Outpost: A Lightweight Responsive Watchtower

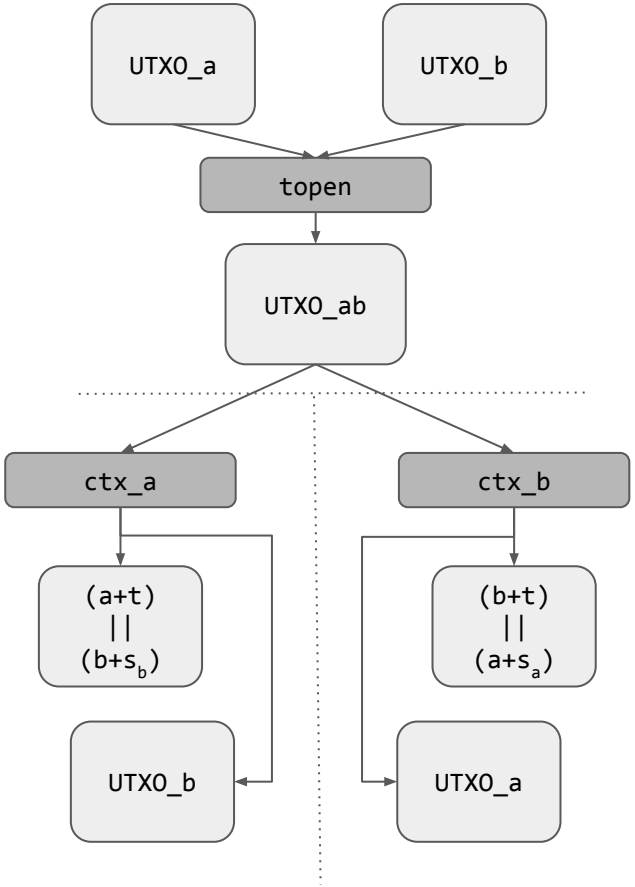
Tejaswi Nadahalli (*ETH Zürich*)
Majid Khabazzian (*Univ. of Alberta*)
Roger Wattenhofer (*ETH Zürich*)

disco.ethz.ch

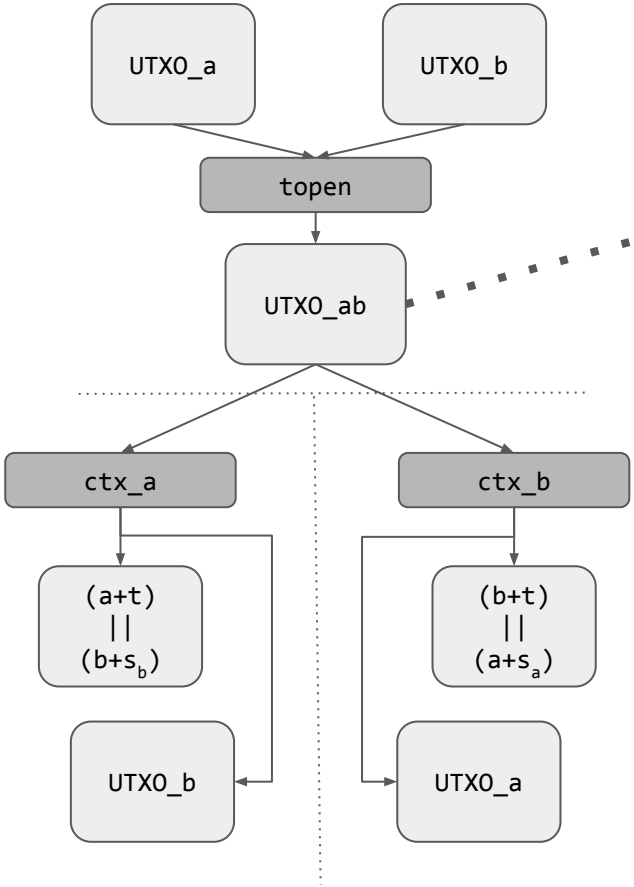
What’s happening?

- 7 TPS
- Channels
- Can’t go offline
- Watchtower
- Watchtower needs to store A LOT
- Does it?
- Alice/Bob have to store things to be able to cheat
- Blockchain!!

