

Blockstream



Blockstream RESEARCH

Mit Null-Wissen zum besseren Bitcoin

Jonas Nick
nickler.ninja
@n1ckler

2023-09-16

₿TC23



Das ist ein sehr interessantes Thema. Wenn eine Lösung [für den Einsatz von Null-Wissen Beweisen] gefunden wird, wäre eine viel bessere, einfachere, bequemere Implementierung von Bitcoin möglich.

- Satoshi



Anwendung benötigt keine Änderung des Bitcoin Protokolls

 Anwendung benötigt keine Änderung des Bitcoin Protokolls

 Anwendung benötigt Änderung

Anwendung benötigt keine Änderung des Bitcoin Protokolls

Anwendung benötigt Änderung





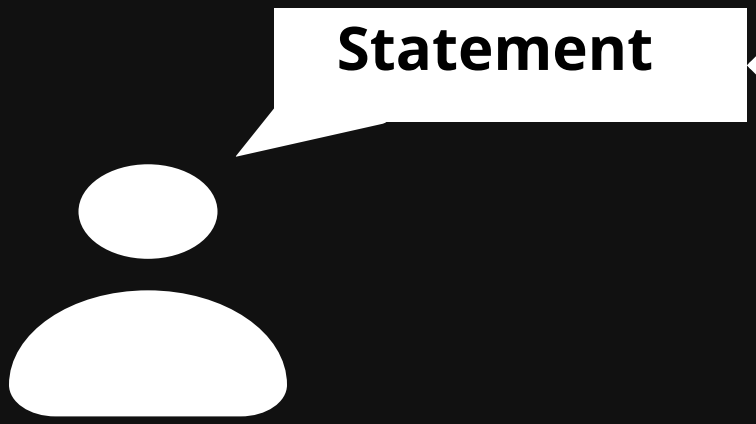






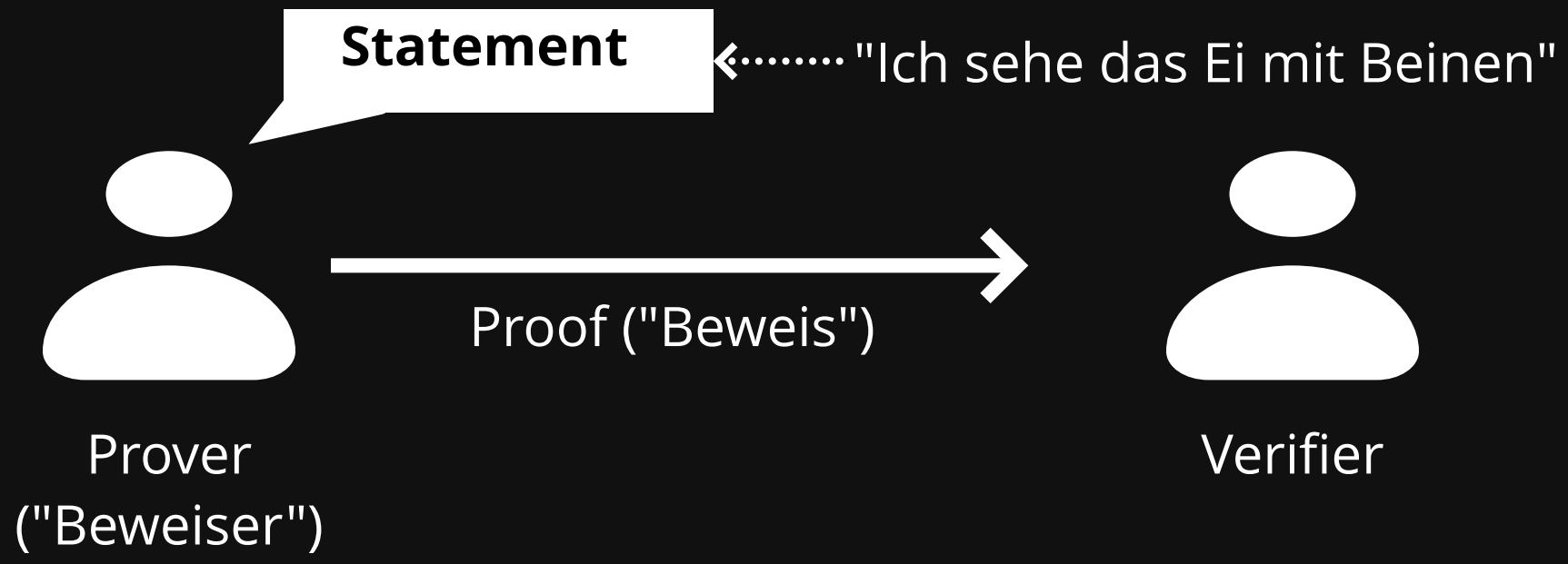
Übersicht

Übersicht

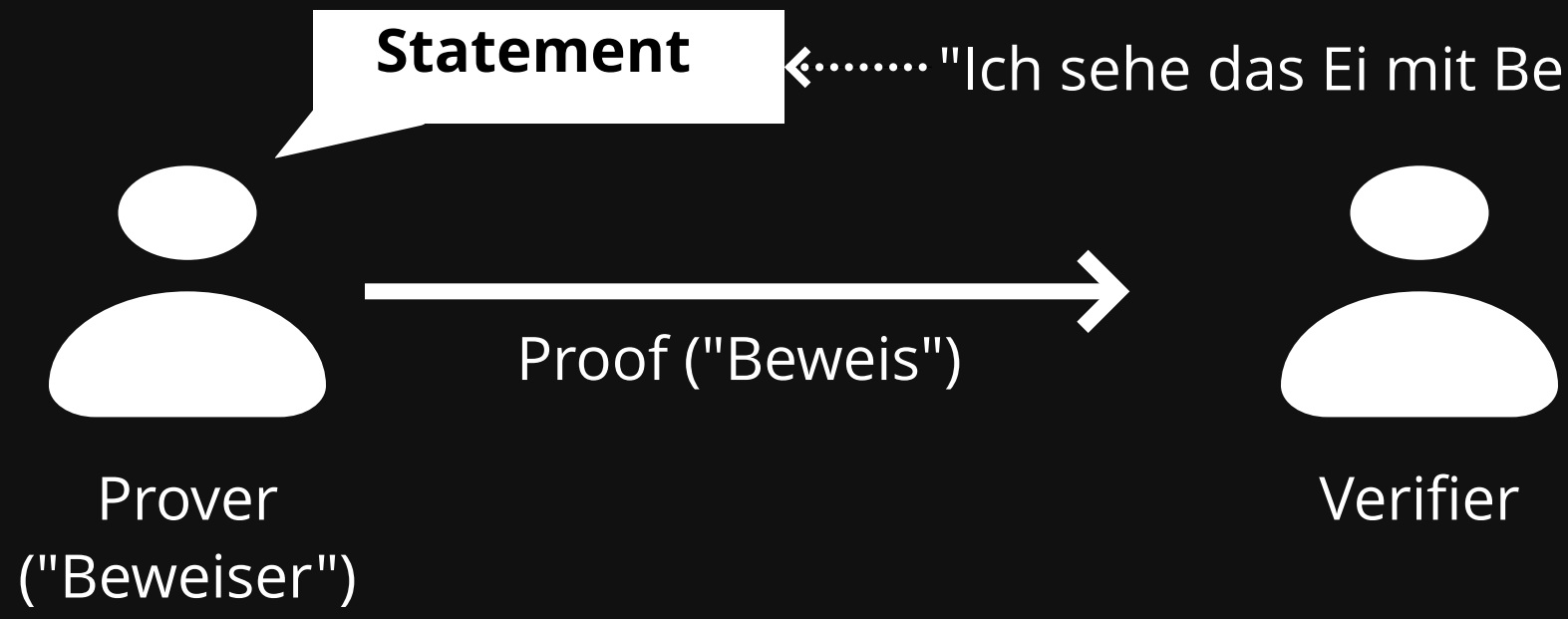


Prover
("Beweiser")

Übersicht


Statement
"Ich sehe das Ei mit Beinen"
Proof ("Beweis")
Prover
("Beweiser")
Verifier

Übersicht



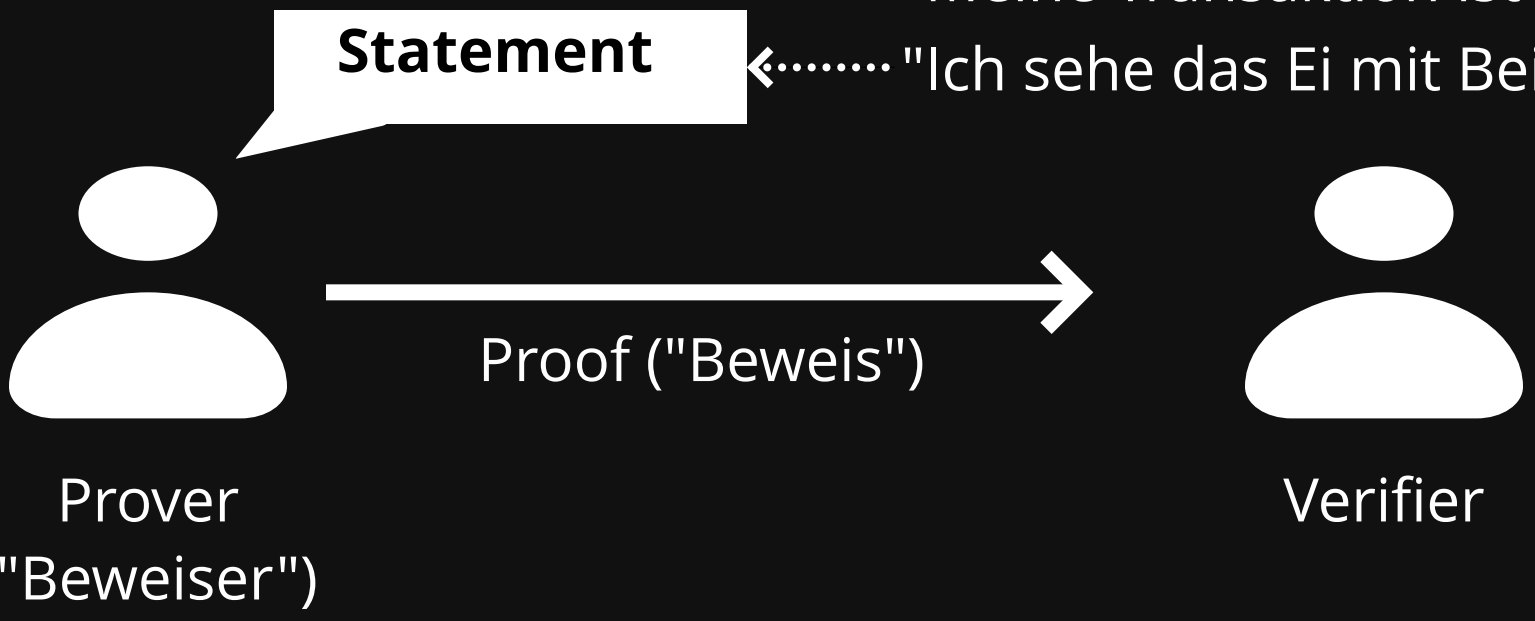
Zero Knowledge (ZK), "Null-Wissen"

Der Verifier lernt nichts Neues, außer dass das Statement wahr ist.

