Foundations of Layer-2 Blockchain Protocols

Matteo Maffei

Croatia Summer School on Real-World Crypto and Privacy
June 4, 2024

European Research Council
Der Wissenschaftsfonds.
FFG
Christian Doppler Forschungsgesellschaft

~~Foundations of~~ Layer-2 ~~Blockchain~~ Protocols

Why everyone should do research on blockchains 😇

Matteo Maffei

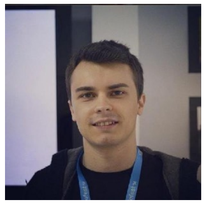
Croatia Summer School on Real-World Crypto and Privacy
June 4, 2024

European Research Council
Der Wissenschaftsfonds.
FFG
Christian Doppler Forschungsgesellschaft

Outline

- Intro to Blockchains, Insights and Challenges
- Layer-2 Protocols for Scalability, Privacy, and more in Bitcoin
- Open Research Directions

Joint Work With...



E. Tairi

L. Aymayr

G. Avarikioti

G. Scaffino

TU WIEN

P. Moreno-Sanchez

institute IMDEA software

A. Kate

PURDUE UNIVERSITY

G. Malavolta

C. Schneidewind

MAX PLANCK INSTITUTE FOR SECURITY AND PRIVACY

S. Mazumdar

भारतीय प्रौद्योगिकी संस्थान इंदौर
IIT Indore
Indian Institute of Technology Indore
ज्ञानम् सर्वजनहिताय

S. Faust

A. Erwig

S. Riahi

TECHNISCHE UNIVERSITÄT DARMSTADT

Sri Aravinda Krishnan Thyagarayan

Carnegie Mellon University

B. Haslofer

COMPLEXITY SCIENCE HUB VIENNA

K. Hostáková

ETH Zürich

Matteo Romiti

AIT AUSTRIAN INSTITUTE OF TECHNOLOGY

Fridhelm Victor

Technische Universität Berlin

P. Nordholt

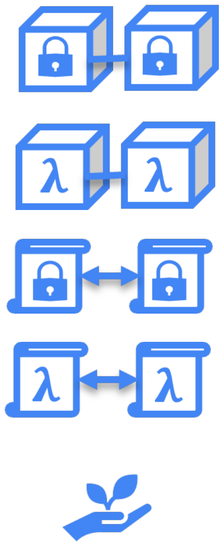
CHAINALYSIS

Blockchain's Evolution

Scientific Innovation

Programmability, Privacy, Scalability, Energy-friendliness,…

- **2008** • Bitcoin (Payments via **Turing-Incomplete Scripting**)
- **2015** • Ethereum (Smart Contracts via **Turing-Complete Scripting**)
- **2016** • Scalability for Bitcoin (Lightning Network)
- **2017** • DeFi Protocols (AMM, Cross-Chain Services)
- **2022** • Ethereum switches to Proof-of-Stake (Energy Friendliness)

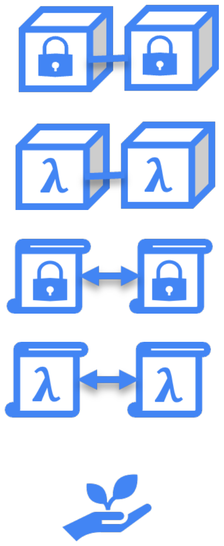


Blockchain's Evolution

Scientific Innovation

Programmability, Privacy, Scalability, Energy-friendliness,…

- 2008 • Bitcoin (Payments via **Turing-Incomplete Scripting**)
- 2015 • Ethereum (Smart Contracts via **Turing-Complete Scripting**)
- 2016 • Scalability for Bitcoin (Lightning Network)
- 2017 • DeFi Protocols (AMM, Cross-Chain Services)
- 2022 • Ethereum switches to Proof-of-Stake (Energy Friendliness)



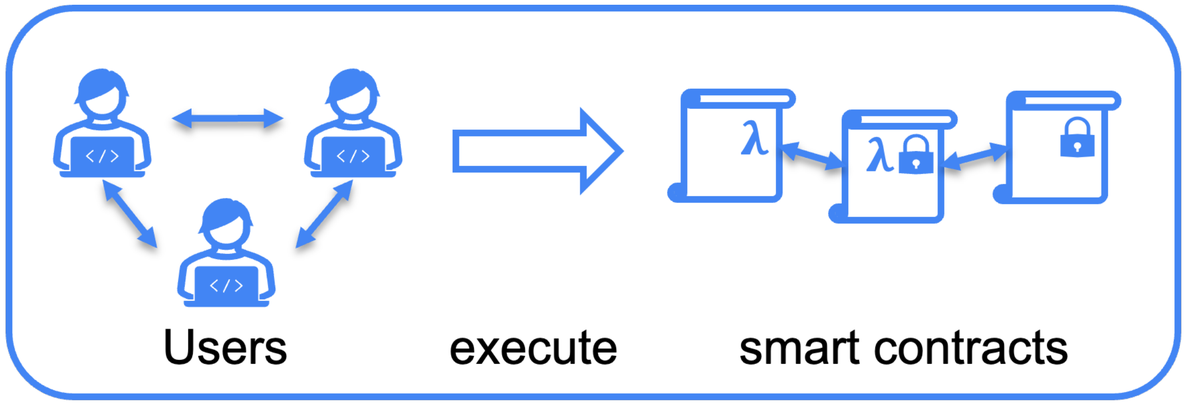
Societal Impact

Decentralized, censorship-resistant, instantaneous, wealth-storing finance

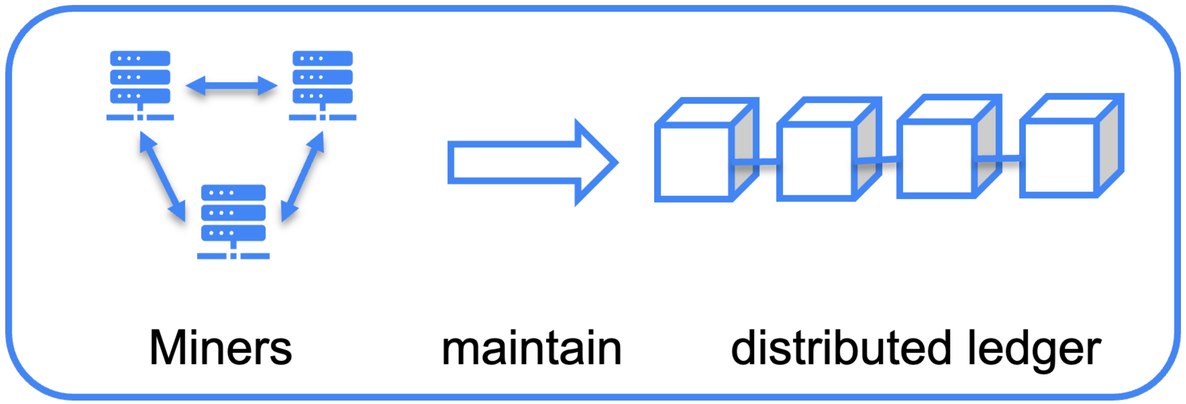
- 2017 • Amex and Visa integrate blockchain technologies
- 2020 • Cryptocurrencies accepted at Starbucks, Microsoft, Paypal etc.
- 2021 • Bitcoin legal tender in El Salvador
- 2022 • UNHCR embraces stable coins for donations in war zones
- 2023 • Digital Euro (2-year testing phase)
- 2024 • Argentina admits cryptocurrencies for contract settlement

Blockchain Architecture

Layer-2
(Application)

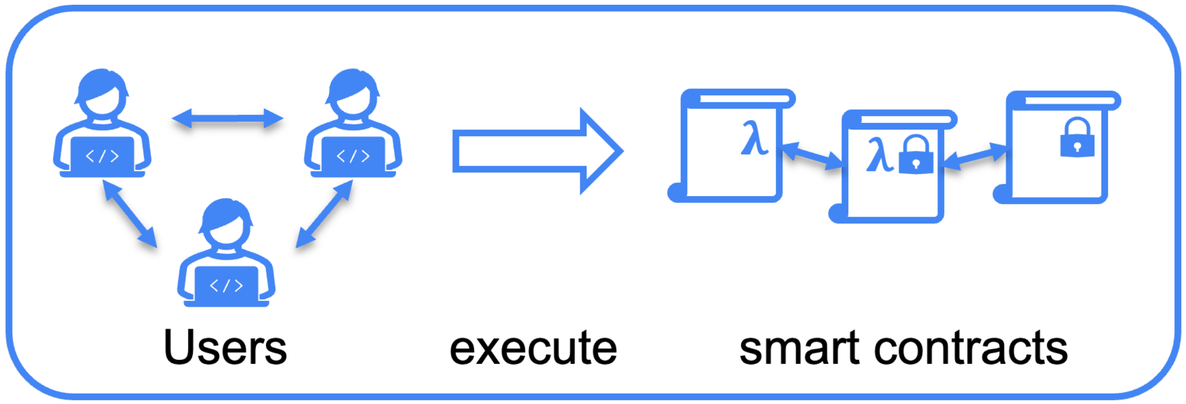


Layer-1
(Consensus)

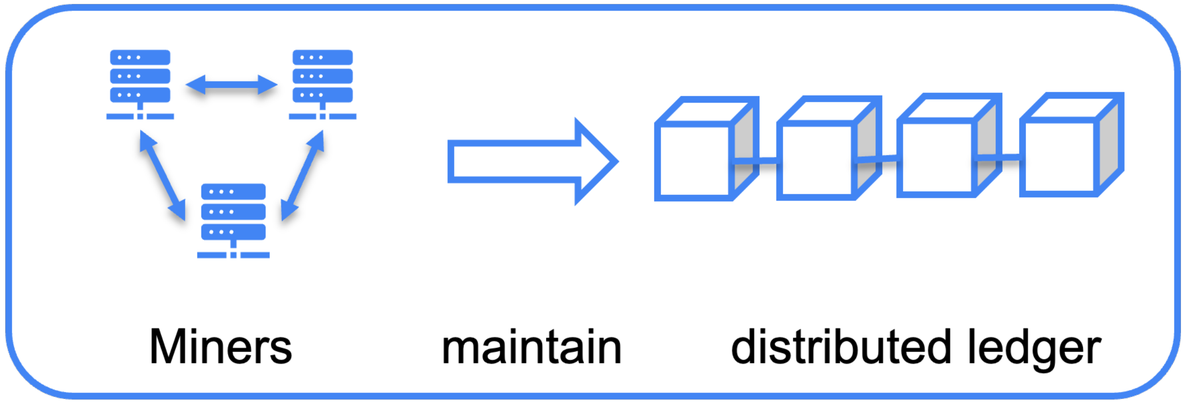


Blockchain Architecture

Layer-2
(Application)

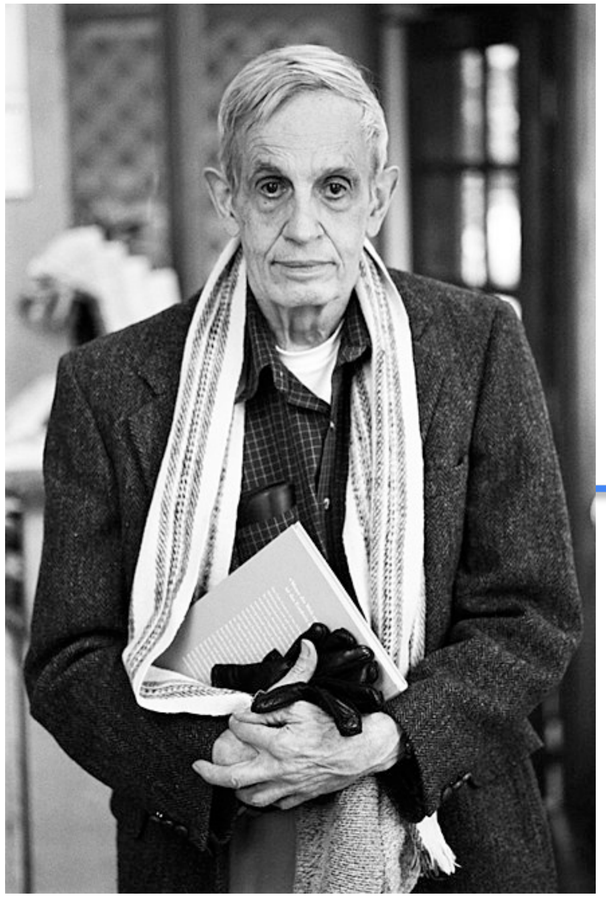


Layer-1
(Consensus)

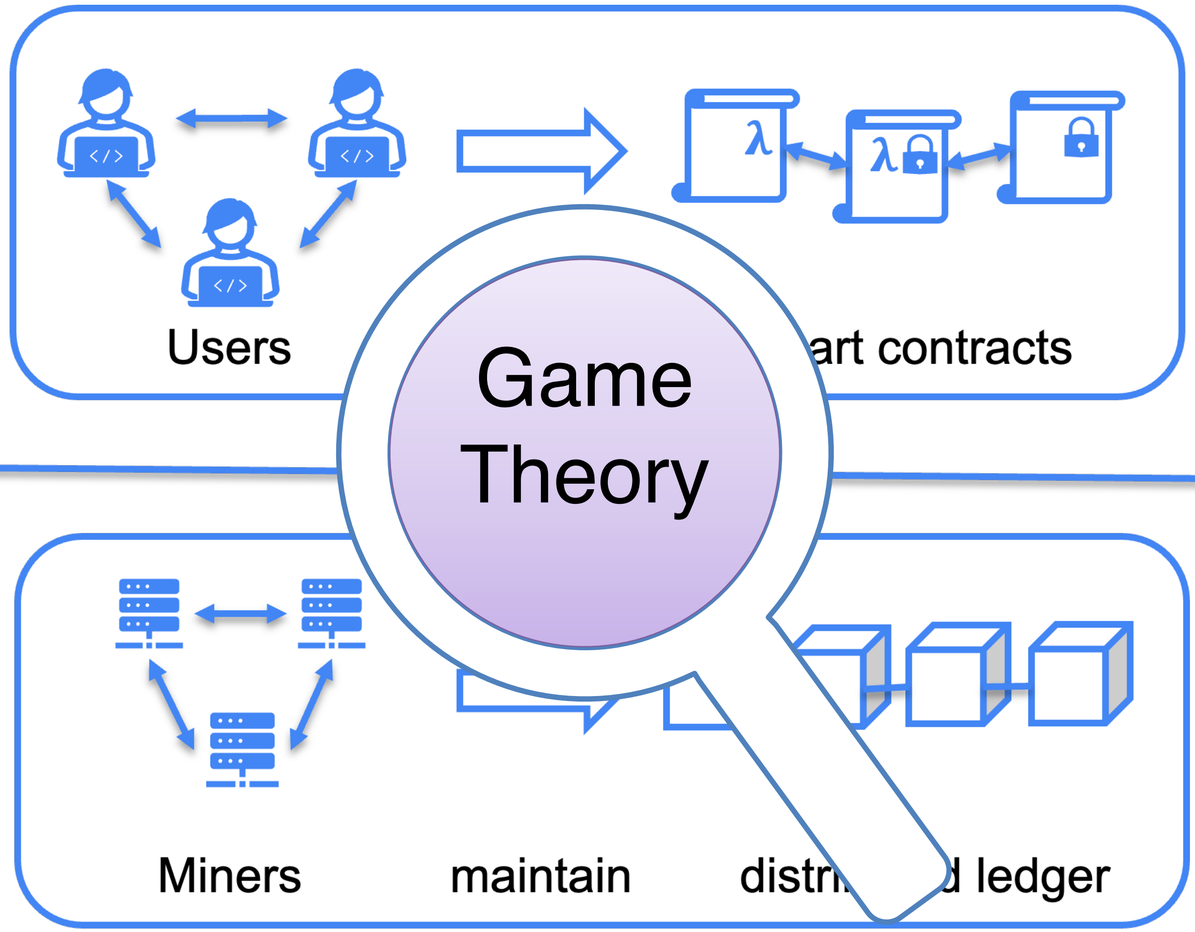


The reason why all of that works goes beyond standard cryptography, distributed system, and secure programming results…

Blockchain Architecture



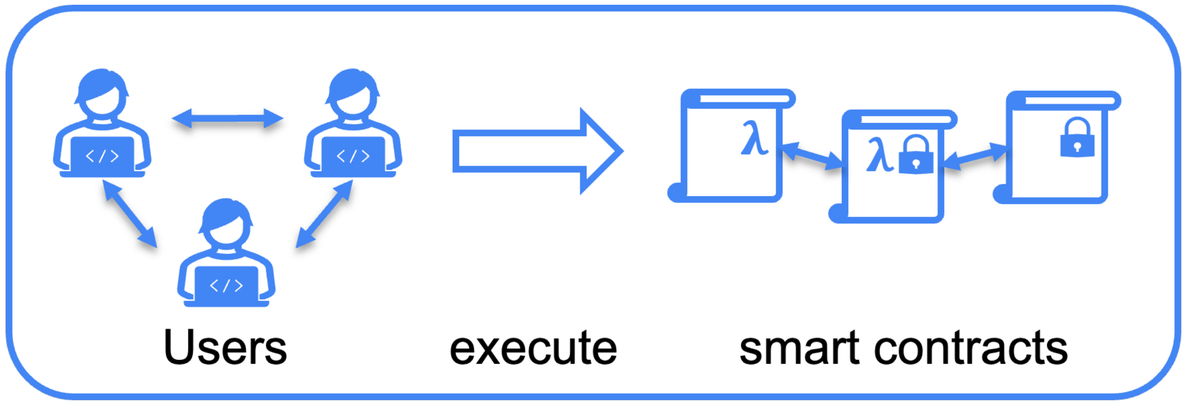
John Nash



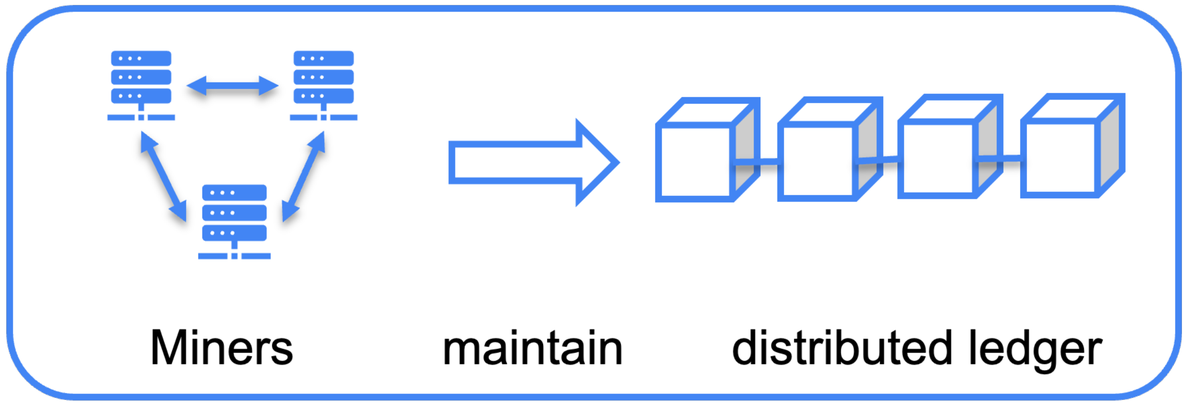
The reason why all of that works goes beyond standard cryptography, distributed system, and secure programming results…

Blockchain Architecture

Layer-2
(Application)



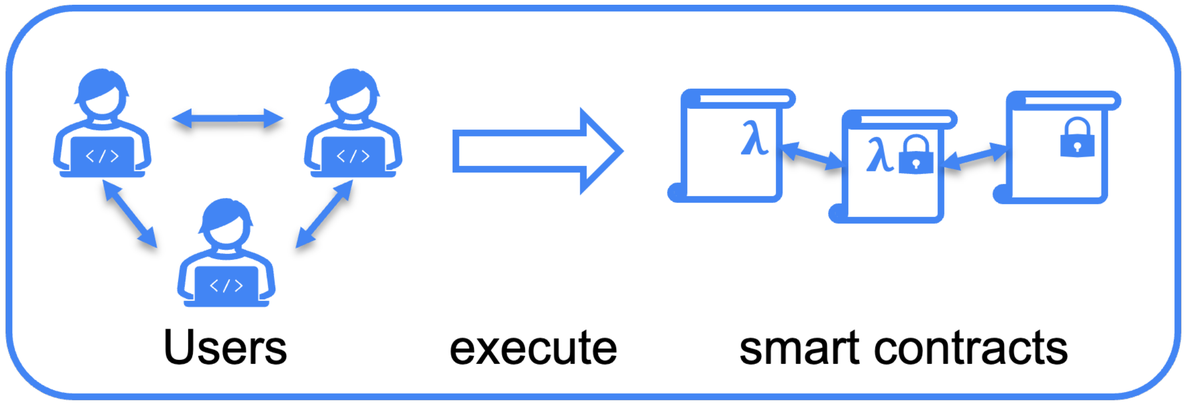
Layer-1
(Consensus)



Scalability issue: each transaction has to be stored and processed

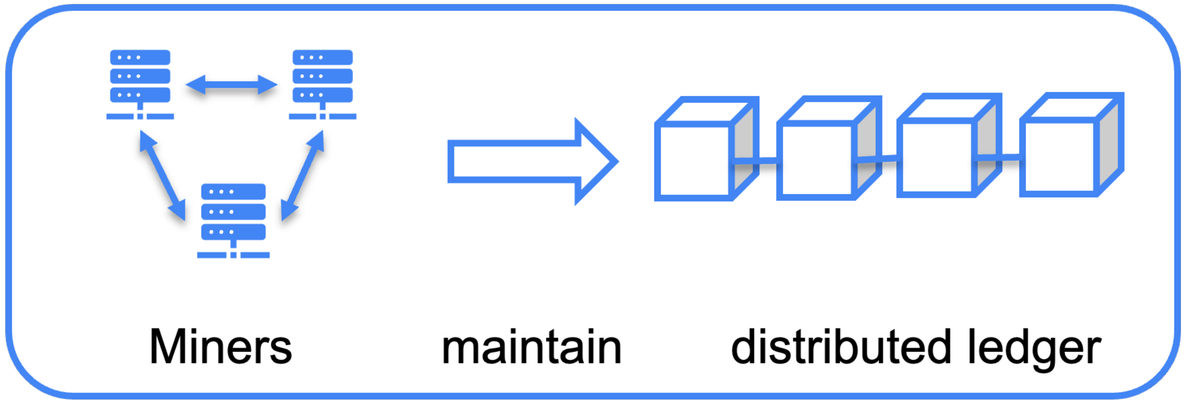
Blockchain Architecture

Layer-2
(Application)



Security issue:
fast deployment
prioritized over
solid foundations

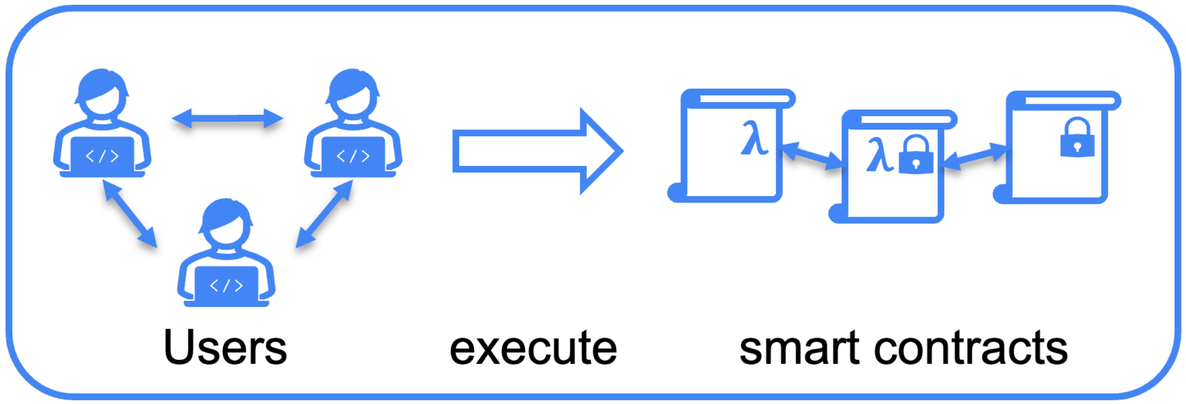
Layer-1
(Consensus)



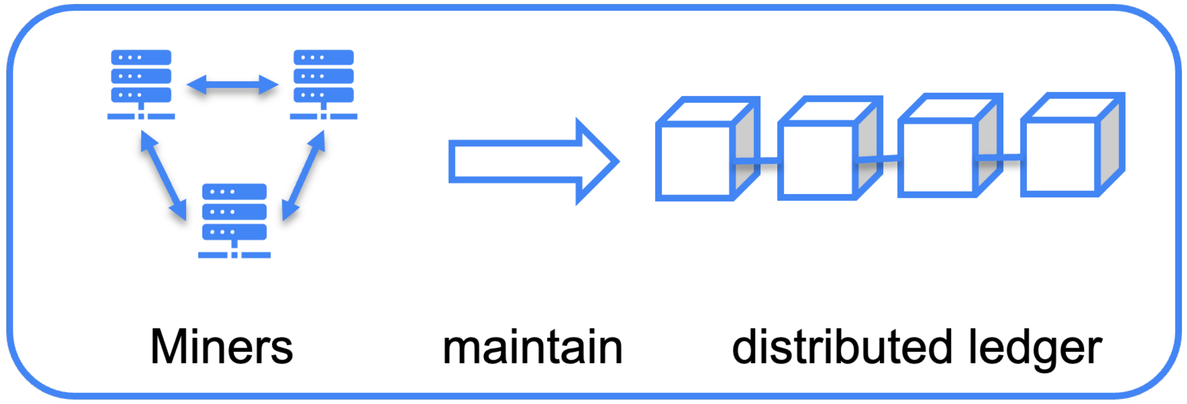
Scalability issue:
each transaction
has to be stored
and processed

Blockchain Architecture

Security issue: incentives not always aligned across layers (applications may break consensus!)



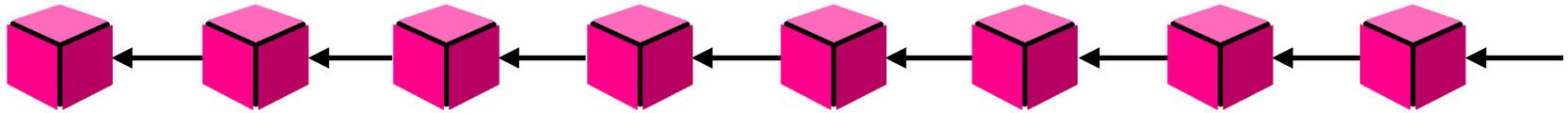
Security issue: fast deployment prioritized over solid foundations



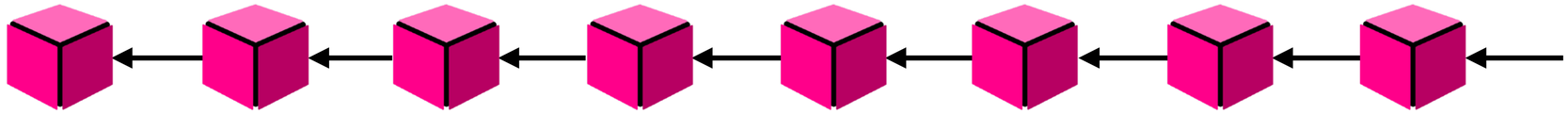
Scalability issue: each transaction has to be stored and processed

Layer-2 Protocols for Bitcoin

Scalability Issue

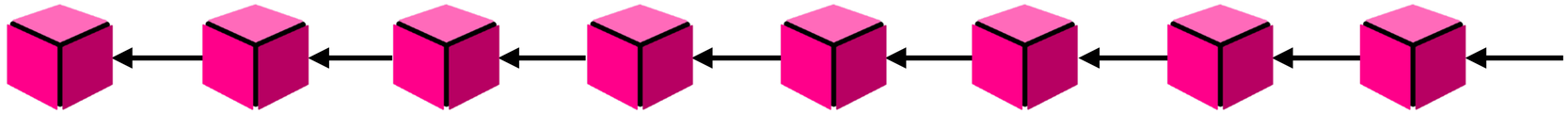


Scalability Issue



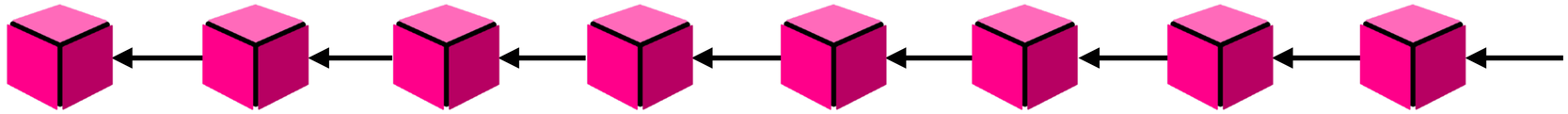
- Blockchain records every transaction

Scalability Issue



- Blockchain records every transaction
- Everyone has to check the whole blockchain

Scalability Issue



- Blockchain records every transaction
- Everyone has to check the whole blockchain

Bitcoin's **transaction rate:** ~10 tx/sec
Visa's transaction rate: ~10K tx/sec

Scalability

- **On-chain, consensus layer**

  e.g., DAG Blockchain, sharding, ...

- **Off-chain, application layer**

  e.g., Payment Channel Networks, Rollups

Scalability

- **On-chain, consensus layer**

  e.g., DAG Blockchain, sharding, ...

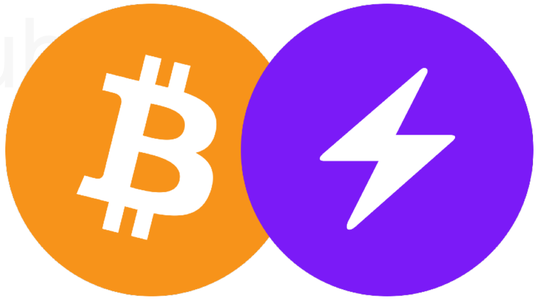
- **Off-chain, application layer**

  e.g., Payment Channel Networks, Rollups

Scalability

- **On-chain, consensus layer**
  e.g., DAG Blockchain, sharding, ...

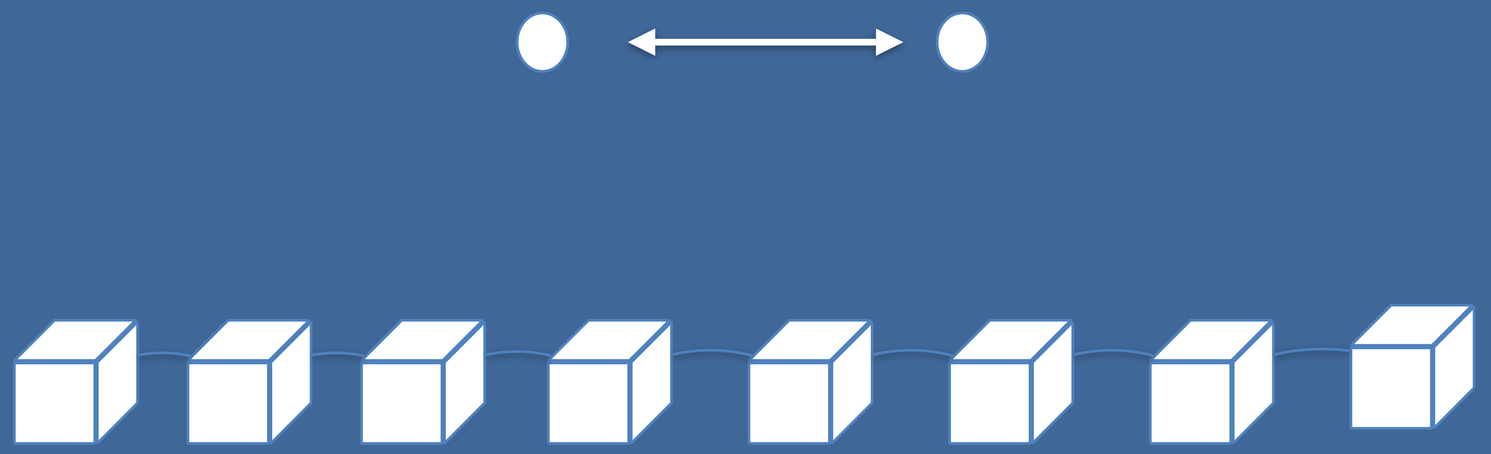
- **Off-chain, application layer**
  e.g., Payment Channel Networks, Rollups



Lightning Network
(300M $ total value locked)

Exchange transactions locally off-chain, blockchain only for disputes

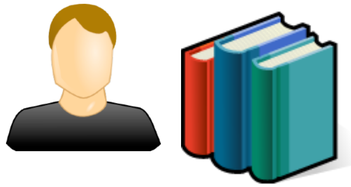
Payment Channels



Two nodes transact with each other without using the blockchain

Payment Channels

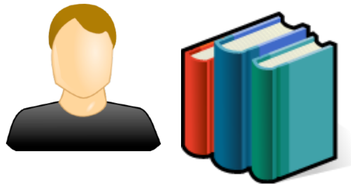
Alice



Bob

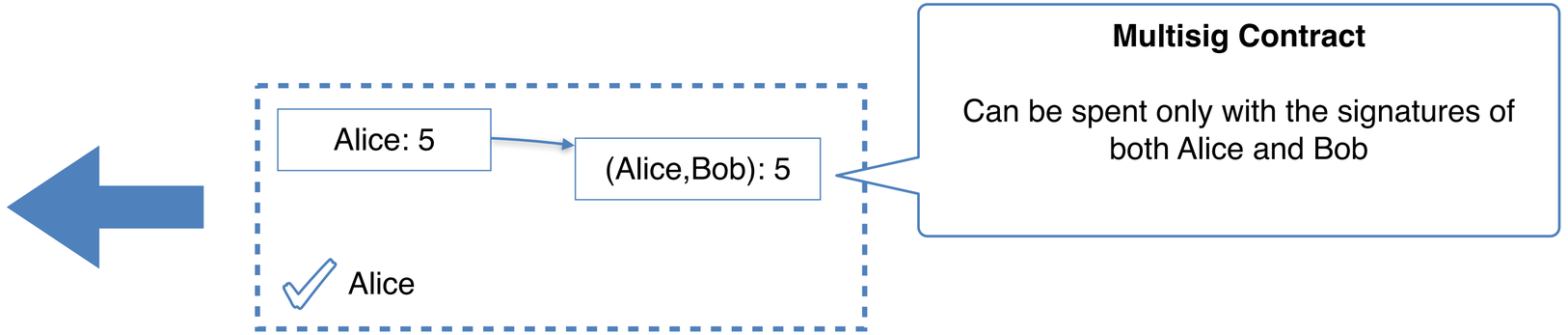
Payment Channels: Open

Alice

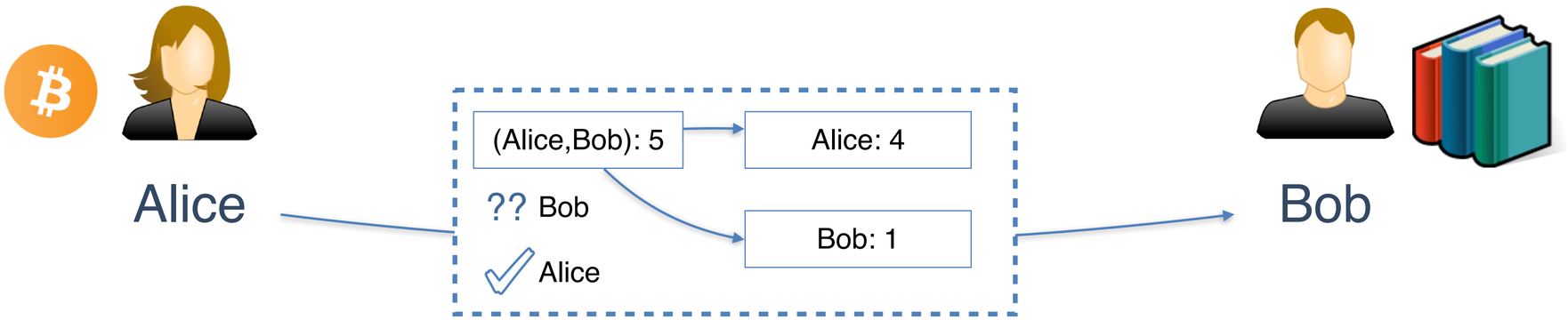


Bob

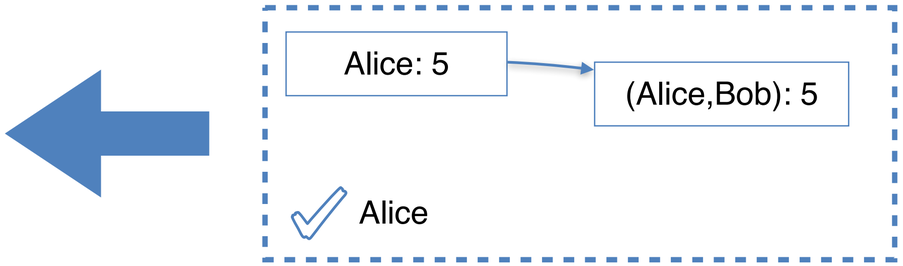
Blockchain



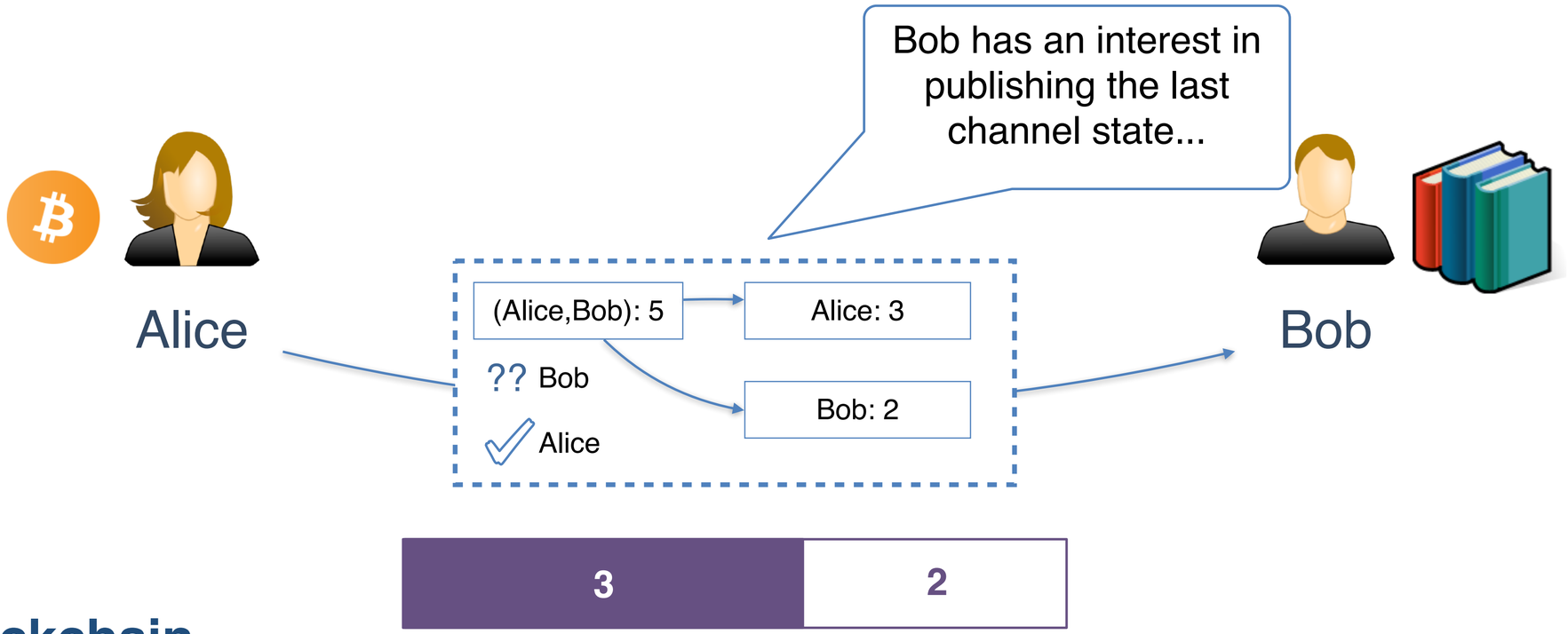
Payment Channels: One-Way Transactions



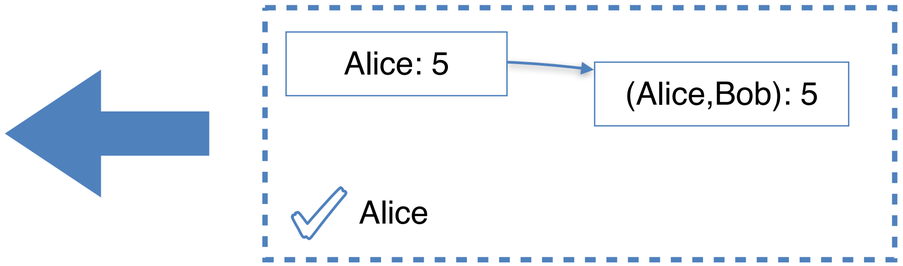
Blockchain



Payment Channels: One-Way Transactions

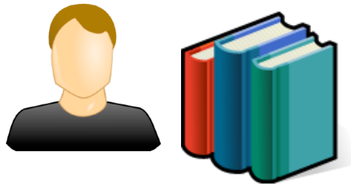


Blockchain



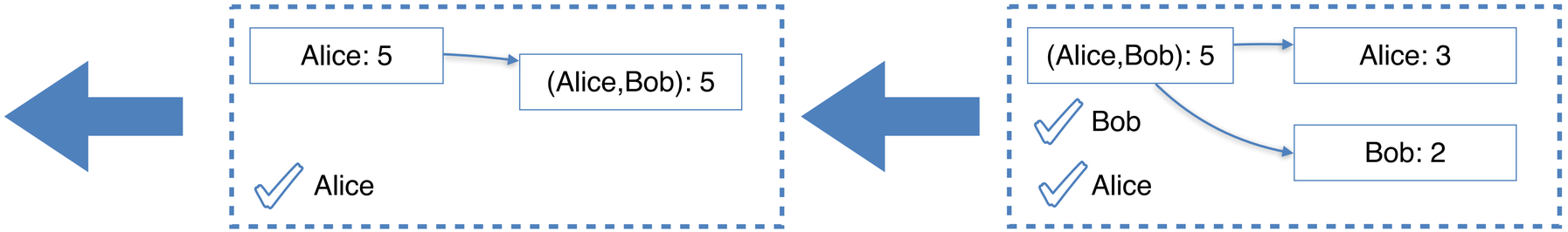
Payment Channels: Closure

Alice



Bob

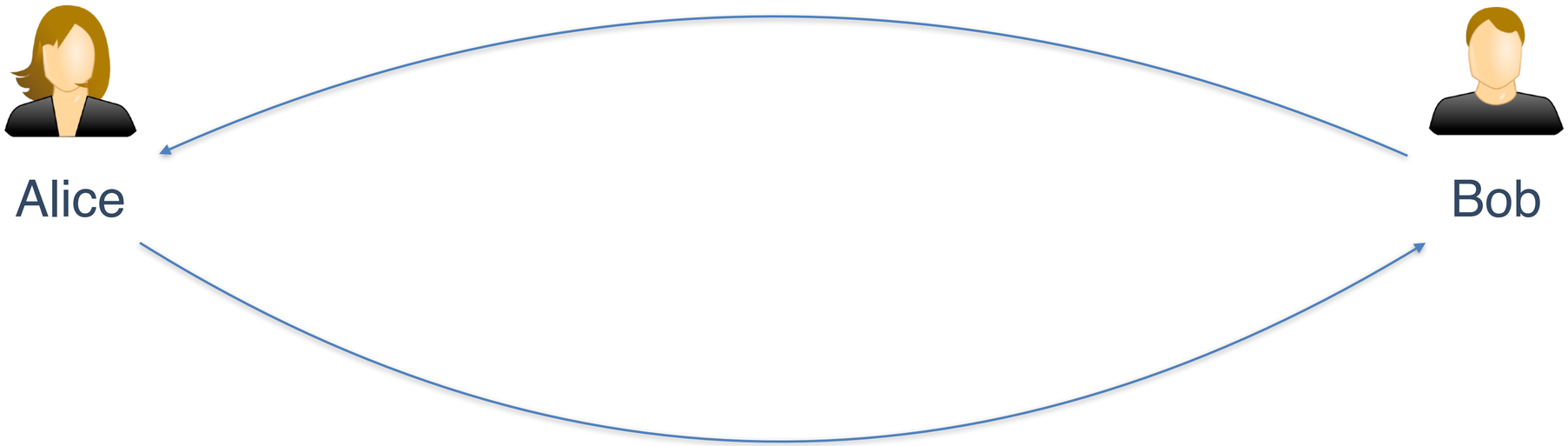
Blockchain



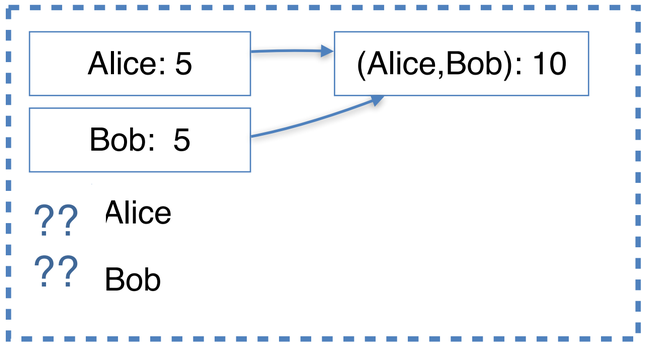
Two Problems

- *What if Bob stops communicating? Alice would lose the money she locked in the channel*
  - We need a way to prevent DOS attacks
- *What if some intermediate state is more advantageous for Bob? He could publish an old channel state*
  - We need a way to prevent channel unrolling attacks...

Payment Channels: First Transaction

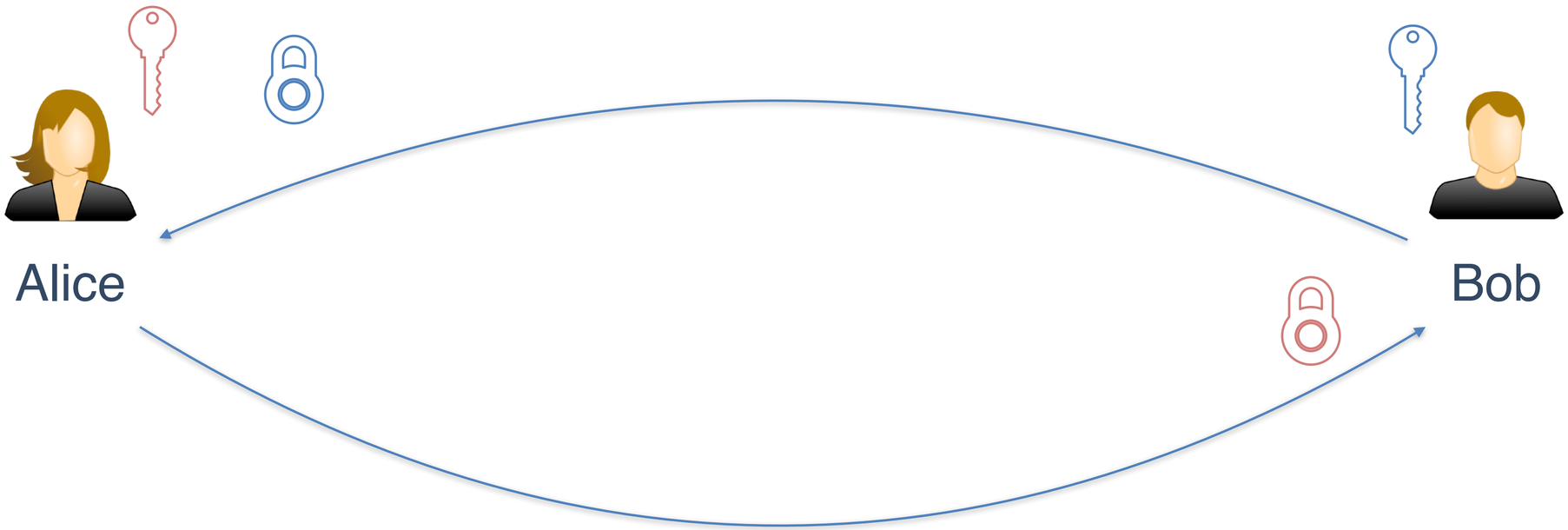


Blockchain

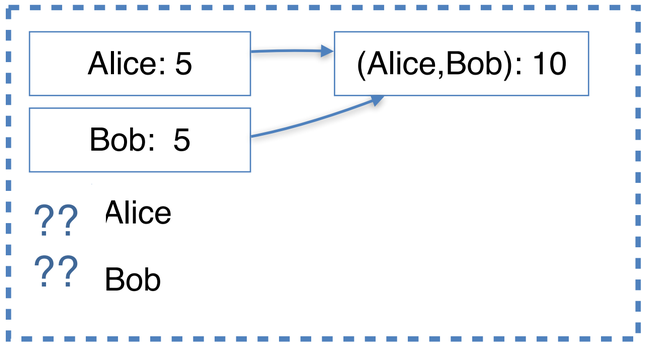


Step 1:
Create Open Transaction (Off-Chain)

Payment Channels: First Transaction



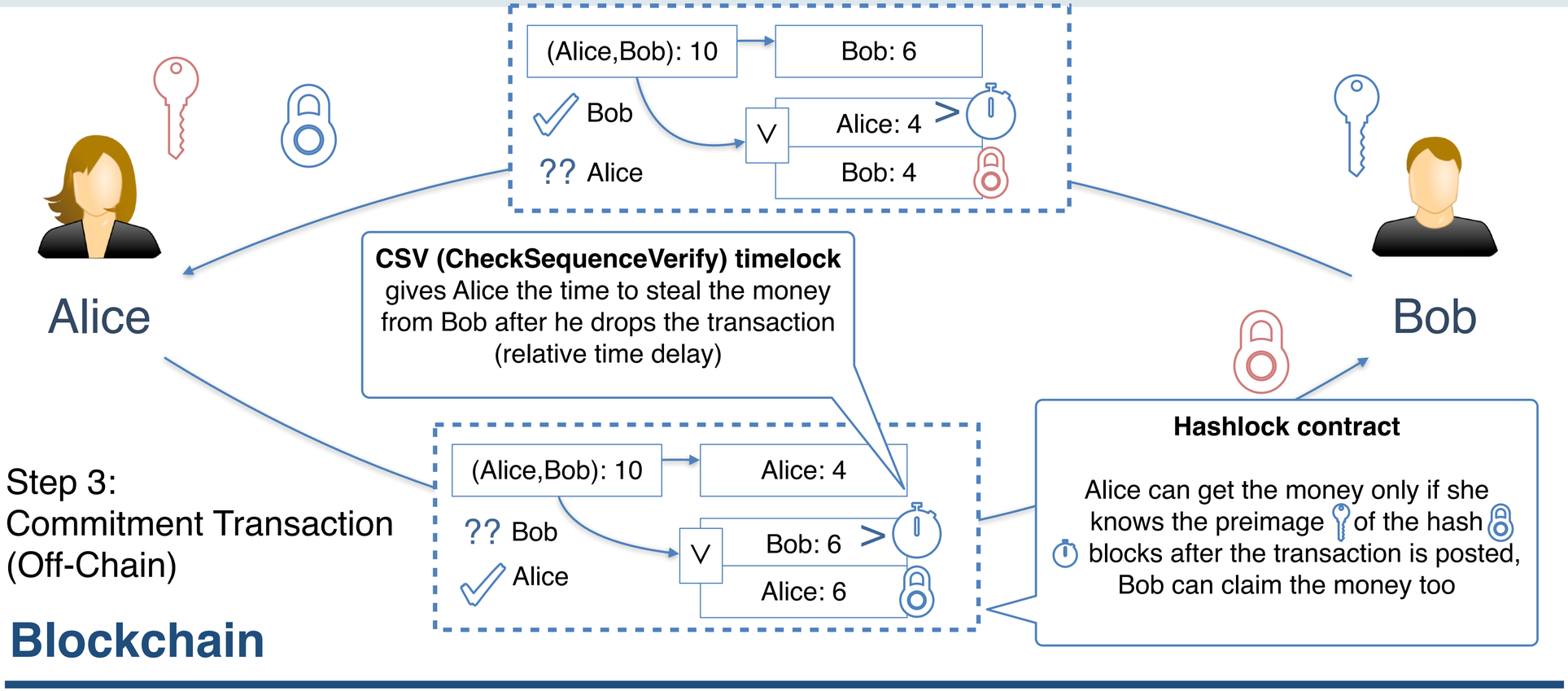
Blockchain

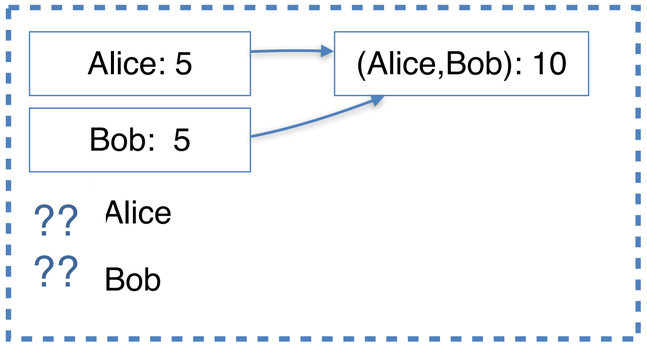


Step 1:
Create Open Transaction (Off-Chain)

Step 2:
Create secrets and exchange respective hashes

Payment Channels: First Transaction

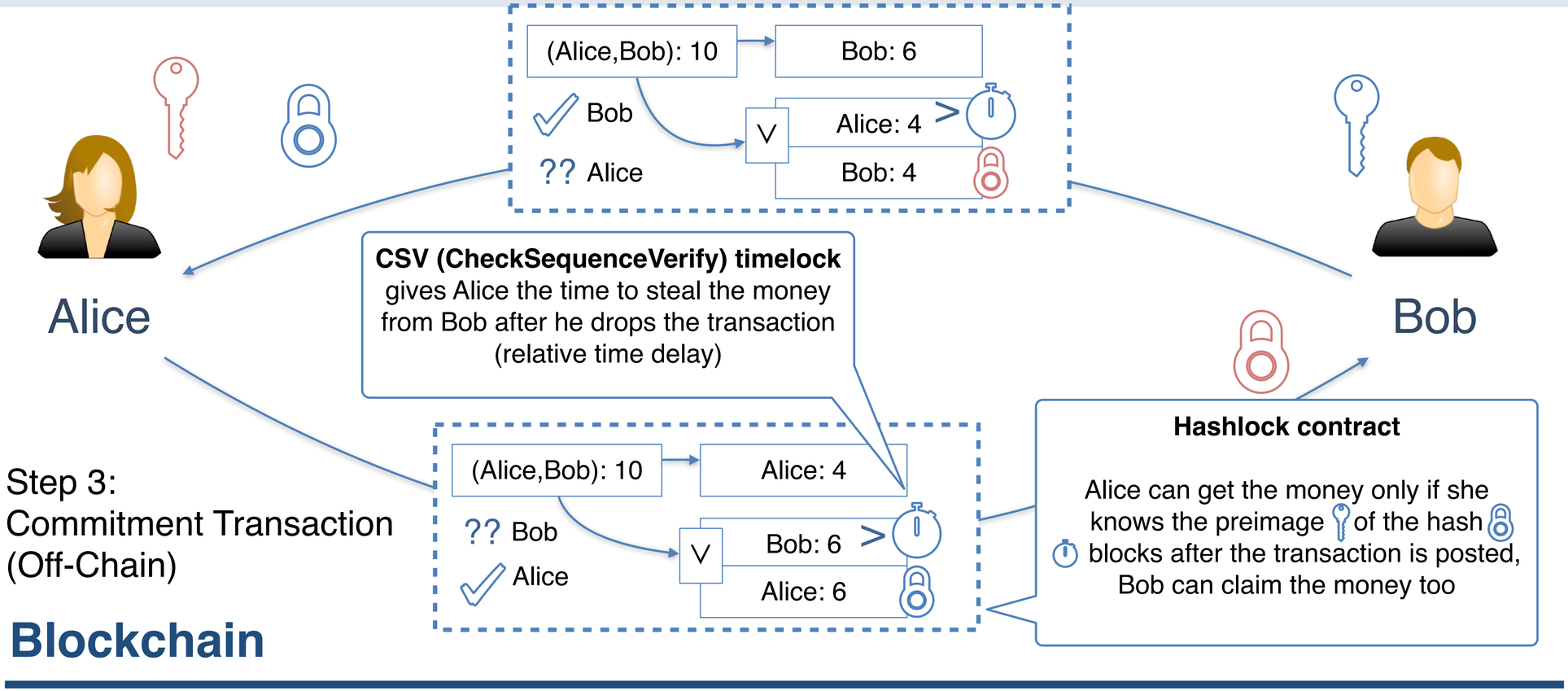


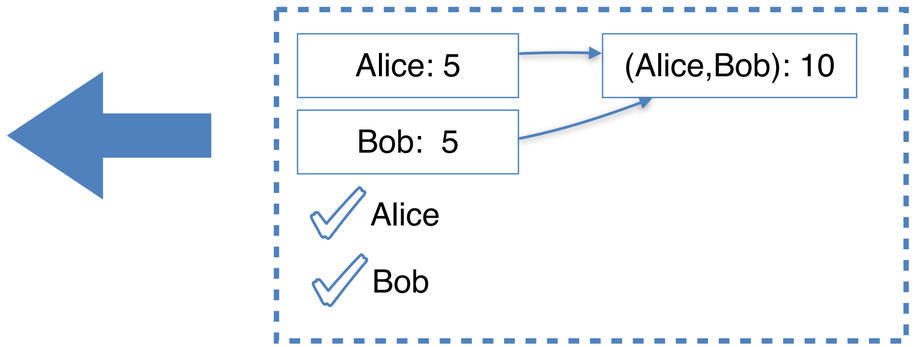


Step 1:
Create Open Transaction (Off-Chain)

