Bitcoin Clique:
Channel-free Off-chain Payments using Two-Shot Adaptor Signatures

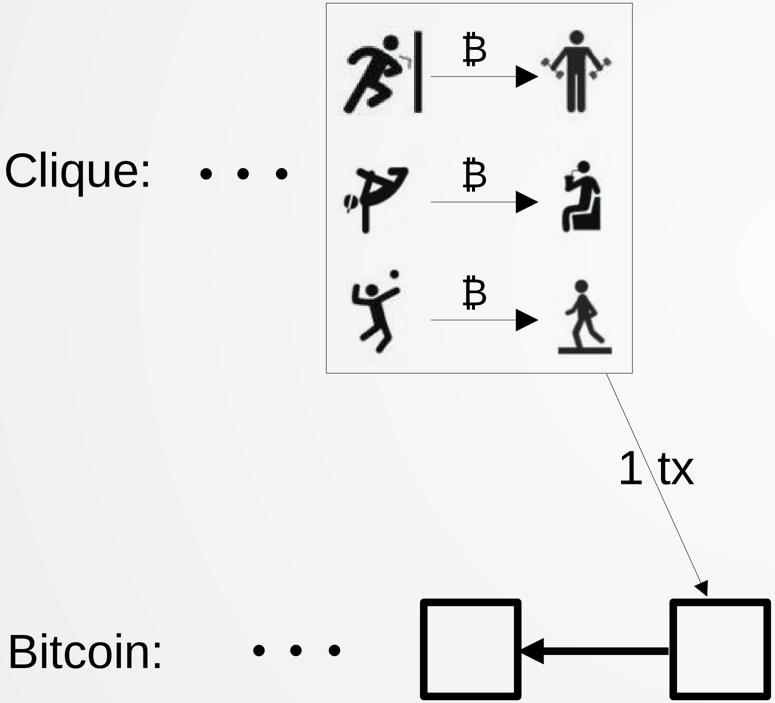
Siavash Riahi
TU Darmstadt

Orfeas Stefanos Thyfronitis Litos
Imperial College London
Common Prefix

SBC
2024-08-09

Commit-chain for Bitcoin*

Commit-chain



Clique: • • •

฿

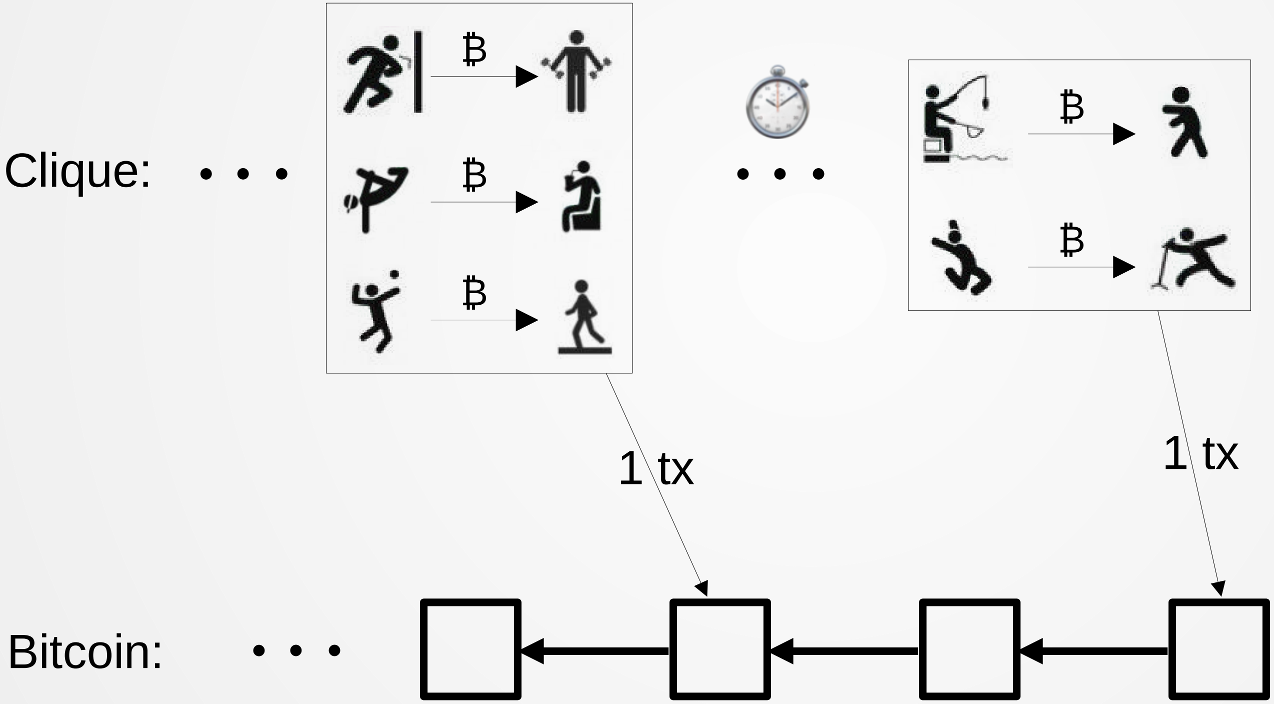
฿

฿

1 tx

Bitcoin: • • •

Commit-chain



Clique:

1 tx

1 tx

Bitcoin:

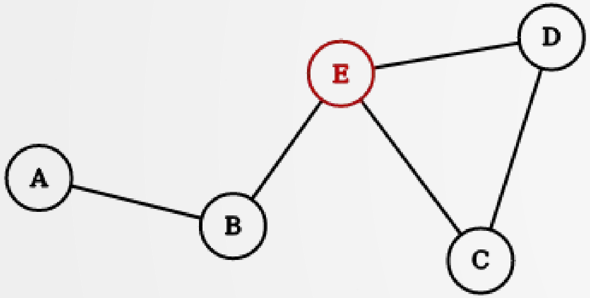
Motivation

- Blockchains are 💸 → Layer-2 solutions
- Why Bitcoin?
  - Security rigorously analyzed 🔐
  - Non-Turing-complete scripting

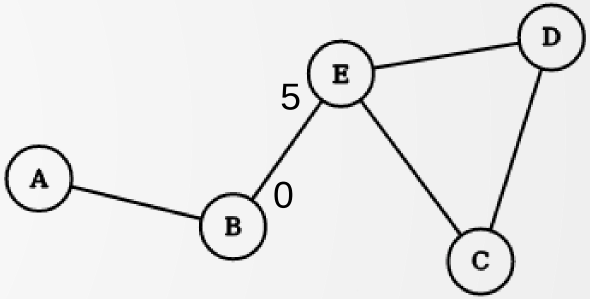
    → solutions should work everywhere
  - Contains > $1T 🤑

Motivation - Why not channels?

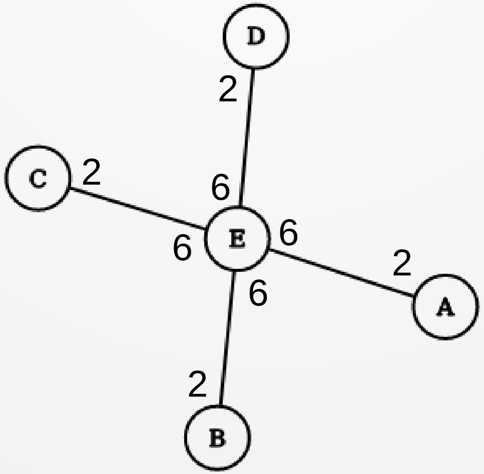
Connectivity



Imbalance



Hub collateral

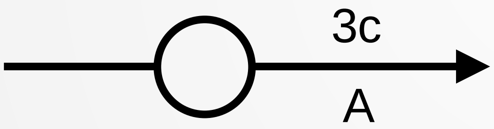


Goals

- Pay anyone in the Clique 🤝
- Free to leave unilaterally 🏃
- Known max time to finality ⏱️
- No extra trust 😇
- Few on-chain TXs 🪶