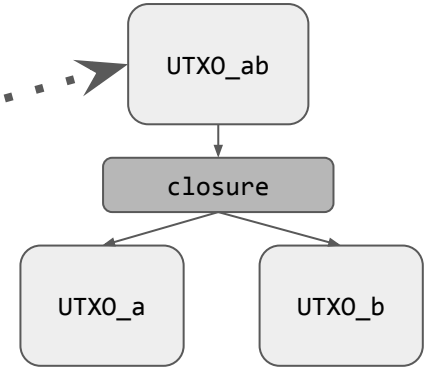
Lightning Channel



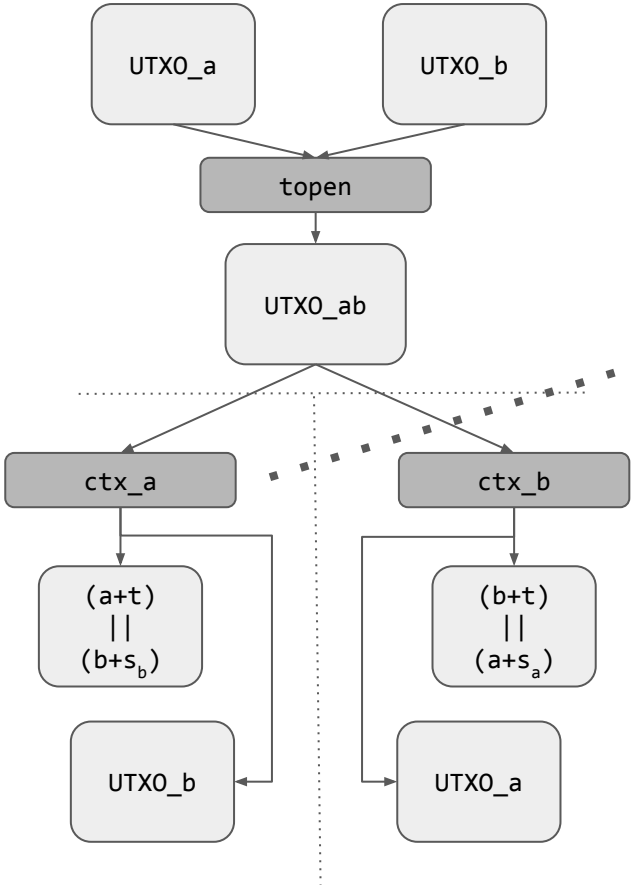
Lightning Channel



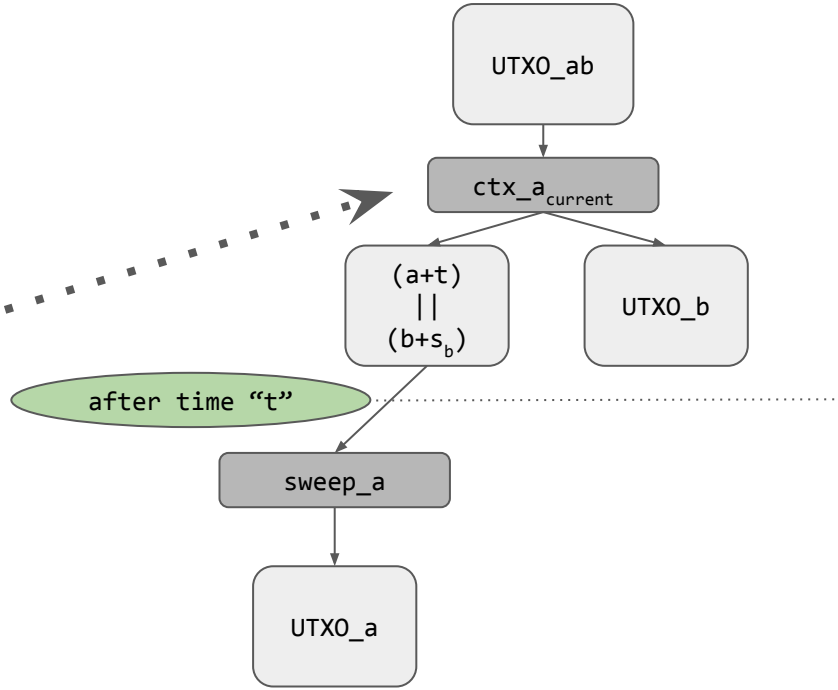
Bilateral Closure



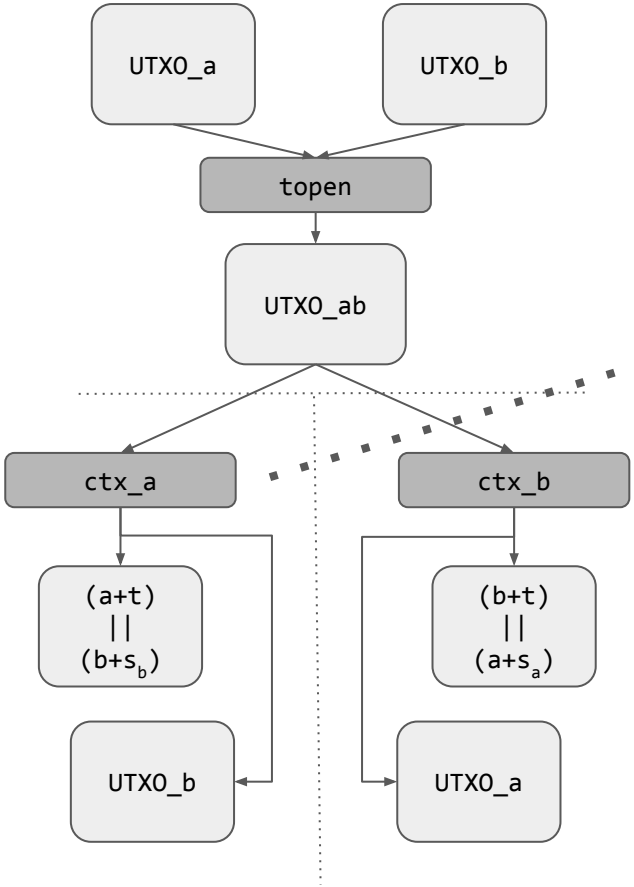
Lightning Channel



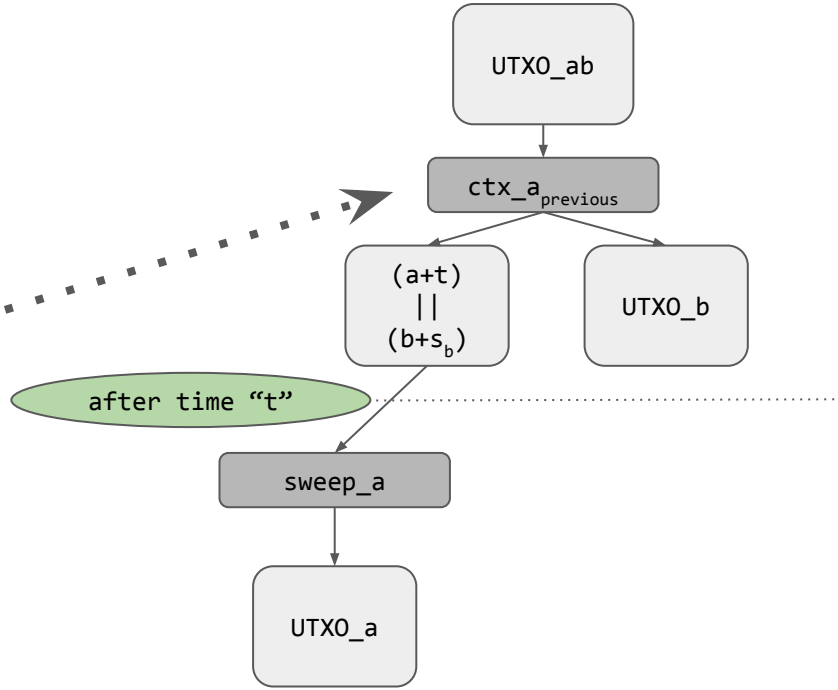
