TECHNISCHE UNIVERSITÄT WIEN
Vienna | Austria

FAKULTÄT FÜR !NFORMATIK
Faculty of Informatics

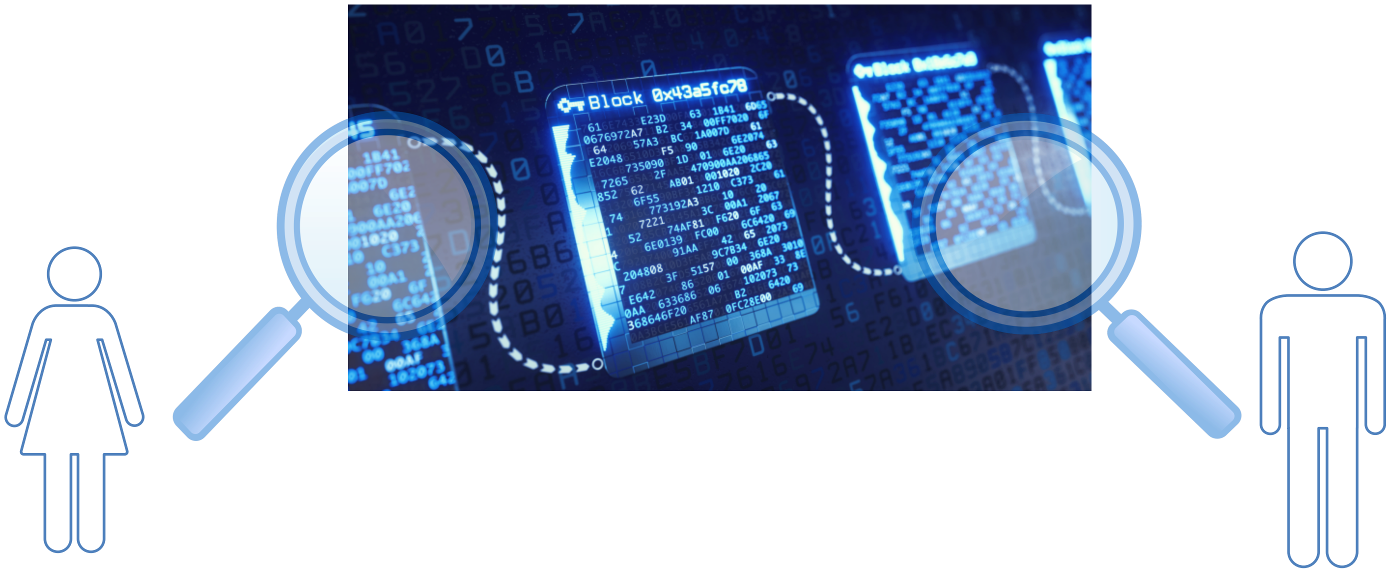
SECURITY & PRIVACY GROUP

Security and Privacy for Payment Channel Networks

Matteo Maffei

DLT 2020 - Ancona 5/2/2020

Scalability Problem



- Decentralized data structure recording each transaction in order to provide public verifiability
- Global consensus: everyone checks the whole blockchain

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

Scalability Solutions

- On-chain, consensus layer (tweak consensus)
  e.g., DAG Blockchain, sharding, ...
- Off-chain, application layer (local consensus, blockchain used only in case of disputes)
  - Payment Channel Networks

    LND Lightning Network (Bitcoin)

    RAIDEN Raiden Network (Ethereum)
  - Many other research projects (Bolt, Z-Channels, Tumblebit, Perun, ...)

Scalability Solutions

- On-chain, consensus layer (tweak consensus)
  e.g., DAG Blockchain, sharding, ...
- Off-chain, application layer (local consensus, blockchain used only in case of disputes)
  - Payment Channel Networks

    Lightning Network (Bitcoin)

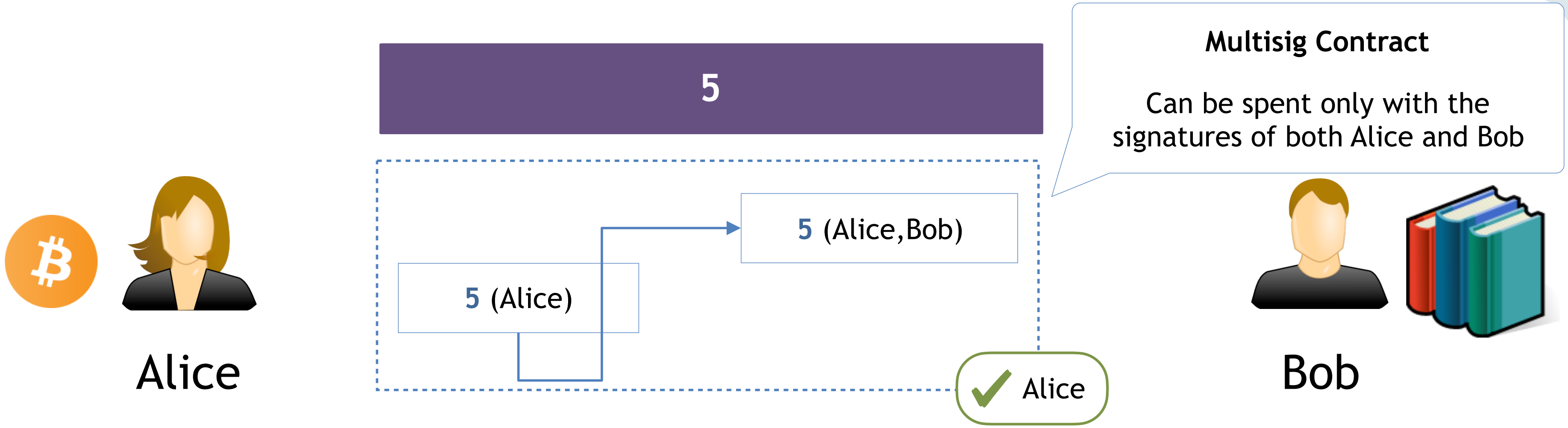
    Raiden Network (Ethereum)
  - Many other research projects (Bolt, Z-Channels, Tumblebit, Perun, ...)

Background on
Payment Channel Networks

Payment Channels: Open

Blockchain

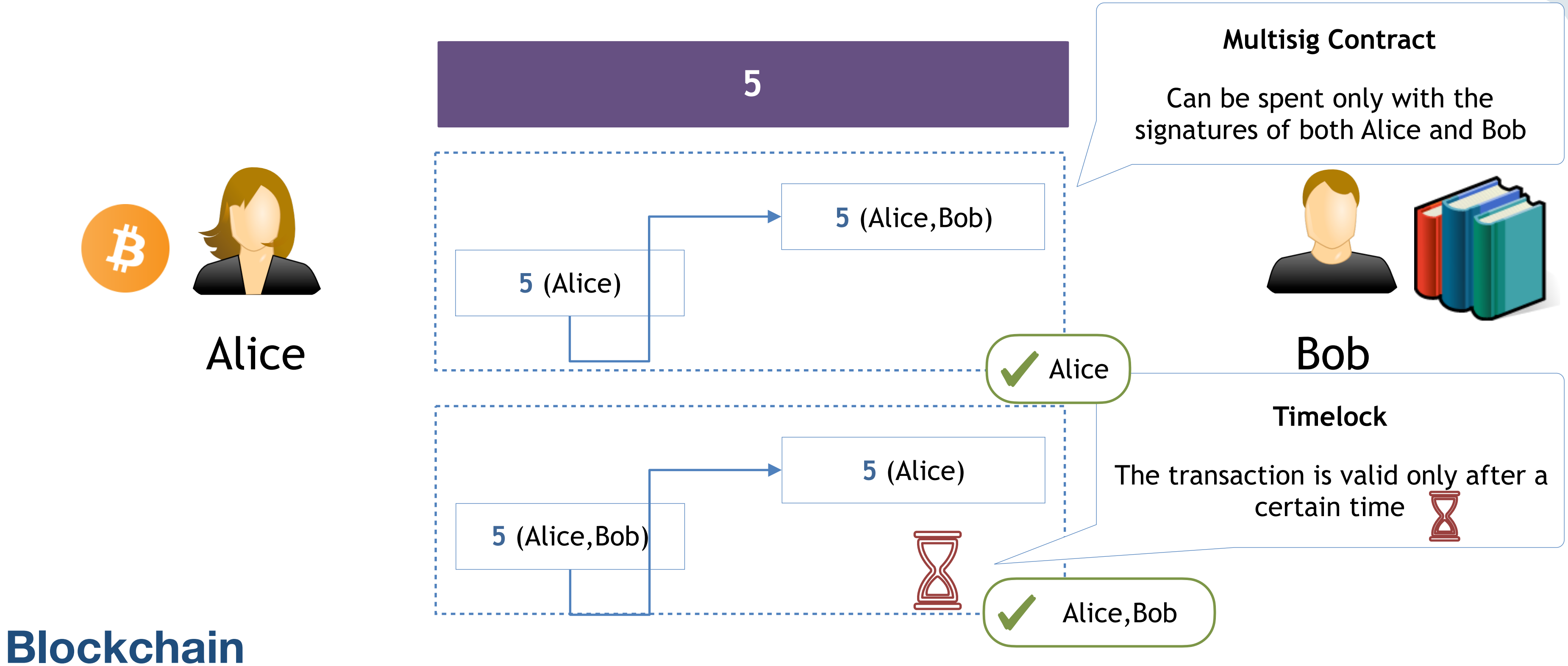
Payment Channels: Open



Blockchain

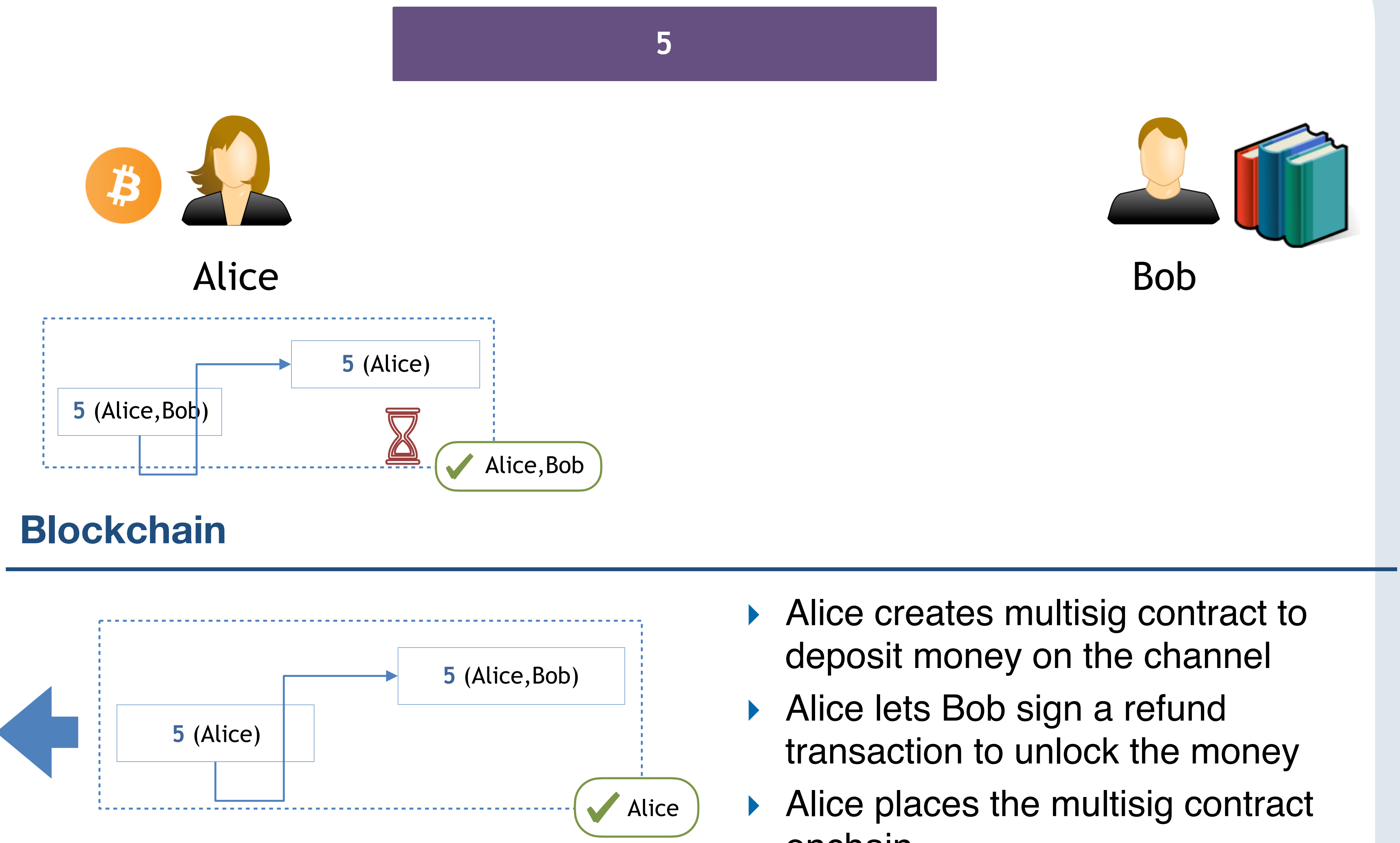
- Alice creates multisig contract to deposit money on the channel

Payment Channels: Open



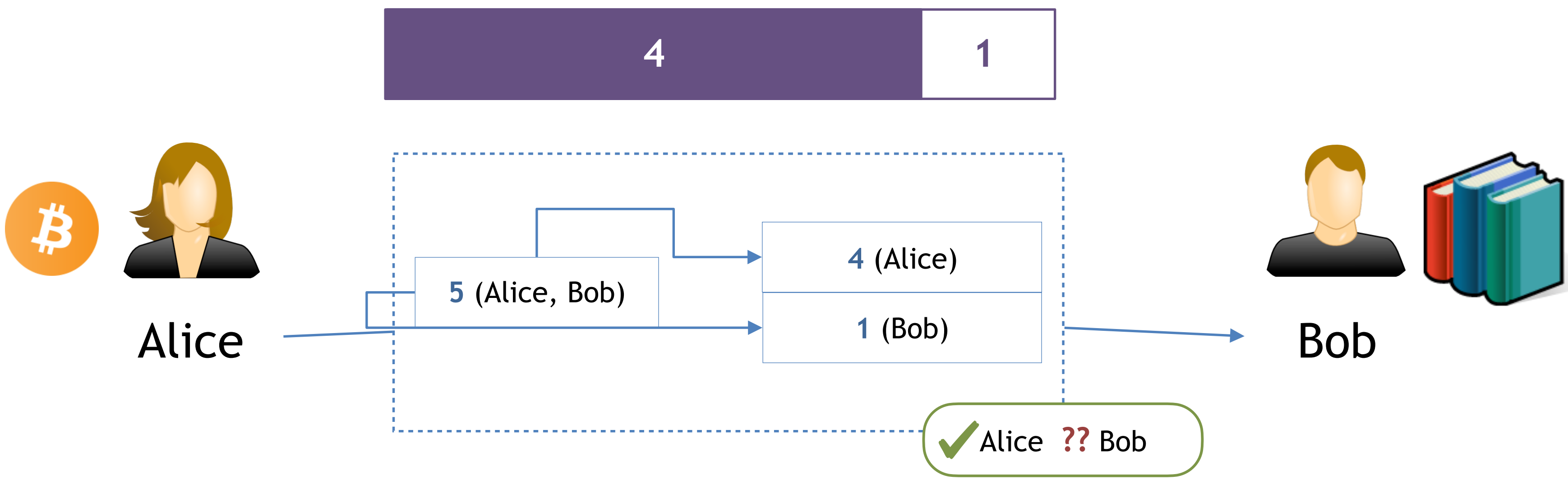
- Alice creates multisig contract to deposit money on the channel
- Alice lets Bob sign a refund transaction to unlock the money

Payment Channels: Open

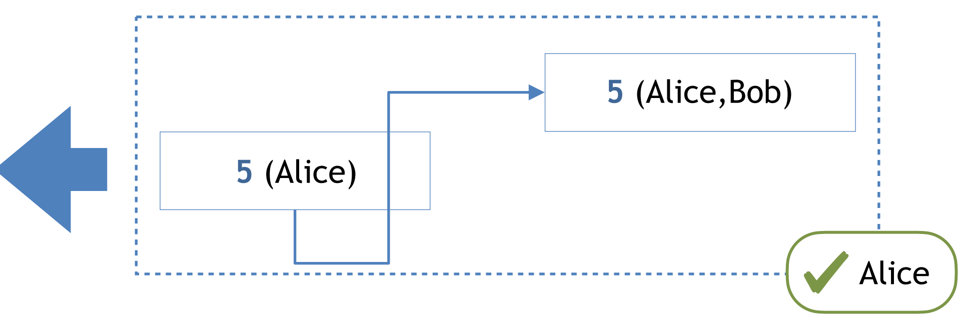


- Alice creates multisig contract to deposit money on the channel
- Alice lets Bob sign a refund transaction to unlock the money
- Alice places the multisig contract onchain