→ bessere Privacy in Bitcoin

Übersicht

"Ich besitze 1 Bitcoin"
"Meine Transaktion ist valide"
"Ich sehe das Ei mit Beinen"



Zero Knowledge (ZK), "Null-Wissen"

Der Verifier lernt nichts Neues, außer dass das Statement wahr ist.

→ bessere Privacy in Bitcoin

Signaturen

Signaturen



privater Schlüssel

öffentlicher Schlüssel

Adresse

Signaturen



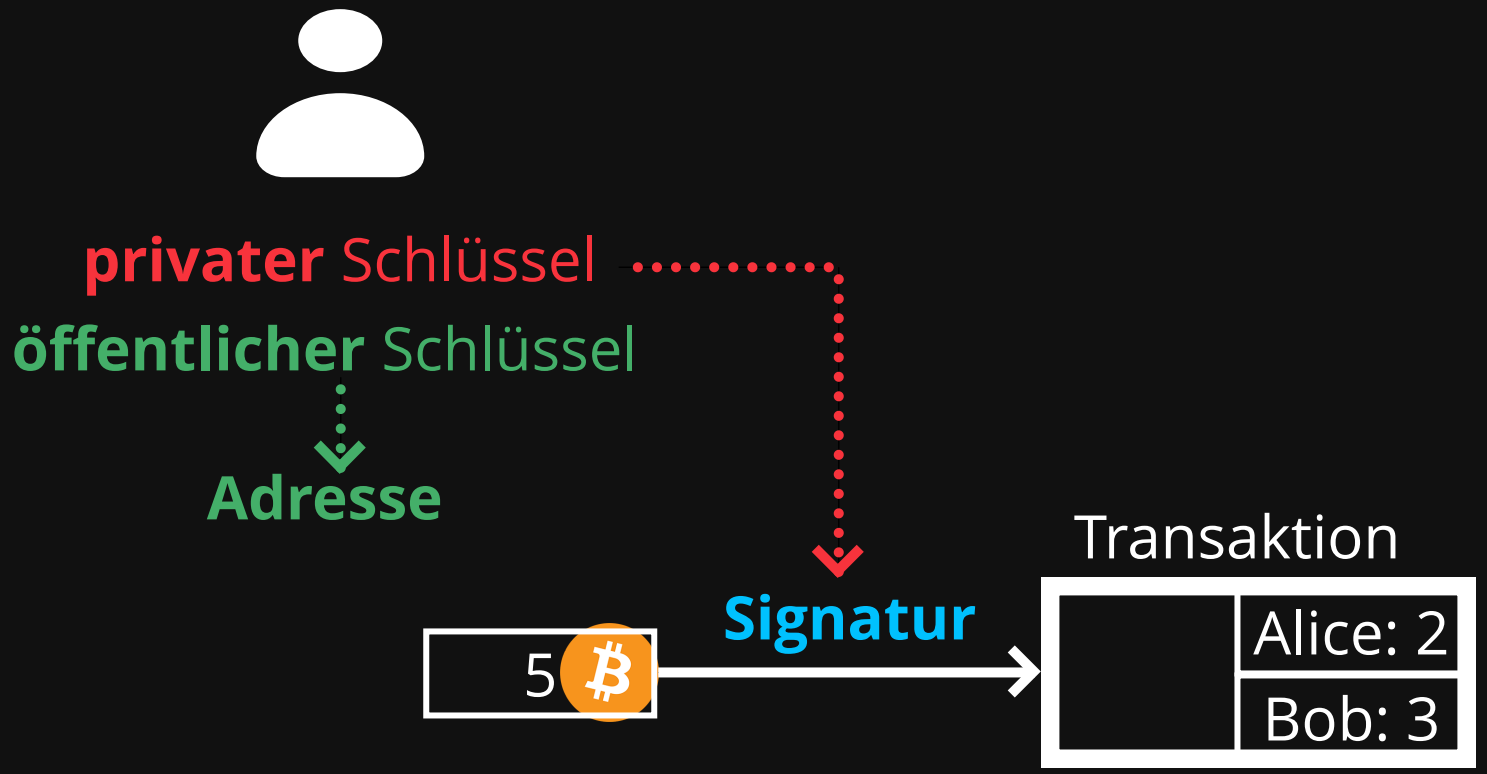
privater Schlüssel

öffentlicher Schlüssel

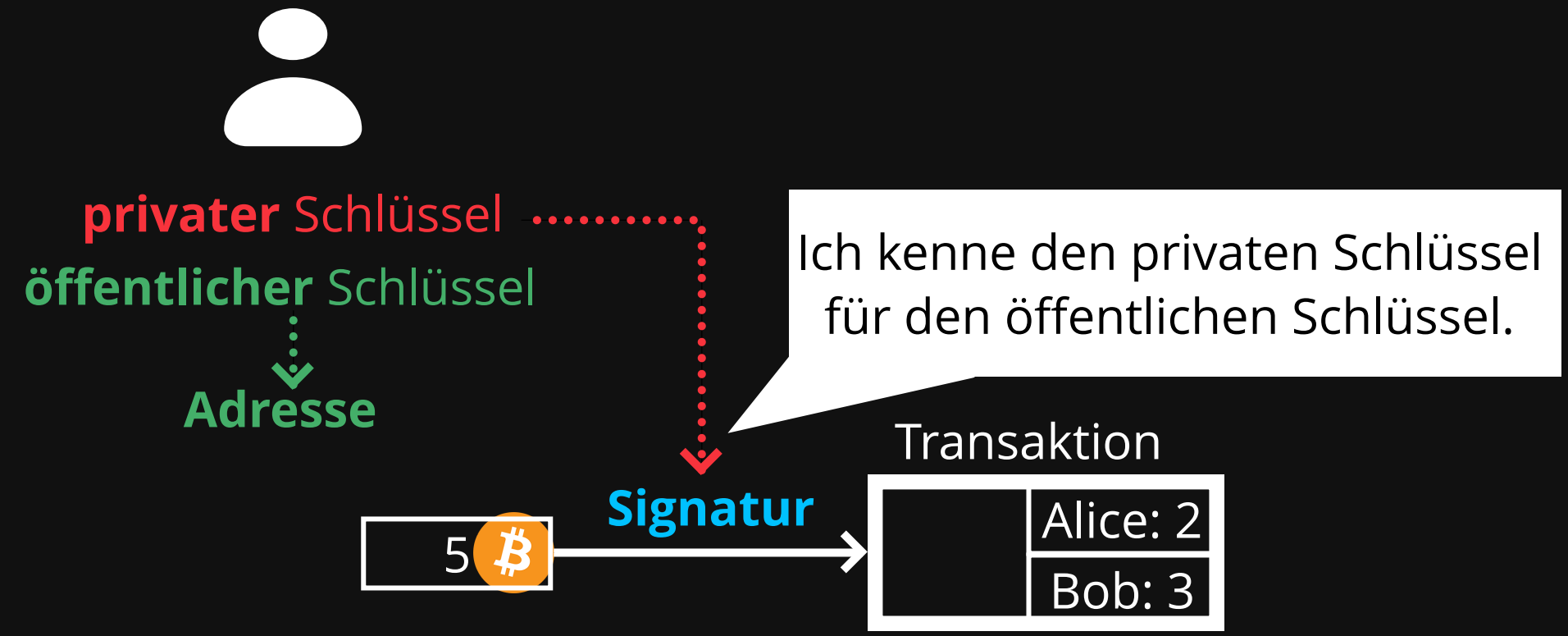
Adresse

5

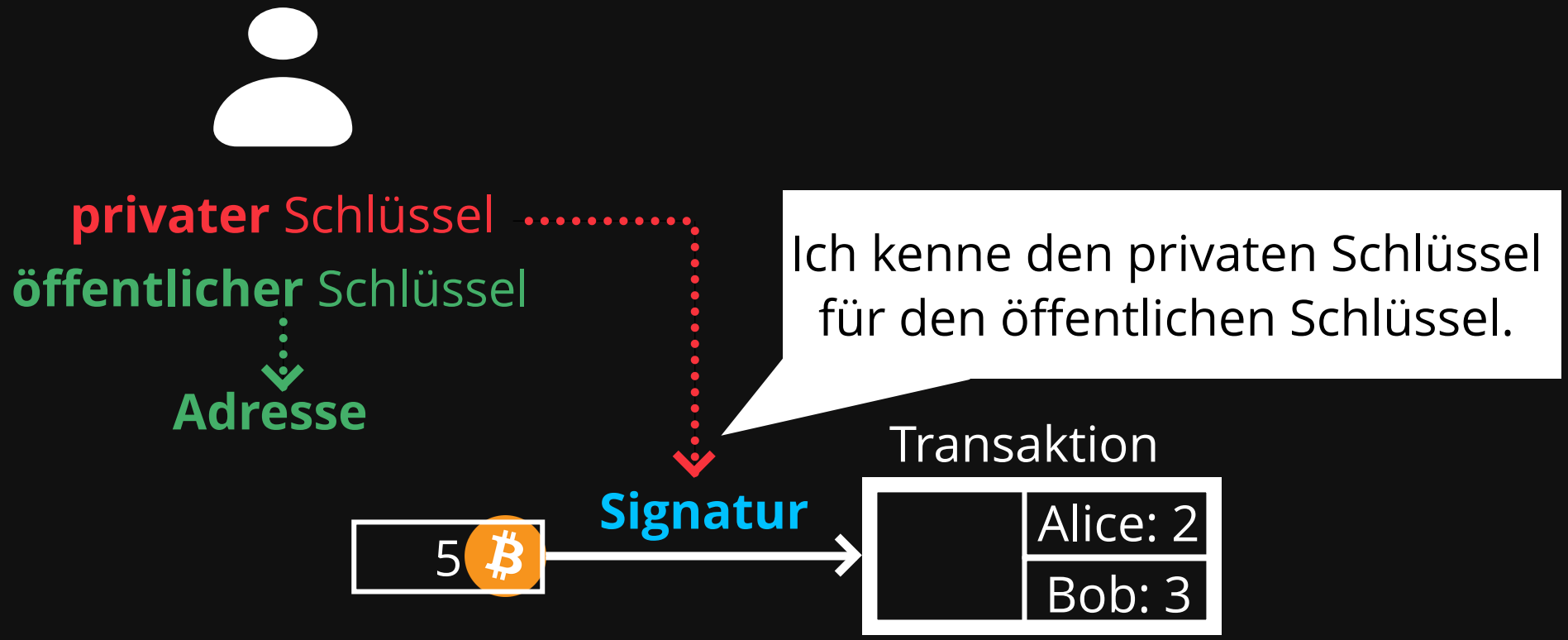
Signaturen


privater Schlüssel
öffentlicher Schlüssel
Adresse
Signatur
5
Transaktion
Alice: 2
Bob: 3

Signaturen


privater Schlüssel
öffentlicher Schlüssel
Adresse
Ich kenne den privaten Schlüssel für den öffentlichen Schlüssel.
Signatur
5
Transaktion
Alice: 2
Bob: 3

Signaturen



- ZK: Signatur gibt keine Information über den privaten Schlüssel preis
- Verifier: Bitcoin Nodes

ZK Beweise über verschlüsselte Werte

ZK Beweise über verschlüsselte Werte

Verschlüsselung: 

ZK Beweise über verschlüsselte Werte

Verschlüsselung: ??? Wert

Ich kenne den Wert in ??? und dieser ist eine Zahl kleiner als 21 Millionen

ZK Beweise über verschlüsselte Werte

Verschlüsselung: ??? Wert

Ich kenne den Wert in ??? und dieser ist eine Zahl kleiner als 21 Millionen

Ist es prinzipiell möglich die Beträge in Transaktionen zu verschlüsseln und ZK-Beweise über deren Korrektheit zu machen?