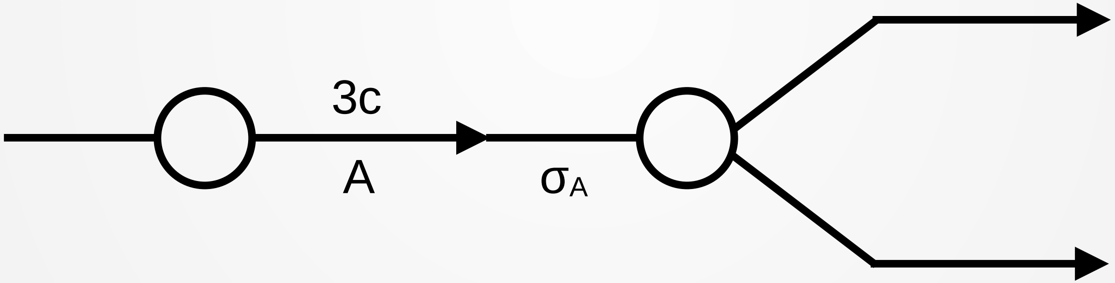
UTXO & Notation

c = e.g., 10,000 satoshis



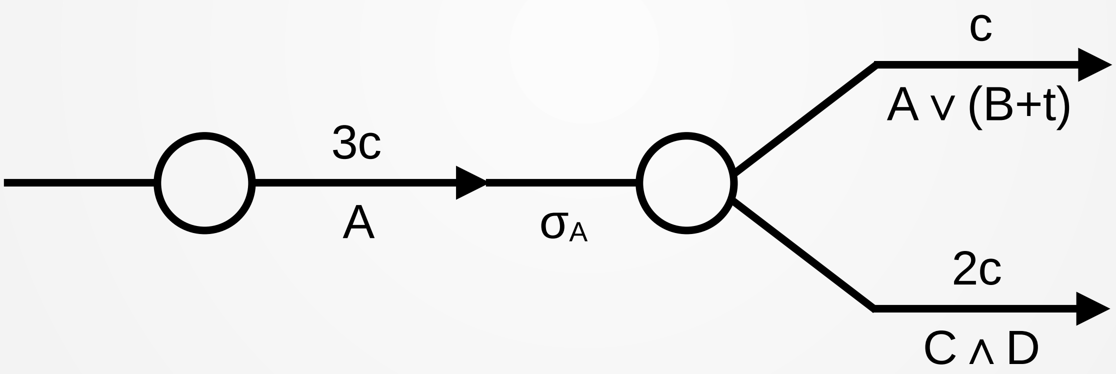
UTXO & Notation

c = e.g., 10,000 satoshis

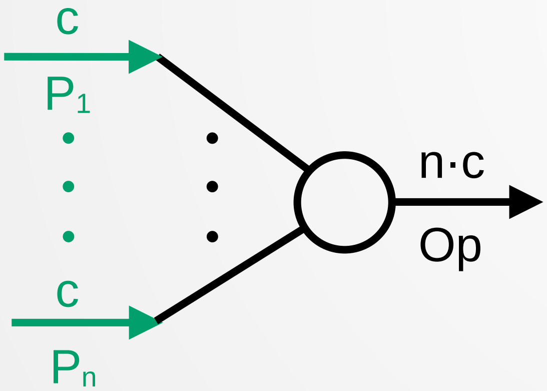


UTXO & Notation

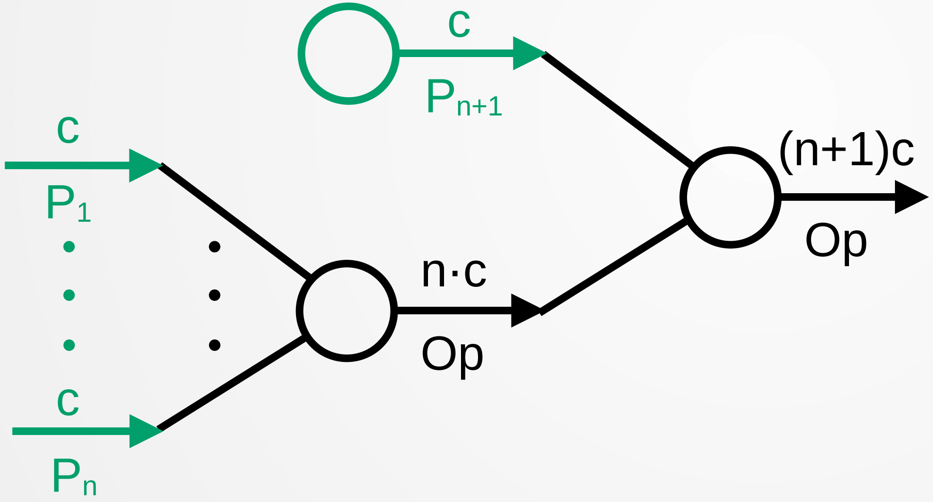
c = e.g., 10,000 satoshis



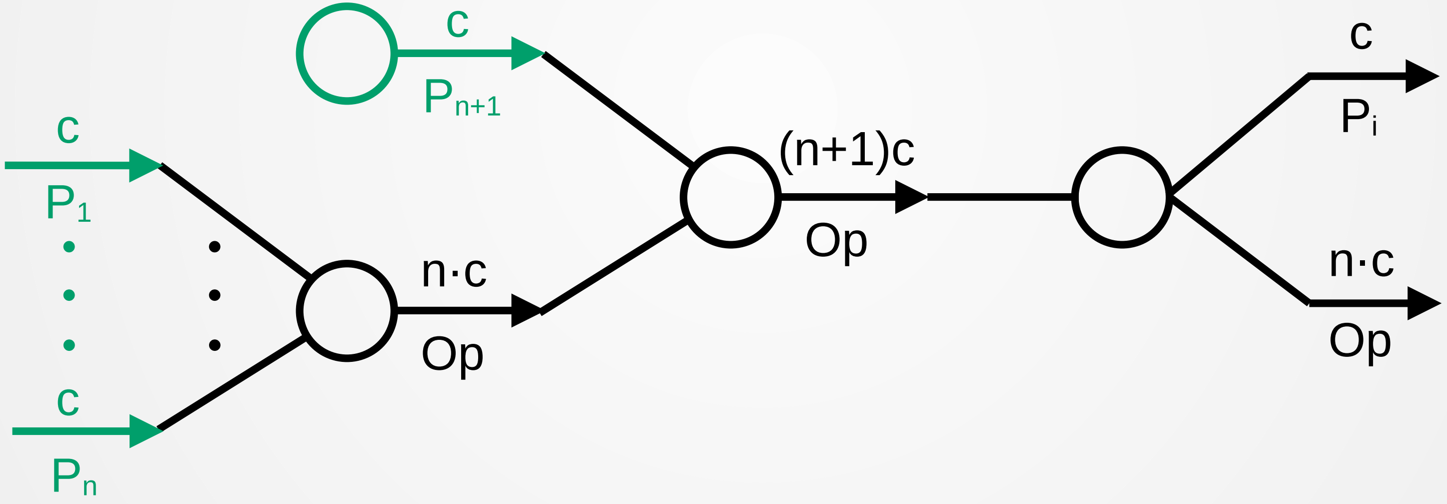
Strawman solution



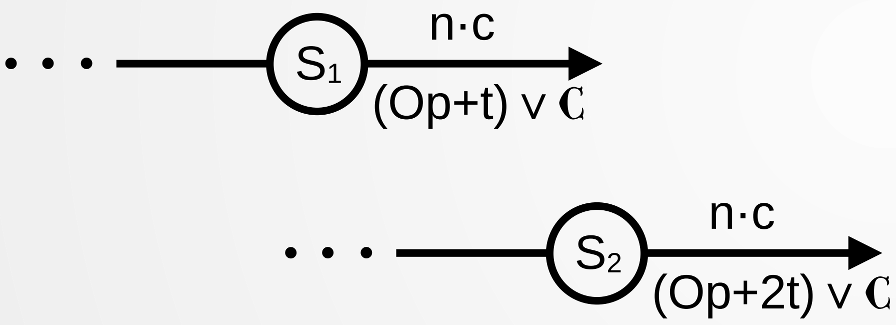
Strawman solution



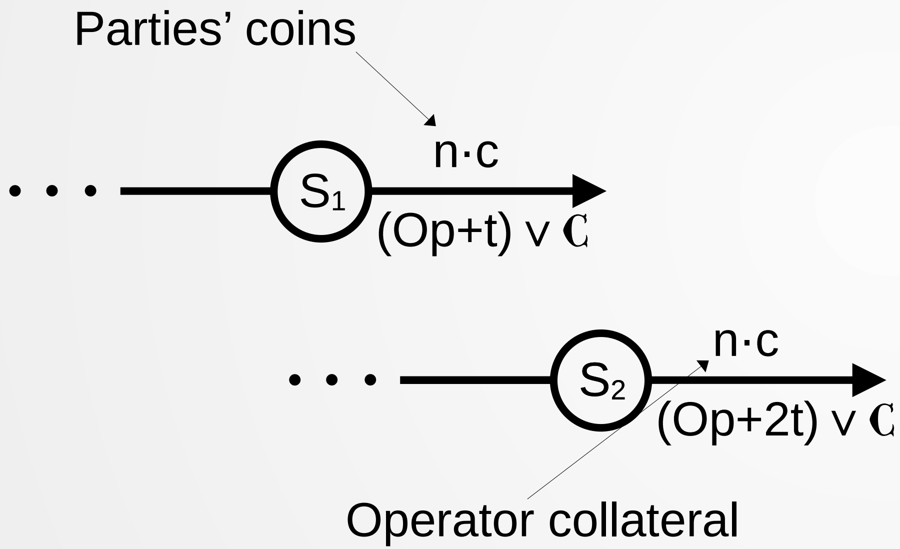
Strawman solution



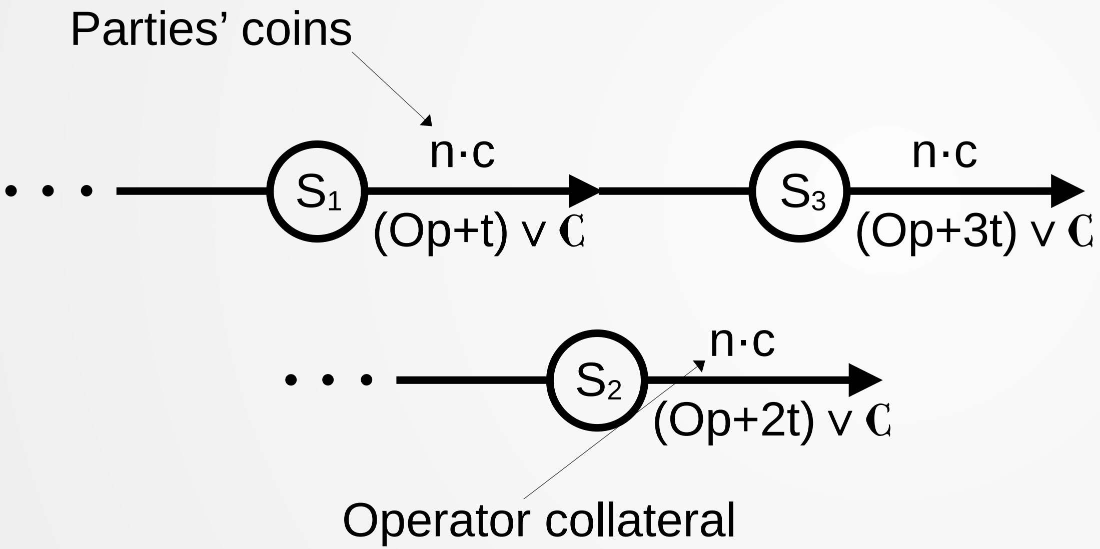
Alternating timelocked outputs



Alternating timelocked outputs