Step 2:
Create secrets and exchange respective hashes

Payment Channels: First Transaction



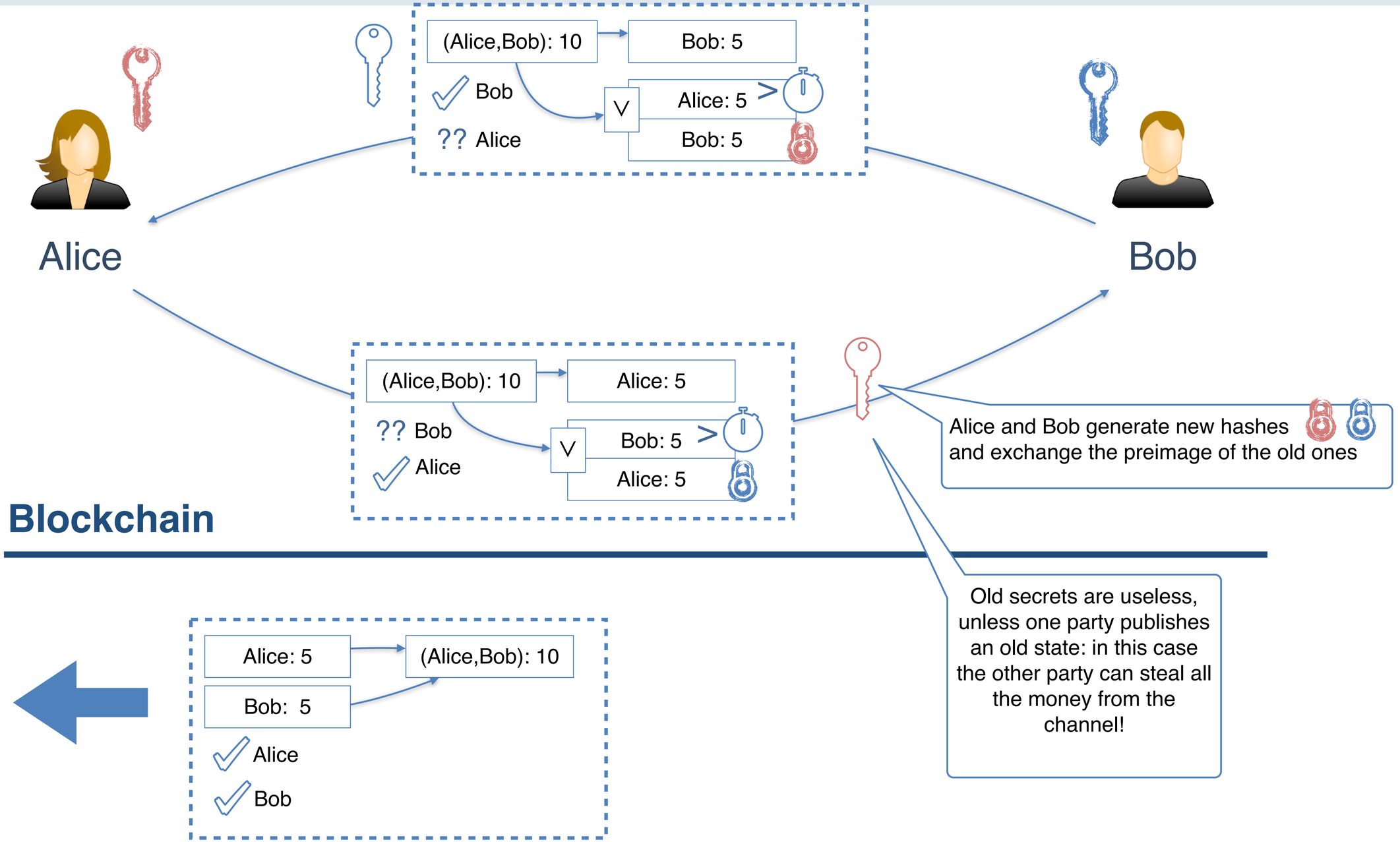


Step 1:
Create Open Transaction (Off-Chain)

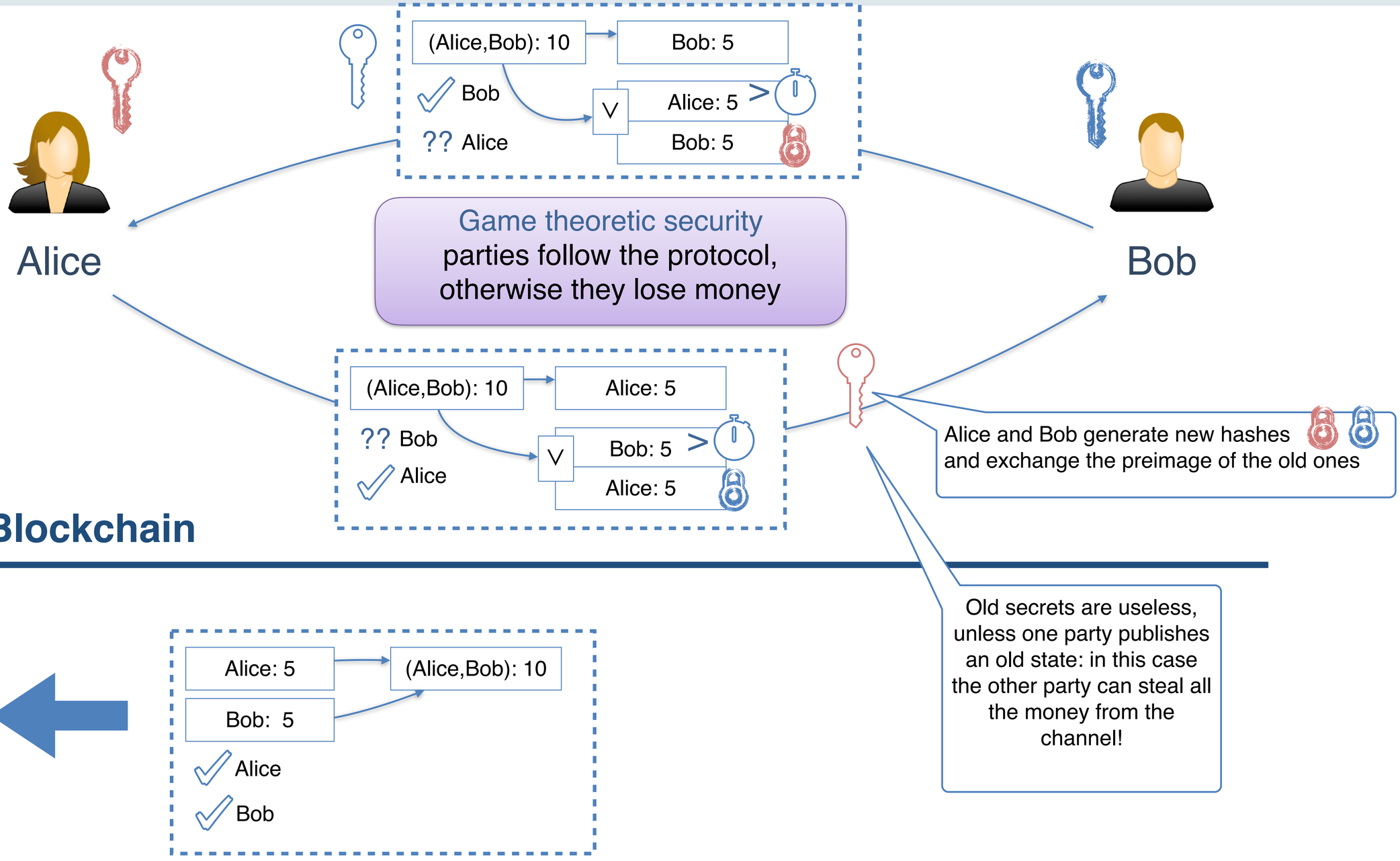
Step 2:
Create secrets and exchange respective hashes

Step 4:
Sign and Push Open Transaction (On-Chain)

Payment Channels: State Change



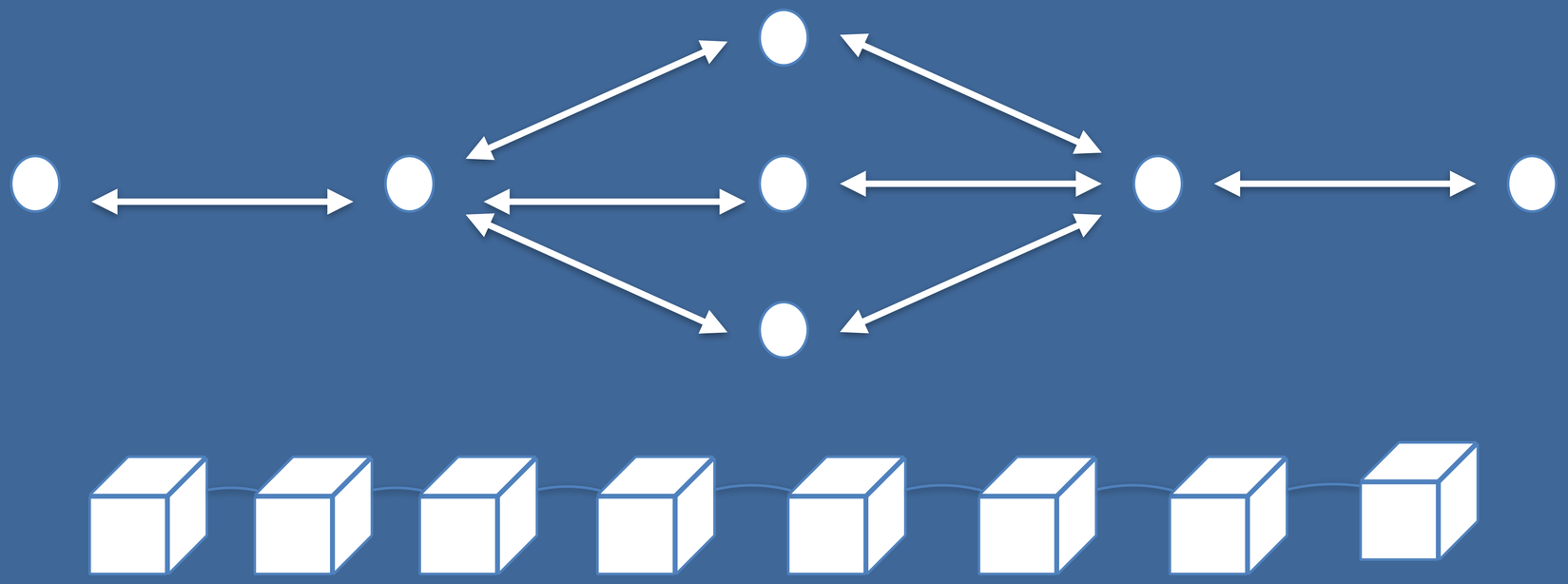
Payment Channels: State Change



Take Home

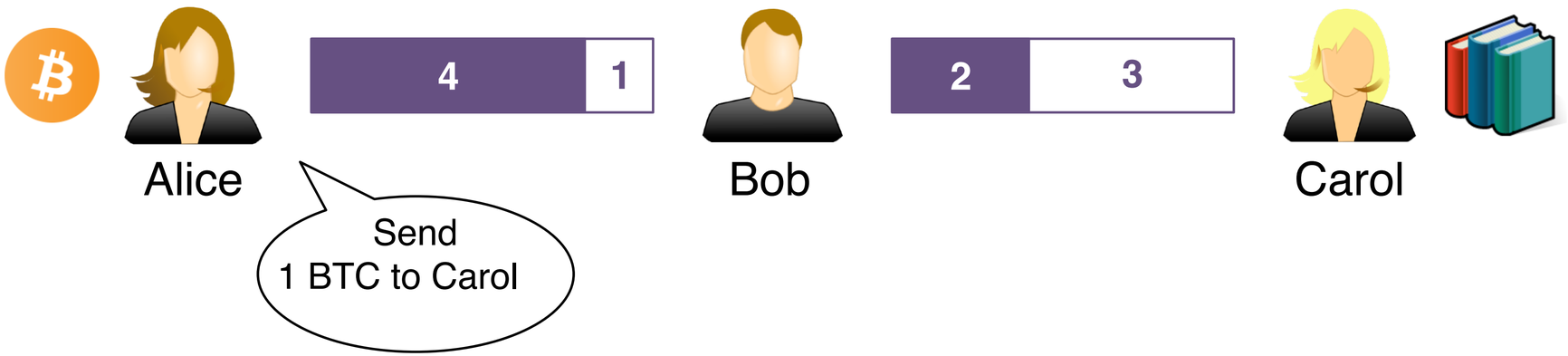
- Arbitrarily many payments with just two messages on-chain (opening and closure) 🙂
- One cannot open a channel with everyone, too expensive (fees plus locked coins) ☹️

Payment Channel Networks

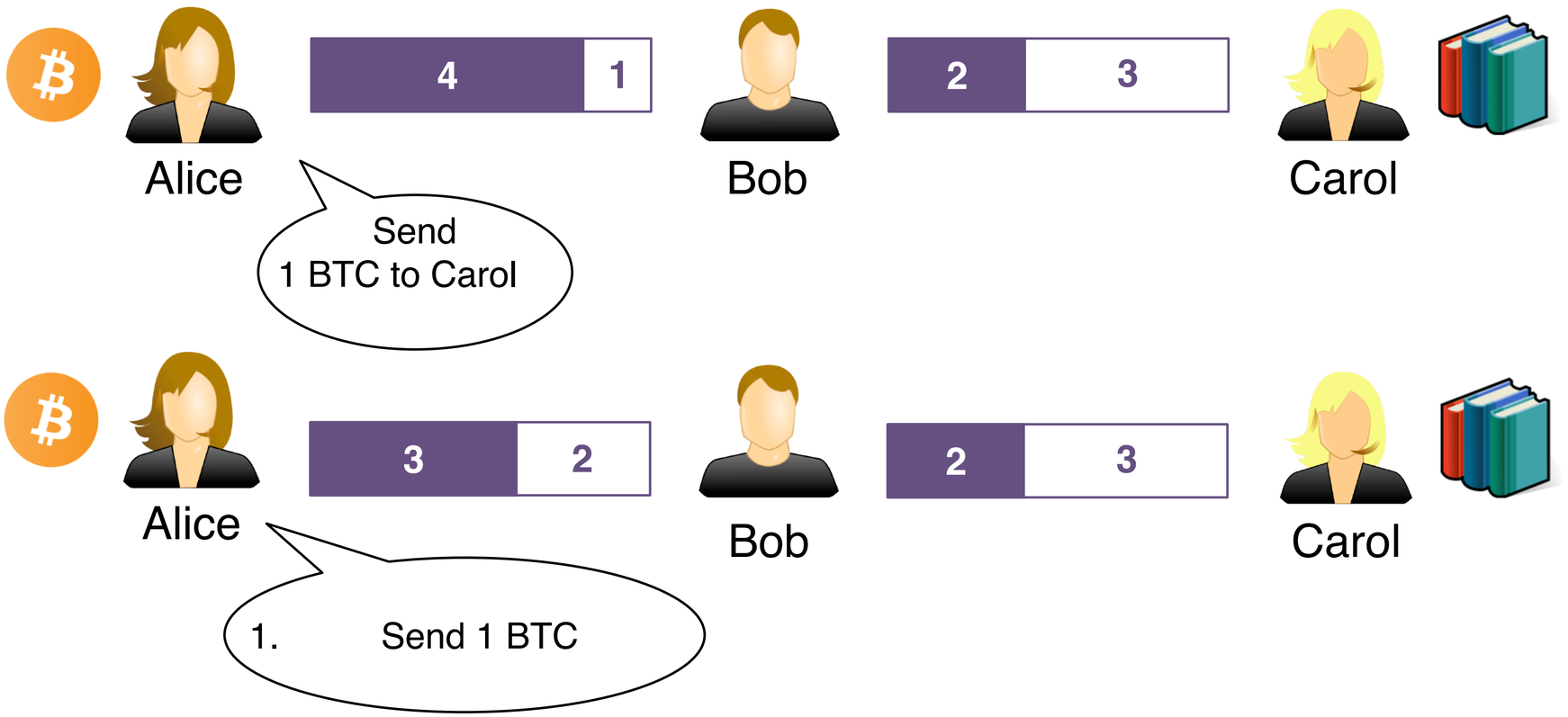


Create a network and perform multi-hop transactions

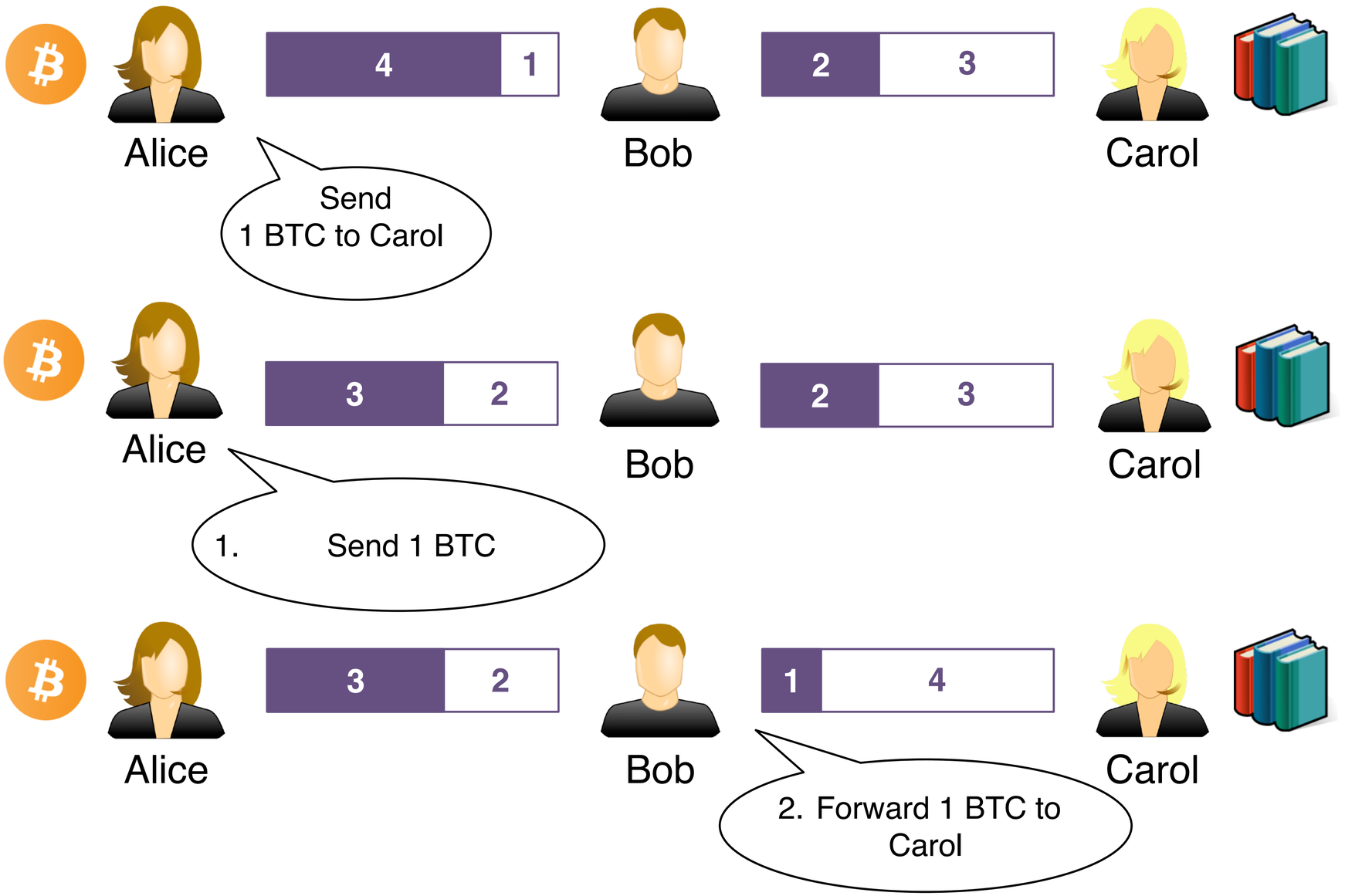
Payment Channel Networks (PCNs)



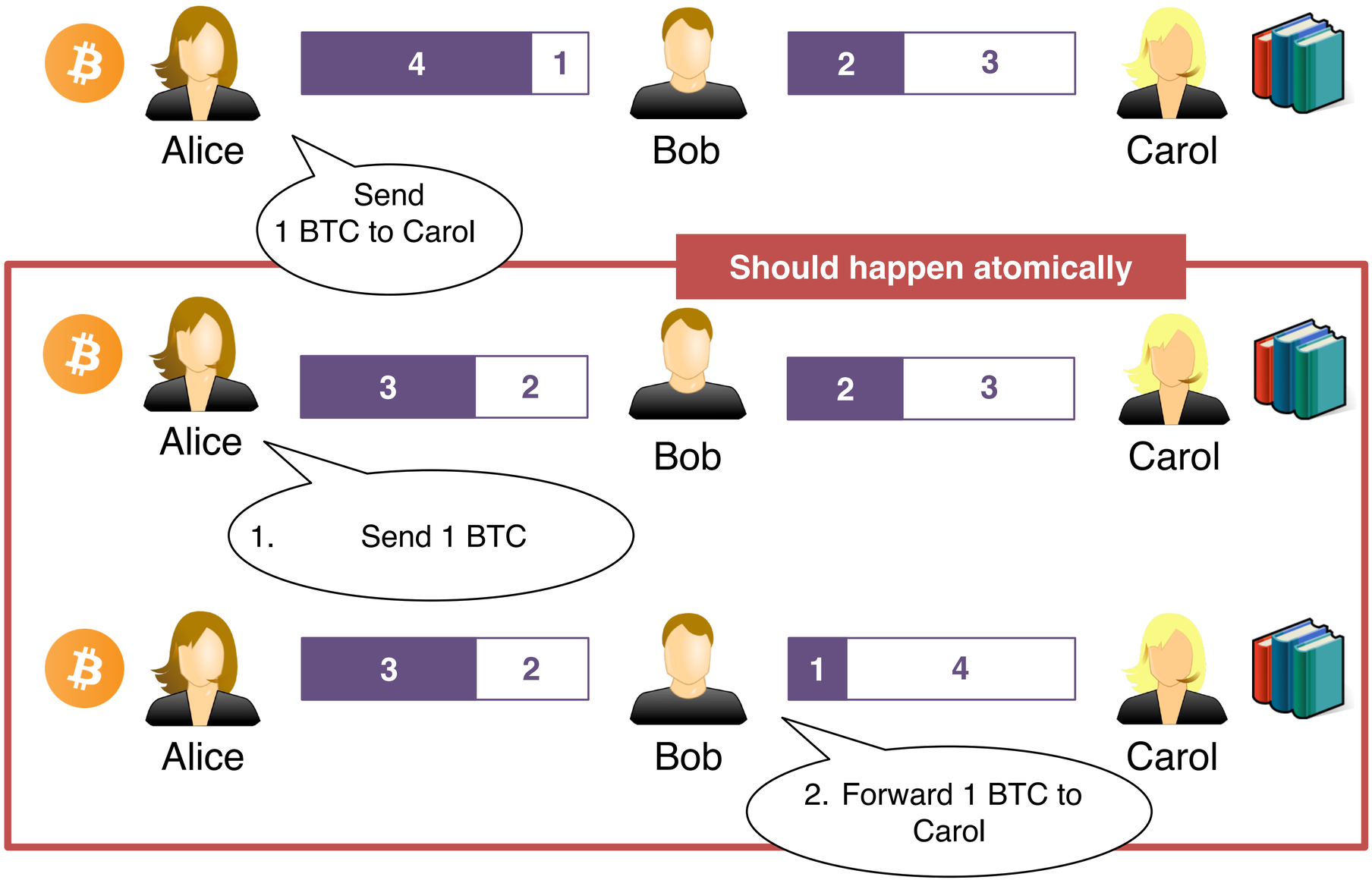
Payment Channel Networks (PCNs)



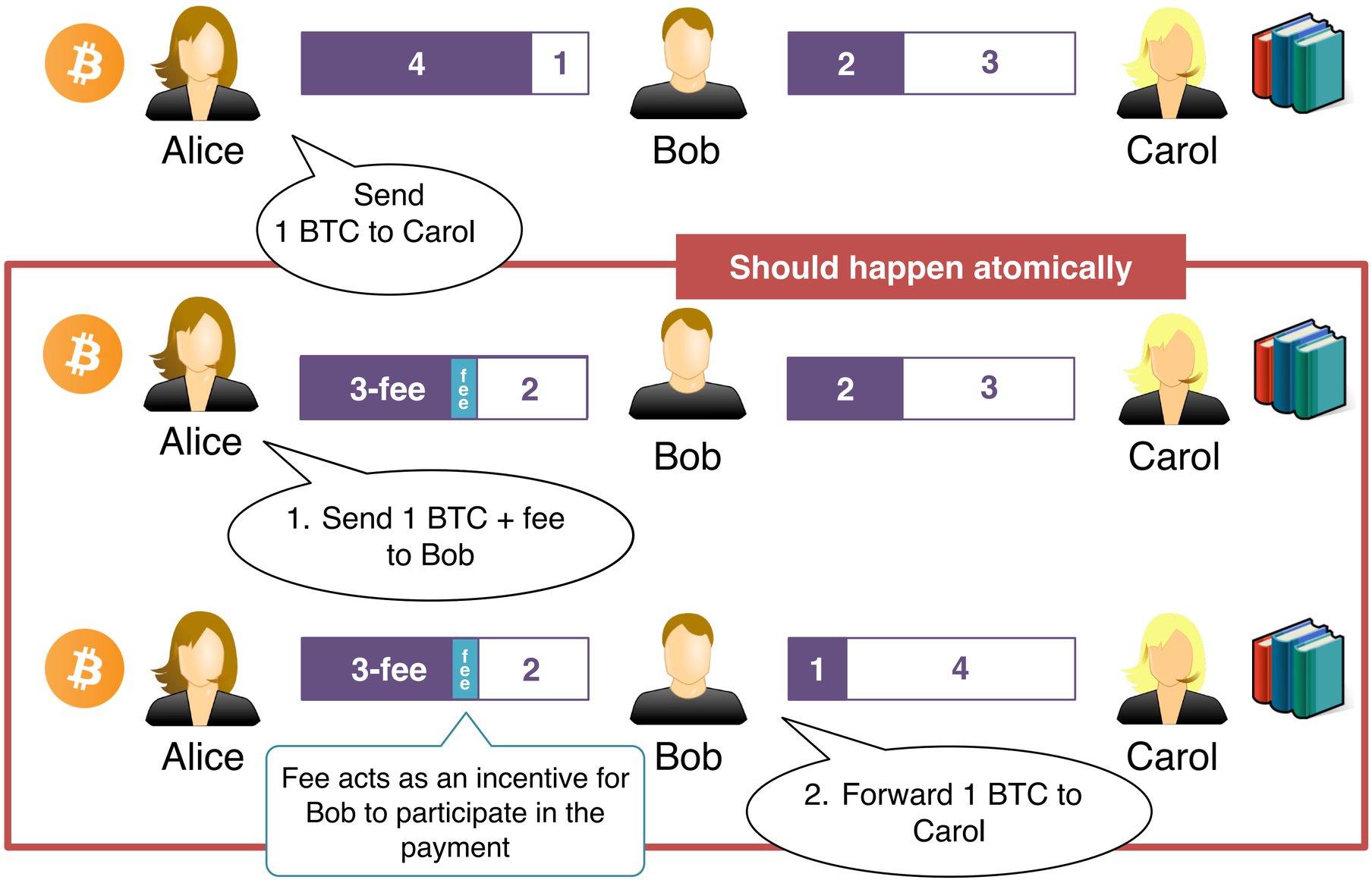
Payment Channel Networks (PCNs)



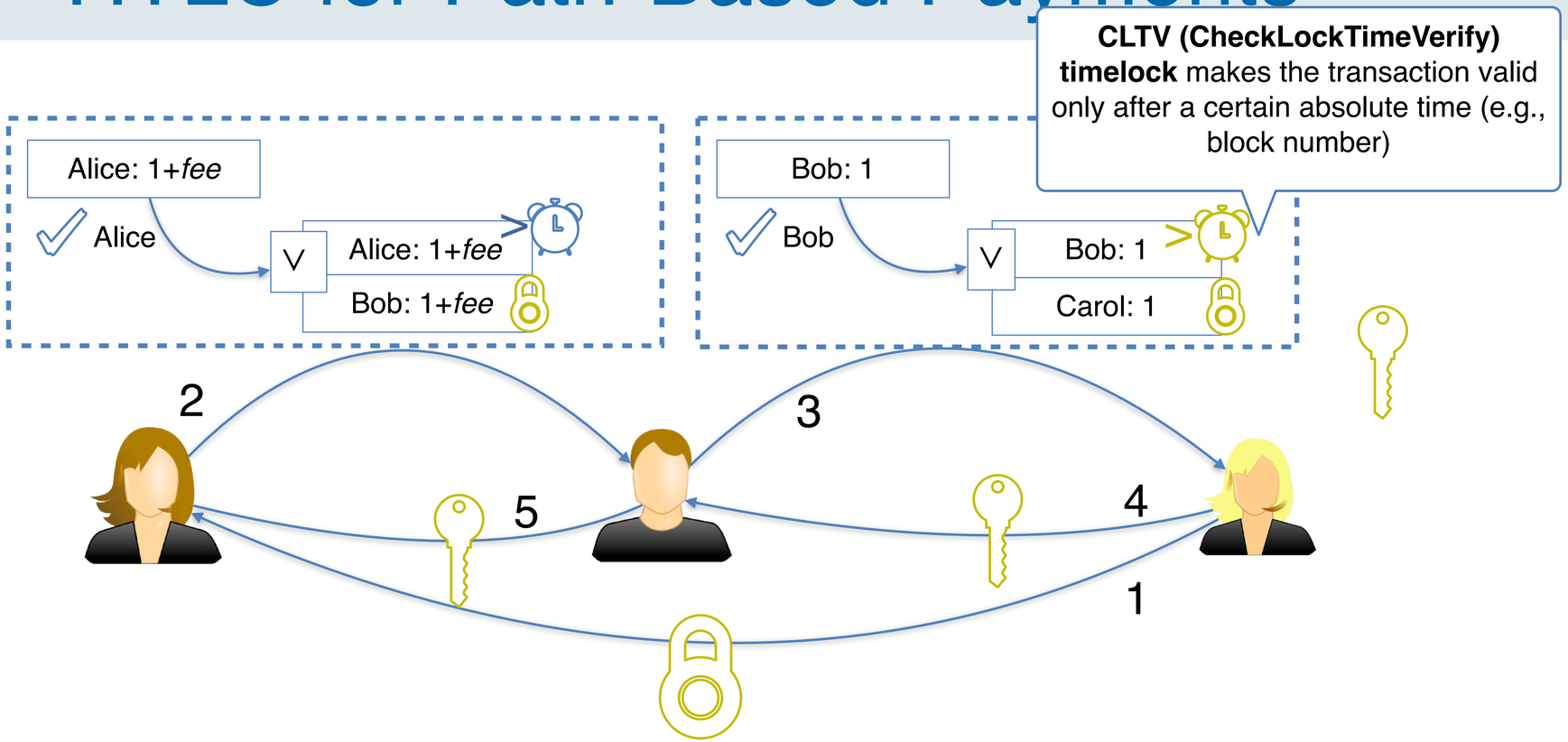
Payment Channel Networks (PCNs)



Payment Channel Networks (PCNs)

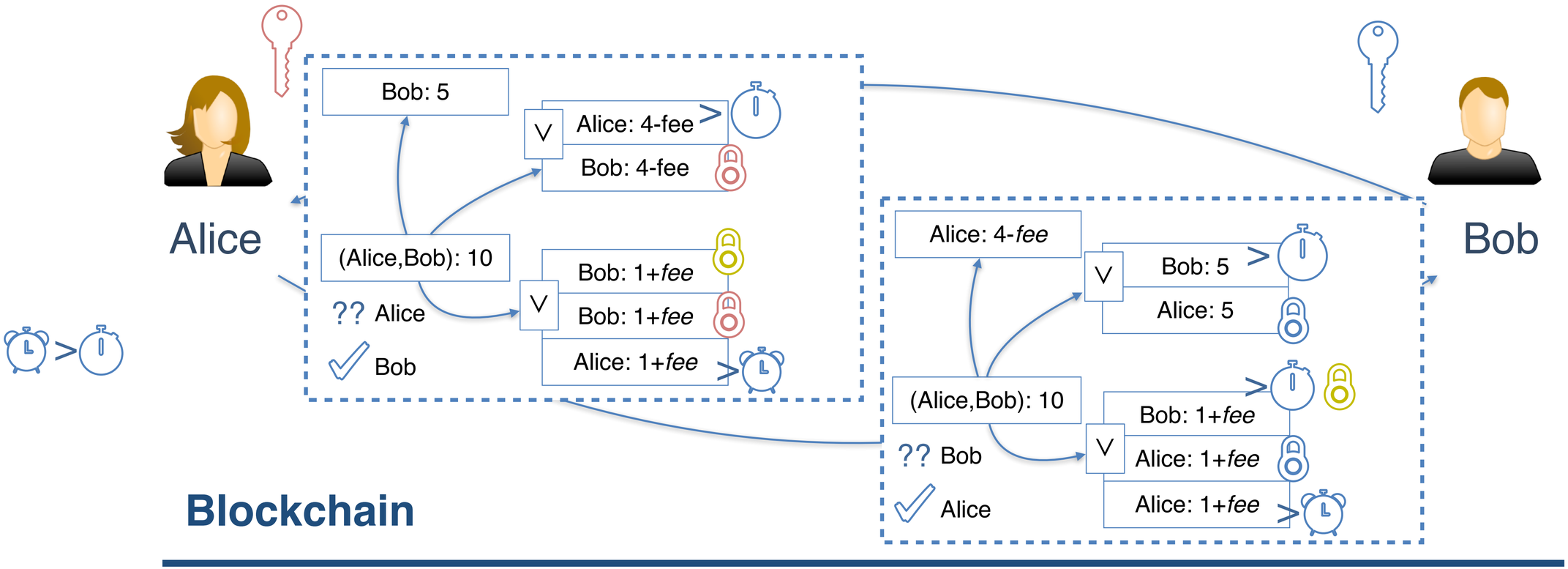


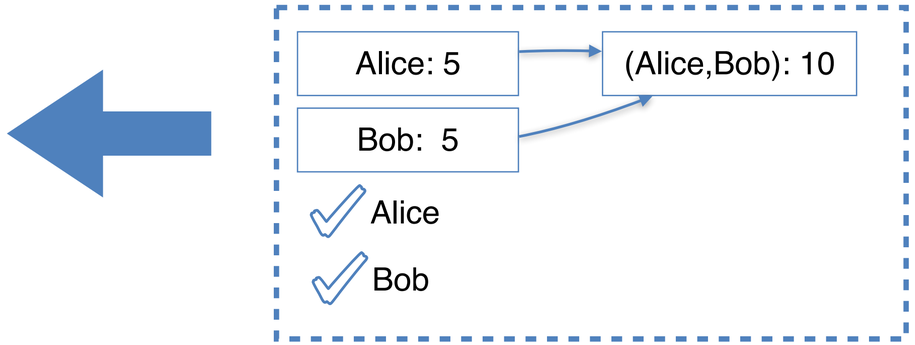
HTLC for Path-Based Payments



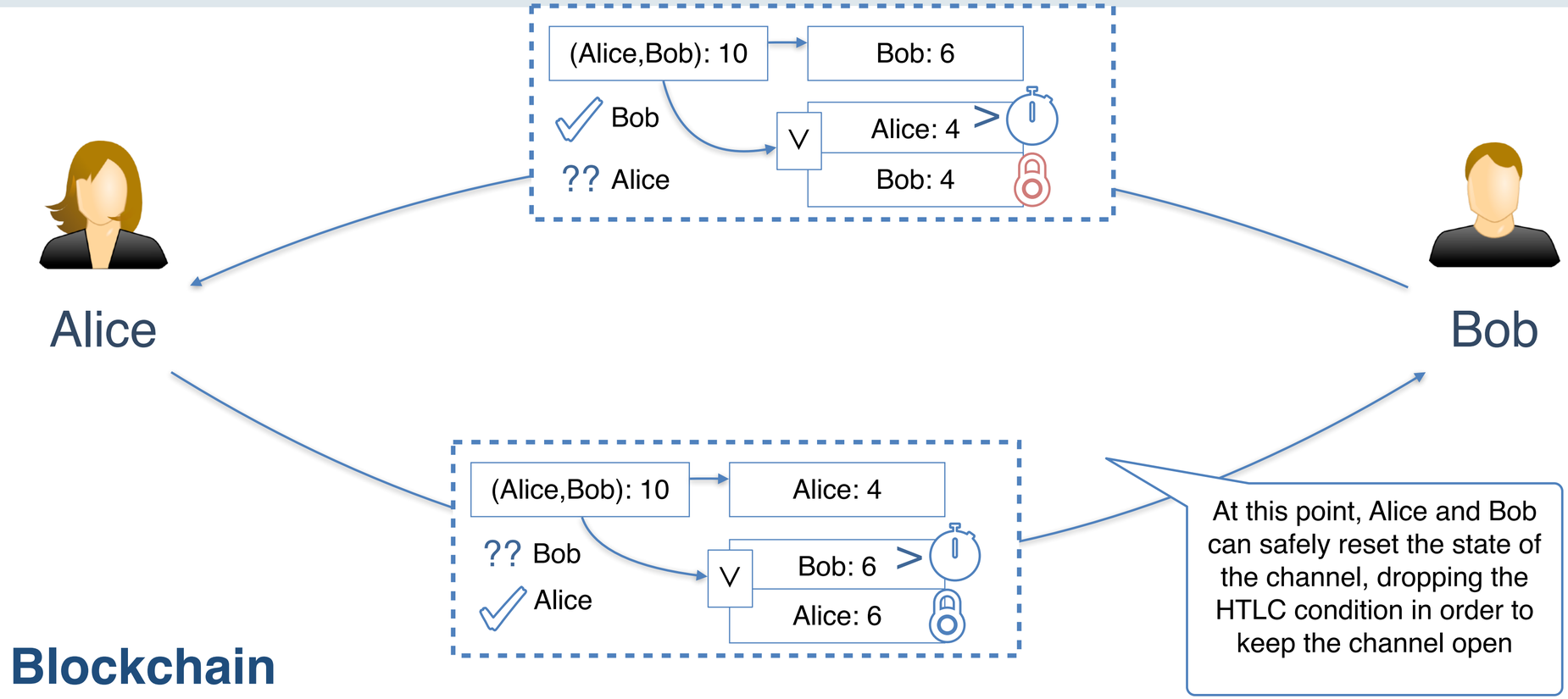
- Since the hash is the same in both transactions, if Carol gets her money then Bob can get her money too!
- It is crucial that  >  in order to give Bob the time to get his money from Alice after Carol posts her transaction

Putting all pieces together...

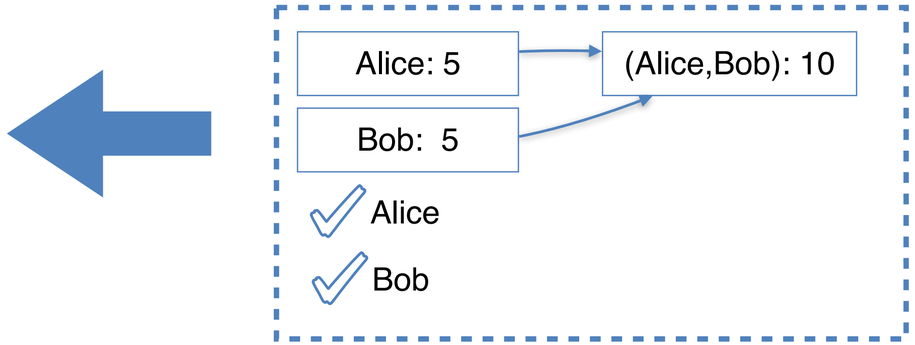




Payment Channels: Optimistic Settlement



Blockchain

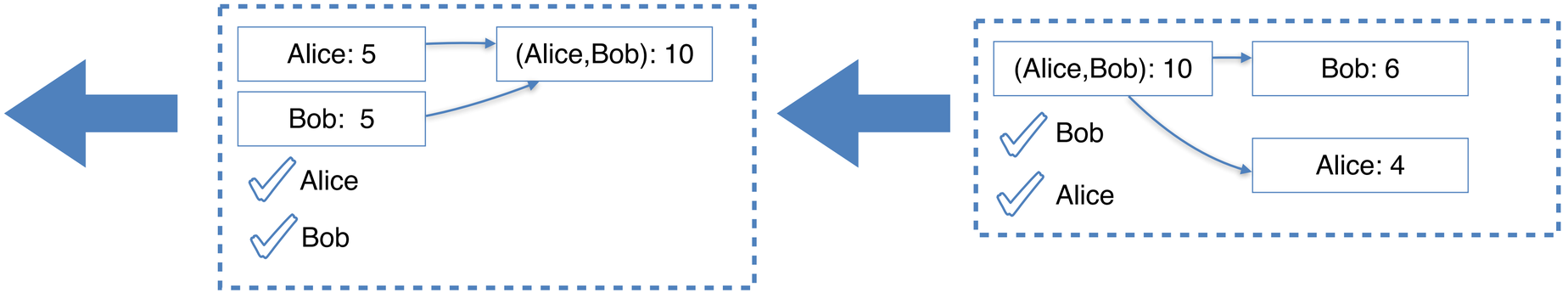


Payment Channels: Closure

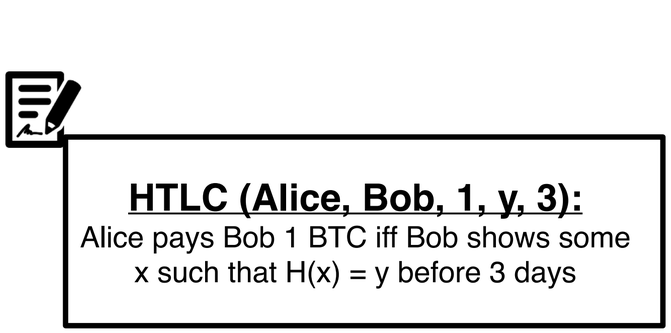
Alice

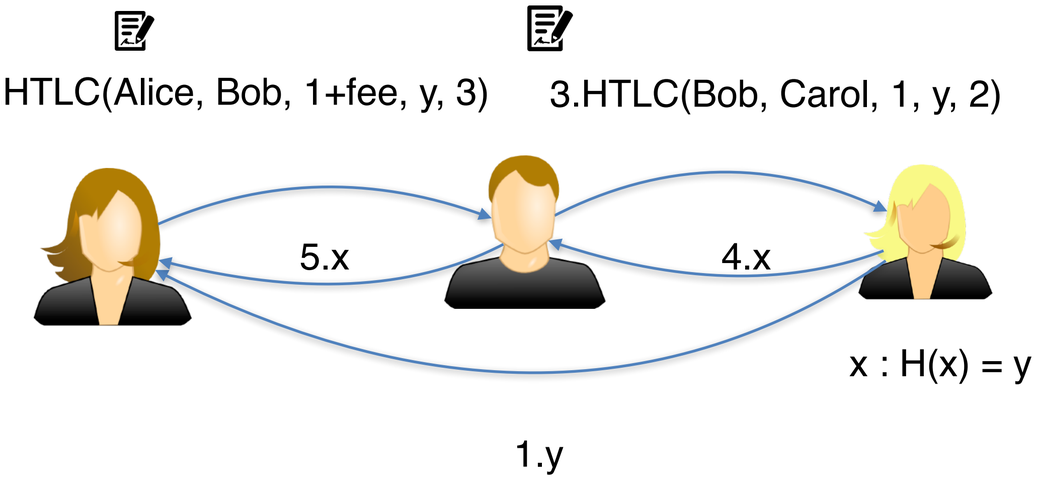
Bob

Blockchain



Take Home

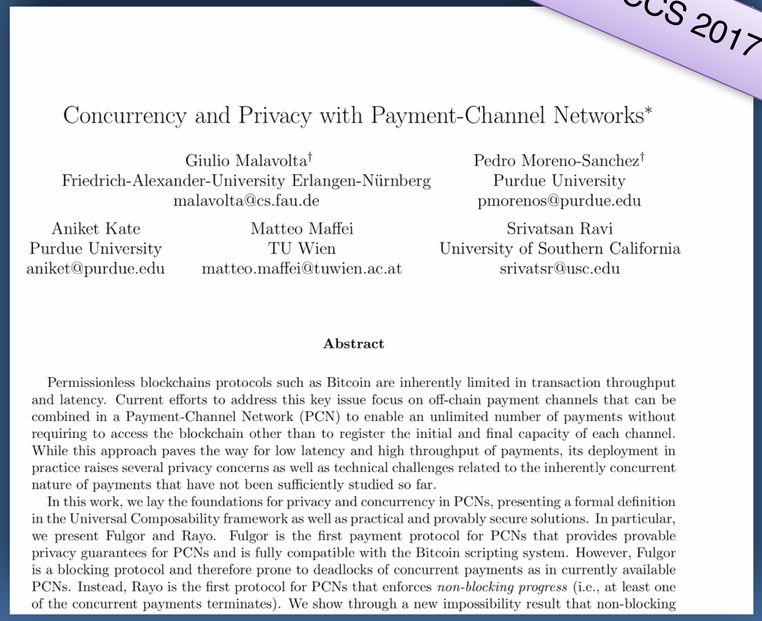




- Lightning Network & Co work allow us to perform *payments offchain*
  - fast, no confirmation delay
  - little fees
  - no blockchain overloading
  - secure and privacy-preserving (at a first glance...)
- The blockchain is used only to mediate disputes

Security and Privacy Issues in Existing PCNs

ACM CCS 2017



Concurrency and Privacy with Payment-Channel Networks*

Giulio Malavolta†
Friedrich-Alexander-University Erlangen-Nürnberg
malavolta@cs.fau.de

Pedro Moreno-Sanchez†
Purdue University
pmorenos@purdue.edu

Aniket Kate
Purdue University
aniket@purdue.edu

Matteo Maffei
TU Wien
matteo.maffei@tuwien.ac.at

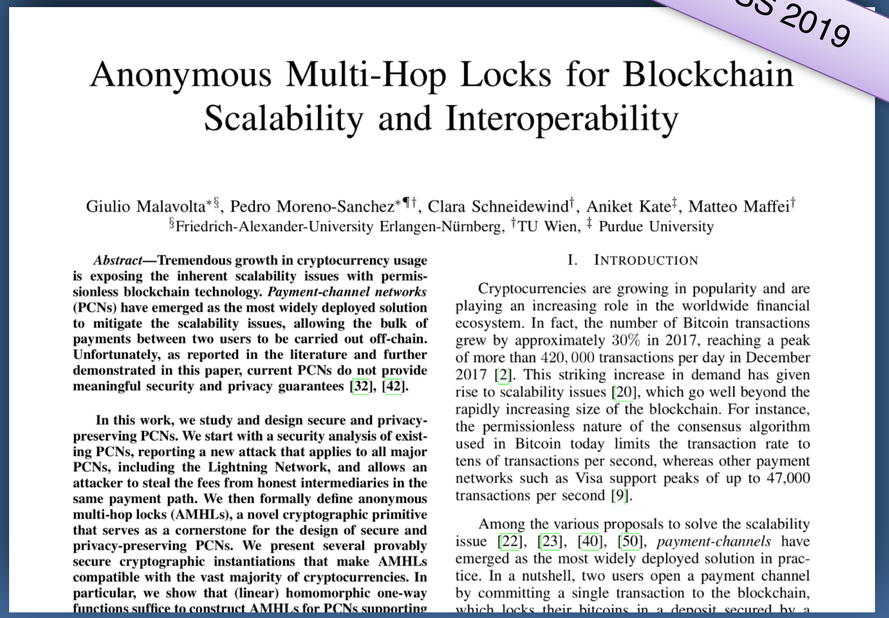
Srivatsan Ravi
University of Southern California
srivatsr@usc.edu

Abstract

Permissionless blockchains protocols such as Bitcoin are inherently limited in transaction throughput and latency. Current efforts to address this key issue focus on off-chain payment channels that can be combined in a Payment-Channel Network (PCN) to enable an unlimited number of payments without requiring to access the blockchain other than to register the initial and final capacity of each channel. While this approach paves the way for low latency and high throughput of payments, its deployment in practice raises several privacy concerns as well as technical challenges related to the inherently concurrent nature of payments that have not been sufficiently studied so far.

In this work, we lay the foundations for privacy and concurrency in PCNs, presenting a formal definition in the Universal Composability framework as well as practical and provably secure solutions. In particular, we present Fulgor and Rayo. Fulgor is the first payment protocol for PCNs that provides provable privacy guarantees for PCNs and is fully compatible with the Bitcoin scripting system. However, Fulgor is a blocking protocol and therefore prone to deadlocks of concurrent payments as in currently available PCNs. Instead, Rayo is the first protocol for PCNs that enforces *non-blocking progress* (i.e., at least one of the concurrent payments terminates). We show through a new impossibility result that non-blocking

NDSS 2019



Anonymous Multi-Hop Locks for Blockchain Scalability and Interoperability

Giulio Malavolta*§, Pedro Moreno-Sanchez*¶†, Clara Schneidewind†, Aniket Kate‡, Matteo Maffei†
§Friedrich-Alexander-University Erlangen-Nürnberg, †TU Wien, ‡ Purdue University

***Abstract*—Tremendous growth in cryptocurrency usage is exposing the inherent scalability issues with permissionless blockchain technology. *Payment-channel networks* (PCNs) have emerged as the most widely deployed solution to mitigate the scalability issues, allowing the bulk of payments between two users to be carried out off-chain. Unfortunately, as reported in the literature and further demonstrated in this paper, current PCNs do not provide meaningful security and privacy guarantees [32], [42].**

In this work, we study and design secure and privacy-preserving PCNs. We start with a security analysis of existing PCNs, reporting a new attack that applies to all major PCNs, including the Lightning Network, and allows an attacker to steal the fees from honest intermediaries in the same payment path. We then formally define anonymous multi-hop locks (AMHLs), a novel cryptographic primitive that serves as a cornerstone for the design of secure and privacy-preserving PCNs. We present several provably secure cryptographic instantiations that make AMHLs compatible with the vast majority of cryptocurrencies. In particular, we show that (linear) homomorphic one-way functions suffice to construct AMHLs for PCNs supporting

I. Introduction

Cryptocurrencies are growing in popularity and are playing an increasing role in the worldwide financial ecosystem. In fact, the number of Bitcoin transactions grew by approximately 30% in 2017, reaching a peak of more than 420,000 transactions per day in December 2017 [2]. This striking increase in demand has given rise to scalability issues [20], which go well beyond the rapidly increasing size of the blockchain. For instance, the permissionless nature of the consensus algorithm used in Bitcoin today limits the transaction rate to tens of transactions per second, whereas other payment networks such as Visa support peaks of up to 47,000 transactions per second [9].

Among the various proposals to solve the scalability issue [22], [23], [40], [50], *payment-channels* have emerged as the most widely deployed solution in practice. In a nutshell, two users open a payment channel by committing a single transaction to the blockchain, which locks their bitcoins in a deposit secured by a

Security + Privacy in PCNs

Are off-chain payments in PCNs secure?

(No honest participant looses money!)

Are off-chain payments in PCNs privacy-preserving by default?

(individual payments are not recorded on the blockchain!)

Security + Privacy in PCNs

Are off-chain payments in PCNs secure?

(No honest participant looses money!)

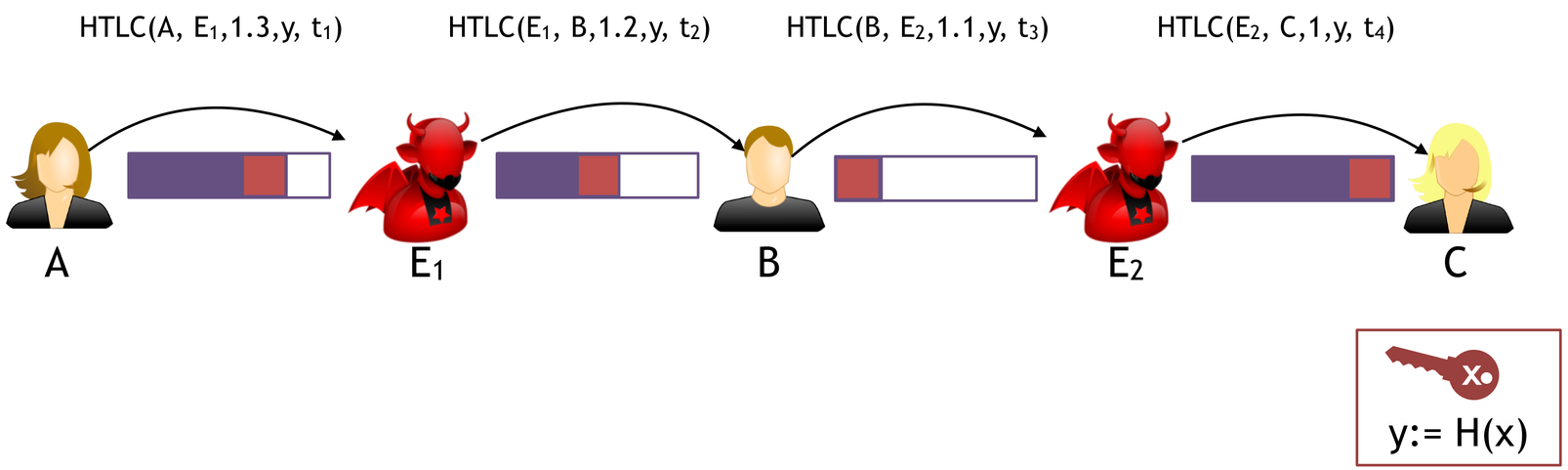
NO!

Are off-chain payments in PCNs privacy-preserving by default?

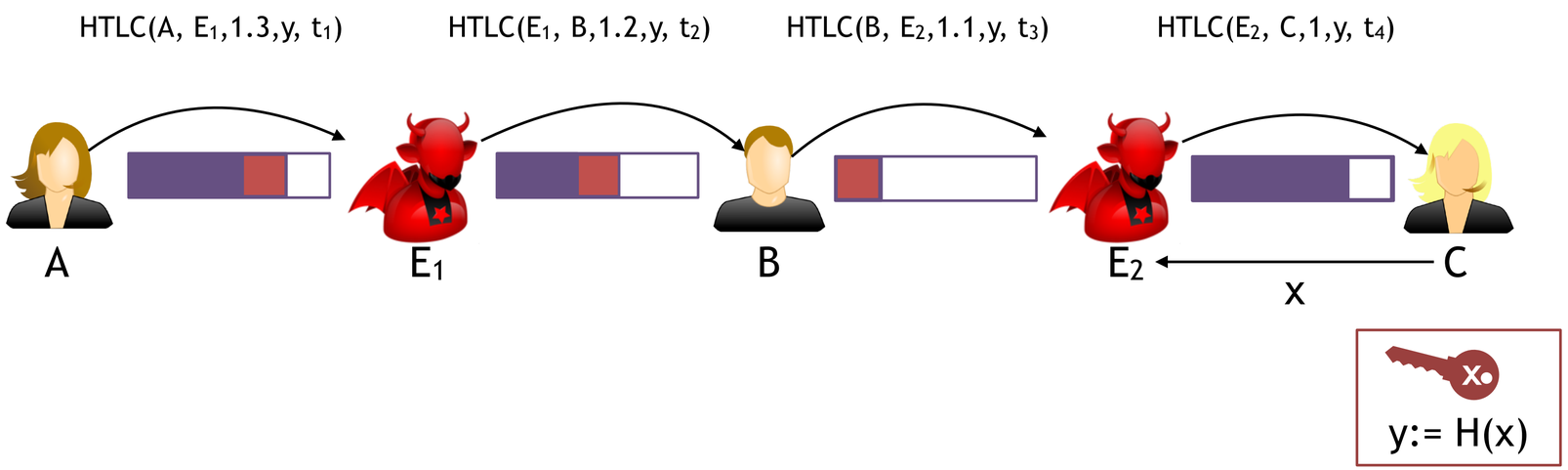
(individual payments are not recorded on the blockchain!)

NO!