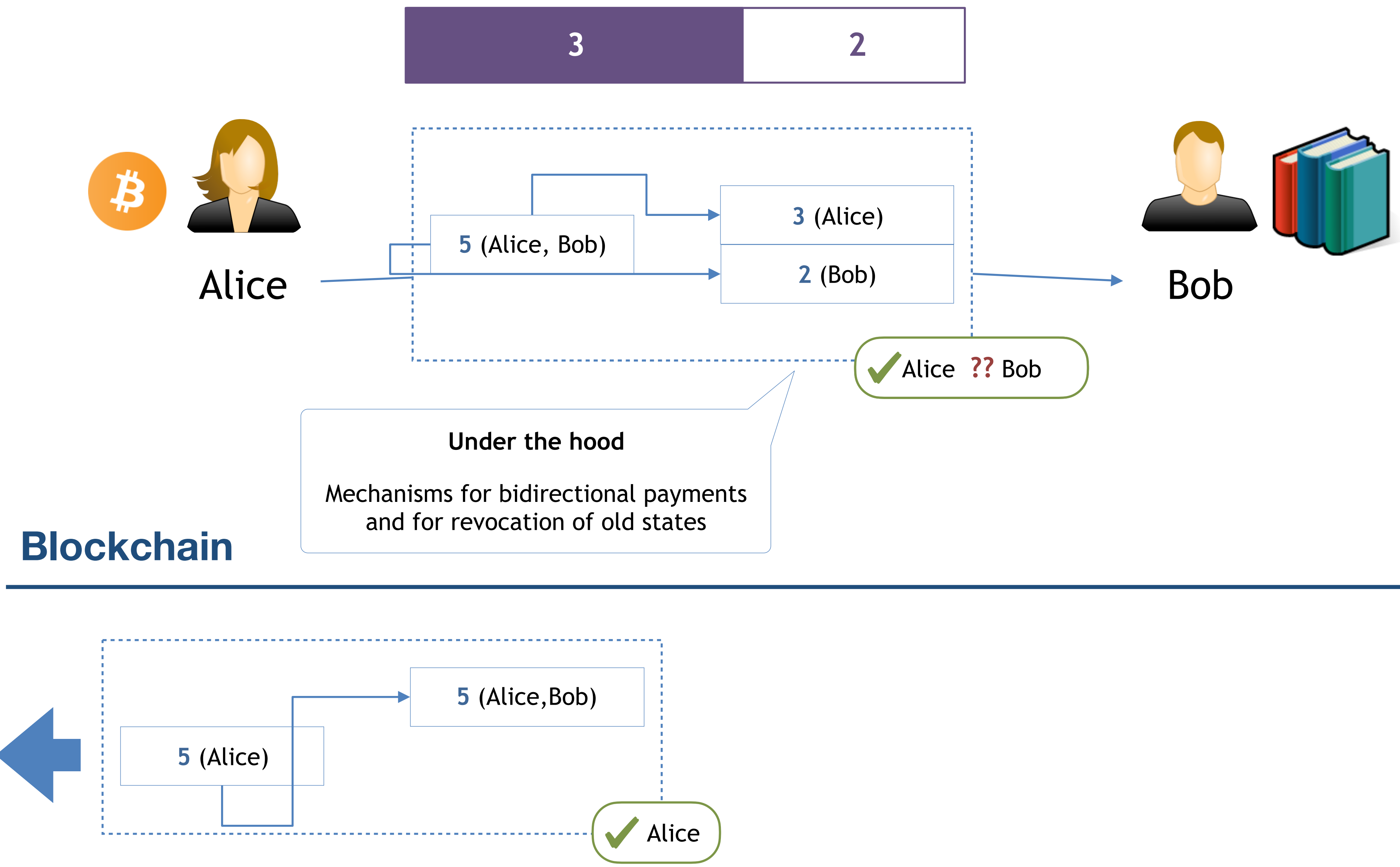
Payment Channels: Transactions



Blockchain

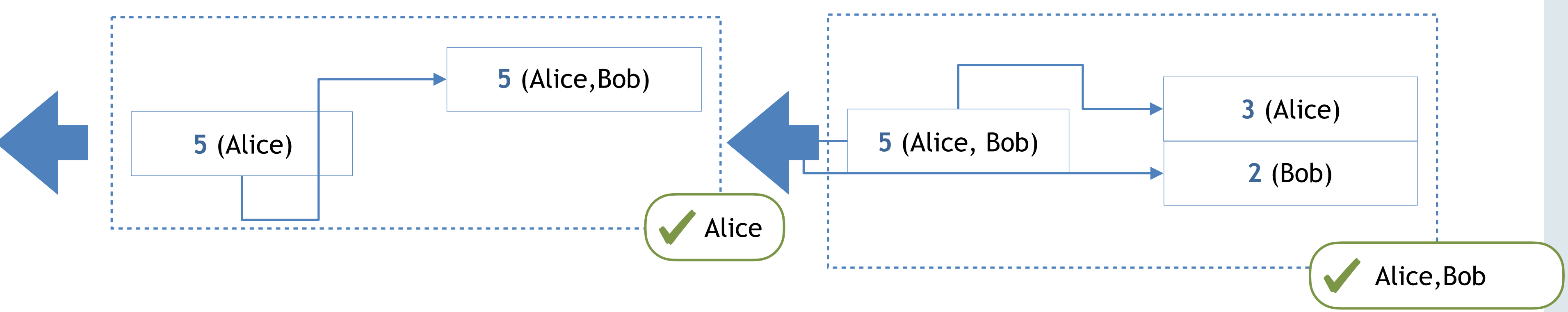


Payment Channels: Transactions

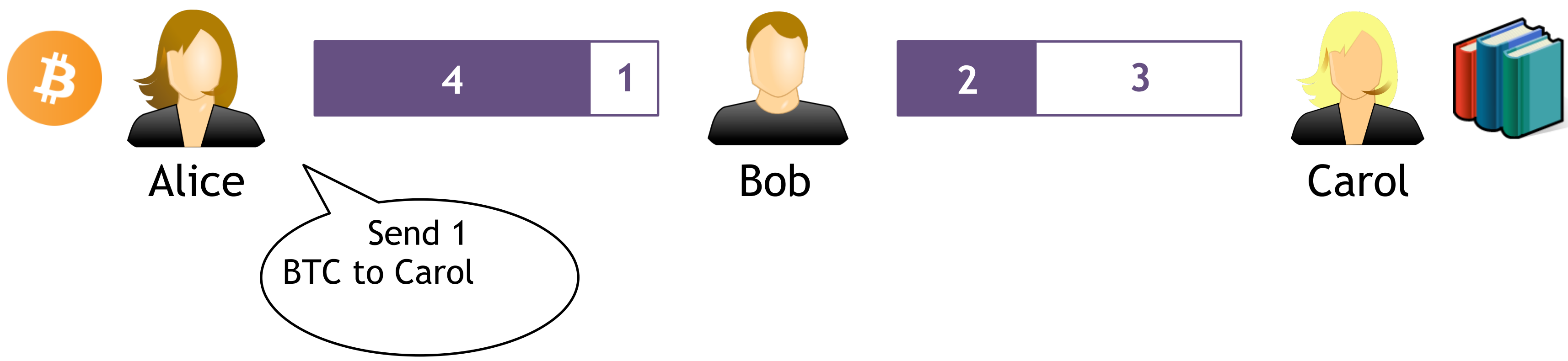


Payment Channels: Close

Blockchain

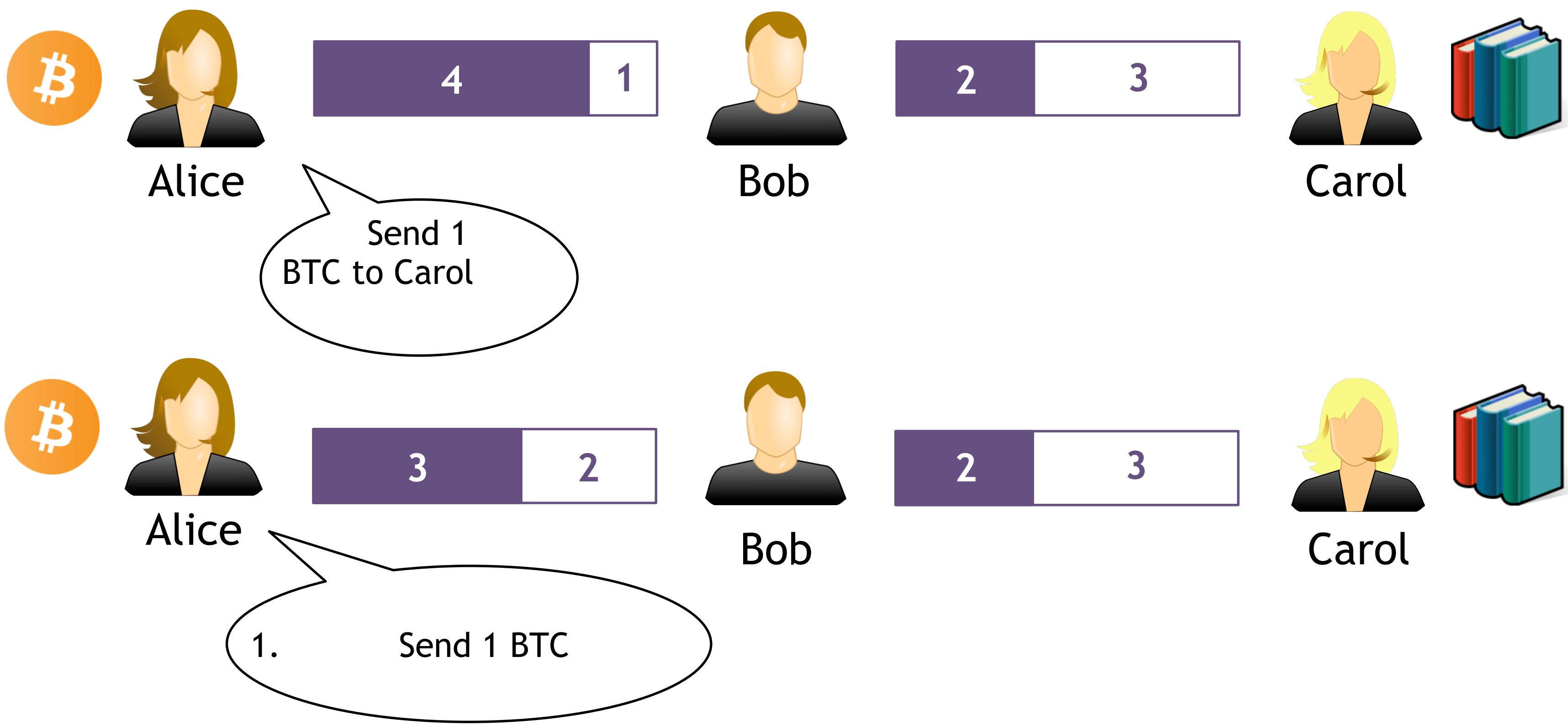


Payment Channel Networks (PCNs)

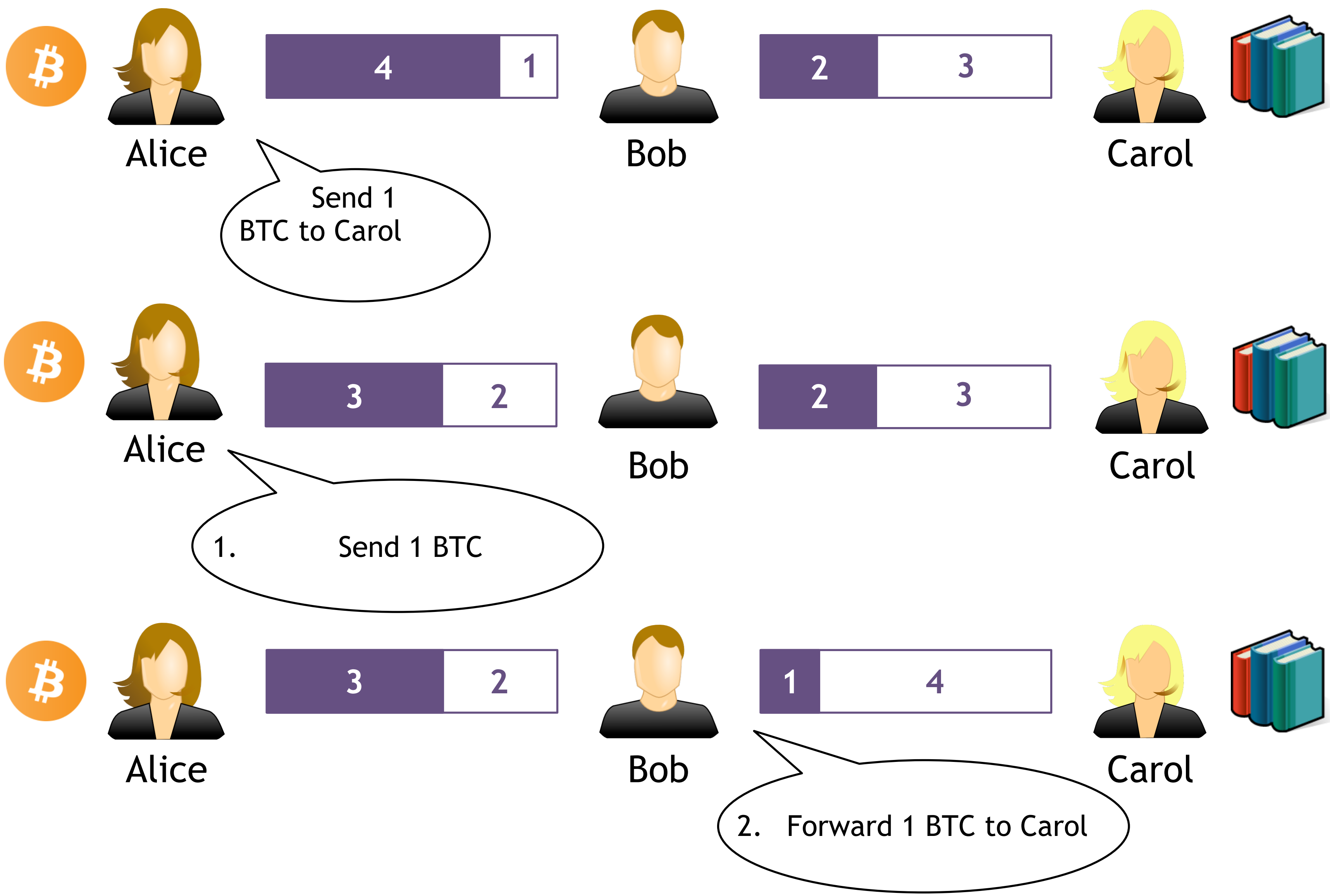


One cannot open channels with everyone...
⇒ exploit channel paths!

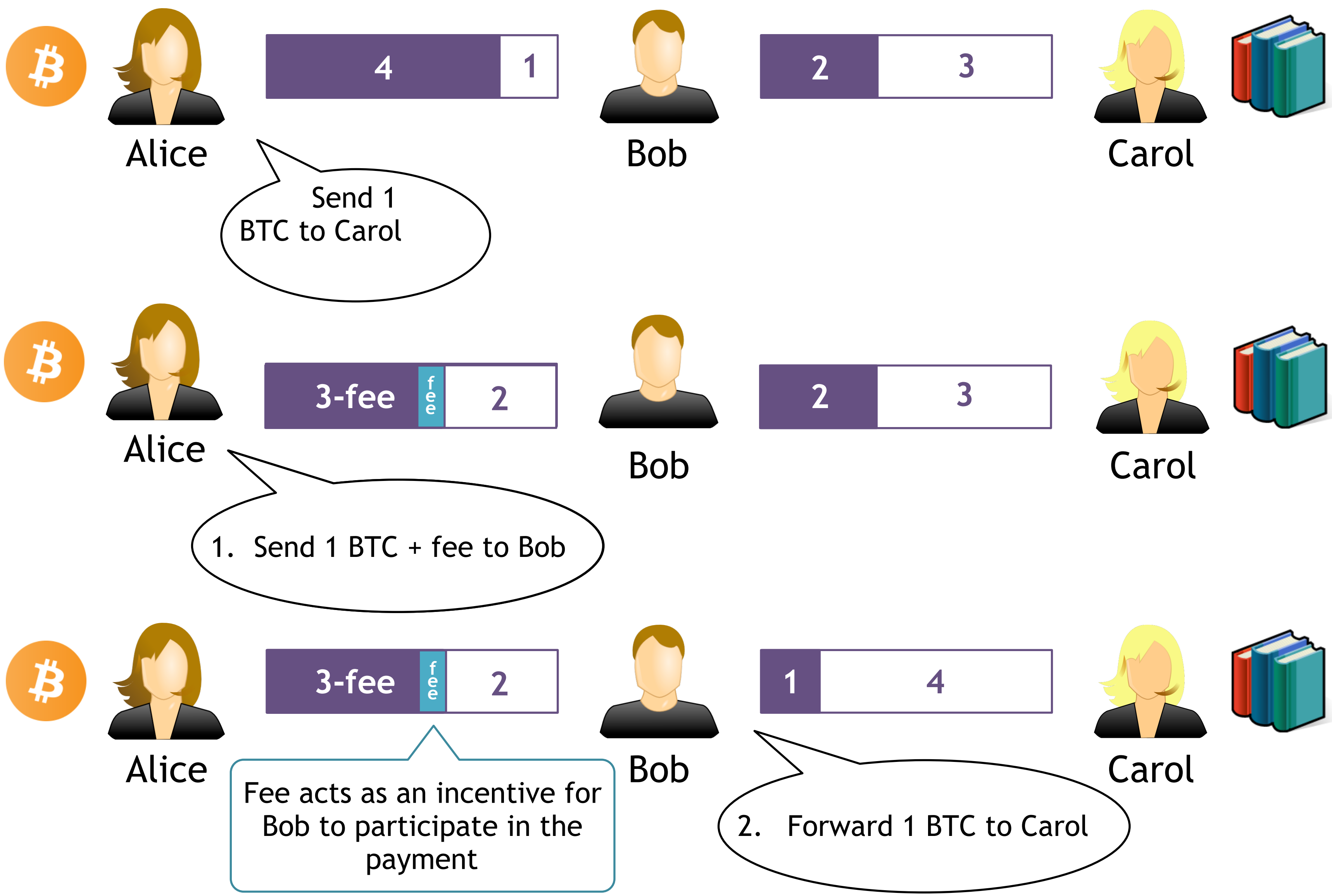
Payment Channel Networks (PCNs)



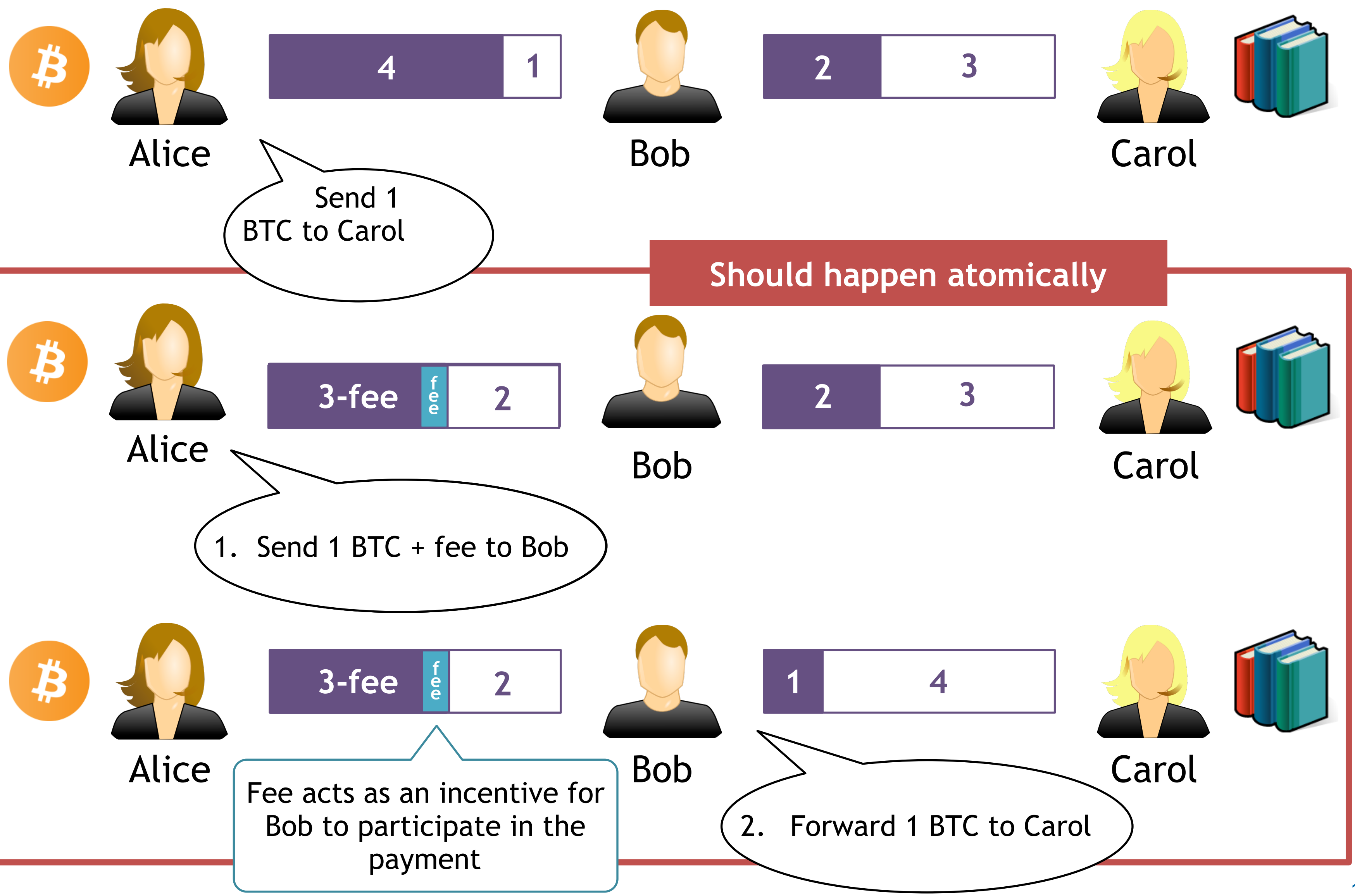
Payment Channel Networks (PCNs)



Payment Channel Networks (PCNs)

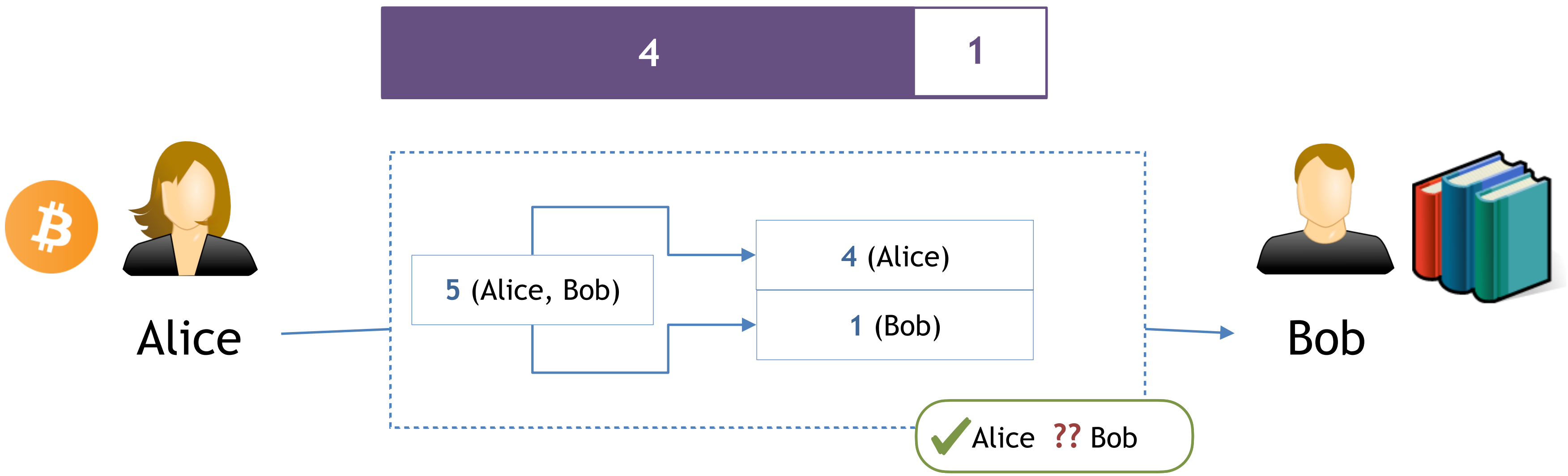


Payment Channel Networks (PCNs)

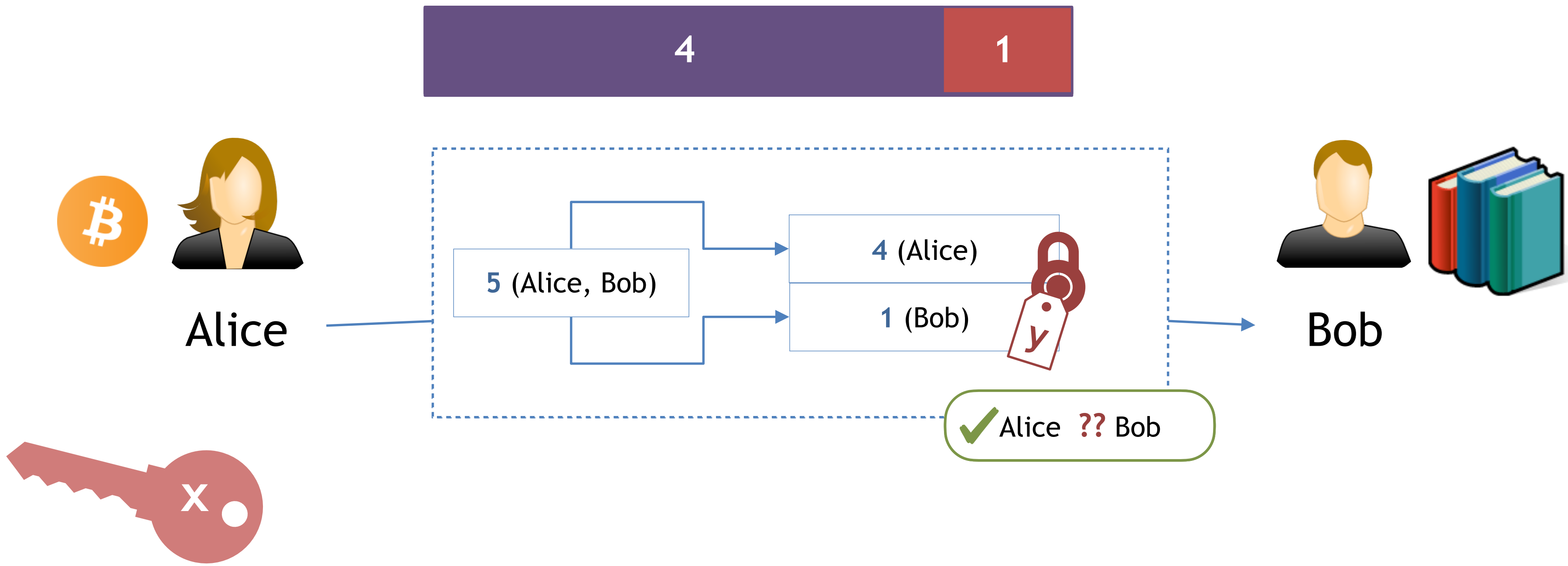


The Lightning Network (LN)

Hashtime Lock Contract (HTLC)

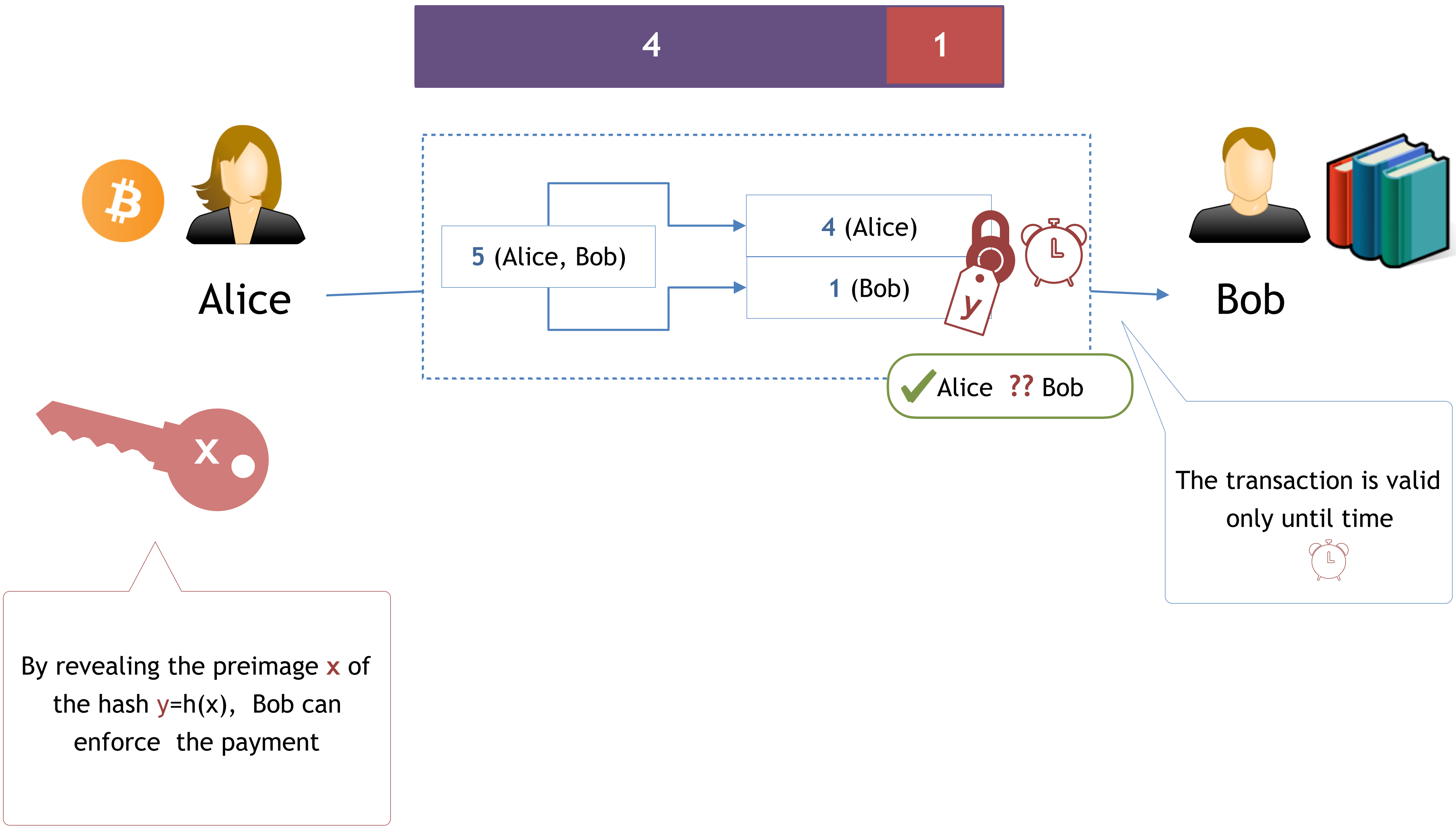


Hashtime Lock Contract (HTLC)