Alternating timelocked outputs



Parties’ coins

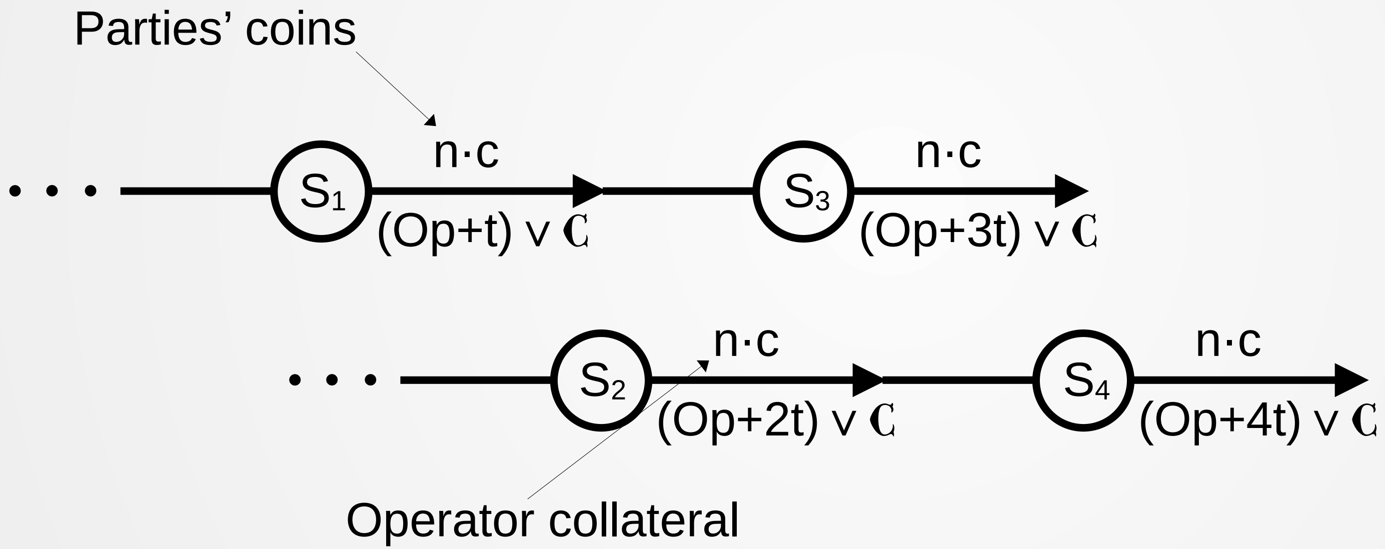
• • • S_1 n·c (Op+t) v $\mathfrak{C}$

S_3 n·c (Op+3t) v $\mathfrak{C}$

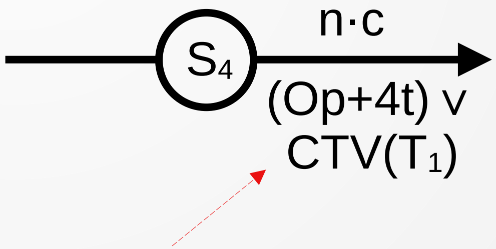
• • • S_2 n·c (Op+2t) v $\mathfrak{C}$

Operator collateral

Alternating timelocked outputs



CHECKTEMPLATEVERIFY (CTV)



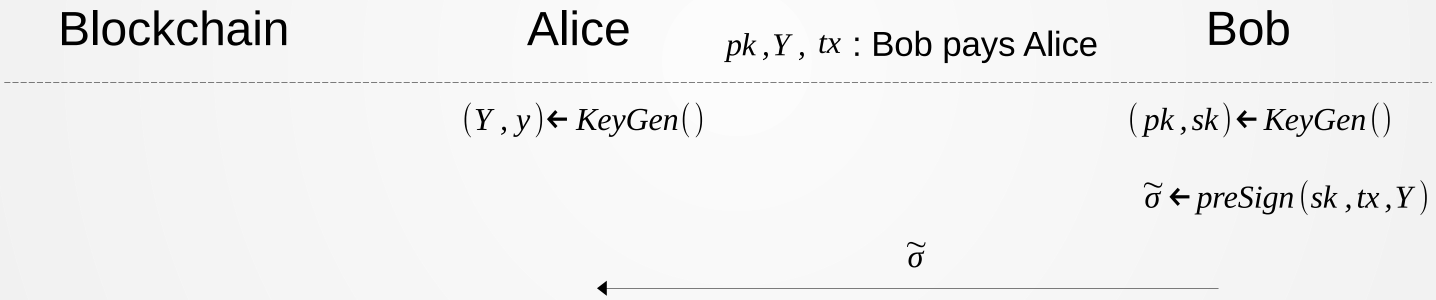
Spendable by:

- Op after time 4t OR
- TX T_1 at any time

Cryptographic intermission: Adaptor Signatures

Blockchain	Alice	pk, Y, tx : Bob pays Alice	Bob
	$(Y, y) \leftarrow KeyGen()$		$(pk, sk) \leftarrow KeyGen()$