Unilateral Closure



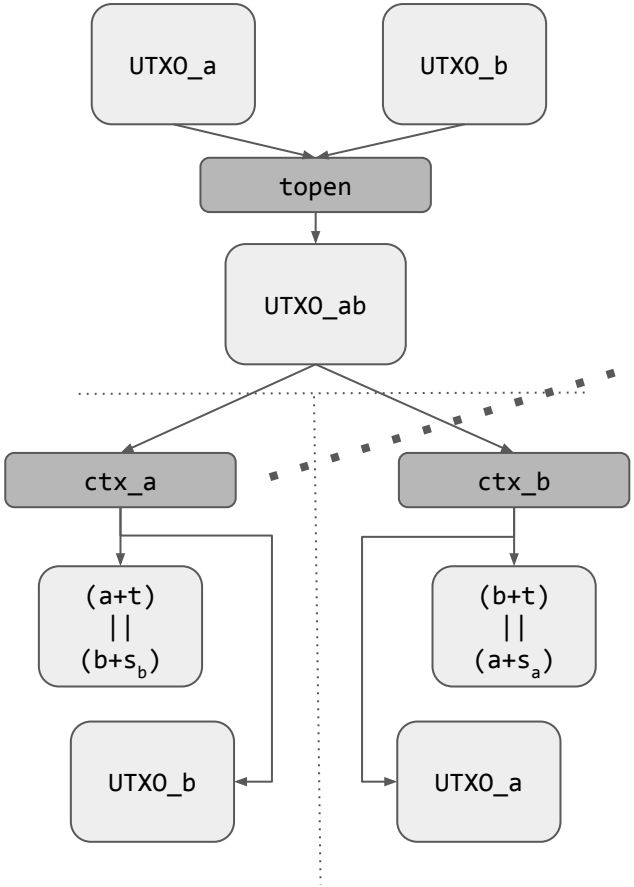
Lightning Channel



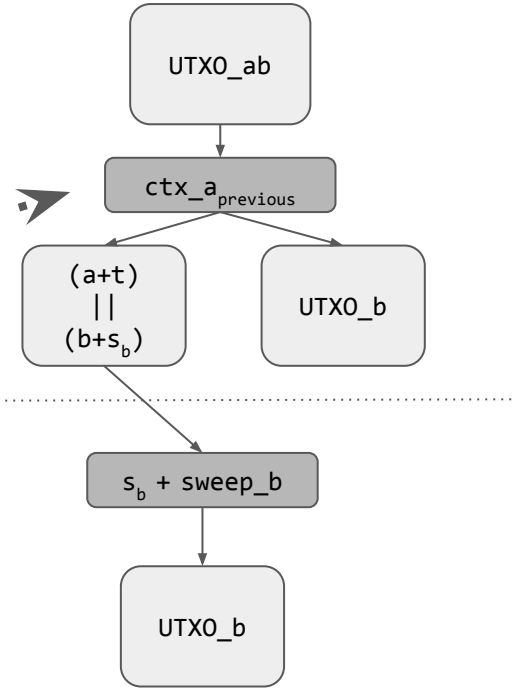
Cheating Closure



Lightning Channel

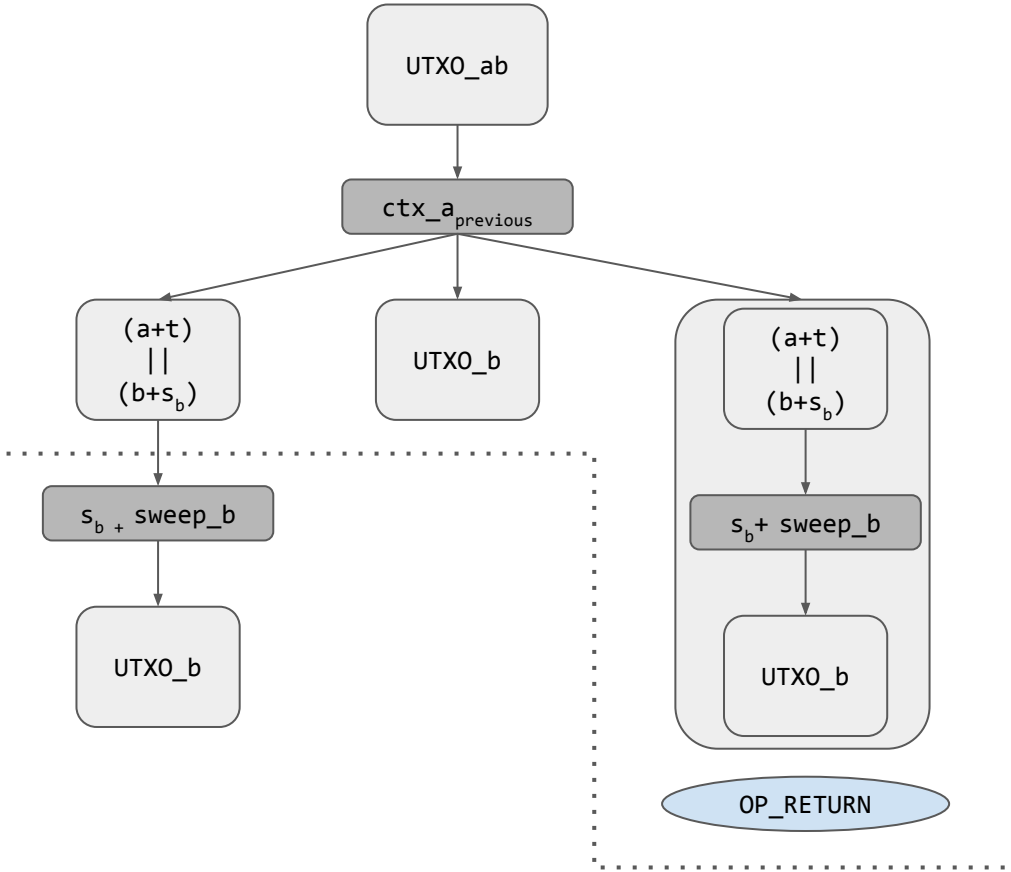


Justice Transaction

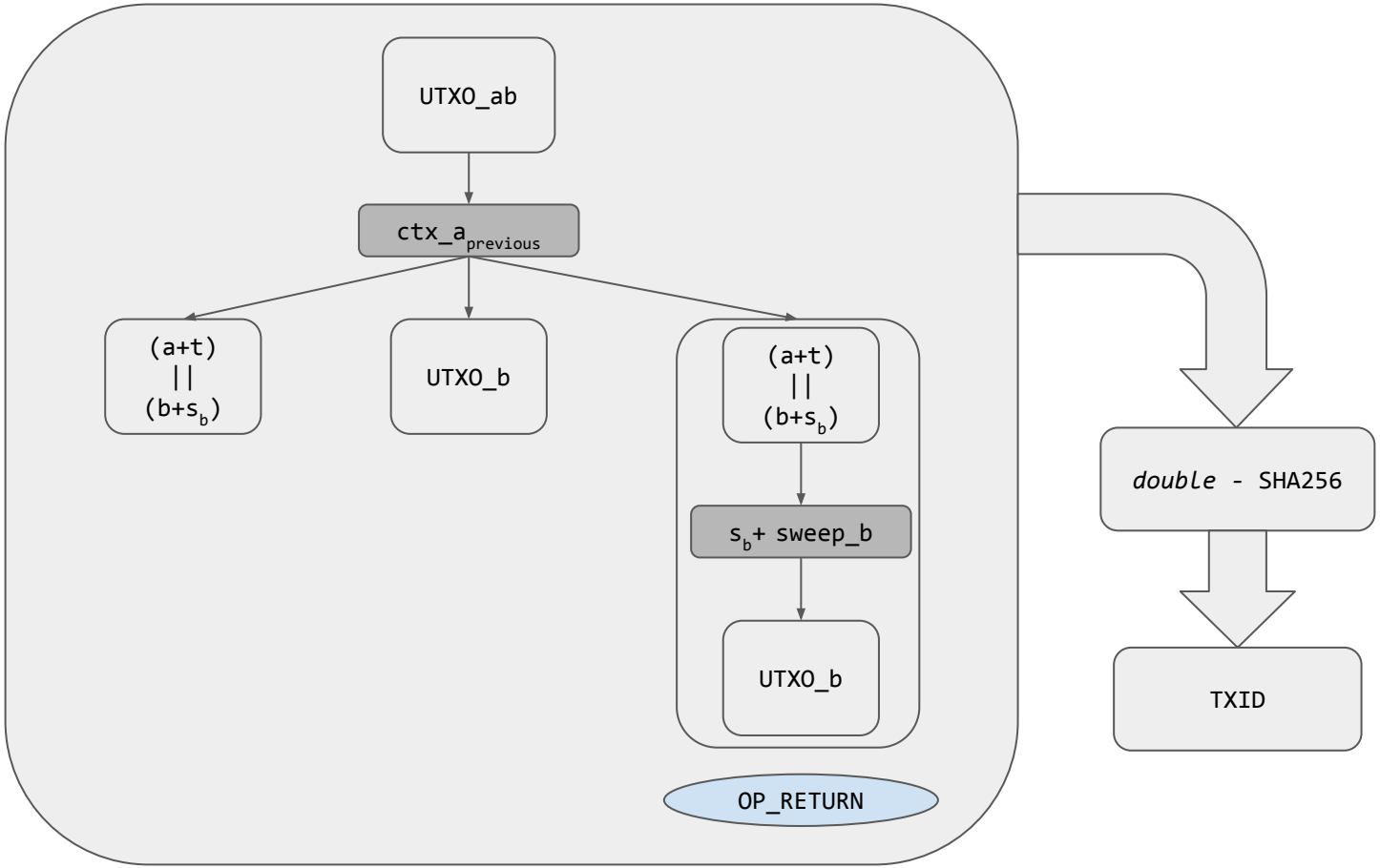


Can we store JTX inside CTX?