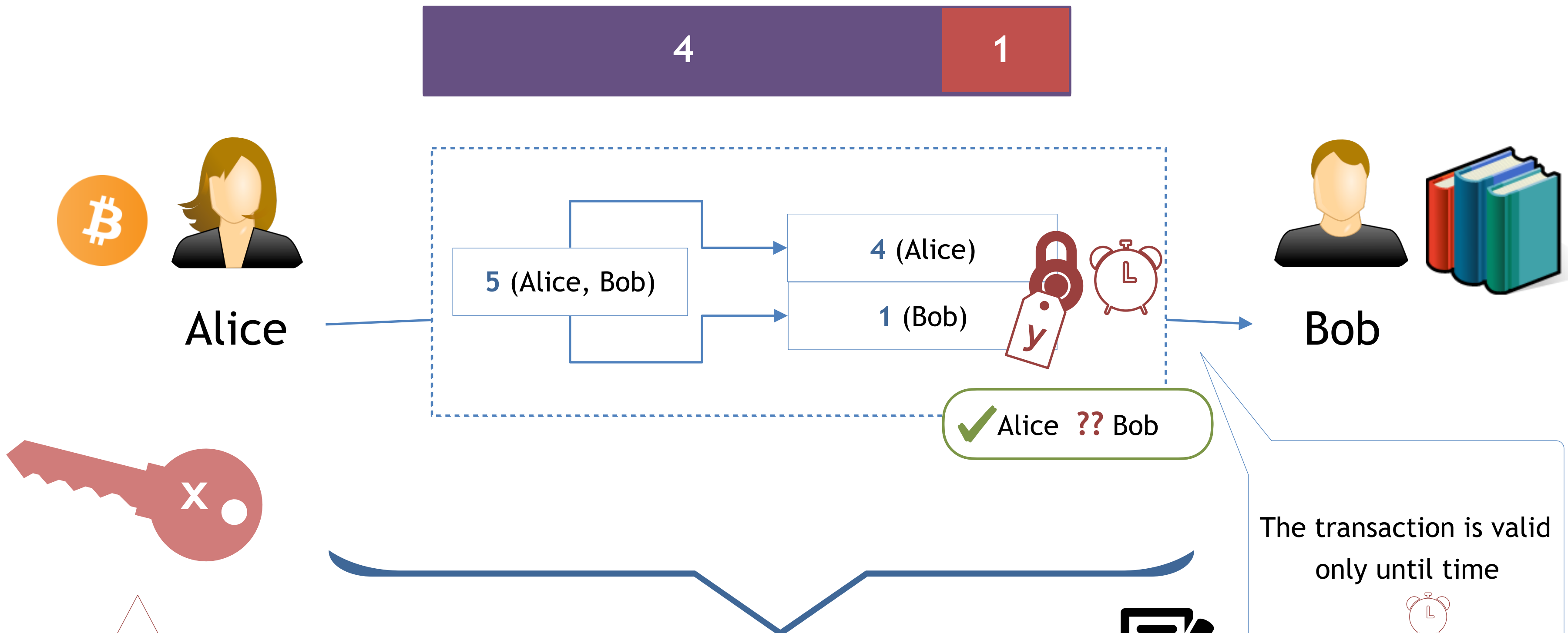


By revealing the preimage x of the hash y=h(x), Bob can enforce the payment

Hashtime Lock Contract (HTLC)



Hashtime Lock Contract (HTLC)

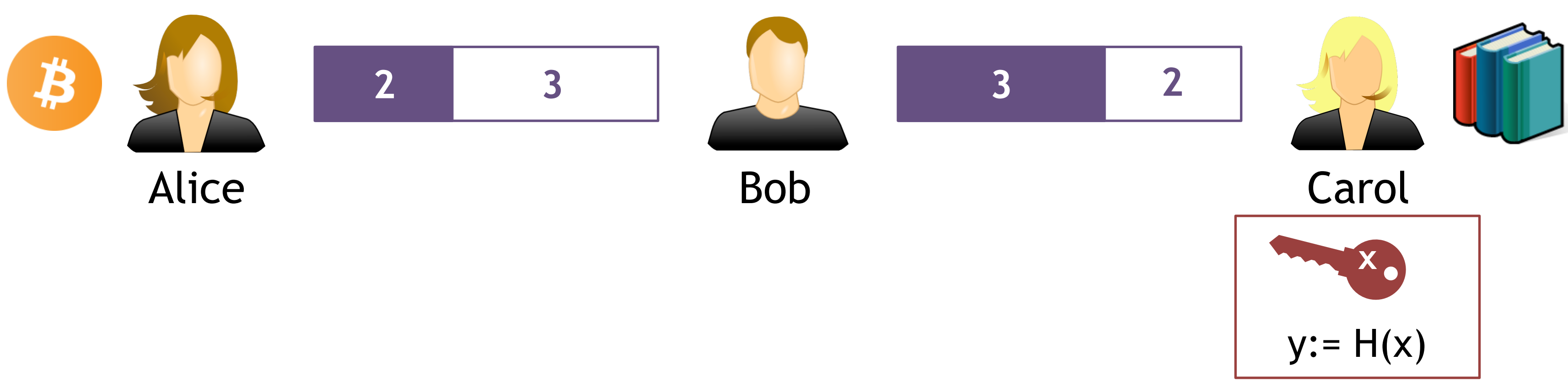


The transaction is valid only until time

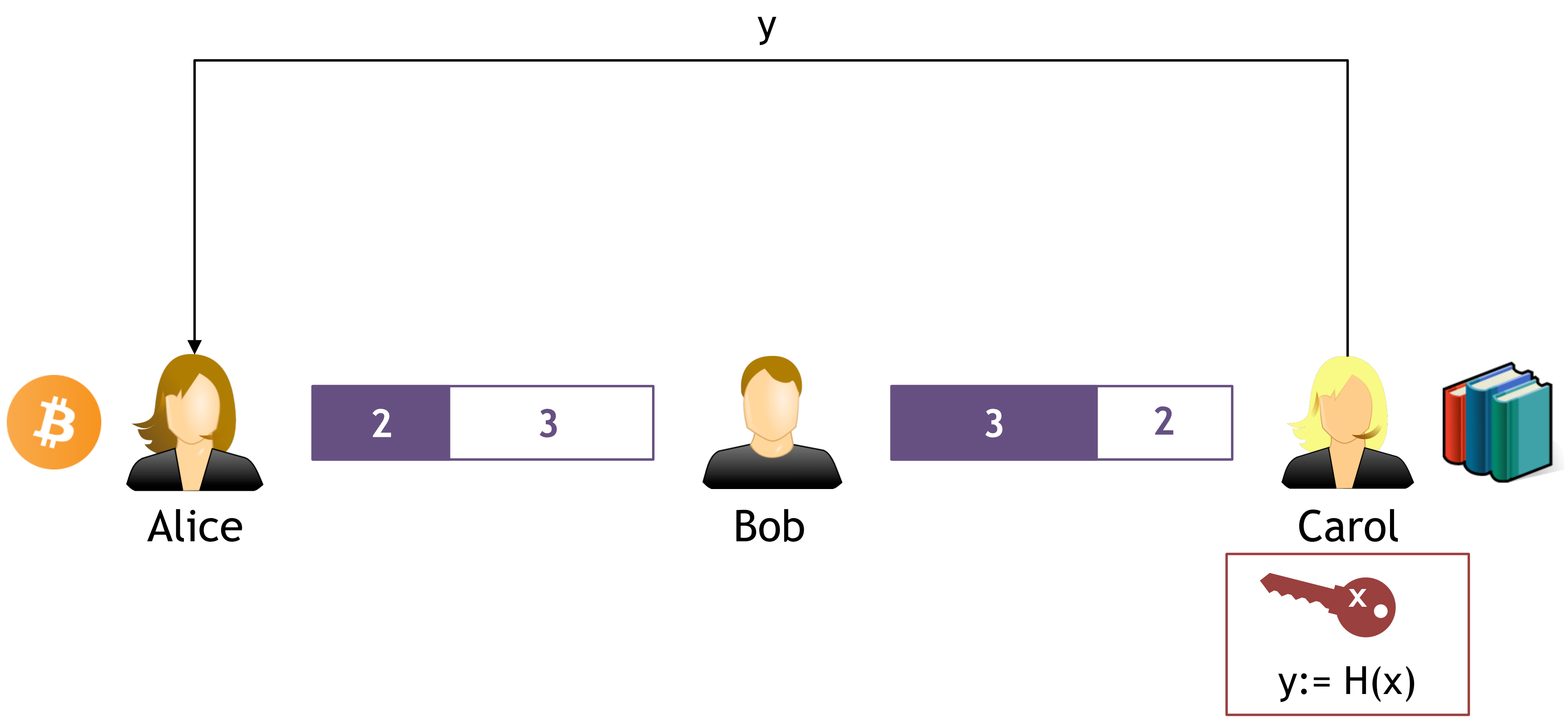
By revealing the preimage x of the hash y=h(x), Bob can enforce the payment

HTLC (Alice, Bob, 1, y,):
Alice pays Bob 1 BTC iff Bob shows some x such that H(x) = y before