Cryptographic intermission: Adaptor Signatures



Blockchain

Alice

$pk, Y,\ tx$: Bob pays Alice

Bob

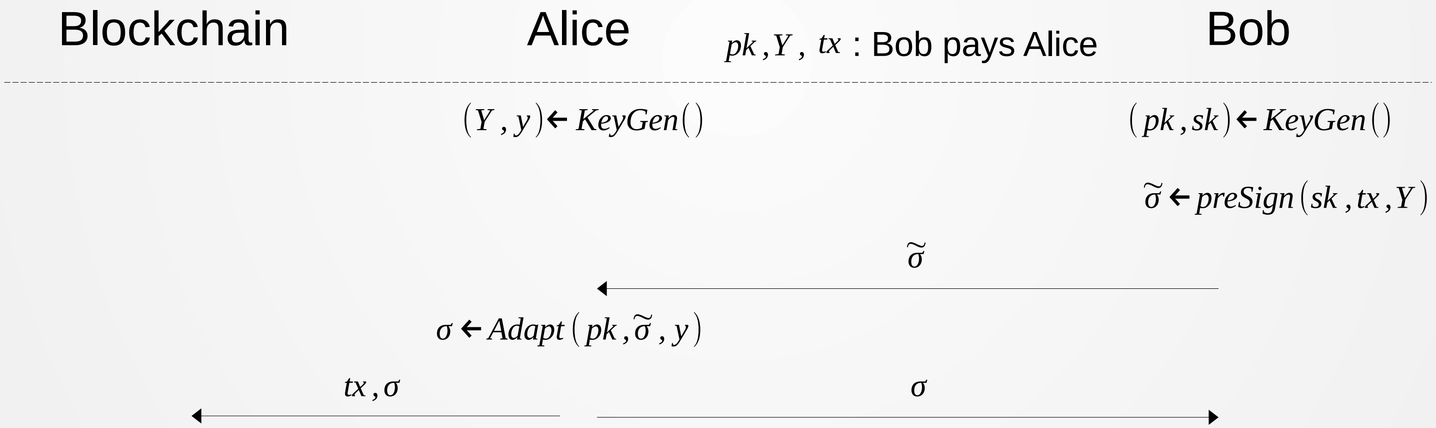
$(Y, y) \leftarrow KeyGen()$

$(pk, sk) \leftarrow KeyGen()$

$\tilde{\sigma} \leftarrow preSign(sk, tx, Y)$

$\tilde{\sigma}$

Cryptographic intermission: Adaptor Signatures



Blockchain

Alice

$pk, Y,$ tx : Bob pays Alice

Bob

$(Y, y) \leftarrow KeyGen()$

$(pk, sk) \leftarrow KeyGen()$

$\tilde{\sigma} \leftarrow preSign(sk, tx, Y)$

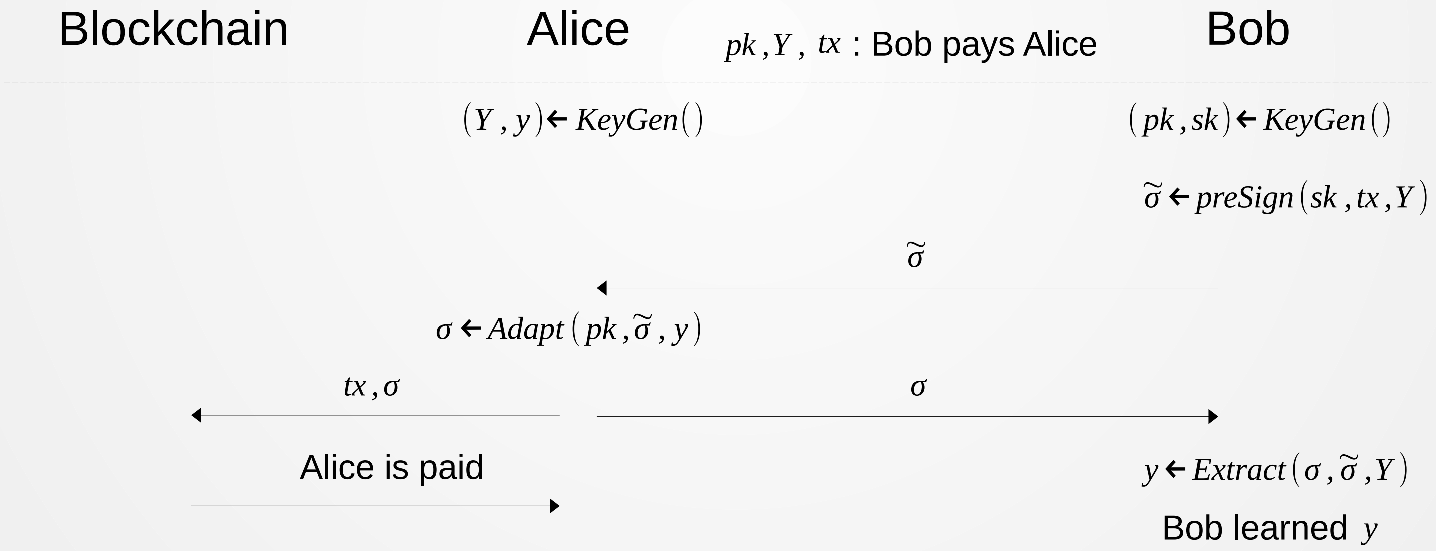
$\tilde{\sigma}$

$\sigma \leftarrow Adapt(pk, \tilde{\sigma}, y)$

tx, σ

σ

Cryptographic intermission: Adaptor Signatures



Blockchain

Alice

pk, Y, tx : Bob pays Alice

Bob

$(Y, y) \leftarrow KeyGen()$

$(pk, sk) \leftarrow KeyGen()$

$\tilde{\sigma} \leftarrow preSign(sk, tx, Y)$

$\tilde{\sigma}$

$\sigma \leftarrow Adapt(pk, \tilde{\sigma}, y)$

tx, σ

σ

Alice is paid

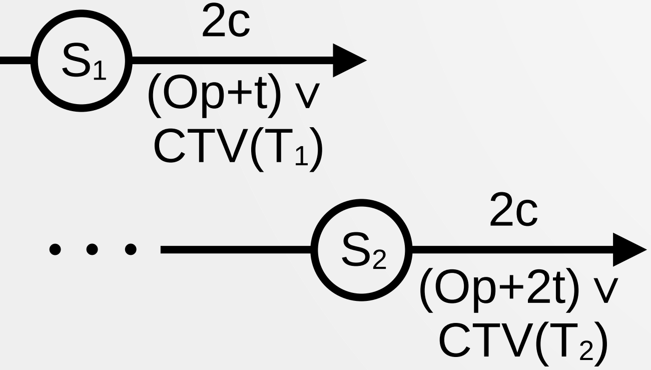
$y \leftarrow Extract(\sigma, \tilde{\sigma}, Y)$

Bob learned y

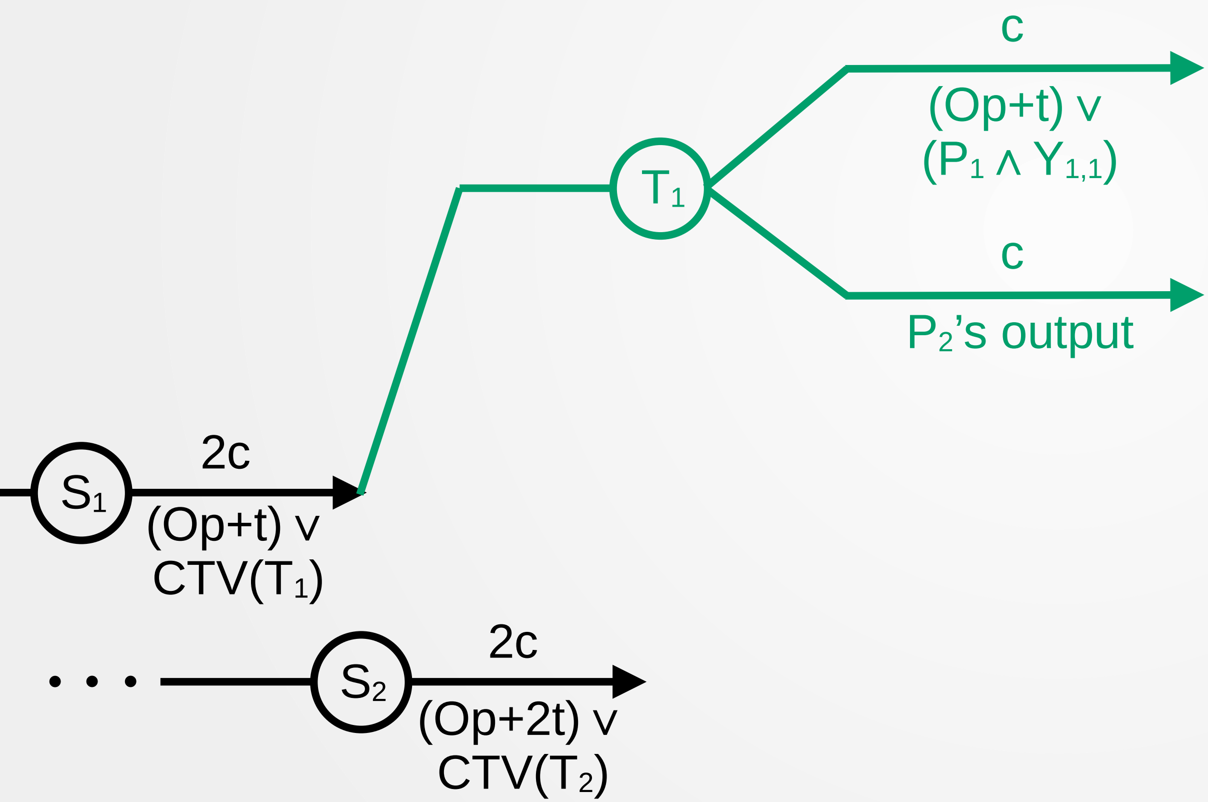
Cryptographic intermission: Two-Shot Adaptor Signatures