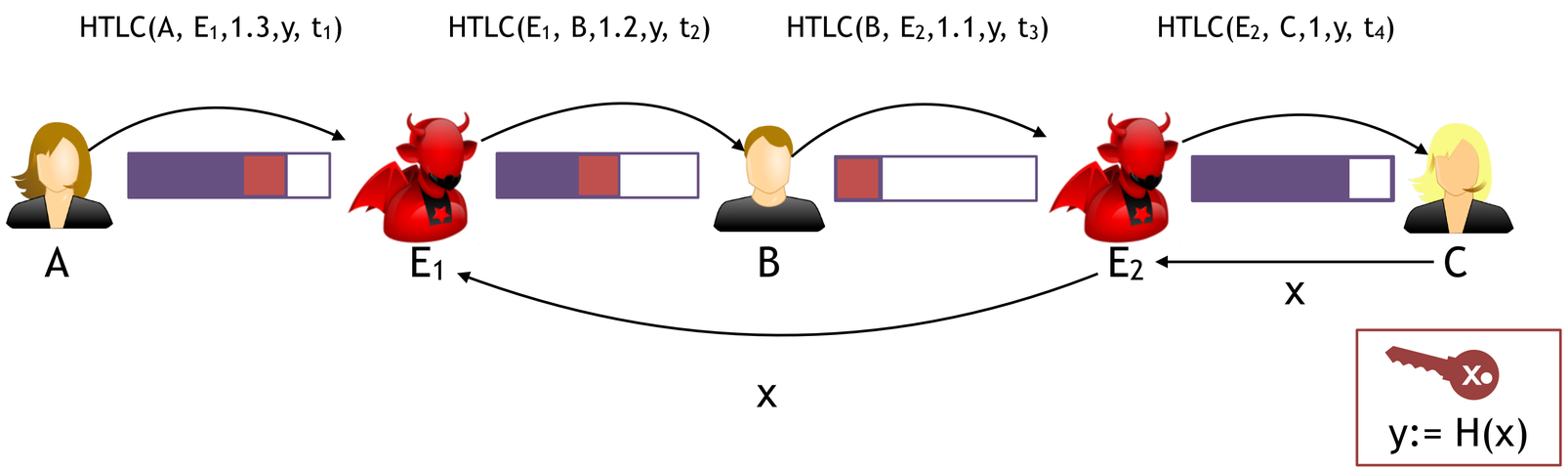
Security Issue: The Wormhole Attack



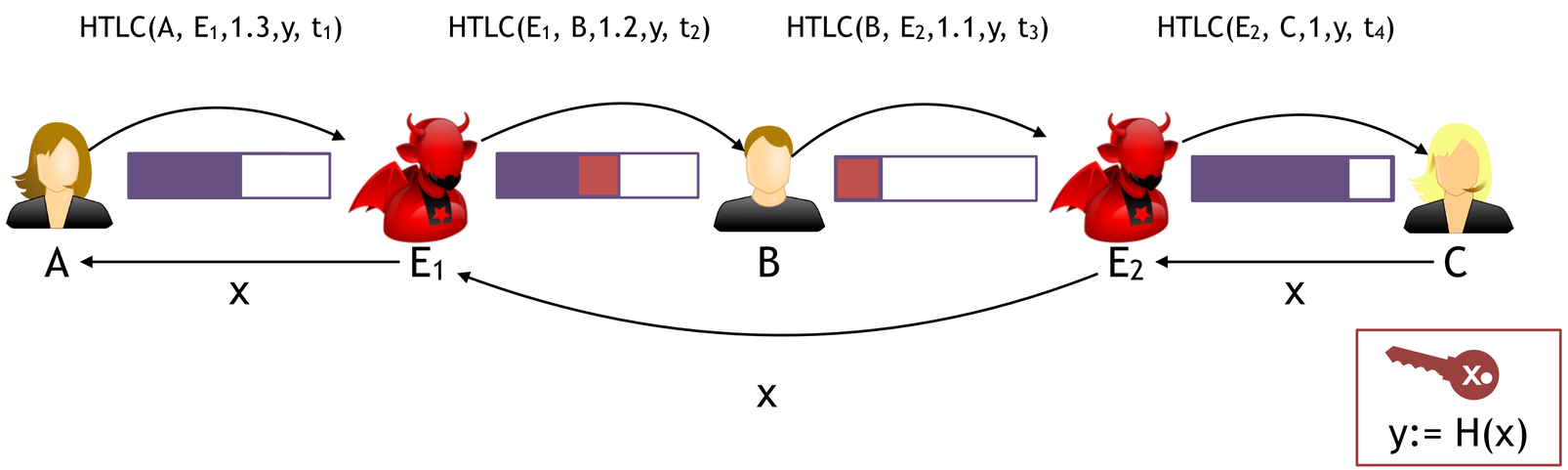
Security Issue: The Wormhole Attack



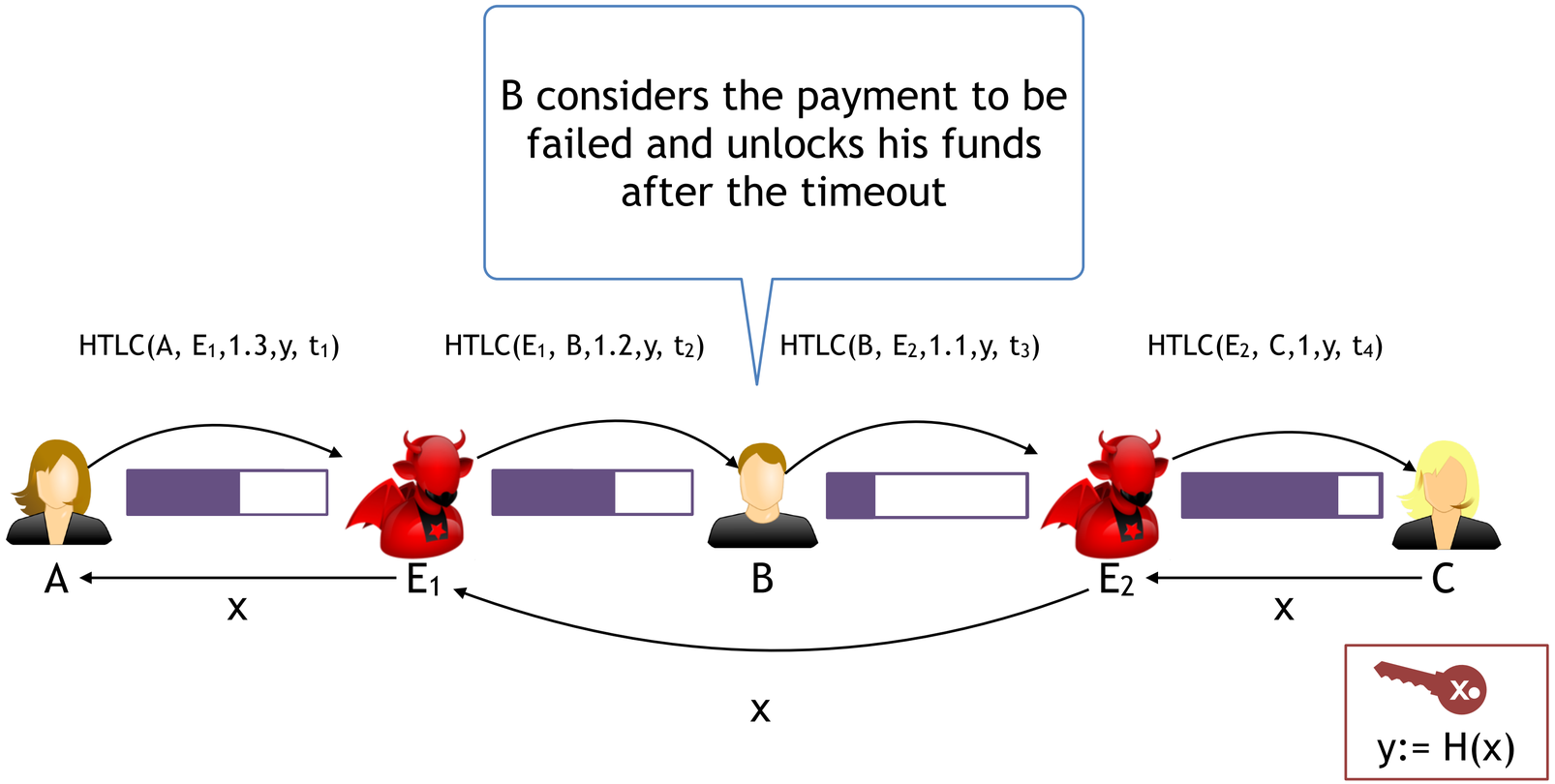
Security Issue: The Wormhole Attack



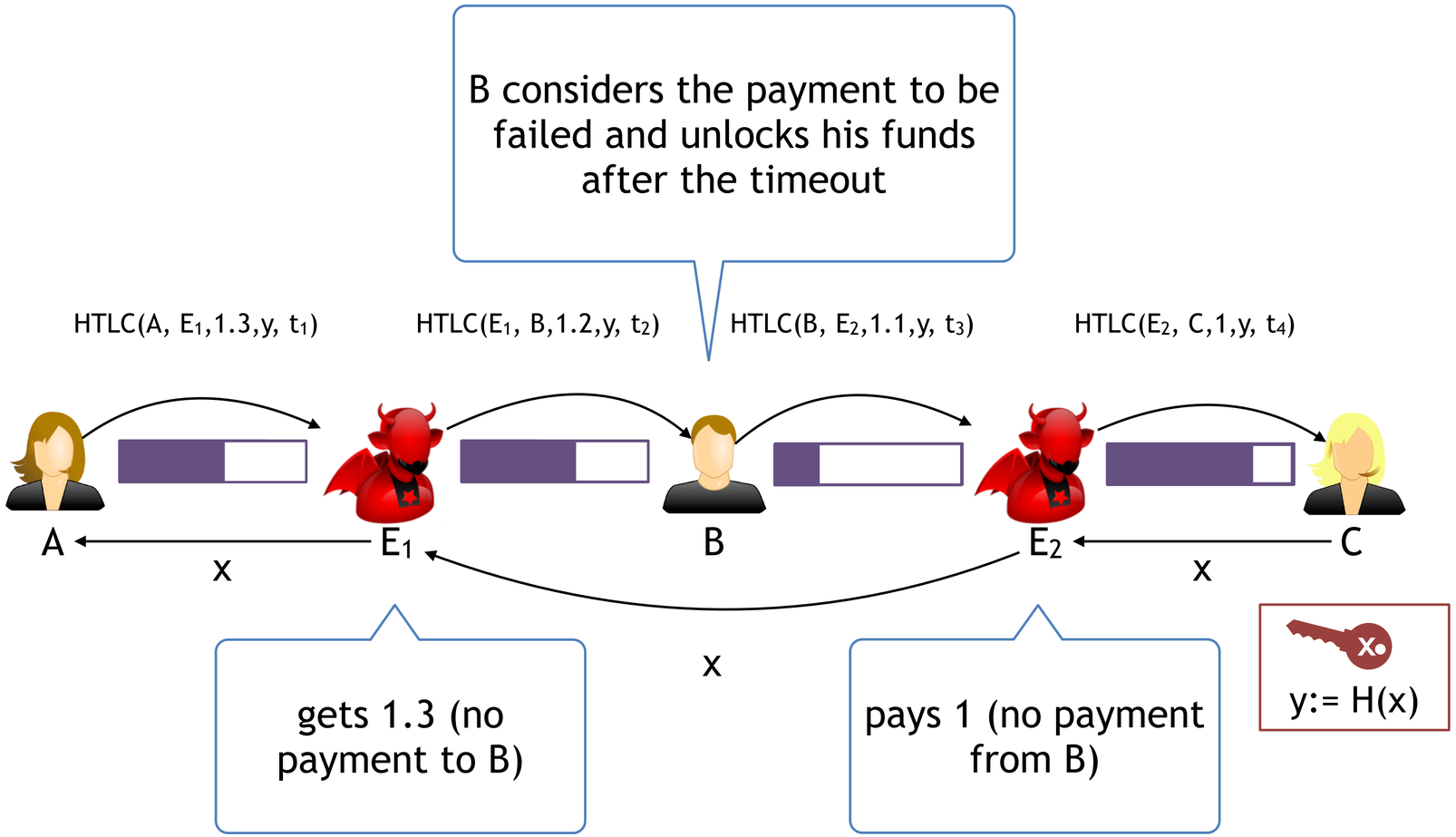
Security Issue: The Wormhole Attack



Security Issue: The Wormhole Attack

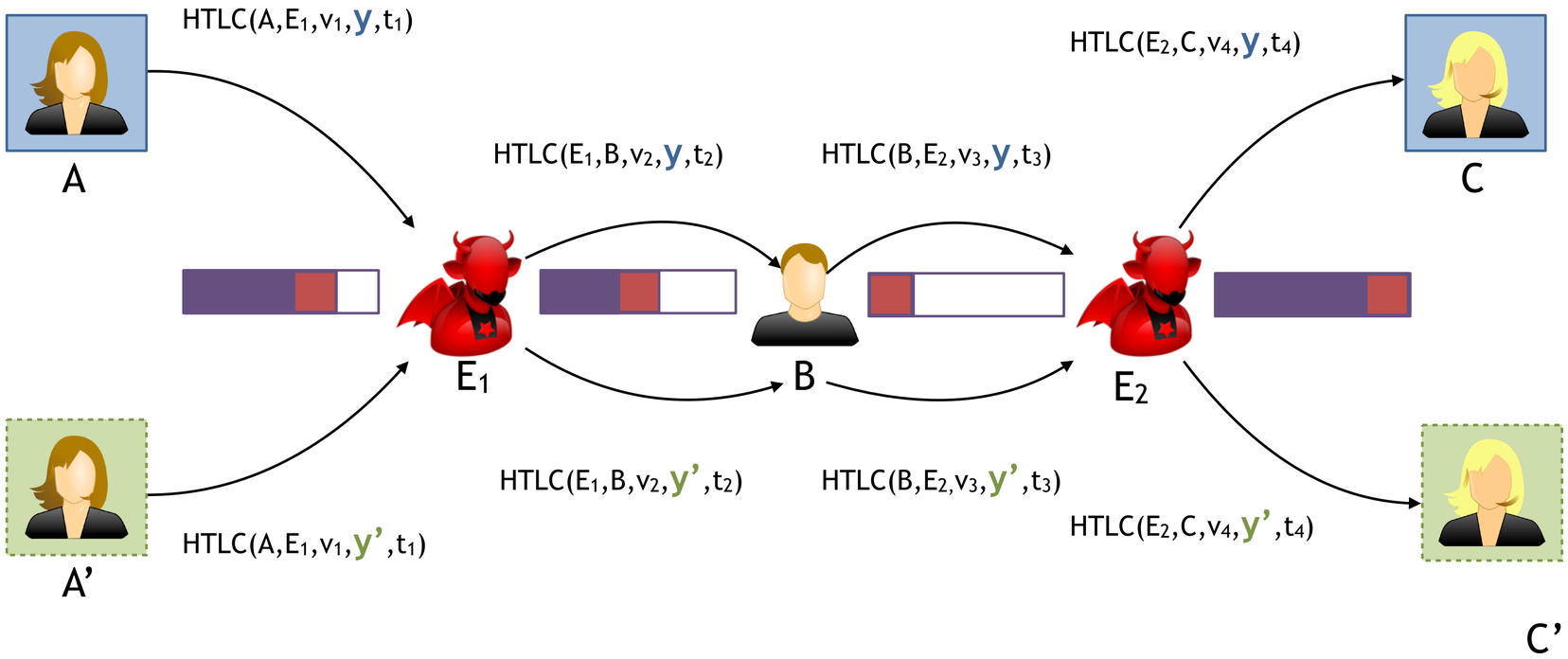


Security Issue: The Wormhole Attack



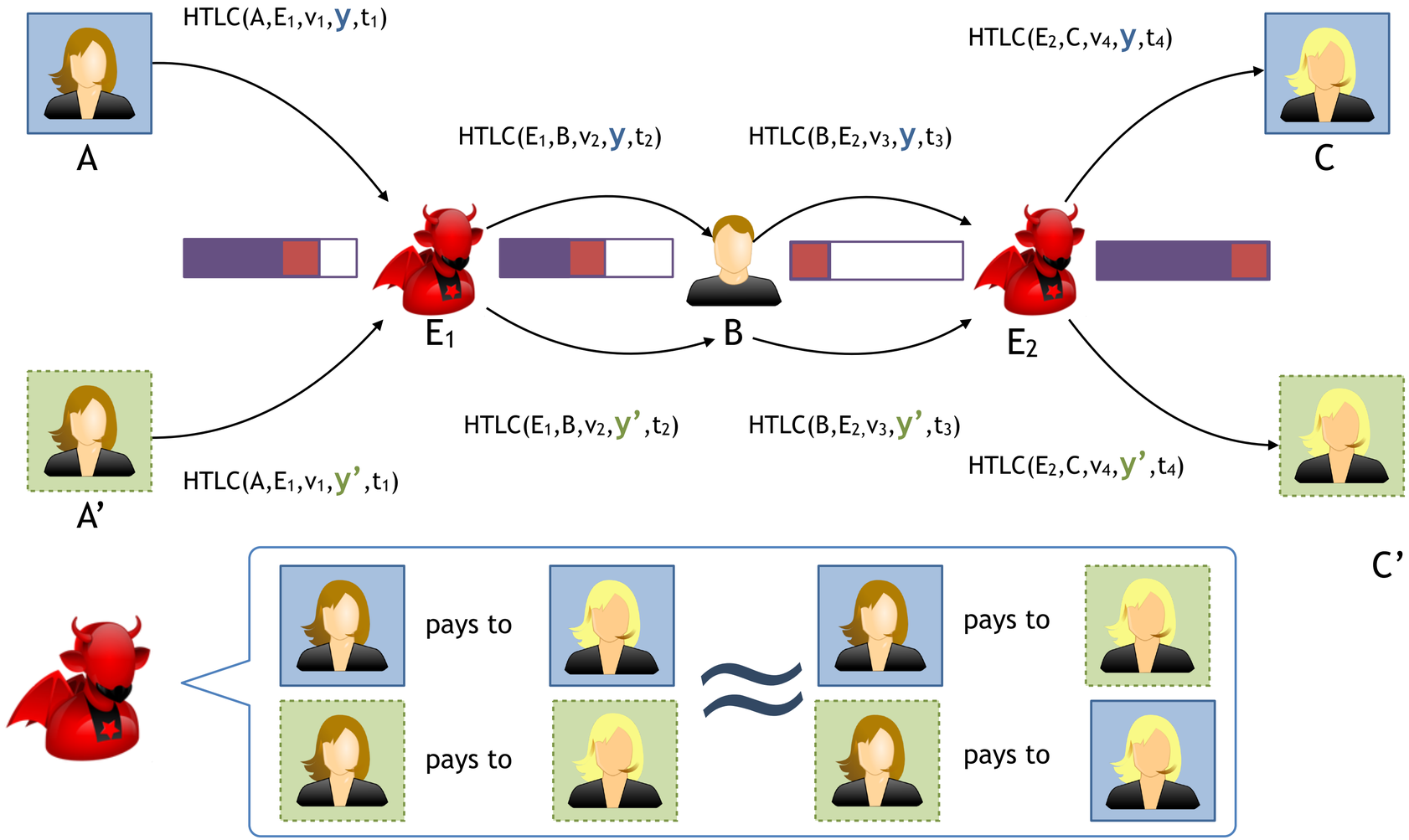
Attacker earns 0.3 BTC (own fees + B's fees)

Privacy Issues in HTLC Payments



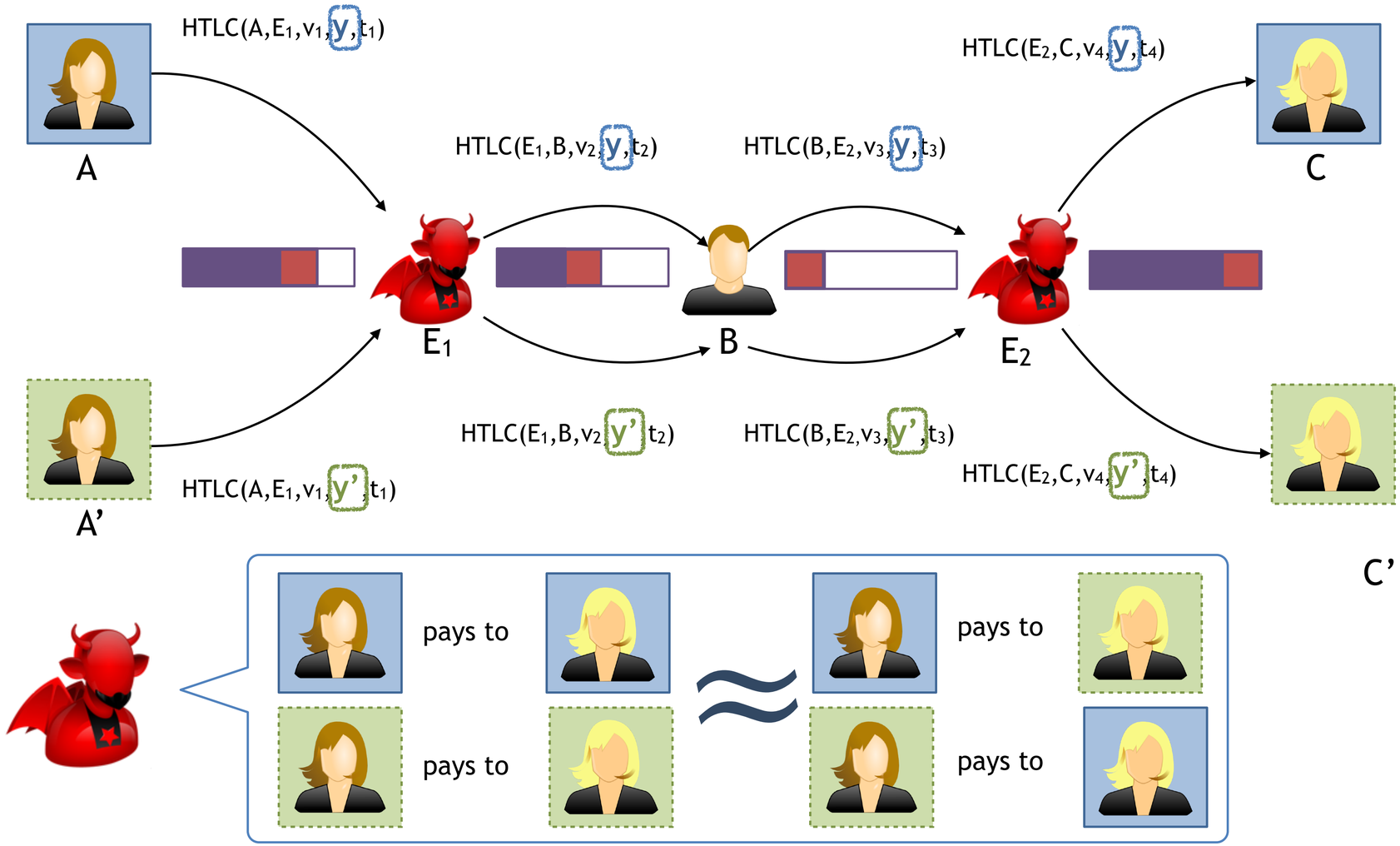
Relationship Anonymity: On-path adversaries do not learn who pays to whom

Privacy Issues in HTLC Payments



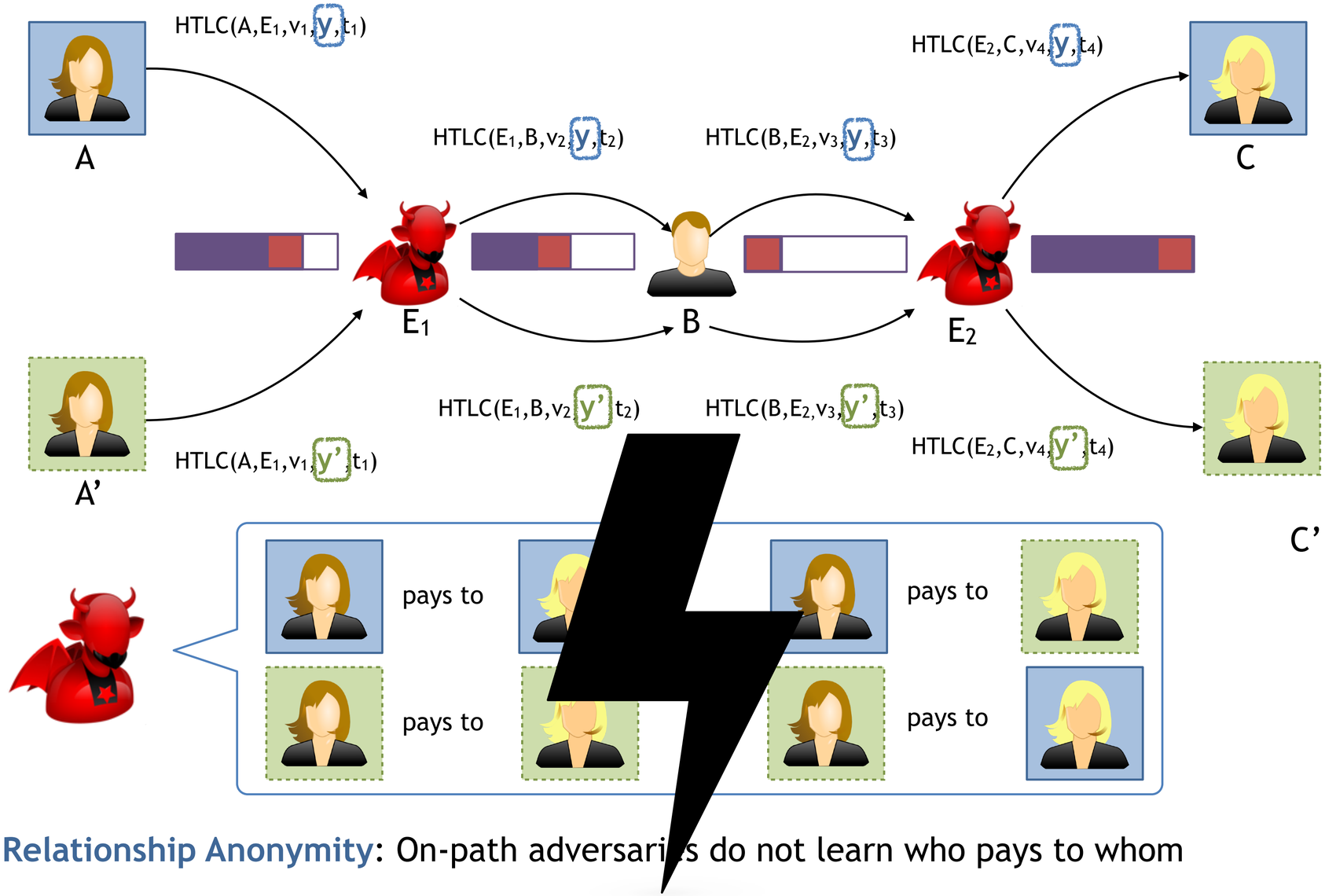
Relationship Anonymity: On-path adversaries do not learn who pays to whom

Privacy Issues in HTLC Payments



Relationship Anonymity: On-path adversaries do not learn who pays to whom

Privacy Issues in HTLC Payments



Relationship Anonymity: On-path adversaries do not learn who pays to whom

Fulgor

ACM CCS 2018

Concurrency and Privacy with Payment-Channel Networks*

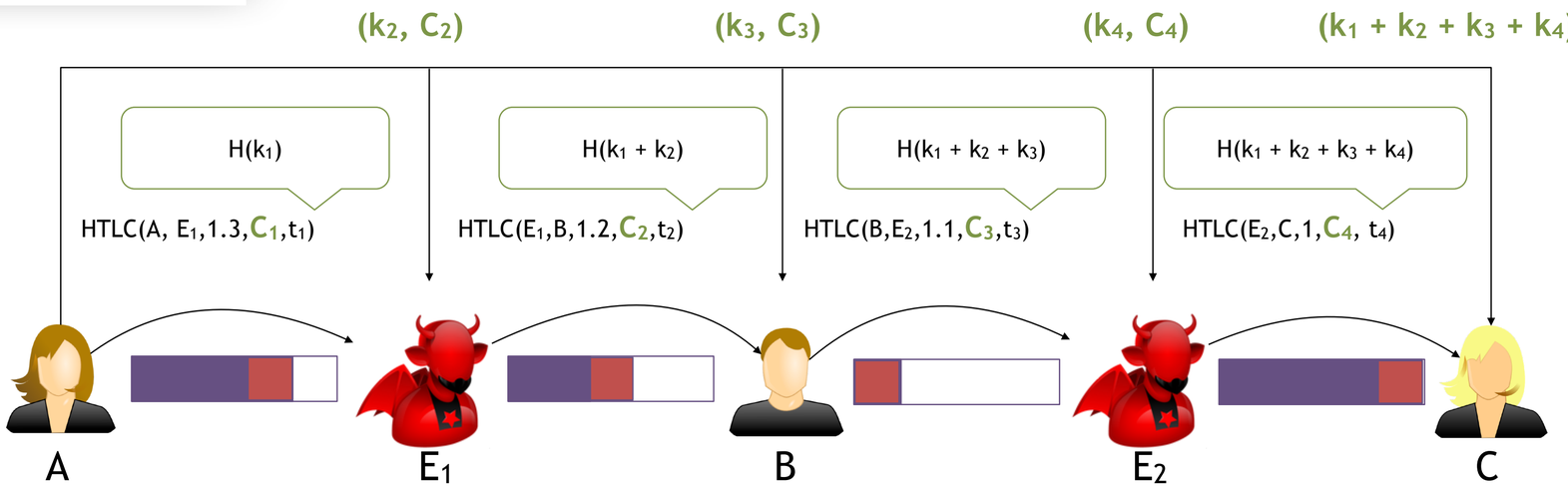
Giulio Malavolta†
Friedrich-Alexander-University Erlangen-Nürnberg
malavolta@cs.fau.de

Pedro Moreno-Sanchez†
Purdue University
pmorenos@purdue.edu

Aniket Kate
Purdue University
aniket@purdue.edu

Matteo Maffei
TU Wien
matteo.maffei@tuwien.ac.at

Srivatsan Ravi
University of Southern California
srivatsr@usc.edu



Fulgor

ACM CCS 2018

Concurrency and Privacy with Payment-Channel Networks*

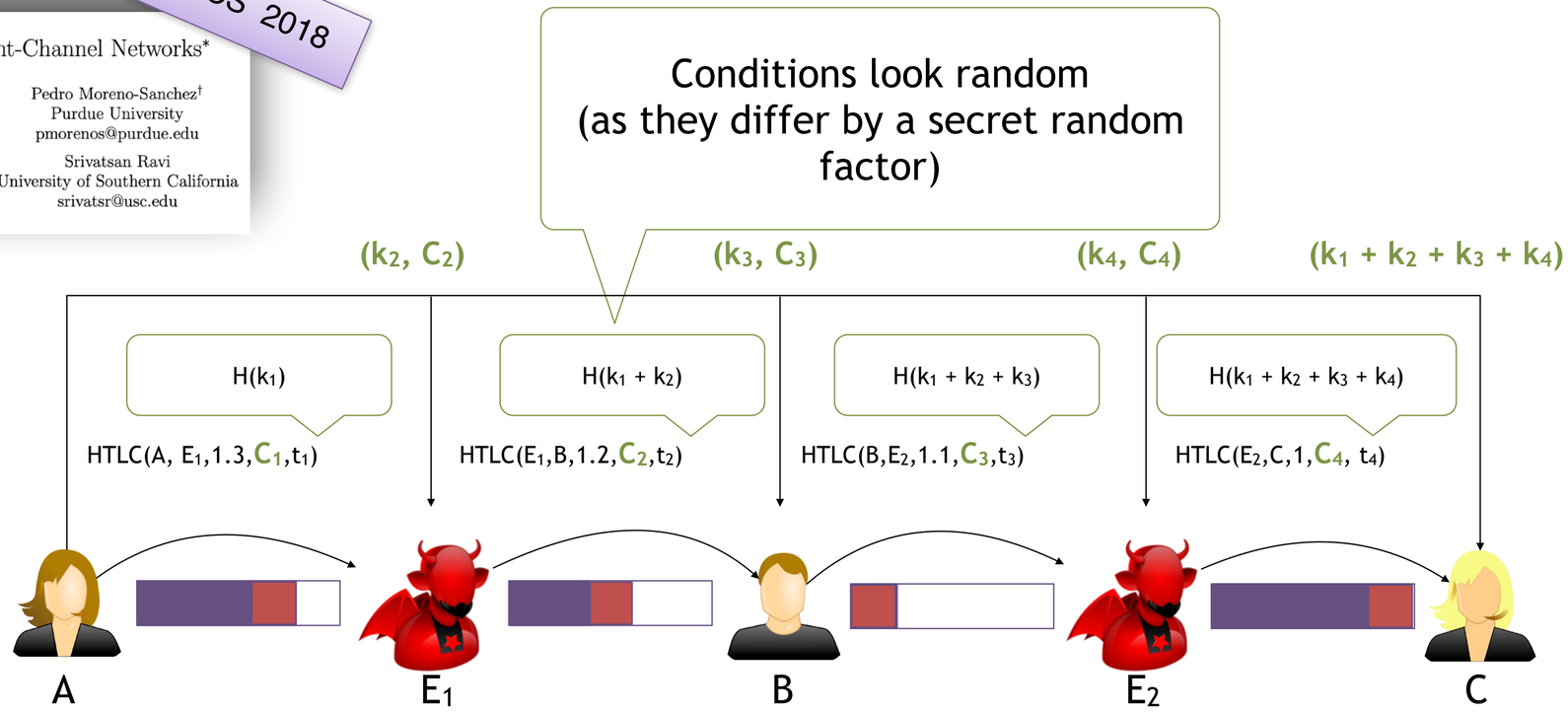
Giulio Malavolta†
Friedrich-Alexander-University Erlangen-Nürnberg
malavolta@cs.fau.de

Pedro Moreno-Sanchez†
Purdue University
pmorenos@purdue.edu

Aniket Kate
Purdue University
aniket@purdue.edu

Matteo Maffei
TU Wien
matteo.maffei@tuwien.ac.at

Srivatsan Ravi
University of Southern California
srivatsr@usc.edu



Fulgor

ACM CCS 2018

Concurrency and Privacy with Payment-Channel Networks*

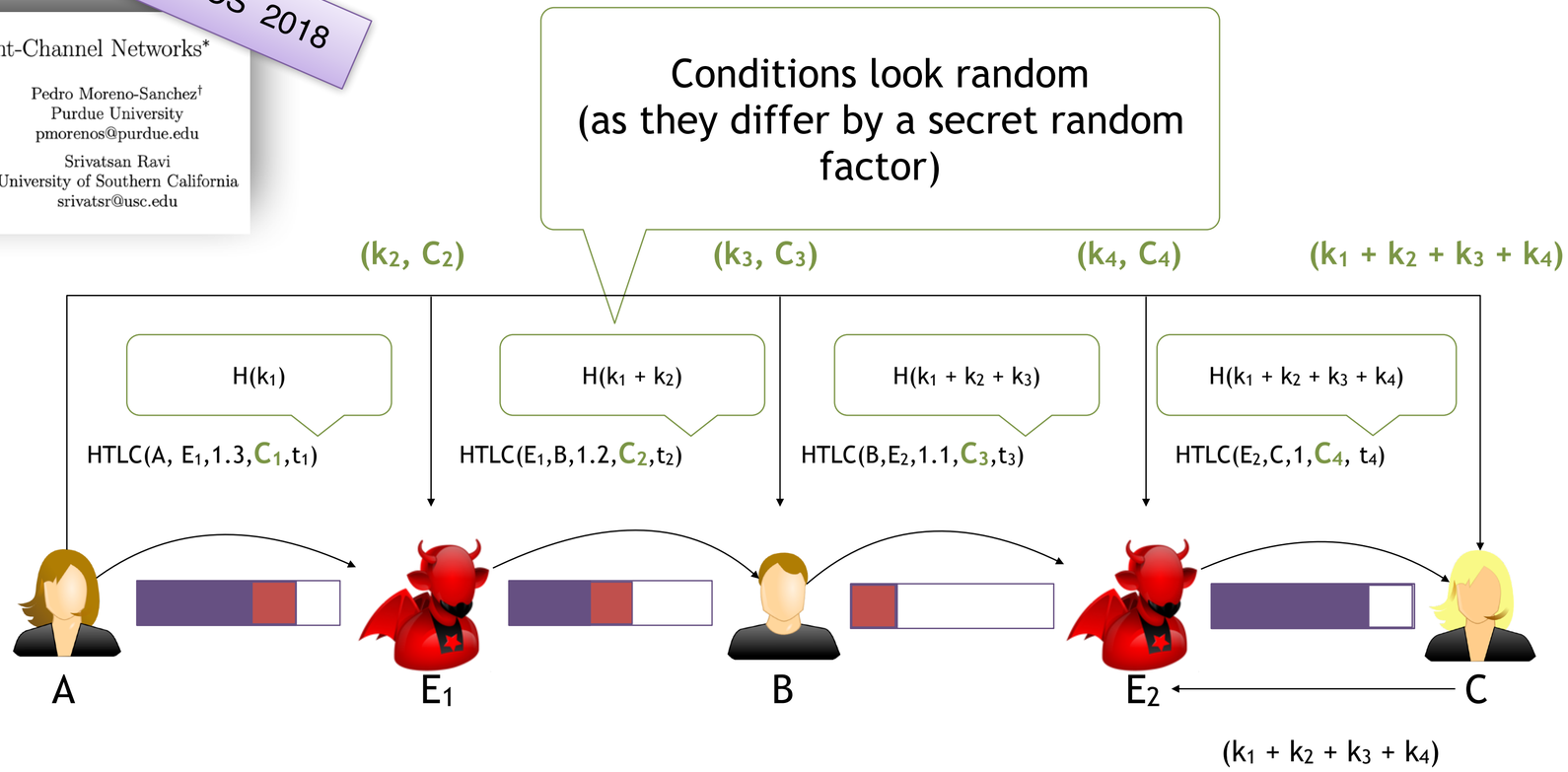
Giulio Malavolta†
Friedrich-Alexander-University Erlangen-Nürnberg
malavolta@cs.fau.de

Pedro Moreno-Sanchez†
Purdue University
pmorenos@purdue.edu

Aniket Kate
Purdue University
aniket@purdue.edu

Matteo Maffei
TU Wien
matteo.maffei@tuwien.ac.at

Srivatsan Ravi
University of Southern California
srivatsr@usc.edu



Fulgor

ACM CCS 2018

Concurrency and Privacy with Payment-Channel Networks*

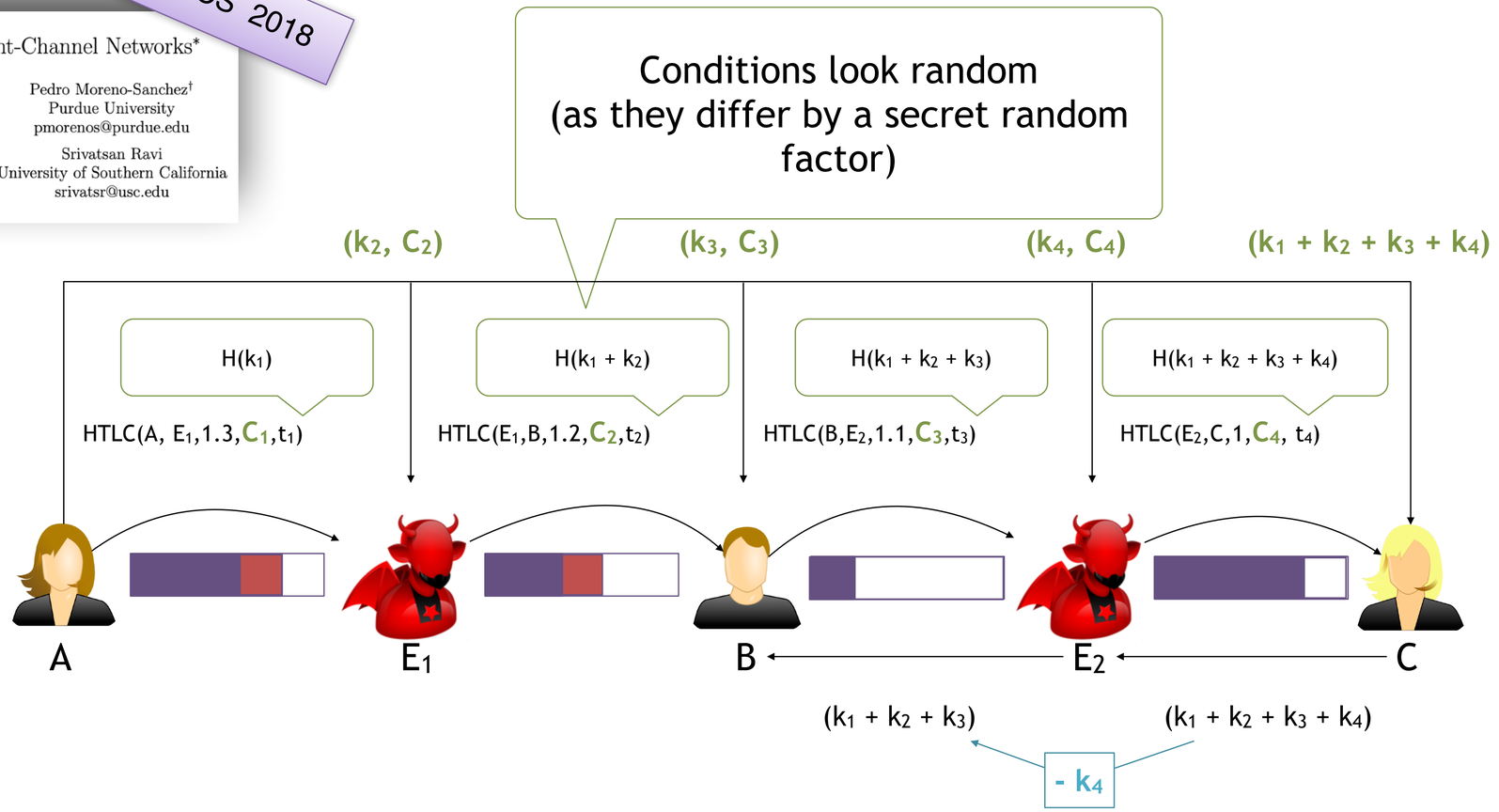
Giulio Malavolta†
Friedrich-Alexander-University Erlangen-Nürnberg
malavolta@cs.fau.de

Pedro Moreno-Sanchez†
Purdue University
pmorenos@purdue.edu

Aniket Kate
Purdue University
aniket@purdue.edu

Matteo Maffei
TU Wien
matteo.maffei@tuwien.ac.at

Srivatsan Ravi
University of Southern California
srivatsr@usc.edu



Fulgor

ACM CCS 2018

Concurrency and Privacy with Payment-Channel Networks*

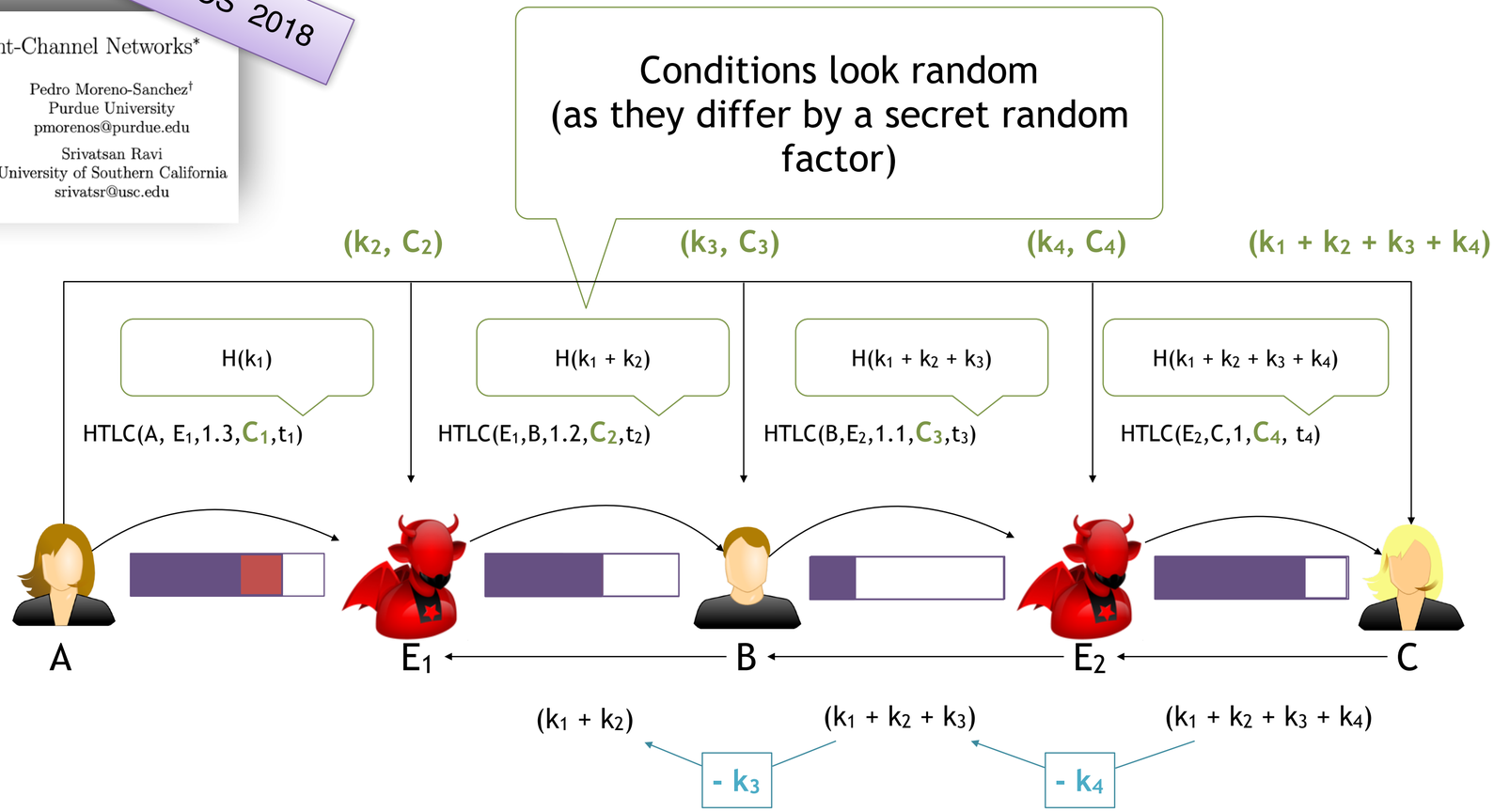
Giulio Malavolta†
Friedrich-Alexander-University Erlangen-Nürnberg
malavolta@cs.fau.de

Pedro Moreno-Sanchez†
Purdue University
pmorenos@purdue.edu

Aniket Kate
Purdue University
aniket@purdue.edu

Matteo Maffei
TU Wien
matteo.maffei@tuwien.ac.at

Srivatsan Ravi
University of Southern California
srivatsr@usc.edu



Fulgor

ACM CCS 2018

Concurrency and Privacy with Payment-Channel Networks*

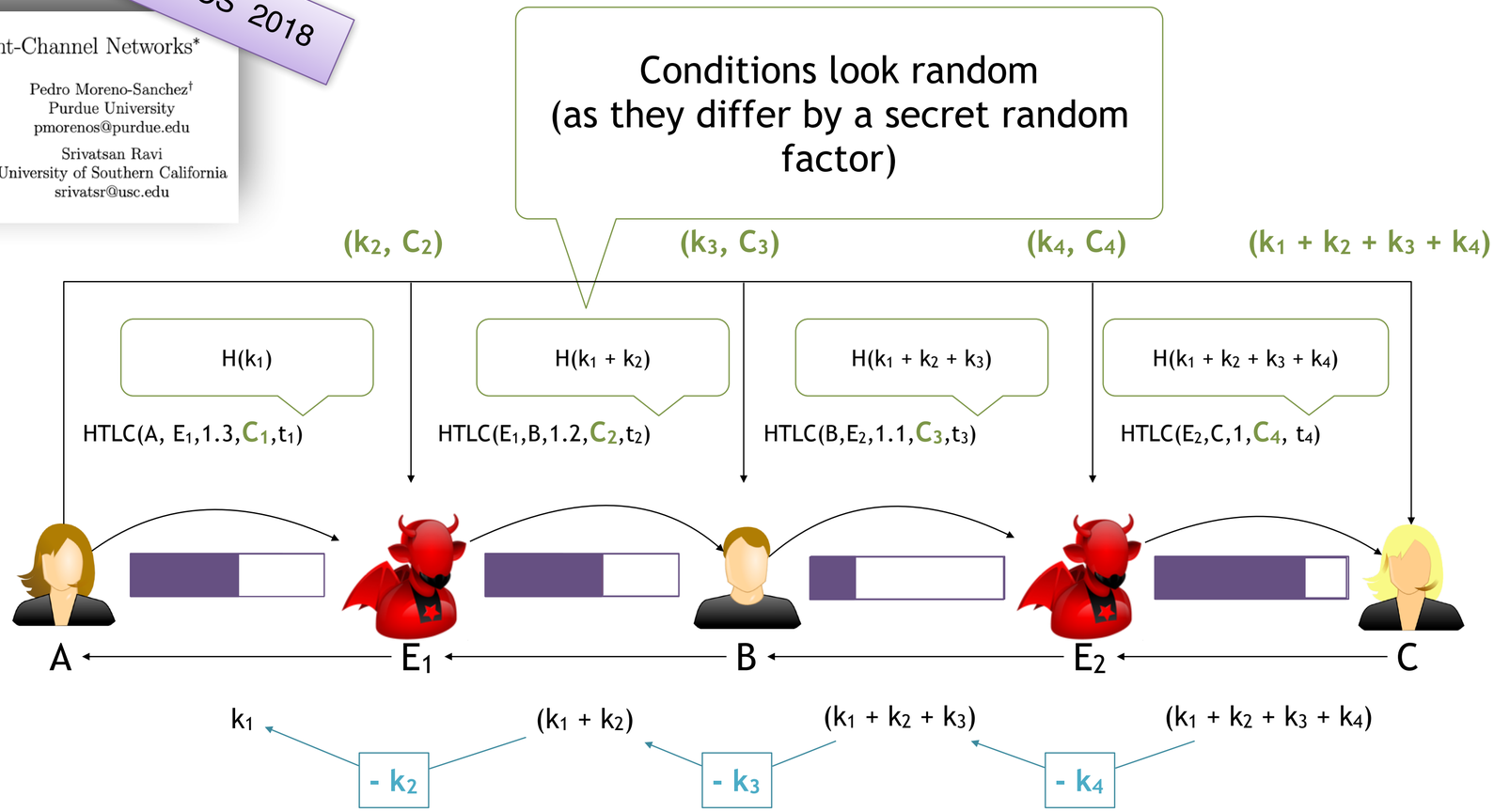
Giulio Malavolta†
Friedrich-Alexander-University Erlangen-Nürnberg
malavolta@cs.fau.de

Pedro Moreno-Sanchez†
Purdue University
pmorenos@purdue.edu

Aniket Kate
Purdue University
aniket@purdue.edu

Matteo Maffei
TU Wien
matteo.maffei@tuwien.ac.at

Srivatsan Ravi
University of Southern California
srivatsr@usc.edu



Fulgor

ACM CCS 2018

Concurrency and Privacy with Payment-Channel Networks*

Giulio Malavolta†
Friedrich-Alexander-University Erlangen-Nürnberg
malavolta@cs.fau.de

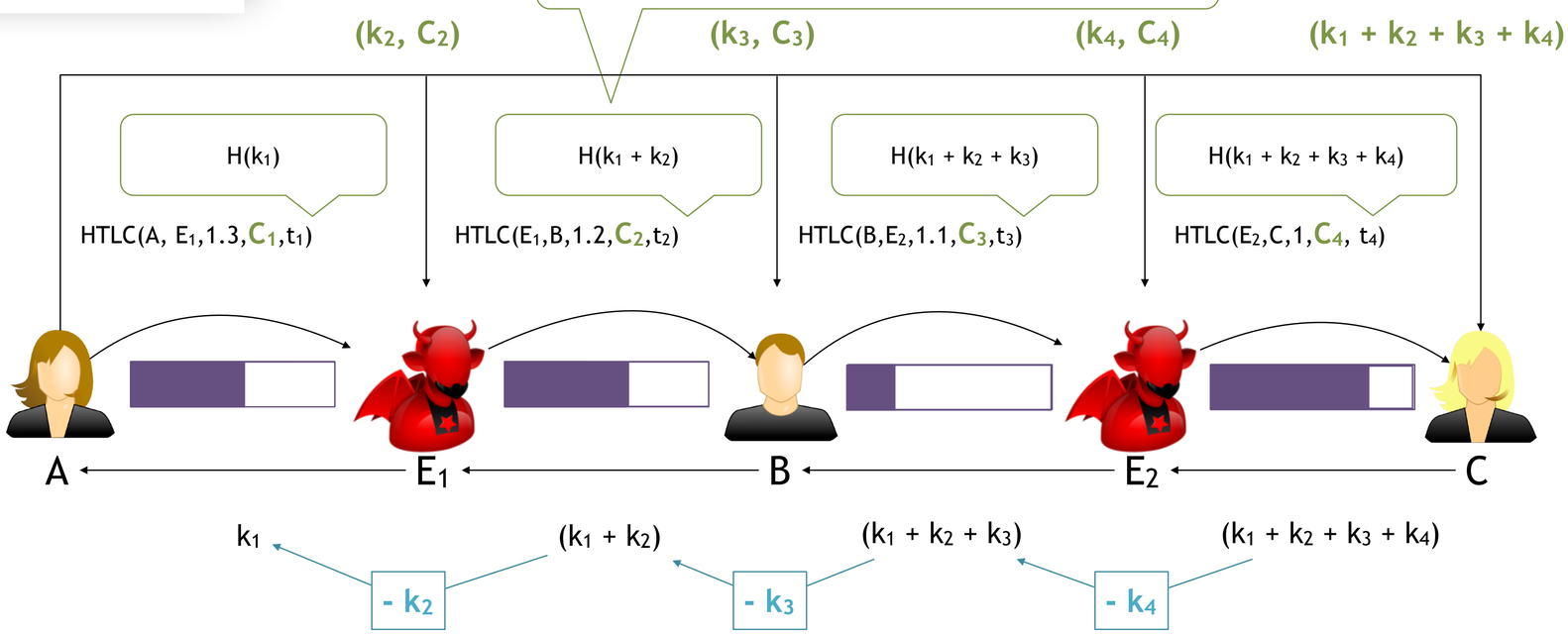
Pedro Moreno-Sanchez†
Purdue University
pmorenos@purdue.edu

Aniket Kate
Purdue University
aniket@purdue.edu

Matteo Maffei
TU Wien
matteo.maffei@tuwien.ac.at

Srivatsan Ravi
University of Southern California
srivatsr@usc.edu

Conditions look random
(as they differ by a secret random factor)



A valid key can only be extracted from a valid key for the right lock

Fulgor

ACM CCS 2018

Concurrency and Privacy with Payment-Channel Networks*

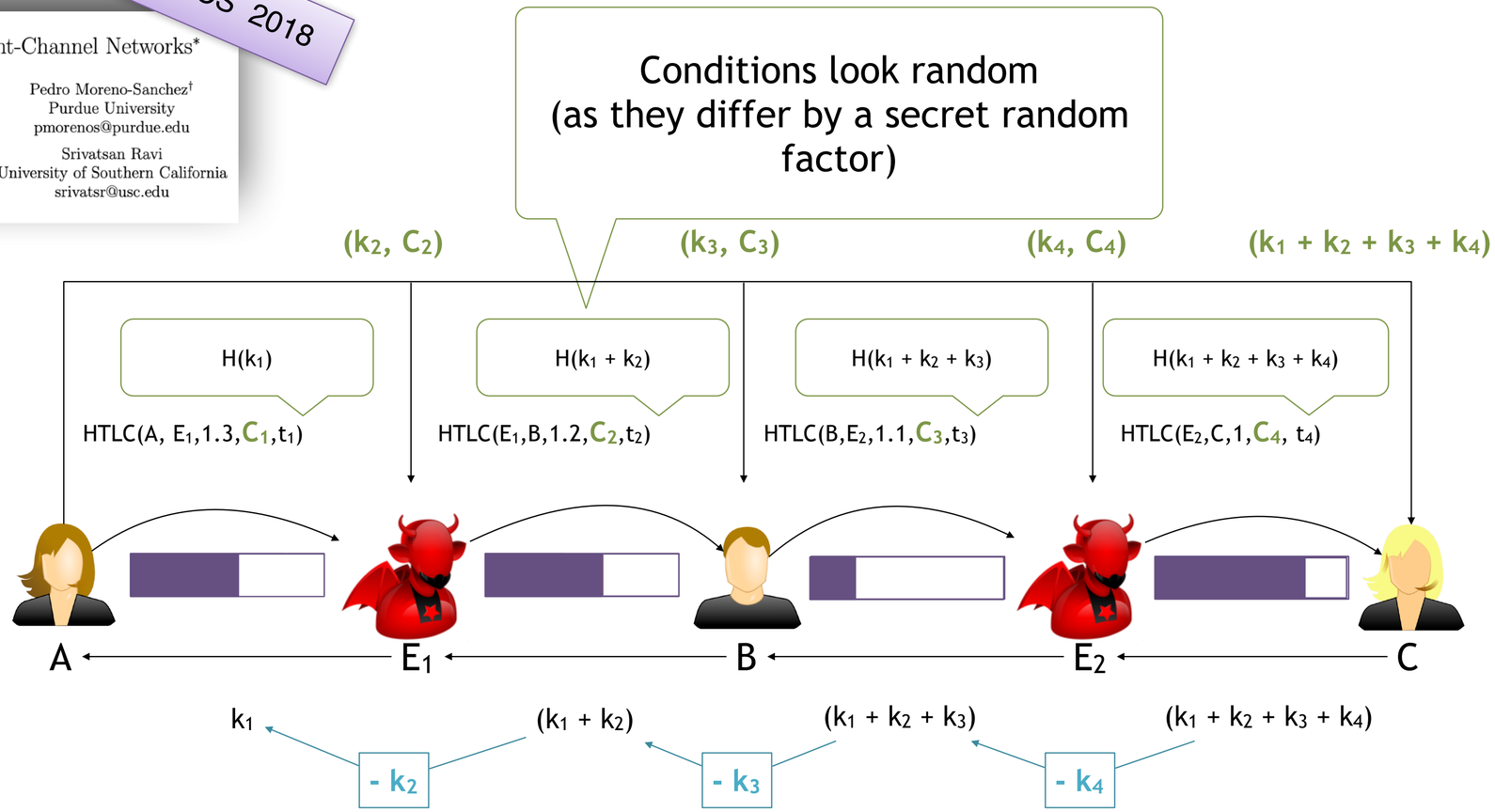
Giulio Malavolta†
Friedrich-Alexander-University Erlangen-Nürnberg
malavolta@cs.fau.de

Pedro Moreno-Sanchez†
Purdue University
pmorenos@purdue.edu

Aniket Kate
Purdue University
aniket@purdue.edu

Matteo Maffei
TU Wien
matteo.maffei@tuwien.ac.at

Srivatsan Ravi
University of Southern California
srivatsr@usc.edu



What if A is compromised?

…be
…key

Fulgor

ACM CCS 2018

Concurrency and Privacy with Payment-Channel Networks*

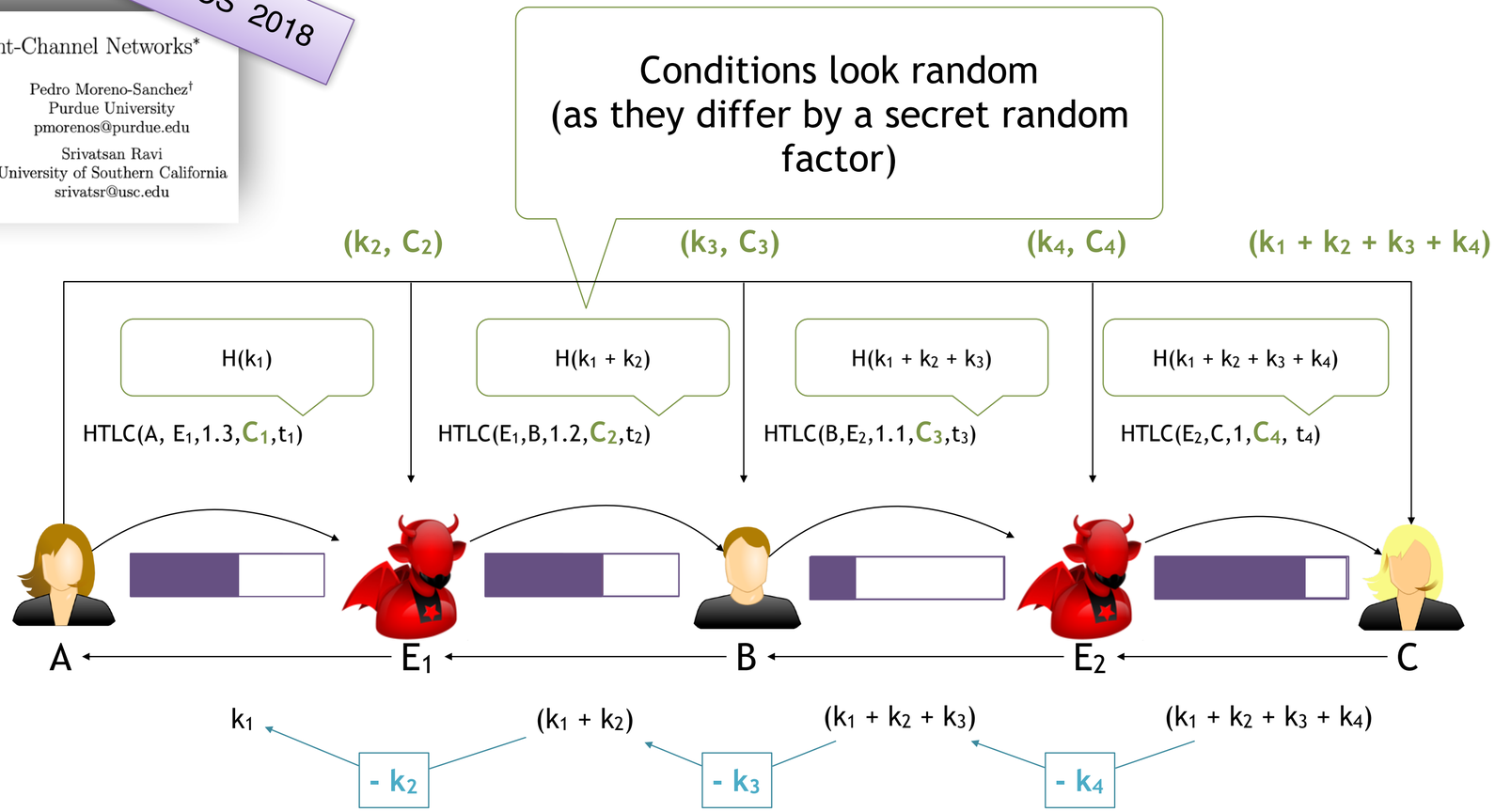
Giulio Malavolta†
Friedrich-Alexander-University Erlangen-Nürnberg
malavolta@cs.fau.de

Pedro Moreno-Sanchez†
Purdue University
pmorenos@purdue.edu

Aniket Kate
Purdue University
aniket@purdue.edu

Matteo Maffei
TU Wien
matteo.maffei@tuwien.ac.at

Srivatsan Ravi
University of Southern California
srivatsr@usc.edu



What if A is compromised?

be key

Intermediaries could lose money!