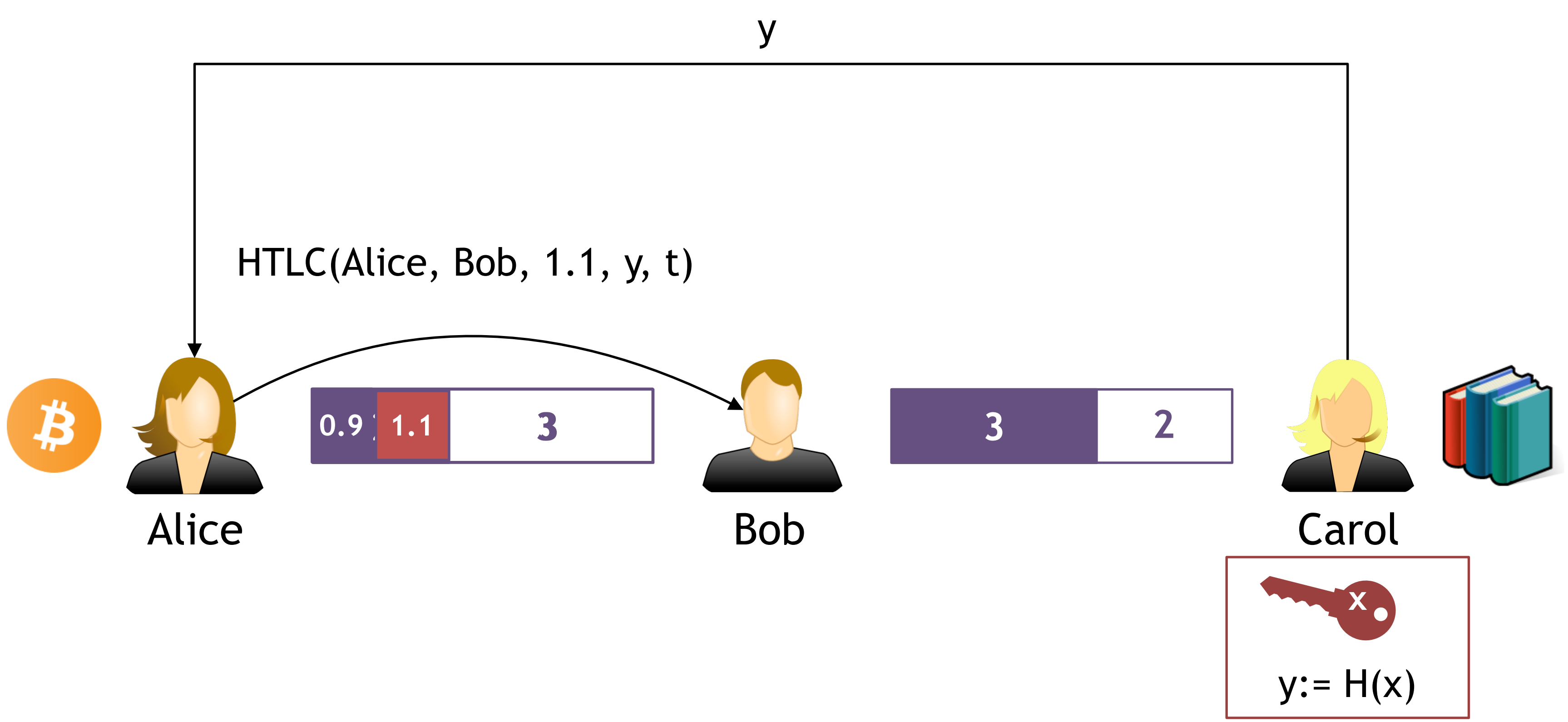
HTLC for Multi-hop Payments



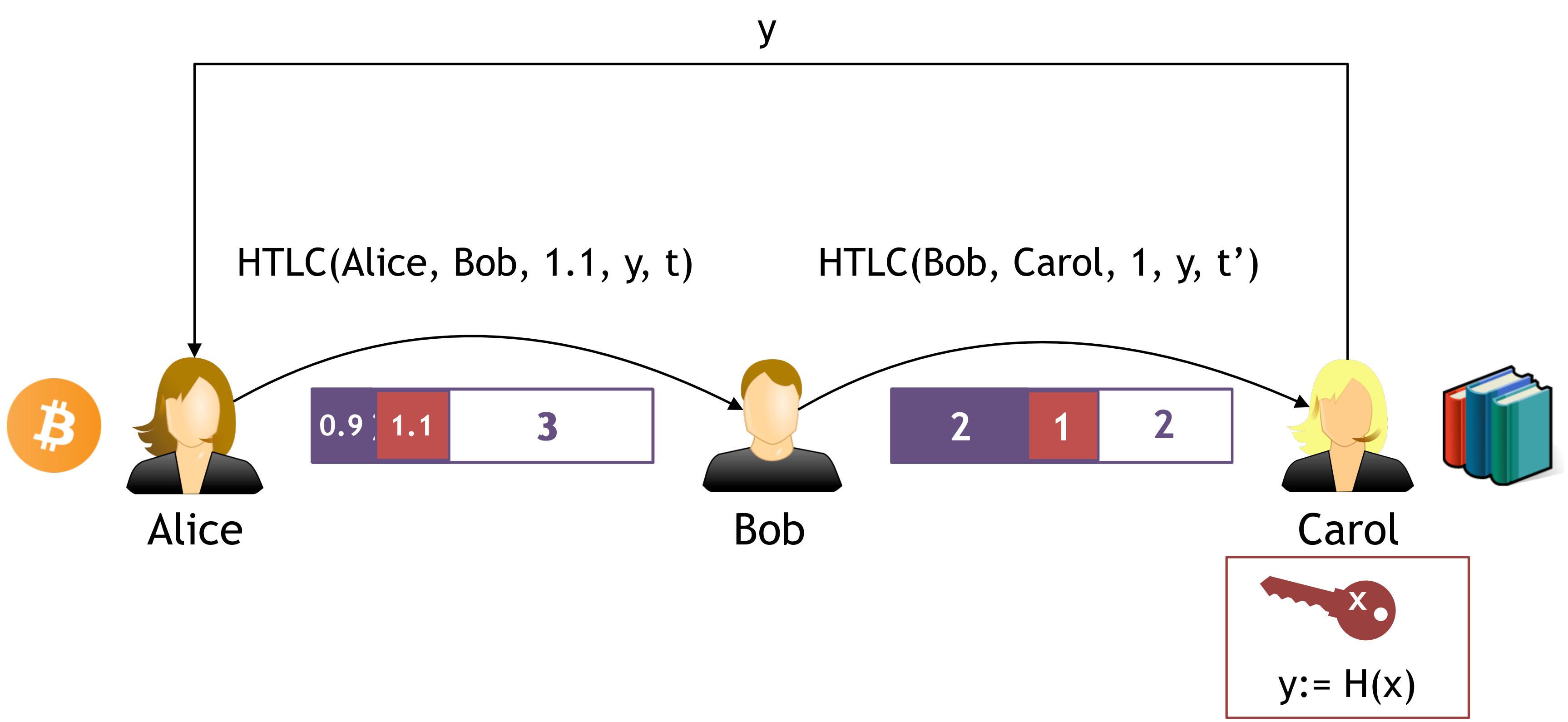
HTLC for Multi-hop Payments



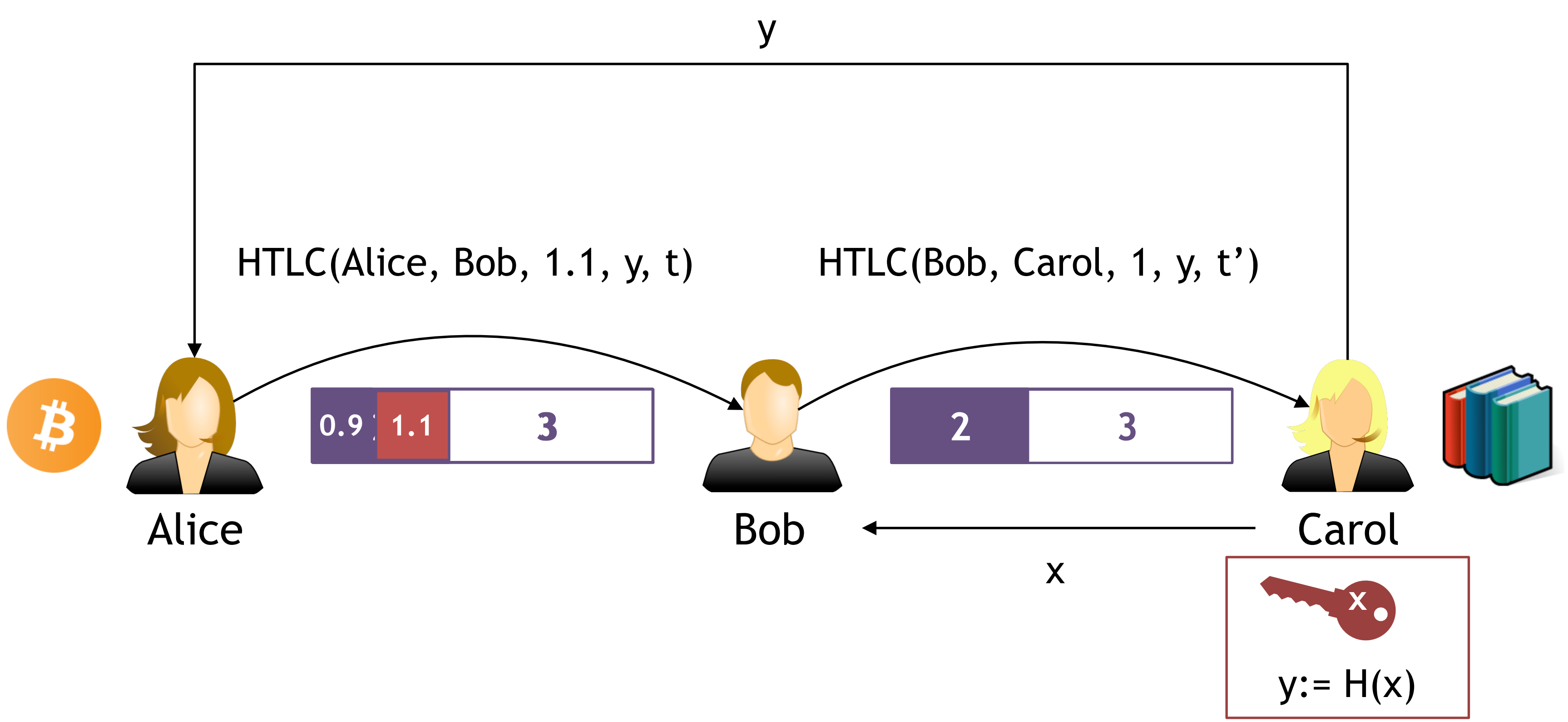
HTLC for Multi-hop Payments



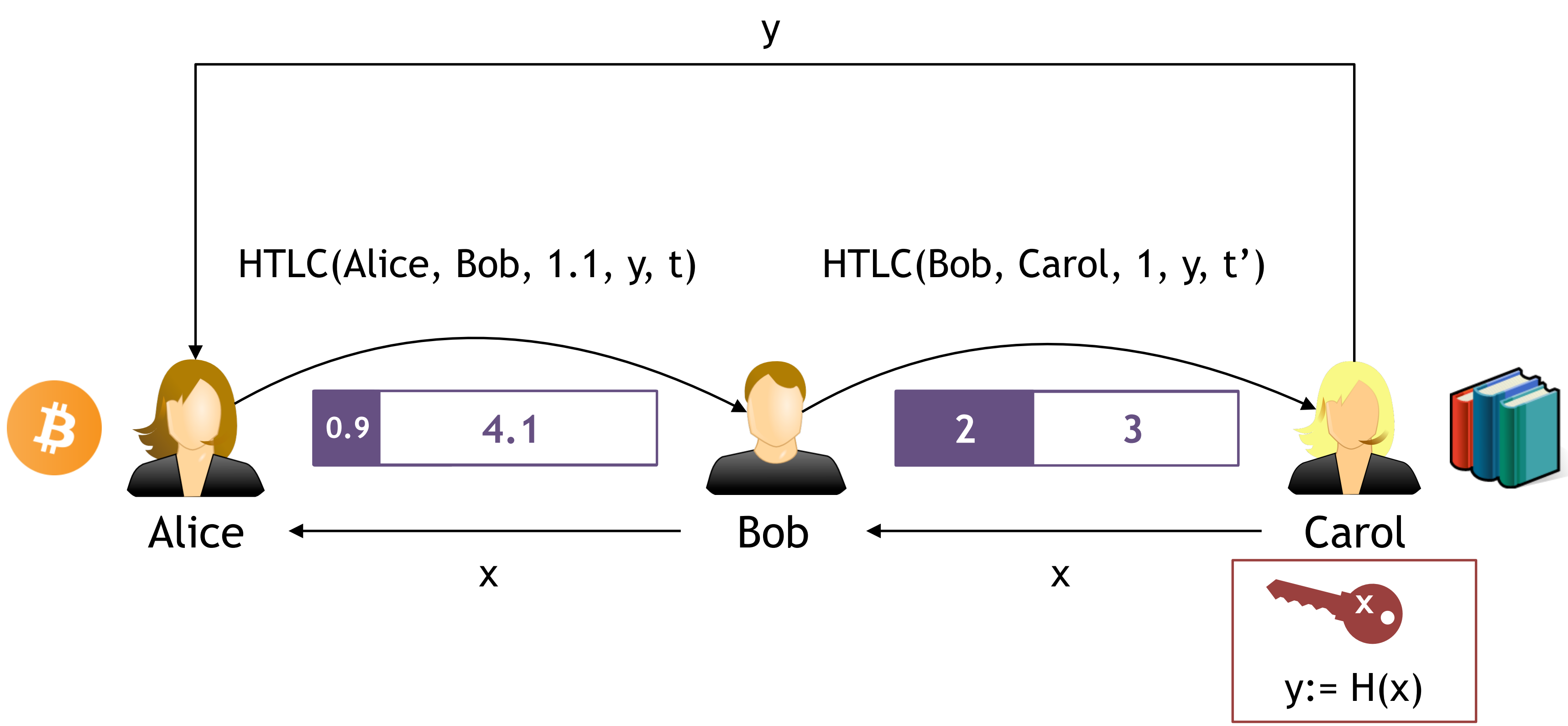
HTLC for Multi-hop Payments



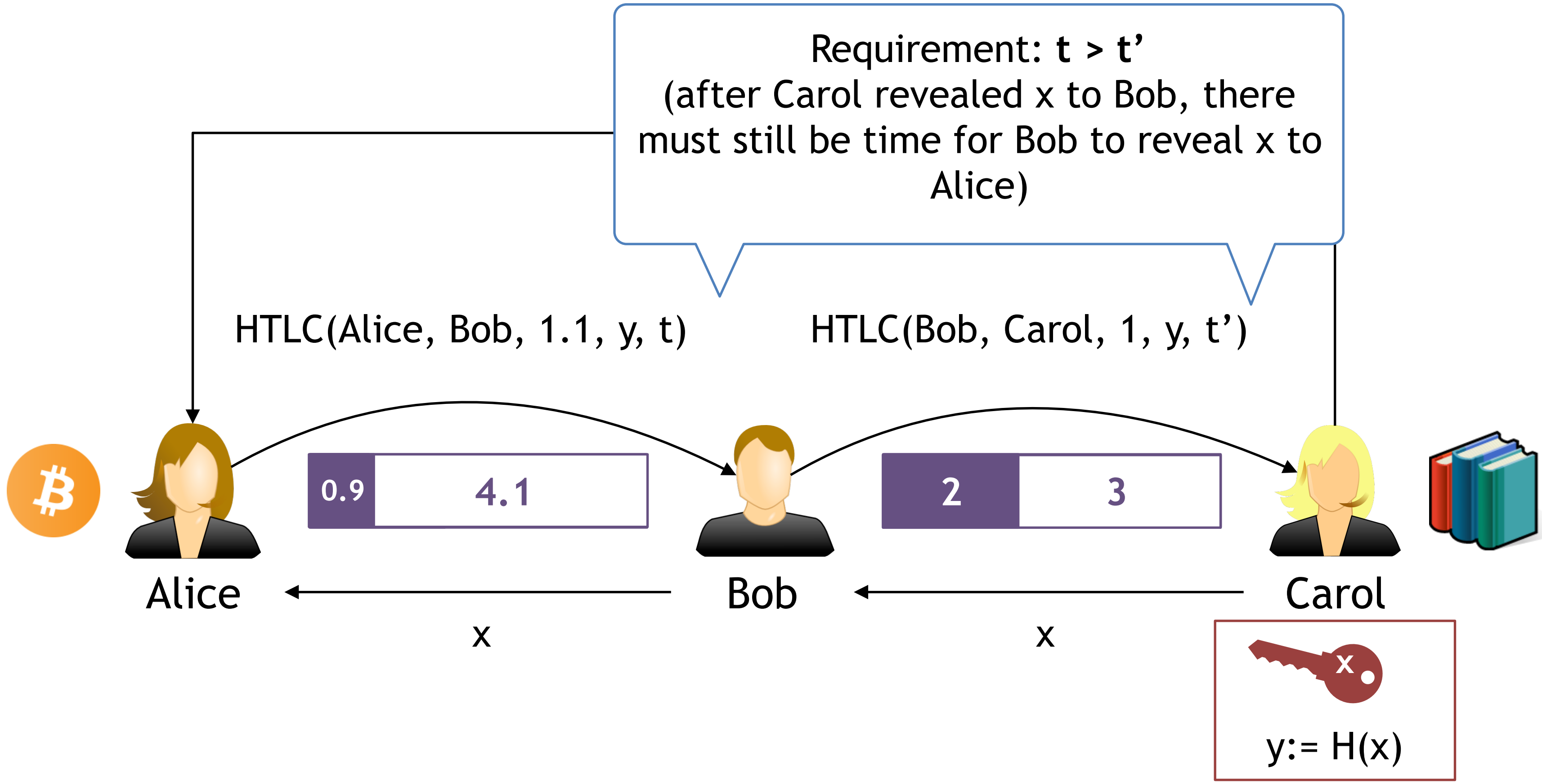
HTLC for Multi-hop Payments



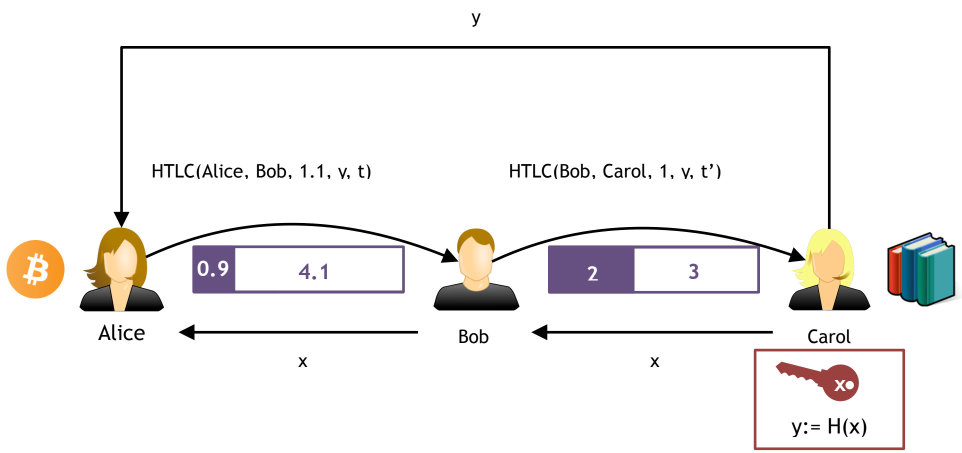
HTLC for Multi-hop Payments



HTLC for Multi-hop Payments



Take home...



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

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 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].**

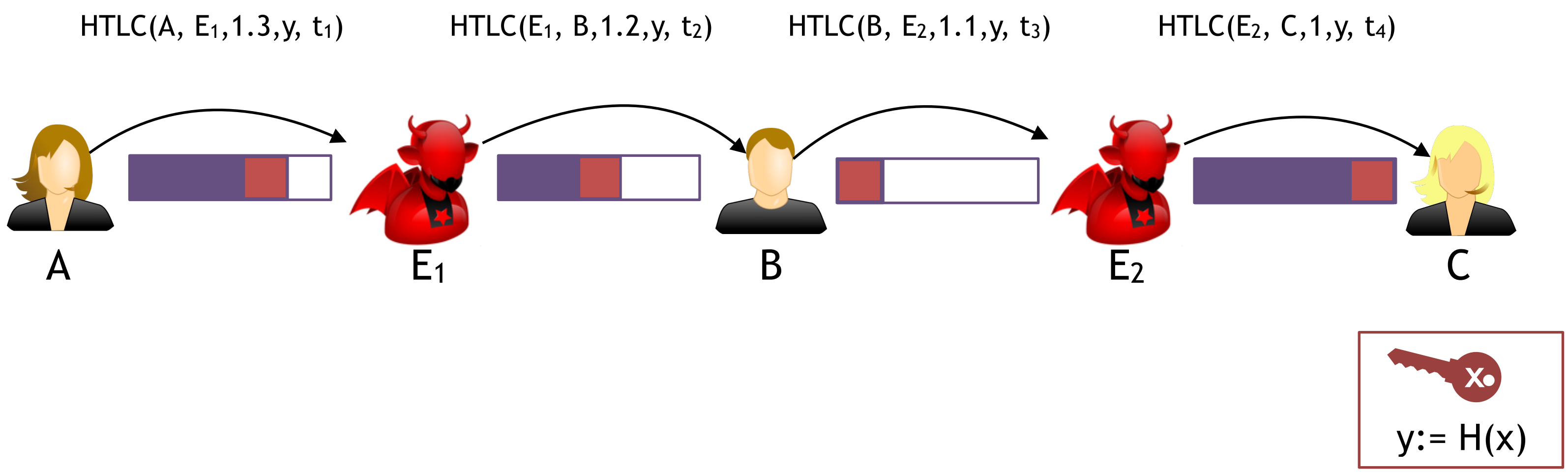
In this work, we study and design secure and privacy-preserving PCNs. We start with a security analysis of existing PCNs, reporting a new attack that applies to all major PCNs, including the Lightning Network, and allows an attacker to steal the fees from honest intermediaries in the same payment path. 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

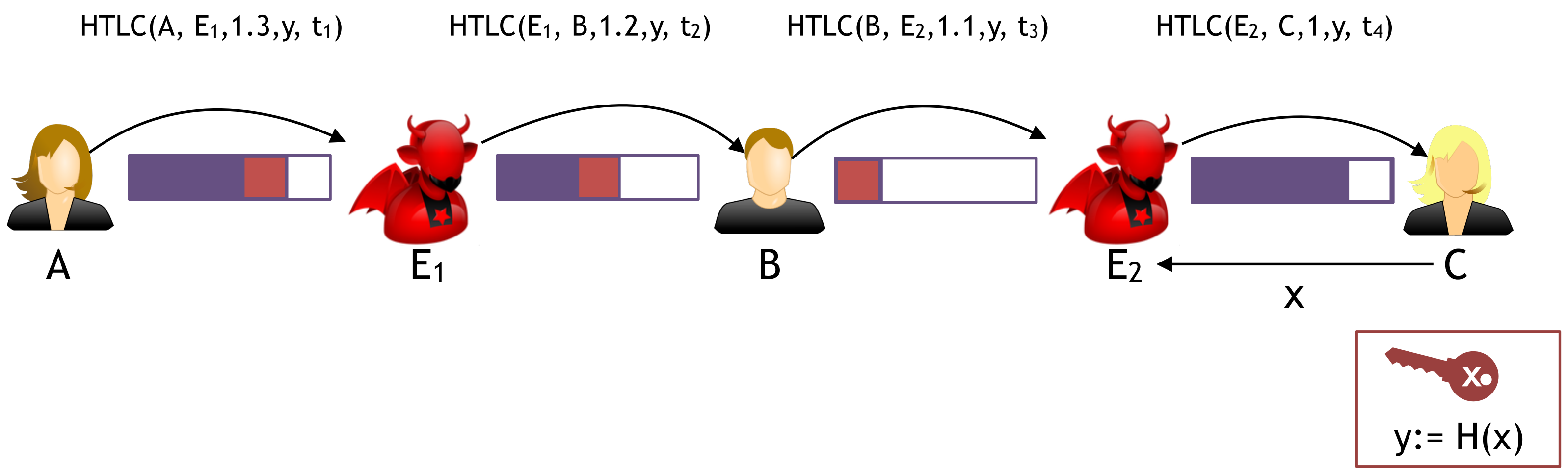
Cryptocurrencies are growing in popularity and are playing an increasing role in the worldwide financial ecosystem. In fact, the number of Bitcoin transactions grew by approximately 30% in 2017, reaching a peak of more than 420,000 transactions per day in December 2017 [2]. This striking increase in 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

