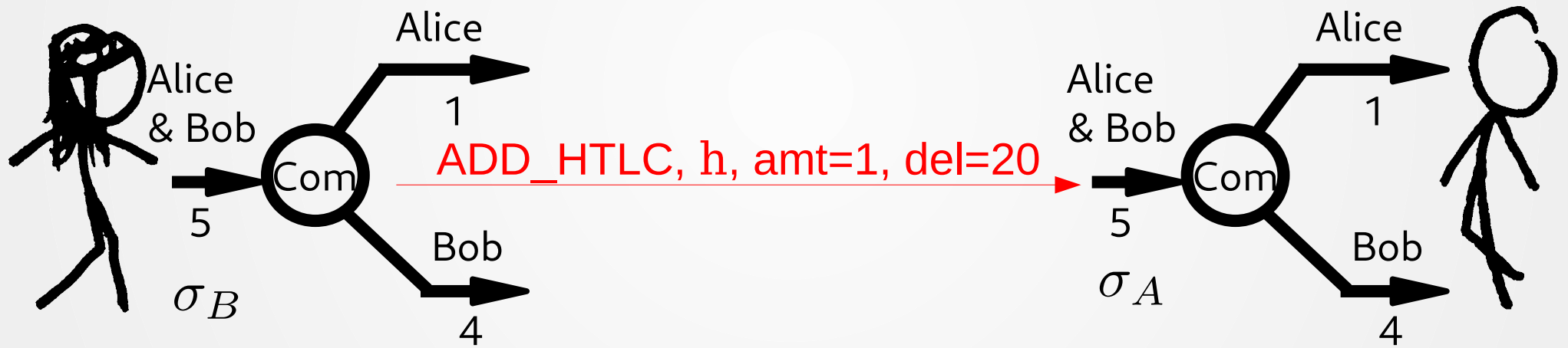
AFTER

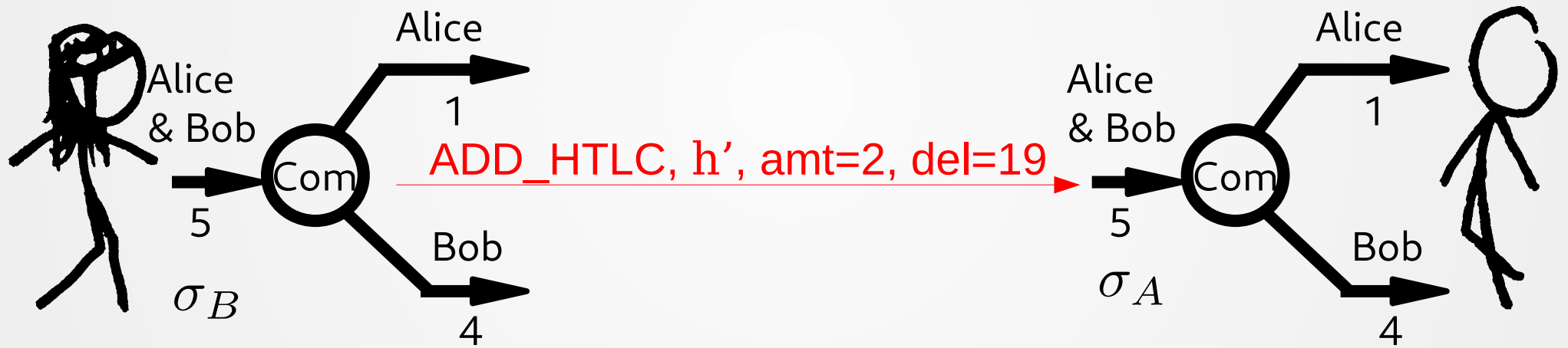


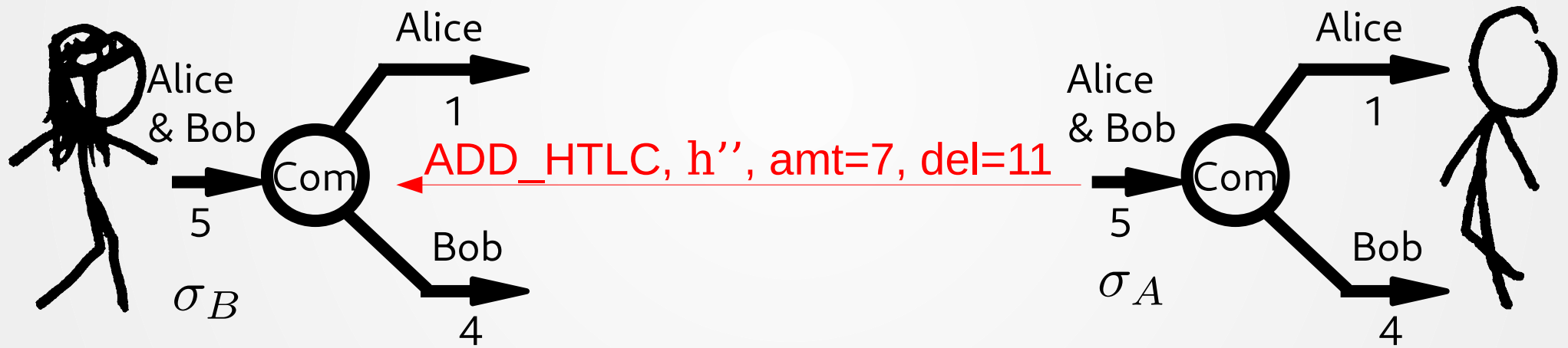
# Part 3

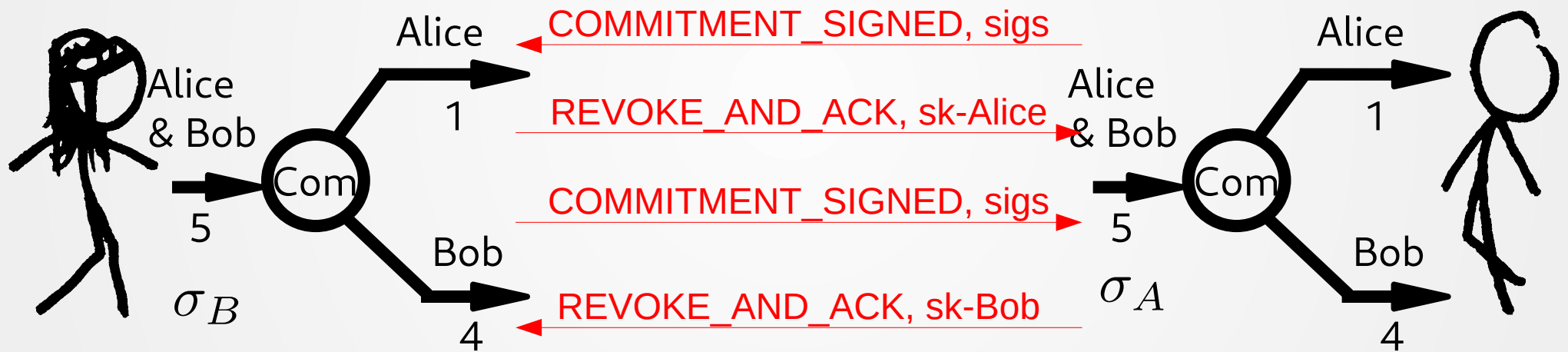
## Optimisations, Privacy, Fees

HTLCs can be batched











# 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, ...

$$\text{Alice\_pubkey\_}\# = \text{basepoint\_}\# + \text{SHA256}(\text{per\_commitment\_point} \parallel \text{basepoint\_}\#) * G$$

(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](https://www.cypherpunks.ca/~iang/pubs/Sphinx_Oakland09.pdf)