ZK Beweise über verschlüsselte Werte

Verschlüsselung: ??? Wert

Ich kenne den Wert in ??? und dieser ist eine Zahl kleiner als 21 Millionen

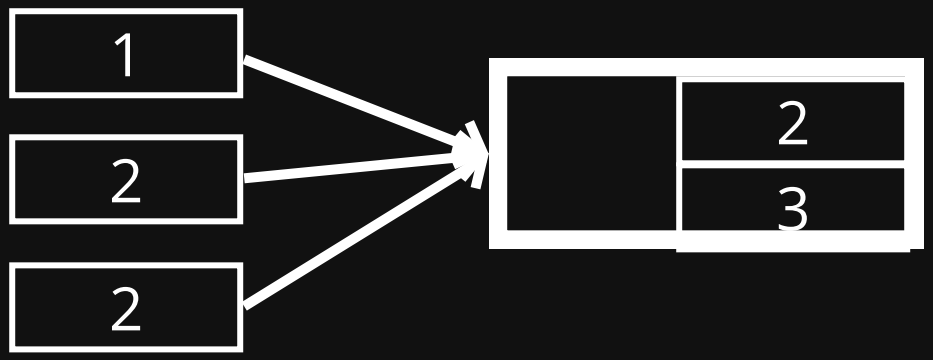
Ist es prinzipiell möglich die Beträge in Transaktionen zu verschlüsseln und ZK-Beweise über deren Korrektheit zu machen?

JA

Confidential Transactions

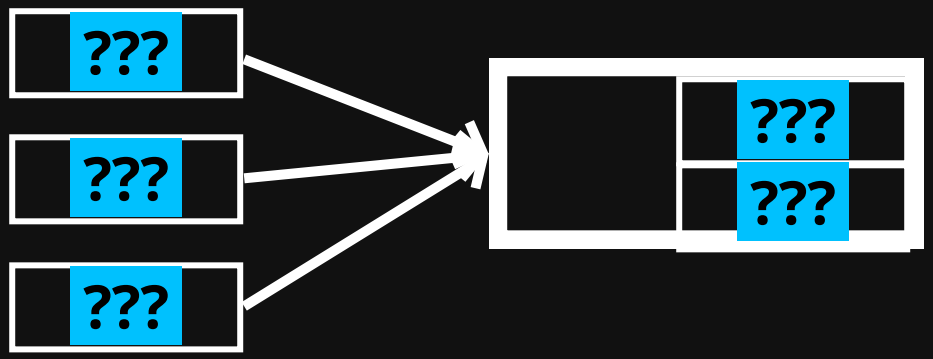


Confidential Transactions

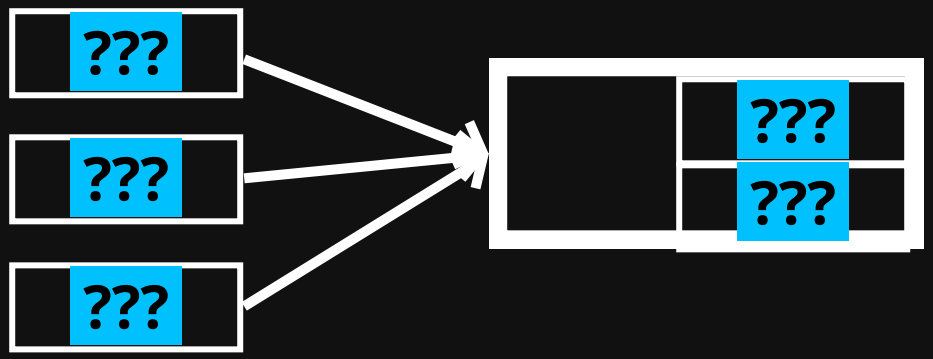


Confidential Transactions

LIQUID NETWORK

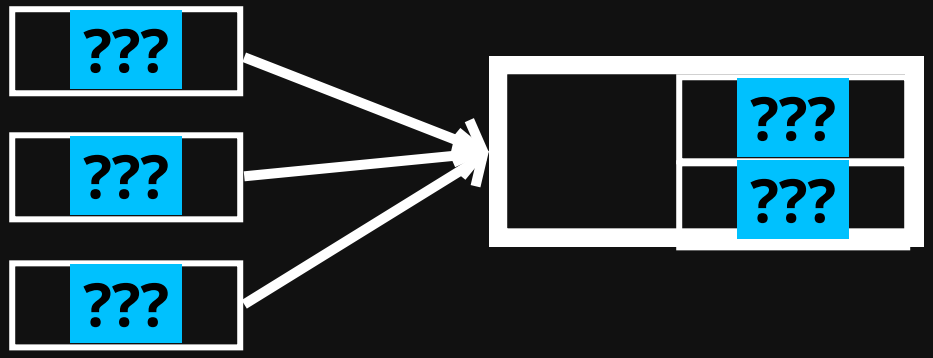


Confidential Transactions



Ich kenne die verschlüsselten Beträge und die Summe der Output Beträge ist gleich der Summe der Input Beträge

Confidential Transactions

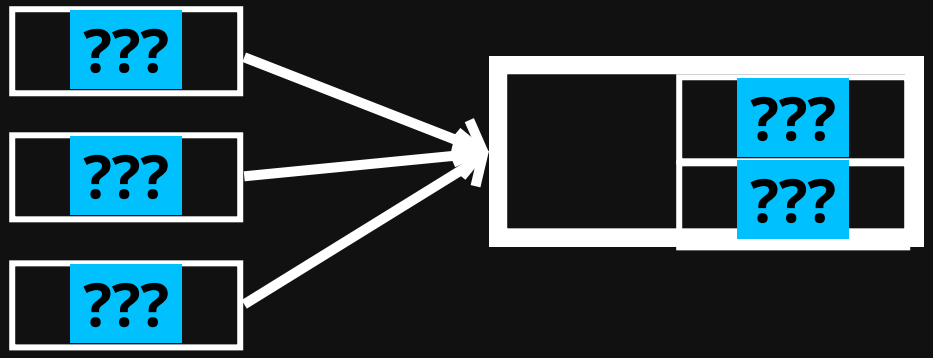


Ich kenne die verschlüsselten Beträge und die Summe der Output Beträge ist gleich der Summe der Input Beträge

Verifier: Liquid Knoten

Confidential Transactions

LIQUID NETWORK



Ich kenne die verschlüsselten Beträge und die Summe der Output Beträge ist gleich der Summe der Input Beträge

Verifier: Liquid Knoten

Bulletproofs++ (2022): Beweis ist etwa 600 Bytes

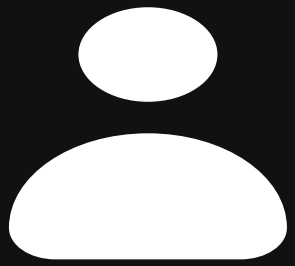
Theorem (1991):