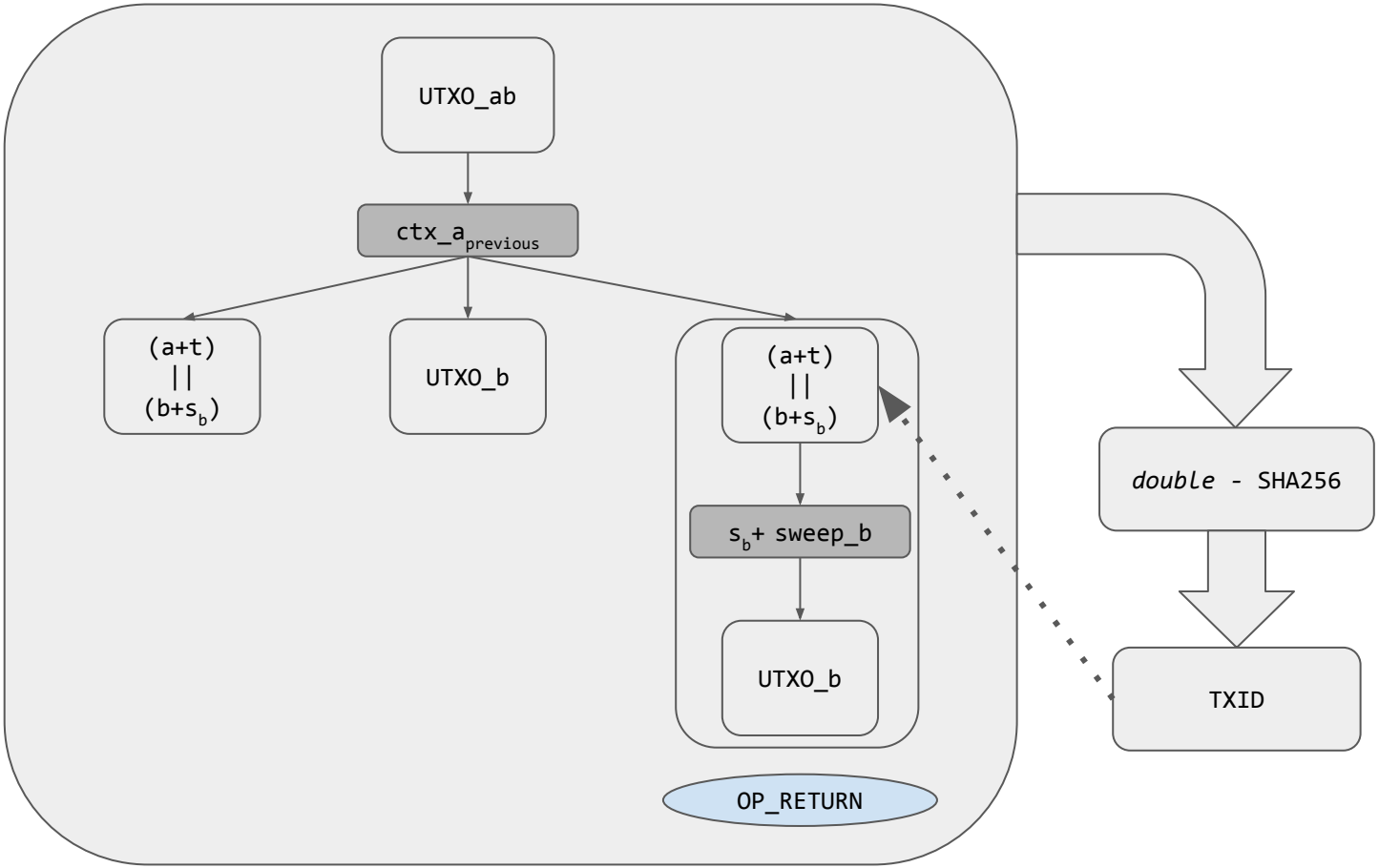
Can we store JTX inside CTX?



TXID


UTXO_ab
ctx_a previous
(a+t)
||
(b+s_b)
UTXO_b
(a+t)
||
(b+s_b)
s_b+ sweep_b
UTXO_b
OP_RETURN
double - SHA256
TXID