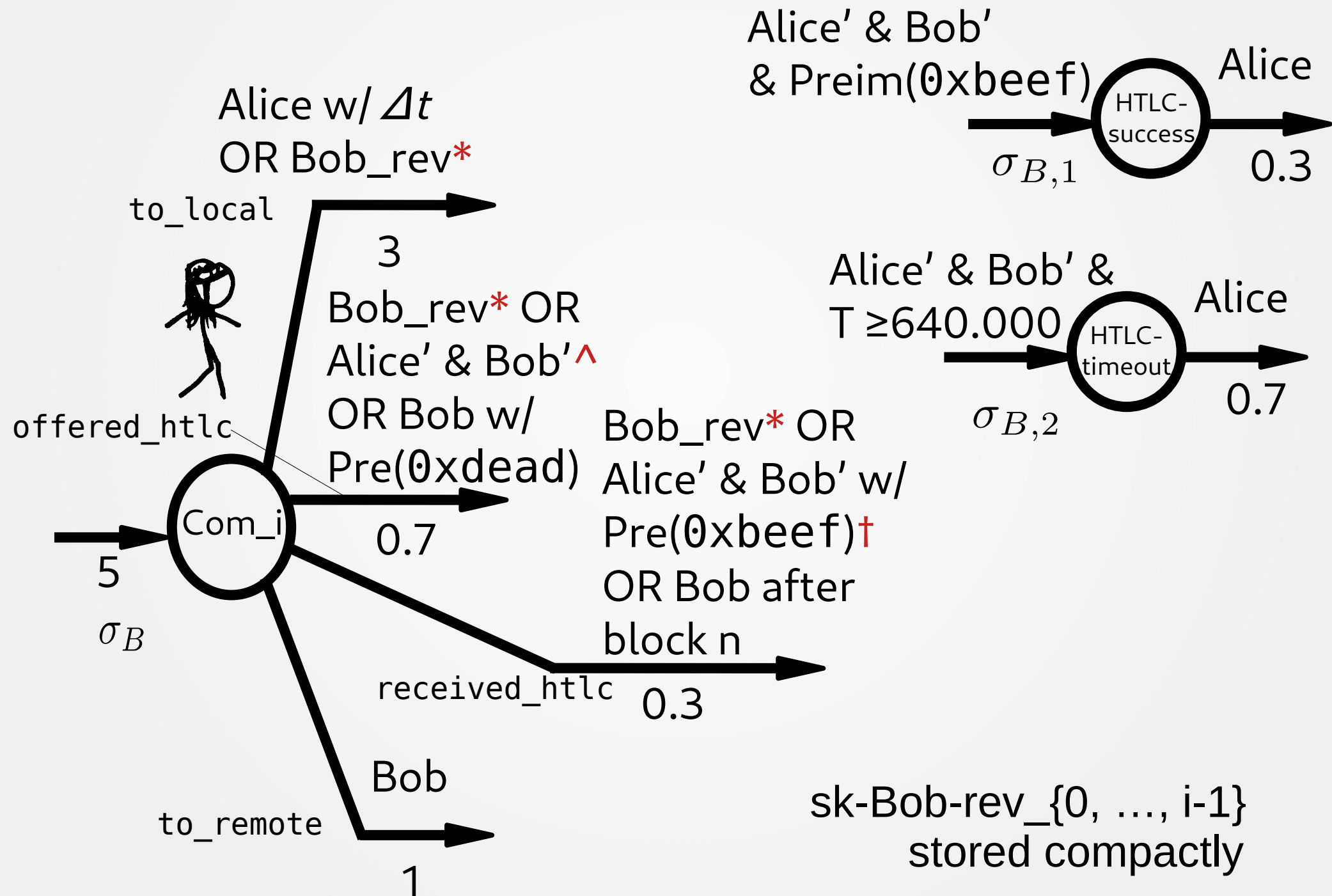
Dust outputs:  
Trim them and stay standard ( )

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!