Alle praktisch relevanten Statements haben Zero Knowledge Beweise

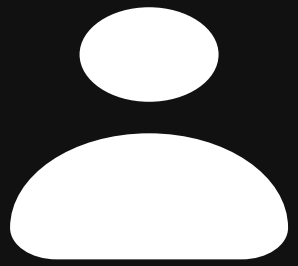
ZKCP

ZKCP

Bsp: Kauf der Lösung eines Sudoku Rätsel
ohne Vertrauen in den Verkäufer



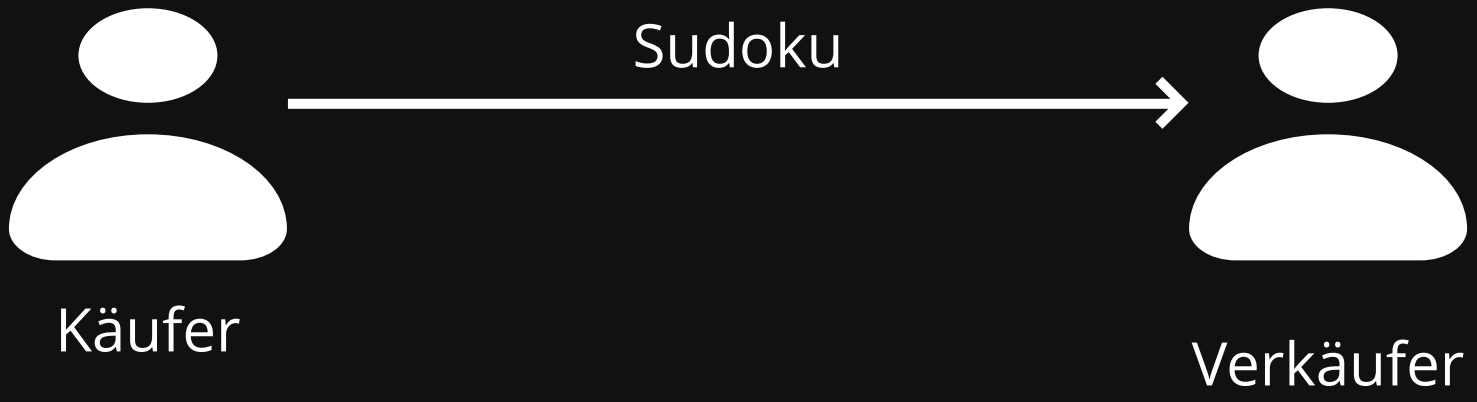
Käufer



Verkäufer

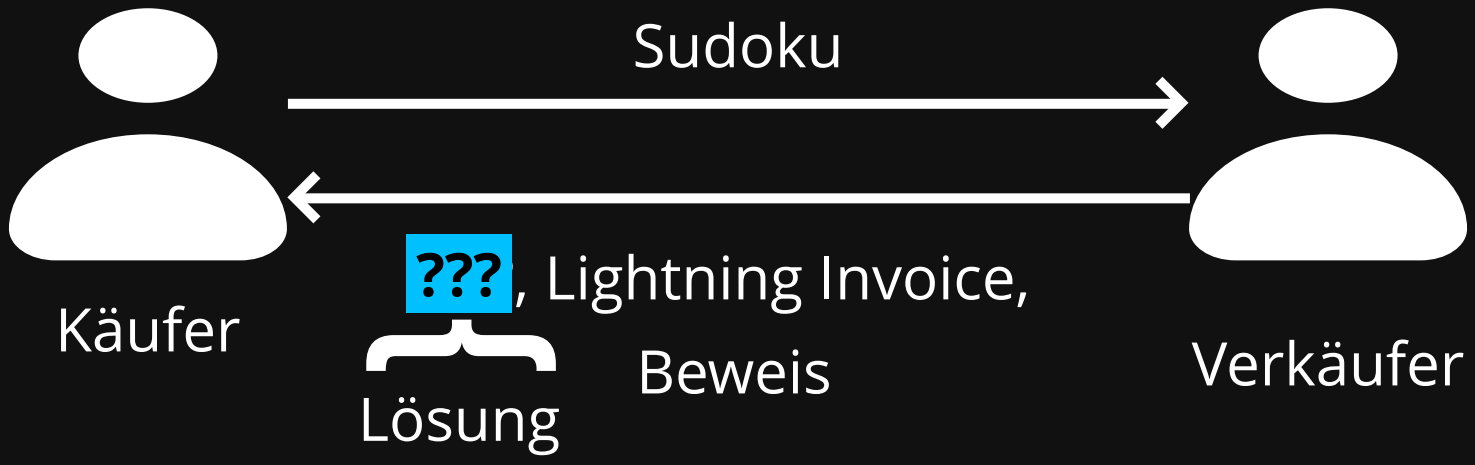
ZKCP

Bsp: Kauf der Lösung eines Sudoku Rätsel
ohne Vertrauen in den Verkäufer



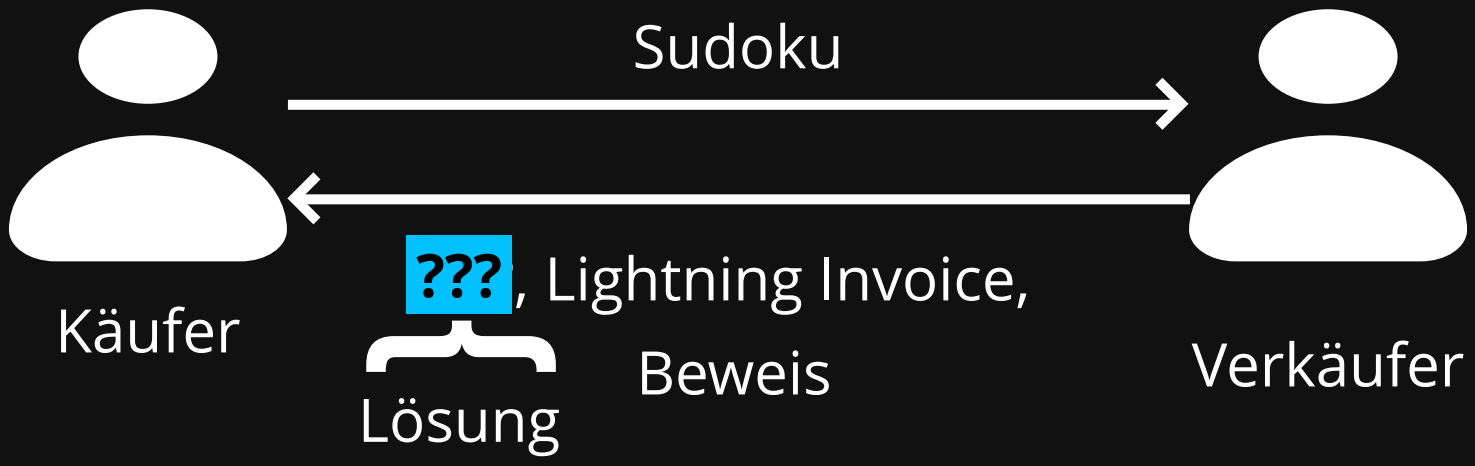
ZKCP

Bsp: Kauf der Lösung eines Sudoku Rätsel
ohne Vertrauen in den Verkäufer



ZKCP

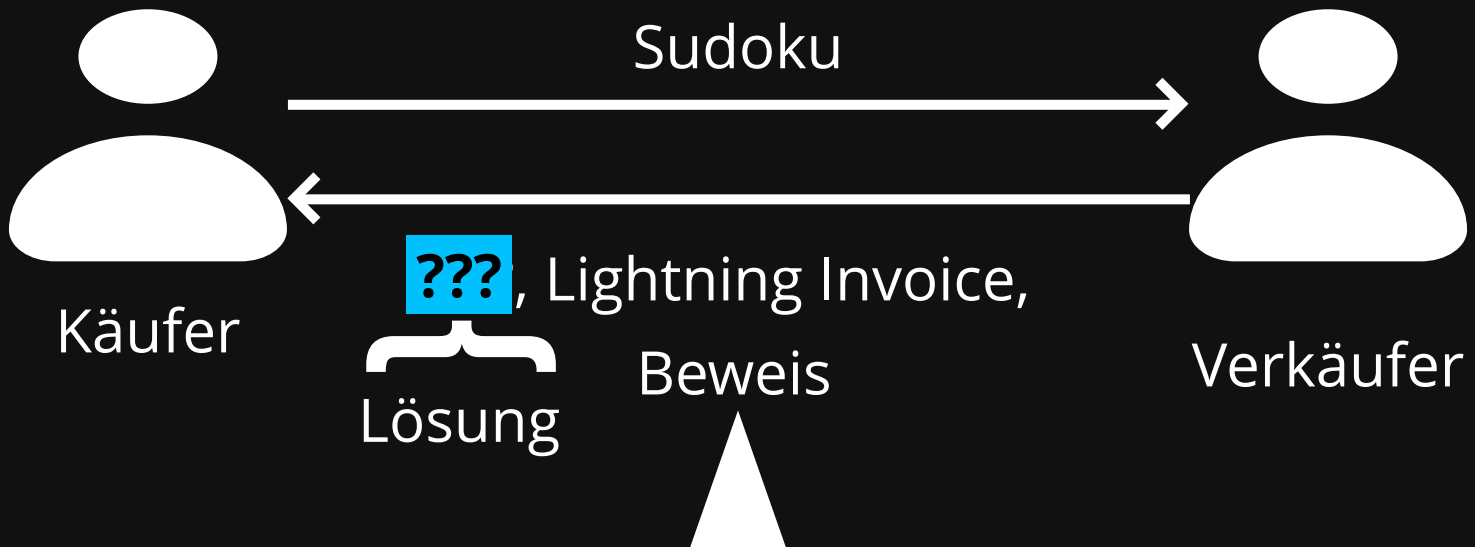
Bsp: Kauf der Lösung eines Sudoku Rätsel
ohne Vertrauen in den Verkäufer



(In Lightning bekommt der Zahlende ein Proof-of-Payment (PoP))

ZKCP

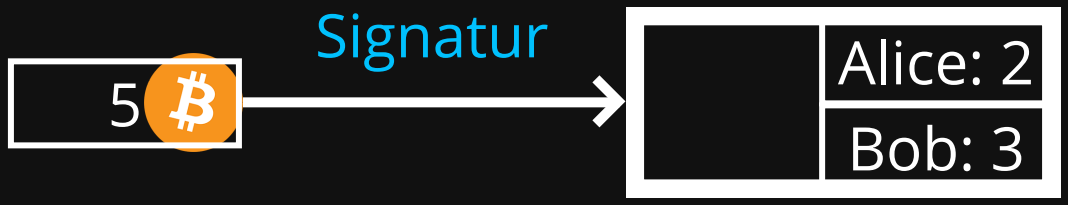
Bsp: Kauf der Lösung eines Sudoku Rätsel
ohne Vertrauen in den Verkäufer



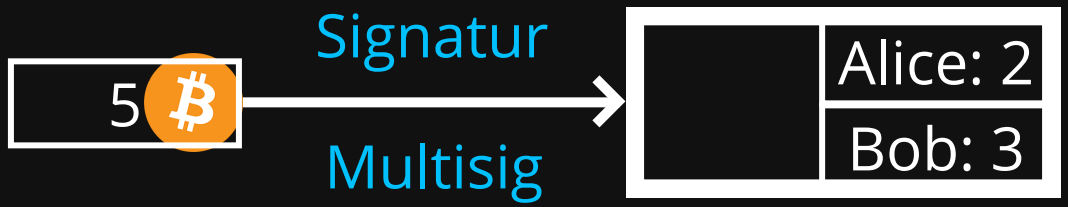
Wenn du dieses Lightning Invoice zahlst, dann erlangst du ein PoP womit du **???** entschlüsseln kannst und das Resultat ist eine Lösung für dein Sudoku.

(In Lightning bekommt der Zahlende ein Proof-of-Payment (PoP))