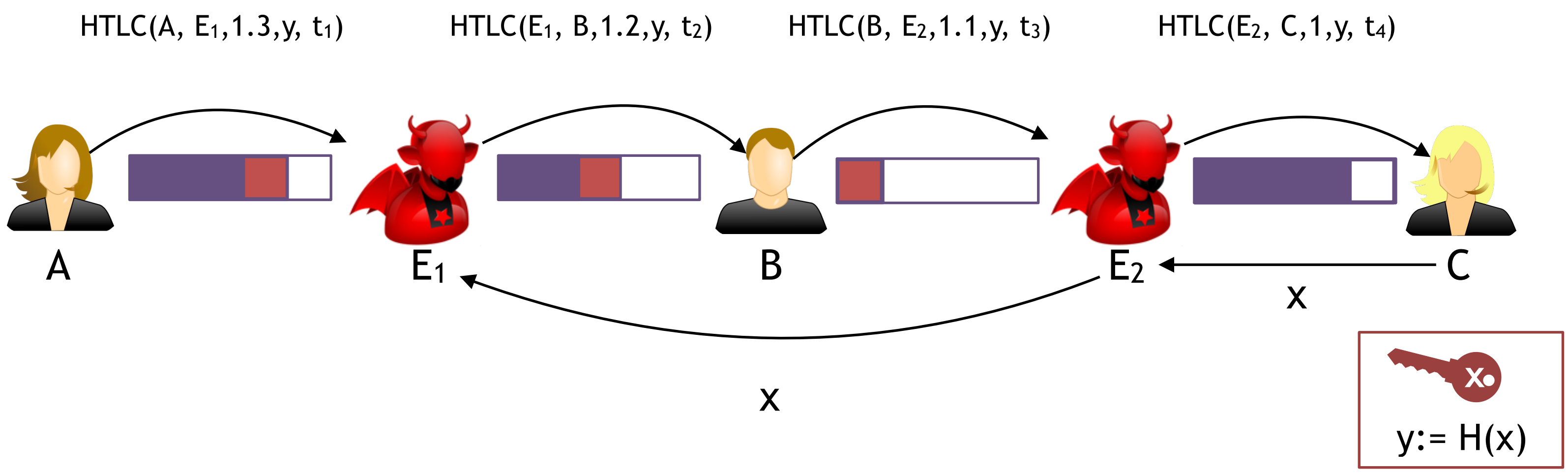
Security Issue: The Wormhole Attack



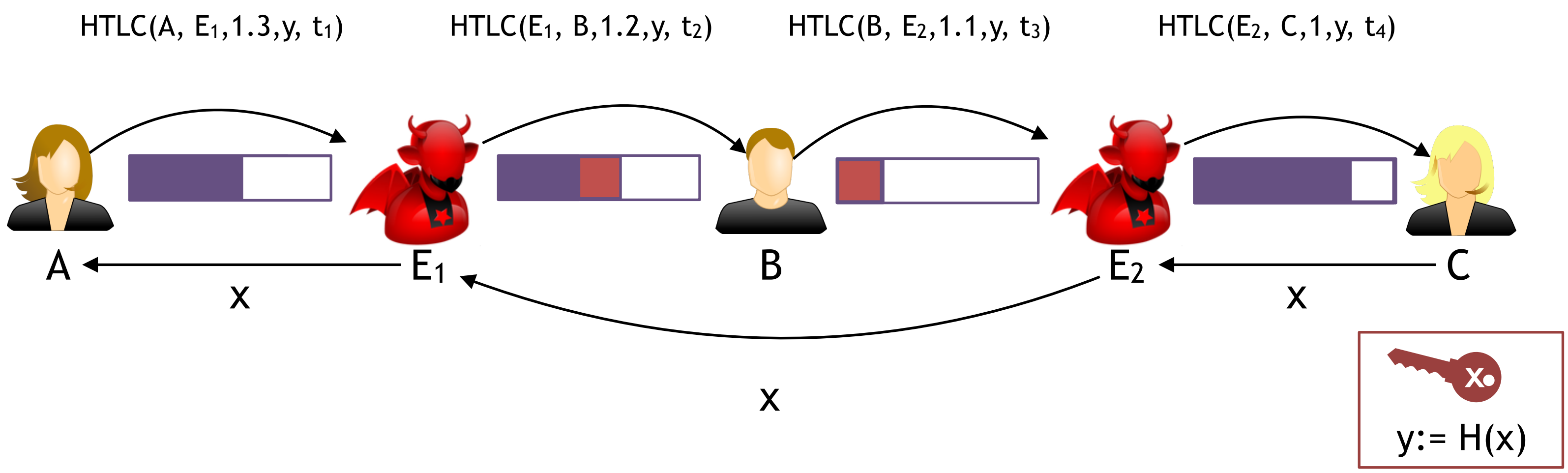
Security Issue: The Wormhole Attack



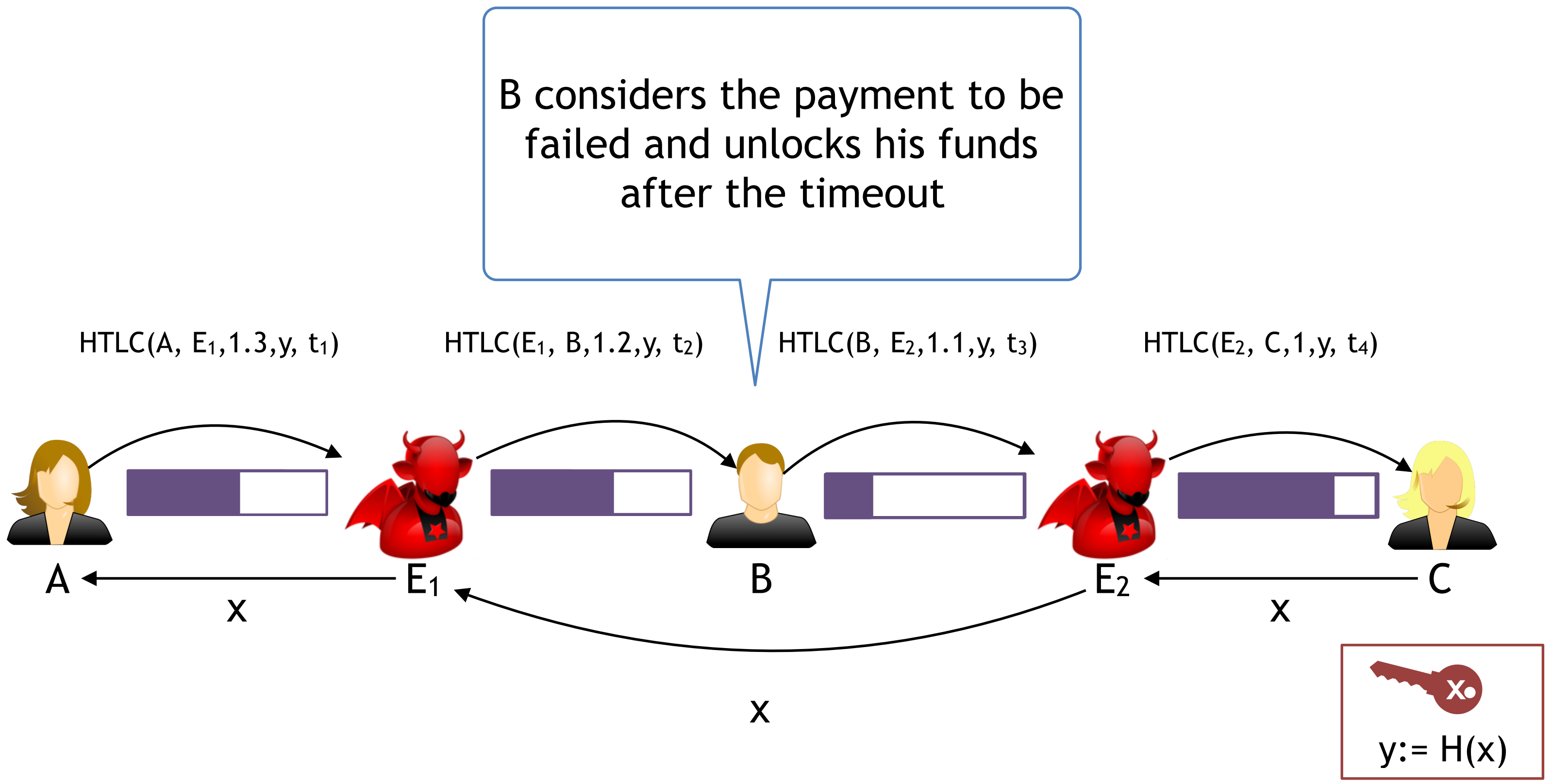
Security Issue: The Wormhole Attack



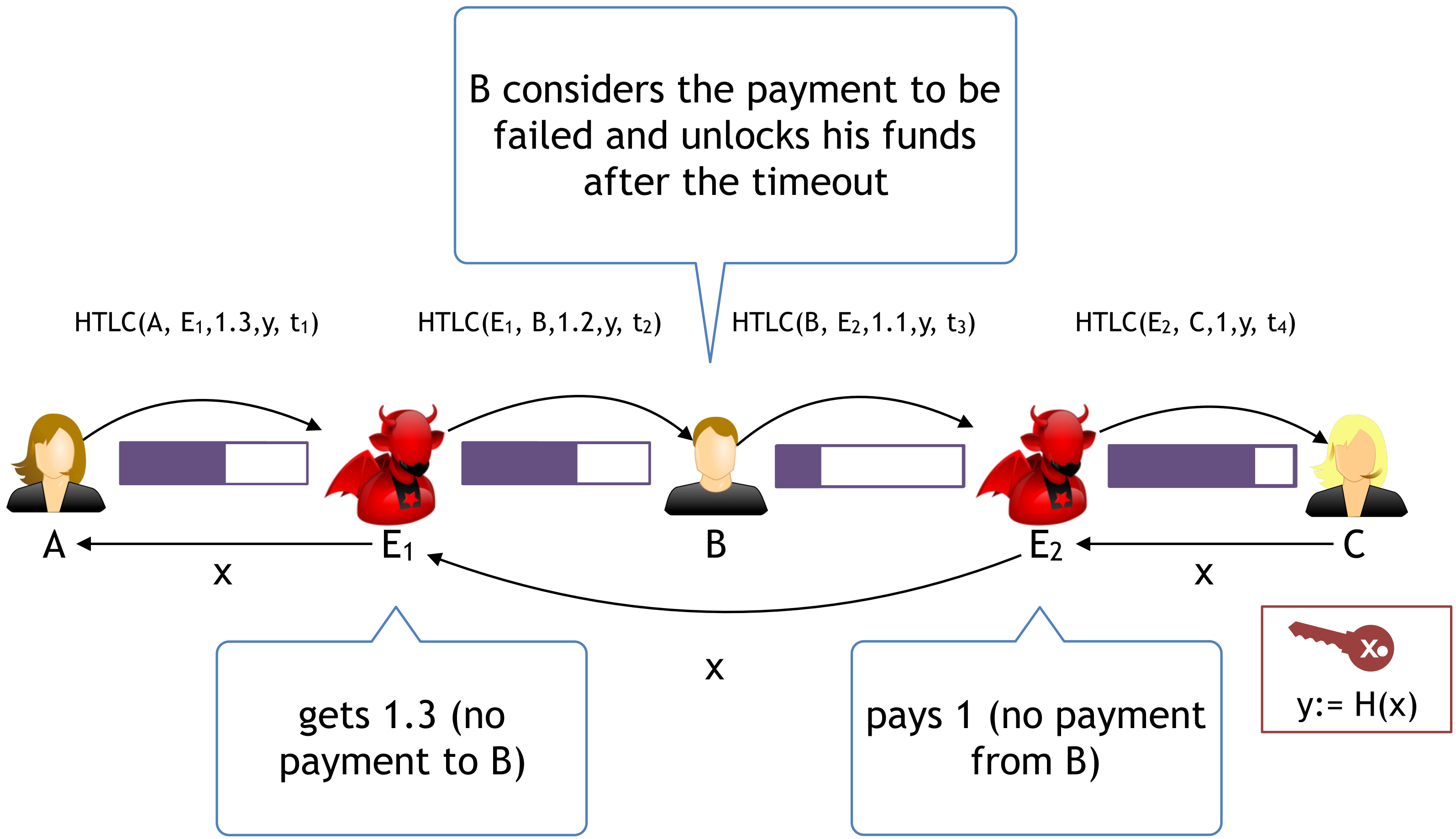
Security Issue: The Wormhole Attack



Security Issue: The Wormhole Attack

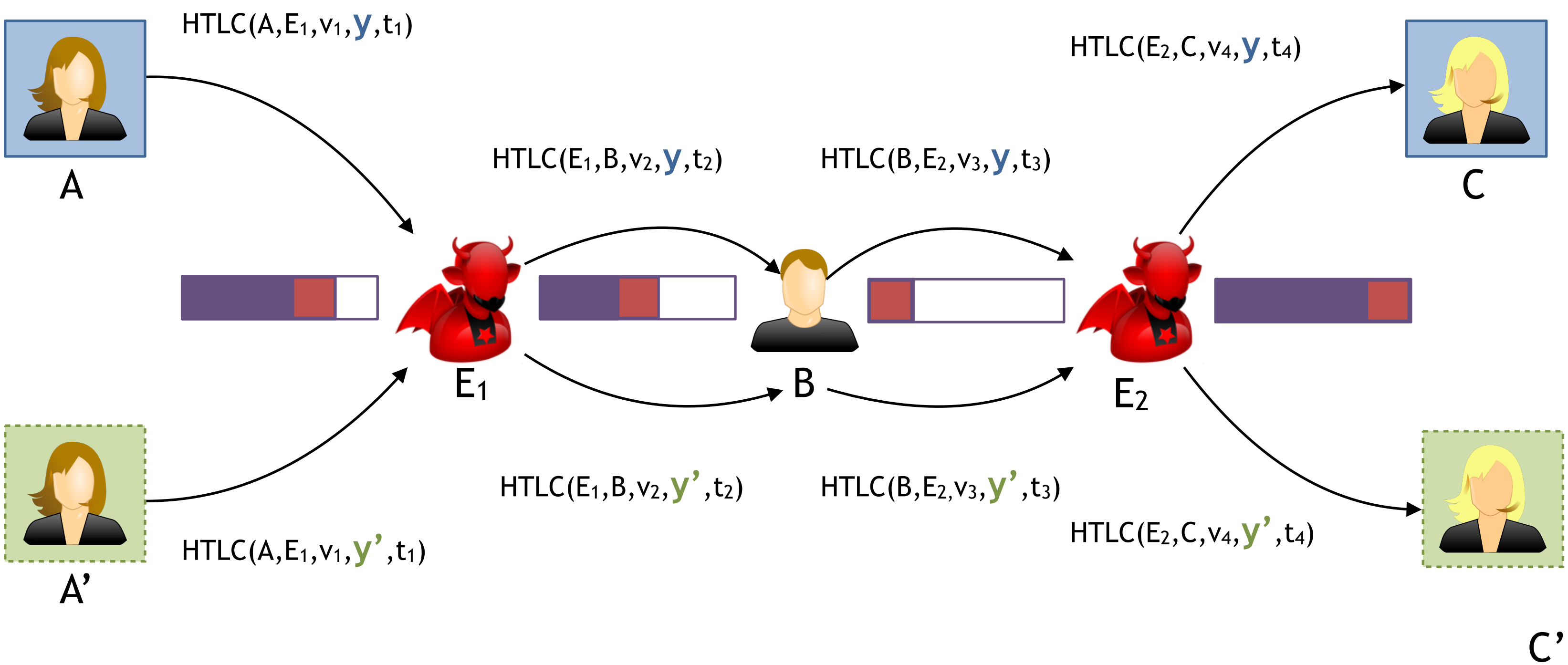


Security Issue: The Wormhole Attack



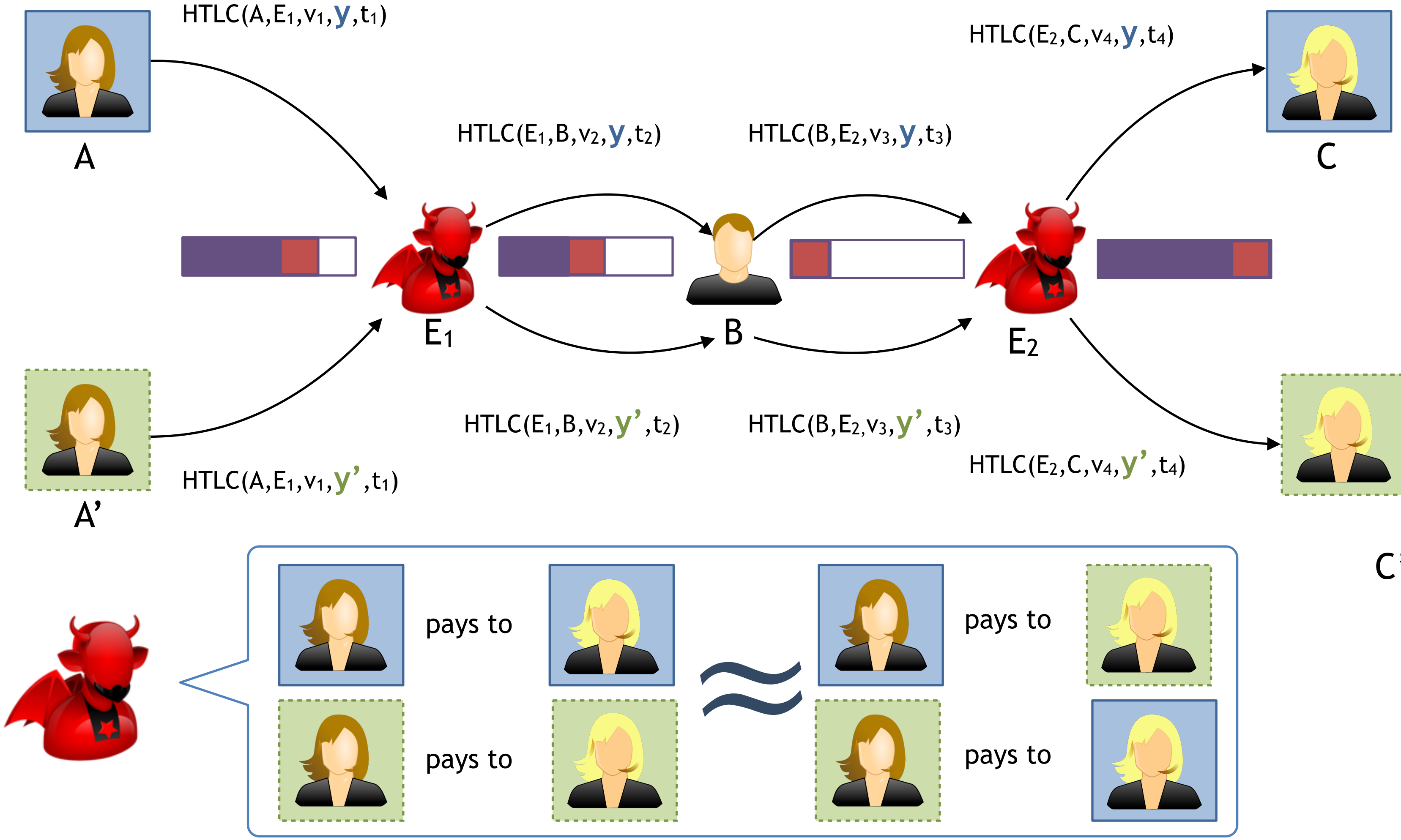
Attacker earns 0.3 BTC (own fees + B's fees)

Privacy Issues in HTLC Payments



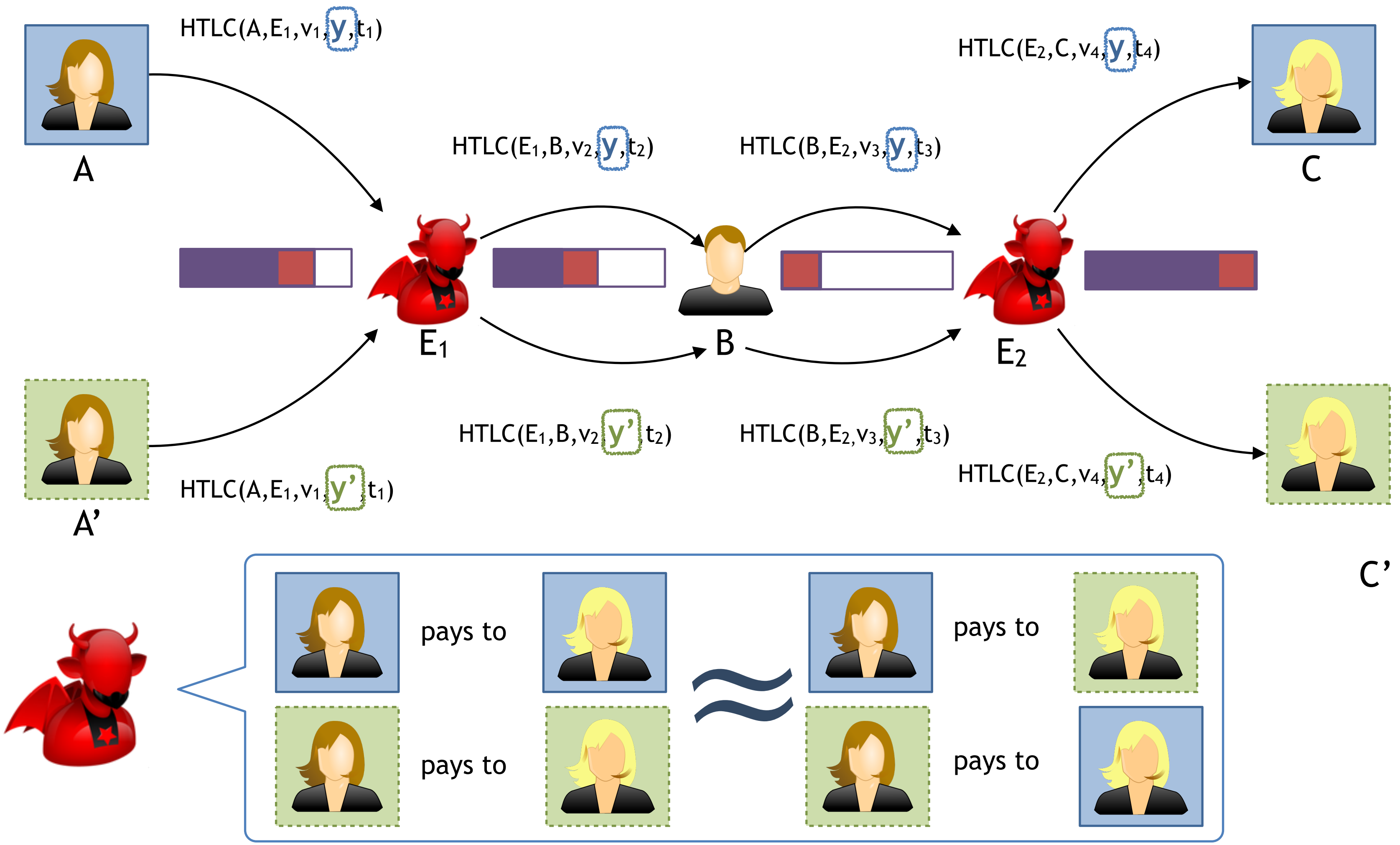
Relationship Anonymity: On-path adversaries do not learn who pays to whom

Privacy Issues in HTLC Payments



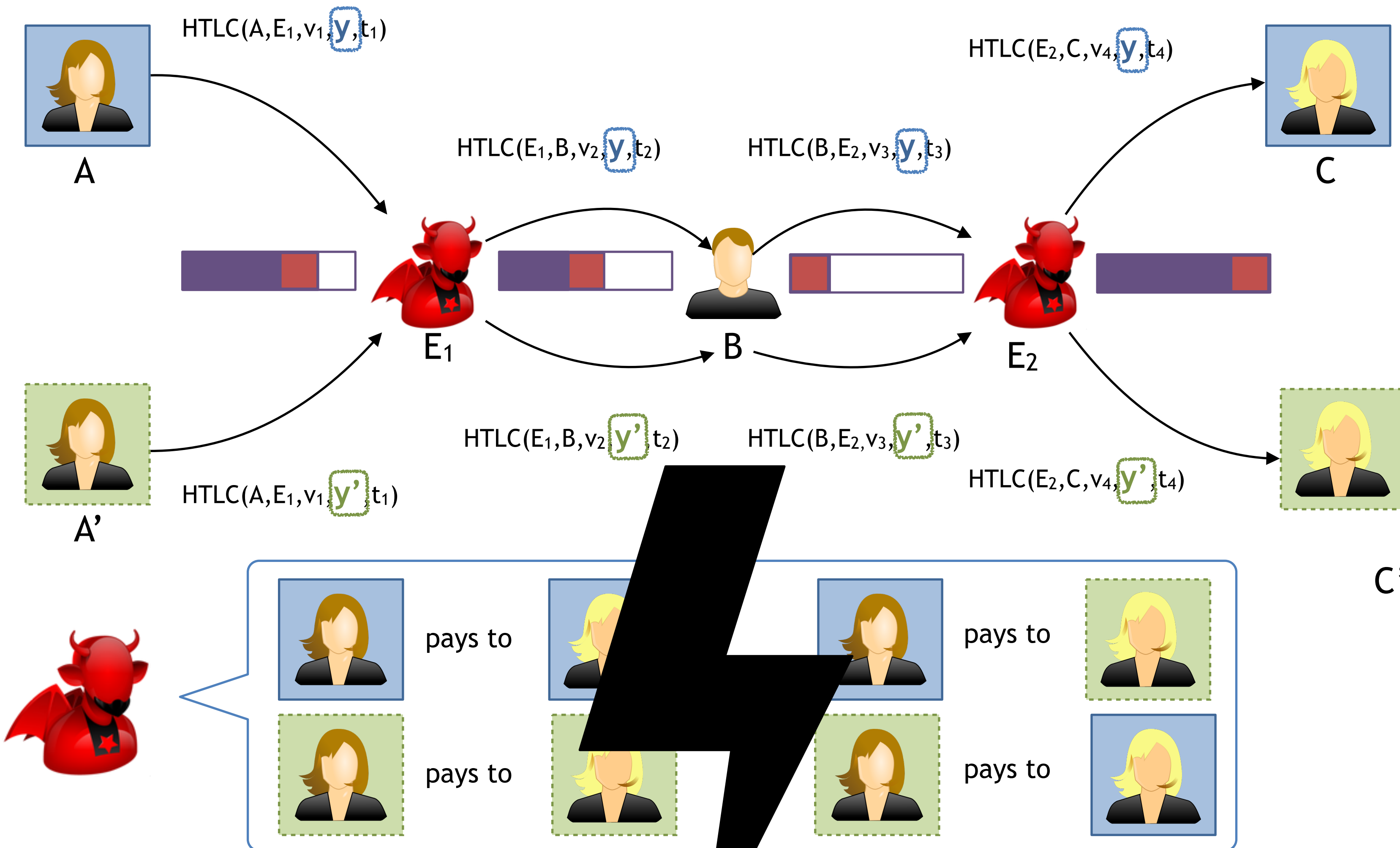
Relationship Anonymity: On-path adversaries do not learn who pays to whom

Privacy Issues in HTLC Payments



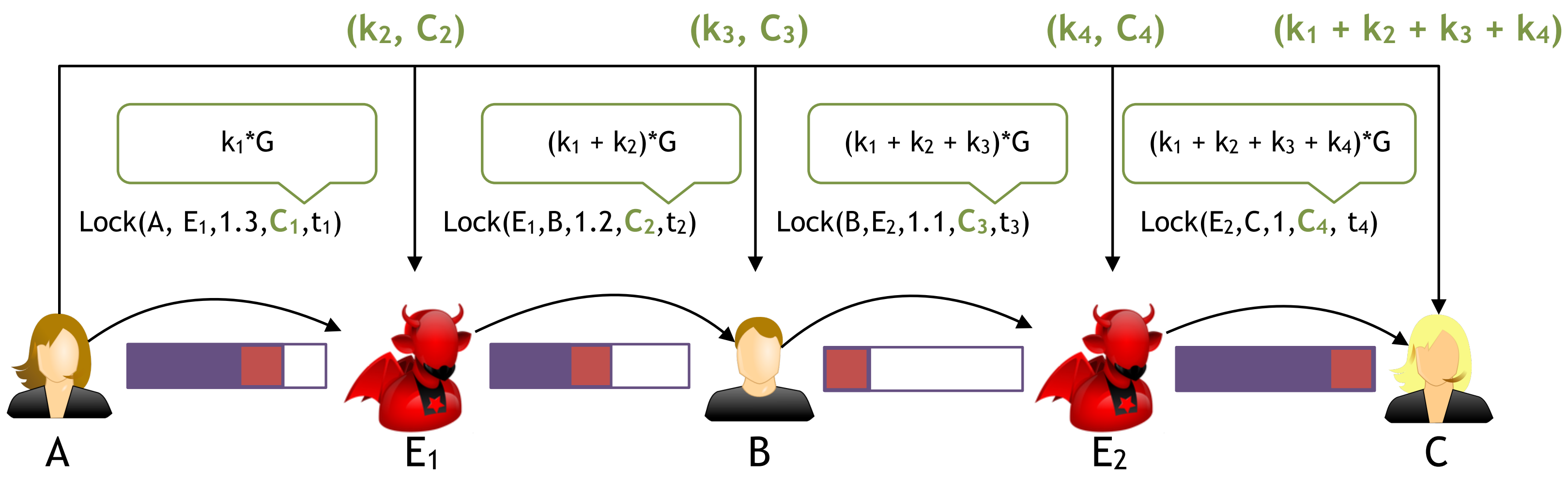
Relationship Anonymity: On-path adversaries do not learn who pays to whom