TXID makes it self-referential



OP_RETURN alternatives

P2SH Data Drop

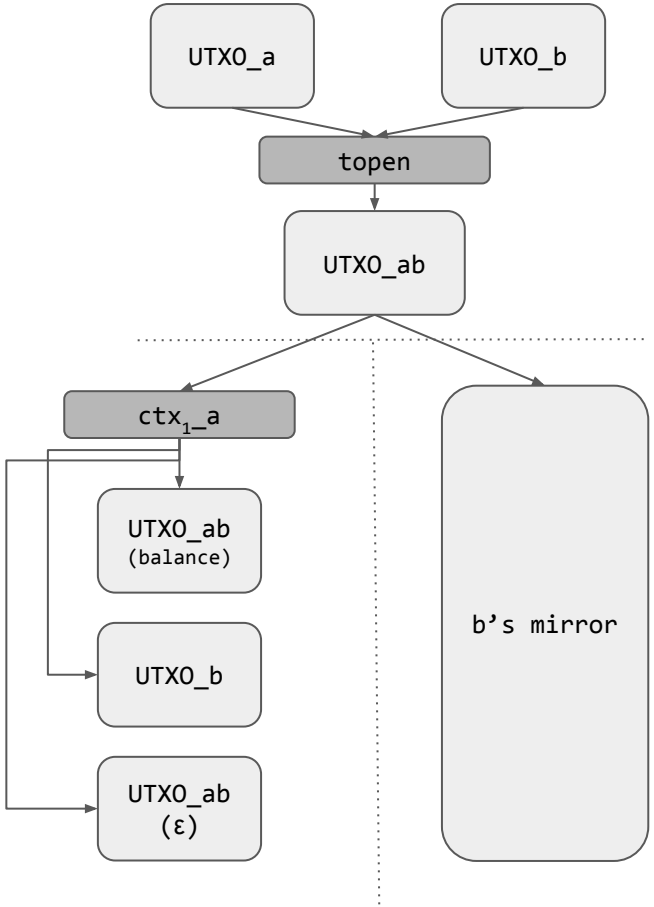
- vulnerable to scriptSig malleability

P2SH Data Hash

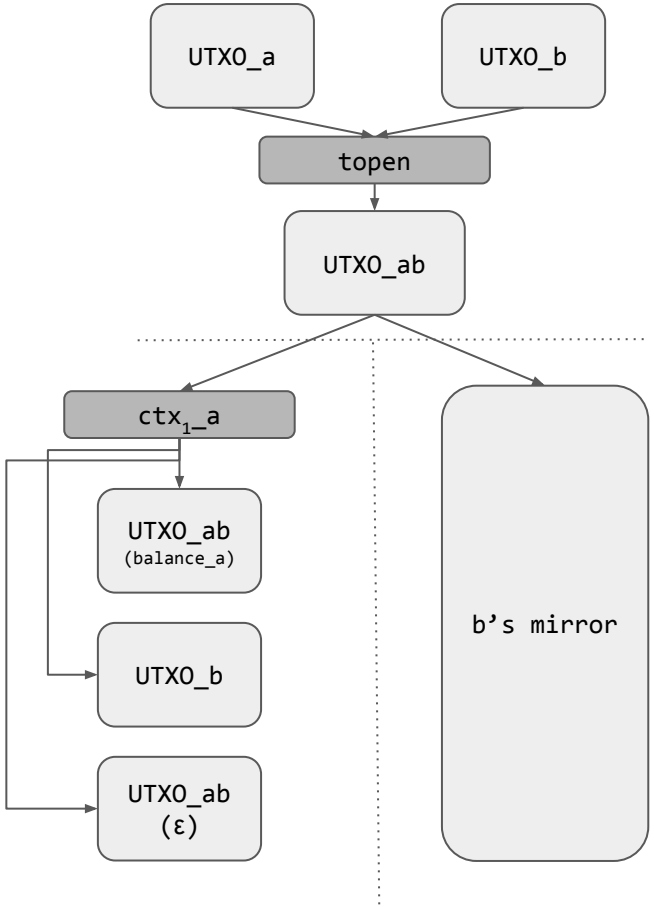
- vulnerable to self-loop in the redeem_script hash

Outpost

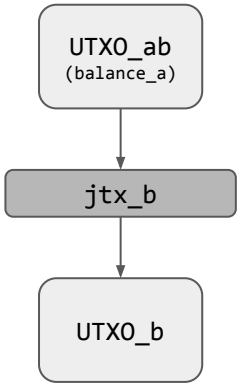
Outpost


UTXO_a
UTXO_b
topen
UTXO_ab
ctx1_a
UTXO_ab
(balance)
UTXO_b
UTXO_ab
(ε)
b's mirror