Bitcoin Script Erweiterung

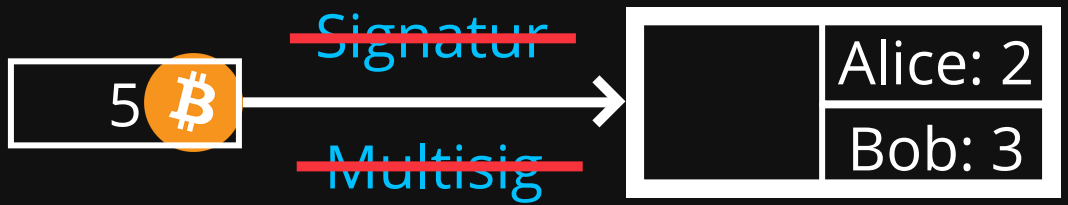


Bitcoin Script Erweiterung



Lightning Contracts

Bitcoin Script Erweiterung😕



~~Lightning Contracts~~

ZKP

Ziel

1. Bessere Privacy

On-Chain Mixer 😕

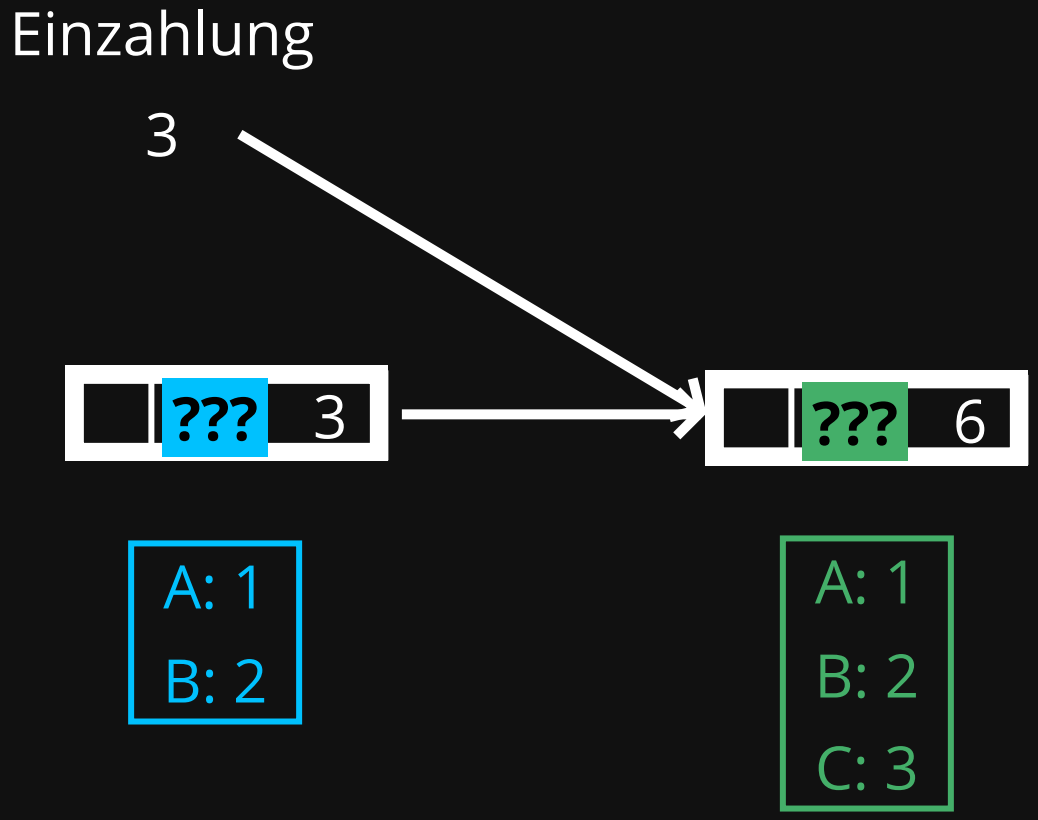
On-Chain Mixer 😕



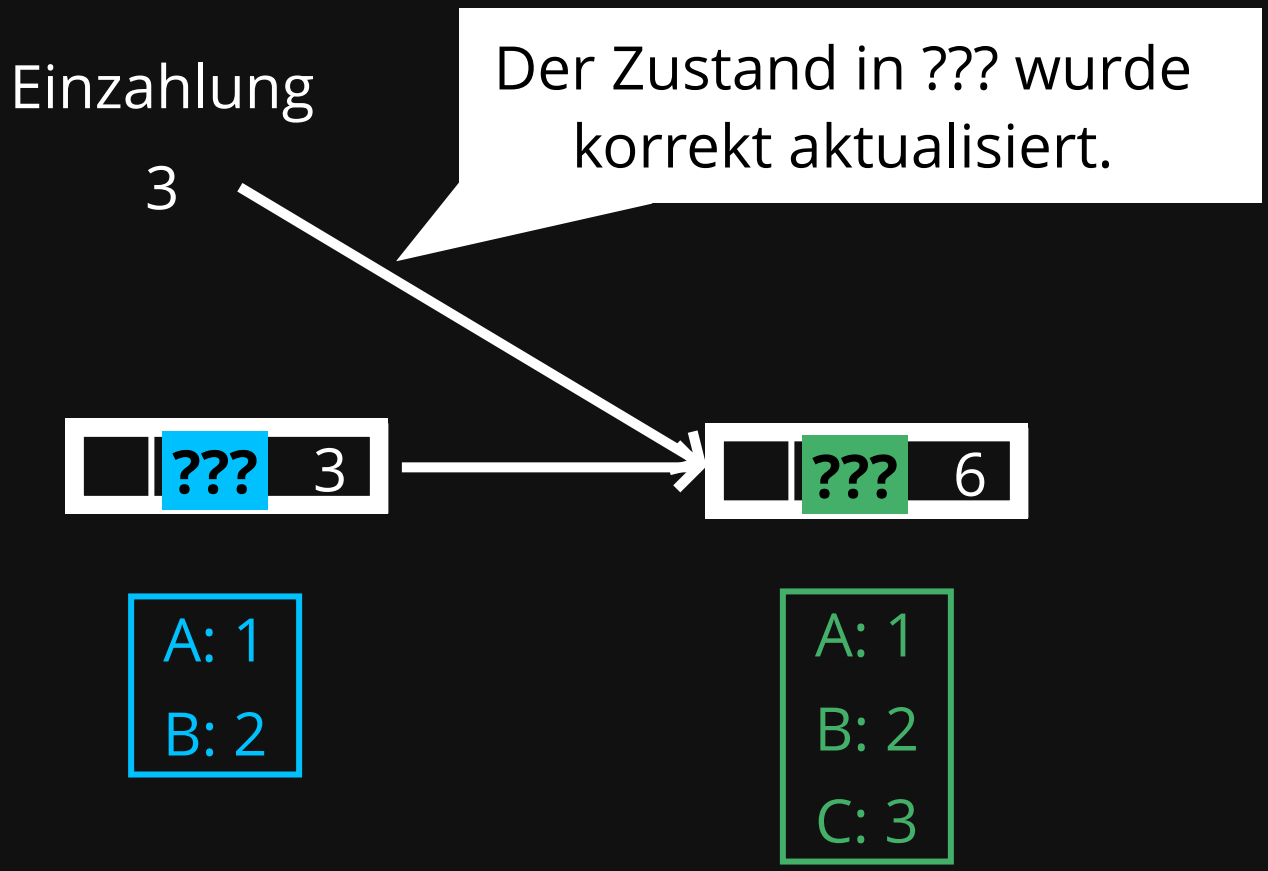
A: 1

B: 2

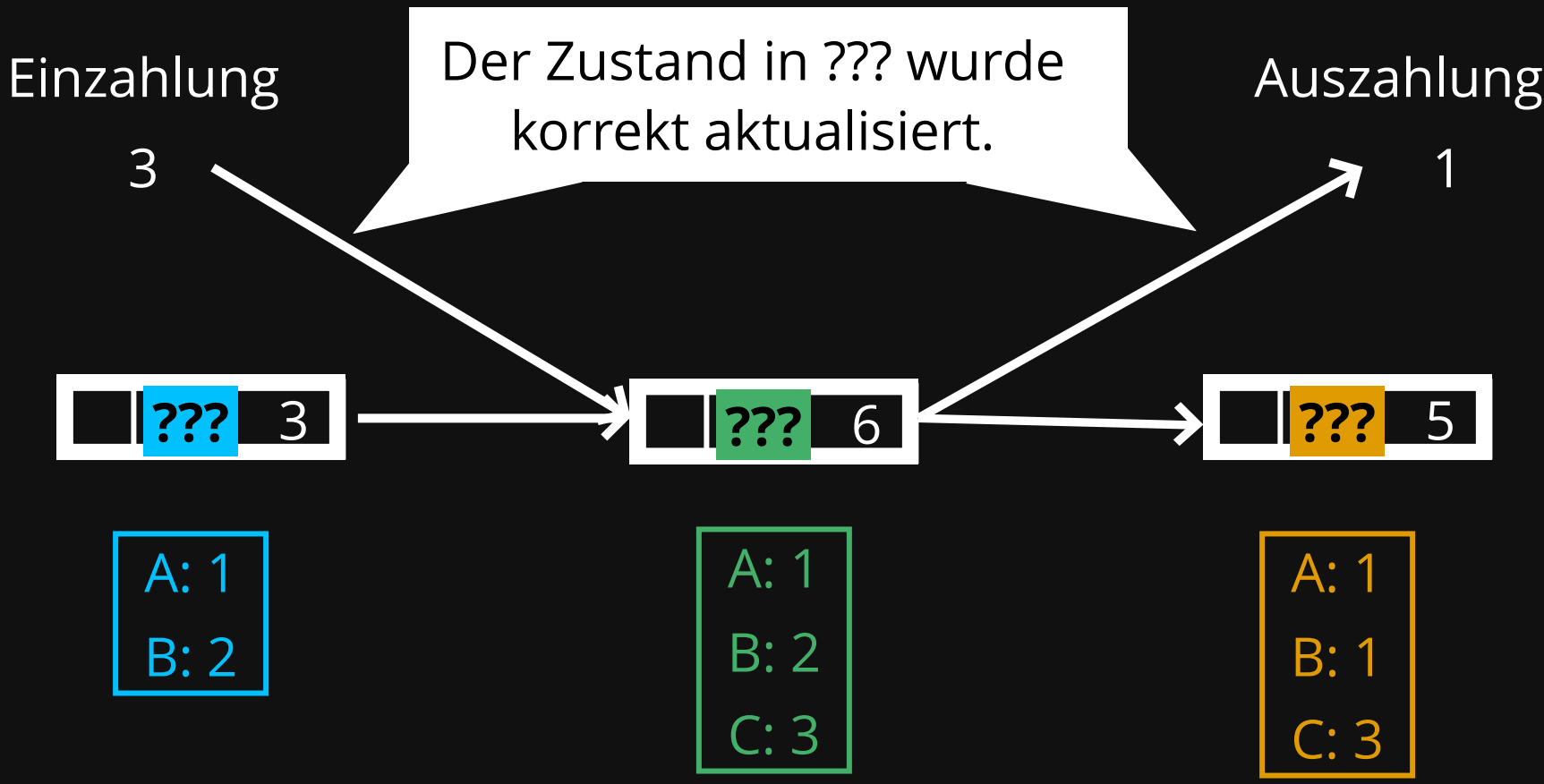
On-Chain Mixer 😕



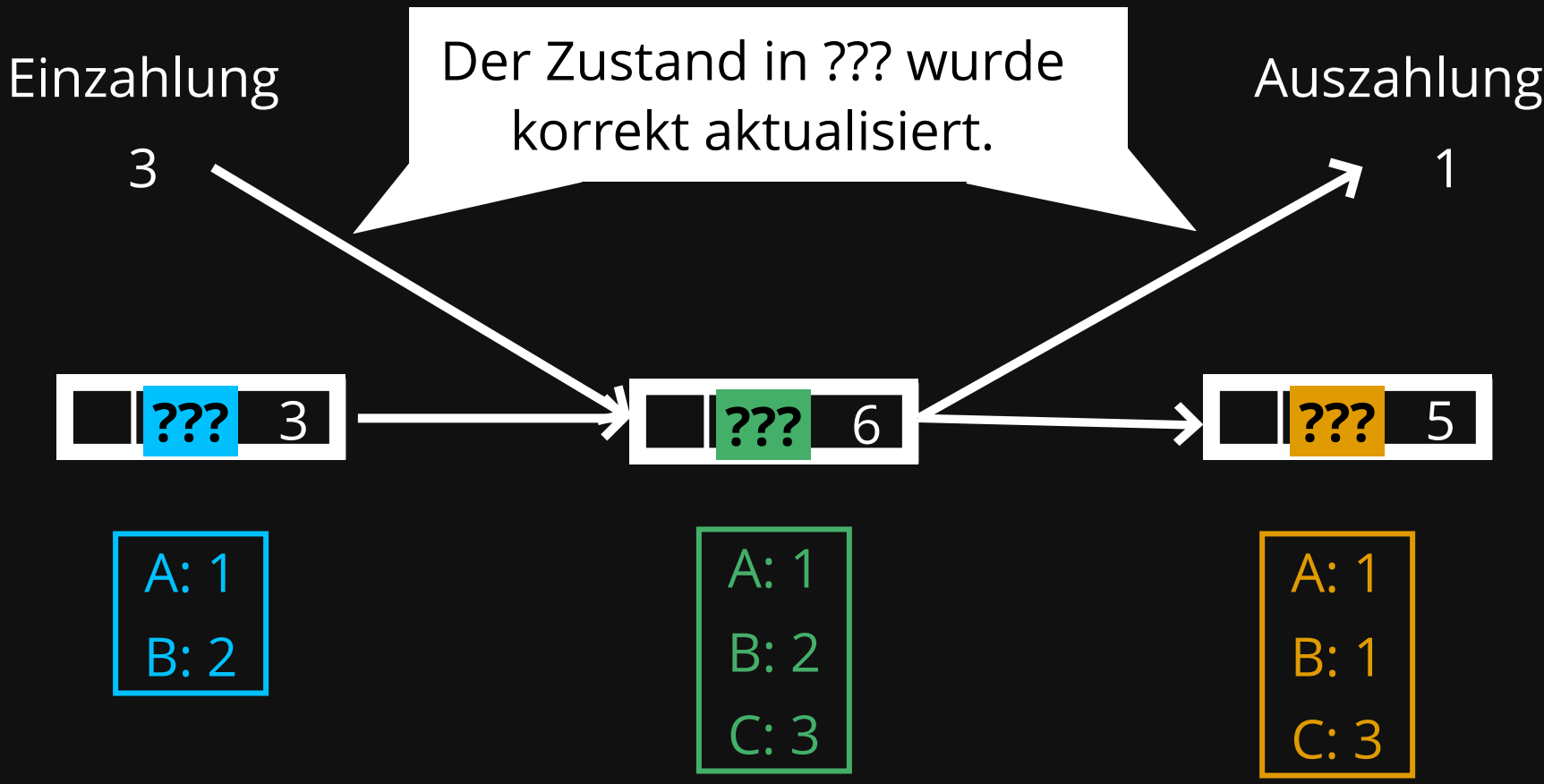
On-Chain Mixer 😕



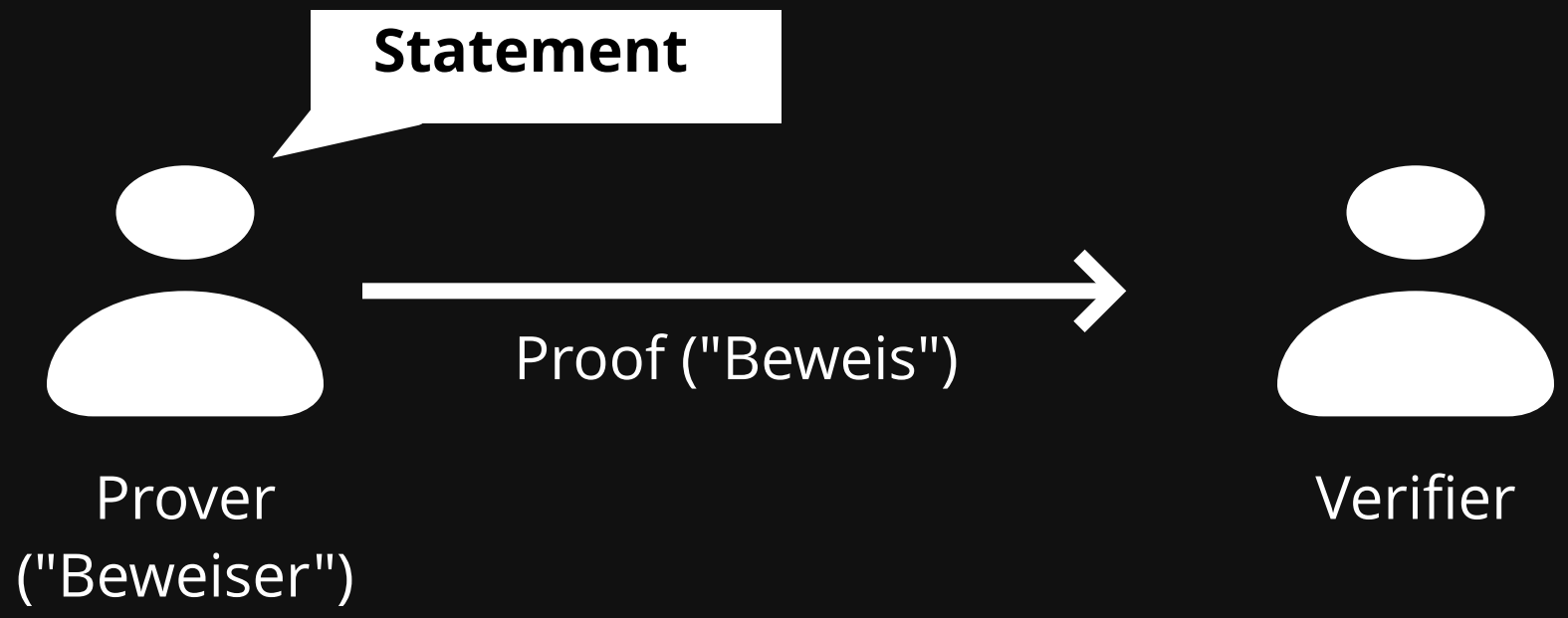