Privacy Issues in HTLC Payments

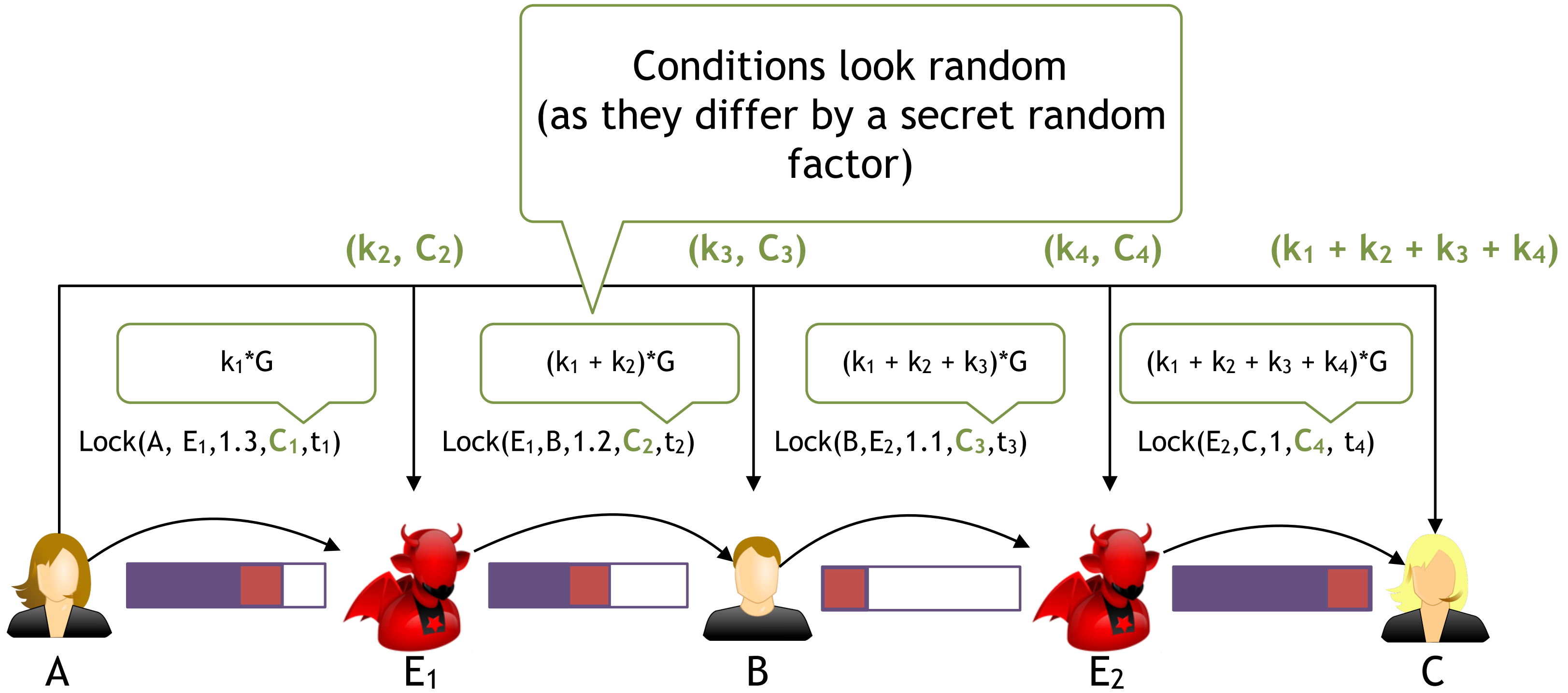


Relationship Anonymity: On-path adversaries do not learn who pays to whom

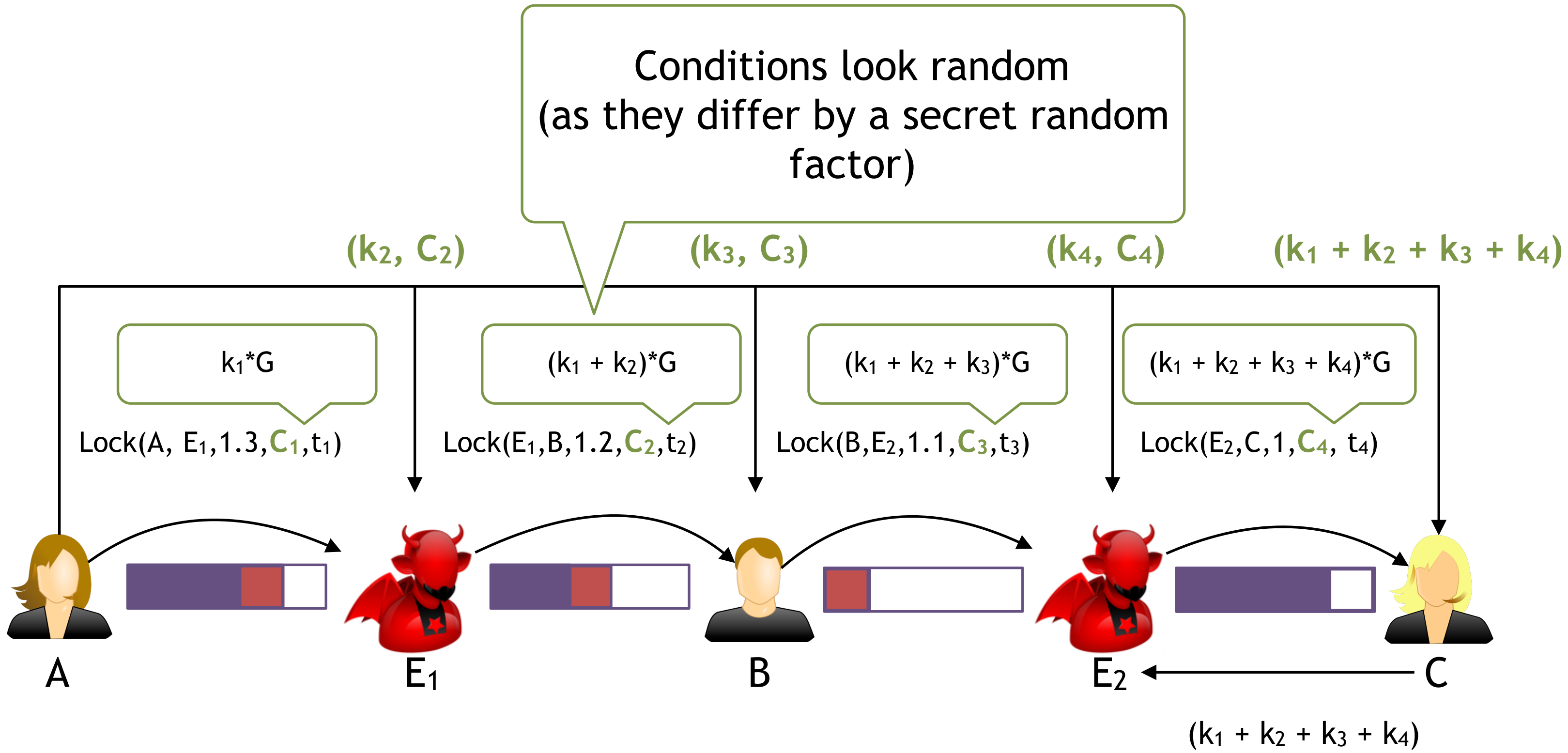
Anonymous Multi-hop Locks



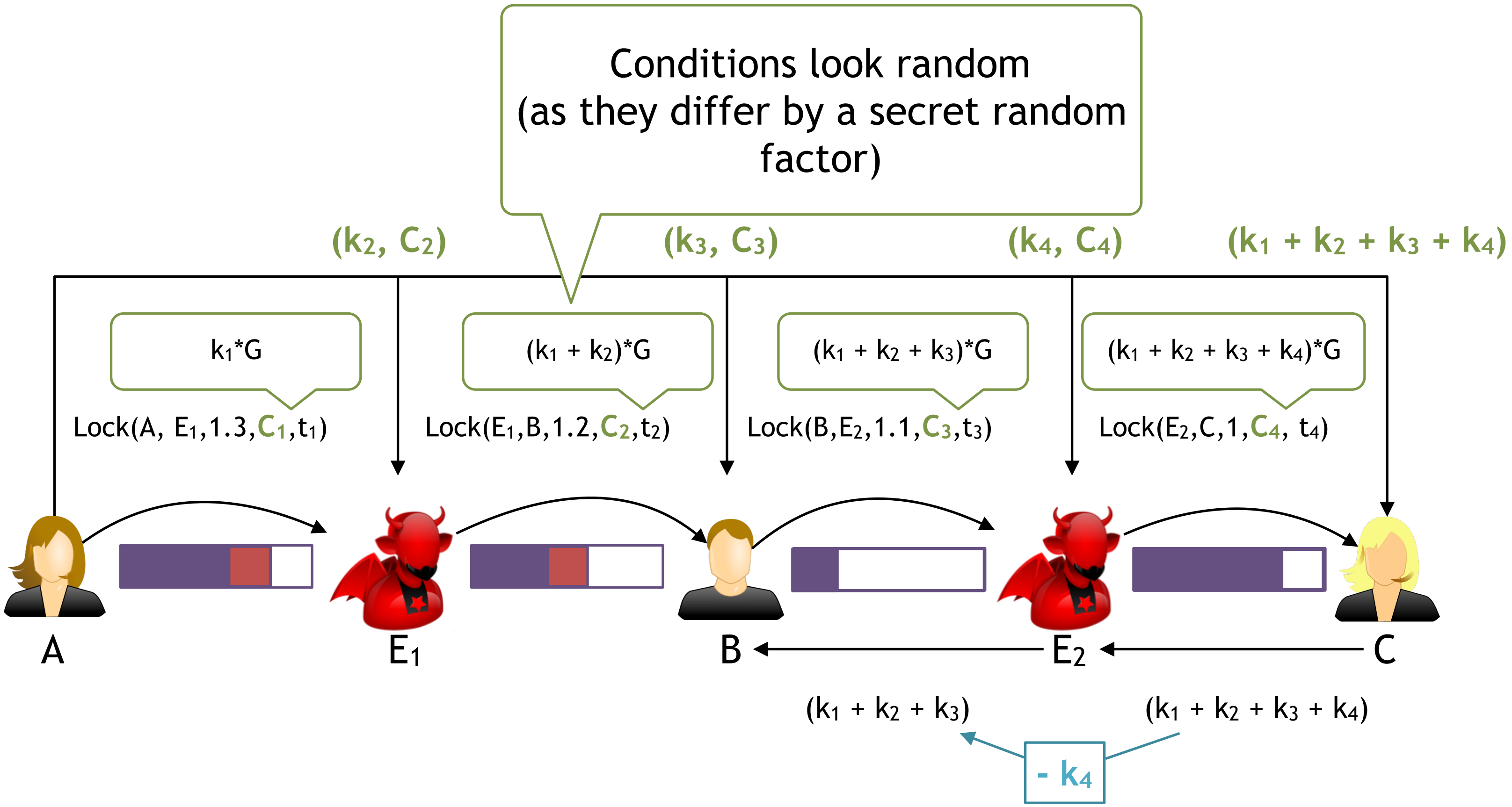
Anonymous Multi-hop Locks



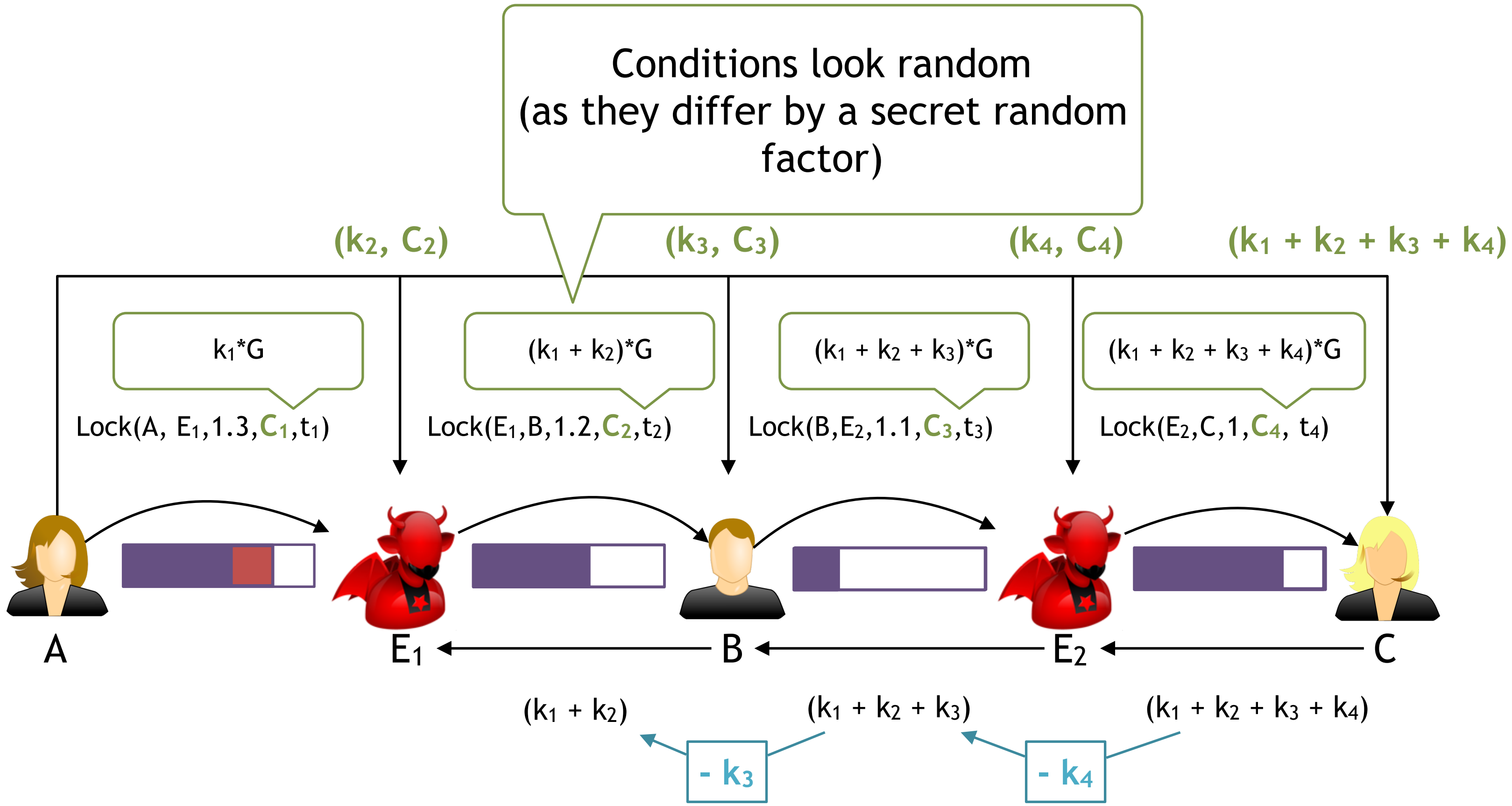
Anonymous Multi-hop Locks



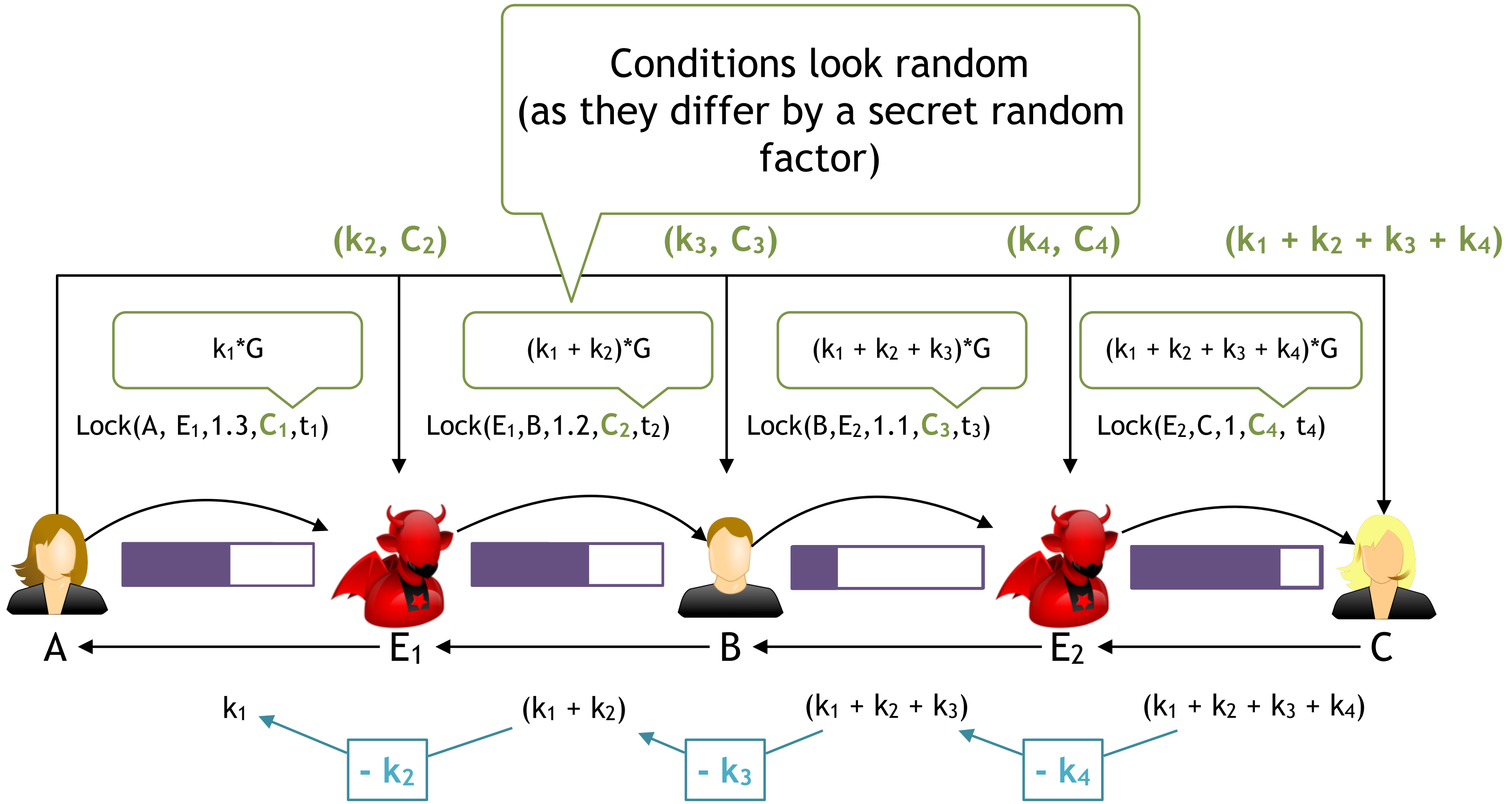
Anonymous Multi-hop Locks



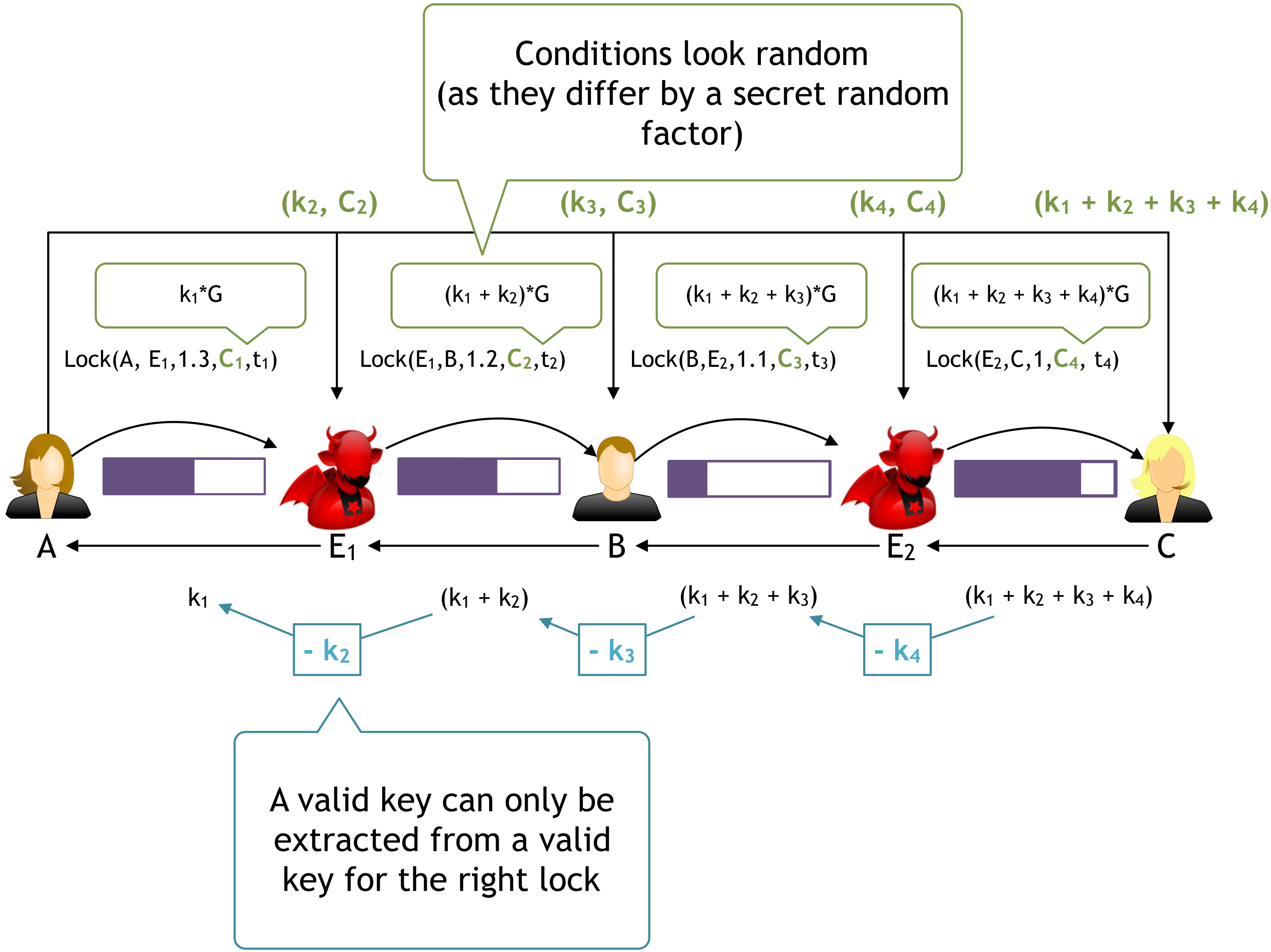
Anonymous Multi-hop Locks



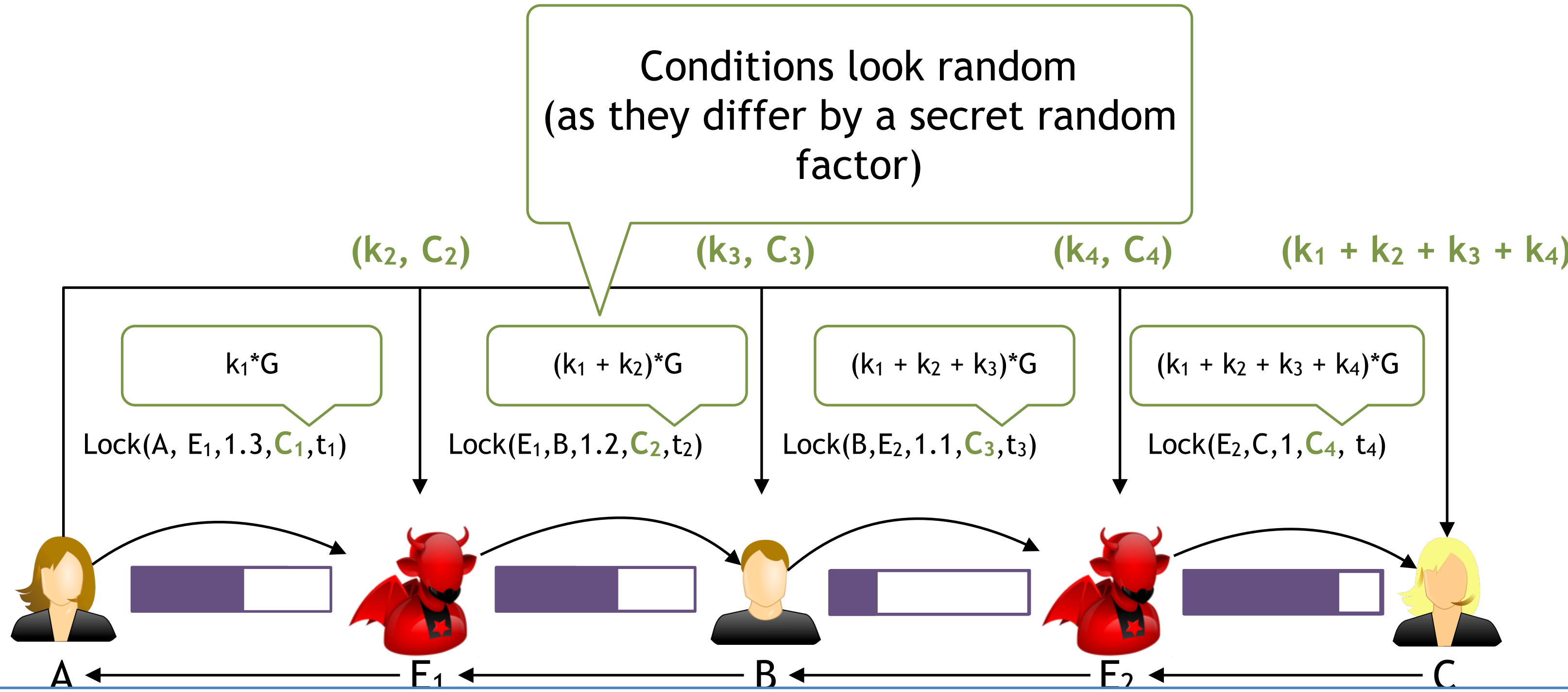
Anonymous Multi-hop Locks



Anonymous Multi-hop Locks



Anonymous Multi-hop Locks



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)

Ideal functionality
(capturing **atomicity**, **consistency** + **relationship anonymity**)

provably realise in the UC framework

Construction from **homomorphic one-way functions**

Schnorr-based construction

ECDSA-based construction

compatible with Bitcoin, Ethereum, etc.

Anonymous Multi-hop-Locks (AMHL)

Ideal functionality
(capturing **atomicity**, **consistency** + **relationship anonymity**)

provably realise in the UC framework

Construction from **homomorphic one-way functions**

Schnorr-based construction

ECDSA-based construction

compatible with
Bitcoin, Ethereum, etc.

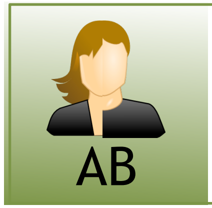
Scriptless Scripts

Scriptless Scripts

5

Alice
(sk_A)

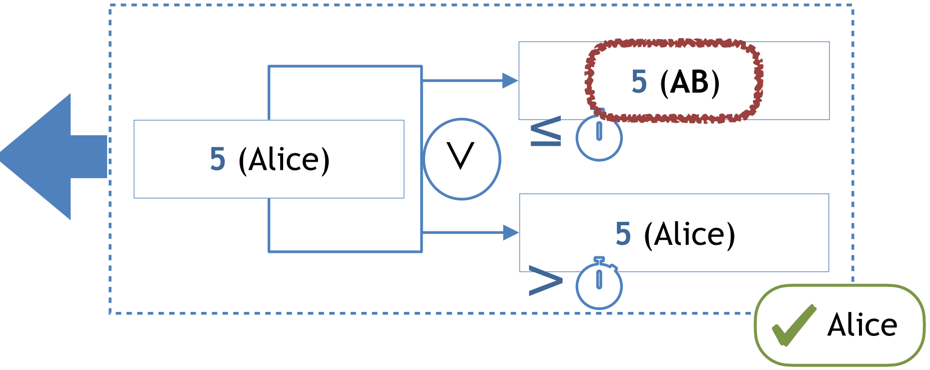
Bob
(sk_B)



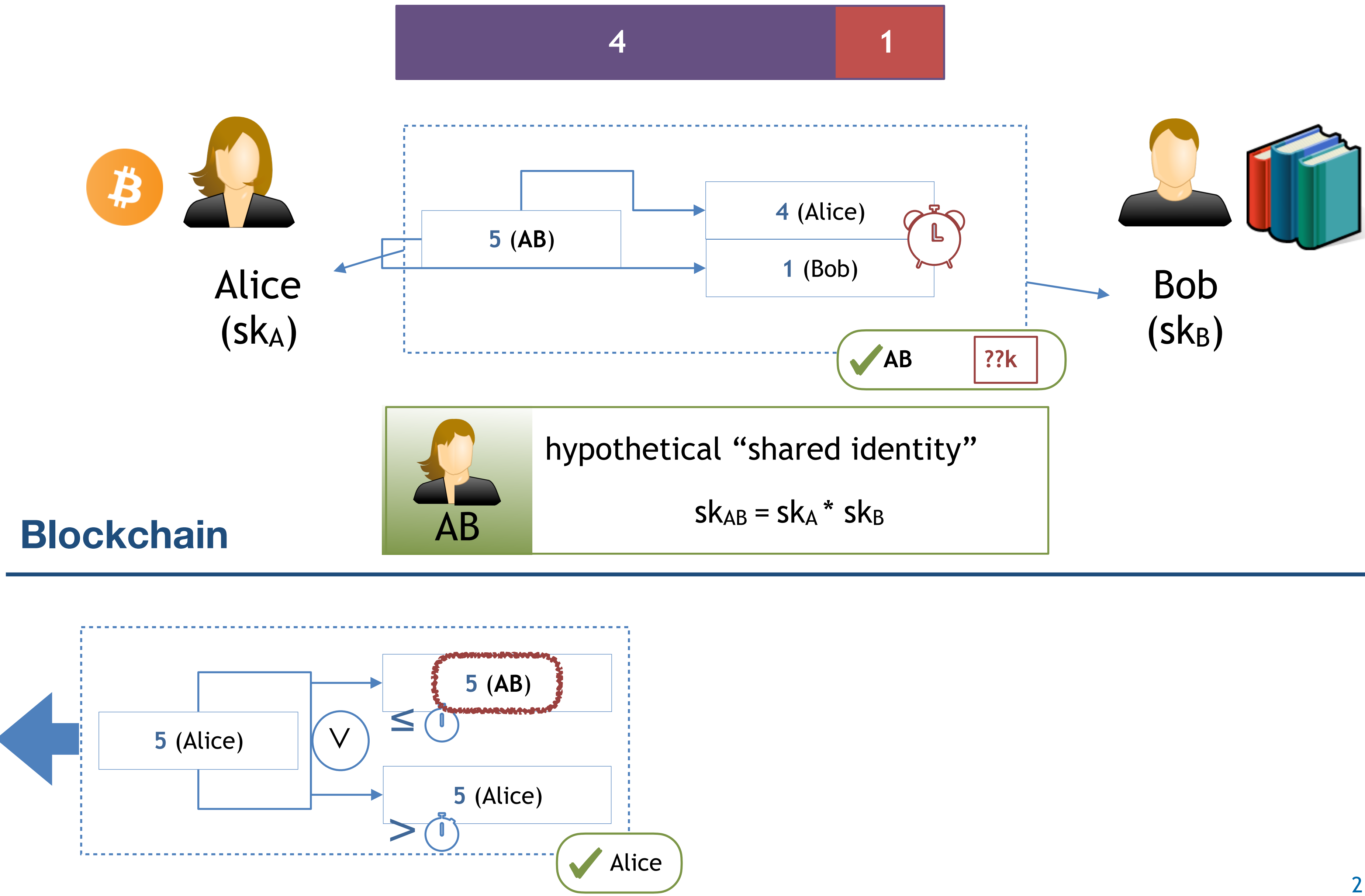
hypothetical "shared identity"