Fulgor

ACM CCS 2018

Concurrency and Privacy with Payment-Channel Networks*

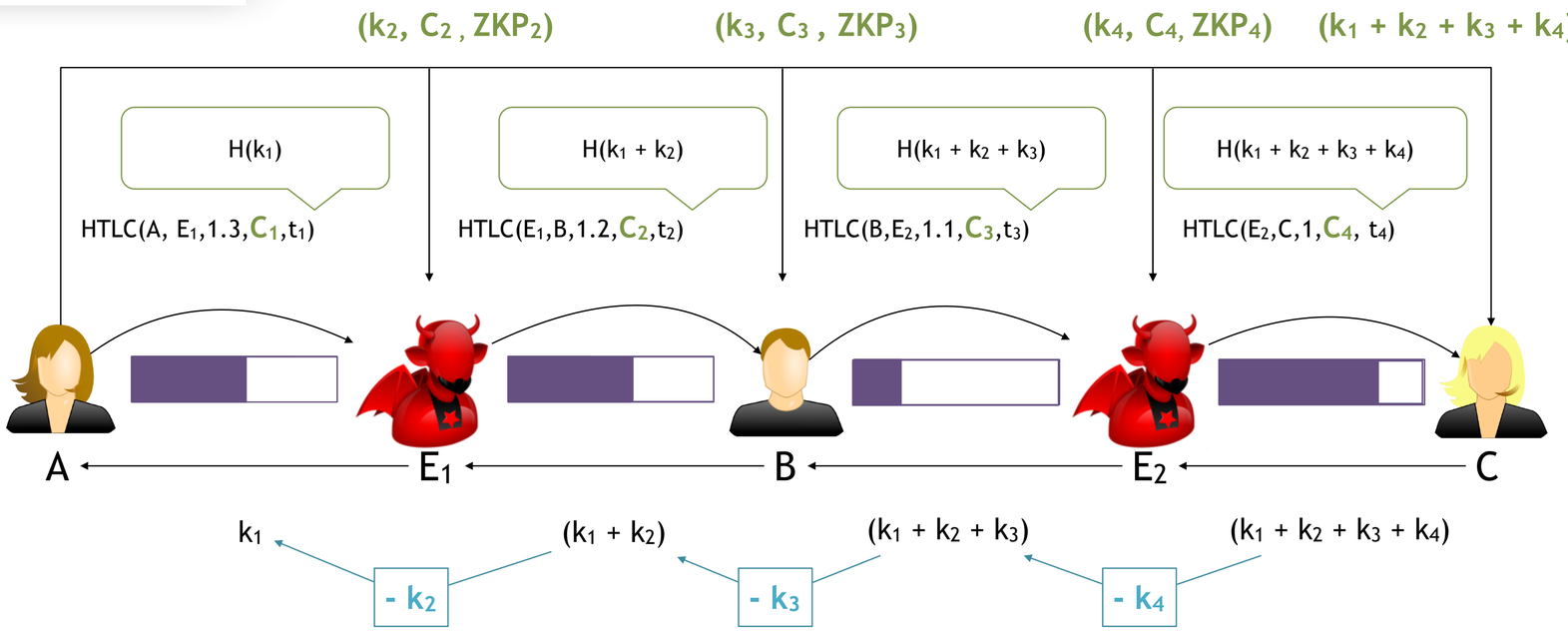
Giulio Malavolta†
Friedrich-Alexander-University Erlangen-Nürnberg
malavolta@cs.fau.de

Pedro Moreno-Sanchez†
Purdue University
pmorenos@purdue.edu

Aniket Kate
Purdue University
aniket@purdue.edu

Matteo Maffei
TU Wien
matteo.maffei@tuwien.ac.at

Srivatsan Ravi
University of Southern California
srivatsr@usc.edu



A sends a Zero-Knowledge Proof that C_i is well formed

ZKP_i= $\{ \exists x \,.\, C_{i-1} = H(x) \wedge C_i = H(k_i + x) \}$

Fulgor

ACM CCS 2018

Concurrency and Privacy with Payment-Channel Networks*

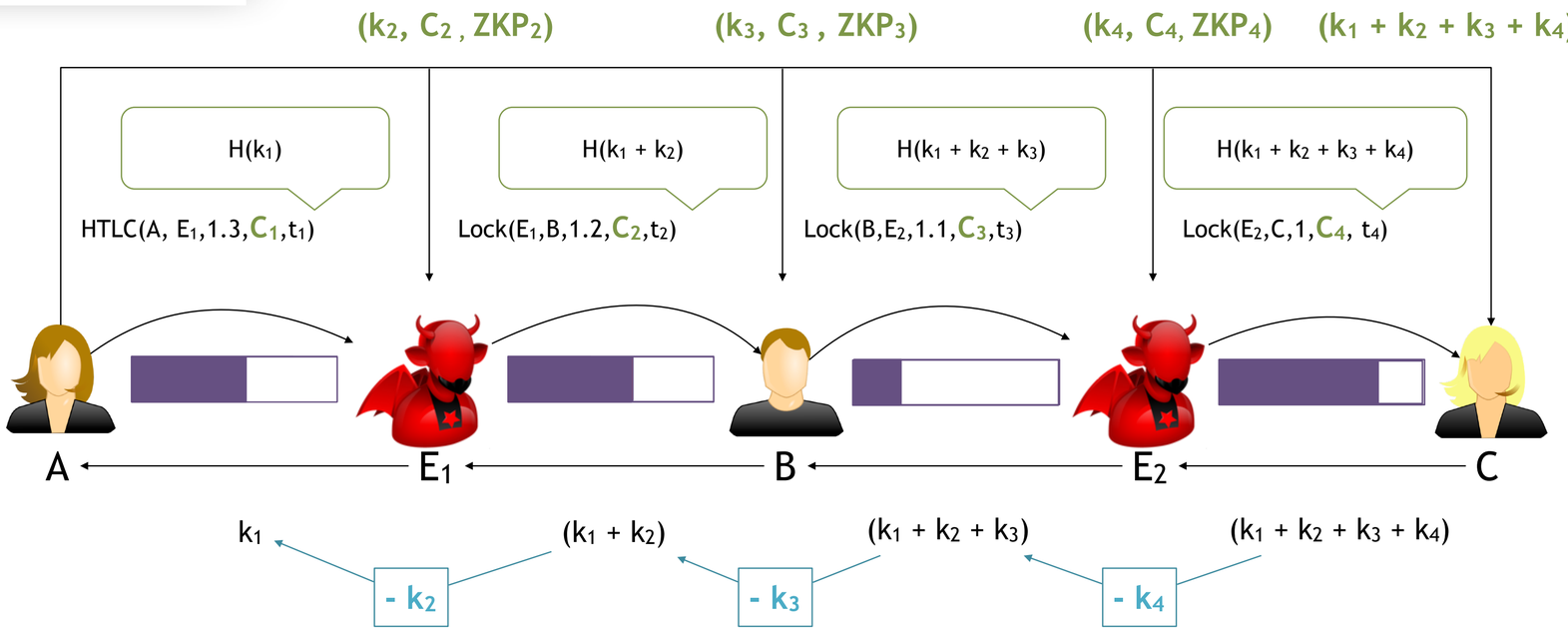
Giulio Malavolta†
Friedrich-Alexander-University Erlangen-Nürnberg
malavolta@cs.fau.de

Pedro Moreno-Sanchez†
Purdue University
pmorenos@purdue.edu

Aniket Kate
Purdue University
aniket@purdue.edu

Matteo Maffei
TU Wien
matteo.maffei@tuwien.ac.at

Srivatsan Ravi
University of Southern California
srivatsr@usc.edu



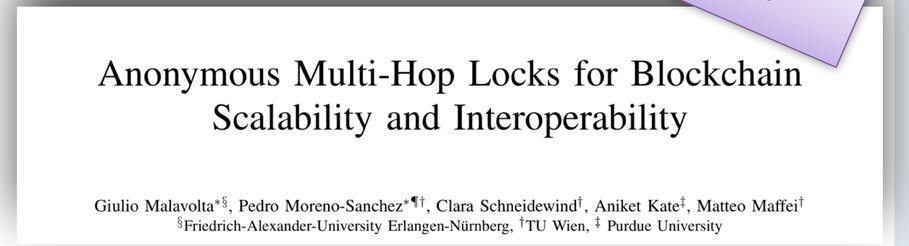
Achieved Properties

1. **Atomicity**:
   If a user's right lock gets opened, he can open his left lock
2. **Consistency**:
   A user can open his left lock only if his right lock was released
3. **Relationship Anonymity:**
   A user learns about no other participant of the payment path than his direct neighbours

No coin loss	No Wormhole Attacks	Privacy

Anonymous Multi-Hop Locks (AMHL)

NDSS 2019



- In a follow-up work, we integrated the randomness in the signature itself (adaptor signatures), getting rid of HTCLs
  - Constructions for ECDSA and Schnorr
  - Implemented in the Lightning Network https://github.com/cfromknecht/tpec
  - Compatibility with currencies without HTLCs (e.g., Monero)
  - Transactions look the same as normal Bitcoin payments (fungibility)
  - More efficient (Fulgor 5 MB communication, AMHL <500 bytes and 50ms computation)
  - Originated the Point Time Locked Constracts (PTLC) BIP proposal

Adaptor Signatures

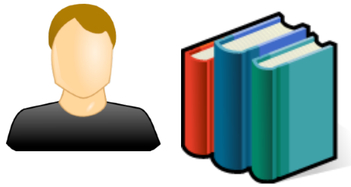
- Invented by the cryptographic community (Polstra, Blockstream)
- An adaptor signature scheme is essentially a two-step signing algorithm bound to a secret, with each step corresponding to a property (adaptability and extractability):
  - a partial signature is generated such that it can be completed only by a party knowing a certain secret (adaptability)
  - the complete signature reveals such a secret (extractability)
- We gave the first construction for ECDSA (used in Bitcoin)
- For a formal definition look at our paper:

Scriptless Scripts

Scriptless Scripts

5

Alice
(sk_A)



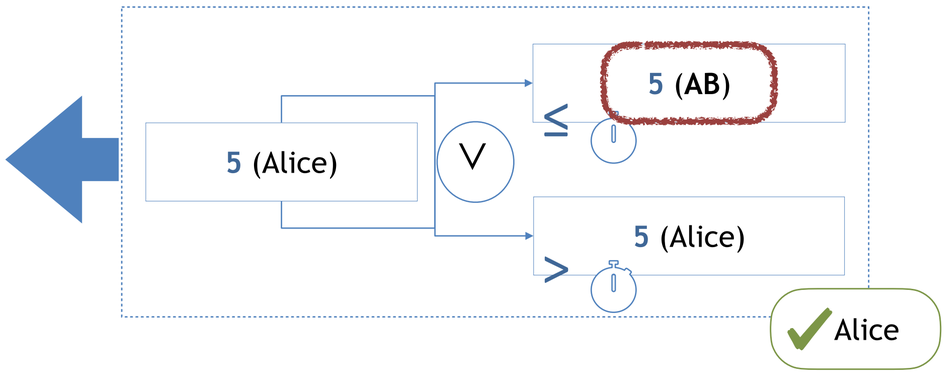
Bob
(sk_B)

AB

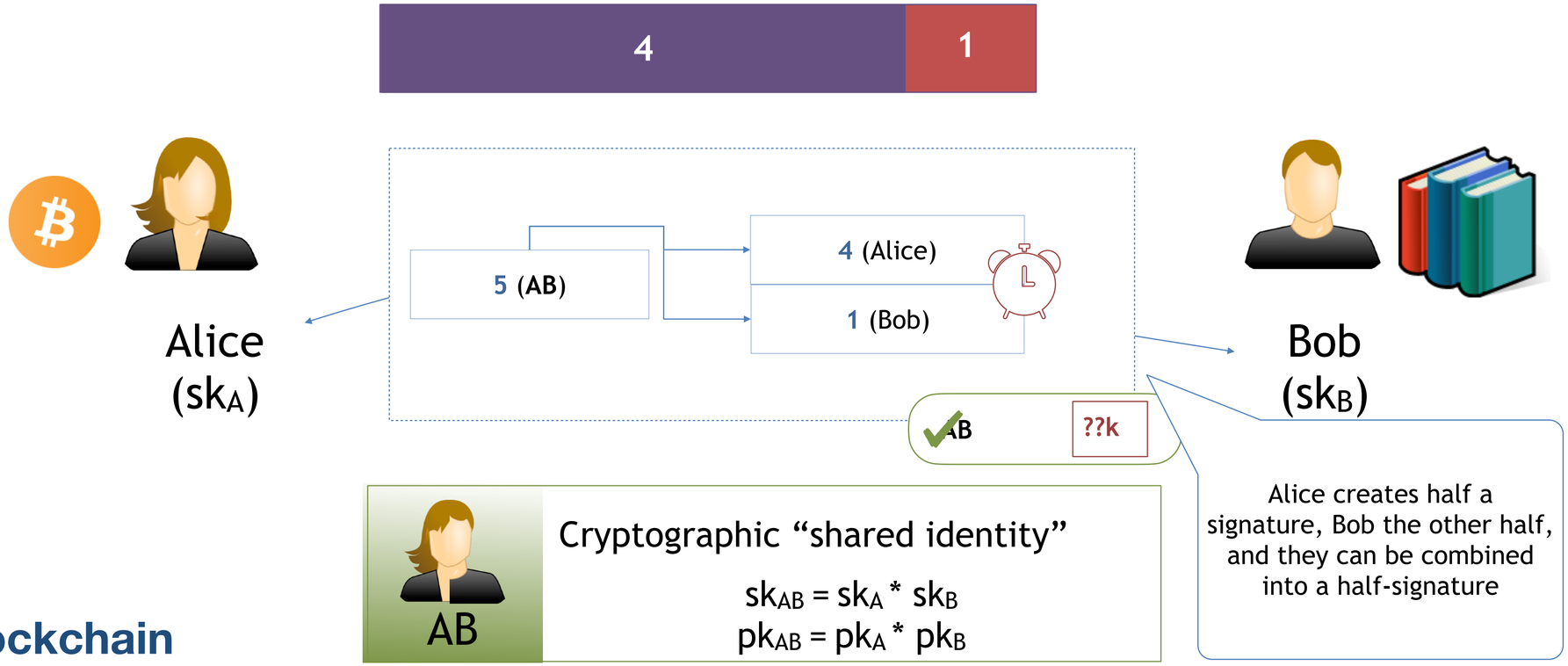
Cryptographic “shared identity”

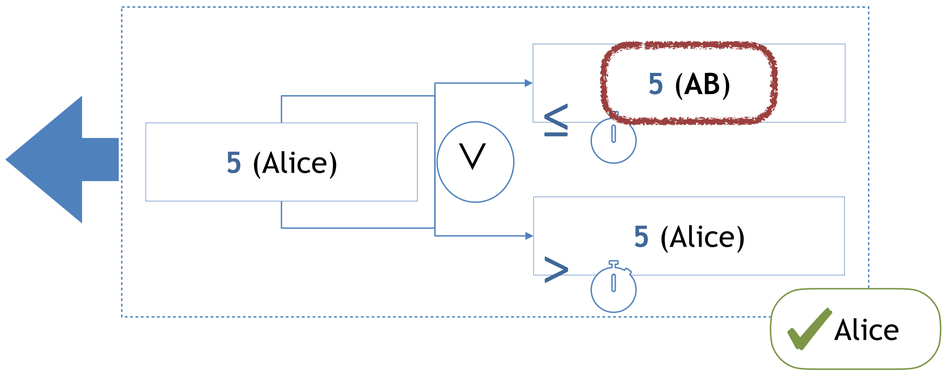
$sk_{AB} = sk_A * sk_B$
$pk_{AB} = pk_A * pk_B$

Blockchain



Scriptless Scripts

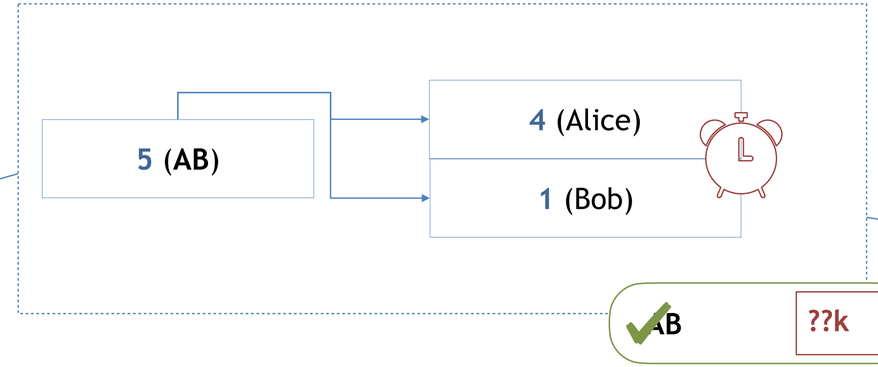




Scriptless Scripts

Alice can retrieve secret **k** from full signature

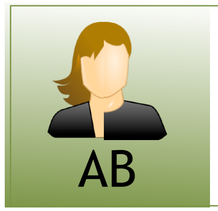
Bob gets sufficient information for checking that the "half signature" produced by Alice and Bob can be completed to a valid signature given **k**



Alice
(sk_A)

Bob
(sk_B)

Alice creates half a signature, Bob the other half, and they can be combined into a half-signature

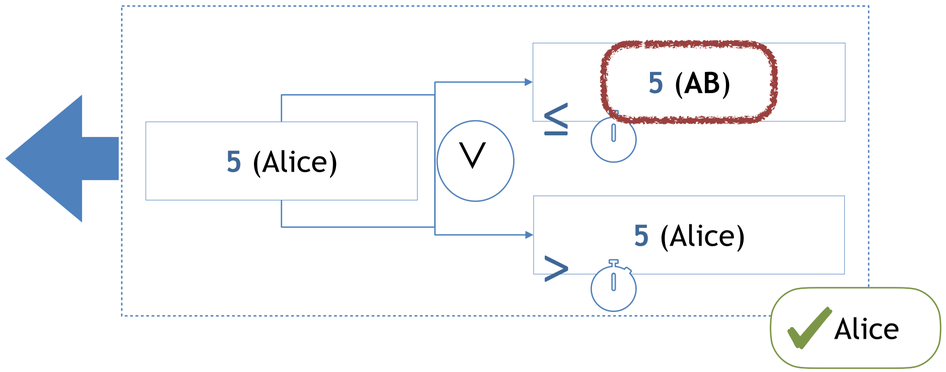


Cryptographic "shared identity"

$sk_{AB} = sk_A * sk_B$

$pk_{AB} = pk_A * pk_B$

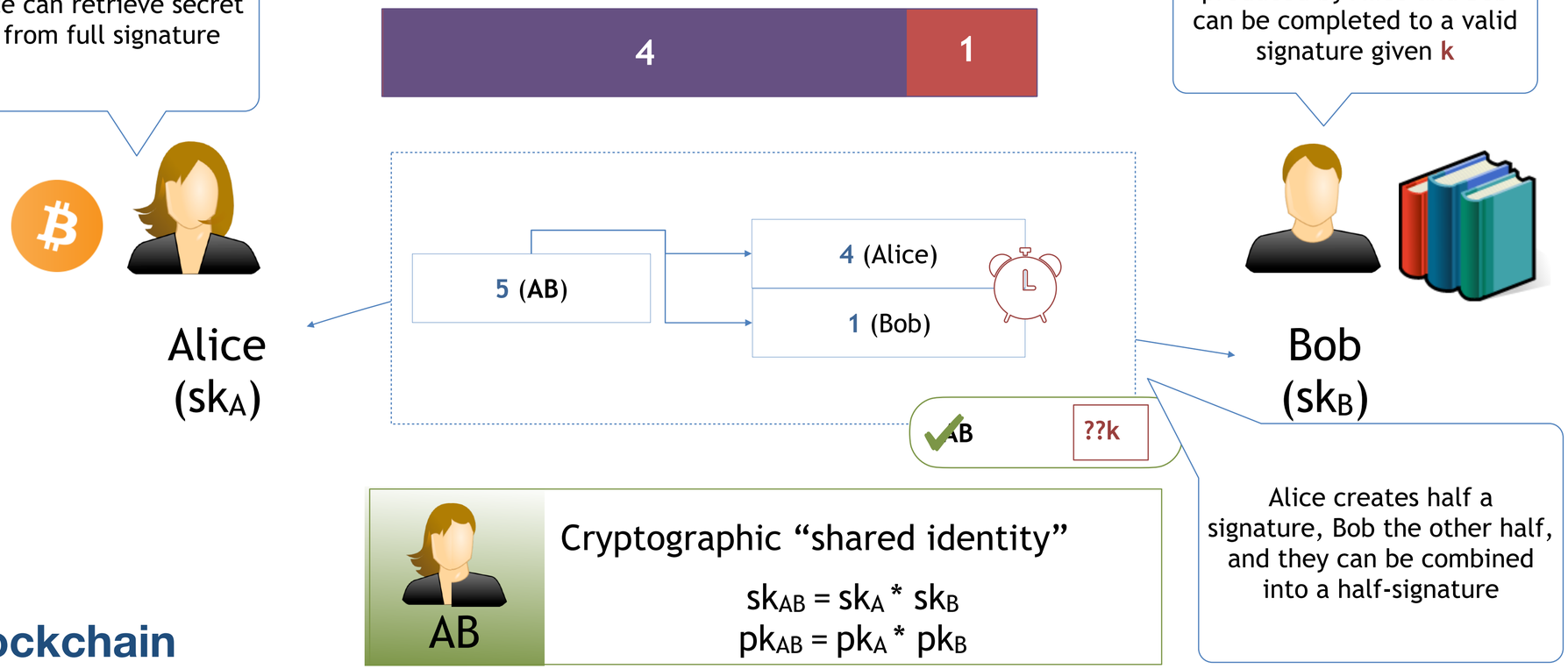
Blockchain



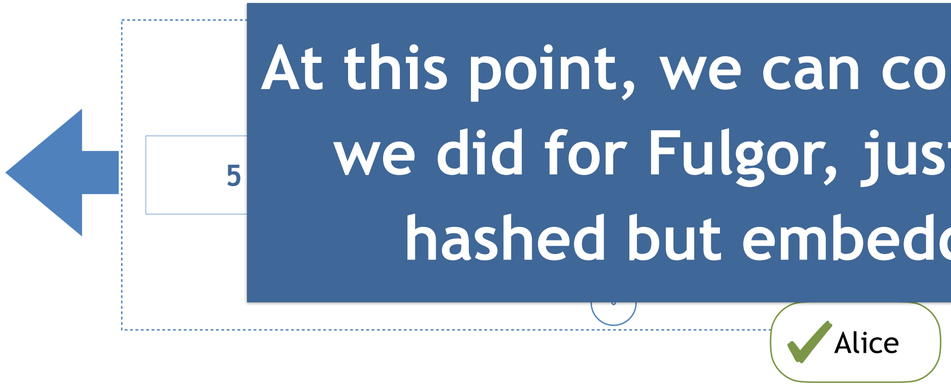
Scriptless Scripts

Alice can retrieve secret **k** from full signature

Bob gets sufficient information for checking that the "half signature" produced by Alice and Bob can be completed to a valid signature given **k**



Blockchain



At this point, we can construct a payment path like we did for Fulgor, just that the secrets are not hashed but embedded into the signatures

Schnorr-based Adaptor Signature

$$sk_I = x_I$$
$$pk_I = x_I \cdot G$$
$$R_I = r_I \cdot G$$

Schnorr Signature for *I*

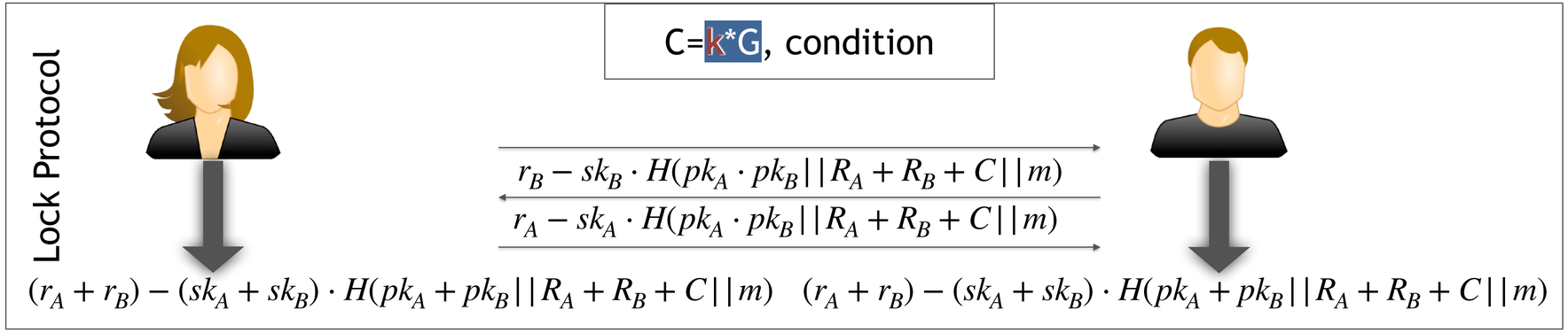
$$sig(r_I, m, sk, pk) = (R_I, r_I - sk_i \cdot H(pk_i || R_I || m))$$

Schnorr-based Adaptor Signature

$$sk_I = x_I$$
$$pk_I = x_I \cdot G$$
$$R_I = r_I \cdot G$$

Schnorr Signature for *I*

$$sig(r_I, m, sk, pk) = (R_I, r_I - sk_i \cdot H(pk_i || R_I || m))$$



Schnorr-based Adaptor Signature

$$sk_I = x_I$$
$$pk_I = x_I \cdot G$$
$$R_I = r_I \cdot G$$

Schnorr Signature for *I*

$$sig(r_I, m, sk, pk) = (R_I, r_I - sk_i \cdot H(pk_i || R_I || m))$$

Alice can retrieve secret **k** from full signature

Bob gets sufficient information for checking that the "half signature" produced by Alice and Bob can be completed to a valid signature given **k**

Lock Protocol

C=k*G, condition

$r_B - sk_B \cdot H(pk_A \cdot pk_B || R_A + R_B + C || m)$

$r_A - sk_A \cdot H(pk_A \cdot pk_B || R_A + R_B + C || m)$

$(r_A + r_B) - (sk_A + sk_B) \cdot H(pk_A + pk_B || R_A + R_B + C || m)$

$(r_A + r_B) - (sk_A + sk_B) \cdot H(pk_A + pk_B || R_A + R_B + C || m)$

Schnorr-based Adaptor Signature

$$sk_I = x_I$$
$$pk_I = x_I \cdot G$$
$$R_I = r_I \cdot G$$

Schnorr Signature for *I*

$$sig(r_I, m, sk, pk) = (R_I, r_I - sk_i \cdot H(pk_i || R_I || m))$$

Alice can retrieve secret **k** from full signature

Bob gets sufficient information for checking that the "half signature" produced by Alice and Bob can be completed to a valid signature given **k**

Lock Protocol

C=k*G, condition

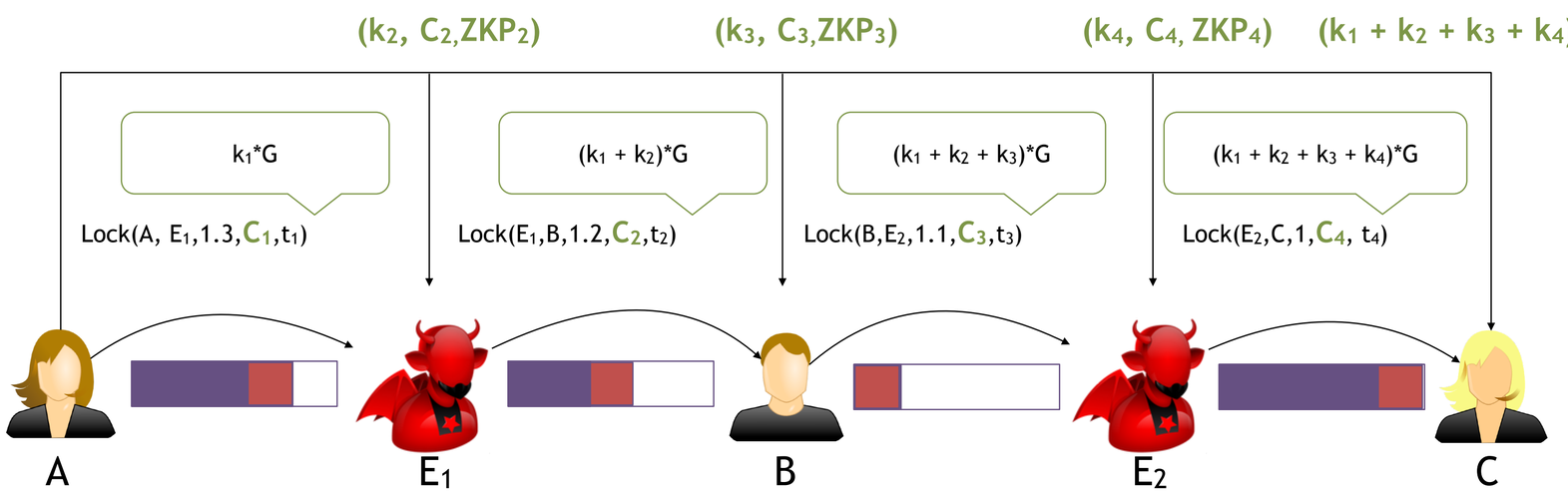
$r_B - sk_B \cdot H(pk_A \cdot pk_B || R_A + R_B + C || m)$

$r_A - sk_A \cdot H(pk_A \cdot pk_B || R_A + R_B + C || m)$

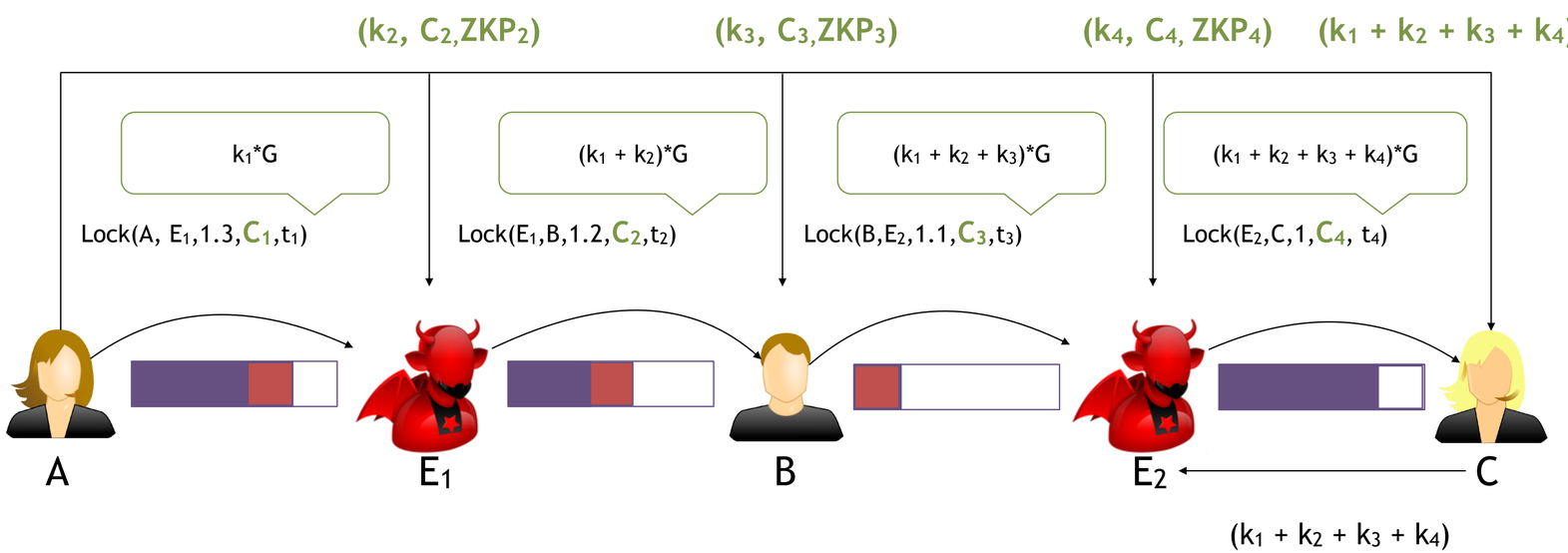
$(r_A + r_B) - (sk_A + sk_B) \cdot H(pk_A$ $pk_B || R_A + R_B + C || m)$ $(r_A + r_B) - (sk_A + sk_B) \cdot H(pk_A + pk_B || R_A + R_B + C || m)$

(Efficient) ZKPs are required to show that half signatures are well-formed with respect to the public R_A and R_B

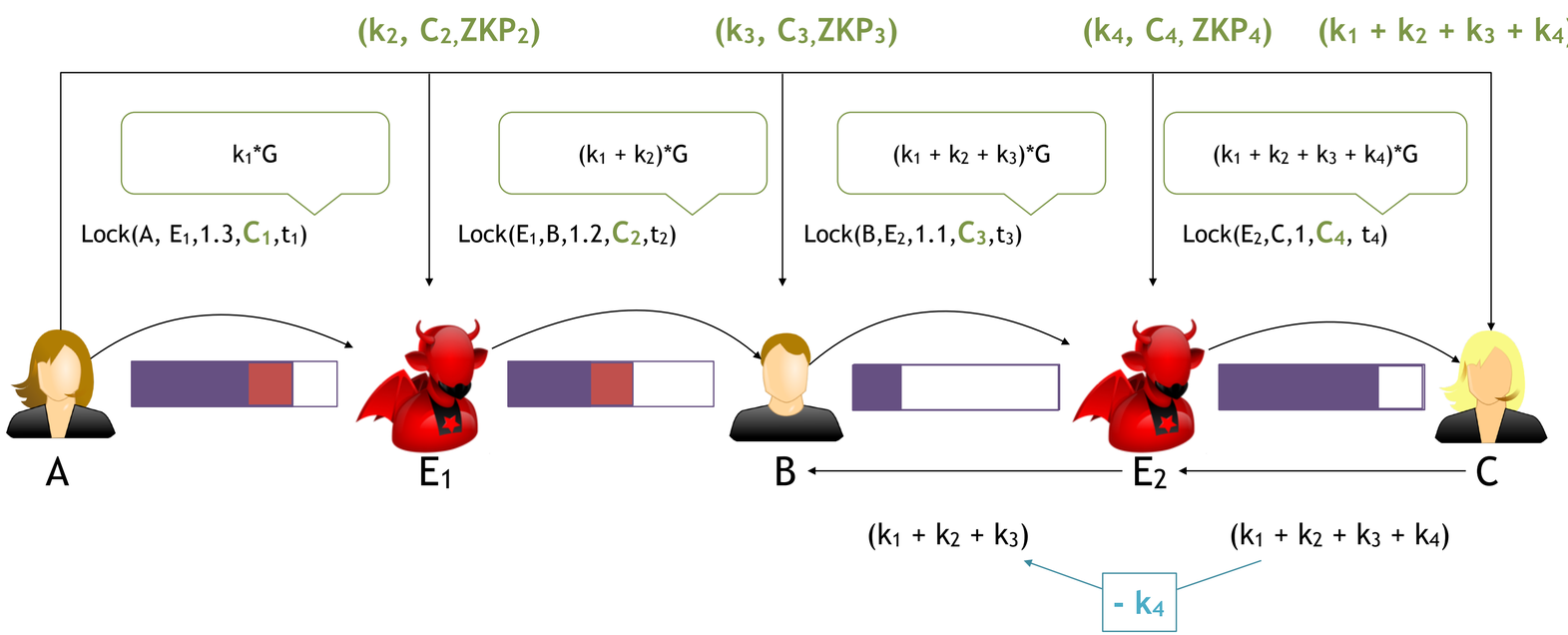
Extension to Multi-hop Locks



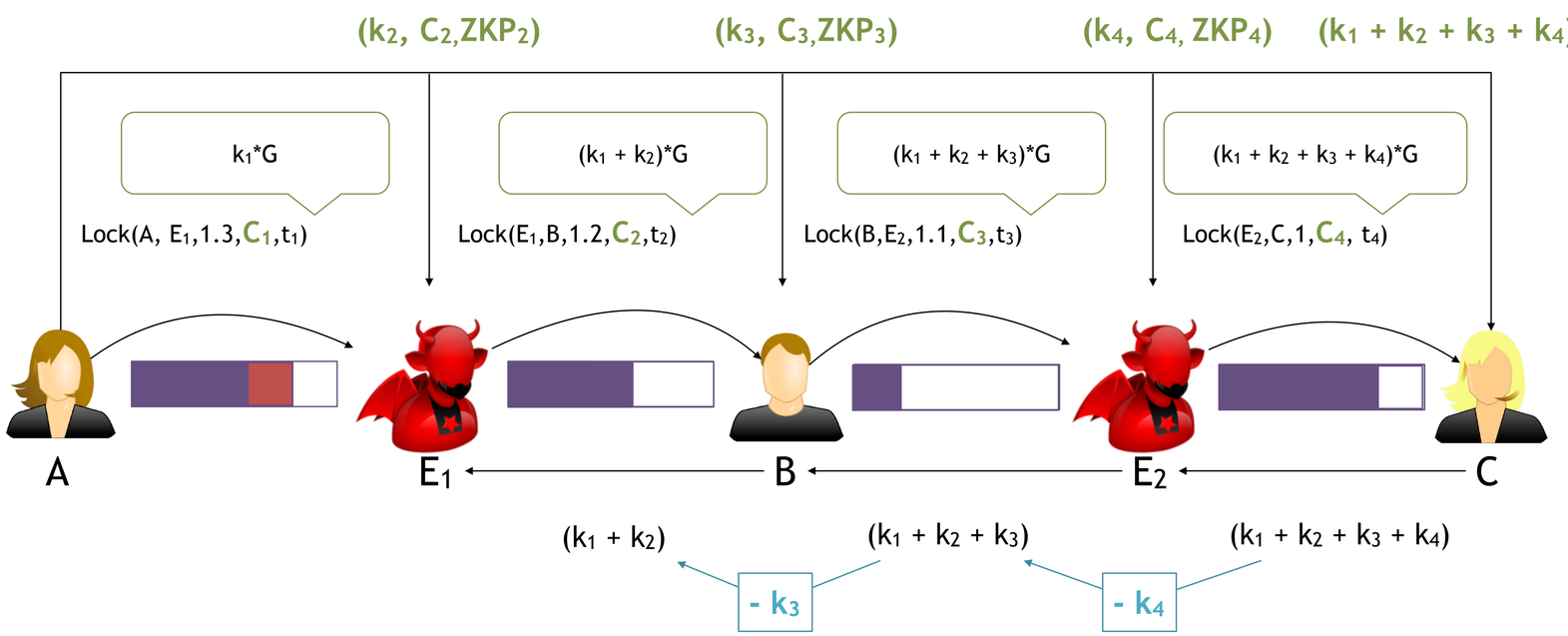
Extension to Multi-hop Locks



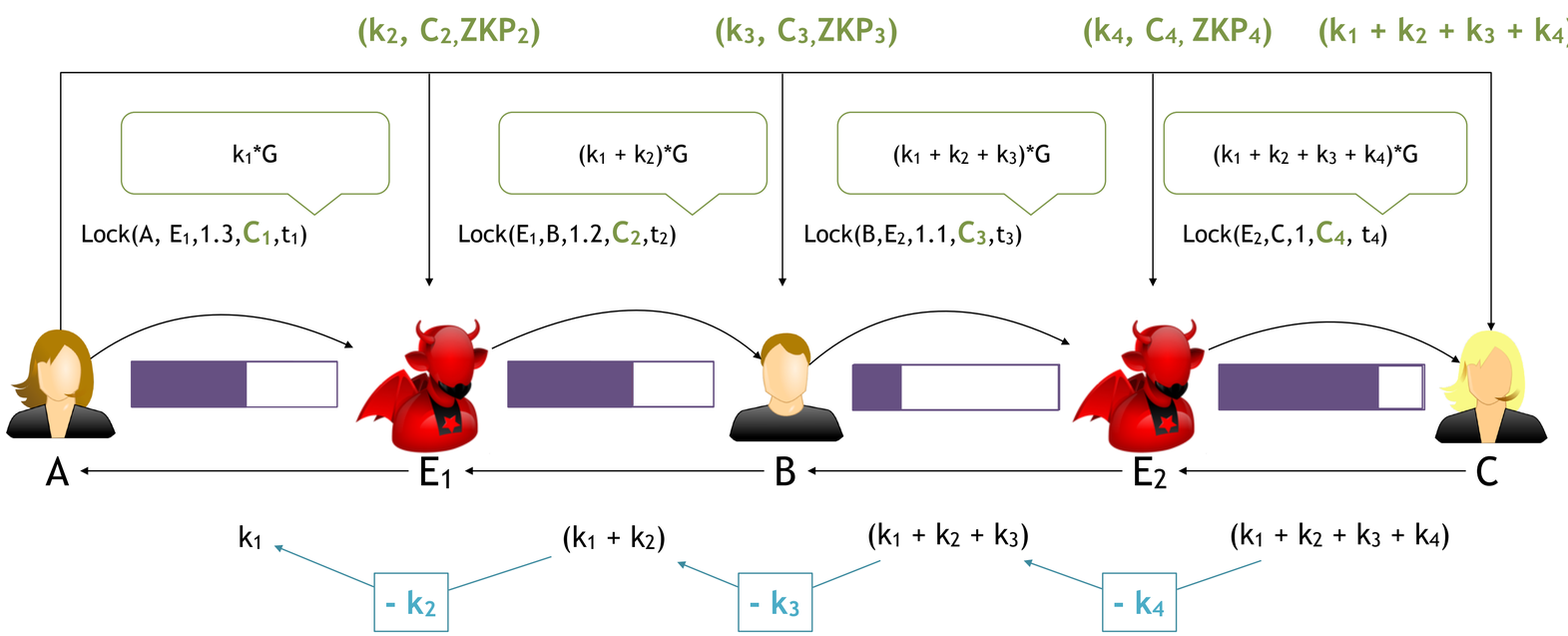
Extension to Multi-hop Locks



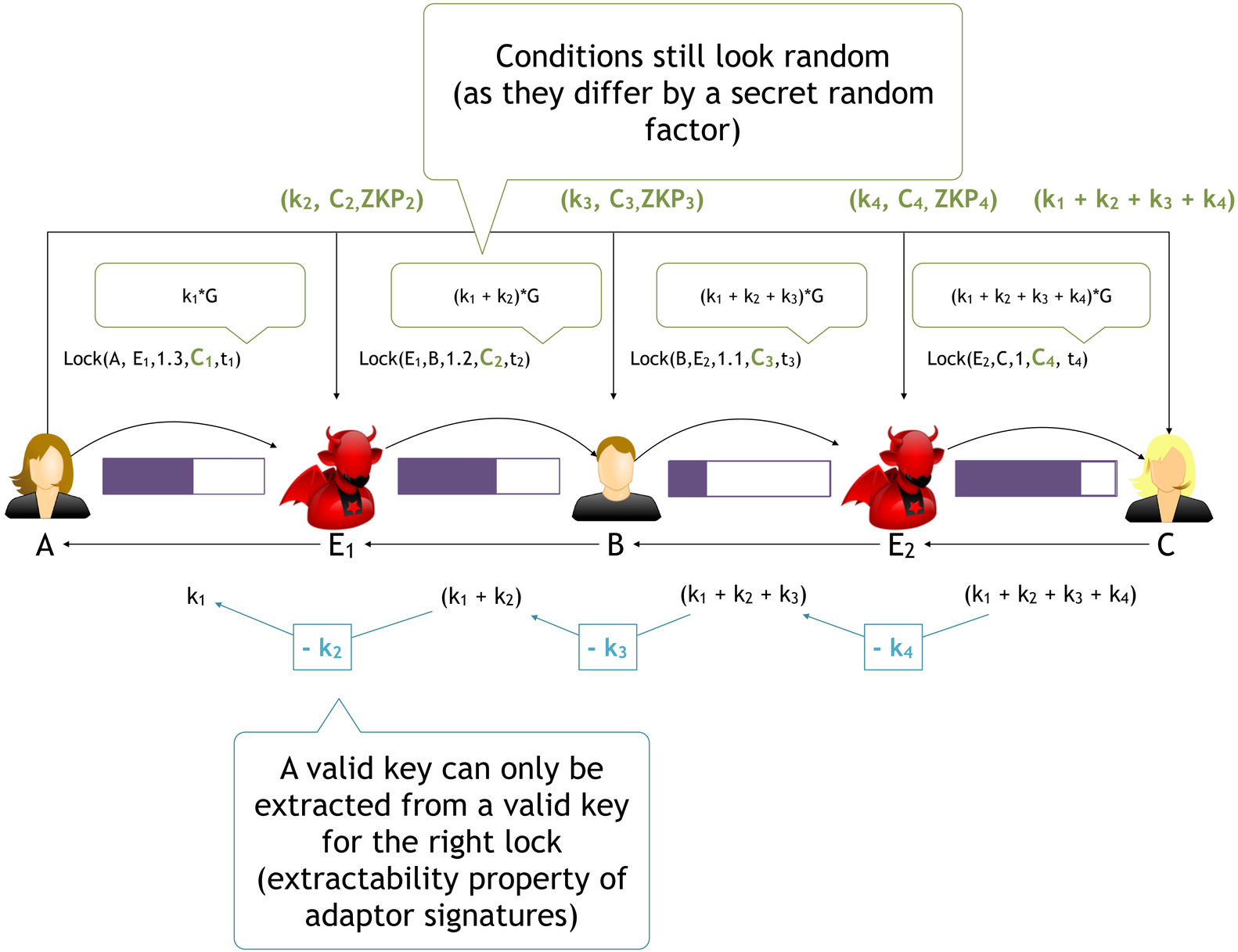
Extension to Multi-hop Locks



Extension to Multi-hop Locks

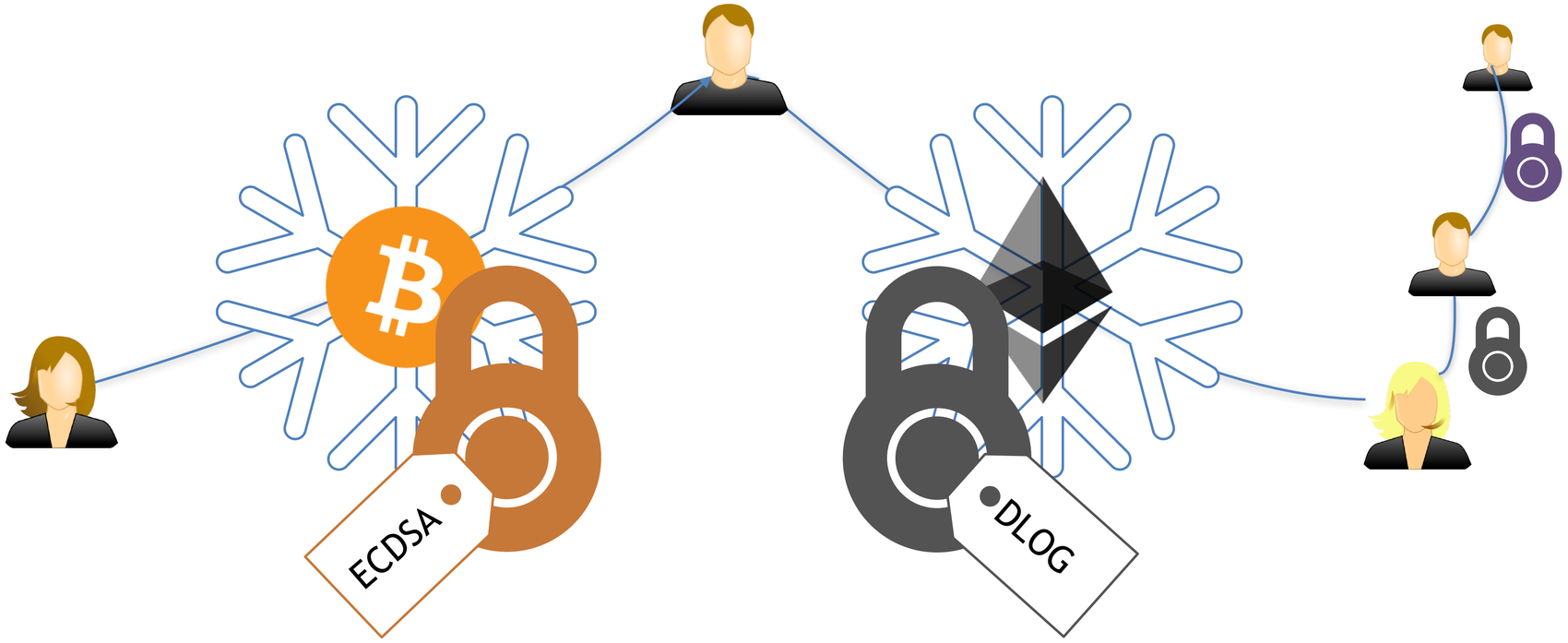


Extension to Multi-hop Locks

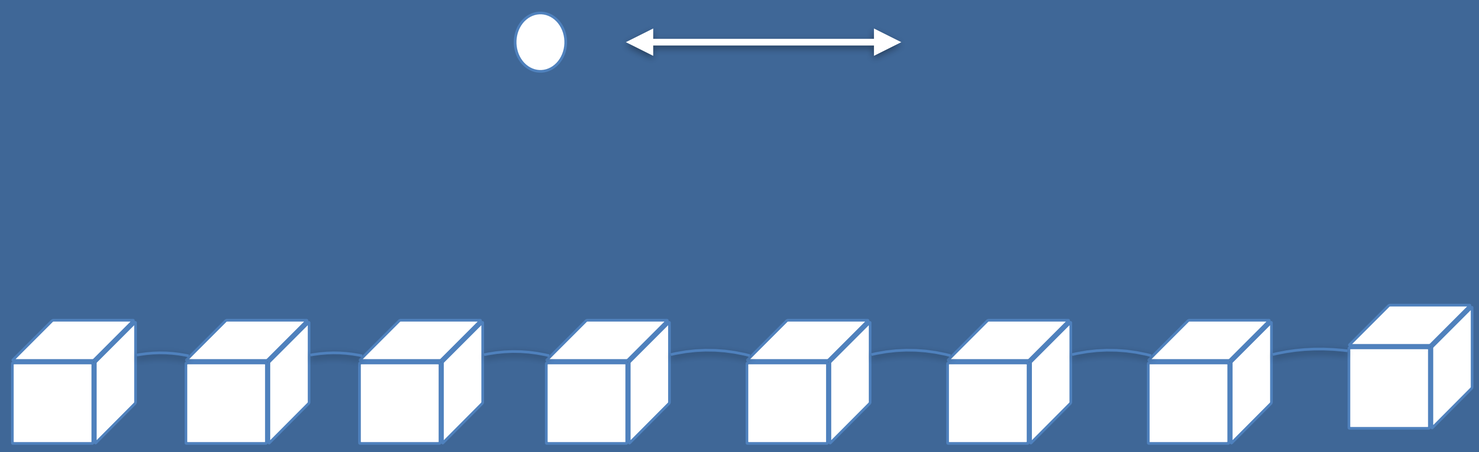


Interoperability

- AMHLs are suitable for cross-currency usage, even with different primitive instantiations
  - ✓ Inter-currency payment channels
  - ✓ Atomic swaps



Watchtowers and sleepy channels



Allow nodes to go offline without losing money

Handling offline nodes

- What if the end-point of a channel is offline?
  - The other end-point can post an old state without being punished…
- Watchtowers: third parties monitoring the blockchain on behalf of offline users
- Challenges:
  - Privacy: avoid to leak all transactions to the watchtower
  - Participation and trust: pay watchtowers if they do their job and punish them otherwise
- Sleepy channels: get rid of watchtowers asking parties to be online only at predetermined time slots

Financial Crypto 2020

Cerberus Channels:
Incentivizing Watchtowers for Bitcoin

Georgia Avarikioti[1], Orfeas Stefanos Thyfronitis Litos[2], and Roger Wattenhofer[1]

[1] ETH Zürich
{zetavar,wattenhofer}@ethz.ch
[2] University of Edinburgh
o.thyfronitis@ed.ac.uk

ACM CCS 2022

Sleepy Channels:
Bitcoin-Compatible Bi-directional Payment Channels without Watchtowers

Lukas Aumayr[1]
TU Wien
lukas.aumayr@tuwien.ac.at

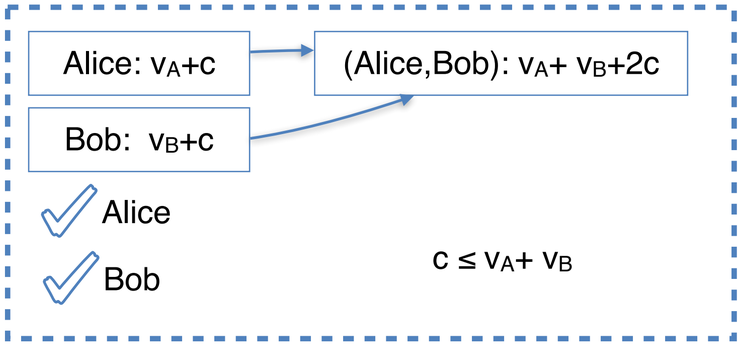
Sri AravindaKrishnan Thyagarajan[1,2]
Carnegie Mellon University
t.srikrishnan@gmail.com

Giulio Malavolta
Max Planck Institute for Security and Privacy
giulio.malavolta@hotmail.it

Pedro Moreno-Sánchez
IMDEA Software Institute
pedro.moreno@imdea.org

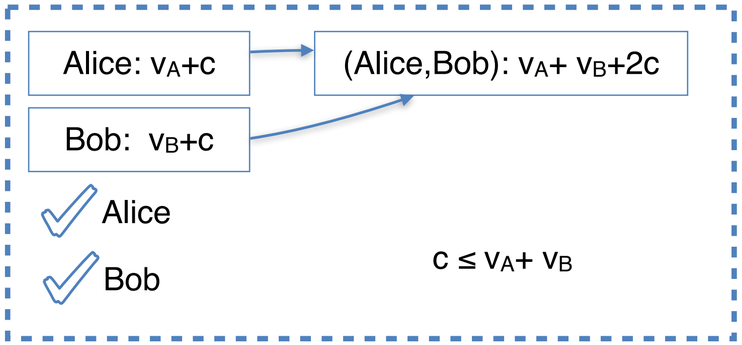
Matteo Maffei
Christian Doppler Laboratory Blockchain Technologies for the Internet of Things, TU Wien
matteo.maffei@tuwien.ac.at

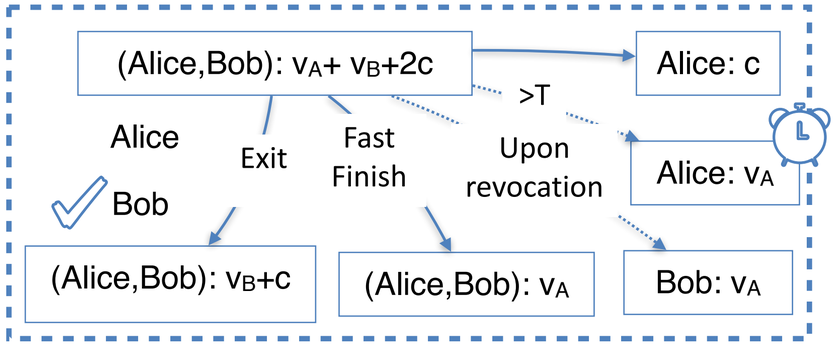
Sleepy Channels



Alice and Bob put a collateral each,
which coincides with the channel capacity
(can be configured depending on trust)

Sleepy Channels

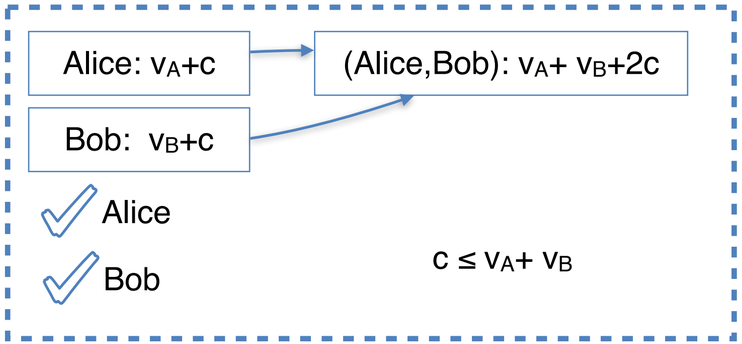


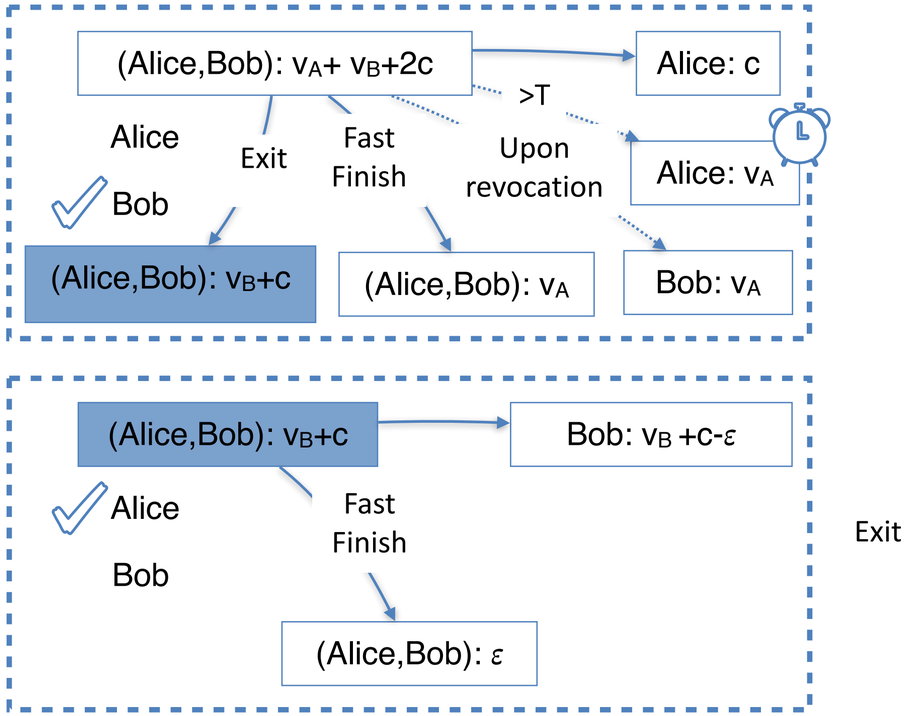


Alice can get her collateral back immediately,
for her money she has to wait until an absolute timelock (channel lifetime),
before which she can be punished if the transaction is old
(Bob has to come online only before T)

We also have a way for Bob to get her money and collateral immediately
(Exit) and then for Alice to get her money (Fast Finish)