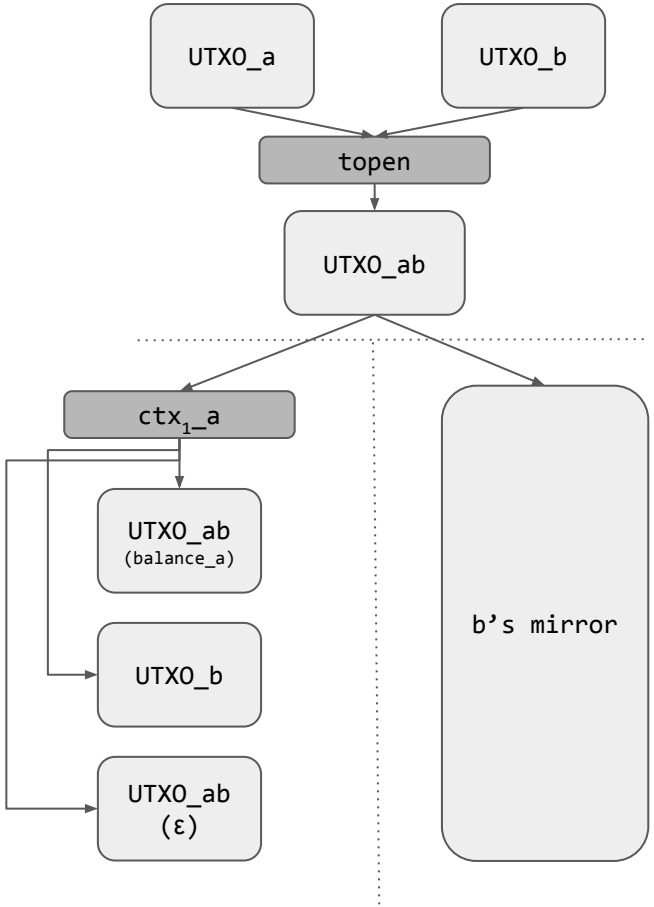
Outpost



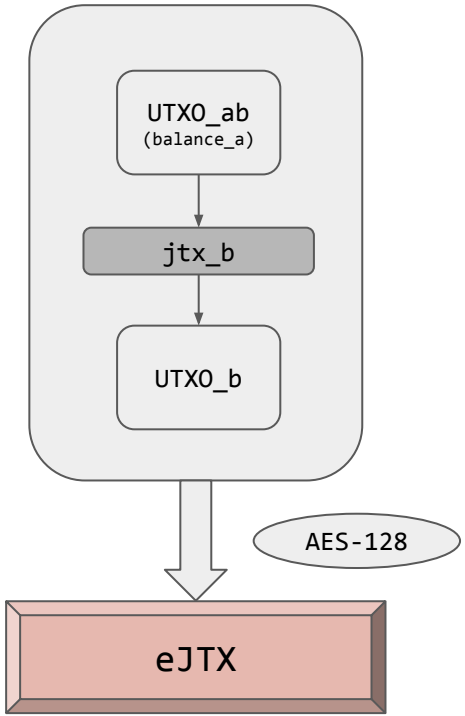
Justice Transaction



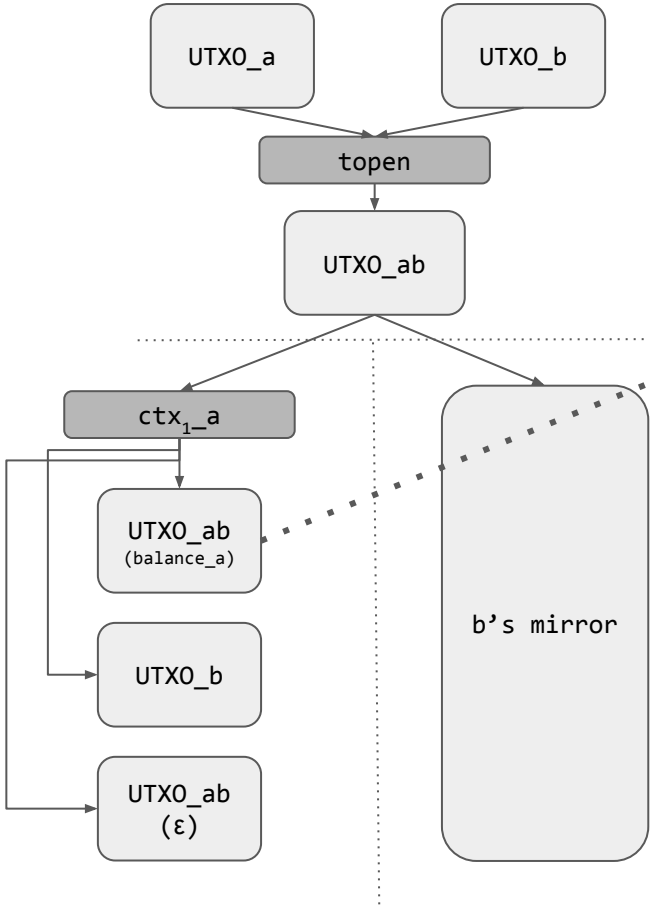
Outpost



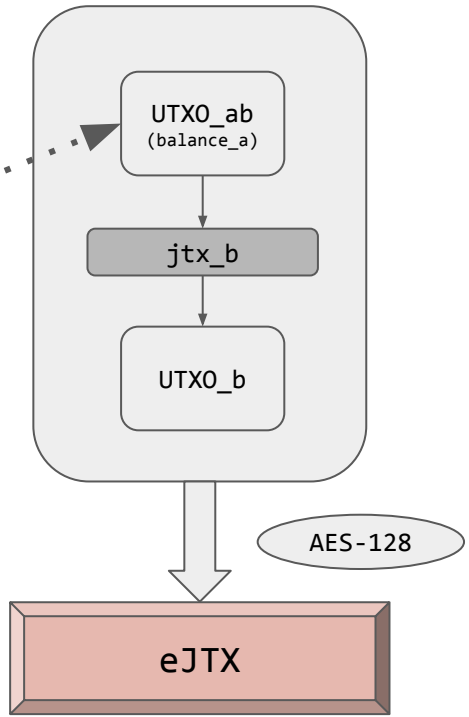
Encrypted Justice Transaction



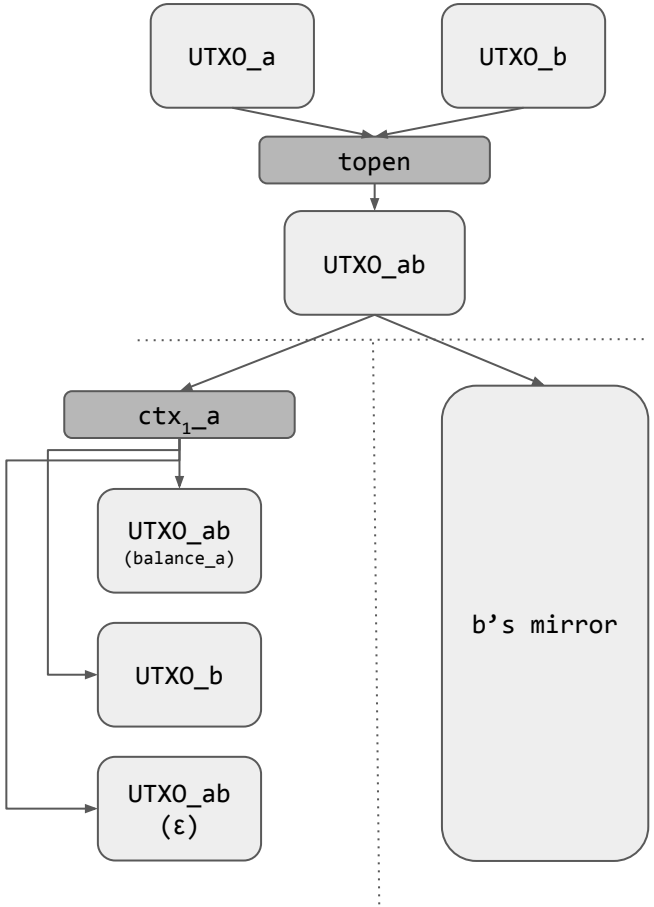
Outpost



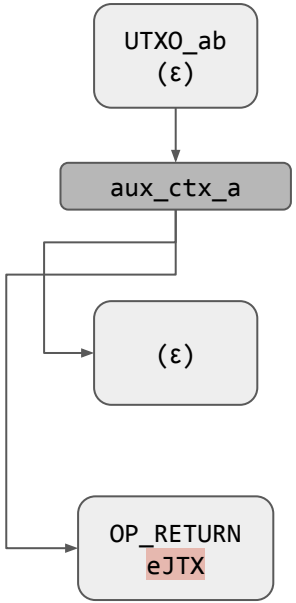
Encrypted Justice Transaction

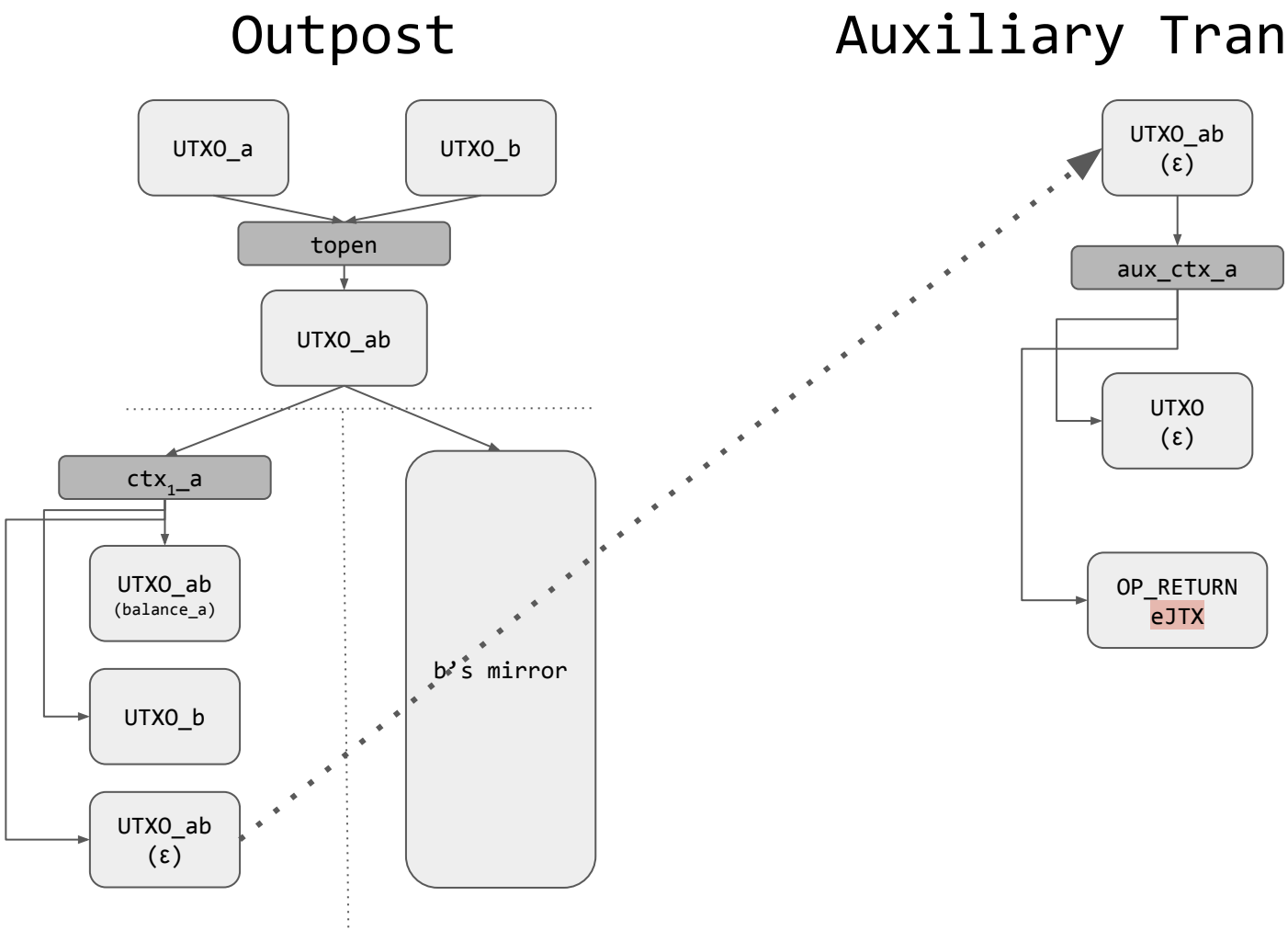


Outpost

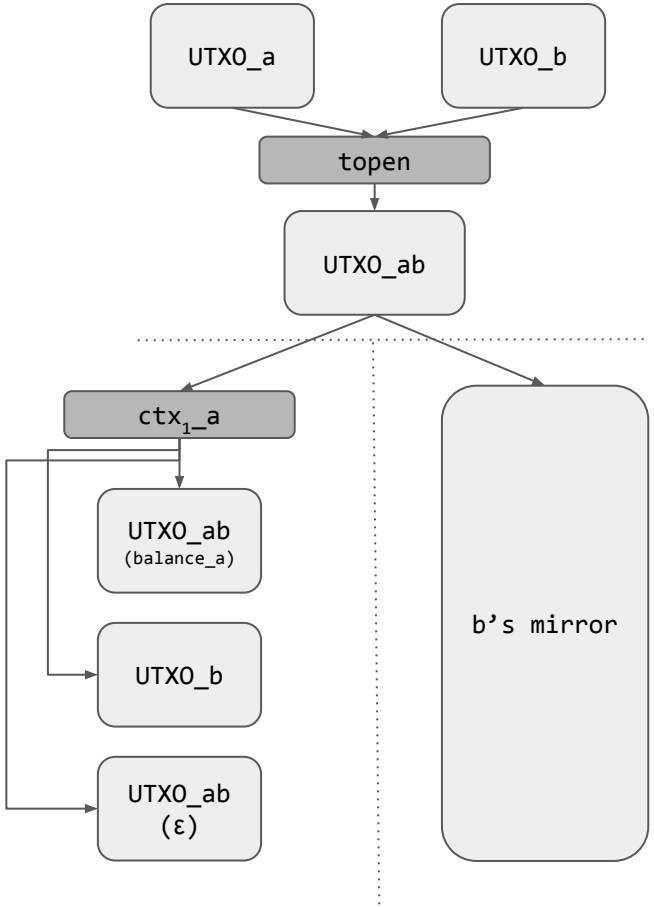


Auxiliary Transaction

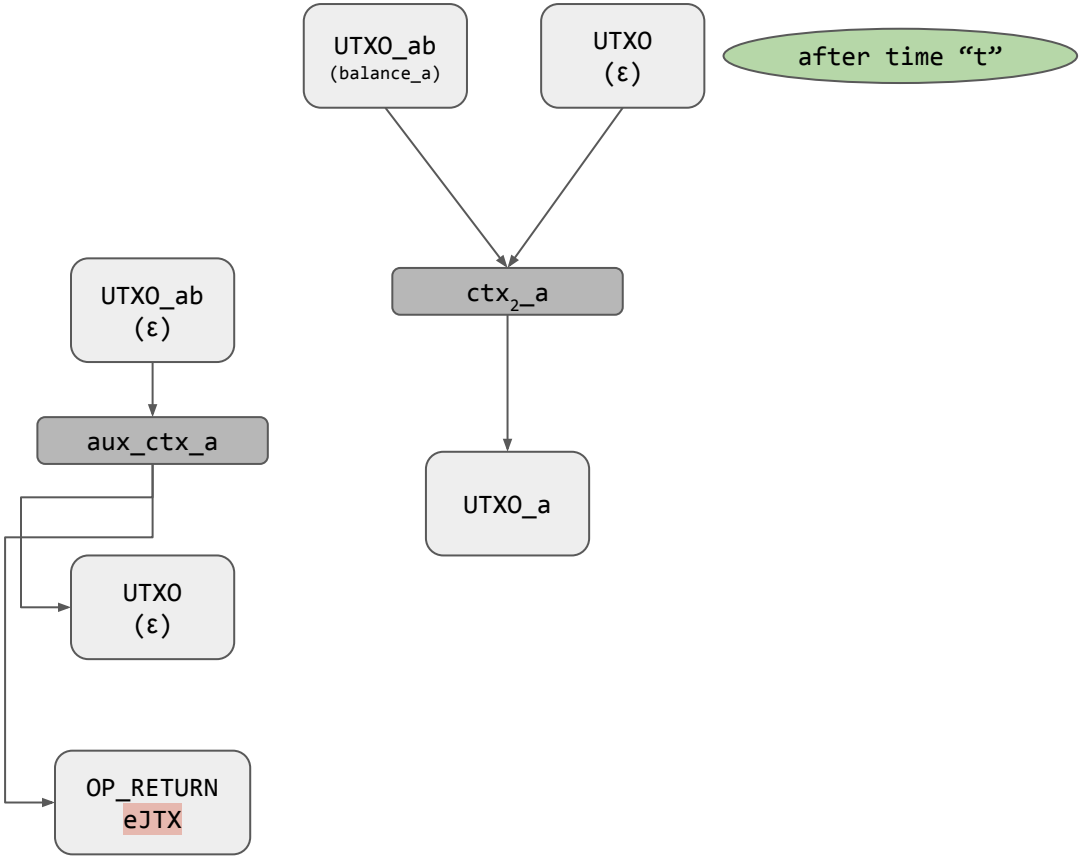


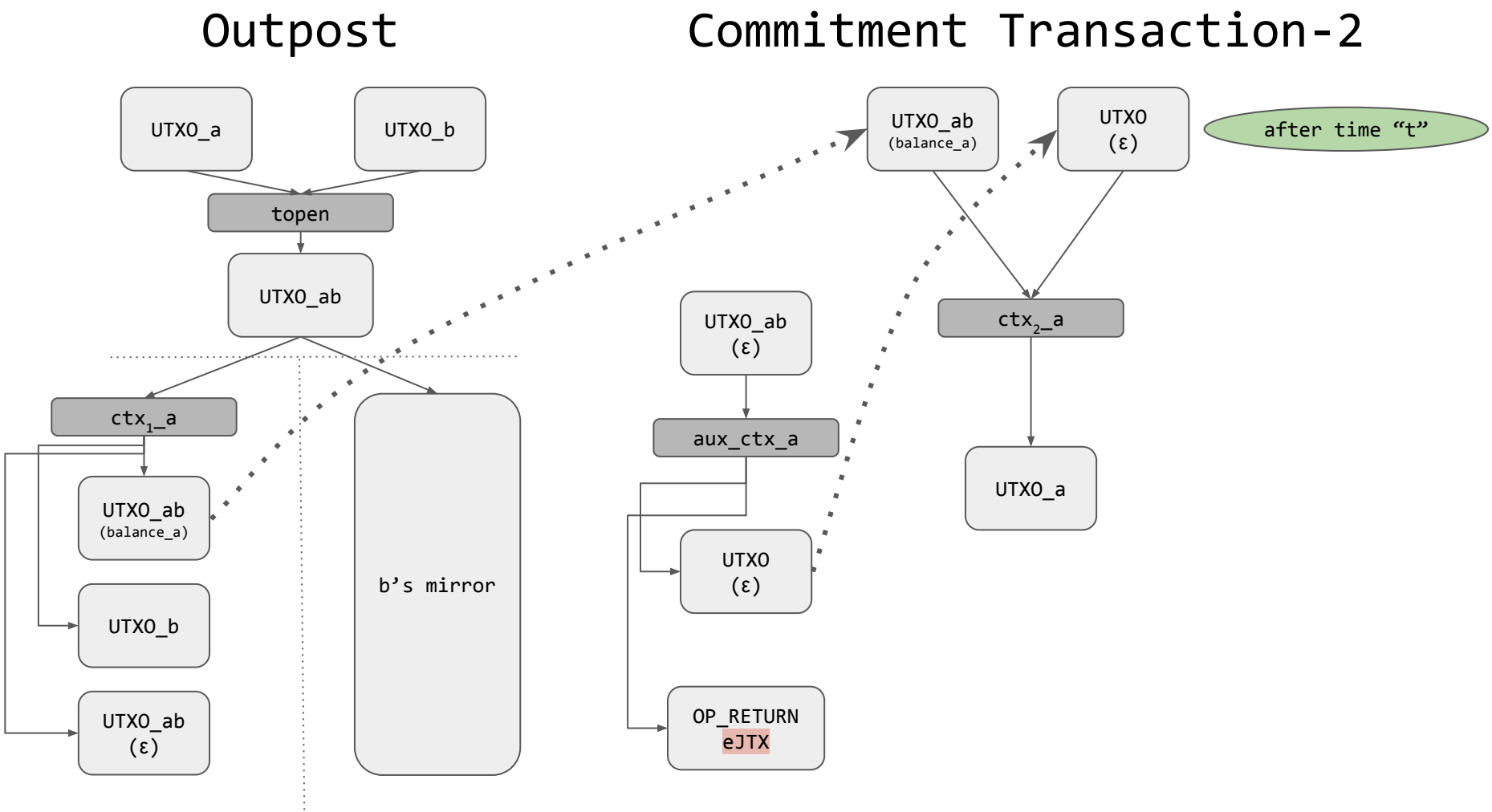

Outpost
Auxiliary Transaction
UTXO_a
UTXO_b
topen
UTXO_ab
ctx1_a
UTXO_ab
(balance_a)
UTXO_b
UTXO_ab
(ε)
b's mirror
UTXO_ab
(ε)
aux_ctx_a
UTXO
(ε)
OP_RETURN
eJTX

Outpost



Commitment Transaction-2




Outpost
Commitment Transaction-2
UTXO_a
UTXO_b
topen
UTXO_ab
ctx1_a
UTXO_ab
(balance_a)
UTXO_b
UTXO_ab
(ε)
b's mirror
UTXO_ab
(balance_a)
UTXO
(ε)
after time "t"
UTXO_ab
(ε)
ctx2_a
aux_ctx_a
UTXO_a
UTXO
(ε)
OP_RETURN
eJTX

Outpost keeps Lightning’s key features