$sk_{AB} = sk_A * sk_B$

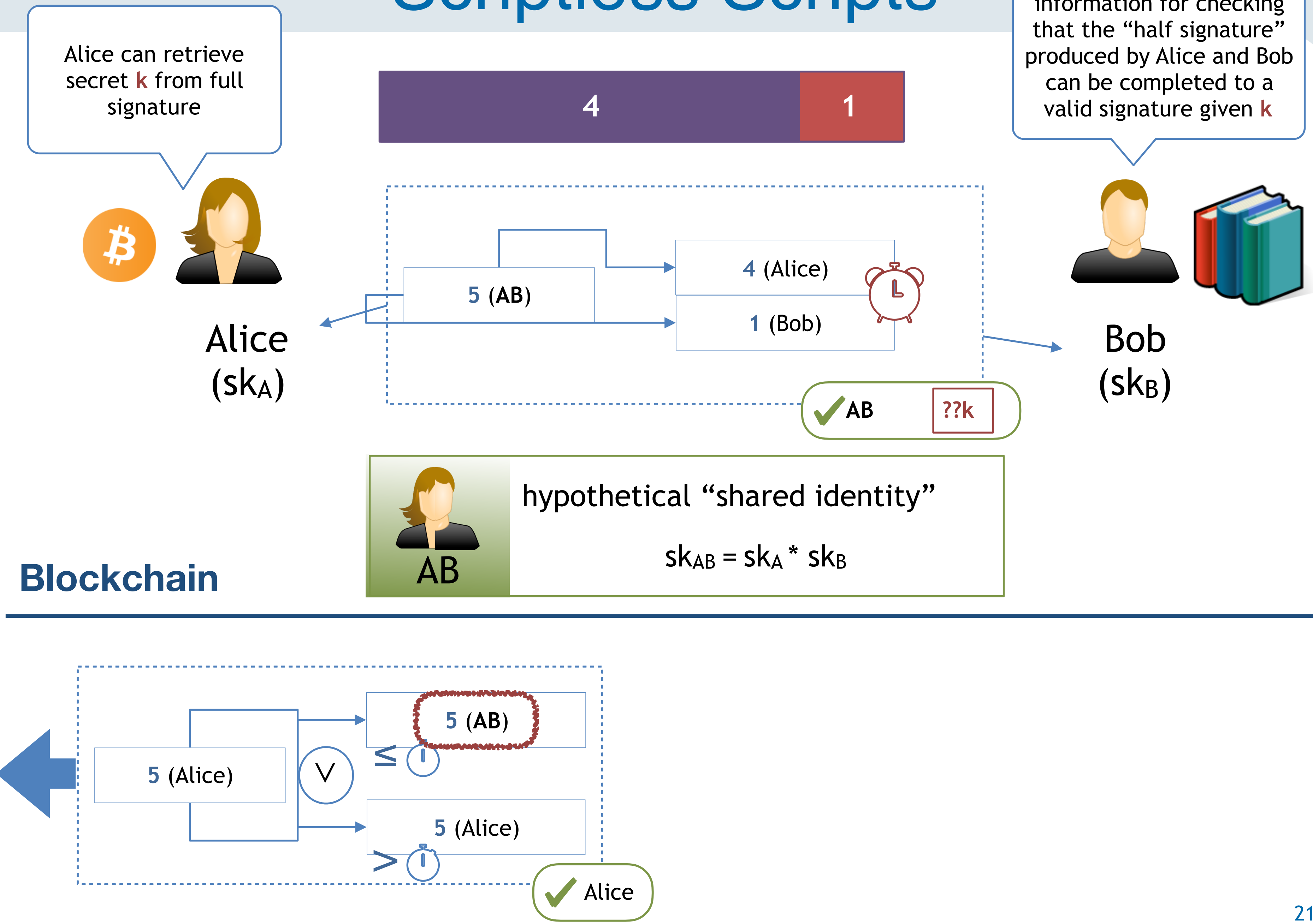
Blockchain



Scriptless Scripts



Scriptless Scripts



Schnorr-based Lock, Simplified

$$sk_I = x_I$$
$$pk_I = x_I \cdot G$$
$$R_I = r_I \cdot G$$
$$sig(r_i, m, sk, pk) = (R_I, r_i - sk_i \cdot H(pk_i || R_I || m))$$

Schnorr Signature for *I*

Schnorr-based Lock, Simplified

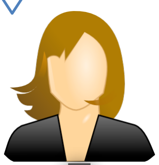
$sk_I = x_I$
$pk_I = x_I \cdot G$
$R_I = r_I \cdot G$
$sig(r_i, m, sk, pk) = (R_I, r_i - sk_i \cdot H(pk_i || R_I || m))$

Schnorr Signature for *I*

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

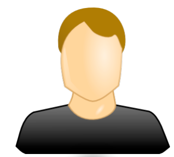


C=k*G, transaction

Jointly, randomly generate C

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

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



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

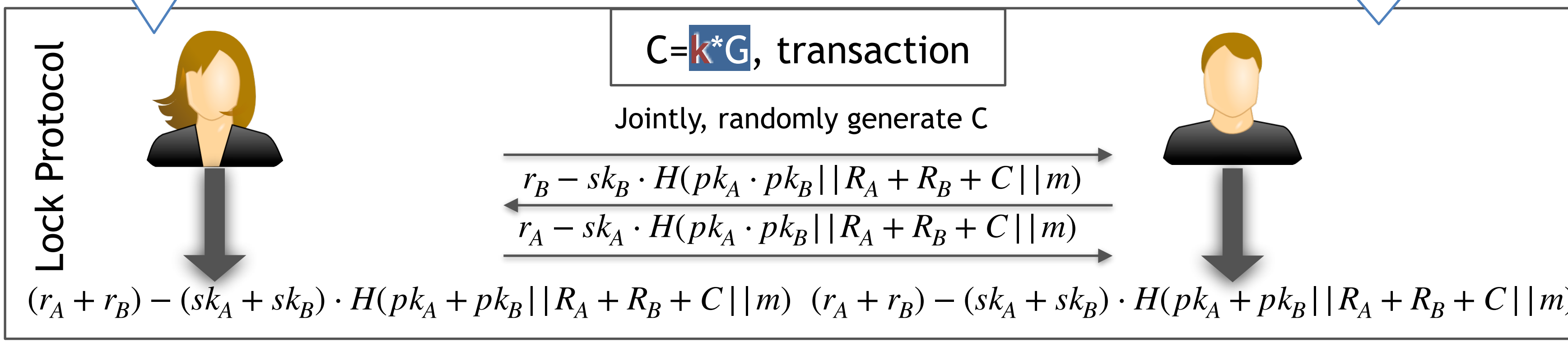
Schnorr-based Lock, Simplified

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

$sk_I = x_I$
$pk_I = x_I \cdot G$
$R_I = r_I \cdot G$
$sig(r_i, m, sk, pk) = (R_I, r_i - sk_i \cdot H(pk_i || R_I || m))$

Schnorr Signature for *I*

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



After learning *k*, Bob can finalise the signature as

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

And Alice can derive *k* from it

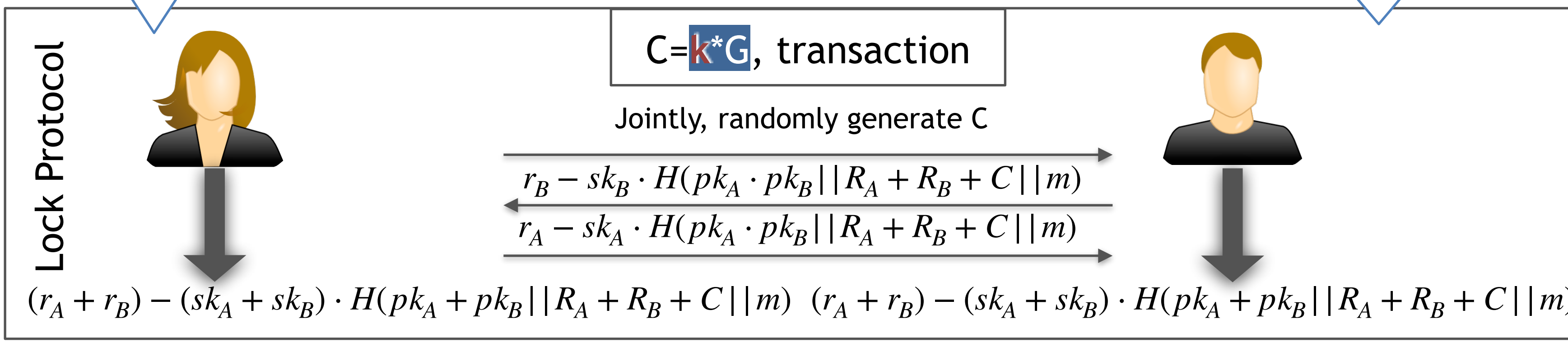
Schnorr-based Lock, Simplified

$$sk_I = x_I$$
$$pk_I = x_I \cdot G$$
$$R_I = r_I \cdot G$$
$$sig(r_i, m, sk, pk) = (R_I, r_i - sk_i \cdot H(pk_i || R_I || m))$$

Schnorr Signature for *I*

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



After learning *k*, Bob can finalise the signature as

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

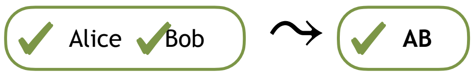
And Alice can derive *k* from it

Hard for ECDSA as σ_R has a non-linear structure, for details please look at the paper

Properties/Evaluation

- Security and Privacy proven in the UC Framework
- Compatible with Bitcoin and current PCNs
  - ✓ Implemented in the Lightning Network (https://github.com/cfromknecht/tpec), KZen, Comet, …
- Reduces transaction size for conditional payments
  - ✓ Encoding of condition within signature

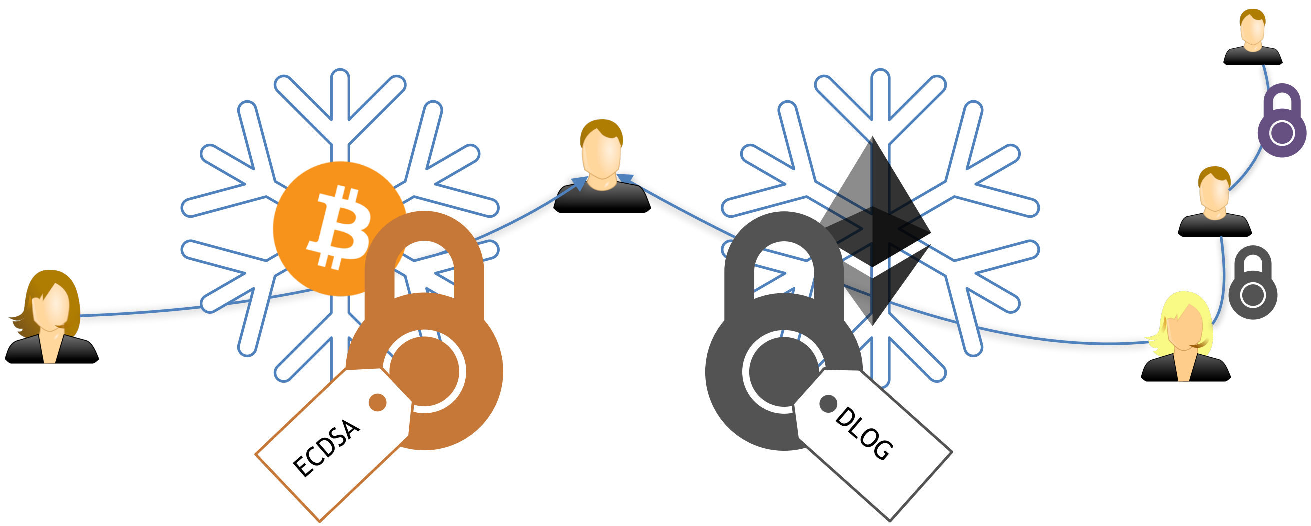
- Makes settlement transactions indistinguishable from regular ones (Fungibility)



- Little overhead:
  - ✓ < 500 bytes communication
  - ✓ few ms computation

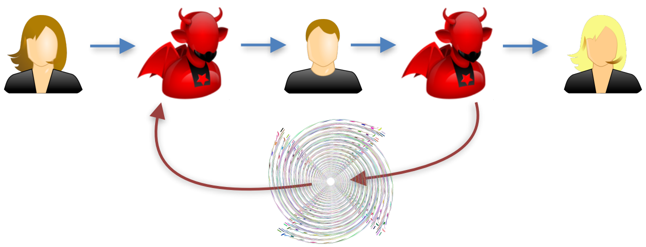
Interoperability

- AMHLs are suitable for cross-currency usage, even with different primitive instantiations
  - ✓ Inter-currency payment channels
  - ✓ Atomic swaps
  - ✓ All major cryptocurrencies (including Monero [Moreno-Sanchez et al., FC'20]) are supported

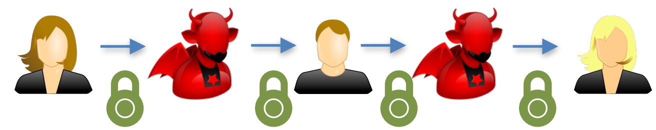


Summary

The **Wormhole Attack**:
A novel attack on Payment Channel Network Security

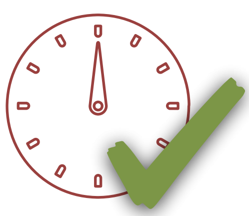


AMHLs: A **new primitive** for secure + anonymous Payment Channel Networks



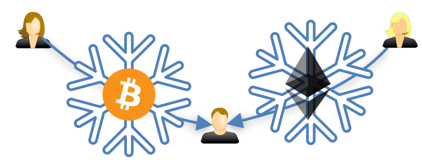
Concrete **constructions** of AMHLs that

… are efficient



… got implemented in Bitcoin's Lightning Network

… enable inter-blockchain Payment Channels



Beyond Path-based Transactions

ACM CCS 2019

Atomic Multi-Channel Updates with Constant Collateral in Payment-Channel Networks

ABSTRACT

Current cryptocurrencies provide a heavily limited transaction throughput that is clearly insufficient to cater their growing adoption. Payment-channel networks (PCNs) have emerged as the most widely deployed scalability solution for today's cryptocurrencies. While PCNs do increase the transaction throughput by processing payments off-chain and using the blockchain only as a dispute arbitrator, they unfortunately require high collateral (i.e., they lock coins for a non-constant time along the payment path) and do not achieve atomicity of the channel updates. These issues have severe consequences in practice. The high collateral enables denial-of-service attacks that hamper the throughput and utility of the PCN. Moreover, the lack of atomicity hinders the applicability of current PCNs in many important application scenarios. Unfortunately, current proposals do not solve either of these issues or they require expressive scripting languages, constraining their deployment to Ethereum.

In this work, we present AMCU, the first protocol for atomic multi-channel updates and reduced collateral that is compatible with Bitcoin (and other cryptocurrencies with reduced scripting capabilities). We provide a formal model in the Universal Composability framework and show that AMCU realizes it, thus demonstrating that AMCU achieves atomicity and state privacy. Moreover, the reduced collateral mitigates the consequences of DoS attacks in PCNs while

Then, both users issue ledger changes with each other through off-chain accountable messages. Finally, when they are done, they set the last agreed ledger state on the blockchain to get the corresponding coins. For instance, Alice can open a channel with Bob by publishing on the blockchain a transaction that transfers x coins from her to an address addr shared by Alice and Bob. Subsequent payments from Alice to Bob only require that Alice sends Bob an off-chain signed transaction of $y < x$ coins from addr to him. Bob can close the channel by signing and adding on-chain the last transaction received by Alice. Interestingly, it is possible to generalize this technique to a network of payment channels where two users can pay each other if they are connected through a path of open payment channels [28].

The Lightning Network (LN) [28] for Bitcoin and the Raiden Network [7] for Ethereum are the most widely deployed PCNs in practice, and several implementations exist today [3, 5, 6]. Several academic efforts have focused on designing solutions to enhance the security [22], privacy [17, 21], concurrency [22, 32], availability [23], and routing mechanisms [29] of PCNs, but many fundamental challenges remain open. In this paper we focus on two fundamental ones, namely, *atomicity* and *collateral*, providing a solution to both.

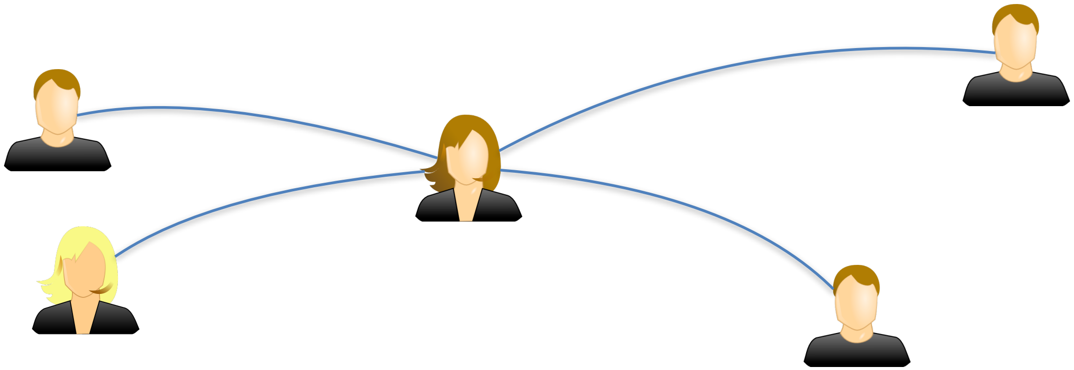
Atomicity Challenge. A long-standing challenge in PCNs is the *atomicity* of updates required in a path of payment channels to perform a multi-hop transaction. Without atomicity, it could be that

Open Challenges

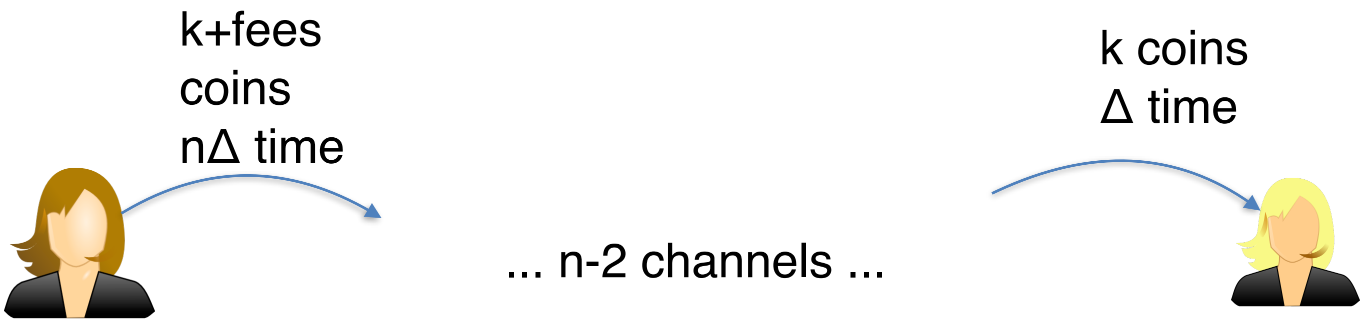
- In this work, we identify two open challenges:
  - Restricted expressiveness (and functionality)
    - Current Bitcoin-compatible PCNs restricted to single path-based payments
  - High collateral
    - A payment requires to put aside coins for a very long time

Our Goal: Full Expressiveness

- Support for arbitrary graph topology
- Enable new applications:
  - Crowd funding
  - Channel rebalancing
  - Netting
  - Your own application?

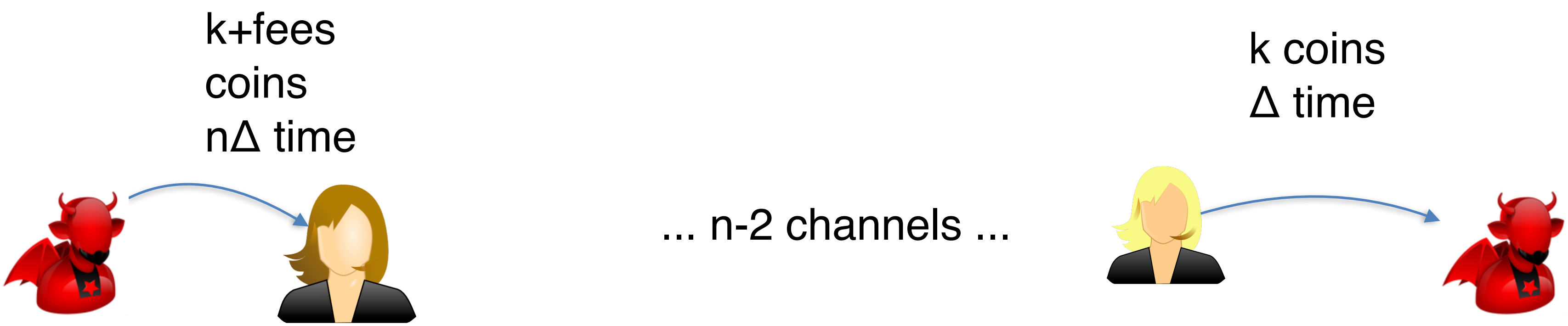


Collateral



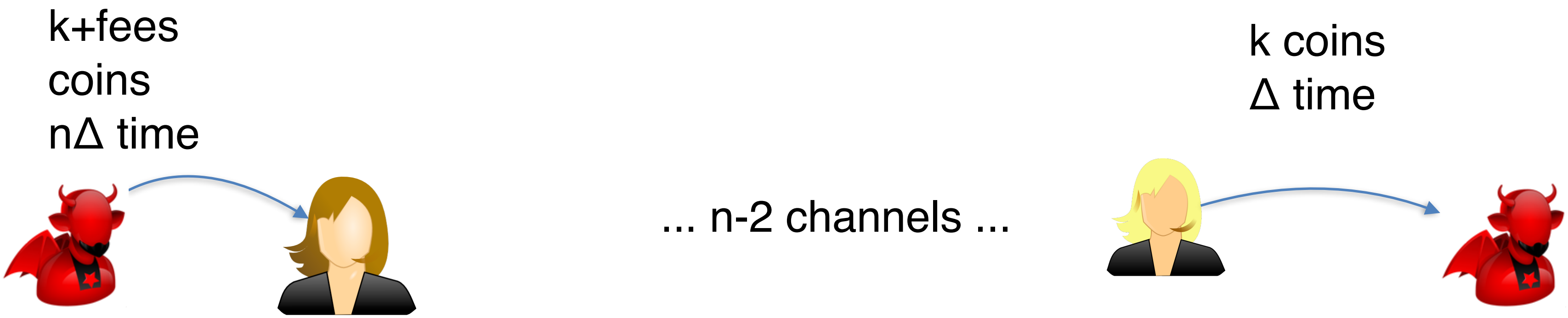
- Each payment of k coins along an n-channel path requires to put aside at least kn coins
- Also, each user i has to lock her coins for a time Δ(n-i) where Δ is the time to safely close a channel
- Coins locked too long!