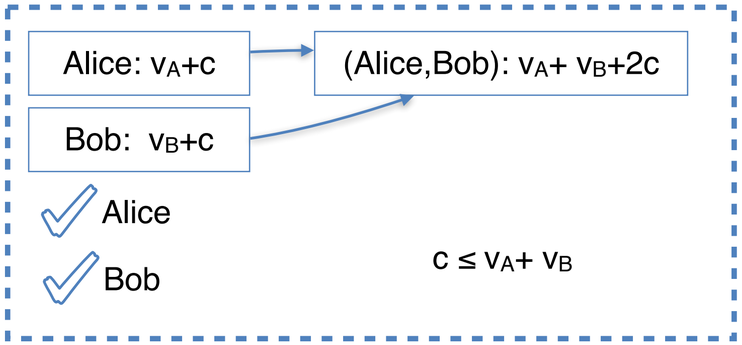
Sleepy Channels



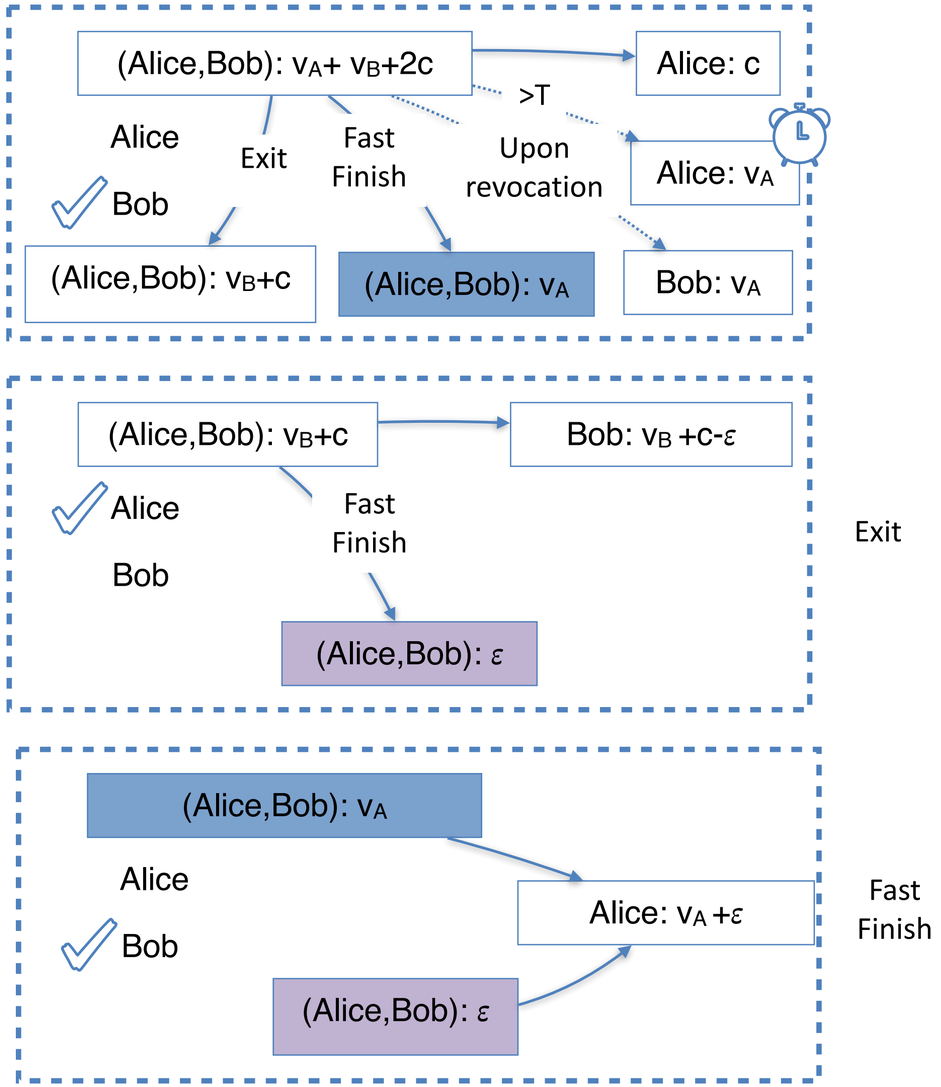


The Exit transaction is pre-signed by Alice, so Bob can post it and get back its money plus collateral, minus a ε : in fact, Bob has an interest to do it, not to lock a collateral larger than Alice's funding

Sleepy Channels



Once Bob is done, Alice can get her money immediately through the Fast Finish transaction



Extensions

- Alice and Bob can update the lifetime of the channel, and also top-up its capacity, with one on-chain transaction (similar to the Splicing protocol in Lightning Network)
- One can get rid of the absolute timelock for better compatibility (e.g., with currencies without timelock scripts like Monero) through verifiable time signatures (VTS)

Blitz

Blitz: Secure Multi-Hop Payments Without Two-Phase Commits*

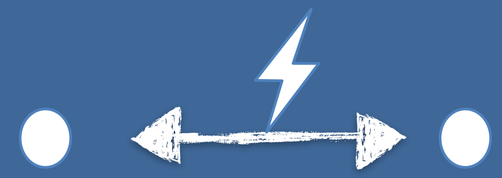
Lukas Aumayr
TU Wien
lukas.aumayr@tuwien.ac.at

Pedro Moreno-Sanchez
IMDEA Software Institute
pedro.moreno@imdea.org

Aniket Kate
Purdue University
aniket@purdue.edu

Matteo Maffei
TU Wien
matteo.maffei@tuwien.ac.at

Usenix Security 2021



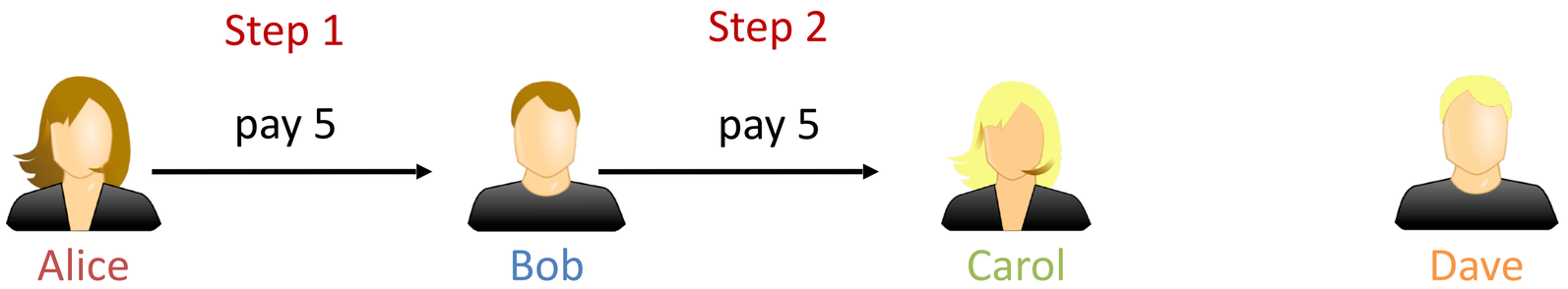
Make payments fast and avoid griefing attacks

Multi-hop payments in one round: Attempt 1

Again: Alice wants to pay 5 coins to Dave, via Bob and Carol

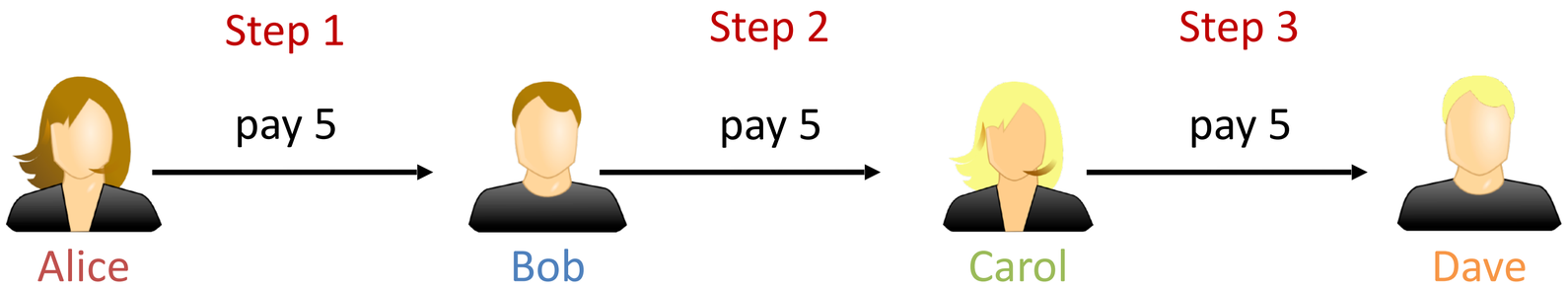
Multi-hop payments in one round: Attempt 1

Again: Alice wants to pay 5 coins to Dave, via Bob and Carol



Multi-hop payments in one round: Attempt 1

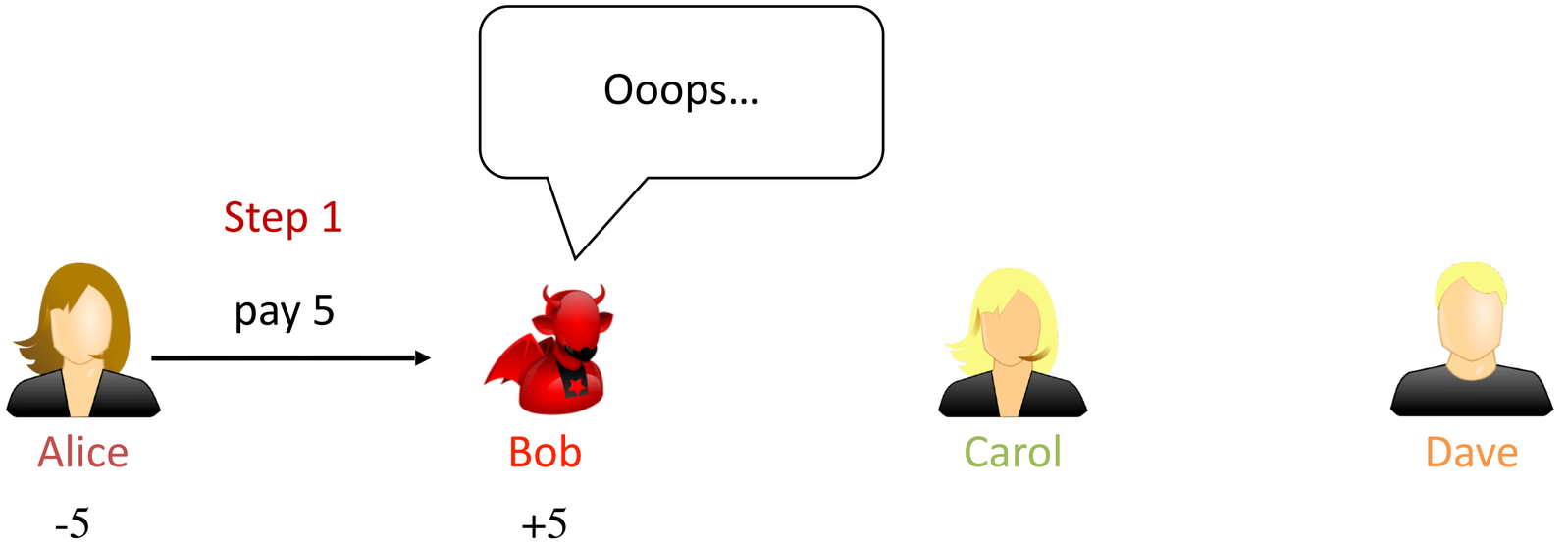
Again: Alice wants to pay 5 coins to Dave, via Bob and Carol



=> Actually used in: Interledger Payments [TS15]

[TS15] S. Thomas and E. Schwartz, "A Protocol for Interledger Payments," *2015*

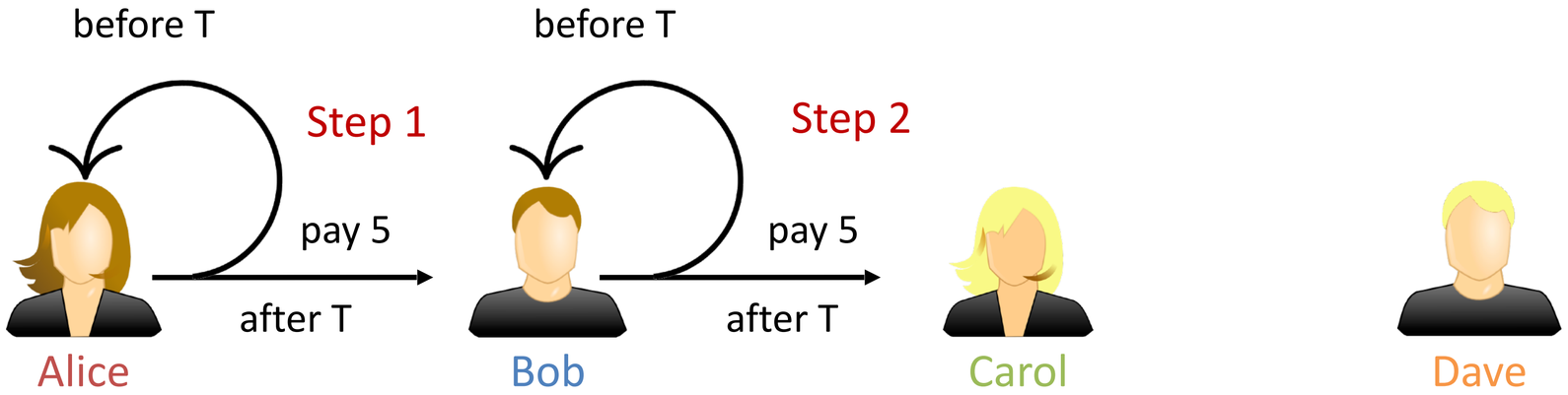
Multi-hop payments in one round: Attempt 1



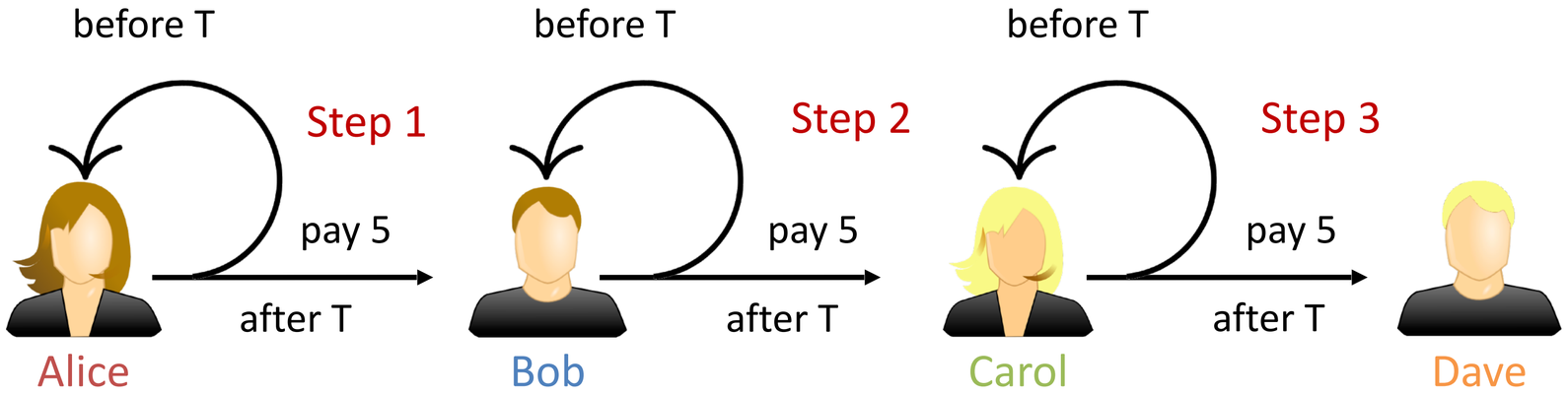
=> A malicious intermediary can stop the payment and effectively steal the 5 coins…

Towards pay-or-revoke: Attempt 2

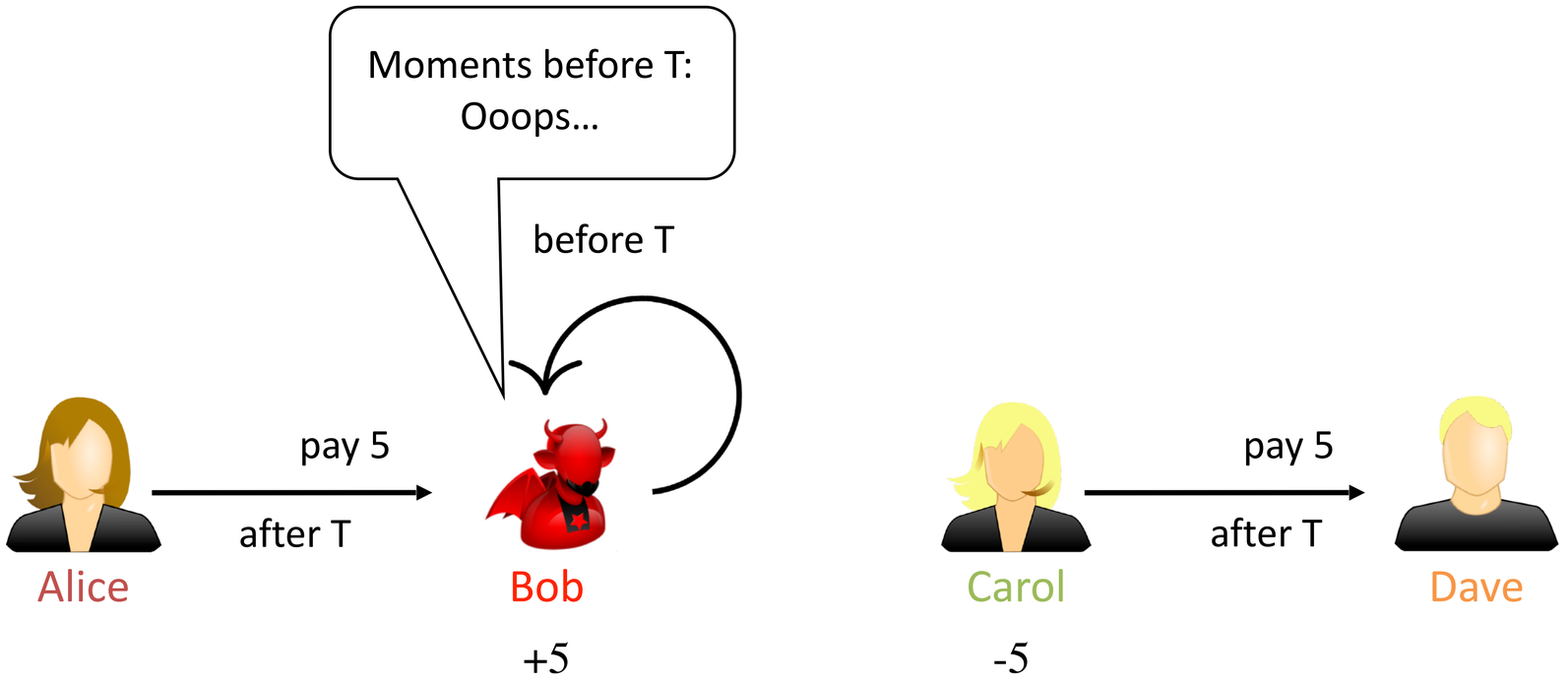
Towards pay-or-revoke: Attempt 2



Towards pay-or-revoke: Attempt 2



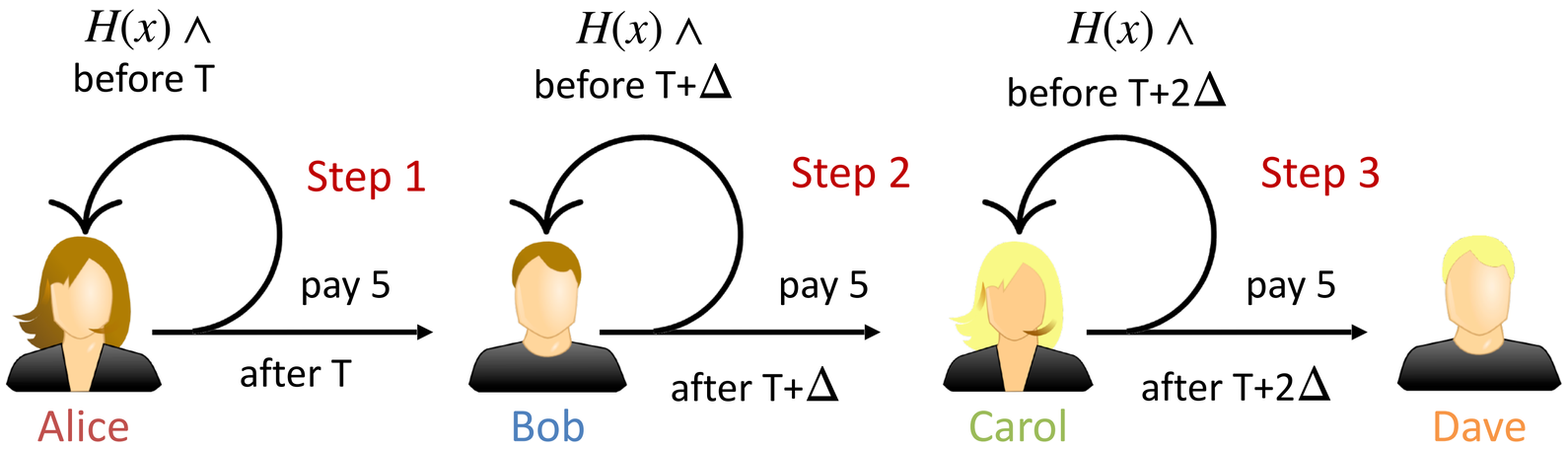
Towards pay-or-revoke: Attempt 2



- Bob refunds in the last moment
- Others won't have time to react

Towards pay-or-revoke: Attempt 3

x chosen by the sender



=> Similar to current Lightning multi-hop payments, has same scripting requirements as Lightning, collateral time grows linearly…

Pay-or-revoke paradigm

Alice

Bob

Carol

Dave

Pay-or-revoke paradigm

Alice

Alice defines a timeout T, independent of the path length

Alice

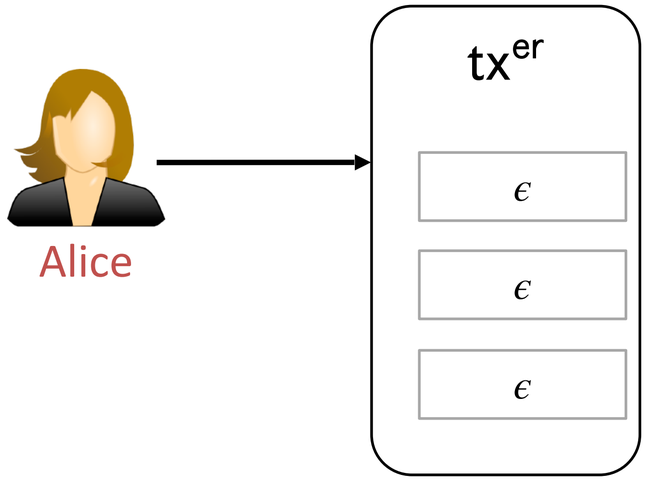
Bob

Carol

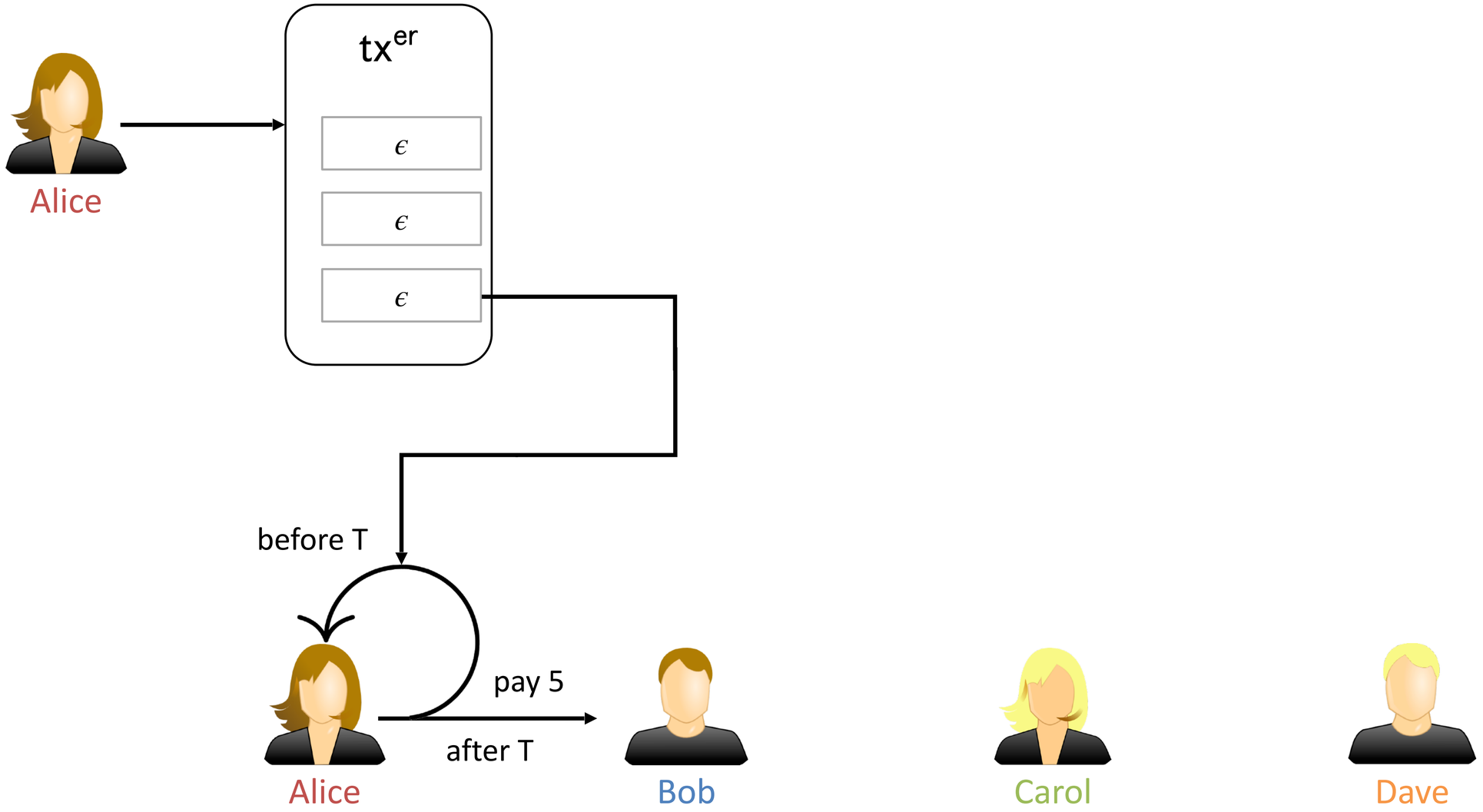
Dave

Pay-or-revoke paradigm

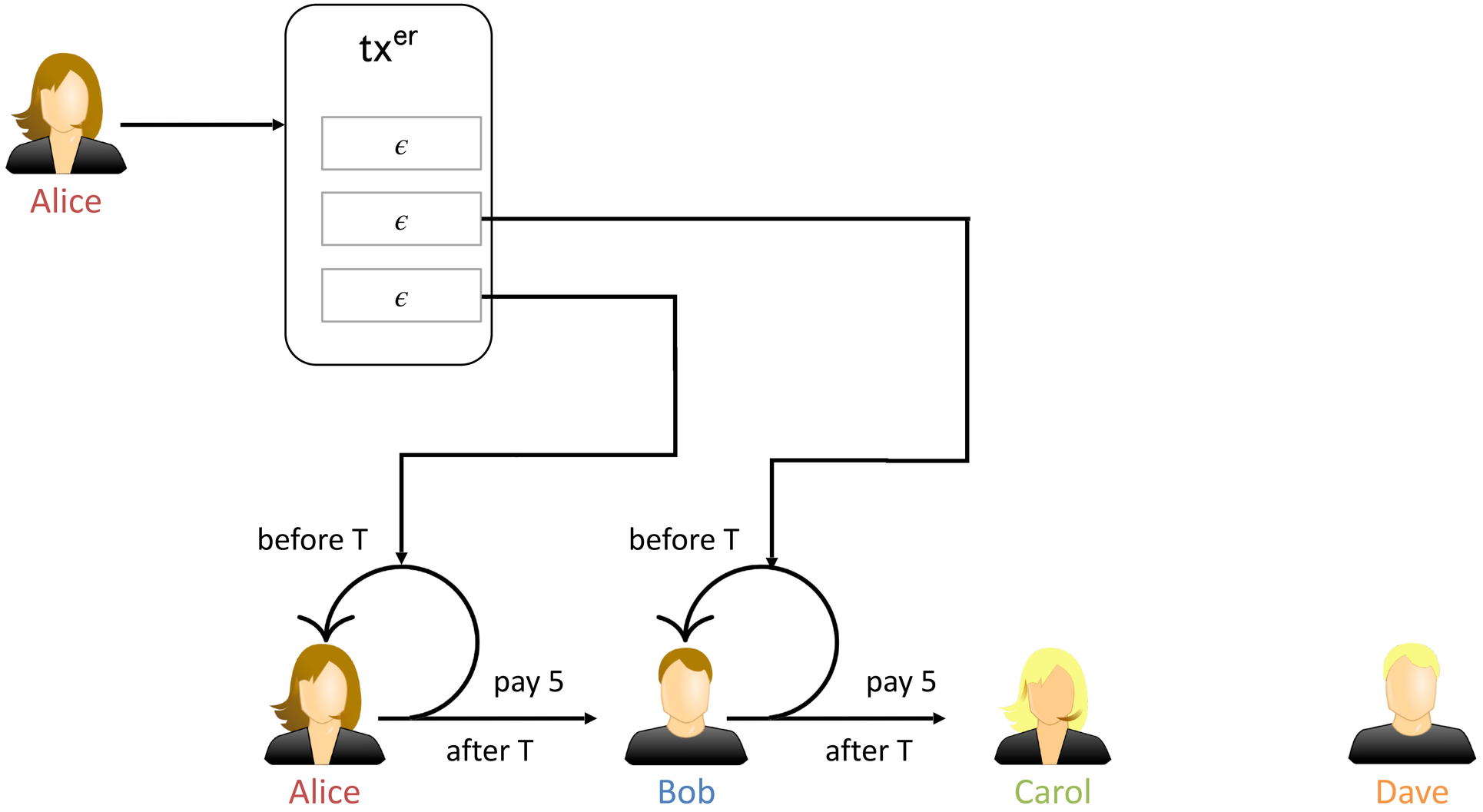
Alice creates refund enabling transaction: tx^{er}



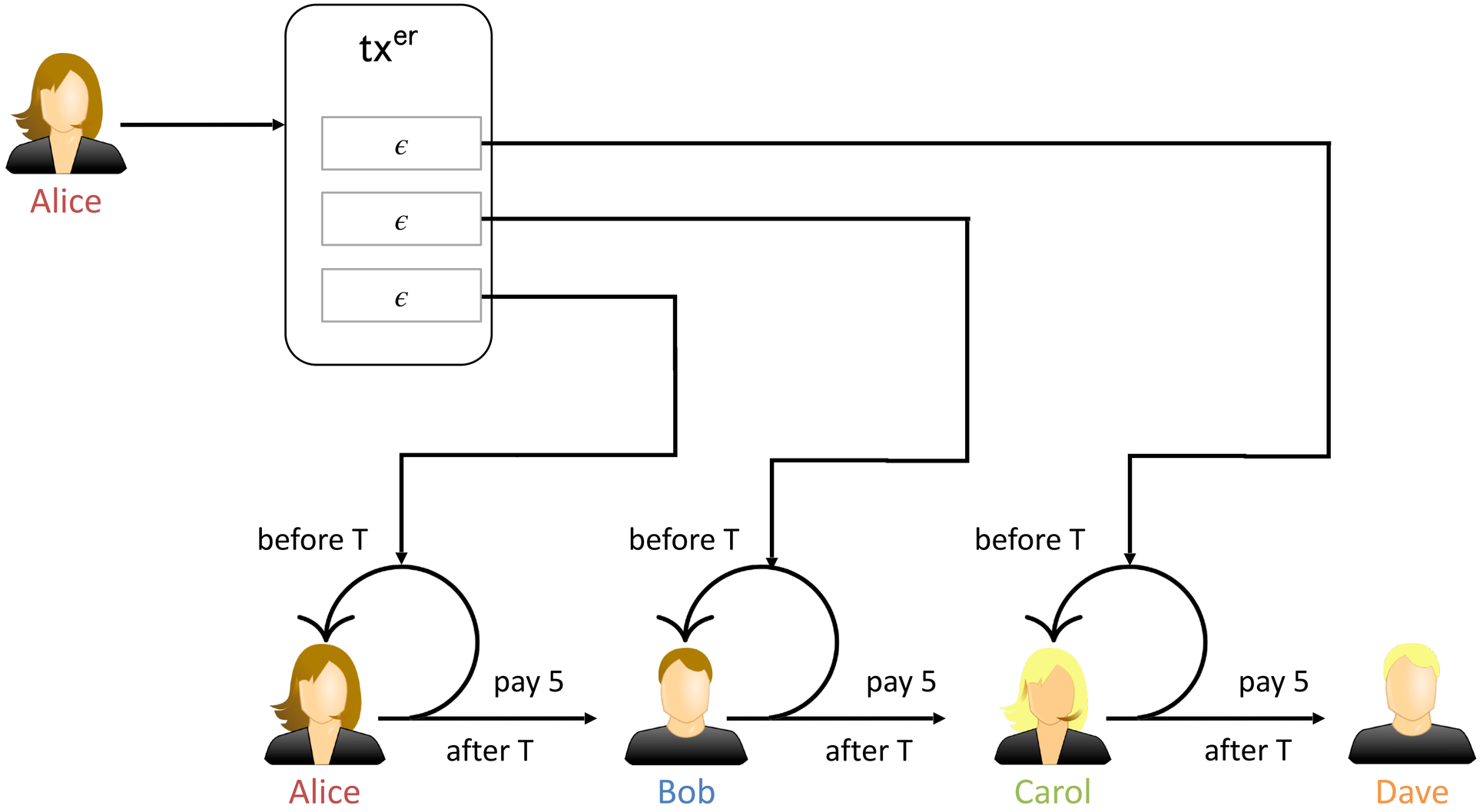
Pay-or-revoke paradigm



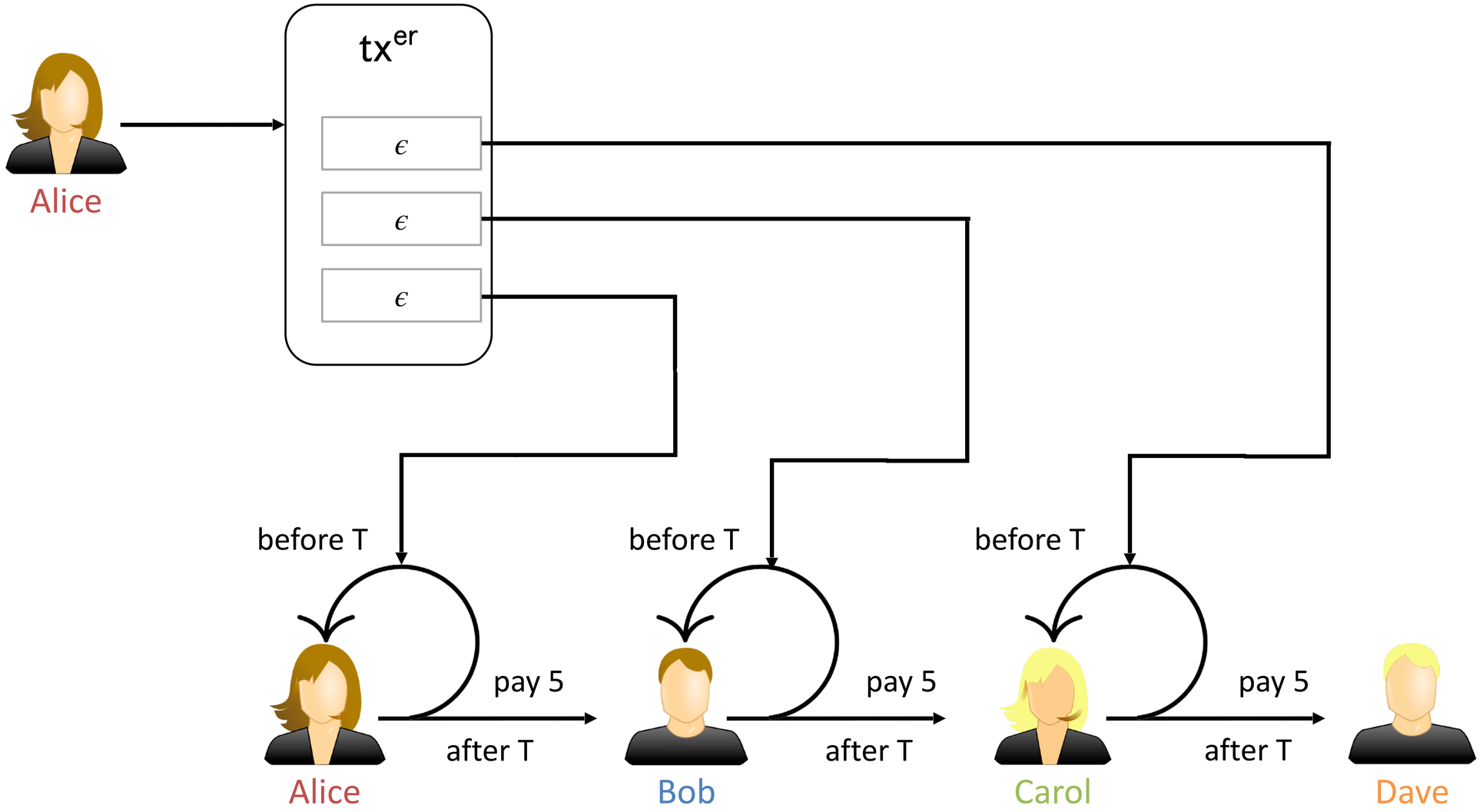
Pay-or-revoke paradigm



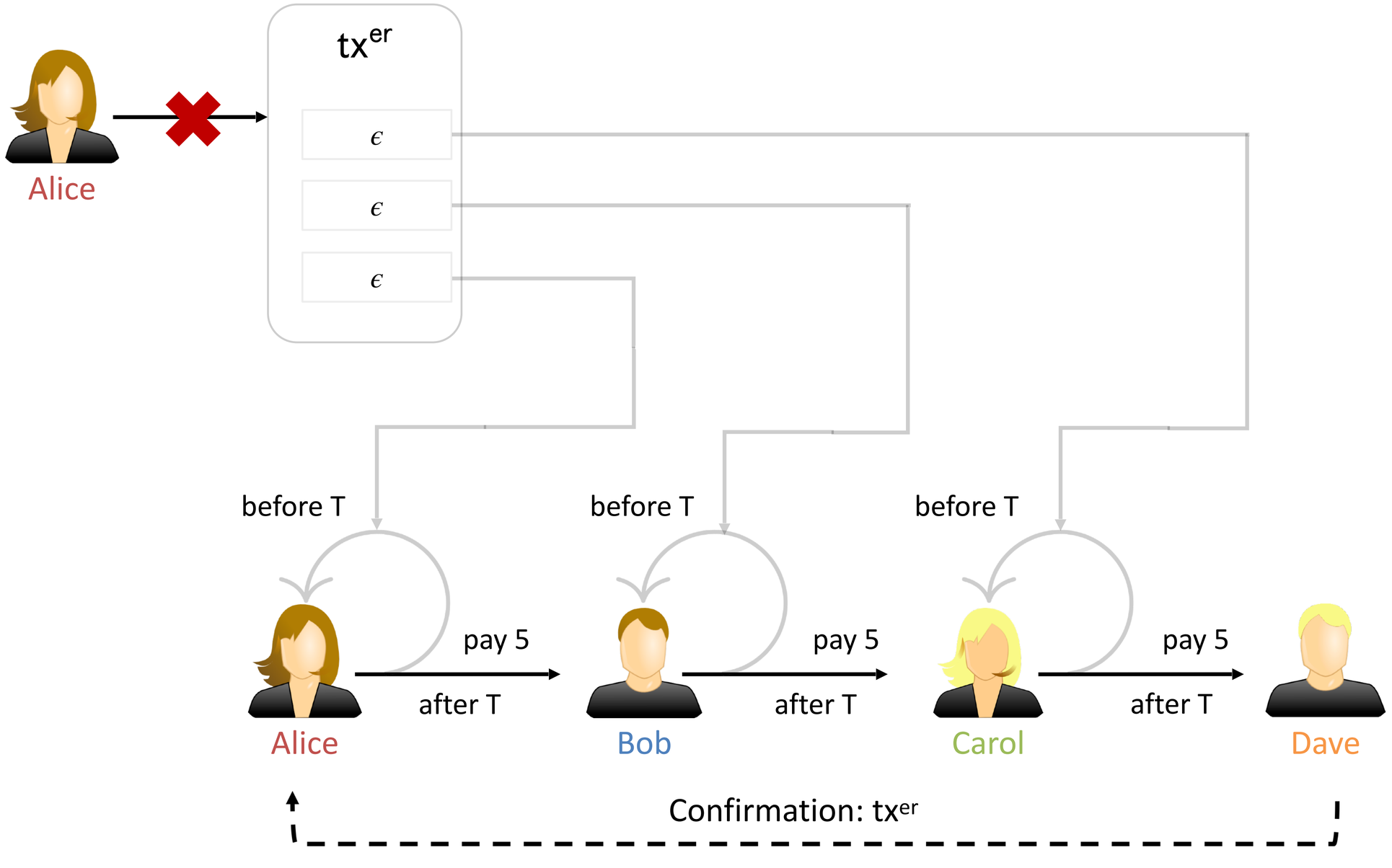
Pay-or-revoke paradigm



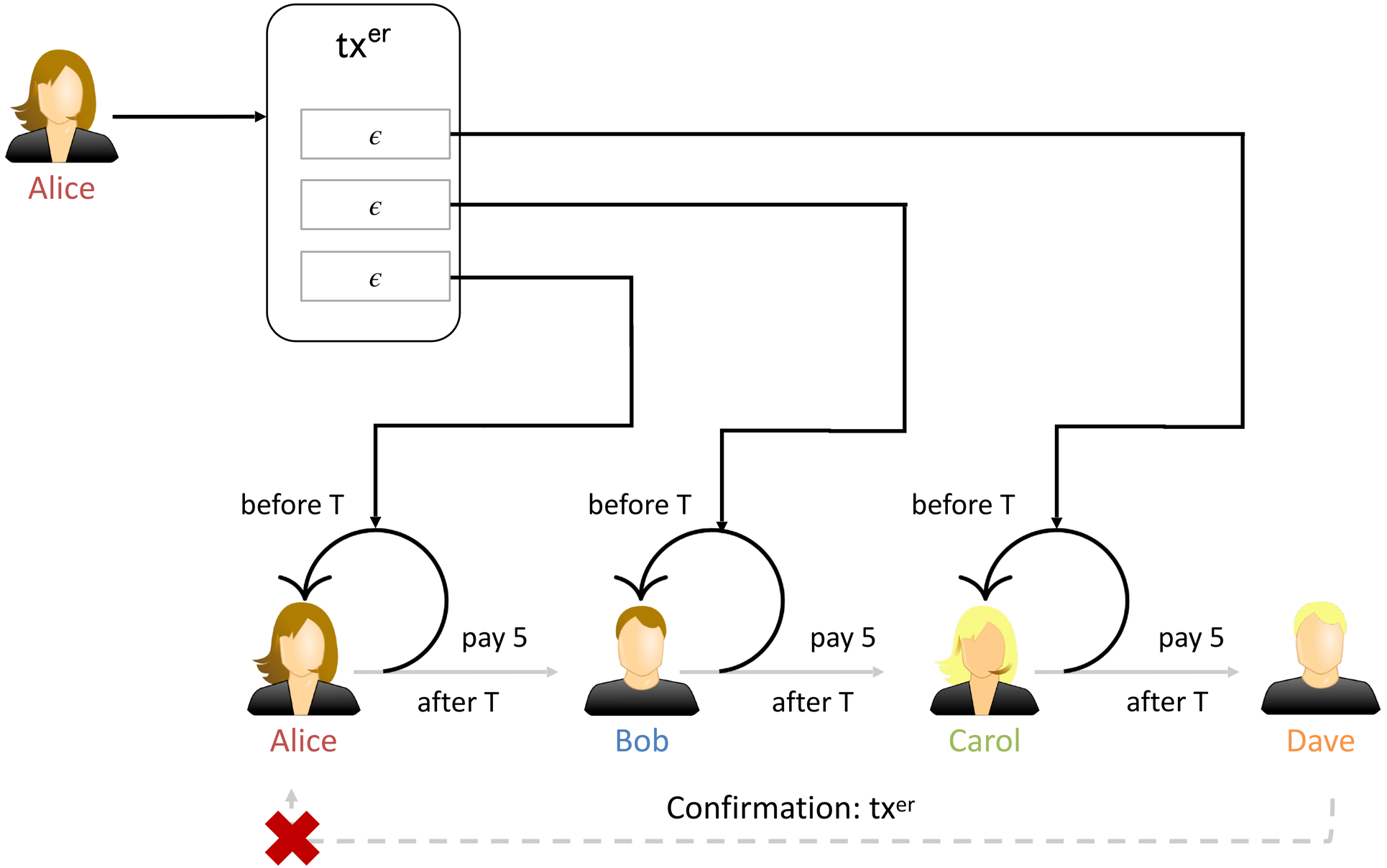
Pay-or-revoke paradigm



Successful payment



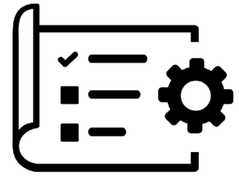
Refund


Alice
tx^er
ϵ
ϵ
ϵ
before T
before T
before T
pay 5
pay 5
pay 5
after T
after T
after T
Alice
Bob
Carol
Dave
Confirmation: tx^er

Evaluation

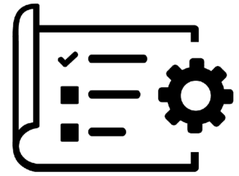
Evaluation

- Blitz contract 26% smaller than Lightning contract (HTLC)
- Can increase number of concurrent payments per channel



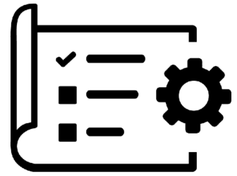
Evaluation

- Blitz contract 26% smaller than Lightning contract (HTLC)
- Can increase number of concurrent payments per channel

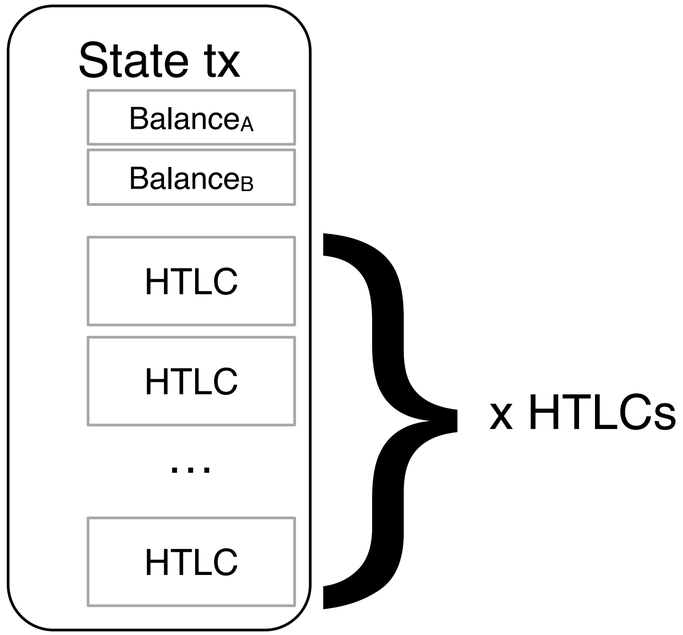


Evaluation

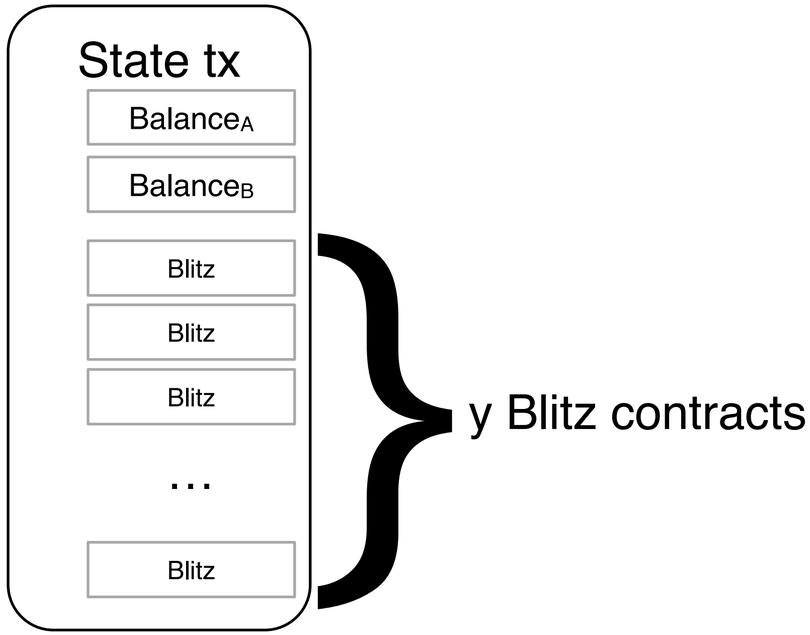
- Blitz contract 26% smaller than Lightning contract (HTLC)
- Can increase number of concurrent payments per channel



Lightning payments

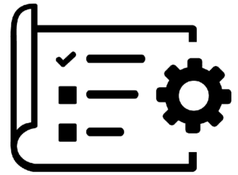


Blitz



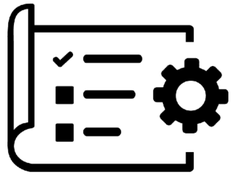
Evaluation

- Blitz contract 26% smaller than Lightning contract (HTLC)
- Can increase number of concurrent payments per channel

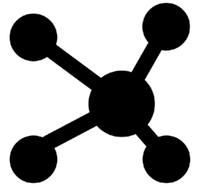


Evaluation

- Blitz contract 26% smaller than Lightning contract (HTLC)
- Can increase number of concurrent payments per channel

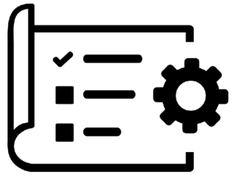


- Simulation on Lightning Network snapshot
- Random payments, some are disrupted

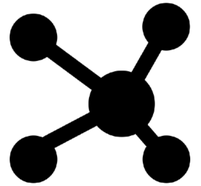


Evaluation

- Blitz contract 26% smaller than Lightning contract (HTLC)
- Can increase number of concurrent payments per channel

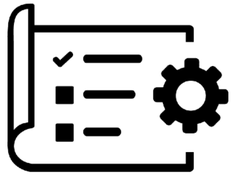


- Simulation on Lightning Network snapshot
- Random payments, some are disrupted
- Constant (Blitz) vs. staggered (Lightning) collateral

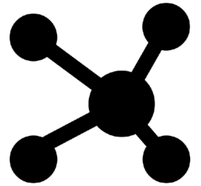


Evaluation

- Blitz contract 26% smaller than Lightning contract (HTLC)
- Can increase number of concurrent payments per channel



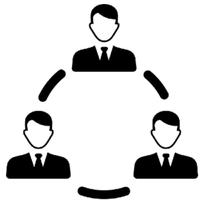
- Simulation on Lightning Network snapshot
- Random payments, some are disrupted
- Constant (Blitz) vs. staggered (Lightning) collateral
- Depending on setting, between 4x and 33x more failed payments in Lightning than Blitz



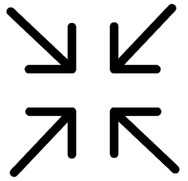
Take home: Blitz

- New multi-hop payment paradigm for Payment Channel Networks

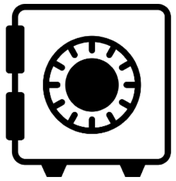
Only one round of communication



Contract size reduced by 26%



Reduced collateral from linear to constant



Security against Wormhole attack

Formalized in UC framework

Nice solution, but …

Limitations of MHPs

What we would like

Only for payments

Nice solution, but …

Limitations of MHPs

Only for payments

Each payment routed via intermediaries

What we would like

Nice solution, but …

Limitations of MHPs

Only for payments

Each payment routed via intermediaries

more fees

less privacy

less reliable

What we would like

Nice solution, but …

Limitations of MHPs

Only for payments

Each payment routed via intermediaries

- more fees
- less privacy
- less reliable

What we would like

DLCs [D17], games, betting, etc.

[D17] T. Dryja,"Discreet Log Contracts," https://adiabat.github.io/dlc.pdf

Nice solution, but …

Limitations of MHPs		What we would like
Only for payments		DLCs [D17], games, betting, etc.
Each payment routed via intermediaries		Involve intermediaries only for setup/closure

- more fees
- less privacy
- less reliable

[D17] T. Dryja,"Discreet Log Contracts," https://adiabat.github.io/dlc.pdf

Nice solution, but …

Limitations of MHPs		What we would like
Only for payments		DLCs [D17], games, betting, etc.
Each payment routed via intermediaries		Involve intermediaries only for setup/closure
more fees		fewer fees
less privacy		more privacy
less reliable		more reliable

[D17] T. Dryja,"Discreet Log Contracts," https://adiabat.github.io/dlc.pdf

Other applications?

Other applications?

- Conditional payments, bets
  - Stock price

Other applications?

- Conditional payments, bets
  - Stock price
  - Weather
  - Sports game
  - etc.
- e.g., Discreet Log Contracts (DLCs) [D17]

Other applications?

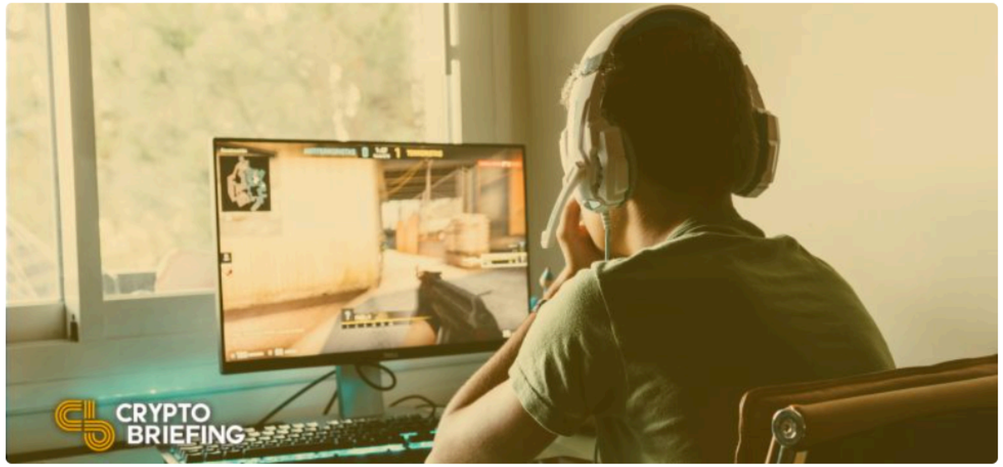
- Conditional payments, bets
  - Stock price
  - Weather
  - Sports game
  - etc.
- e.g., Discreet Log Contracts (DLCs) [D17]

Business

Lightning Network Integration Now Lets Counter-Strike Players Earn Bitcoin

by Nivesh Rustgi Dec. 28, 2020

Counter-Strike players will be able to bet Bitcoin and earn sats for each kill.



Bitcoin-focused gaming developer ZEBEDEE has designed a prototype to play Counter-Strike and earn BTC through the lightning network.

Counter-Strike to Add Lightning Network

"There is a lot of low hanging fruit to simply add Bitcoin to existing games," said the co-founder of ZEBEDEE on a Twitch video, demonstrating the latest Infuse app.

The application integrated seamlessly via Steam, the largest online

Trending News

Dog Coin Shiba Inu Looks to Resume Its Uptrend
Markets · 3 days ago

Bitcoin Looks Set to Dip After Traders Lose $700M in Liquidations
Markets · 3 days ago

Cardano Could Retrace Before Targeting $2.70
Markets · Nov. 9, 2021

Kart Racing League Announces Public Sale of Governance Token

https://cryptobriefing.com/lightning-network-counter-strike-players-earn-bitcoin/

Other applications?

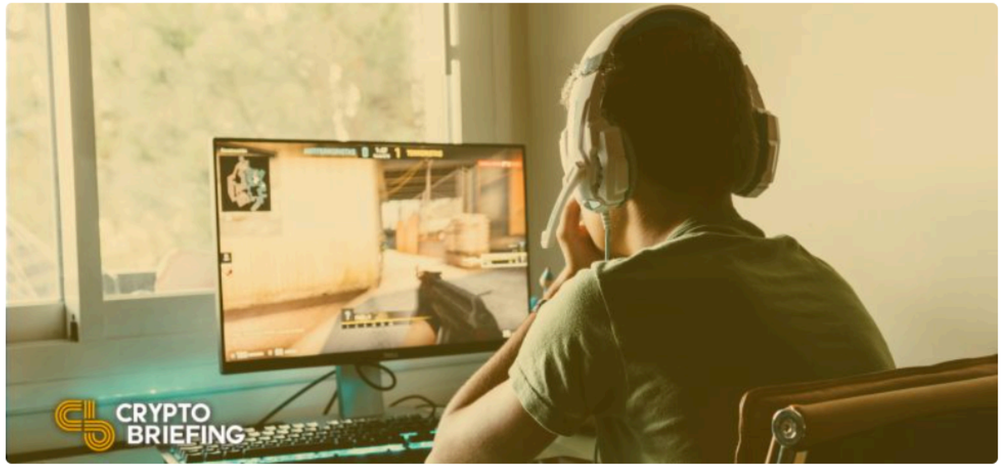
- Conditional payments, bets
  - Stock price
  - Weather
  - Sports game
  - etc.
- e.g., Discreet Log Contracts (DLCs) [D17]
- Works in individual channels, but not between any two users in the network

Business

Lightning Network Integration Now Lets Counter-Strike Players Earn Bitcoin

by Nivesh Rustgi Dec. 28, 2020

Counter-Strike players will be able to bet Bitcoin and earn sats for each kill.



Bitcoin-focused gaming developer ZEBEDEE has designed a prototype to play Counter-Strike and earn BTC through the lightning network.

Counter-Strike to Add Lightning Network

"There is a lot of low hanging fruit to simply add Bitcoin to existing games," said the co-founder of ZEBEDEE on a Twitch video, demonstrating the latest Infuse app.

The application integrated seamlessly via Steam, the largest online

Trending News

Dog Coin Shiba Inu Looks to Resume Its Uptrend
Markets · 3 days ago

Bitcoin Looks Set to Dip After Traders Lose $700M in Liquidations
Markets · 3 days ago

Cardano Could Retrace Before Targeting $2.70
Markets · Nov. 9, 2021

Kart Racing League Announces Public Sale of Governance Token

https://cryptobriefing.com/lightning-network-counter-strike-players-earn-bitcoin/

[D17] T. Dryja,"Discreet Log Contracts," https://adiabat.github.io/dlc.pdf

Other applications?

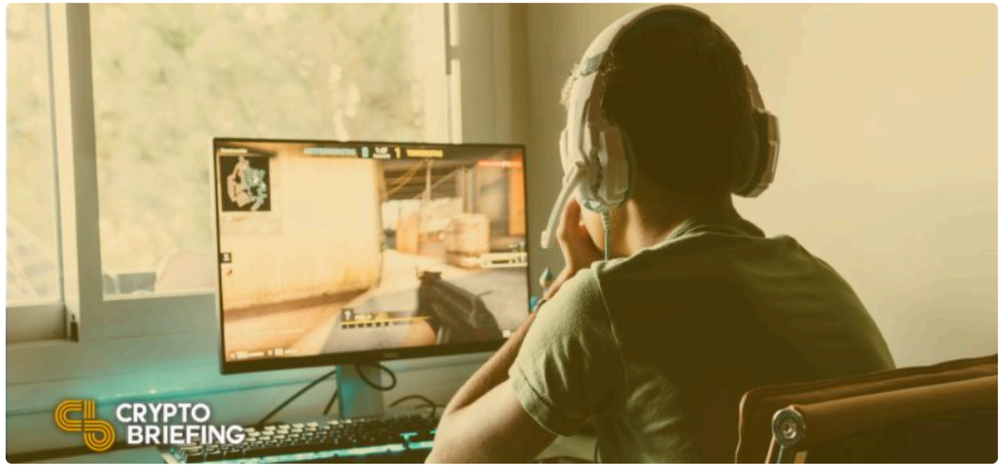
- Conditional payments, bets
  - Stock price
  - Weather
  - Sports game
  - etc.
- e.g., Discreet Log Contracts (DLCs) [D17]
- Works in individual channels, but not between any two users in the network
- MHP only for payments!