- Alice has 2 pairs $(Y_1, y_1), (Y_2, y_2)$
- Bob wants to learn $y_1 + y_2$
- Alice’s secret is safe if she only discloses y_1 or y_2

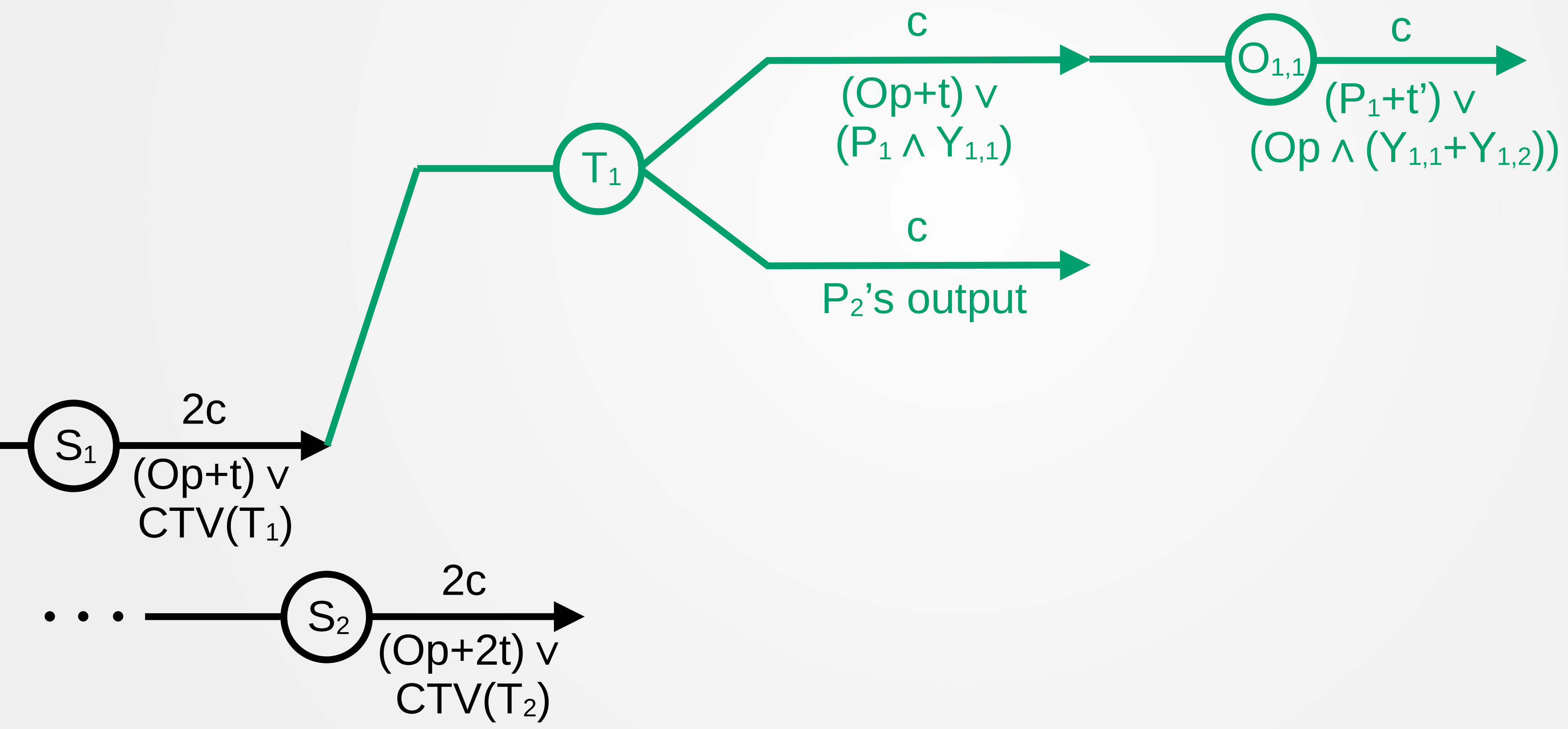
Unilateral exit



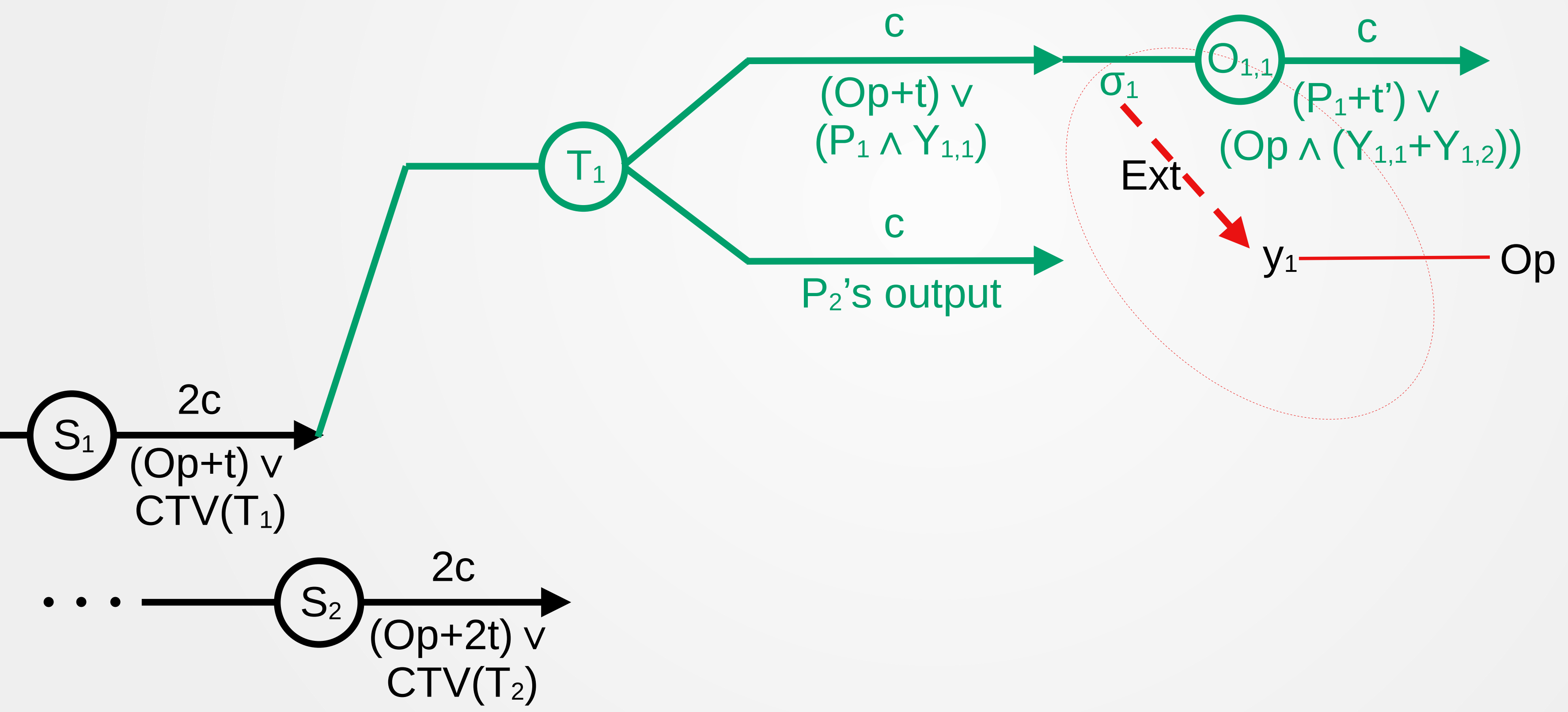
Unilateral exit



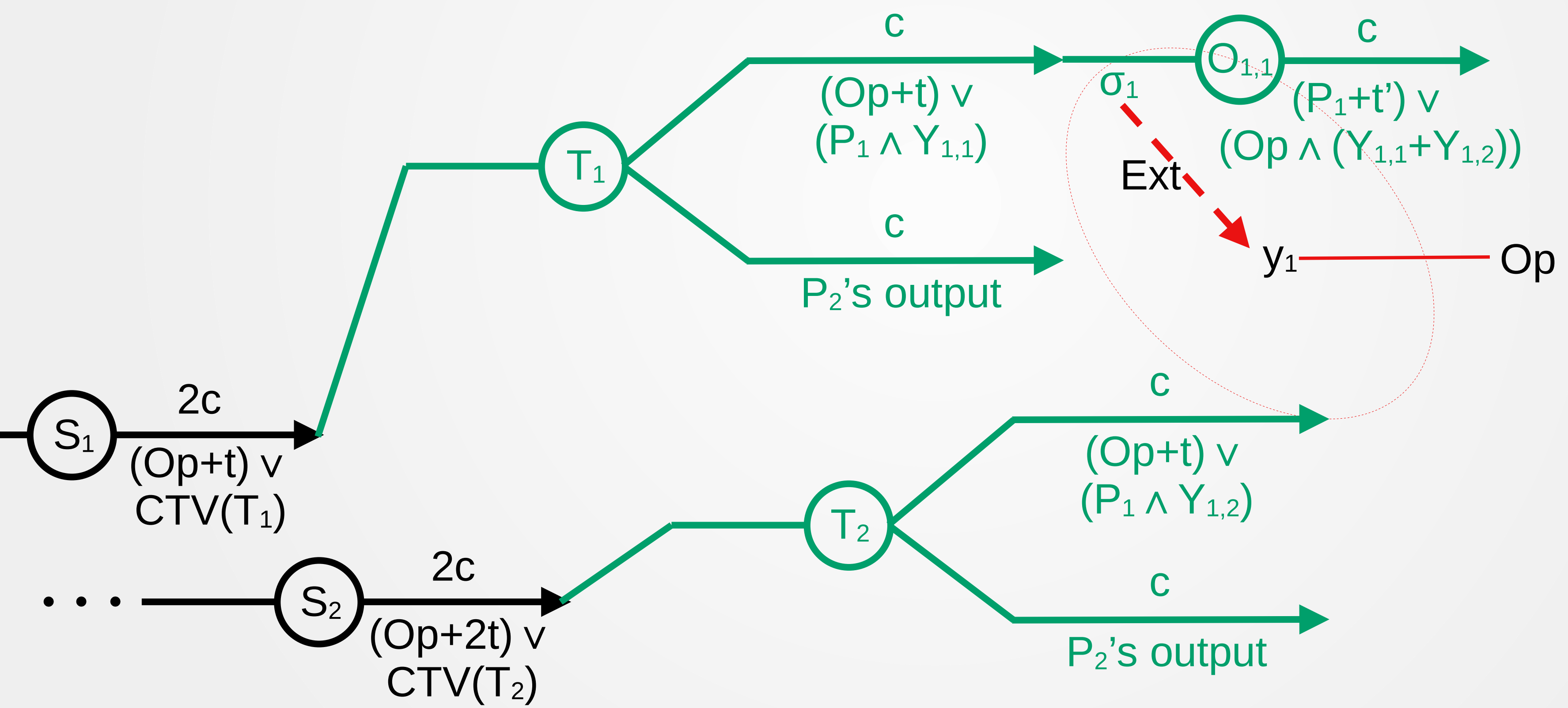
Unilateral exit



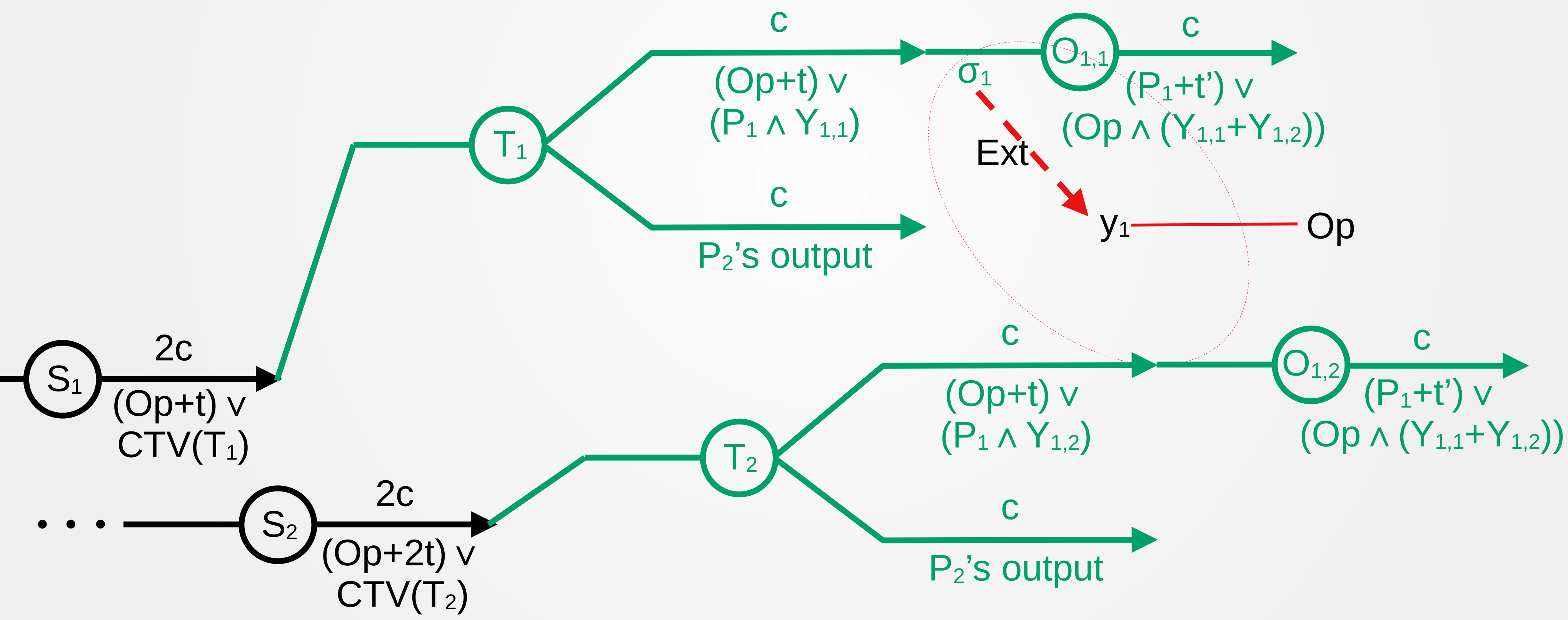
Operator security



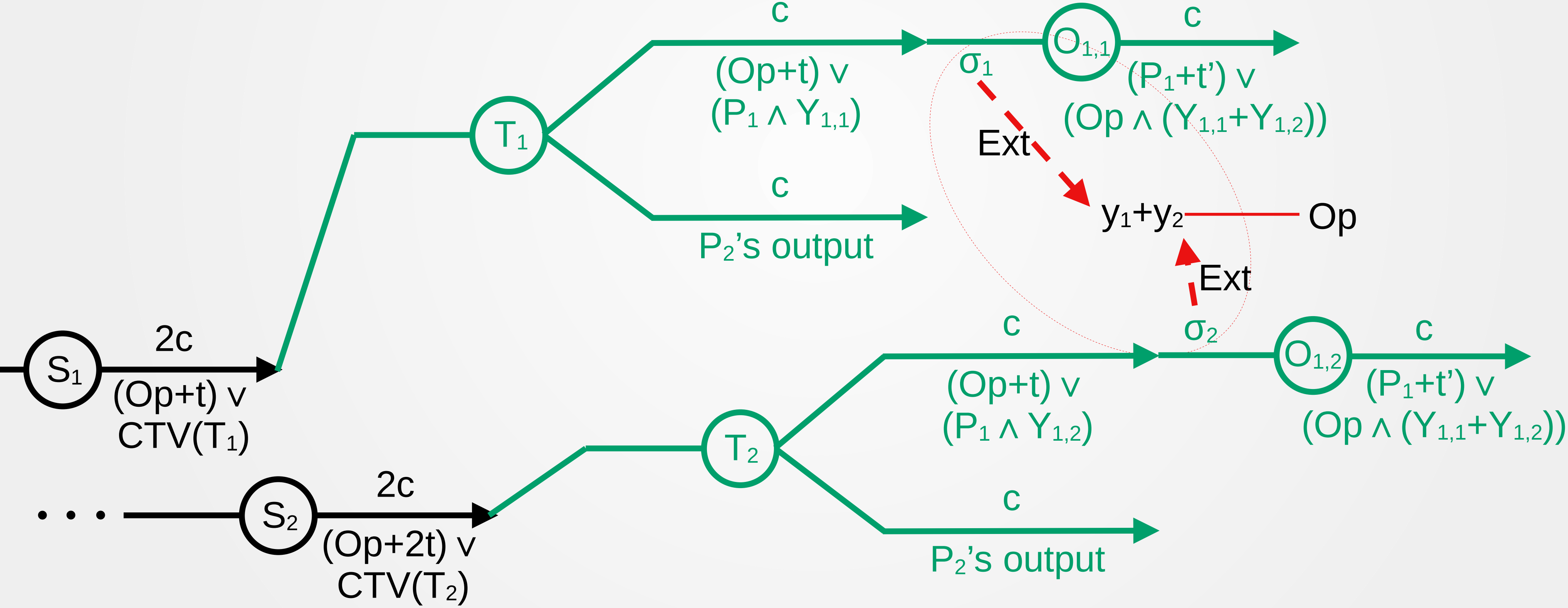
Operator security