On-Chain Mixer 😕



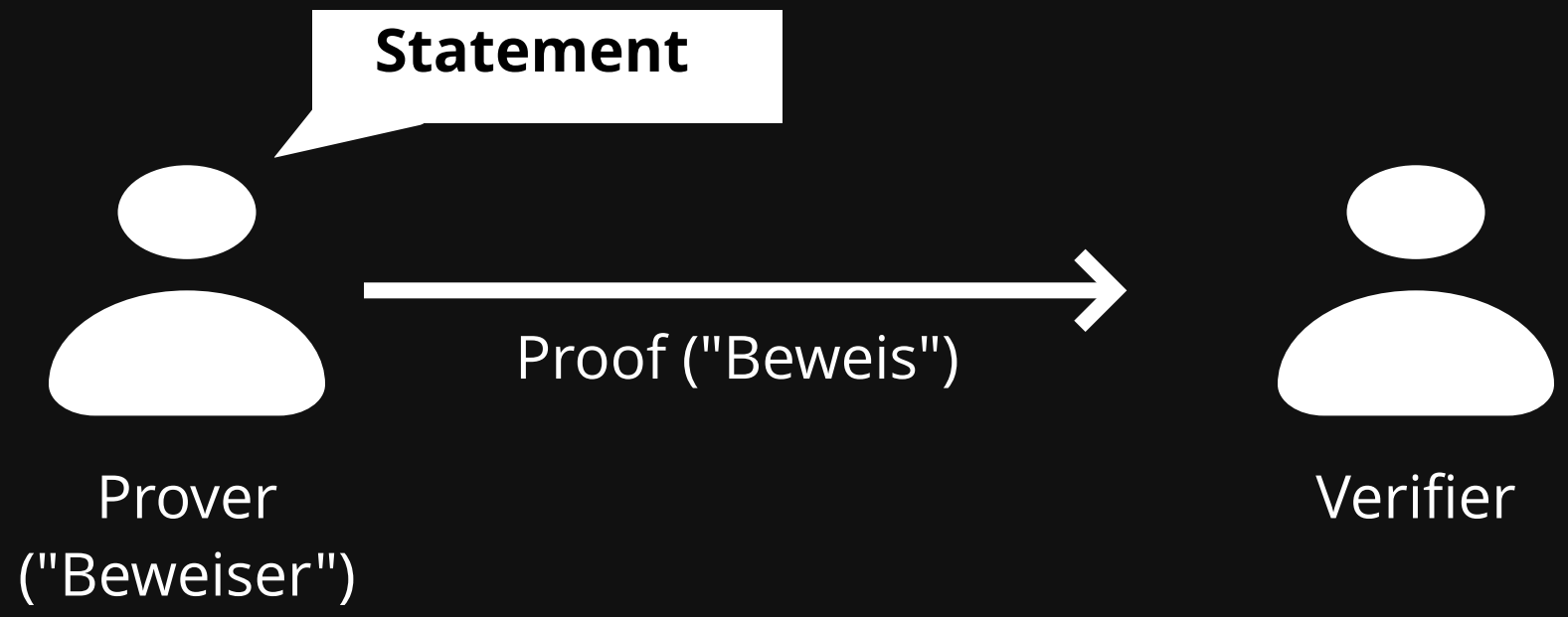
On-Chain Mixer 😕



Zustand verschlüsselt, Zustandsänderung in ZK →
Auszahlung kann nicht mit Einzahlung assoziiert werden.


Statement
Proof ("Beweis")
Prover
("Beweiser")
Verifier

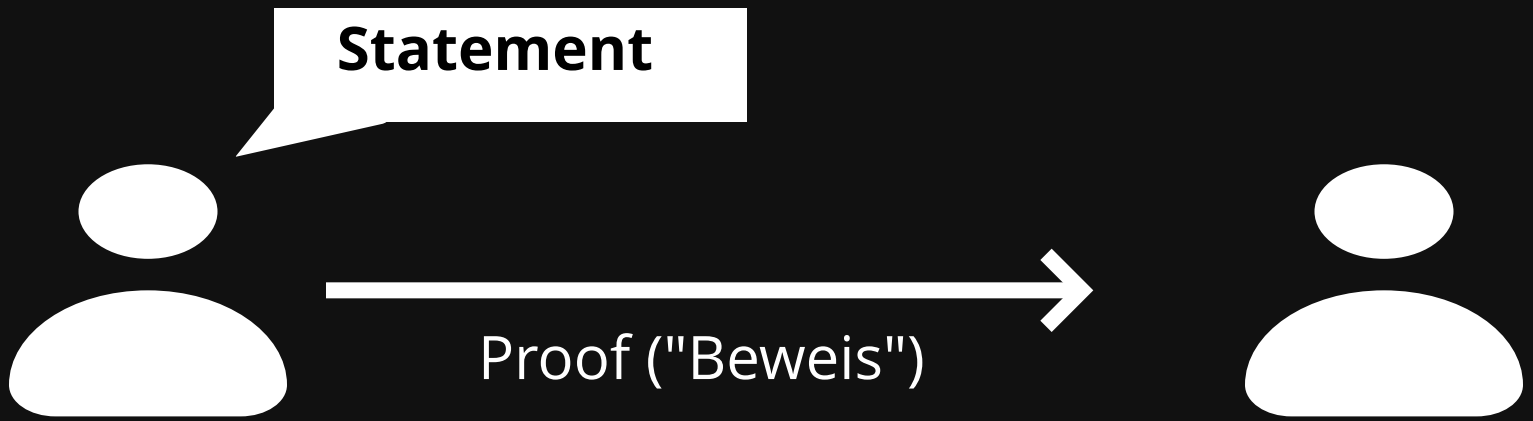
Succinctness



Succinctness, "Kürze"

Der Beweis ist klein und die Verifizierung ist schnell.

Succinctness



Succinctness, "Kürze"

Der Beweis ist klein und die Verifizierung ist schnell.