Business

Lightning Network Integration Now Lets Counter-Strike Players Earn Bitcoin

by Nivesh Rustgi Dec. 28, 2020

Counter-Strike players will be able to bet Bitcoin and earn sats for each kill.



Bitcoin-focused gaming developer ZEBEDEE has designed a prototype to play Counter-Strike and earn BTC through the lightning network.

Counter-Strike to Add Lightning Network

"There is a lot of low hanging fruit to simply add Bitcoin to existing games," said the co-founder of ZEBEDEE on a Twitch video, demonstrating the latest Infuse app.

The application integrated seamlessly via Steam, the largest online

Trending News

Dog Coin Shiba Inu Looks to Resume Its Uptrend
Markets · 3 days ago

Bitcoin Looks Set to Dip After Traders Lose $700M in Liquidations
Markets · 3 days ago

Cardano Could Retrace Before Targeting $2.70
Markets · Nov. 9, 2021

Kart Racing League Announces Public Sale of Governance Token

https://cryptobriefing.com/lightning-network-counter-strike-players-earn-bitcoin/

[D17] T. Dryja,"Discreet Log Contracts," https://adiabat.github.io/dlc.pdf

Other applications?

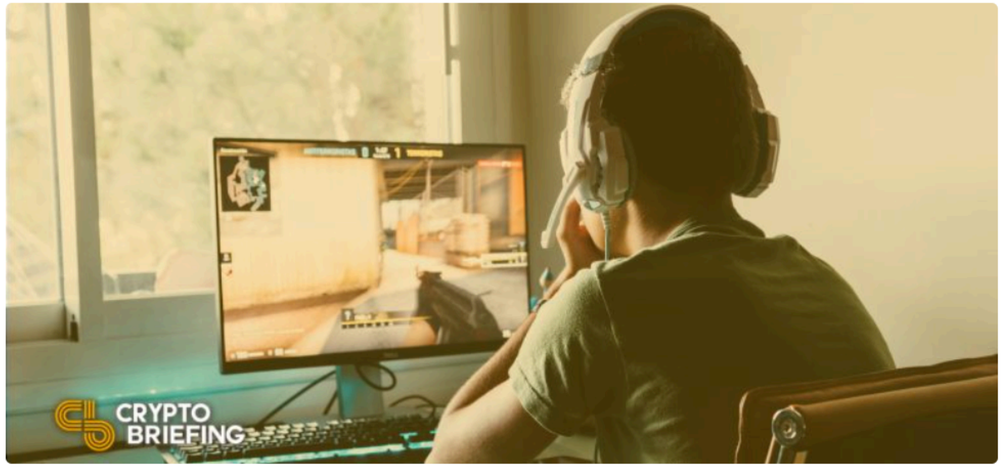
- Conditional payments, bets
  - Stock price
  - Weather
  - Sports game
  - etc.
- e.g., Discreet Log Contracts (DLCs) [D17]
- Works in individual channels, but not between any two users in the network
- MHP only for payments!

Business

Lightning Network Integration Now Lets Counter-Strike Players Earn Bitcoin

by Nivesh Rustgi Dec. 28, 2020

Counter-Strike players will be able to bet Bitcoin and earn sats for each kill.



Bitcoin-focused gaming developer ZEBEDEE has designed a prototype to play Counter-Strike and earn BTC through the lightning network.

Counter-Strike to Add Lightning Network

"There is a lot of low hanging fruit to simply add Bitcoin to existing games," said the co-founder of ZEBEDEE on a Twitch video, demonstrating the latest Infuse app.

The application integrated seamlessly via Steam, the largest online

Trending News

Dog Coin Shiba Inu Looks to Resume Its Uptrend
Markets · 3 days ago

Bitcoin Looks Set to Dip After Traders Lose $700M in Liquidations
Markets · 3 days ago

Cardano Could Retrace Before Targeting $2.70
Markets · Nov. 9, 2021

Kart Racing League Announces Public Sale of Governance Token

https://cryptobriefing.com/lightning-network-counter-strike-players-earn-bitcoin/

[D17] T. Dryja,"Discreet Log Contracts," https://adiabat.github.io/dlc.pdf

Virtual Channels

Breaking and Fixing Virtual Channels:
Domino Attack and Donner

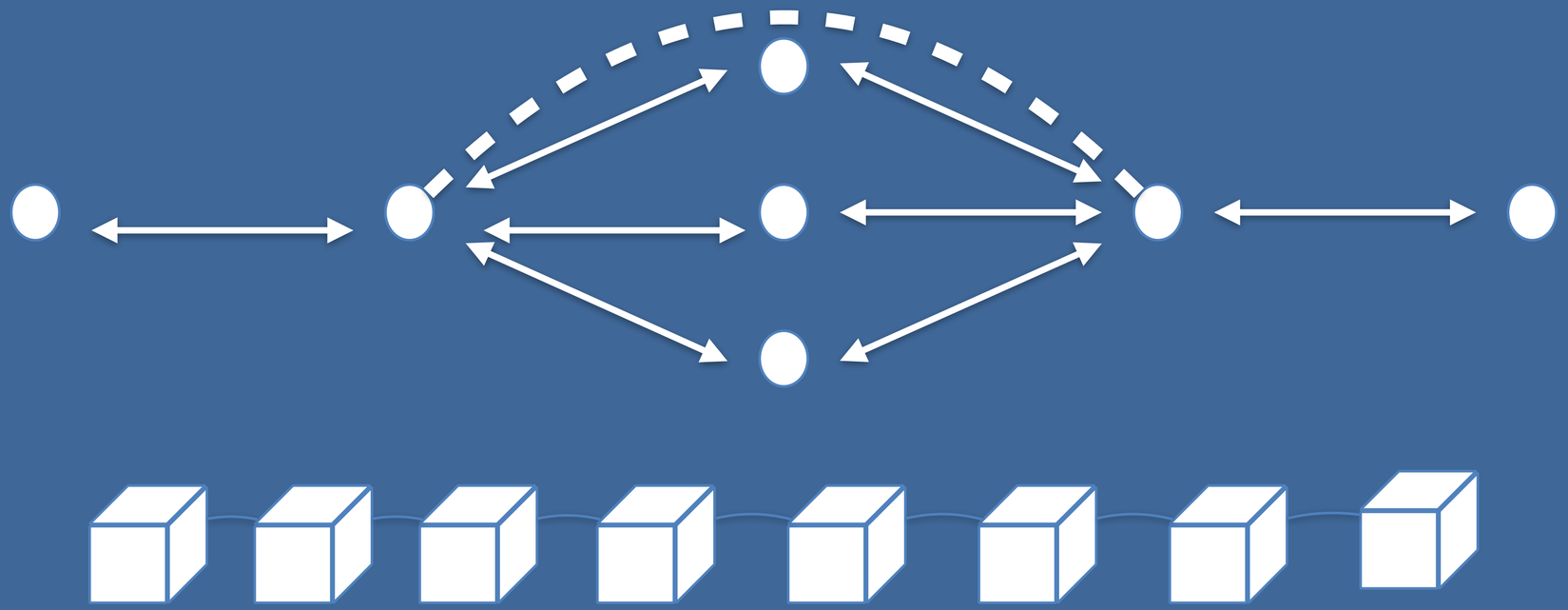
Lukas Aumayr
TU Wien
lukas.aumayr@tuwien.ac.at

Pedro Moreno-Sanchez
IMDEA Software Institute
pedro.moreno@imdea.org

Aniket Kate
Purdue University / Supra
aniket@purdue.edu

Matteo Maffei
Christian Doppler Laboratory
Blockchain Technologies for the
Internet of Things / TU Wien
matteo.maffei@tuwien.ac.at

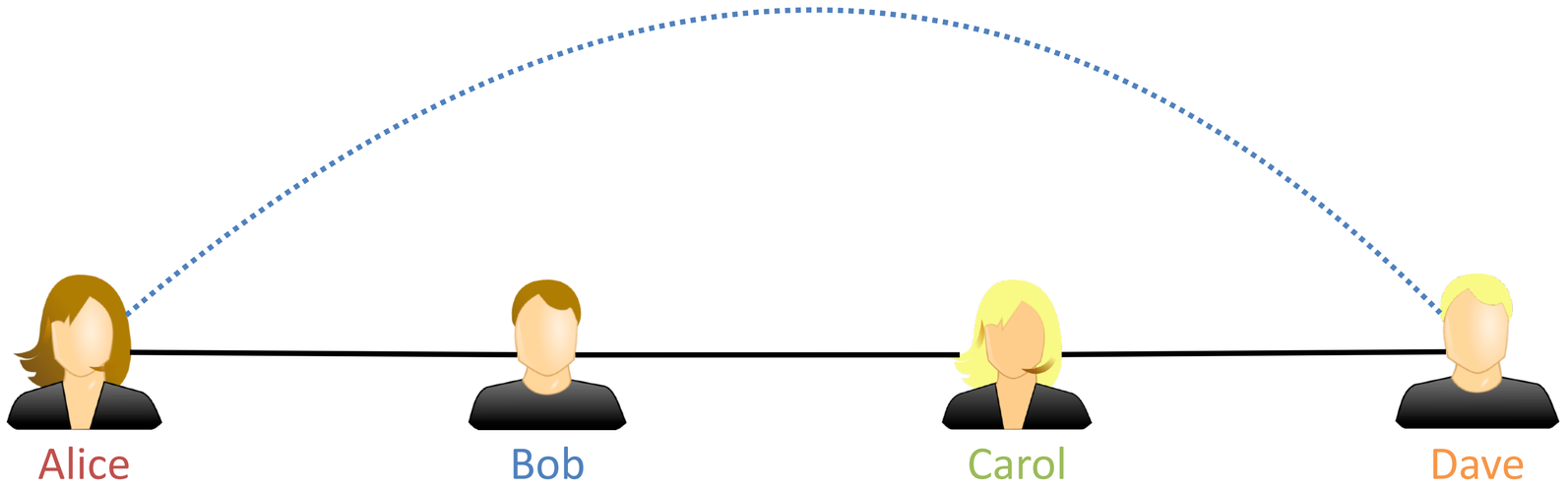
NDSS 2023



Establish bridges over channels off-chain

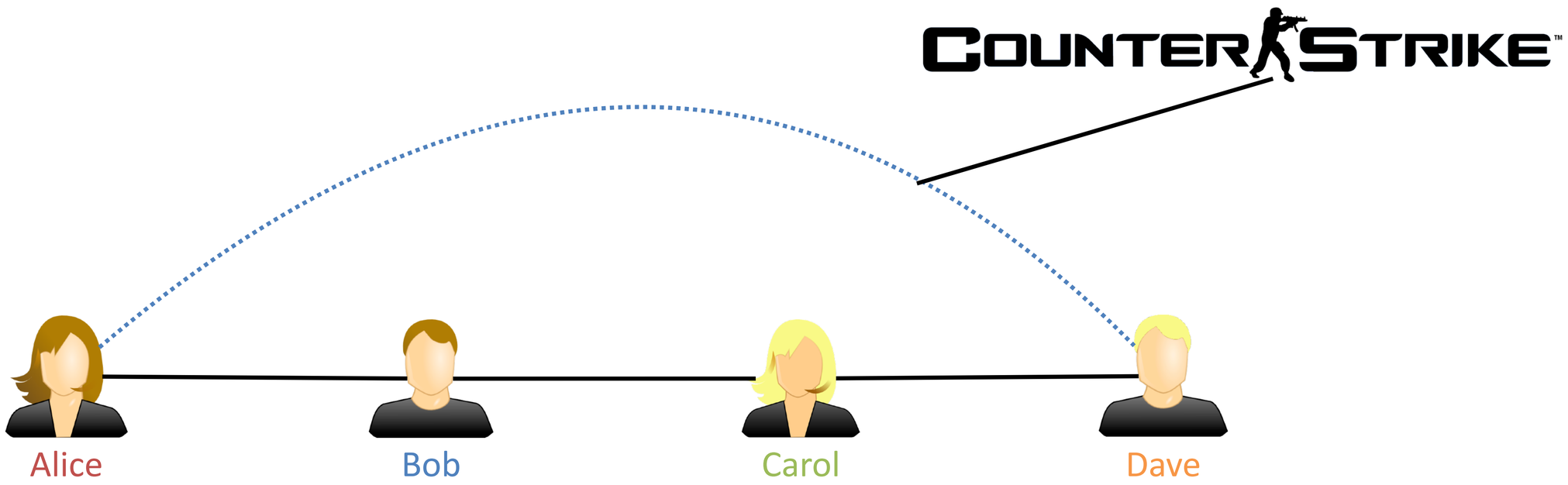
Virtual channel (VC)

Key idea:
- Open a virtual channel, without modifying the PCN
- VC is same as PC, but funding transaction (FT) off-chain



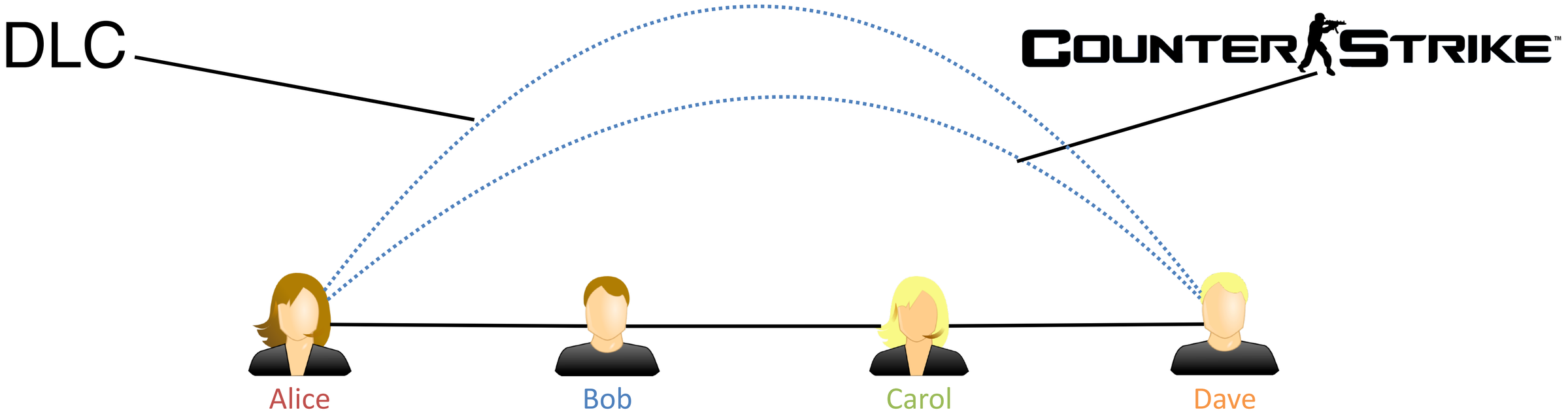
Virtual channel (VC)

Key idea:
- Open a virtual channel, without modifying the PCN
- VC is same as PC, but funding transaction (FT) off-chain



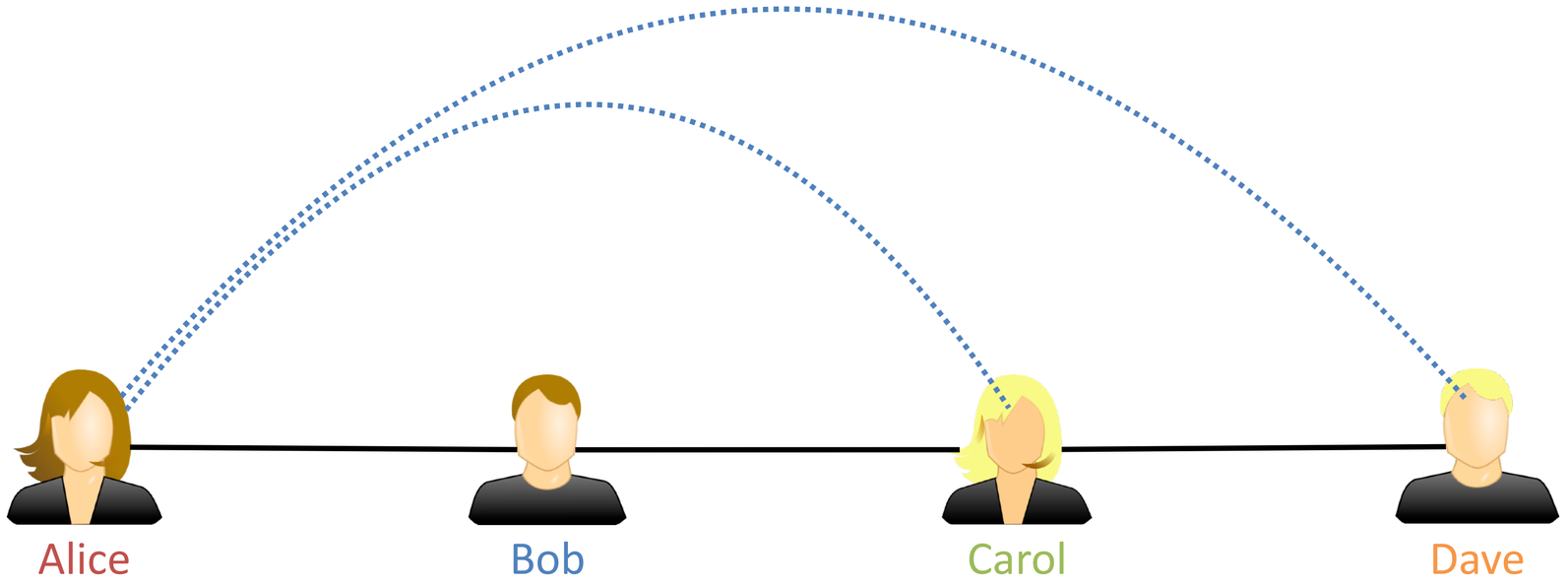
Virtual channel (VC)

Key idea:
- Open a virtual channel, without modifying the PCN
- VC is same as PC, but funding transaction (FT) off-chain

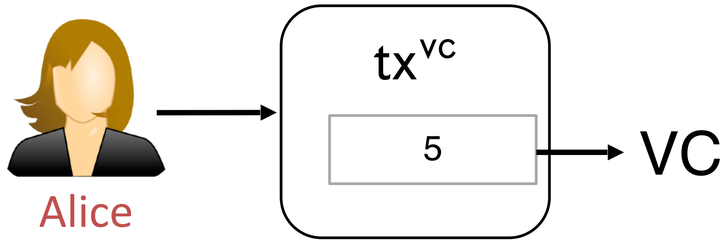


Virtual Channel (VC)

- Existing constructions based on recursive paradigm
- We present a new attack (Domino attack) on all of them, which would shut down the Lighting Network
- We need a new design paradigm!



Virtual channel



Funding transaction
of the virtual channel

Idea:

- Alice funds the channel with amount 5 off-chain

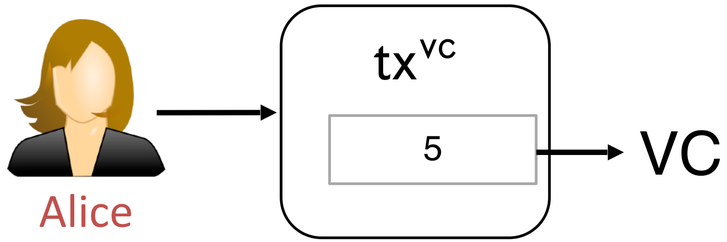

Alice


Bob


Carol


Dave

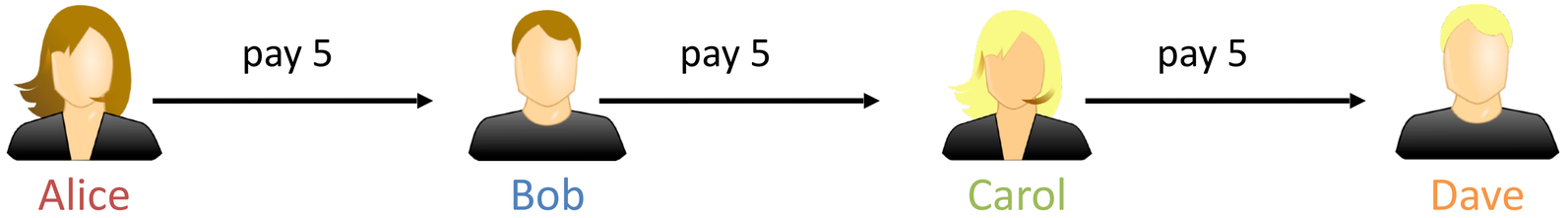
Virtual channel



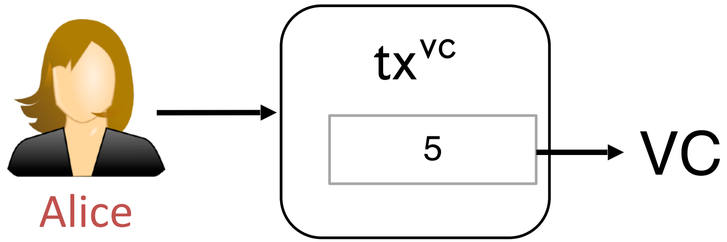
Funding transaction
of the virtual channel

Idea:

- Alice funds the channel with amount 5 off-chain
- Set up a collateral payment of 5 coins



Virtual channel

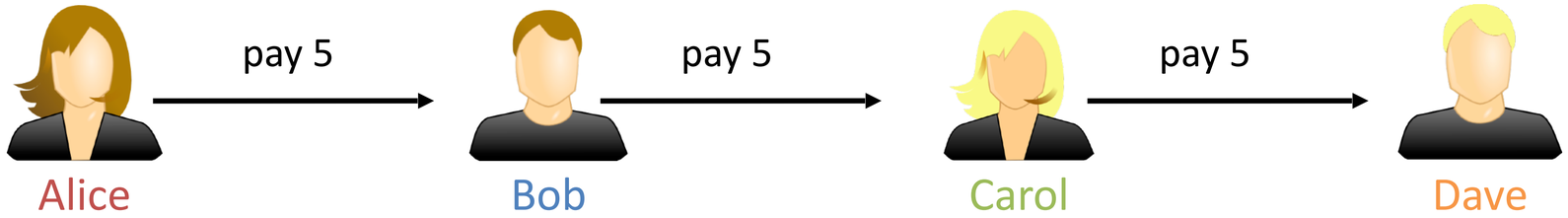


Funding transaction
of the virtual channel

Idea:

- Alice funds the channel with amount 5 off-chain
- Set up a collateral payment of 5 coins
- Connect funding and payment 5, s.t.,
  - If funding is published, Alice gets collateral back
  - Otherwise, Dave gets 5 coins through payment

?????

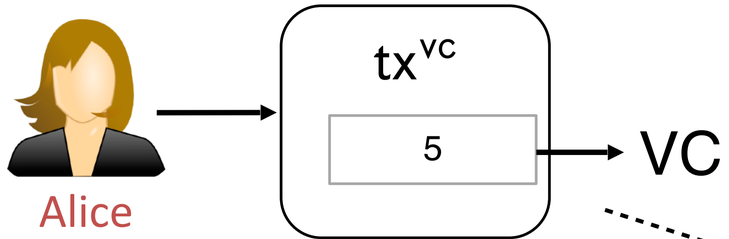


Virtual channel

Rationale

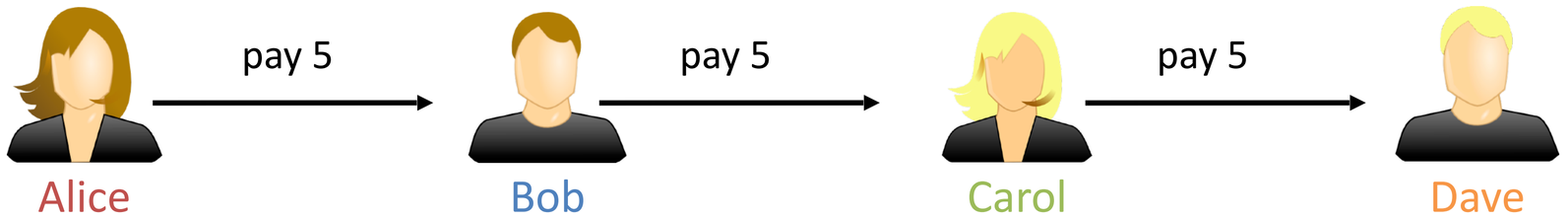
- Posting FT, means that the VC is now funded on-chain -> payment channel (PC)
- Dave is safe
  - Either gets money from payment
  - Or can claim from transformed PC

Virtual channel

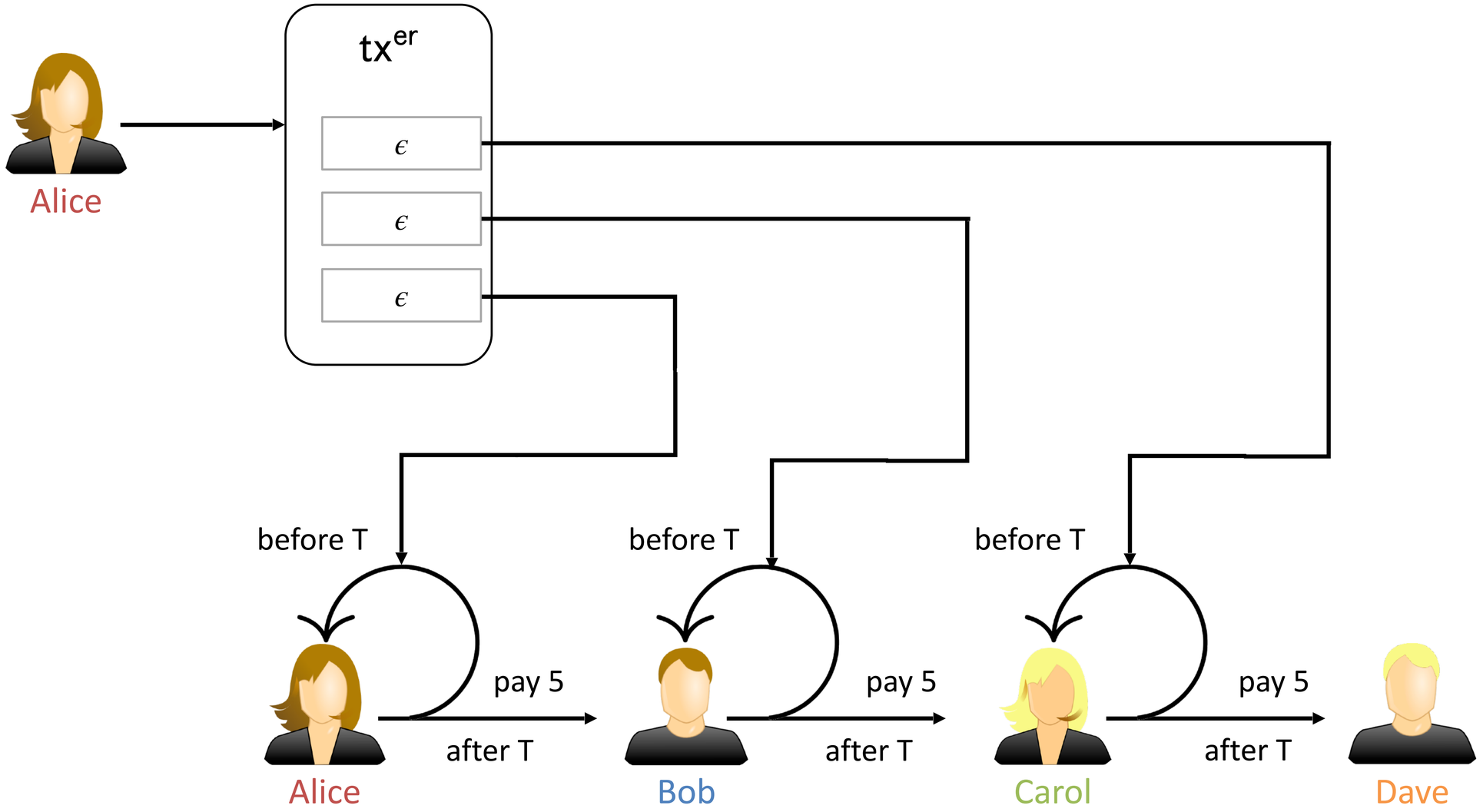


Funding transaction
of the virtual channel

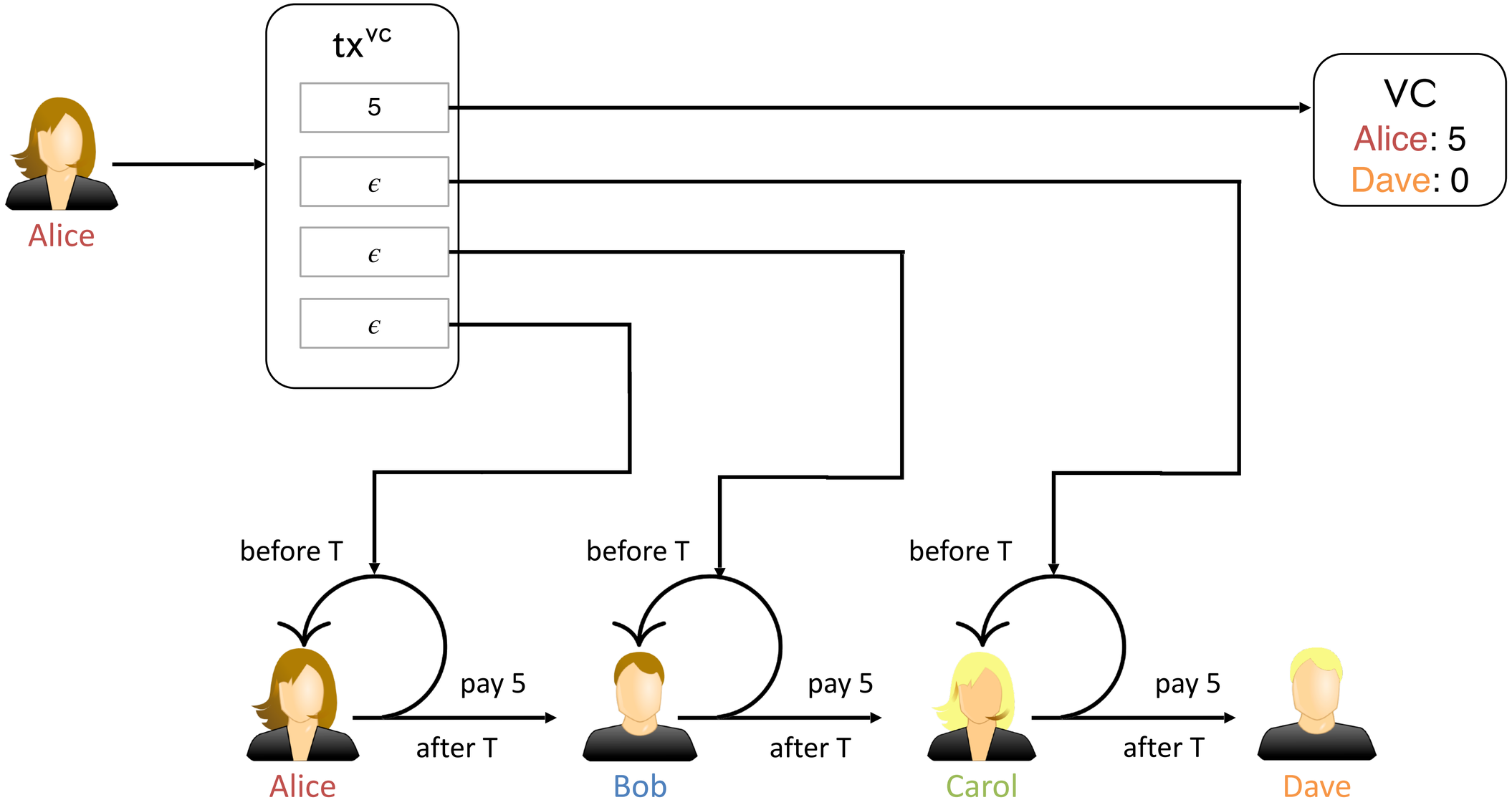
- Challenge: FT and payment must be mutually exclusive!



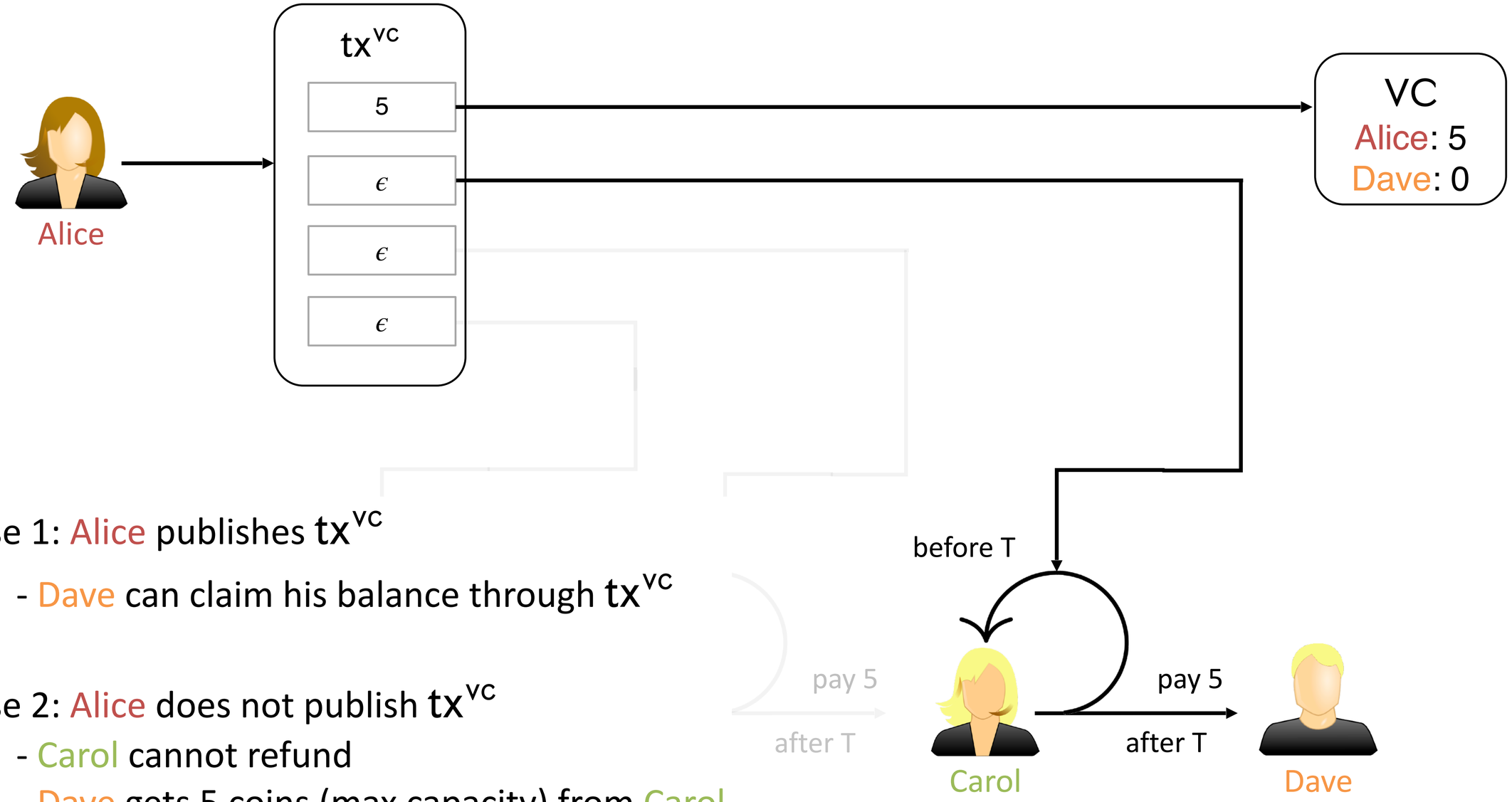
Recall our Blitz payment scheme!



We can fund the VC



Dave?



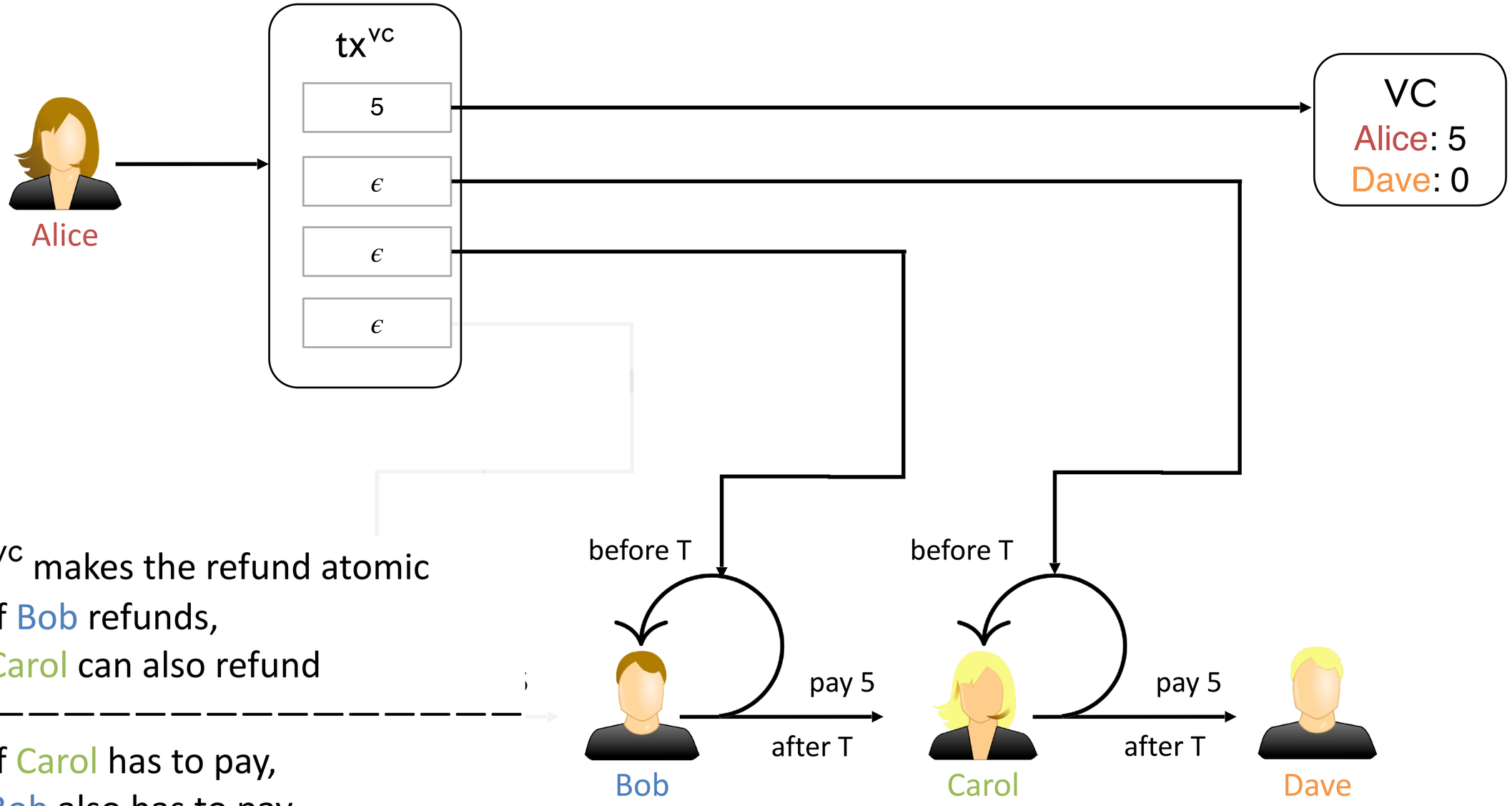
Case 1: Alice publishes tx^{vc}

- Dave can claim his balance through tx^{vc}

Case 2: Alice does not publish tx^{vc}

- Carol cannot refund
- Dave gets 5 coins (max capacity) from Carol

Carol (or other intermediaries)?

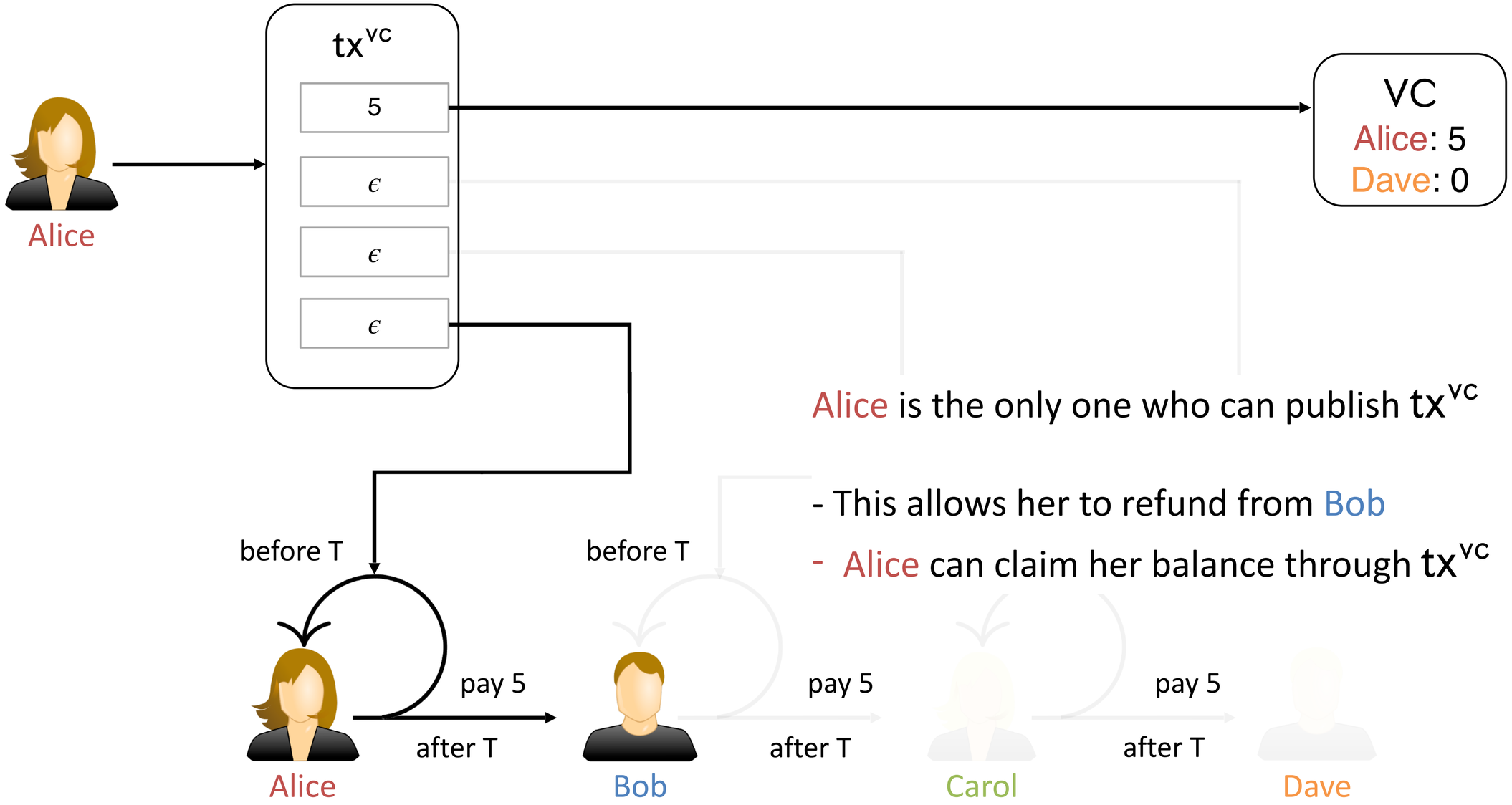


tx^{vc} makes the refund atomic

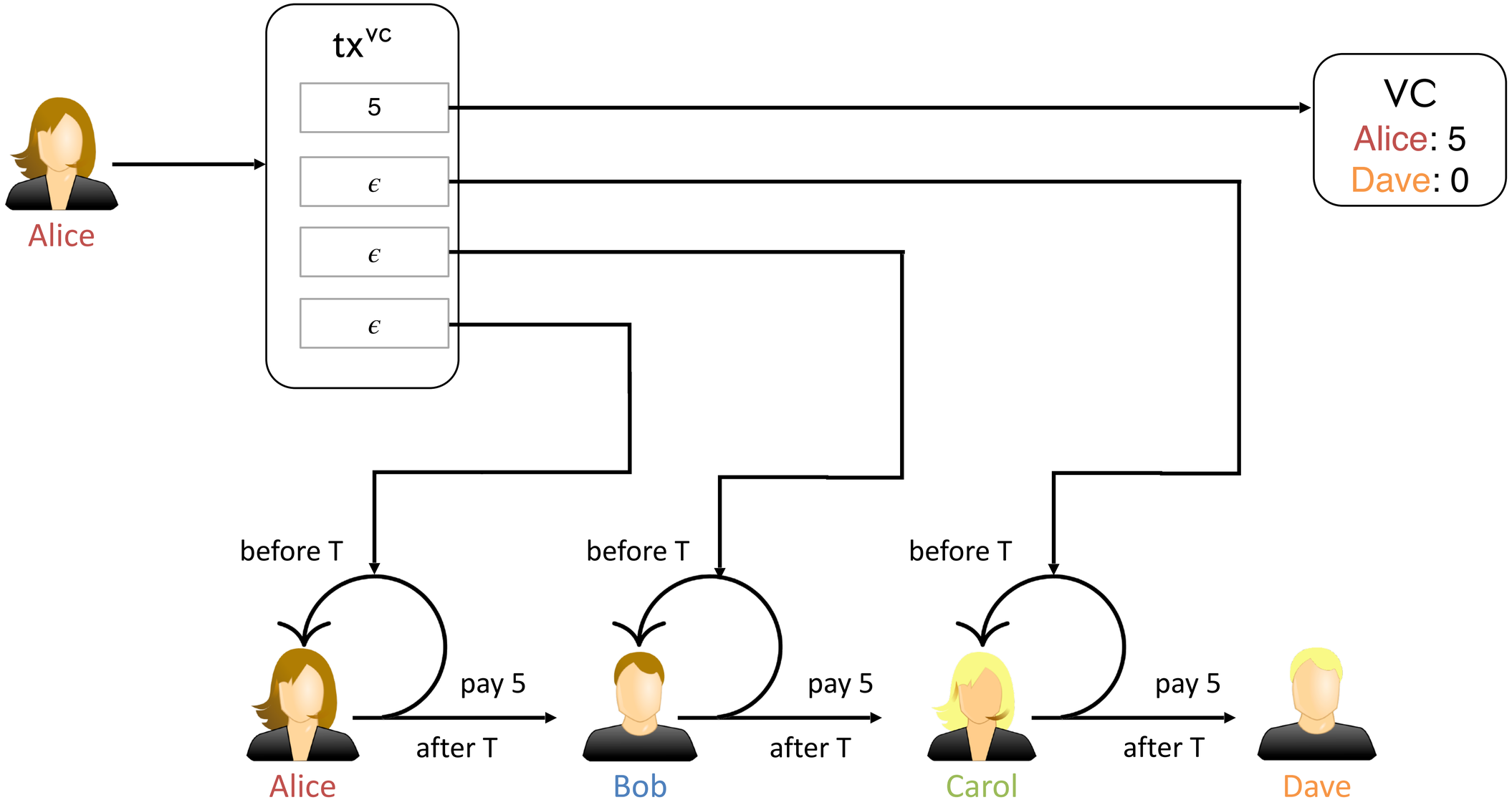
- if Bob refunds,
  Carol can also refund

- if Carol has to pay,
  Bob also has to pay

Alice?

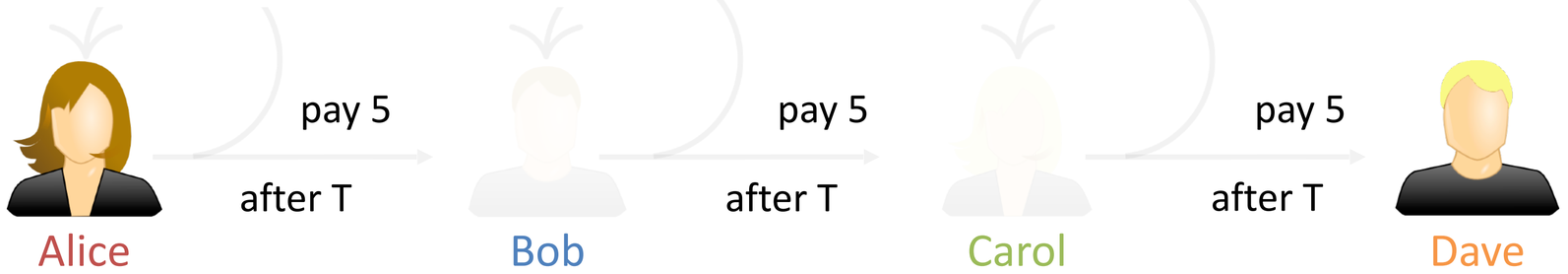

tx^{vc}
5
ϵ
ϵ
ϵ
Alice
VC
Alice: 5
Dave: 0
Alice is the only one who can publish tx^{vc}
- This allows her to refund from Bob
- Alice can claim her balance through tx^{vc}
before T
before T
pay 5
after T
pay 5
after T
pay 5
after T
Alice
Bob
Carol
Dave

How to use the VC



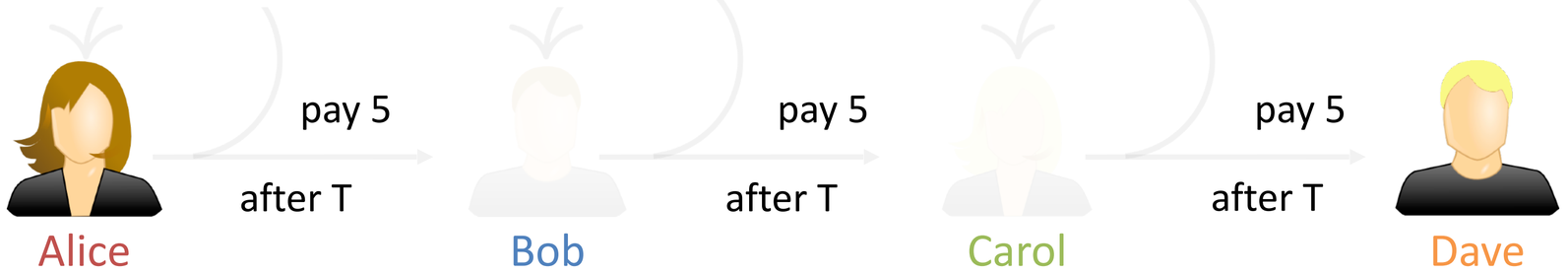
How to use the VC

Alice and Dave update the VC by exchanging new commitment txs and revoking the previous ones.

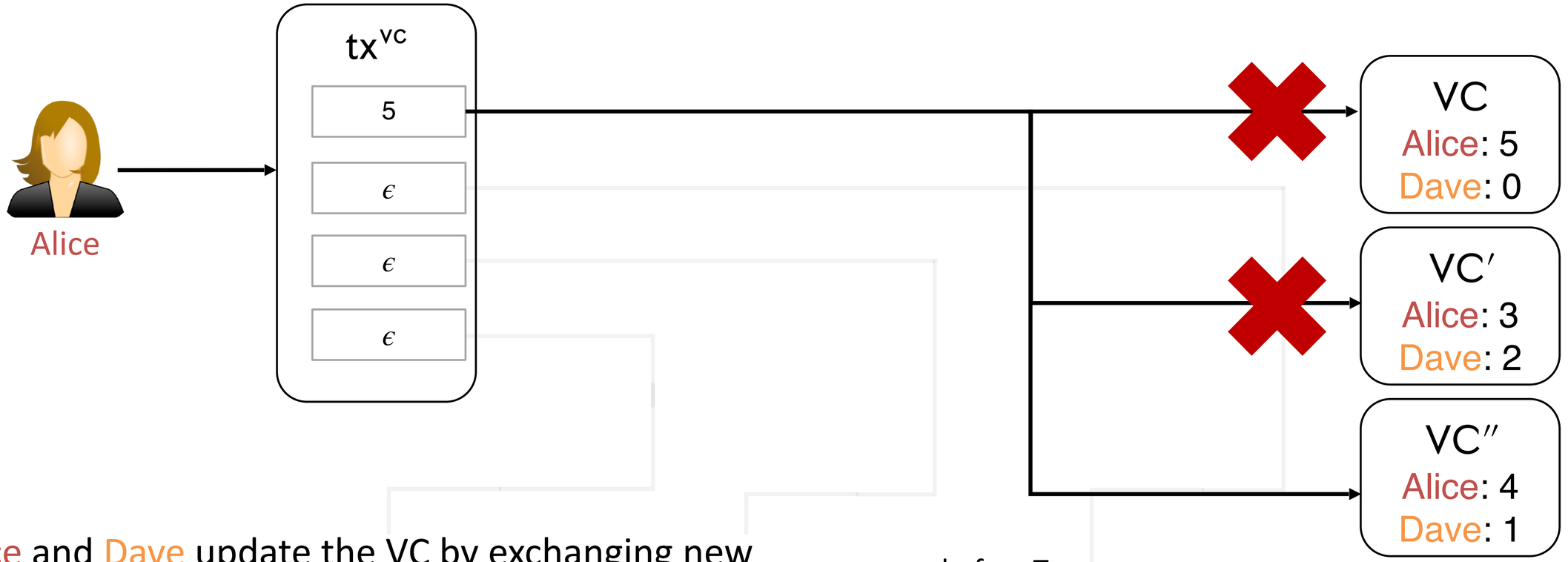


How to use the VC

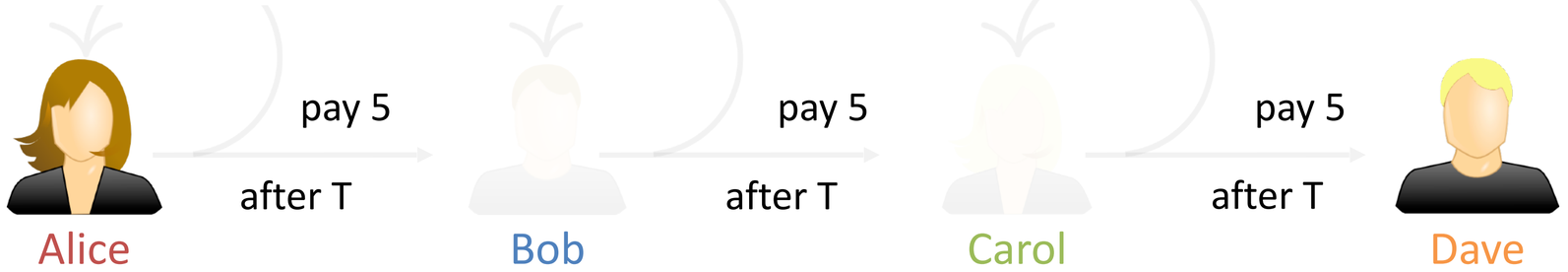
Alice and Dave update the VC by exchanging new commitment txs and revoking the previous ones.



How to use the VC

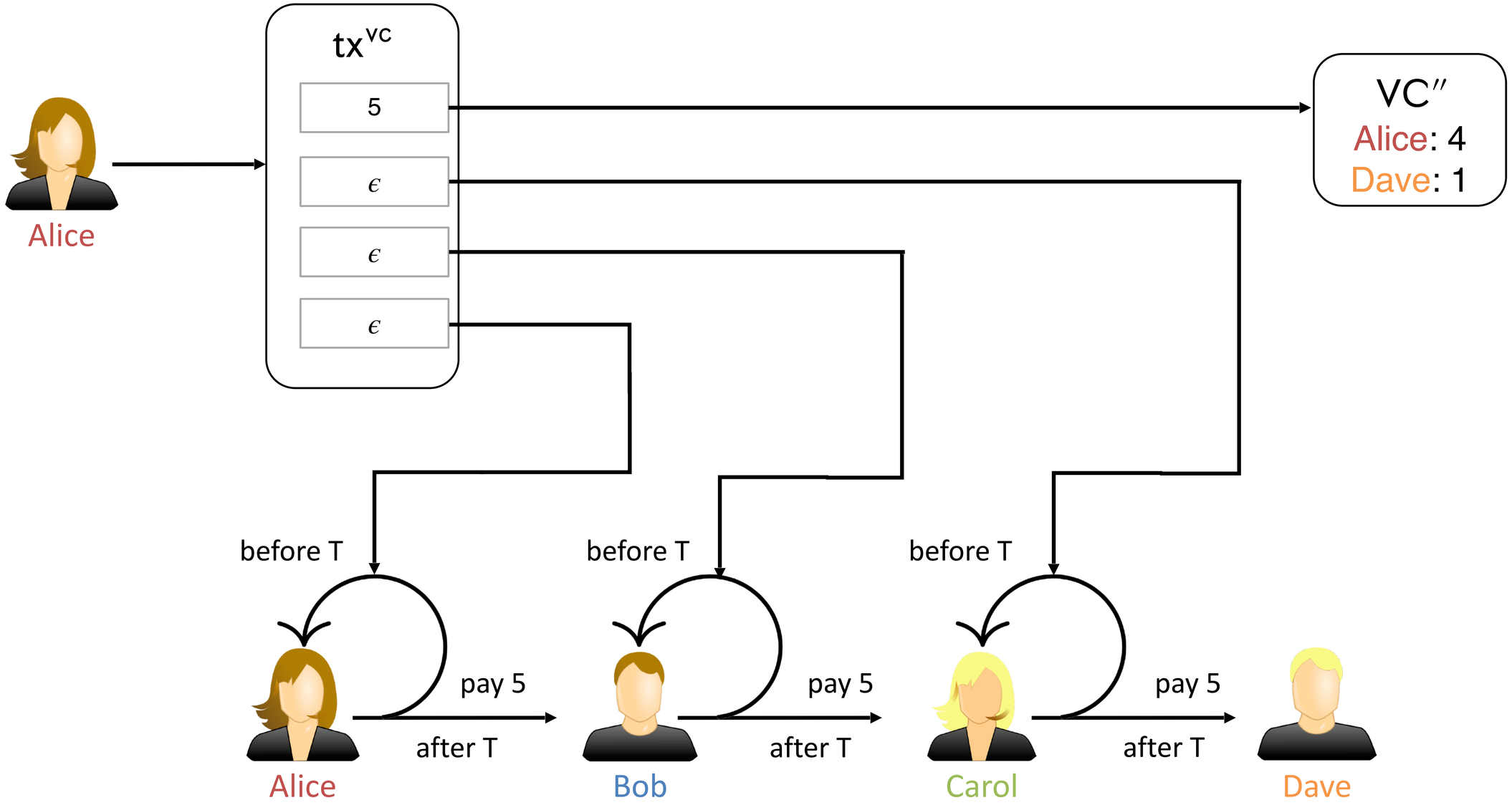


Alice and Dave update the VC by exchanging new commitment txs and revoking the previous ones.