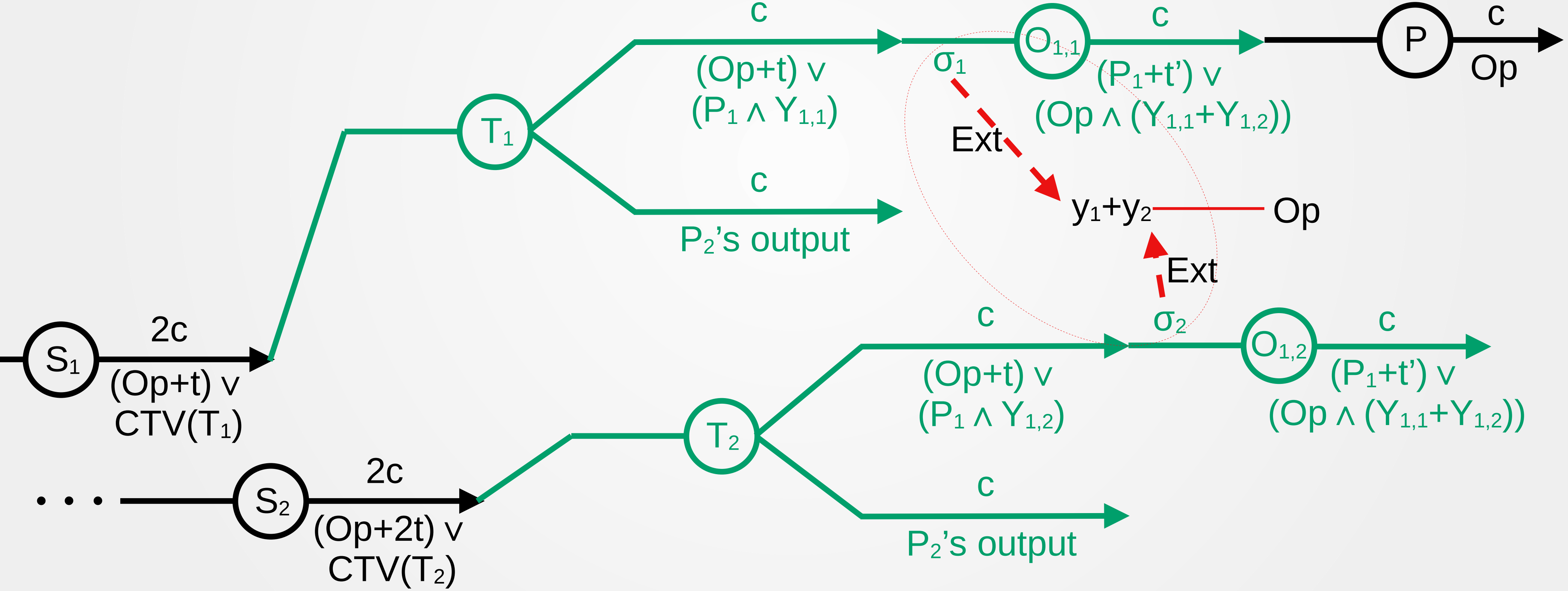
Operator security



Operator security



Operator security



Limitations & Future work

- Need for (untrusted) Operator
- Large collateral by Operator
- Single denomination
- No privacy

Goals achieved ✅

- Pay anyone in the Clique: Output transfers
- Free to leave unilaterally: CTV tree of TXs
- Known max time to finality: 2t+slack
- Users need no extra trust: Operator Byzantine
- Few on-chain TXs: Constant in #users, payments