→ bessere Skalierbarkeit von Bitcoin

Zustandsänderungen

Zustandsänderungen

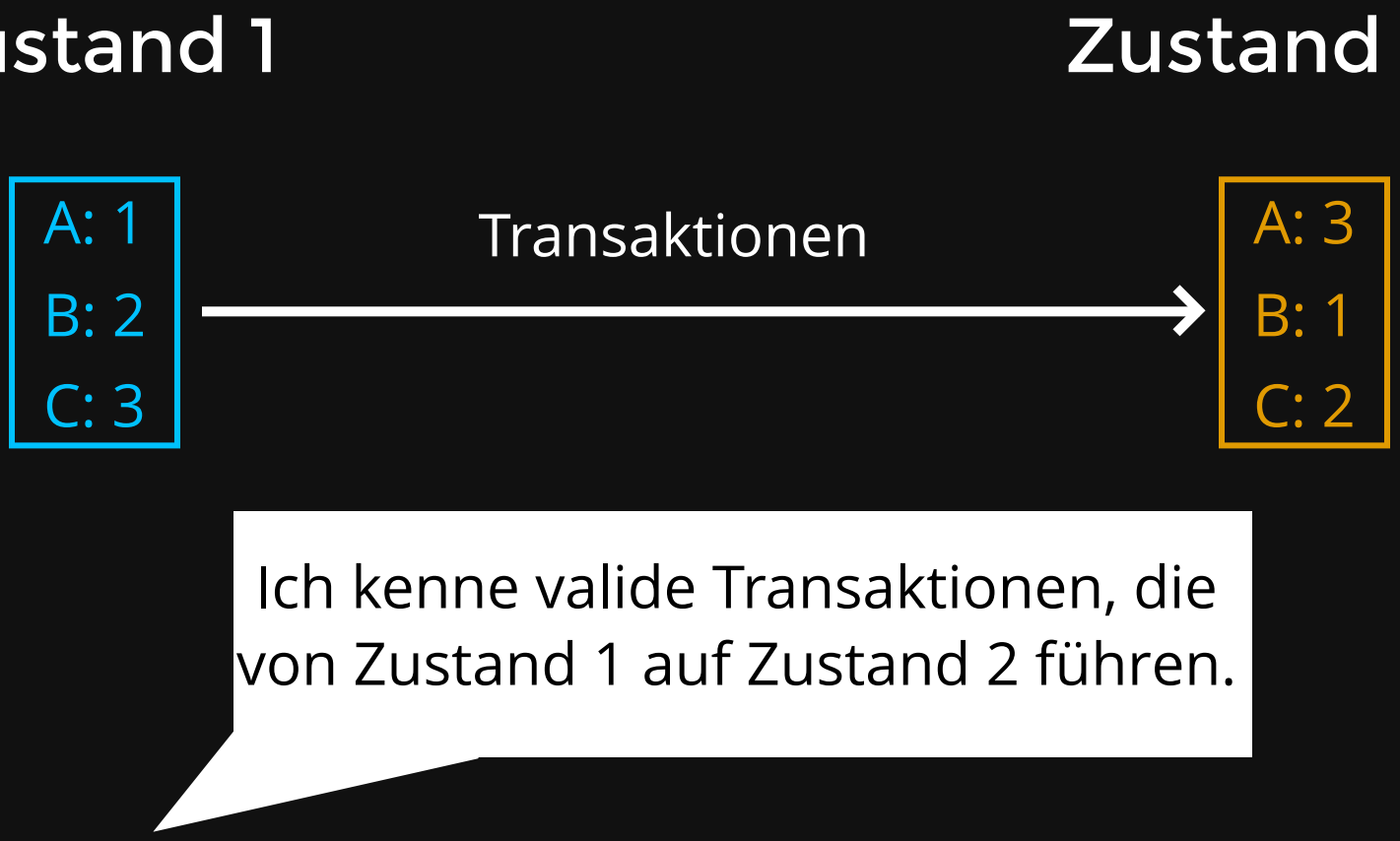
Zustand 1

A: 1
B: 2
C: 3

Zustand 2

A: 3
B: 1
C: 2

Zustandsänderungen



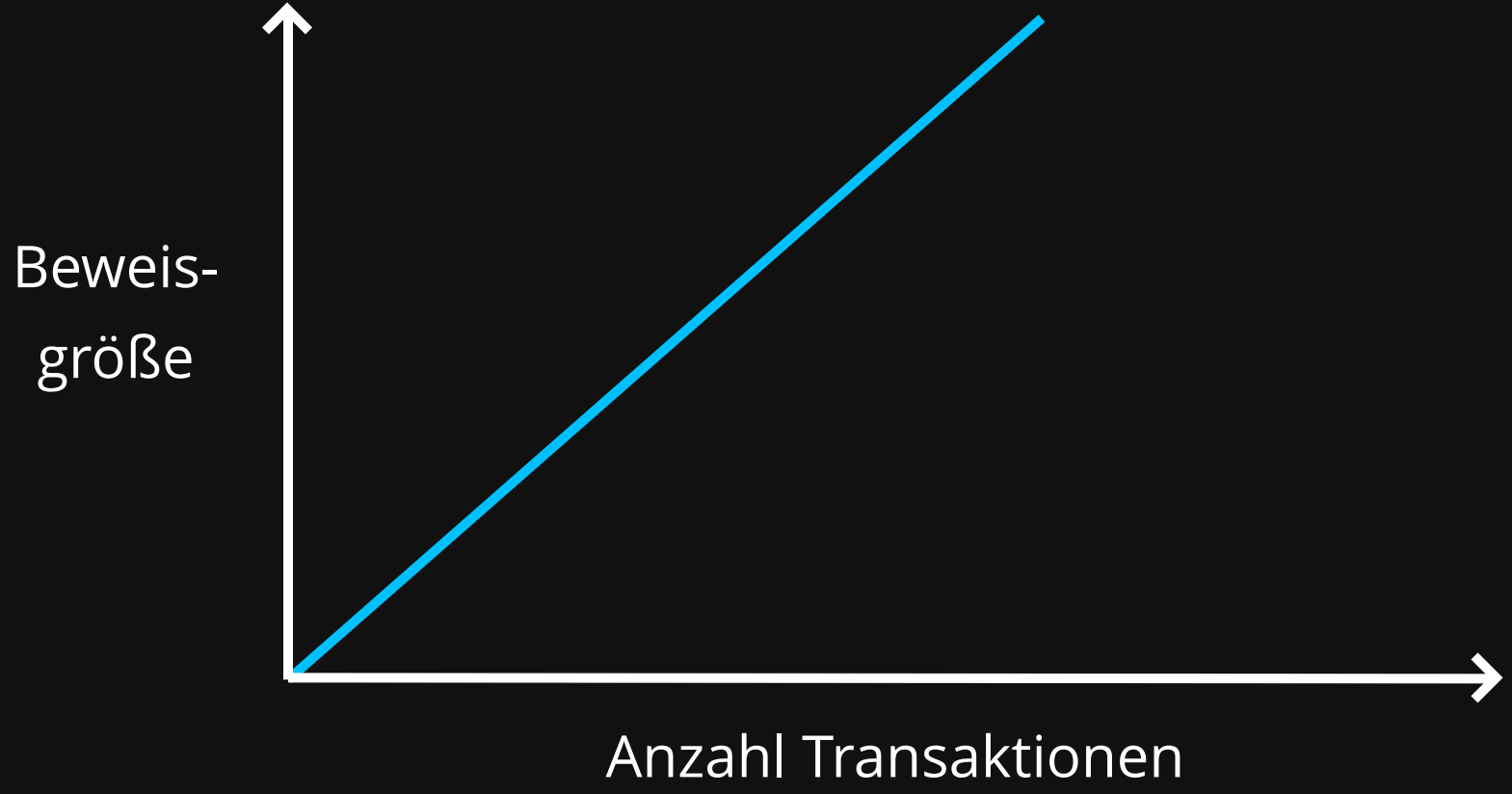
Zustandsänderungen



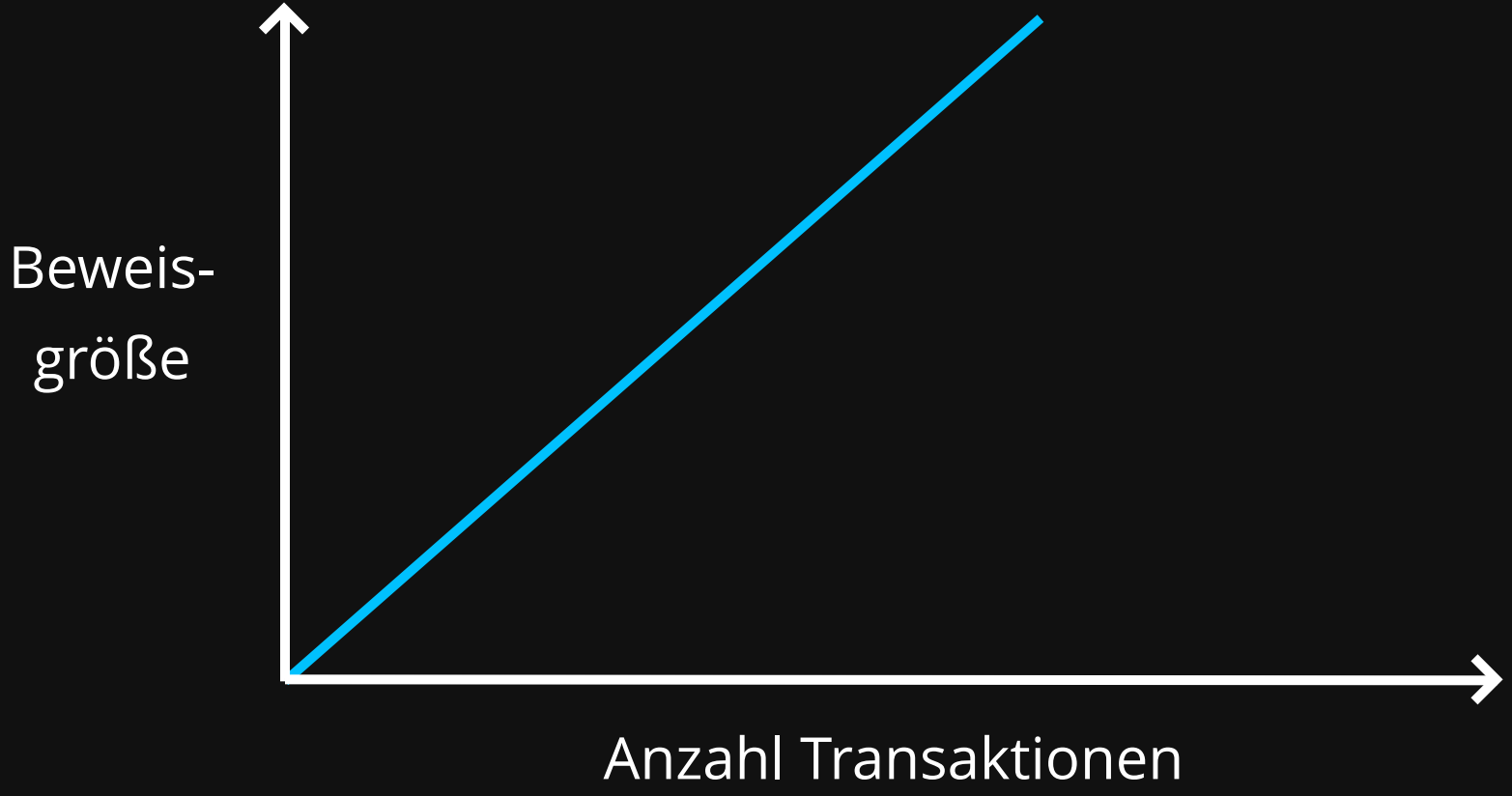
Ich kenne valide Transaktionen, die von Zustand 1 auf Zustand 2 führen.

Wie groß ist der Beweis und wie lange dauert die Verifikation?