Griefing attack

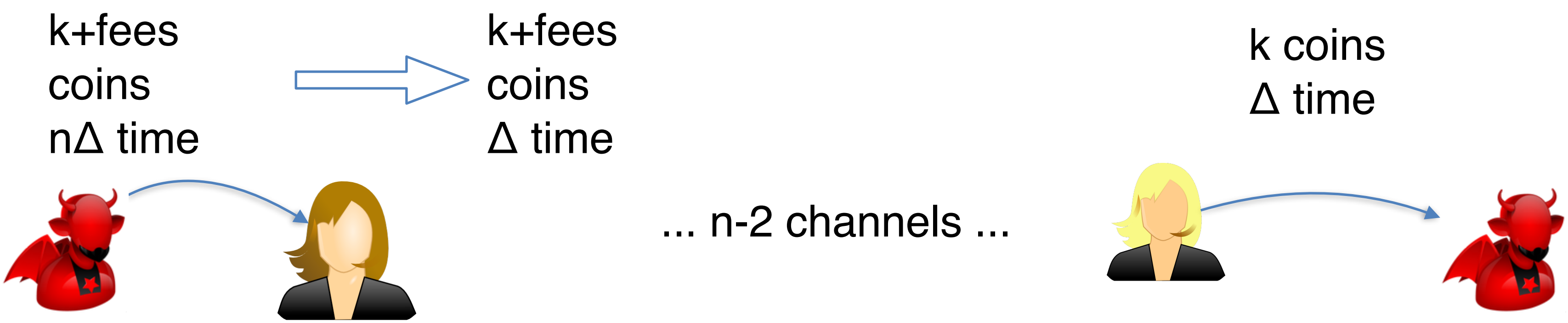


- The adversary has a time amplification factor of n-1
- Δ is 1 day in the Lightning network!
- The attacker can use several paths

Our Goal: Constant Collateral

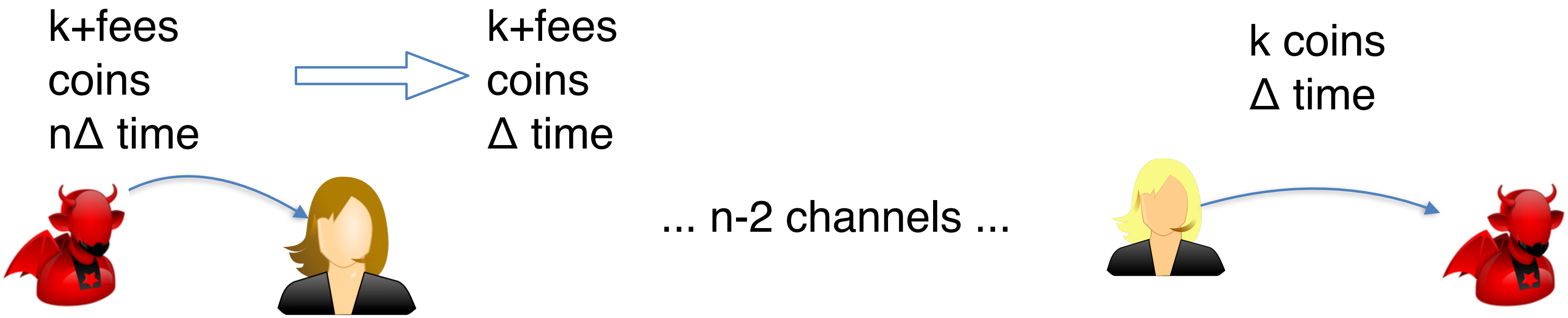


Our Goal: Constant Collateral



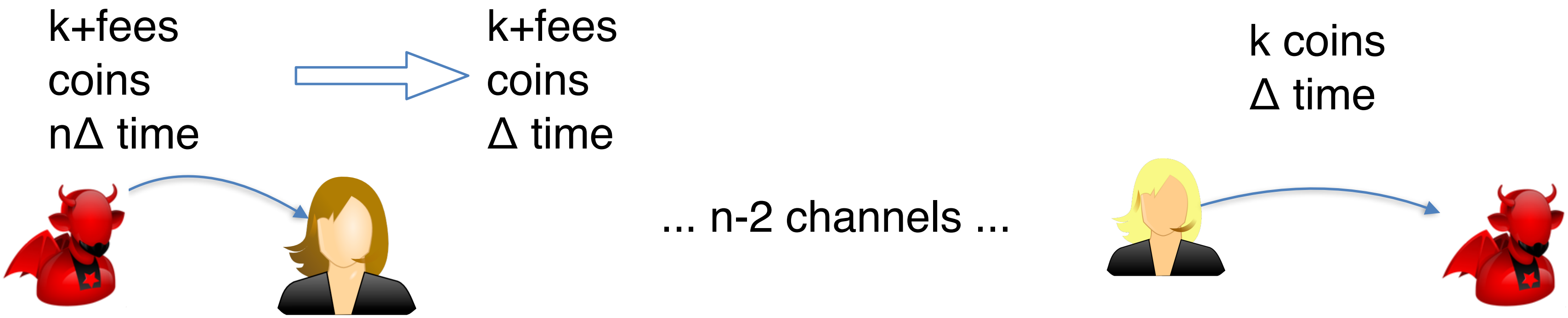
- Constant collateral: Coins are locked only for Δ time, independently of the number of channels

Our Goal: Constant Collateral



- Constant collateral: Coins are locked only for Δ time, independently of the number of channels
- Reduces the amplification factor

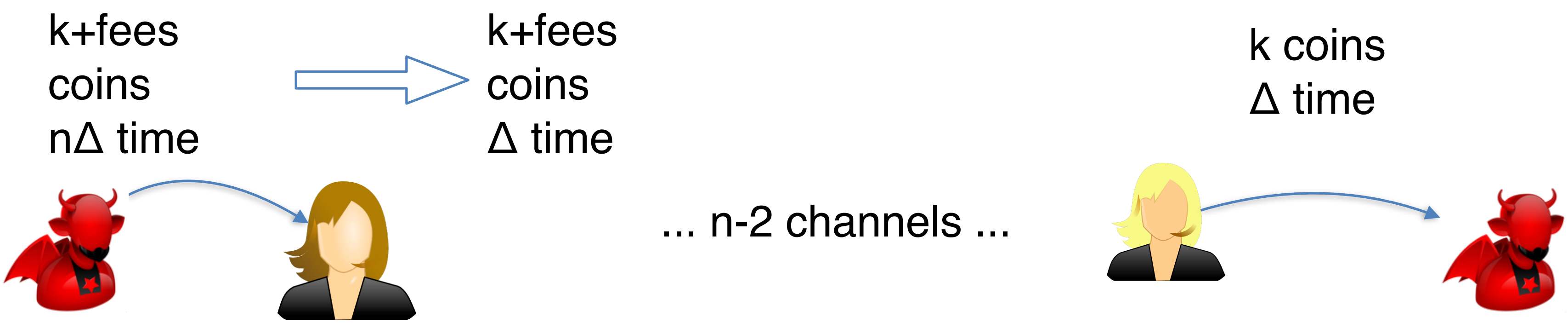
Our Goal: Constant Collateral



- Constant collateral: Coins are locked only for Δ time, independently of the number of channels
- Reduces the amplification factor
- Feasible in Ethereum-based PCNs: Sprites[1]

1 A. Miller et al. Sprites and State Channels: Payment Networks that Go Faster than Lightning.

Our Goal: Constant Collateral

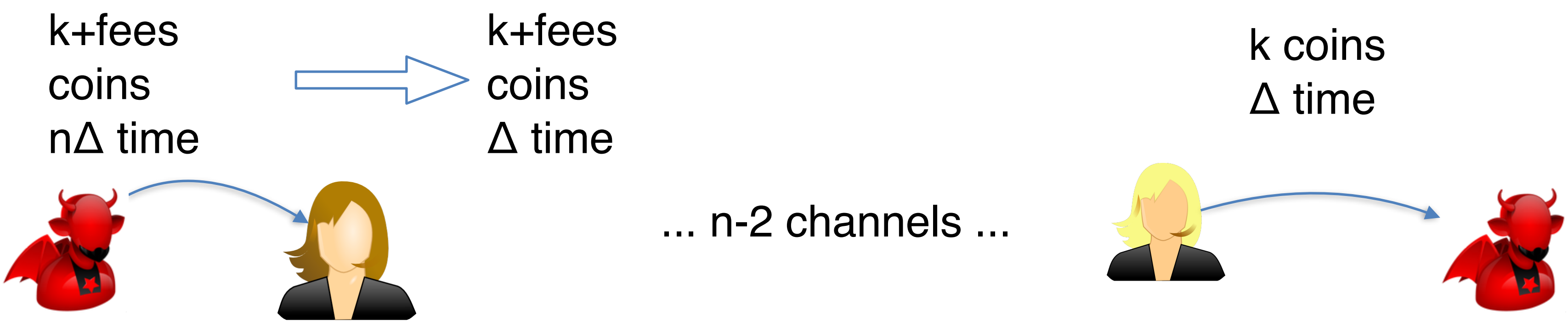


- Constant collateral: Coins are locked only for Δ time, independently of the number of channels
- Reduces the amplification factor
- Feasible in Ethereum-based PCNs: Sprites[1]

a) Feasibility of constant locktimes in Bitcoin: Our constant locktimes construction relies on a global contract mechanism, which is easily expressed in Ethereum, but cannot (we conjecture) be emulated in Bitcoin without some modification to its scripting system. Are there minimal modifications to Bitcoin script that would enable constant locktimes?

1 A. Miller et al. Sprites and State Channels: Payment Networks that Go Faster than Lightning.

Our Goal: Constant Collateral



- Constant collateral: Coins are locked only for Δ time, independently of the number of channels
- Reduces the amplification factor
- Feasible in Ethereum-based PCNs: Sprites[1]

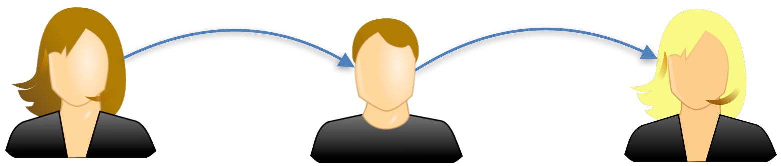
a) Feasibility of constant locktimes in Bitcoin: Our constant locktimes construction relies on a global contract mechanism, which is easily expressed in Ethereum, but cannot (we conjecture) be emulated in Bitcoin without some modification to its scripting system. Are there minimal modifications to Bitcoin script that would enable constant locktimes?

AMCU: Constant collateral and backwards compatible with Bitcoin script

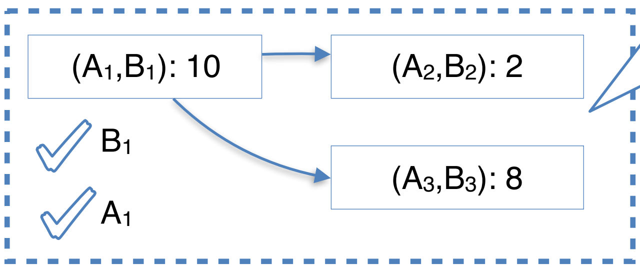
[1] A. Miller et al. Sprites and State Channels: Payment Networks that Go Faster than Lightning.

Atomic Multi-Channel Updates (ACMU)

8 (out of 10)   7 (out of 30)



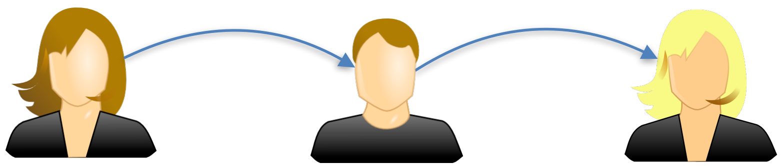
Phase 1 (Setup for A,B)



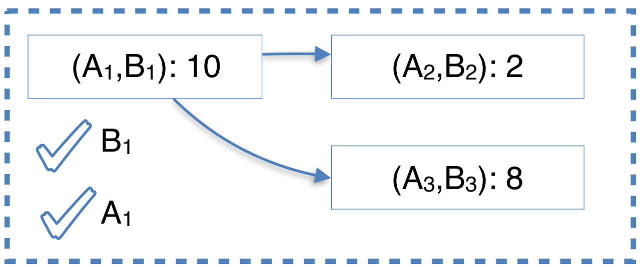
Split the channel so that 2 coins are still available

Atomic Multi-Channel Updates (ACMU)

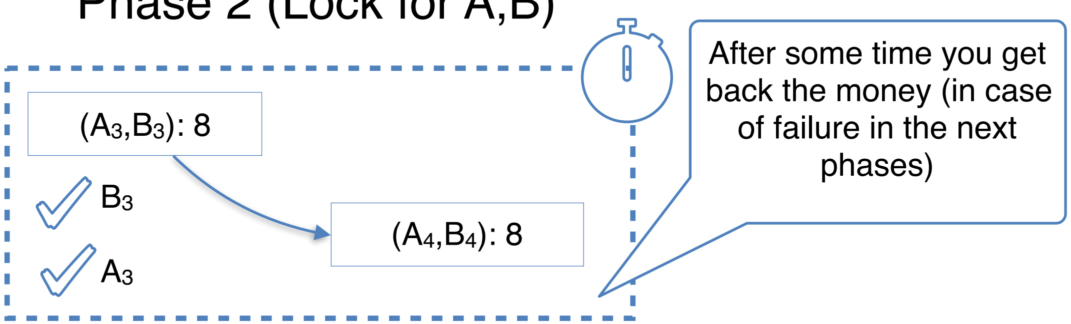
8 (out of 10) 7 (out of 30)



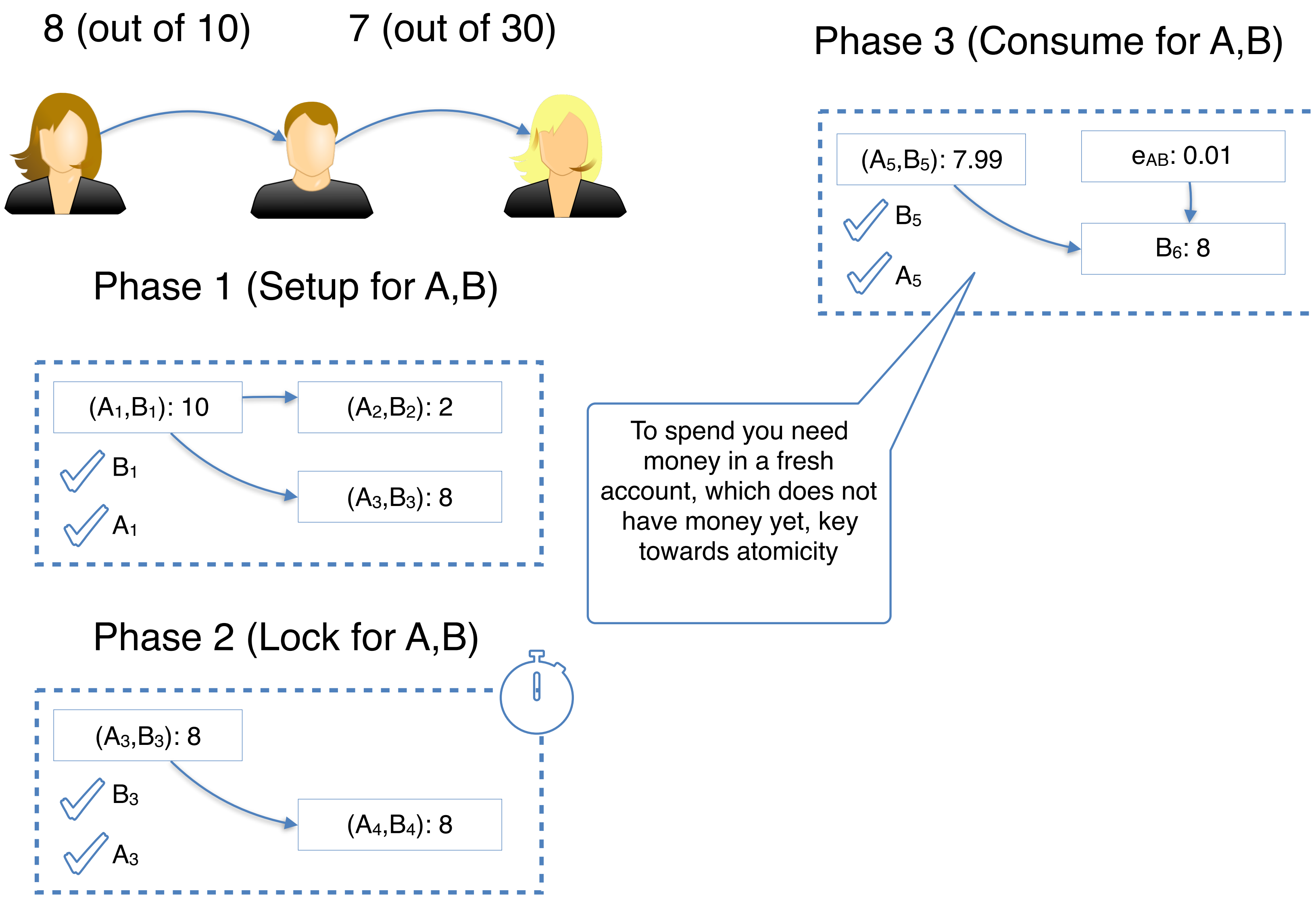
Phase 1 (Setup for A,B)



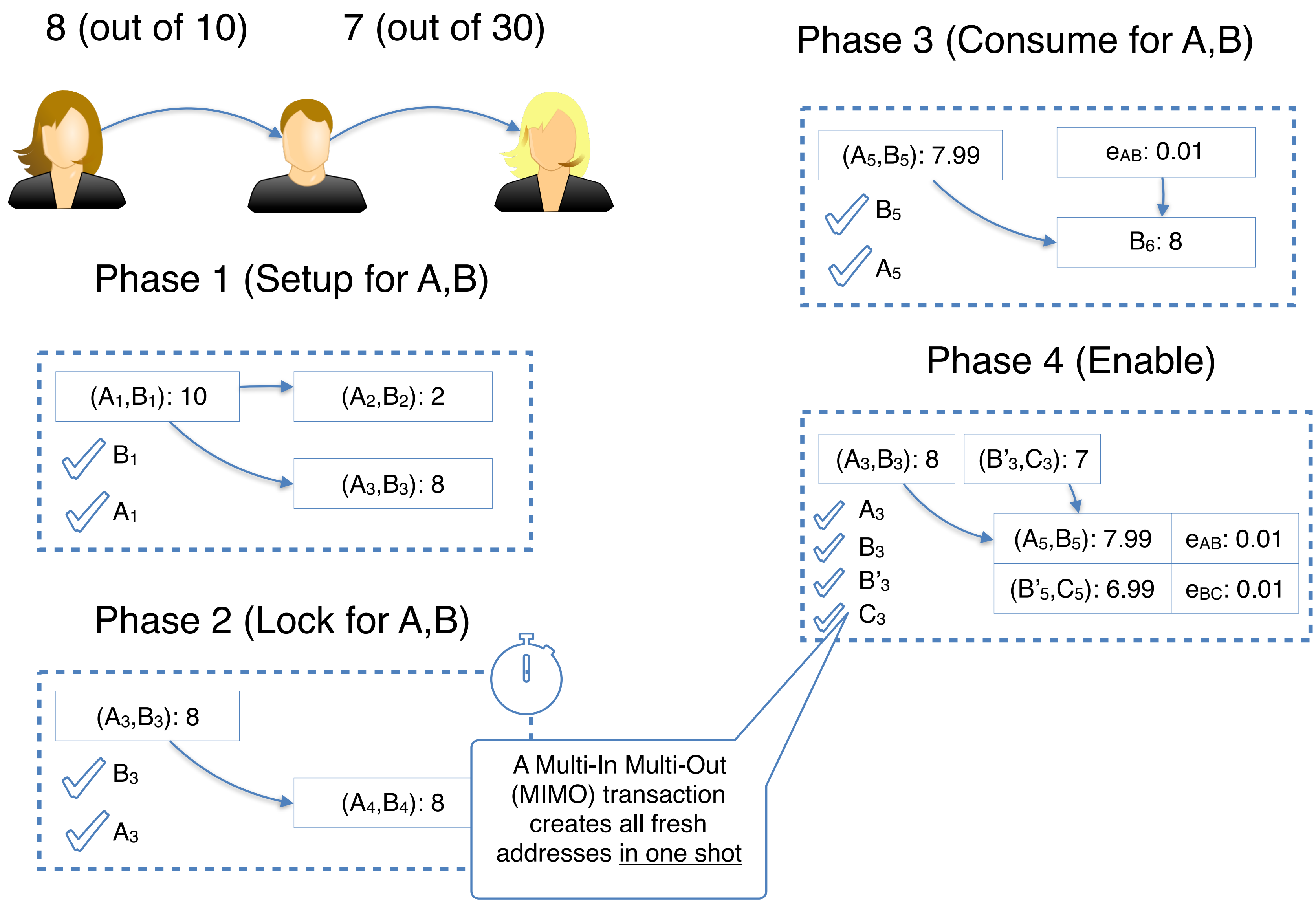
Phase 2 (Lock for A,B)



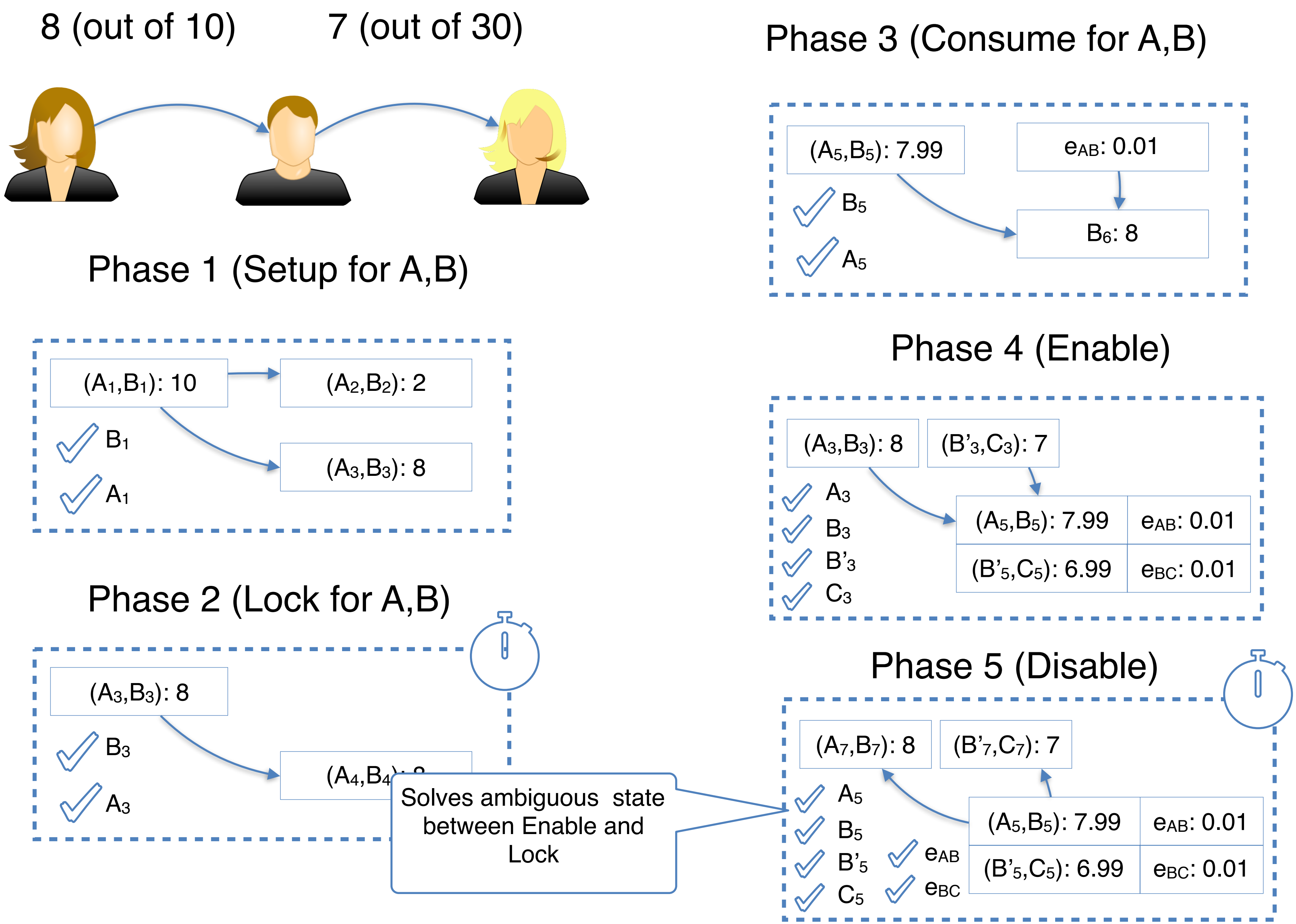
Atomic Multi-Channel Updates (ACMU)



Atomic Multi-Channel Updates (ACMU)



Atomic Multi-Channel Updates (ACMU)



Take Home

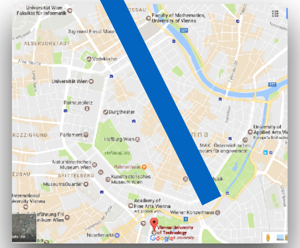
- We can perform off-chain transactions without decreasing security, supporting
  - cross-chain payments
  - synchronisation across arbitrary channels

Take Home

- We can perform off-chain transactions without decreasing security, supporting
  - cross-chain payments
  - synchronisation across arbitrary channels

- We are currently working on
  - Virtual channels to support offline intermediary nodes
  - Payment channel hubs for enhancing connectivity
  - Routing protocols for enhancing resilience
  - as well as on several other blockchain-related topics, like automated verification of smart contracts

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



Vienna Security and Privacy Research Center

VISP

Founding members

TU WIEN

IST AUSTRIA

universität wien

Collaborating partners

AIT AUSTRIAN INSTITUTE OF TECHNOLOGY TOMORROW TODAY

SBA Research

Numbers

- 7 ERC grants
- >10 professors working on S&P and related fields
- >100 doctoral and postdoctoral researchers

ViSP Research Areas

Pietrazk

Fuchsbauer

Cryptography

Lindorfer

Weippl

System Security

Schmid

Zseby

Network Security

Maffei

Kovacs

Henzinger

S&P Verification

Shafique

S&P in Machine Learning

Bartocci

IoT/CPS Security

Kastner

Hardware Security