✯ Operator security: Refunded on exit/malicious user

✯ Graceful recovery after closing starts

Goals achieved ✅

- Pay anyone in the Clique: Output transfers
- Free to leave unilaterally: CTV tree of TXs
- Known max time to finality: 2t+slack
- Users need no extra trust: Operator Byzantine
- Few on-chain TXs: Constant in #users, payments

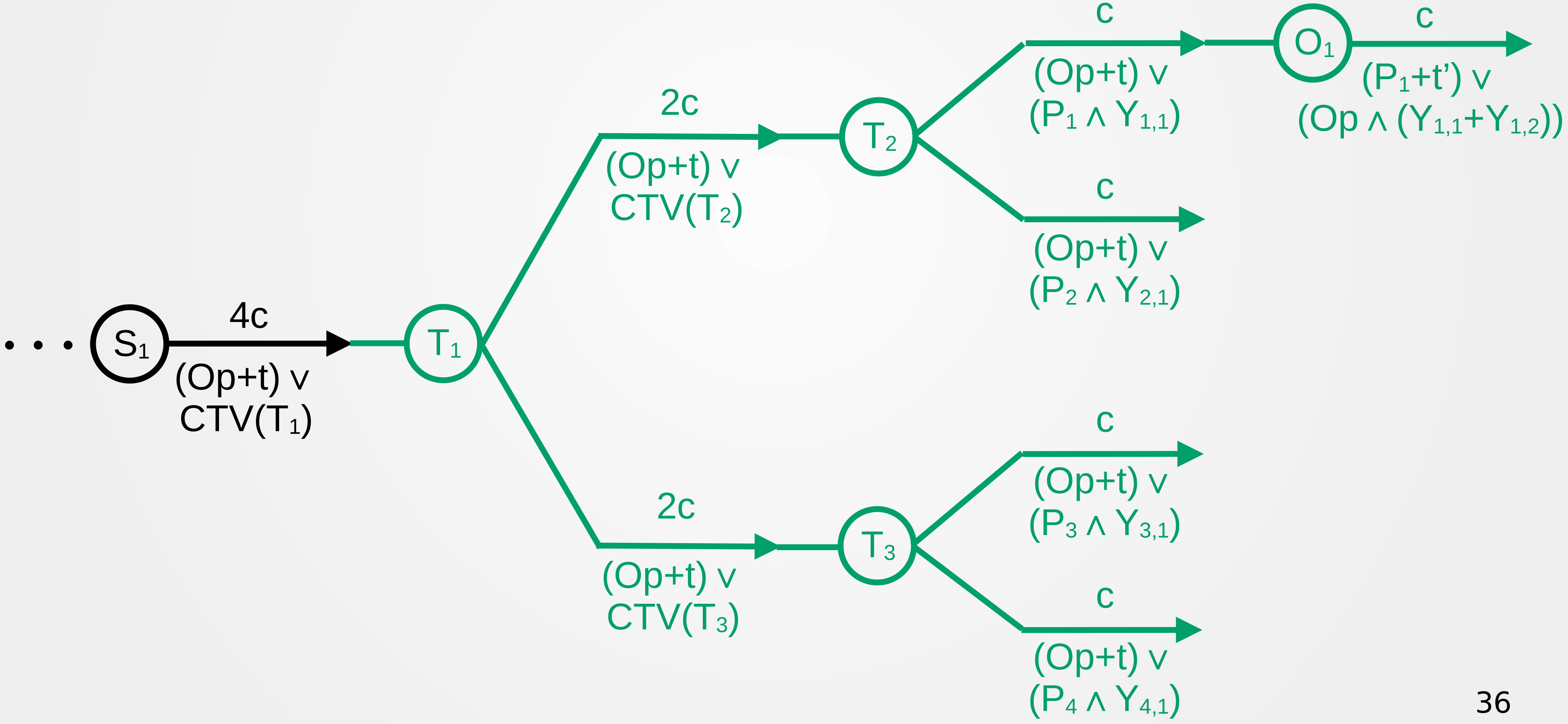
★ Operator security: Refunded on exit/malicious user