- Unilateral closure: broadcaster has to wait
    - Not cheating
    - Cheating

- Exchange revocation keys vs. AES-128 decryption keys

Cheating (Alice wants to profit)

Time

Alice broadcasts older ctx_1, aux_ctx	Bob is watching the blockchain
Wait	Bob sees aux_ctx
Wait	Bob has key to decrypt eJTX & get JTX_b
Wait	Bob broadcasts JTX_b
Wait	JTX_b is confirmed on the blockchain 😎
Wait	ctx_2 is now invalid
Alice can broadcast ctx_2 - but…	

Unilateral Closure(Bob has disappeared)

Time

Alice broadcasts latest ctx_1, aux_ctx	Bob is secretly watching the blockchain
Wait	Bob sees aux_ctx
Wait	Bob cannot decrypt eJTX
Wait	ctx_2 is still valid
Alice can broadcast ctx_2 - 😎	

Griefing ¯_(ツ)_/¯

Alice broadcasts ctx_1	Bob is watching the blockchain
Wait	Bob sees ctx_1_a confirmed
Wait	Bob's own ctx_1_b is now invalid 😡
Wait	??????
Gone...	

Can we have ctx_1 send all its balance to Bob?

Profit

Outpost

- CTX and eJTX on the blockchain
- eJTX’s key in the node (16 bytes)

Classic Lightning

- CTX on the blockchain