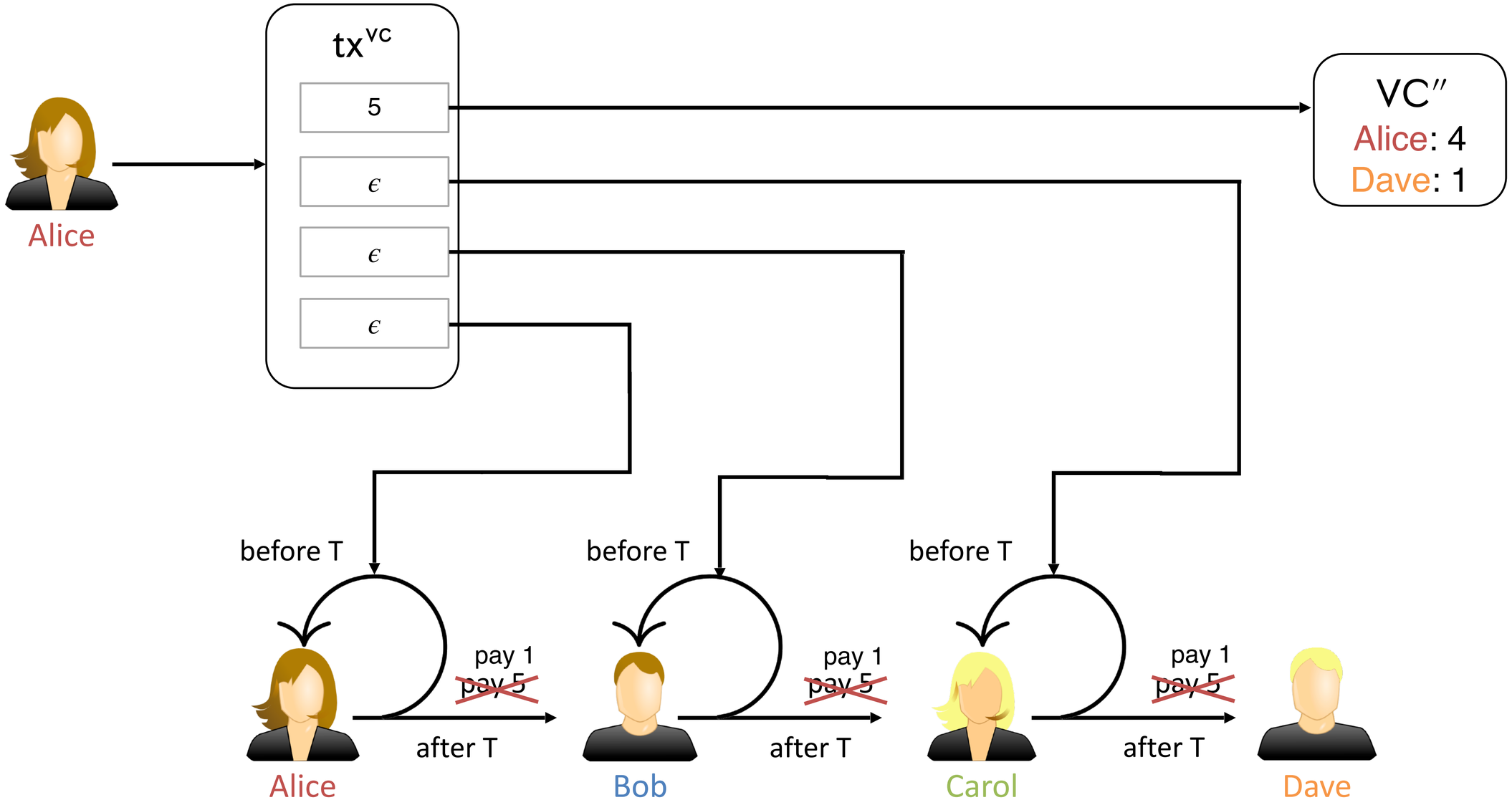


before T

Close VC

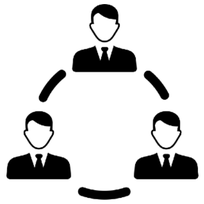

tx^{vc}
5
ϵ
ϵ
ϵ
Alice
VC″
Alice: 4
Dave: 1
before T
before T
before T
pay 5
pay 5
pay 5
after T
after T
after T
Alice
Bob
Carol
Dave

Close VC

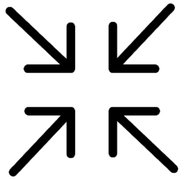

Alice
tx^vc
5
ϵ
ϵ
ϵ
VC″
Alice: 4
Dave: 1
before T
pay 1
pay 5
after T
Alice
before T
pay 1
pay 5
after T
Bob
before T
pay 1
pay 5
after T
Carol
Dave

Take home: Donner

- New virtual channel construction



Generic scalability solution for apps over multiple hops



Constant overhead

Fair, unlimited lifetime and fee model

Better security, privacy & latency

Formalized in UC framework

S&P'21

A^{2}L: Anonymous Atomic Locks for Scalability and Interoperability in Payment Channel Hubs

Erkan Tairi, Pedro Moreno-Sanchez and Matteo Maffei
TU Wien
{erkan.tairi,pedro.sanchez,matteo.maffei}@tuwien.ac.at

CCS'22

Foundations of Coin Mixing Services

Noemi Glaeser
University of Maryland & Max Planck Insitute for Security and Privacy
nglaeser@umd.edu

Matteo Maffei
TU Wien & Christian Doppler Laboratory Blockchain Technologies for the Internet of Things
matteo.maffei@tuwien.ac.at

Giulio Malavolta
Max Planck Institute for Security and Privacy
giulio.malavolta@hotmail.it

Sanchez

Erkan Tairi
TU Wien & Christian Doppler Laboratory Blockchain Technologies for the Internet of Things
erkan.tairi@tuwien.ac.at

Sri AravindaKrishnan Thyagarajan
Carnegie Mellon University
t.srikrishnan@gmail.com

Payment Channel Hubs

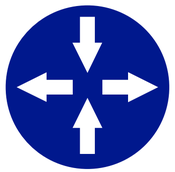
Miners accept to deviate from consensus if bribed

Payment Channel Hubs (PCH)

- The idea is to simplify setup, routing, and payments by having a central (untrusted) hub connecting users
  - Similar to a bank
- Challenge: how do we guarantee atomicity and privacy at the same time?
  - If the payer tells the bank whom to pay, privacy is gone (in contrast to Ligthning, the path has just length 2)

Payment in PCH: First Attempt

$C = k*G$



$C = k*G$

(pk_A, sk_A)

(pk_G, sk_G, k)

(pk_B, sk_B)

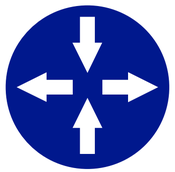
The first idea, for atomicity, is to rely on conditional payments and adaptor signatures, like in Lightning

AS

AS

Payment in PCH: First Attempt

C = k*G



C = k*G

(pk_A, sk_A)

(pk_G, sk_G, k)

(pk_B, sk_B)

AS

Lock

tx_G, sk_G

σ_G

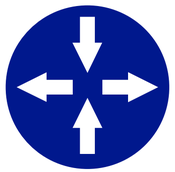
σ_G

AS

The second idea, for privacy, is to start a conditional payment from the payee's side!

Payment in PCH: First Attempt

C = k*G



C = k*G

(pk_A, sk_A)

(pk_G, sk_G, k)

(pk_B, sk_B)

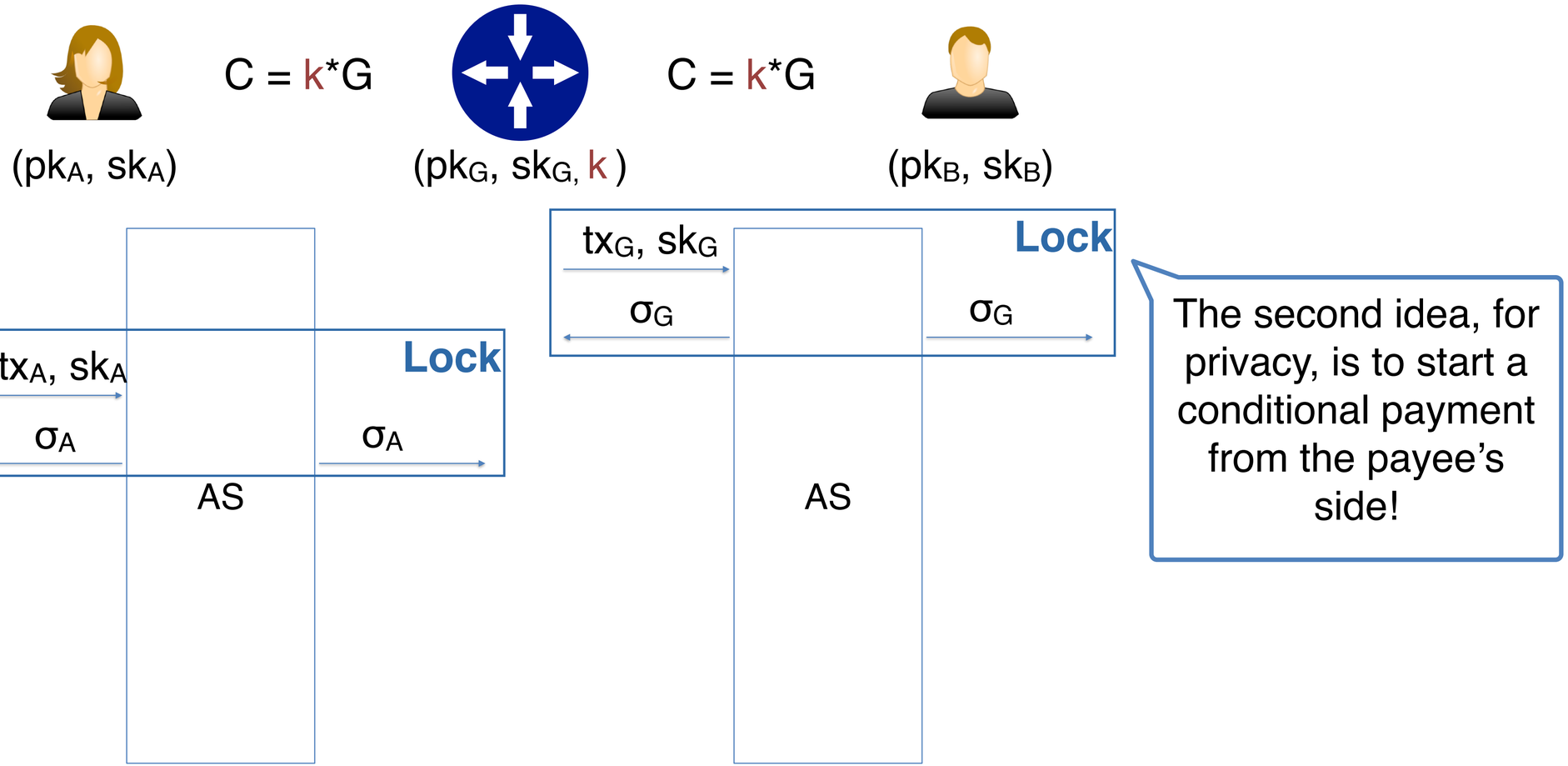
AS

Puzzle Promise

AS

The second idea, for privacy, is to start a conditional payment from the payee's side!

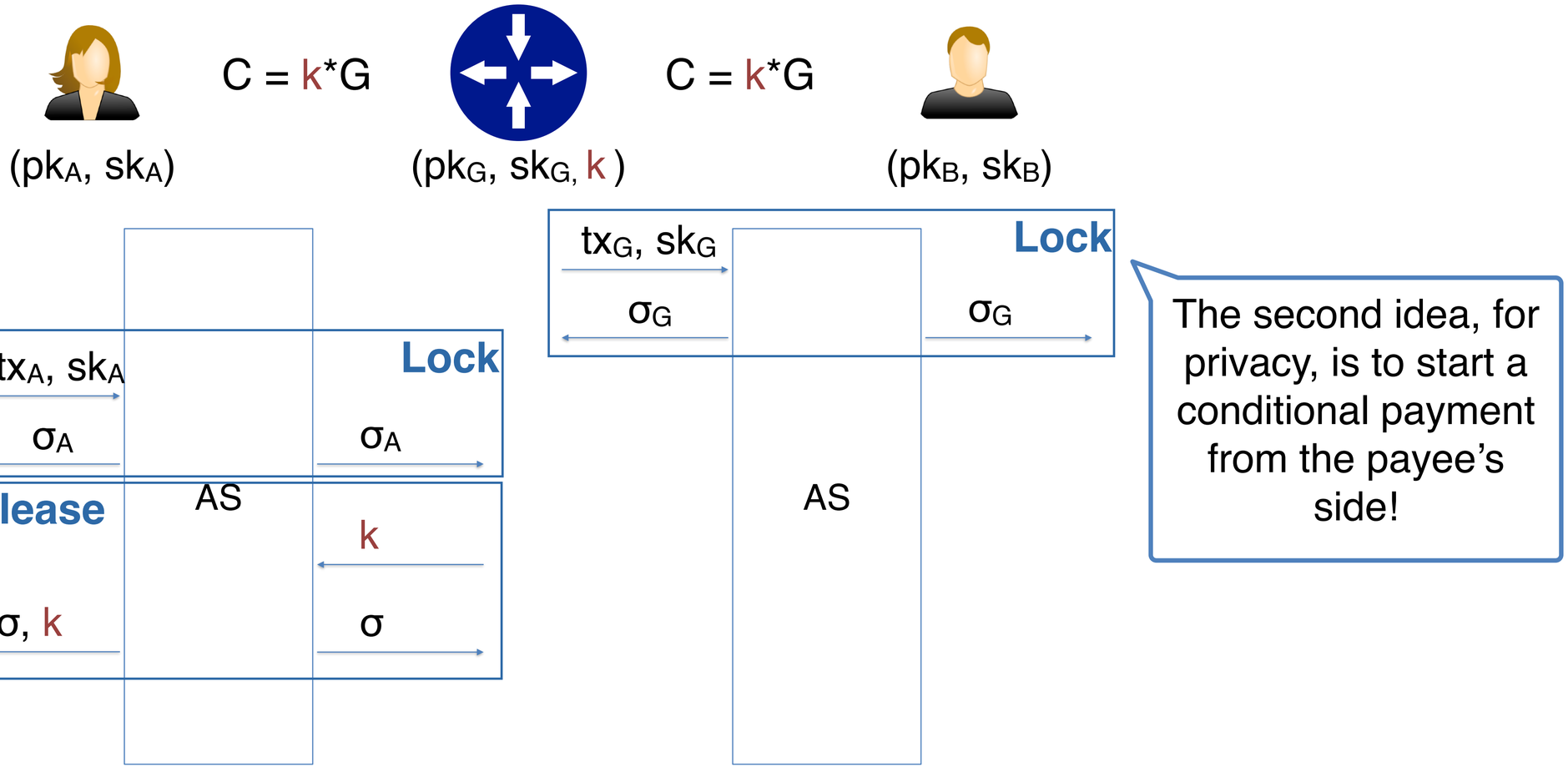
Payment in PCH: First Attempt



After the hub has issued a puzzle promise, Bob tells Alice to start the payment

The second idea, for privacy, is to start a conditional payment from the payee's side!

Payment in PCH: First Attempt

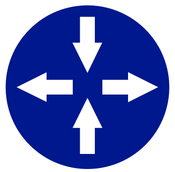


After the hub has issued a puzzle promise, Bob tells Alice to start the payment

The second idea, for privacy, is to start a conditional payment from the payee's side!

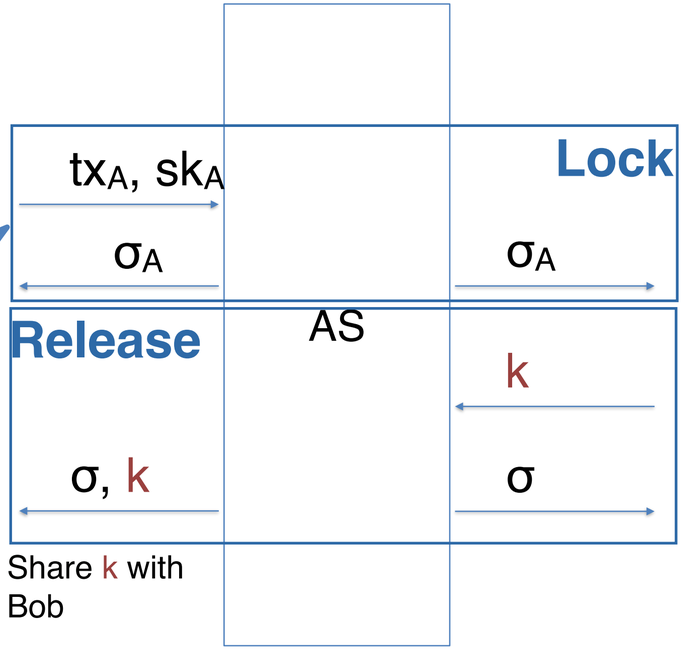
Payment in PCH: First Attempt

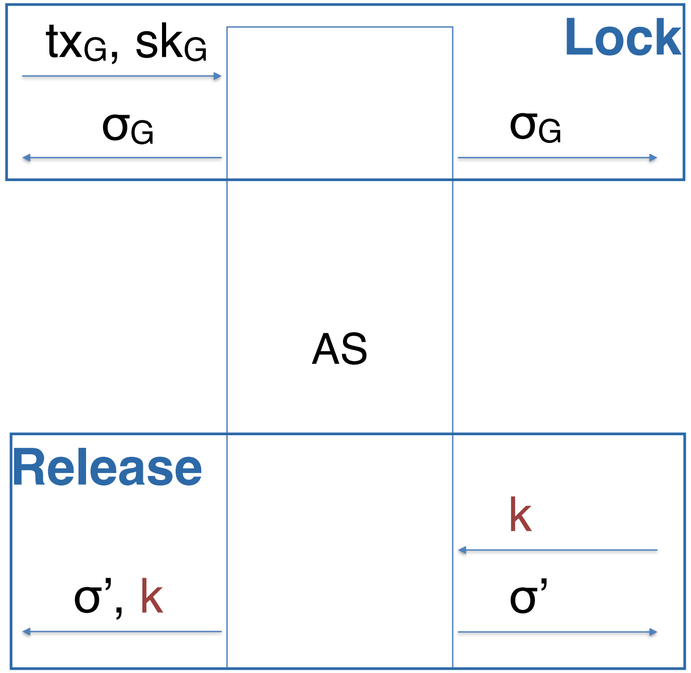
$C = k*G$



$C = k*G$

(pk_A, sk_A) (pk_G, sk_G, k) (pk_B, sk_B)



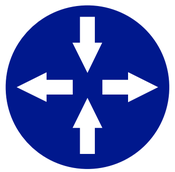


After the hub has issued a puzzle promise, Bob tells Alice to start the payment

The second idea, for privacy, is to start a conditional payment from the payee's side!

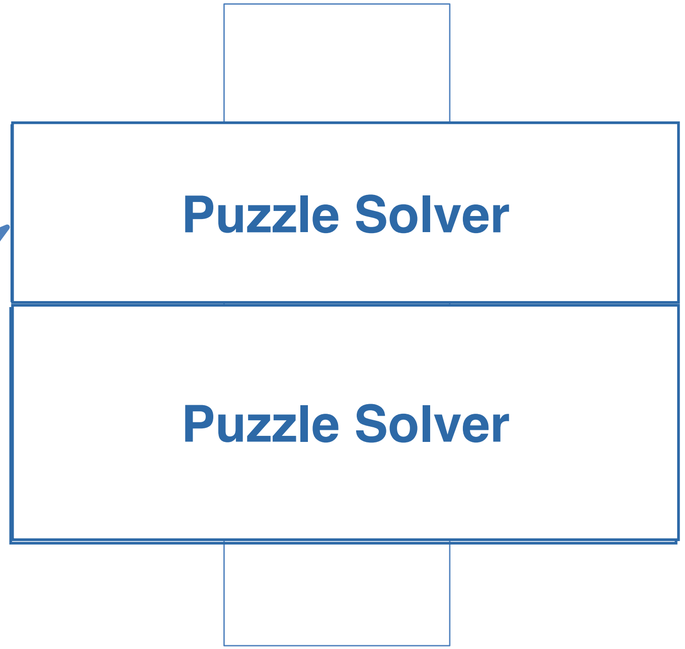
Payment in PCH: First Attempt

$C = k*G$

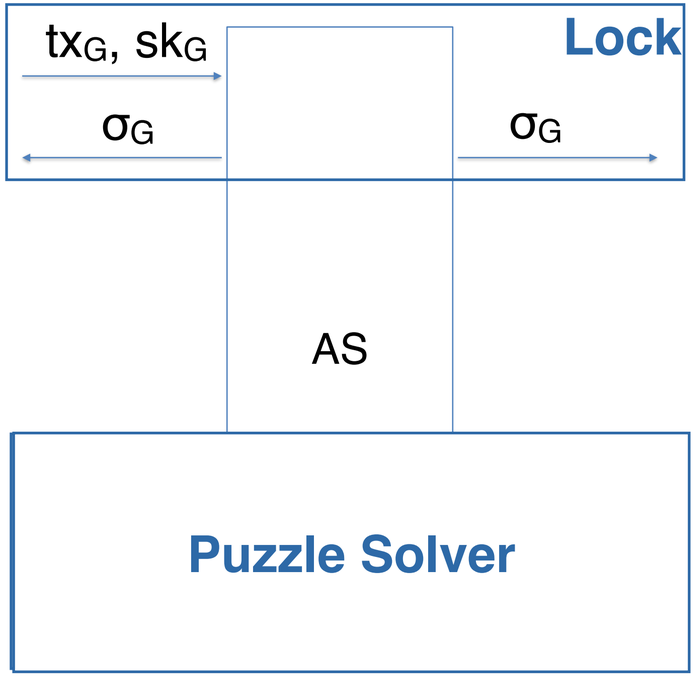


$C = k*G$

(pk_A, sk_A) (pk_G, sk_G, k) (pk_B, sk_B)

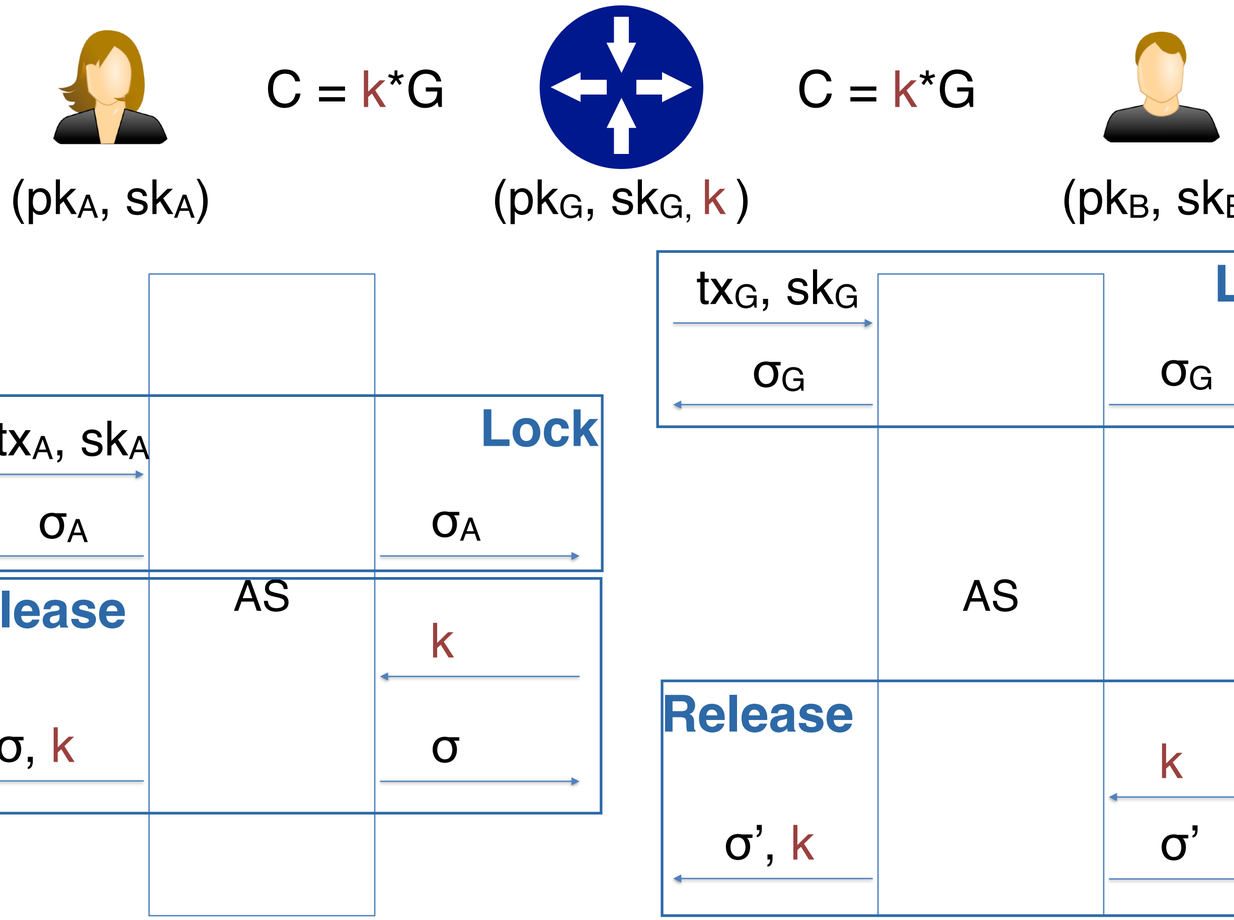


After the hub has issued a puzzle promise, Bob tells Alice to start the payment



The second idea, for privacy, is to start a conditional payment from the payee's side!

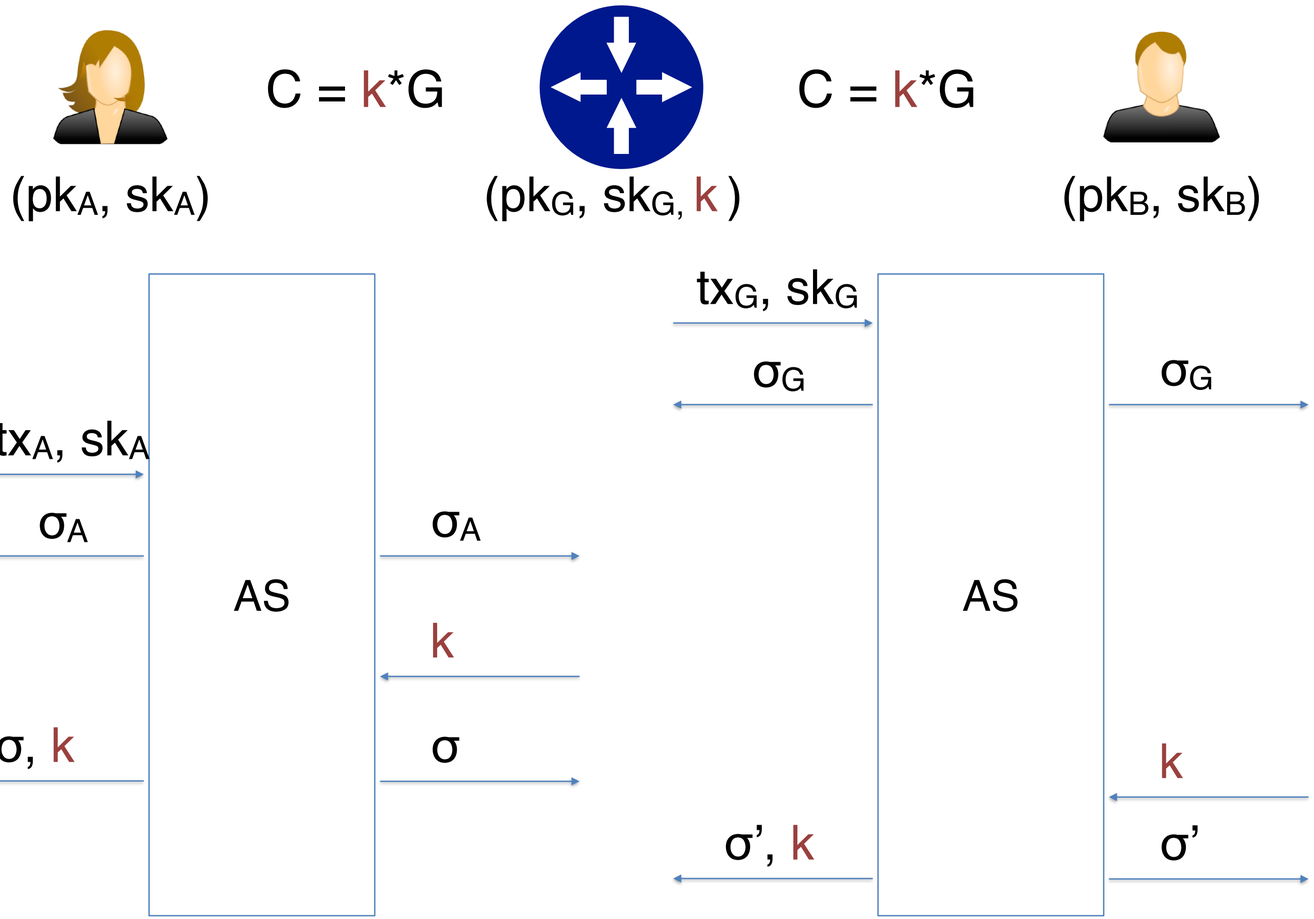
Payment in PCH: First Attempt



After the hub has issued a puzzle promise, Bob tells Alice to start the payment

The second idea, for privacy, is to start a conditional payment from the payee's side!

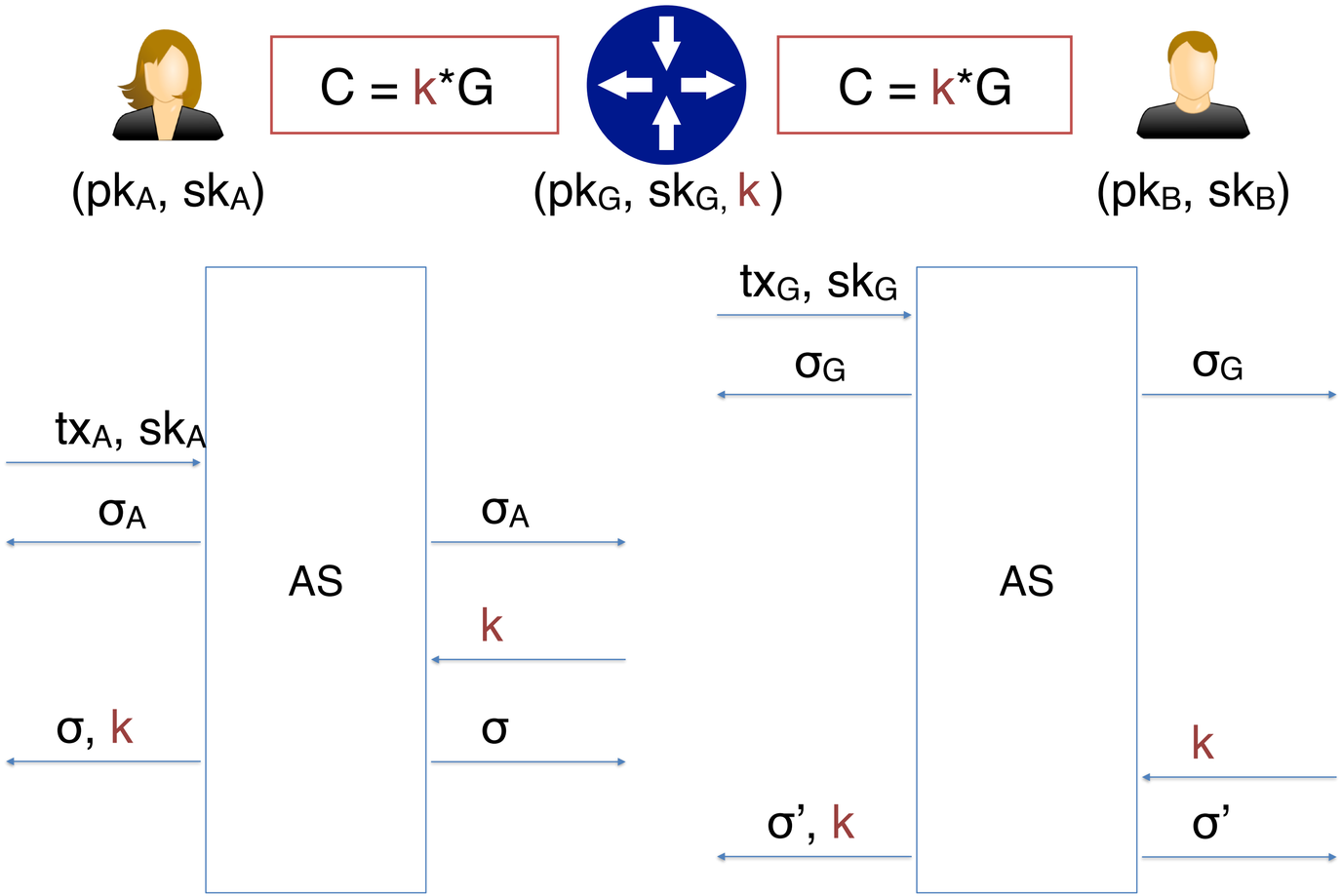
Privacy Issue



✓ The payee does not have to tell the hub whom she wants to pay!

✗ But…the condition is the same on both signatures, so payer and payee can be linked!

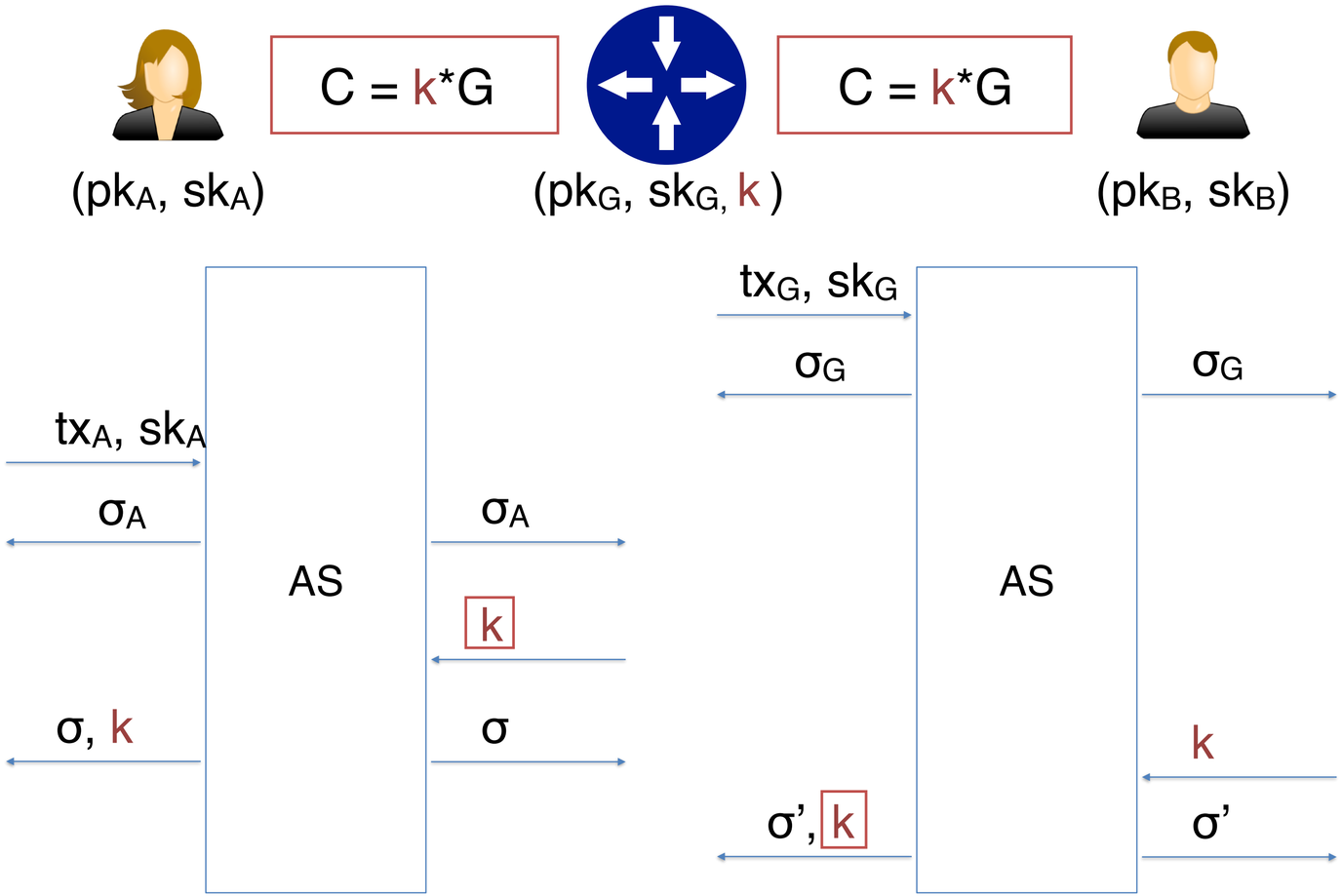
Privacy Issue



✓ The payee does not have to tell the hub whom she wants to pay!

✗ But…the condition is the same on both signatures, so payer and payee can be linked!

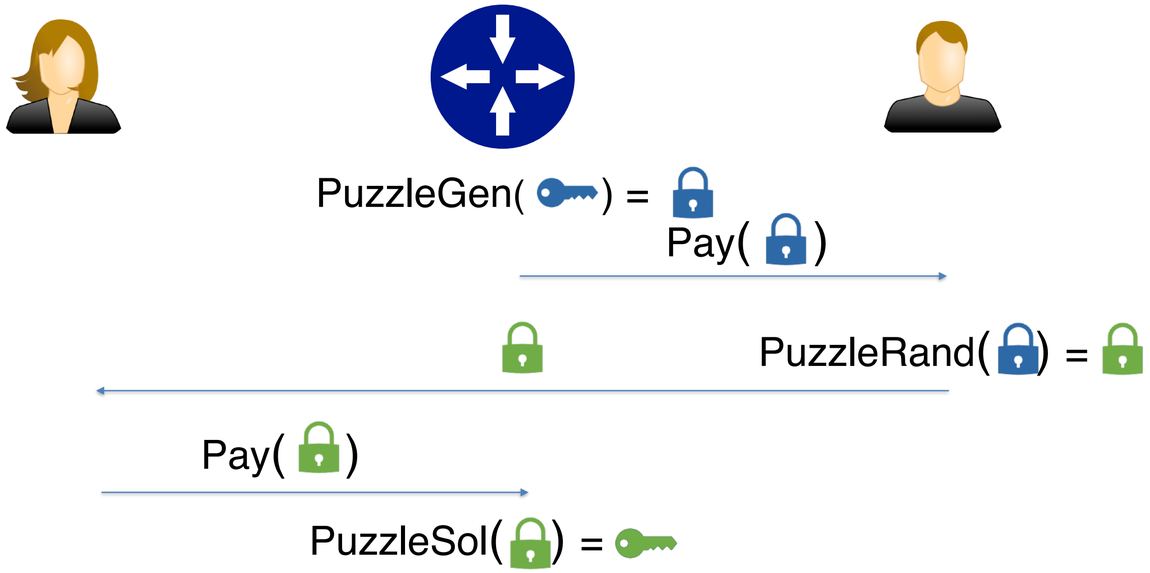
Privacy Issue



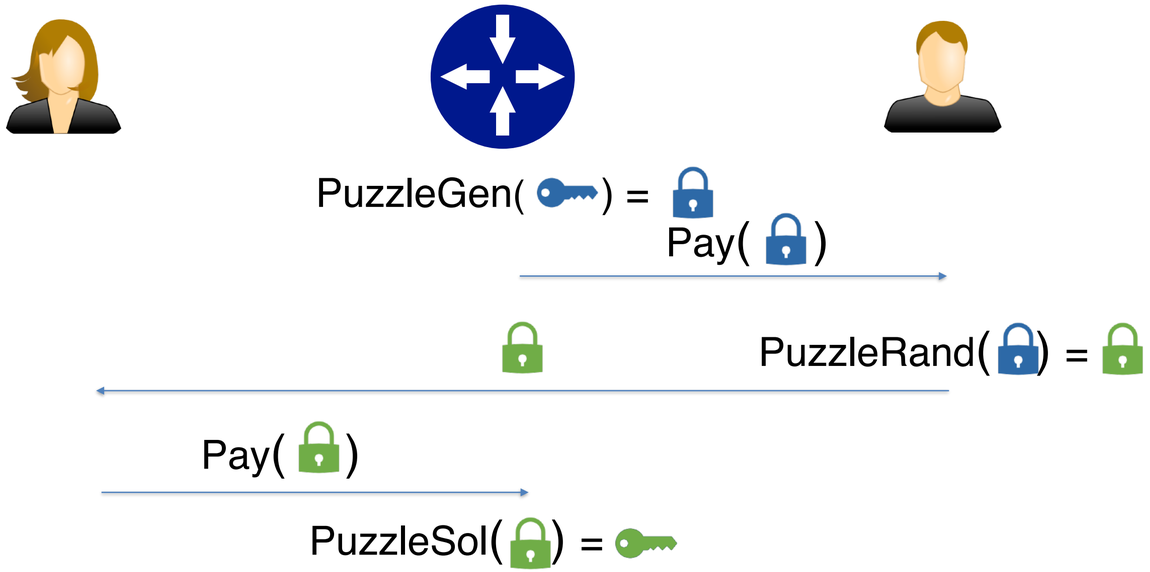
The payee does not have to tell the hub whom she wants to pay!

But…the condition is the same on both signatures, so payer and payee can be linked!

Privacy Solution

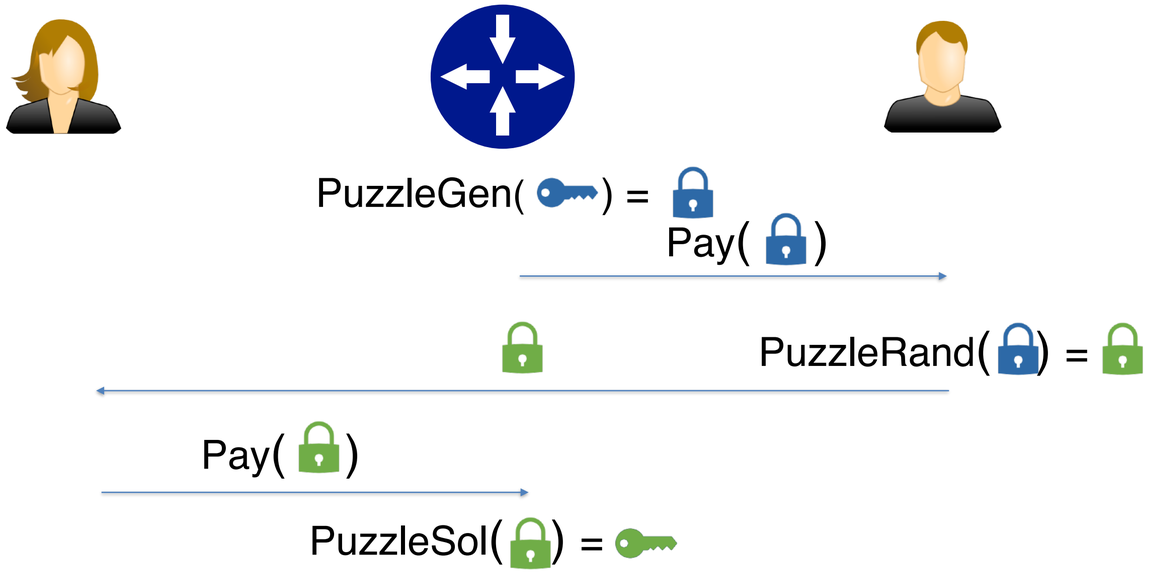


Privacy Solution



- Recall in our case the puzzle is the condition C = k*G, and the solution is the secret k. Hence, the randomized puzzle would correspond to computing C' = r*k*G, for a random scalar r, and randomized solution is r*k.

Privacy Solution



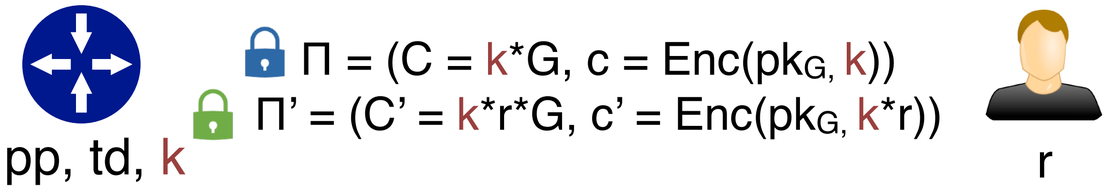
- Recall in our case the puzzle is the condition C = k*G, and the solution is the secret k. Hence, the randomized puzzle would correspond to computing C' = r*k*G, for a random scalar r, and randomized solution is r*k.
- Gateway cannot solve the puzzle now as it does not know r. The solution is to extend the puzzle with the encryption of the secret k under the gateway's key.

Building Block: Randomizable Puzzle

- Randomizable puzzle combines the condition of adaptor signature with an encryption under additively homomorphic encryption scheme
- Goals:
  - Gateway creates a puzzle  that can be solved using a trapdoor (e.g., secret key)
  - The puzzle can be randomized to create a fresh looking version 

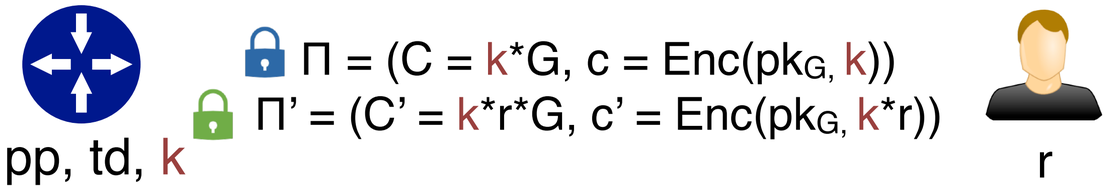
Building Block: Randomizable Puzzle

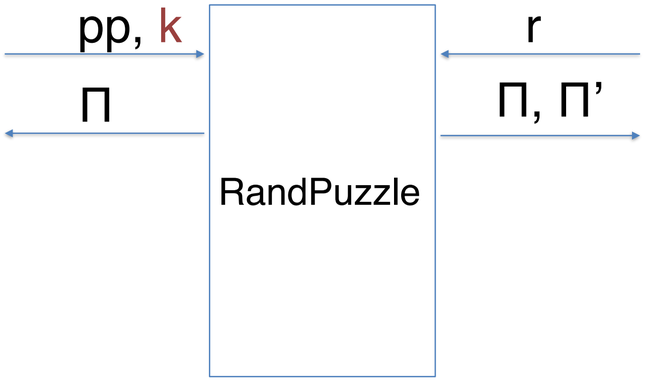
- Randomizable puzzle combines the condition of adaptor signature with an encryption under additively homomorphic encryption scheme
- Goals:
  - Gateway creates a puzzle that can be solved using a trapdoor (e.g., secret key)
  - The puzzle can be randomized to create a fresh looking version



Building Block: Randomizable Puzzle

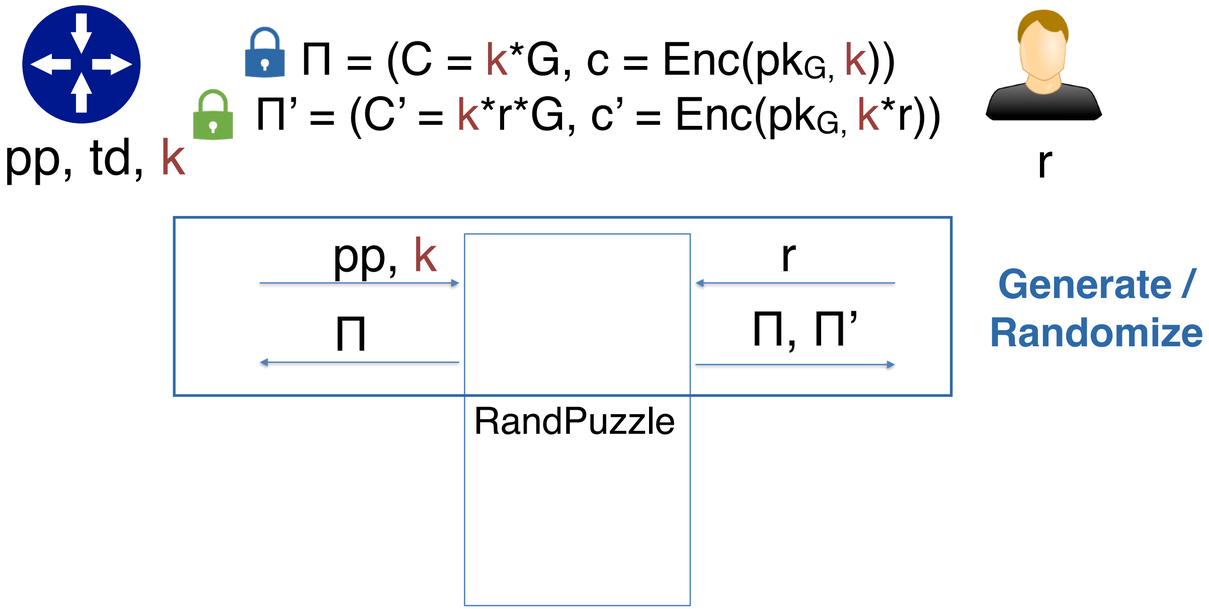
- Randomizable puzzle combines the condition of adaptor signature with an encryption under additively homomorphic encryption scheme
- Goals:
  - Gateway creates a puzzle that can be solved using a trapdoor (e.g., secret key)
  - The puzzle can be randomized to create a fresh looking version





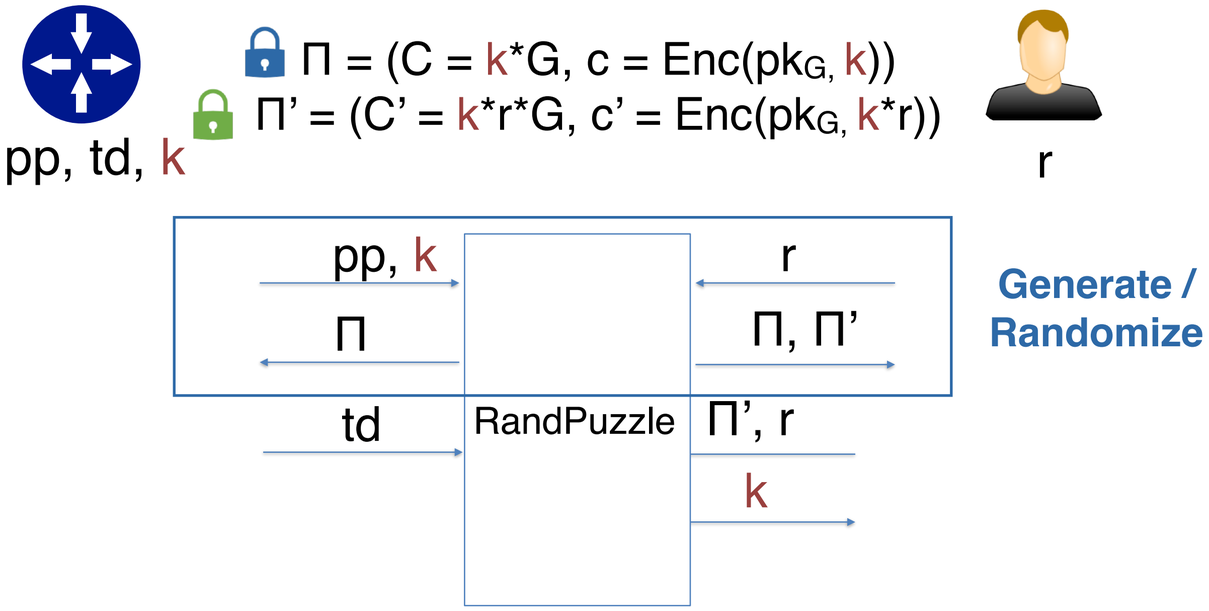
Building Block: Randomizable Puzzle

- Randomizable puzzle combines the condition of adaptor signature with an encryption under additively homomorphic encryption scheme
- Goals:
  - Gateway creates a puzzle that can be solved using a trapdoor (e.g., secret key)
  - The puzzle can be randomized to create a fresh looking version



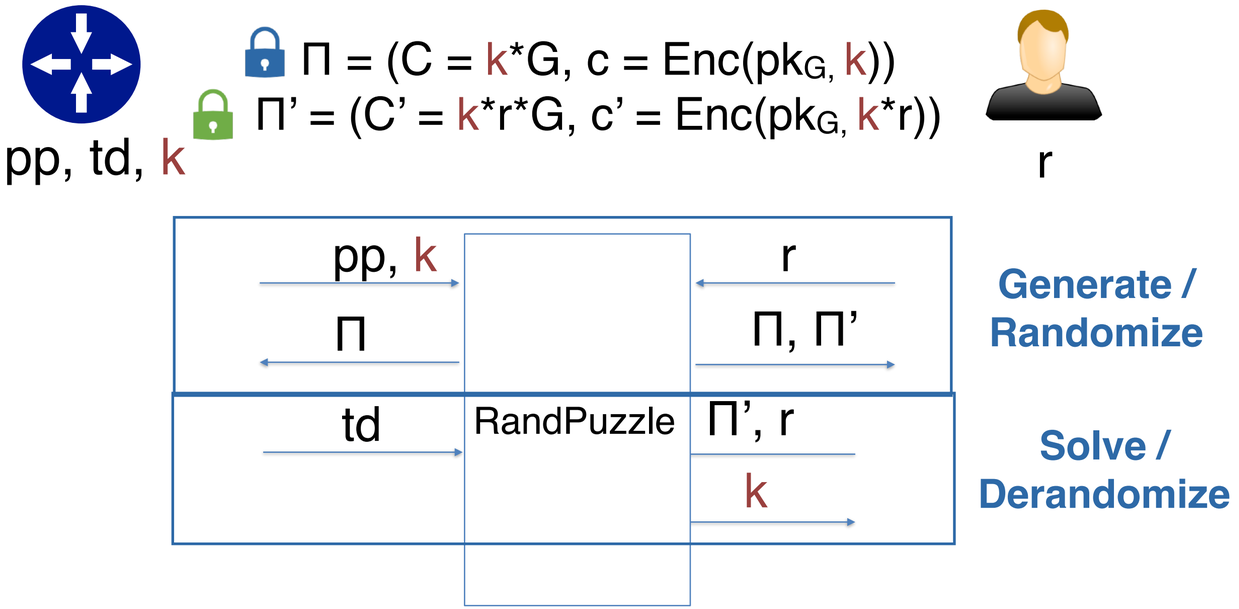
Building Block: Randomizable Puzzle

- Randomizable puzzle combines the condition of adaptor signature with an encryption under additively homomorphic encryption scheme
- Goals:
  - Gateway creates a puzzle that can be solved using a trapdoor (e.g., secret key)
  - The puzzle can be randomized to create a fresh looking version



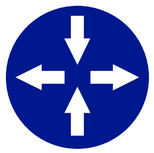
Building Block: Randomizable Puzzle

- Randomizable puzzle combines the condition of adaptor signature with an encryption under additively homomorphic encryption scheme
- Goals:
  - Gateway creates a puzzle that can be solved using a trapdoor (e.g., secret key)
  - The puzzle can be randomized to create a fresh looking version



A²L: Protocol Overview

(pk_A, sk_A)



$(pk_G, sk_G, pp_G, td_G, k)$

$C = k*G$

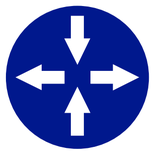
r_B

AS + RandPuzzle

AS + RandPuzzle

A²L: Protocol Overview

(pk_A, sk_A)



$C = k*G$

$(pk_G, sk_G, pp_G, td_G, k)$

r_B

AS + RandPuzzle

tx_G, sk_G, pp_G

σ_G, Π

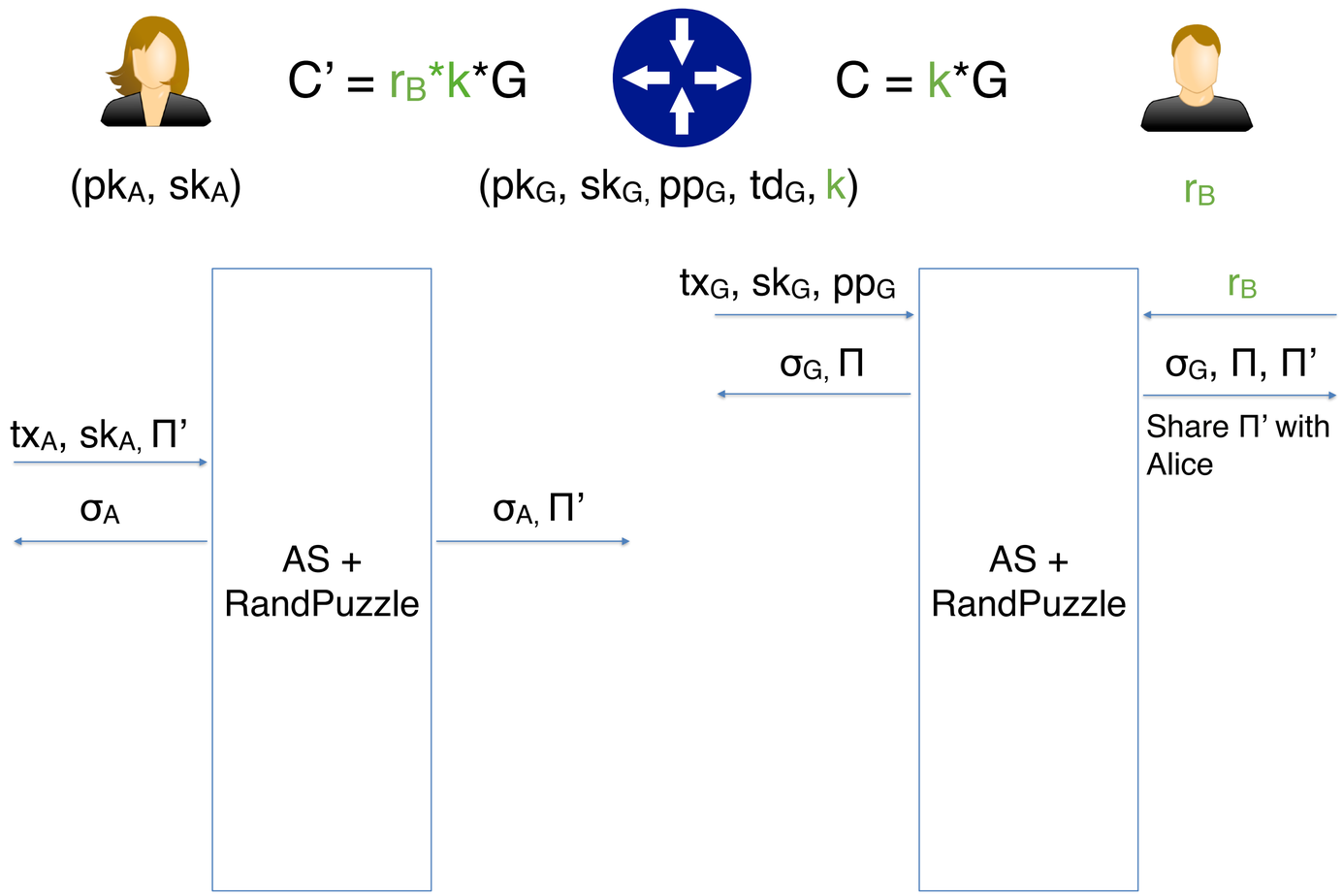
r_B

σ_G, Π, Π'