★ Graceful recovery after closing starts

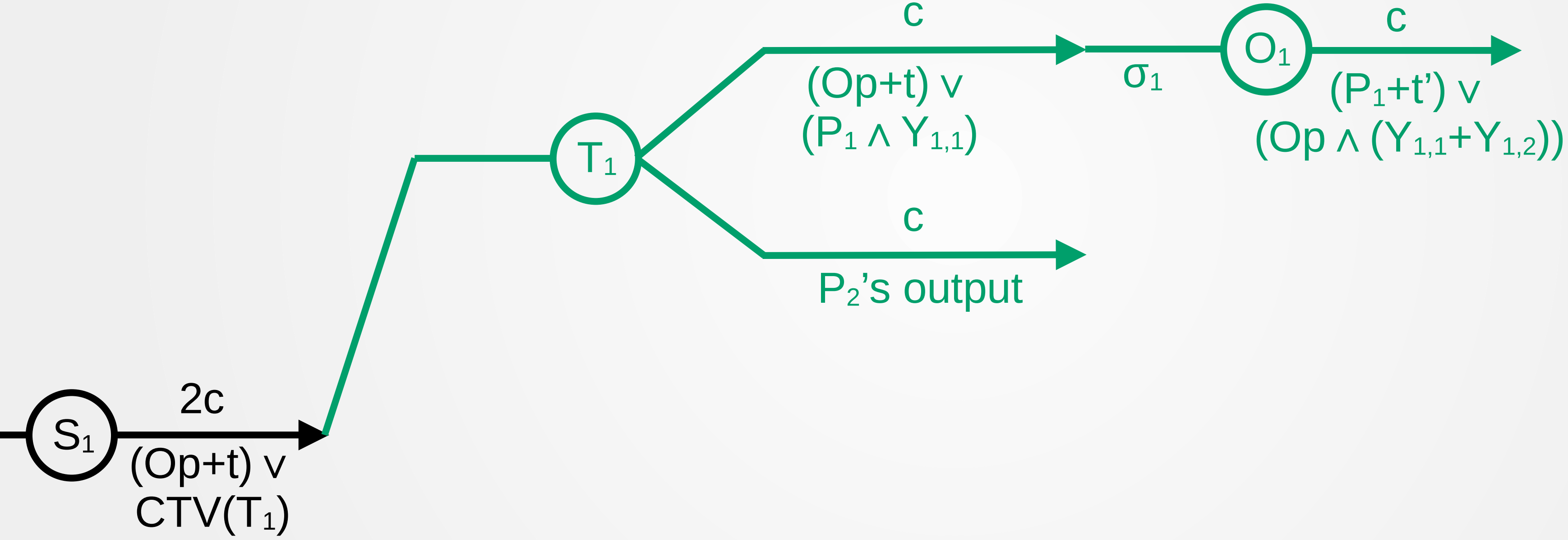
https://eprint.iacr.org/2024/25

Thank you!
Questions?

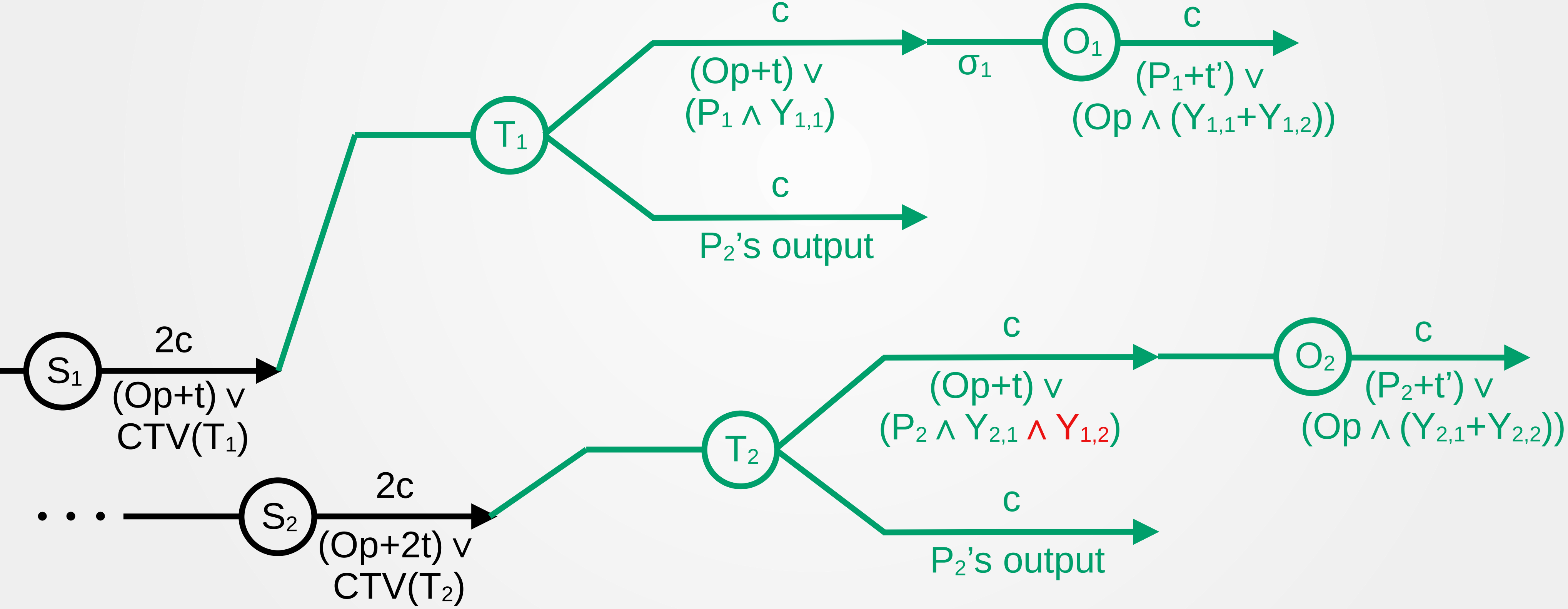
Unilateral exit - CTV tree



P_1 pays P_2


c
O_1
c
σ_1
(Op+t) ∨
$(P_1 \wedge Y_{1,1})$
$(P_1\text{+t'})$ ∨
$(\text{Op} \wedge (Y_{1,1}+Y_{1,2}))$
T_1
c
P_2's output
S_1
2c
(Op+t) ∨
$\text{CTV}(T_1)$

P_1 pays P_2



S_1
2c
(Op+t) v
CTV(T_1)
T_1
c
(Op+t) v
(P_1 ∧ $Y_{1,1}$)
σ_1
O_1
c
(P_1+t’) v
(Op ∧ ($Y_{1,1}$+$Y_{1,2}$))
c
P_2’s output
. . .
S_2
2c
(Op+2t) v
CTV(T_2)
T_2
c
(Op+t) v
(P_2 ∧ $Y_{2,1}$ ∧ $Y_{1,2}$)
O_2
c
(P_2+t’) v
(Op ∧ ($Y_{2,1}$+$Y_{2,2}$))
c
P_2’s output

Off-chain storage & actions

- Operator stores everyone’s data
- Operator posts every new payment (keys, etc.)
- Every user calculates every new tree locally
- If new root tx unexpected, users leave unilaterally