- JTX in the node (~350 bytes)

Cost

Outpost

- 3 txns
- ε

Classic Lightning

- 1 txn
- No ε

Limitations

- OP_RETURN limited to 80 bytes.
    - IsStandard ಠ_ಠ
    - Split aux_ctx into 2; P2SH Data-hash across them

- Bloat
    - Not on the blockchain (happy case)

Optimize!!

SHACHAIN

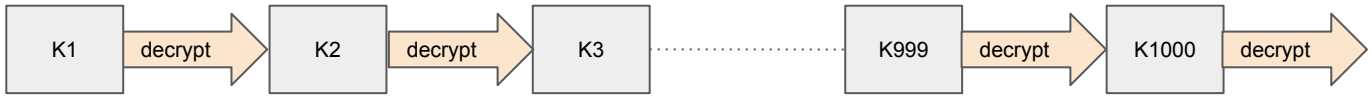
Pre-compute

Use*

RSA-CHAIN

Pre-compute No

Use*



*Key derivation function

Size estimates

Classic Lightning

		20k channels, 1M updates
Known Channel	N·size(ejtx) + 1·size(txid)	7.00 TB
Unknown Channel	N·size(ejtx) + N·size(txid))	7.65 TB

Outpost

Known Channel	size(key) + size(txid)	0.96 MB (WTF)
Unknown Channel	N·size(key) + N·size(txid)	0.96 TB

Note: size(key) << size(ejtx) i.e. 16 << 350

Pay per update

- Online
    - Every Bitcoin block

- Proof of storage
    - Merkle Tree of state updates
    - Ask for proof

Thanks

- Store justice transactions in commitment transactions

- Pay per update, and ask for proof

- tejaswin@ethz.ch
- mkhabbazian@ualberta.ca
- wattenhofer@ethz.ch