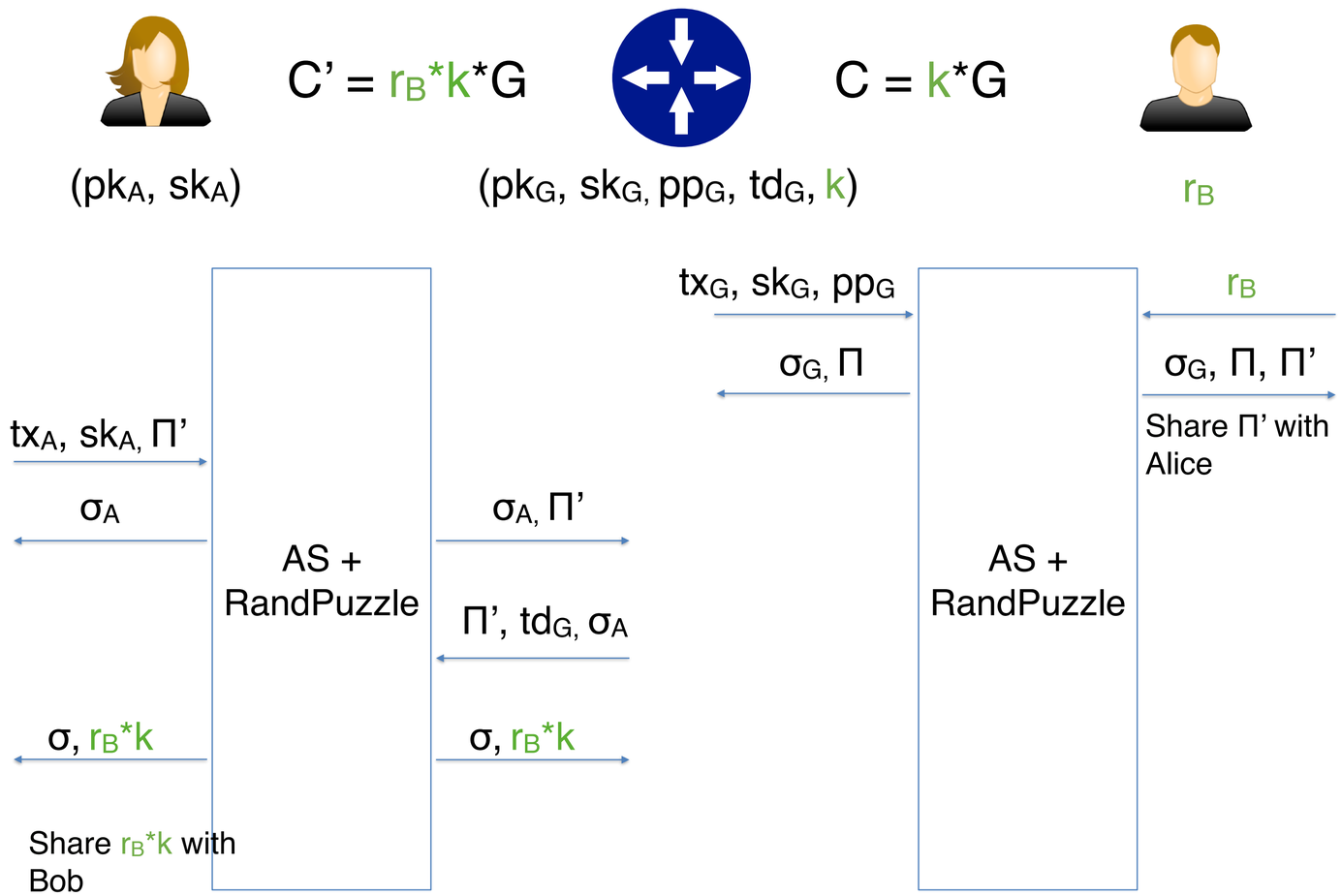
Share Π' with Alice

AS + RandPuzzle

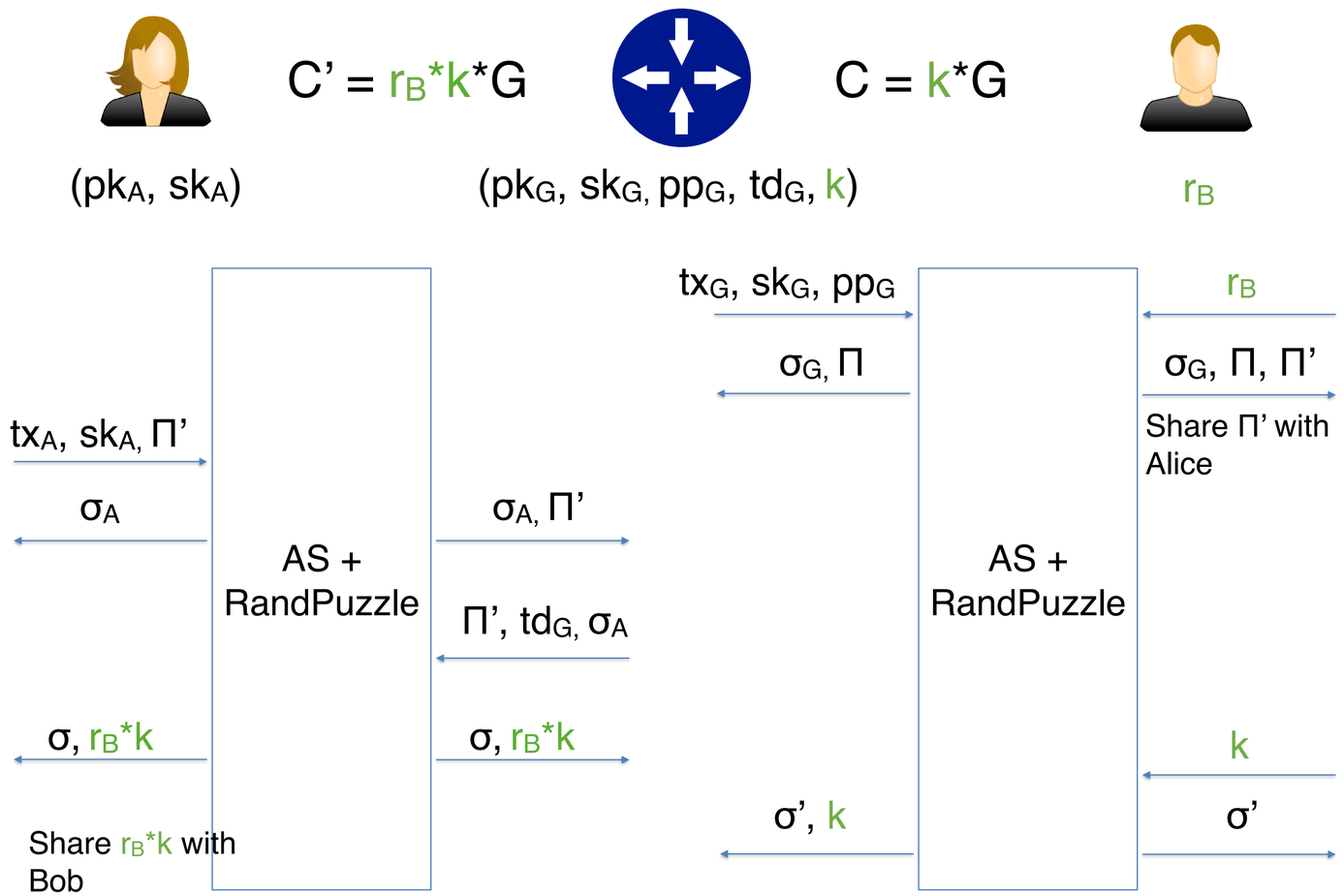
A²L: Protocol Overview



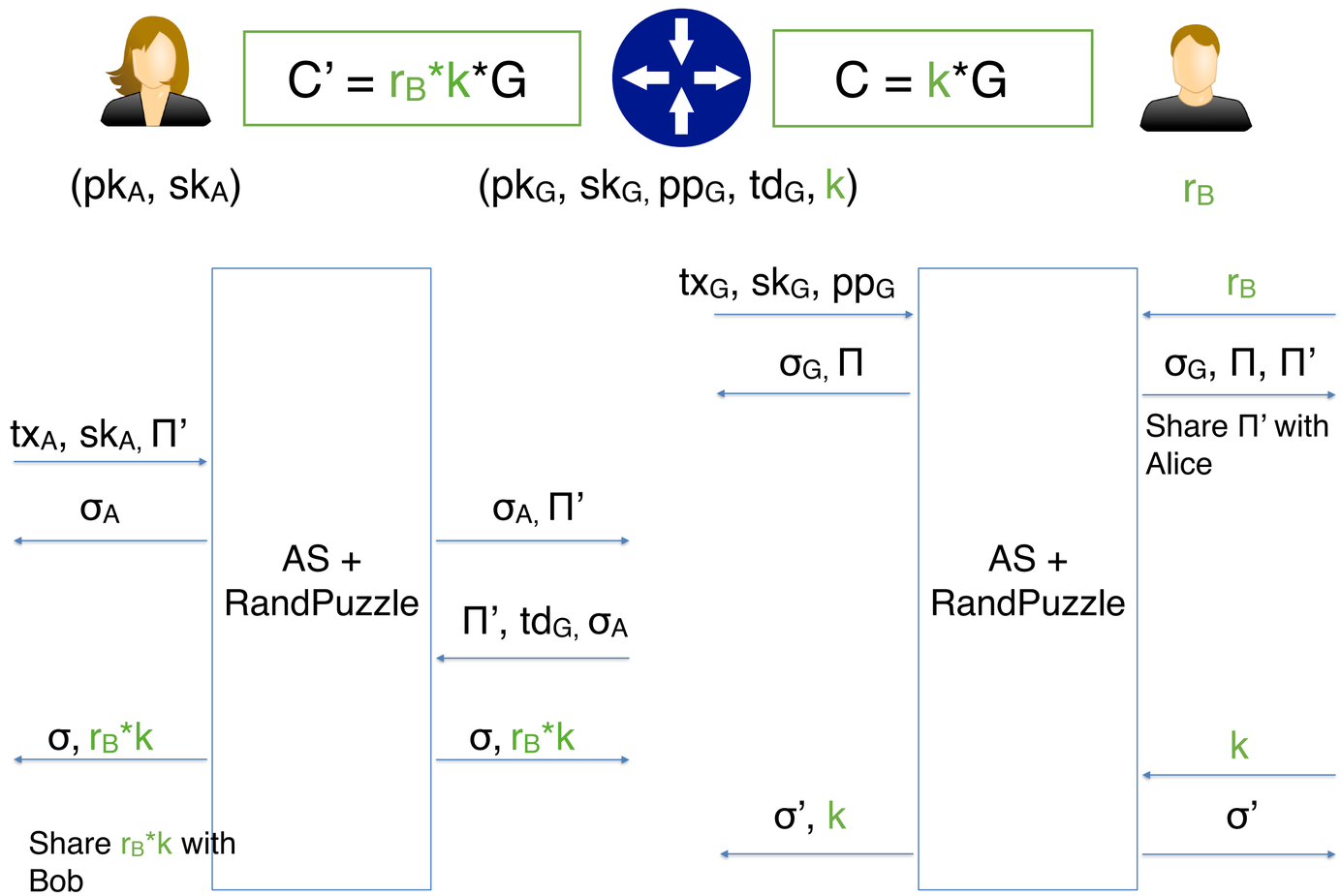
A²L: Protocol Overview



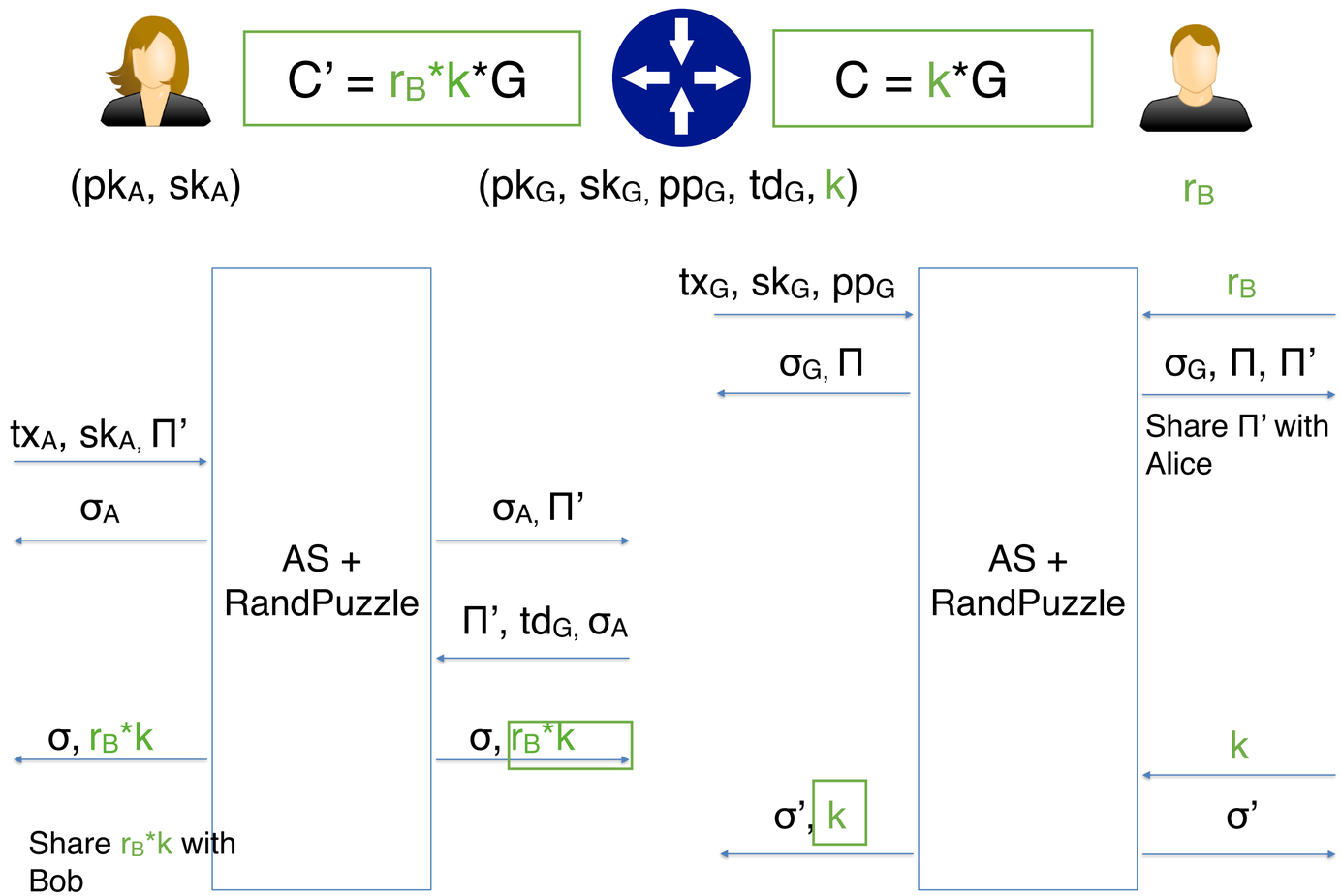
A²L: Protocol Overview



A²L: Protocol Overview



A²L: Protocol Overview

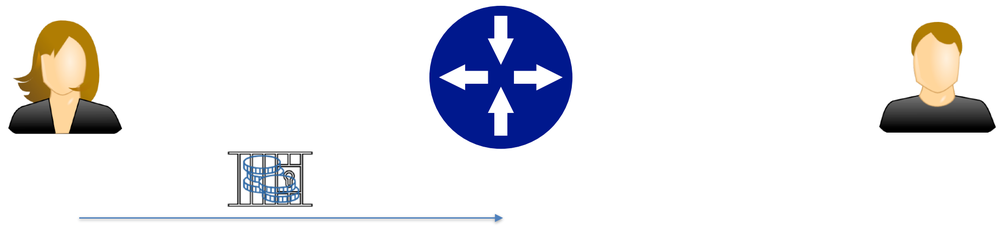


Griefing Protection

- Privacy-preserving registration protocol to protect against griefing attacks (like a user forcing the hub to lock money in a lot of puzzles…)

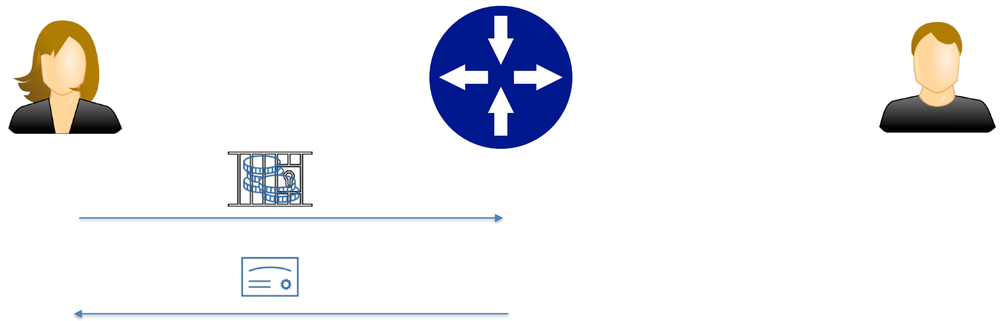
Griefing Protection

- Privacy-preserving registration protocol to protect against griefing attacks (like a user forcing the hub to lock money in a lot of puzzles…)



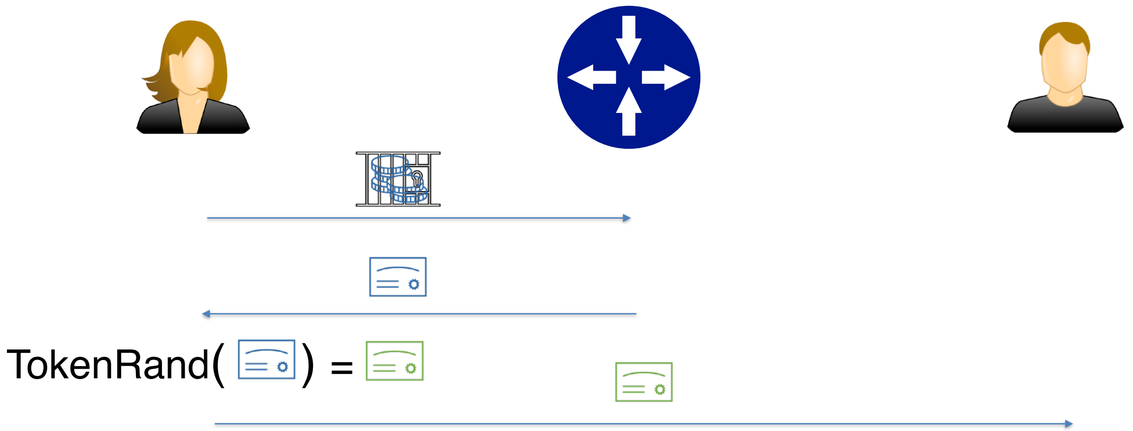
Griefing Protection

- Privacy-preserving registration protocol to protect against griefing attacks (like a user forcing the hub to lock money in a lot of puzzles…)



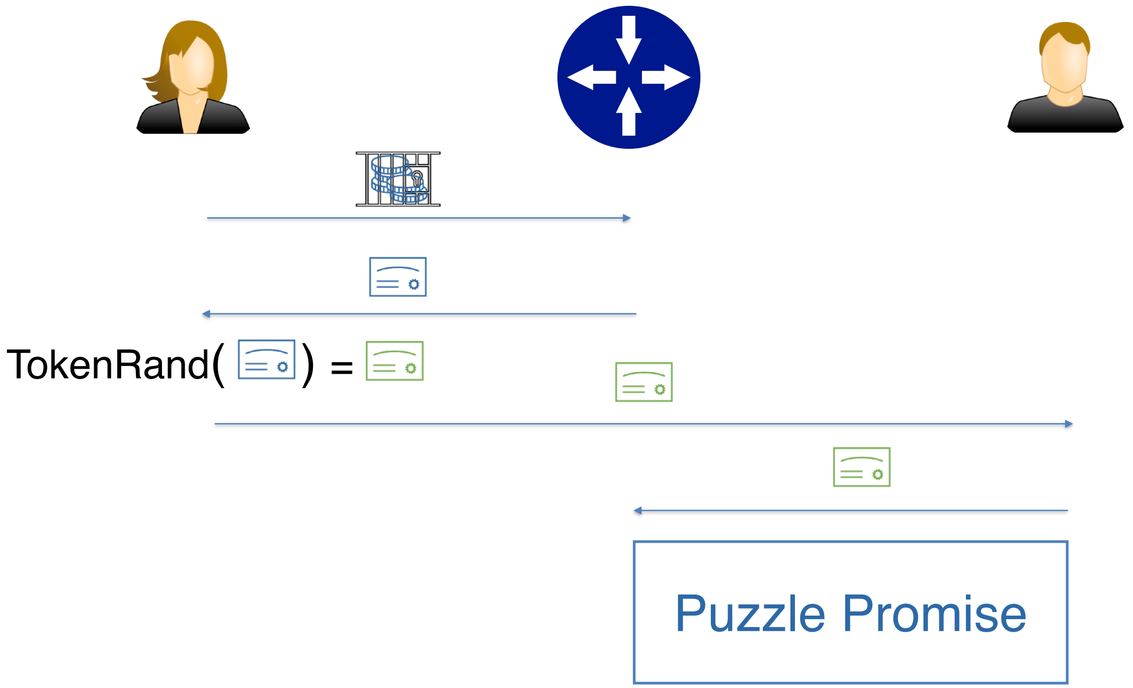
Griefing Protection

- Privacy-preserving registration protocol to protect against griefing attacks (like a user forcing the hub to lock money in a lot of puzzles…)

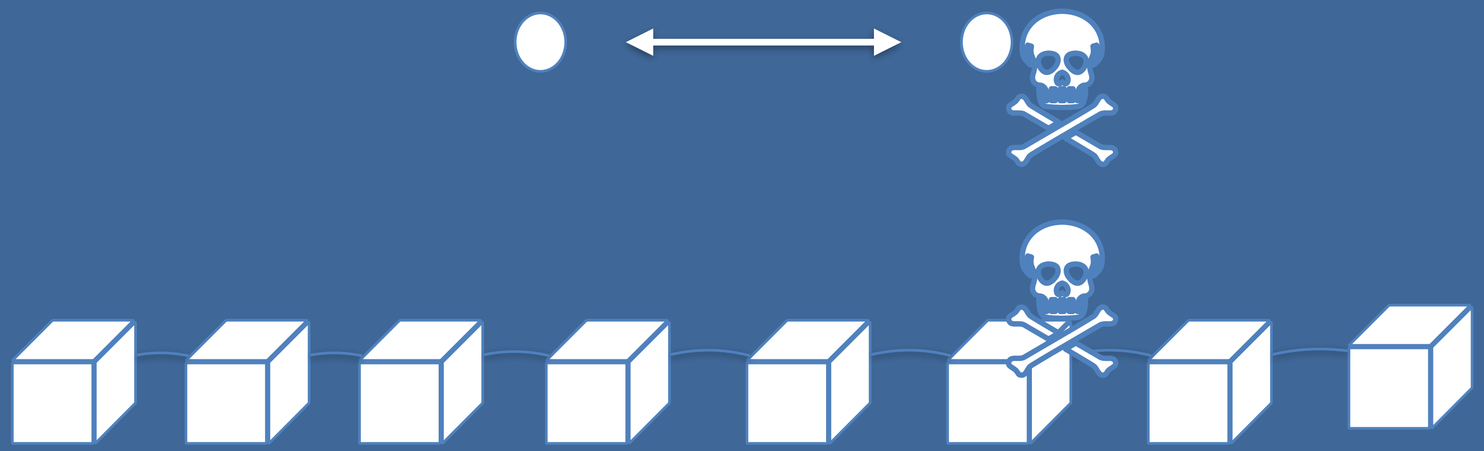


Griefing Protection

- Privacy-preserving registration protocol to protect against griefing attacks (like a user forcing the hub to lock money in a lot of puzzles…)

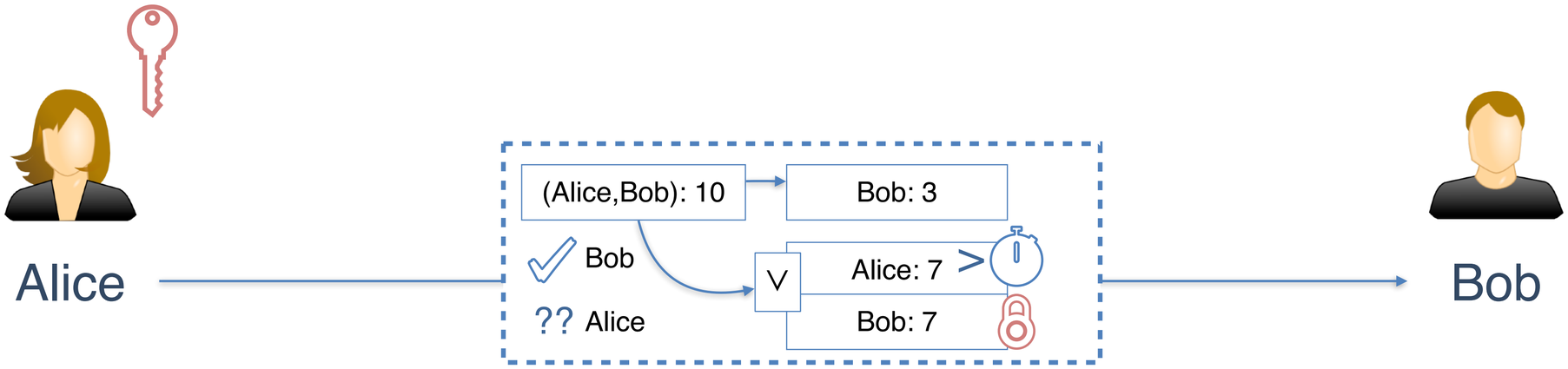


Bribing Attacks
(Or Layer-2 breaks Layer-1)



Miners accept to deviate from consensus if bribed

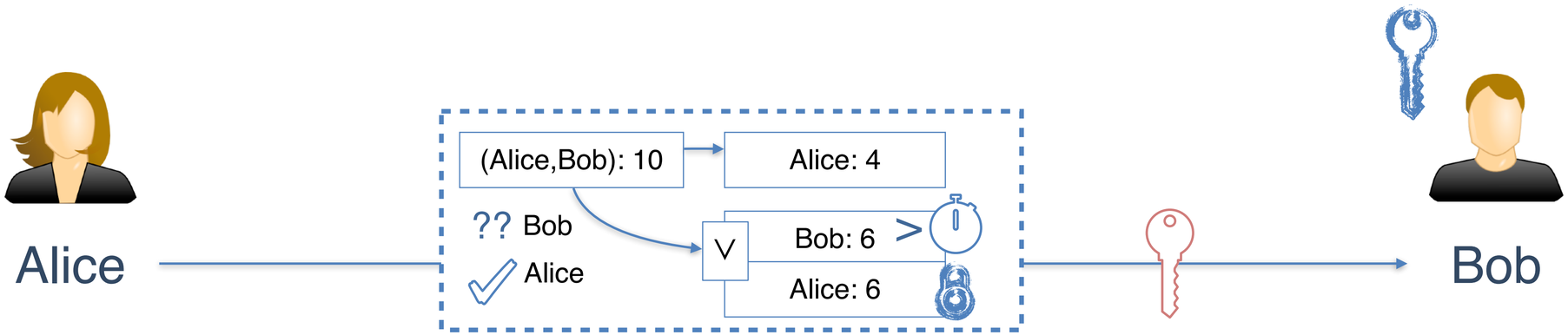
What if miners are bribed?



Blockchain

- Alice first has 7 coins…

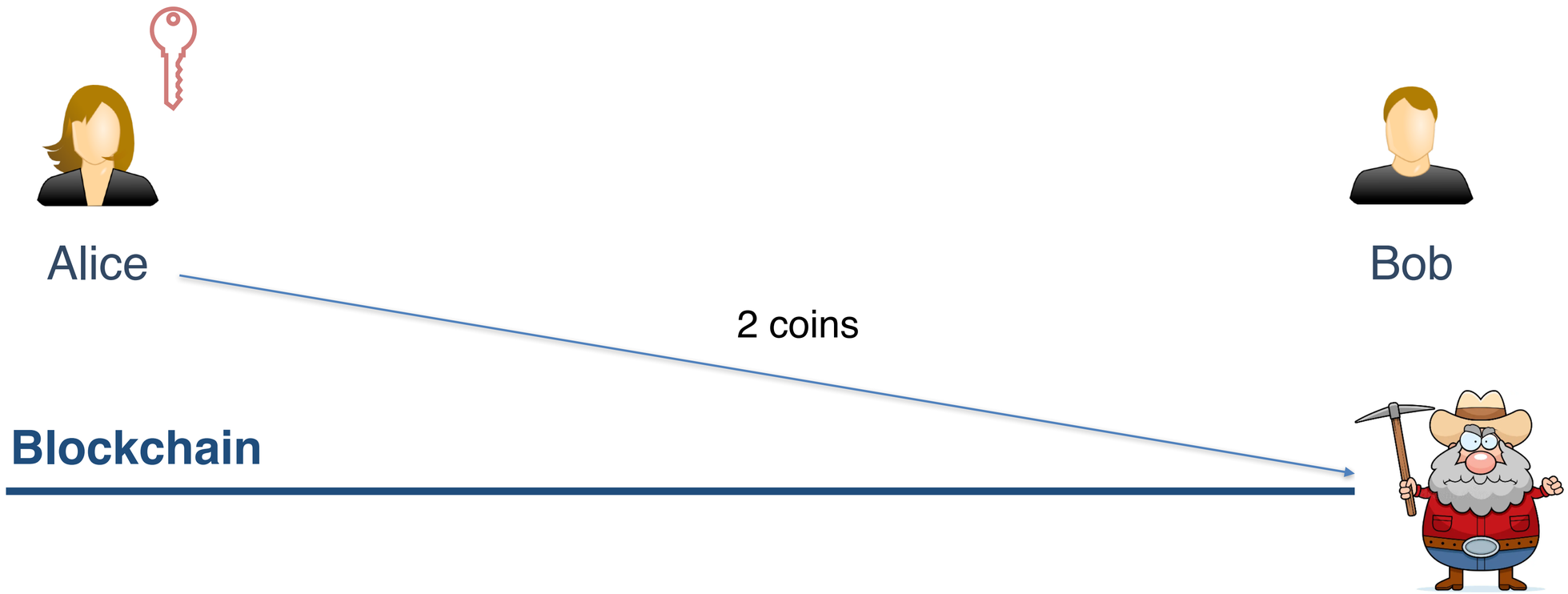
What if miners are bribed?



Blockchain

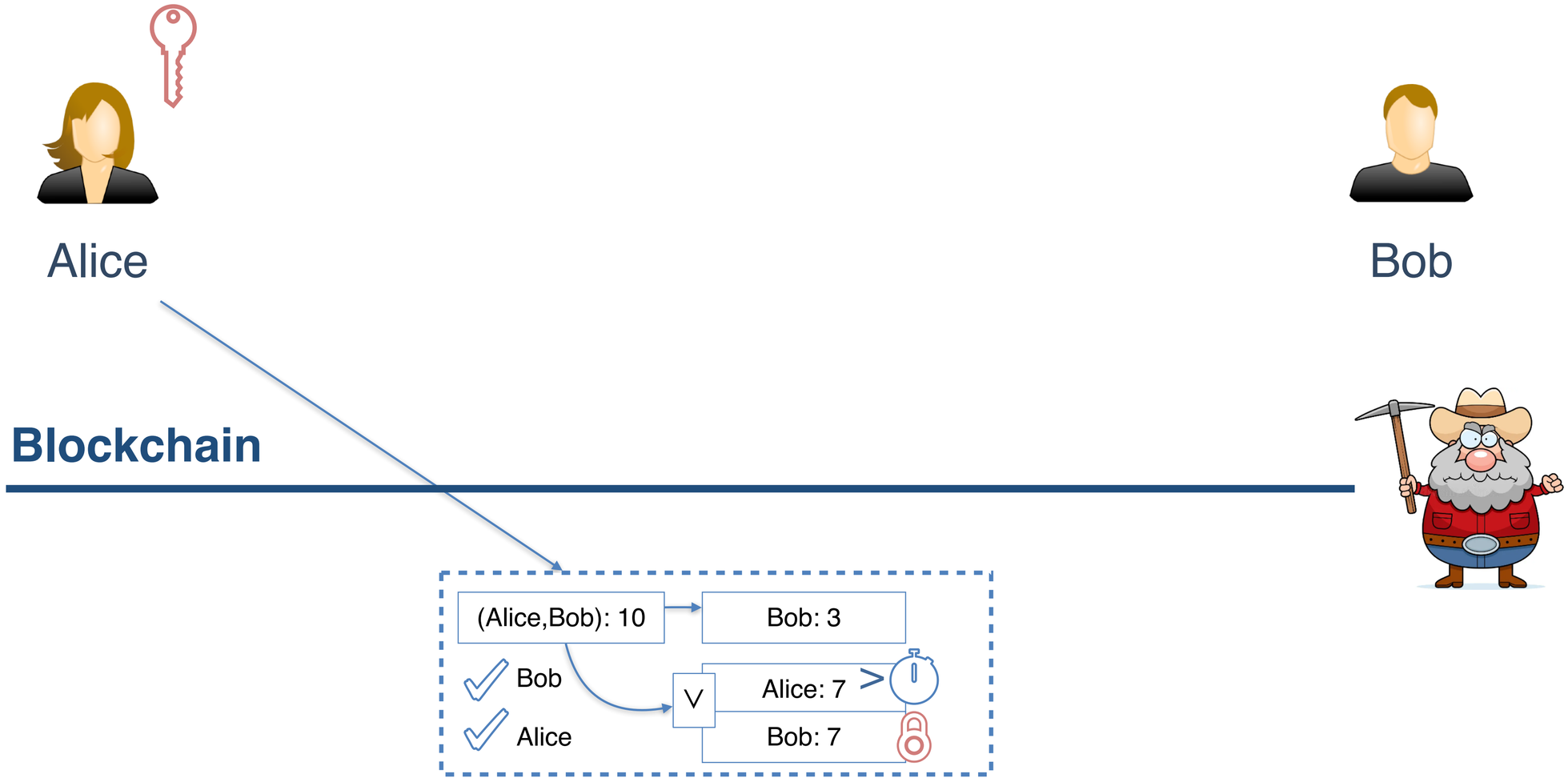
- Then she pays 3 to Bob and reveals the old key

What if miners are bribed?



- Now Alice first bribes the miner…

What if miners are bribed?



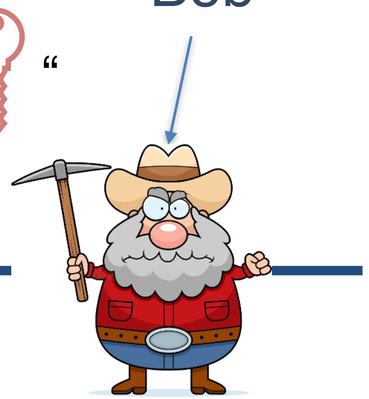
- And then posts the old channel balance on-chain

What if miners are bribed?

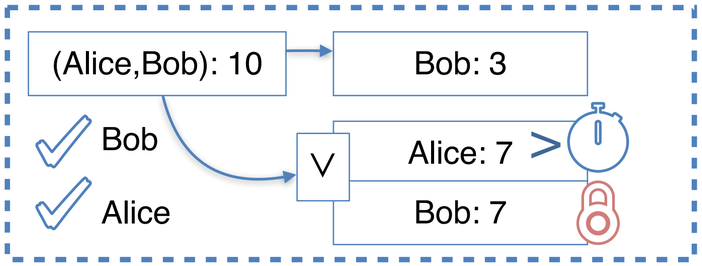
Alice

Bob

“I claim 7 coins with “



Blockchain



- Bob tries to punish Alice before the timeout, but the miners do not post the transaction on chain

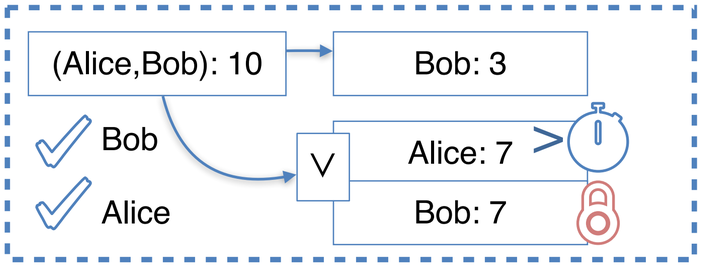
What if miners are bribed?

Alice

Bob

"I claim 7 coins"

Blockchain



- After the timeout, Alice gets 7 coins.

State-of-the-art

- Currently covers just HTLCs (not payment channels)
- Mad-HTCL:
  - Incentivize miners to punish misbehaving users
  - Game-theoretic security against passive miner strategies
- HE-HTLC
  - Game-theoretic security against active miner strategies

IEEE S&P 2021

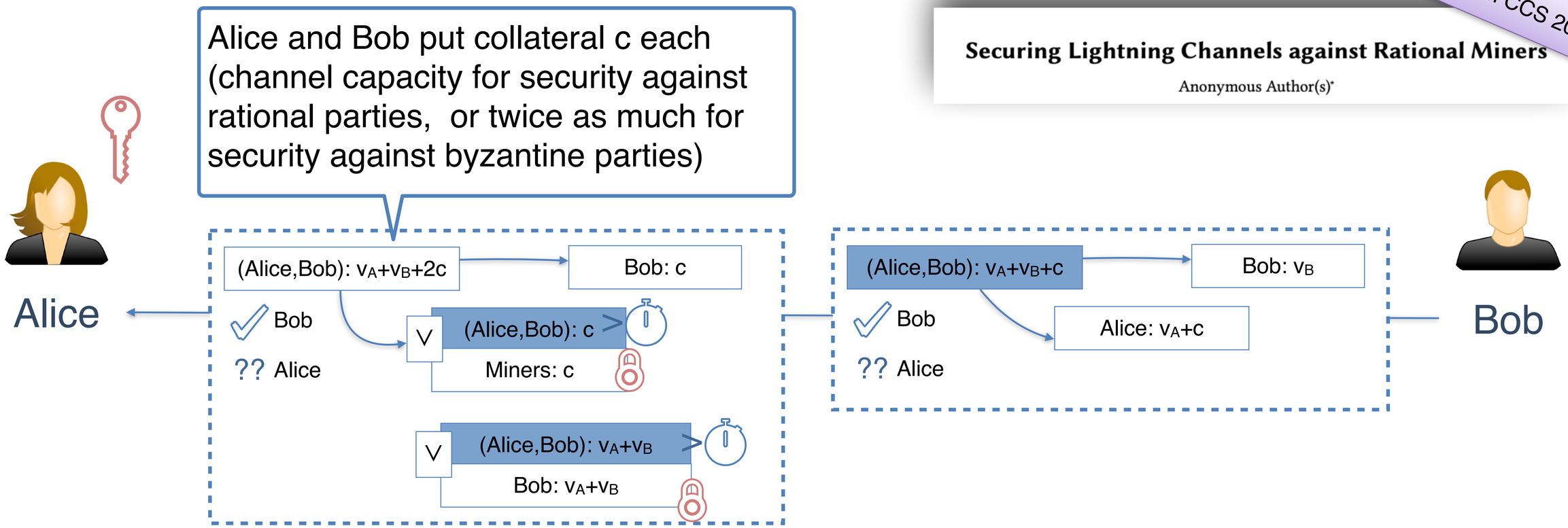
NDSS 2023

CRAB (Channel Resistant Against Bribery)

ACM CCS 2024

Securing Lightning Channels against Rational Miners

Anonymous Author(s)*



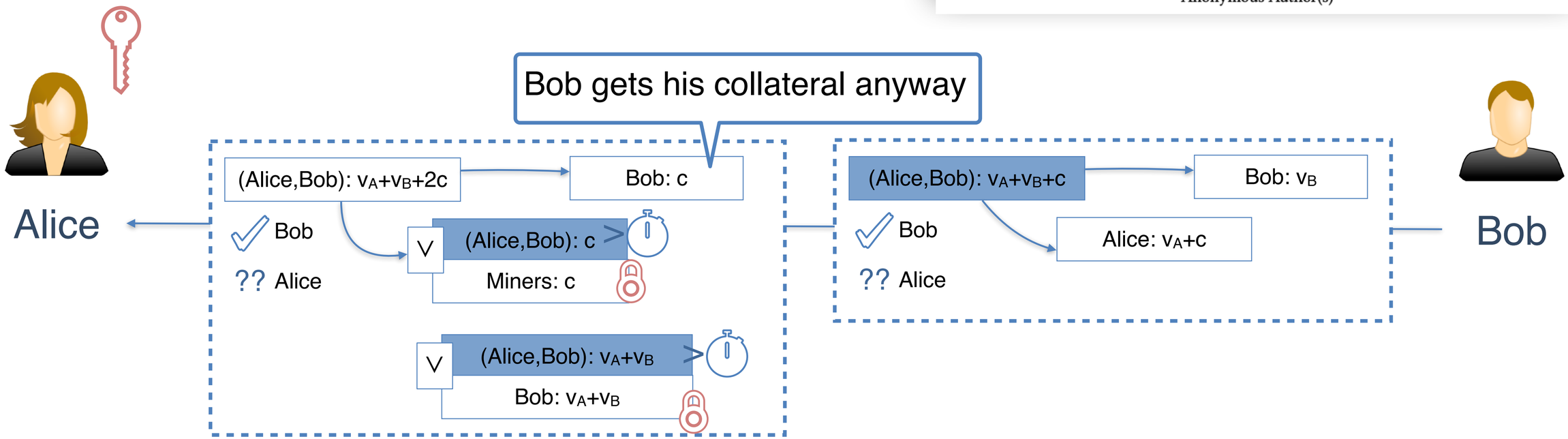
- First game-theoretically secure payment channel construction against byzantine adversaries and rational miners
- Supports offline users without requiring watchtowers nor limited channel lifetime

CRAB (Channel Resistant Against Bribery)

ACM CCS 2024

Securing Lightning Channels against Rational Miners

Anonymous Author(s)*



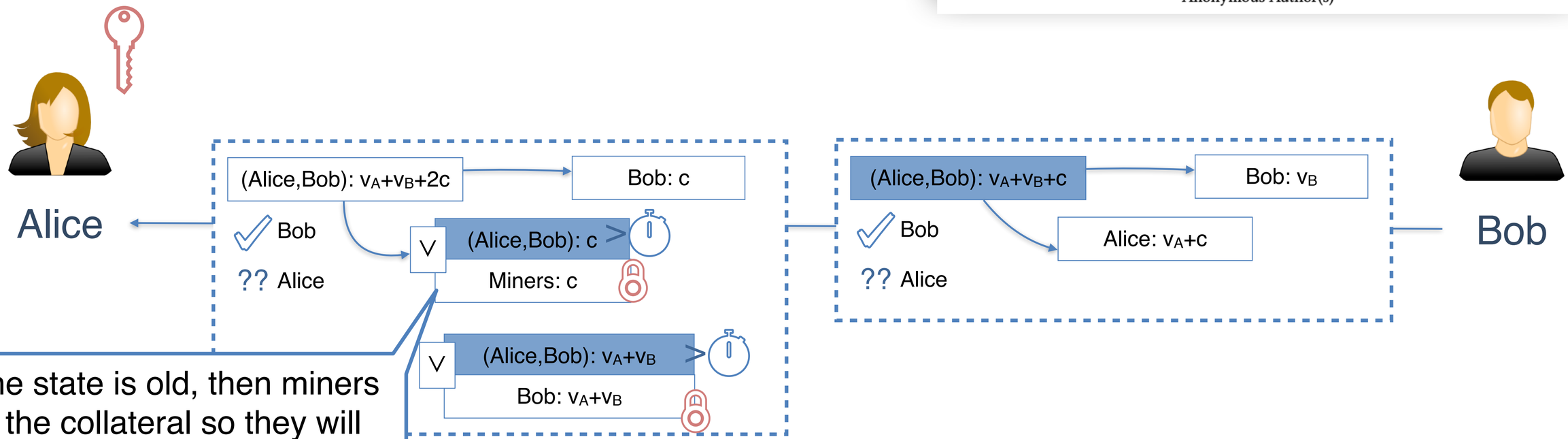
- First game-theoretically secure payment channel construction against byzantine adversaries and rational miners
- Supports offline users without requiring watchtowers nor limited channel lifetime

CRAB (Channel Resistant Against Bribery)

ACM CCS 2024

Securing Lightning Channels against Rational Miners

Anonymous Author(s)*



If the state is old, then miners get the collateral so they will post the transaction on-chain, otherwise the collateral goes to the next transaction

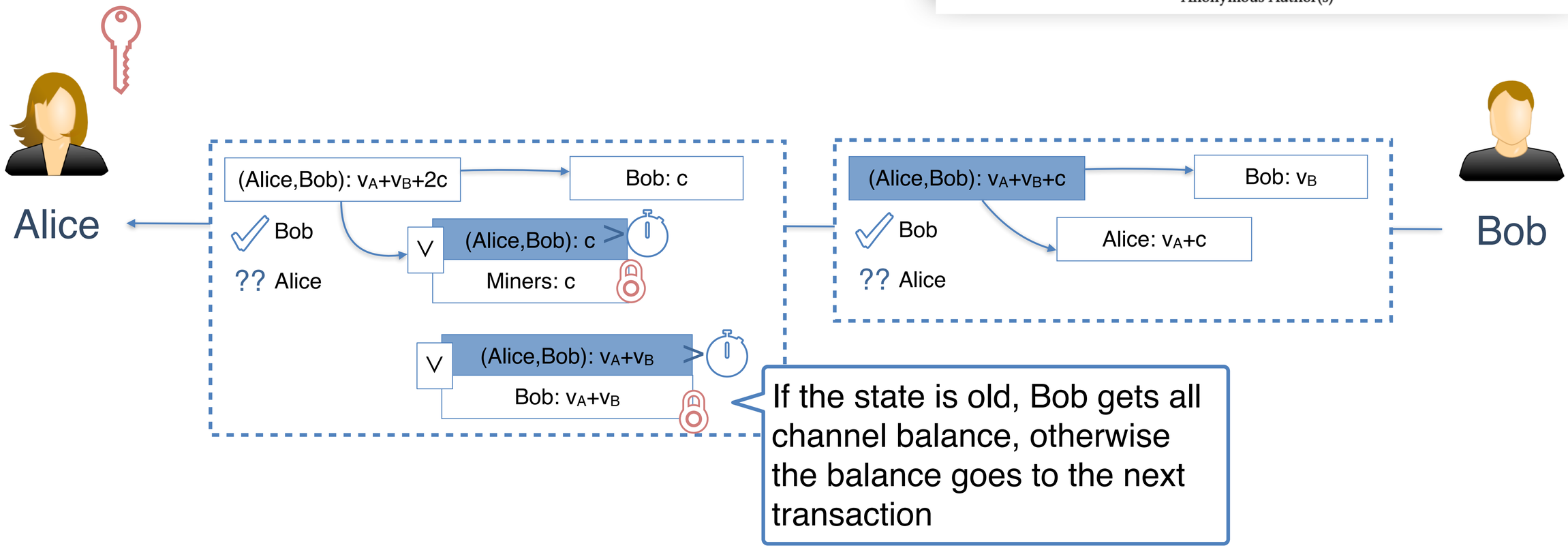
- First game-theoretically secure payment channel construction against byzantine adversaries and rational miners
- Supports offline users without requiring watchtowers nor limited channel lifetime

CRAB (Channel Resistant Against Bribery)

ACM CCS 2024

Securing Lightning Channels against Rational Miners

Anonymous Author(s)*



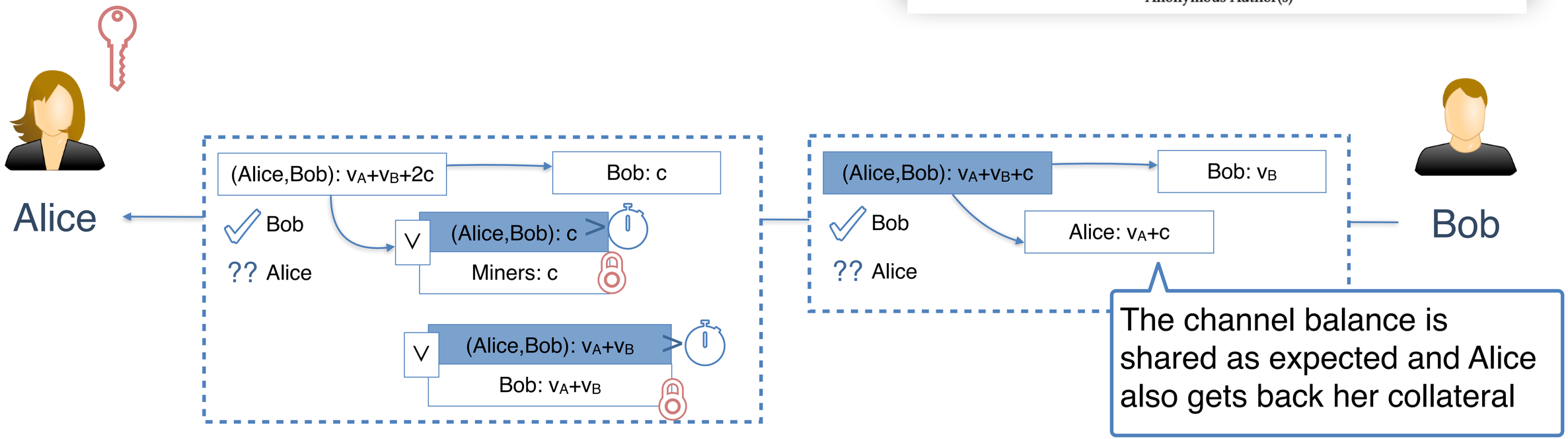
- First game-theoretically secure payment channel construction against byzantine adversaries and rational miners
- Supports offline users without requiring watchtowers nor limited channel lifetime

CRAB (Channel Resistant Against Bribery)

ACM CCS 2024

Securing Lightning Channels against Rational Miners

Anonymous Author(s)*



- First game-theoretically secure payment channel construction against byzantine adversaries and rational miners
- Supports offline users without requiring watchtowers nor limited channel lifetime

Research Questions

Research Questions for PL Folks

- Characterize the class of functions expressable in Bitcoin scripting
- Characterize the gains in expressiveness that opcodes currently discussed would offer (e.g., different forms of covenance)
- Provide semantic foundations, verification tools, etc.

ACM CCS 2018

BitML: A Calculus for Bitcoin Smart Contracts

Massimo Bartoletti
University of Cagliari
bart@unica.it

Roberto Zunino
University of Trento
roberto.zunino@unitn.it

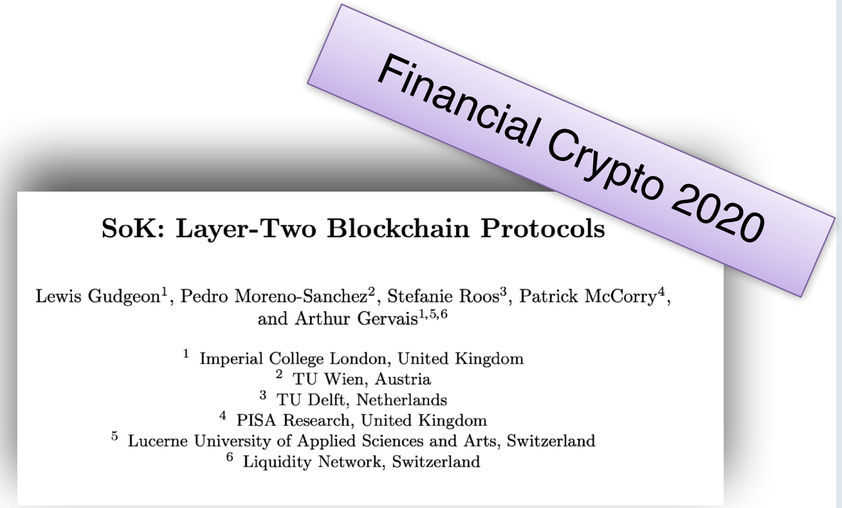
BitVM: Compute Anything on Bitcoin

Robin Linus
robin@zerosync.org

December 12, 2023

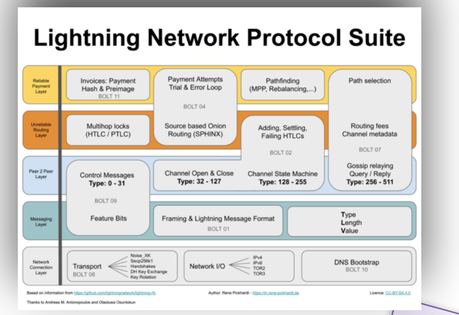
Research Questions for Distributed and Crypto Folks

- Which properties would we like to achieve via Layer-2 protocols?
  - Privacy, scalability, accountability, what more?
- Which classes of protocols can we design to achieve them?
  - Payment channel networks, rollups, what else?



Research Questions for Network Folks

- Lightning Network assumes a public topology to compute the route to the receiver (scalability and privacy issues)
- How can we route messages over a private topology?
- Can we characterize the privacy properties (e.g., like we do in Tor)?
- How can we make routing more efficient and resiliant?



NDSS 2018

Settling Payments Fast and Private: Efficient Decentralized Routing for Path-Based Transactions

Stefanie Roos
University of Waterloo
sroos@uwaterloo.ca

Pedro Moreno-Sanchez
Purdue University
pmorenos@purdue.edu

Aniket Kate
Purdue University
aniket@purdue.edu

Ian Goldberg
University of Waterloo
iang@cs.uwaterloo.ca

NSDI 2020

High Throughput Cryptocurrency Ro
Payment Channel Networks

Vibhaalakshmi Sivaraman[1], Shaileshh Bojja Venkatakrishnan[2], Kathleen Ruan[3], Parimarjan Negi[1], Lei Yang[1], Radhika Mittal[4], Mohammad Alizadeh[1], and Giulia Fanti[3]

[1]Massachusetts Institute of Technology
[2]Ohio State University
[3]Carnegie Mellon University
[3]University of Illinois at Urbana-Cham

IFIP Networking 2021

LightPIR: Privacy-Preserving Route Discovery for Payment Channel Networks

Krzysztof Pietrzak[‡*]
krzysztof.pietrzak@ist.ac.at

Iosif Salem[§]
iosif.salem@univie.ac.at

Stefan Schmid[§†]
stefan_schmid@univie.ac.at

Michelle Yeo[‡]
michelle.yeo@ist.ac.at

[‡]IST Austria
[§]Faculty of Computer Science, University of Vienna

Research Questions for ML and Measurement Folks

- How can we leverage the on-chain footprint to
  - Break user anonymity, both on-chain (Layer-1) and off-chain (Layer-2)?
  - Track payments and identify cybercrime activities?
  - Quantify the guarantees offered by privacy-preserving protocols?
  - Understand and optimize Miner Extractable Value algorithms?

Research Questions for Game-Theory Folks

- Design Layer-2 protocols that are game-theoretic secure against rational miners
- Game-theoretically secure the composition of
  - Layer-1 and Layer-2
  - Layer-2 applications

NDSS 2023


He-HTLC: Revisiting Incentives in HTLC

Sarisht Wadhwa[§] *Duke University* sarisht.wadhwa@duke.edu
Jannis Stöter[§] *Duke University* jannis.stoeter@alumni.duke.edu
Fan Zhang *Duke University* fan.zhang@duke.edu
Kartik Nayak *Duke University* kartik@cs.duke.edu

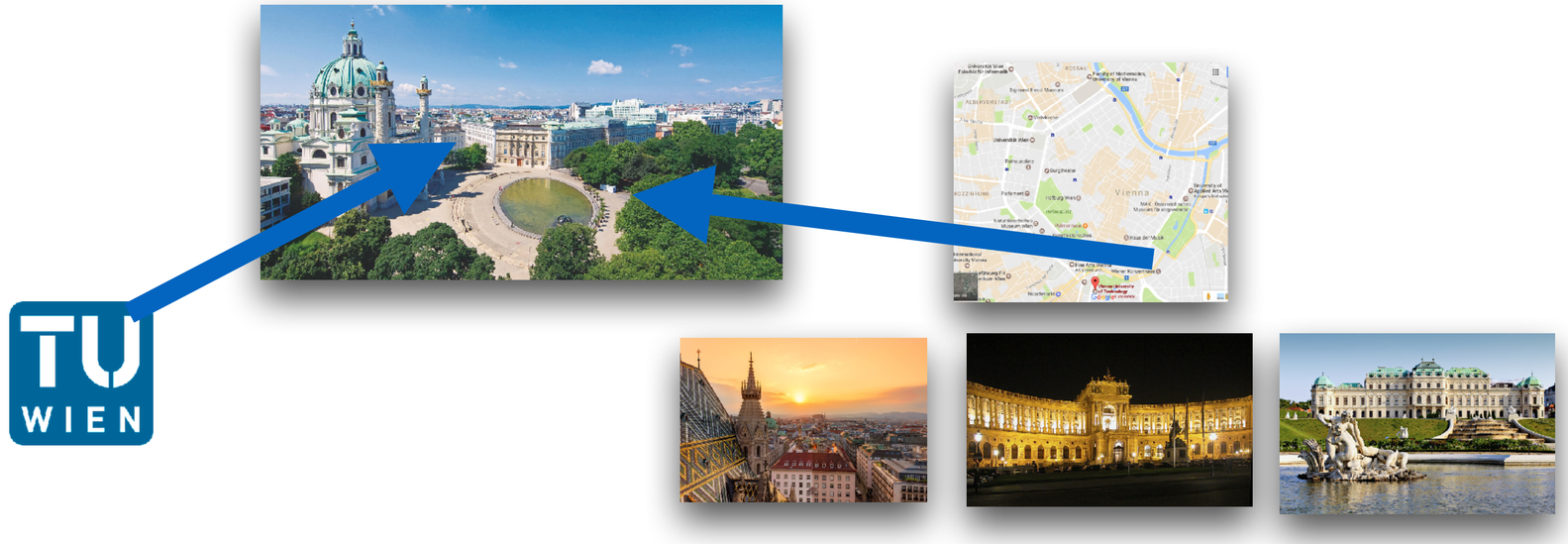
IEEE CSF 2023


Towards a Game-Theoretic Security Analysis of Off-Chain Protocols

Sophie Rain *TU Wien*, Austria
Georgia Avarikioti *TU Wien*, Austria
Laura Kovács *TU Wien*, Austria
Matteo Maffei *Christian Doppler Lab Blockchain Technologies for the Internet of Things* *TU Wien*, Austria

Interested in an internship, PhD, PostDoc, research visit, talk?

ERC Advanced Grant
BlockSec
Formal Methods for Secure Blockchain-Oriented Programming
2024-2029



Take Home

Scaling blockchains and making them more secure and privacy-preserving is a grand challenge that requires groundbreaking, interdisciplinary research

(PL, game theory, networks, ML, cryptography, distributed systems…)