Beweis-
größe
Anzahl Transaktionen

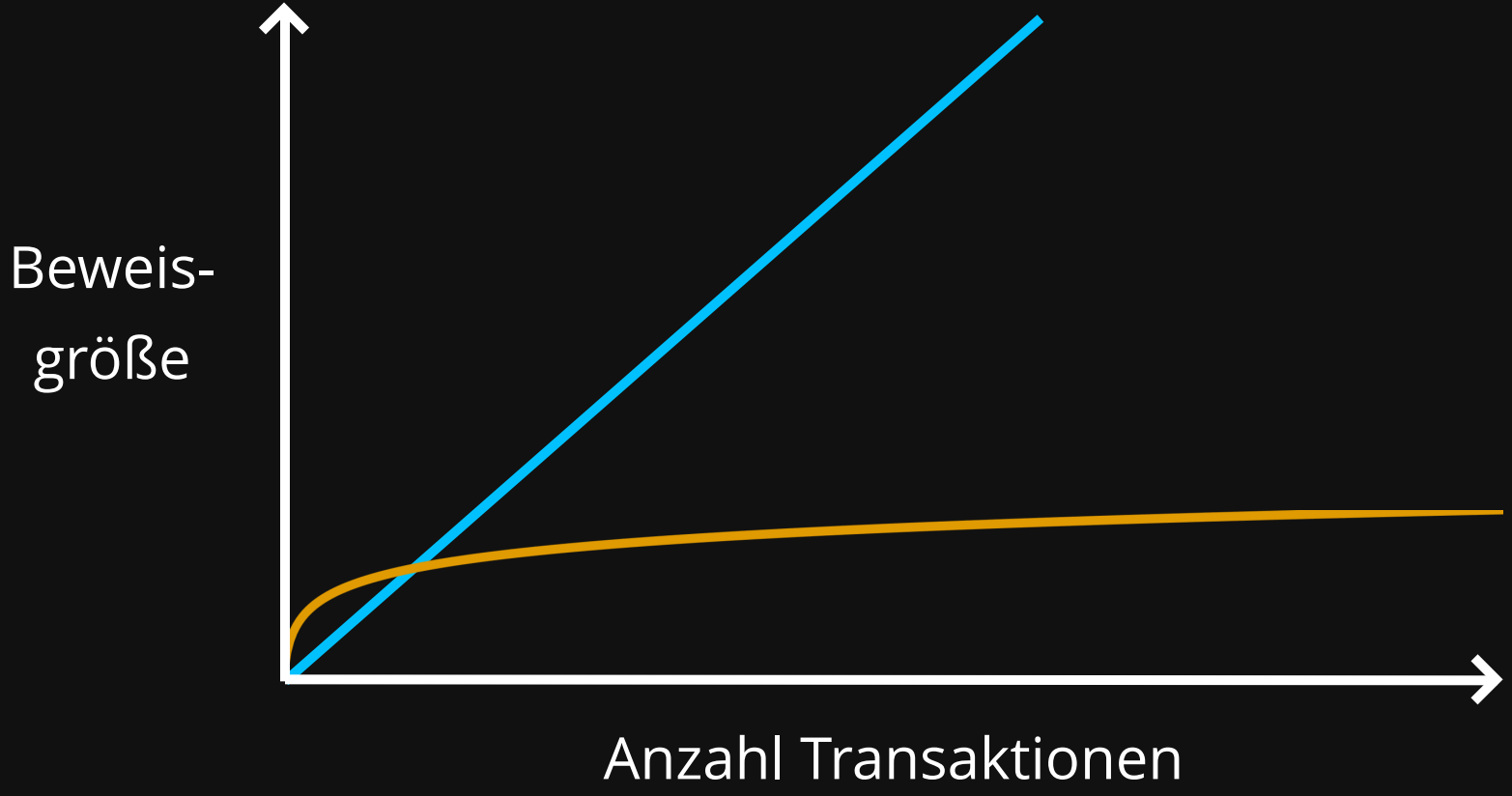


Trivial: Alle Transaktionen werden an den Verifier geschickt.



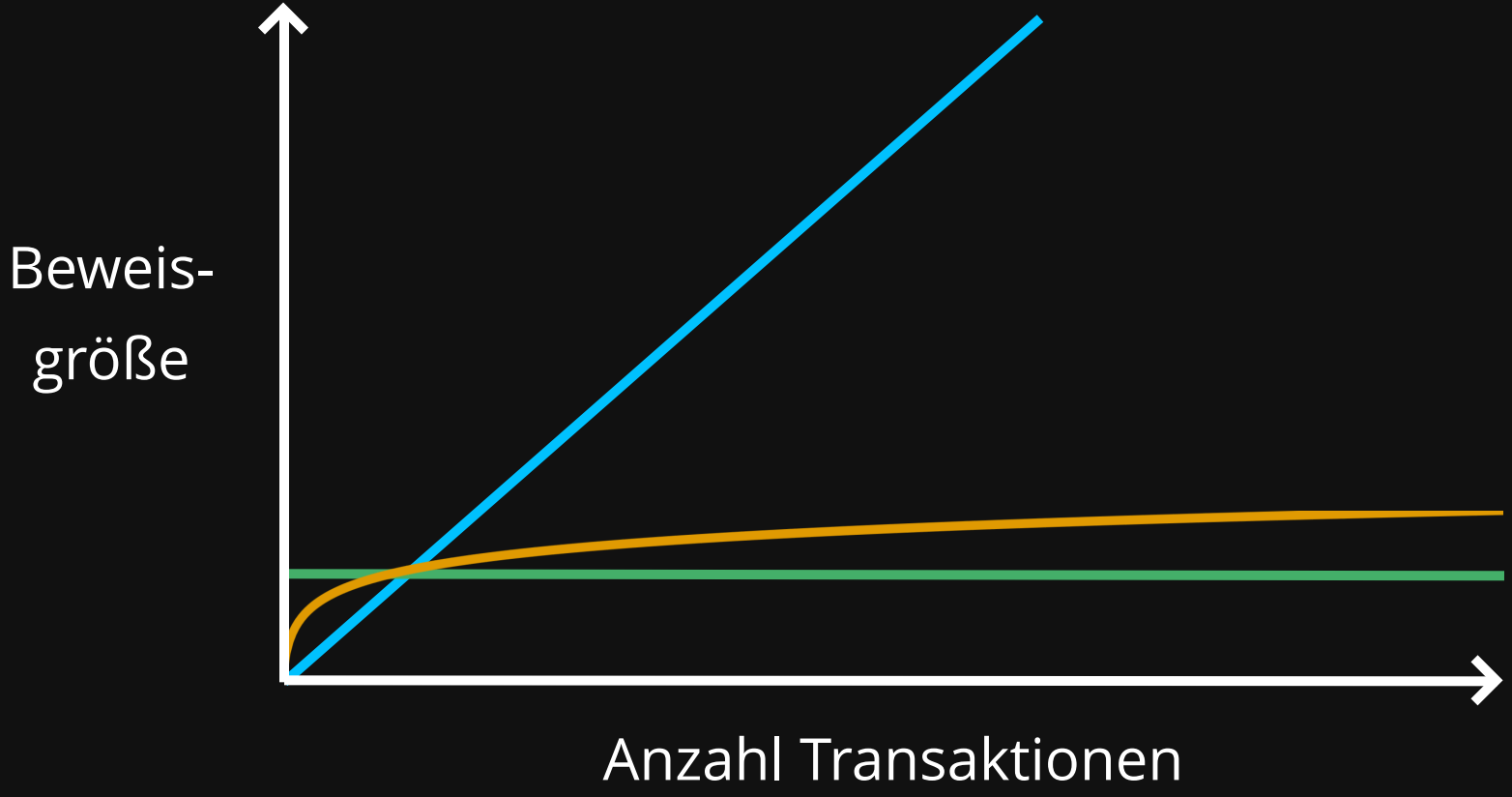
Trivial: Alle Transaktionen werden an den Verifier geschickt.

Beweissysteme mit Succinctness nennt man SNARKs



Trivial: Alle Transaktionen werden an den Verifier geschickt.
Logarithmische SNARKs, wie z.B. STARKs oder Bulletproofs

Beweissysteme mit Succinctness nennt man SNARKs



Trivial: Alle Transaktionen werden an den Verifier geschickt.
Logarithmische SNARKs, wie z.B. STARKs oder Bulletproofs
Konstante SNARKs (z.B. Groth16, Plonk / Marlin)

Beweissysteme mit Succinctness nennt man SNARKs

SNARK Zoo

Es gibt eine ganze Menge unterschliedlicher SNARKs (sehr aktives Forschungsfeld) mit verschiedenen Tradeoffs.

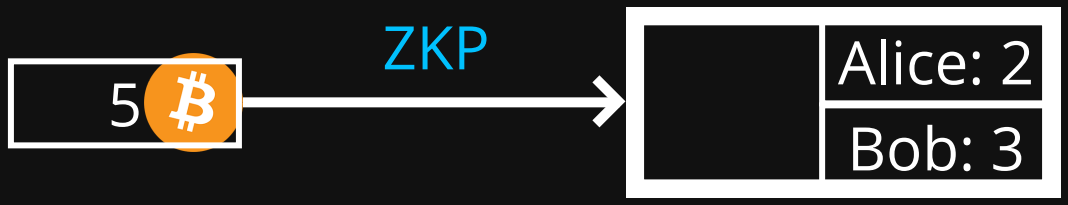
SNARK Zoo

Es gibt eine ganze Menge unterschliedlicher SNARKs (sehr aktives Forschungsfeld) mit verschiedenen Tradeoffs.

Bsp.:

Konstante SNARKs nur mit trusted Setup (1-aus-n).

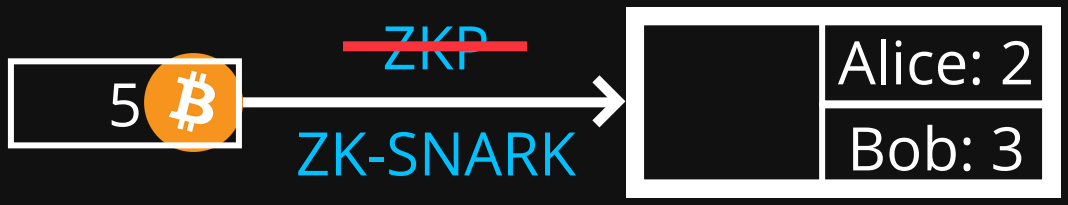
Bitcoin Script Erweiterung😕



Ziel

1. Bessere Privacy

Bitcoin Script Erweiterung😕



Ziel

1. Bessere Privacy
2. Bessere Skalierbarkeit

On-Off-Chain Mixer ("Validity Rollup")😕

On-Off-Chain Mixer ("Validity Rollup")😕

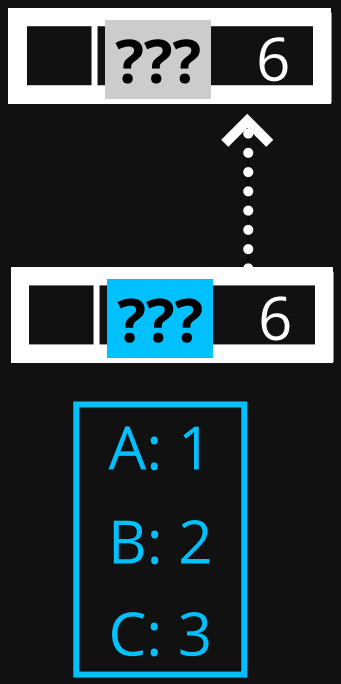


A: 1

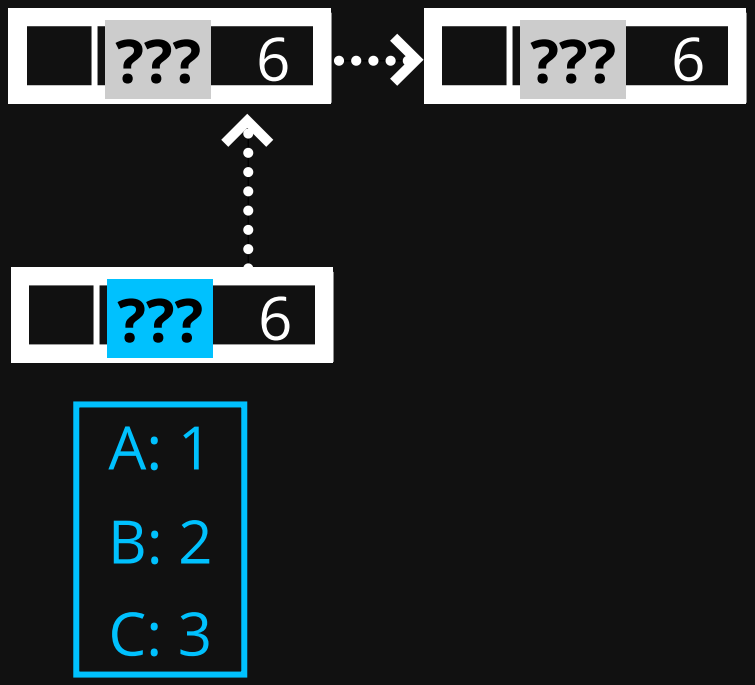
B: 2

C: 3

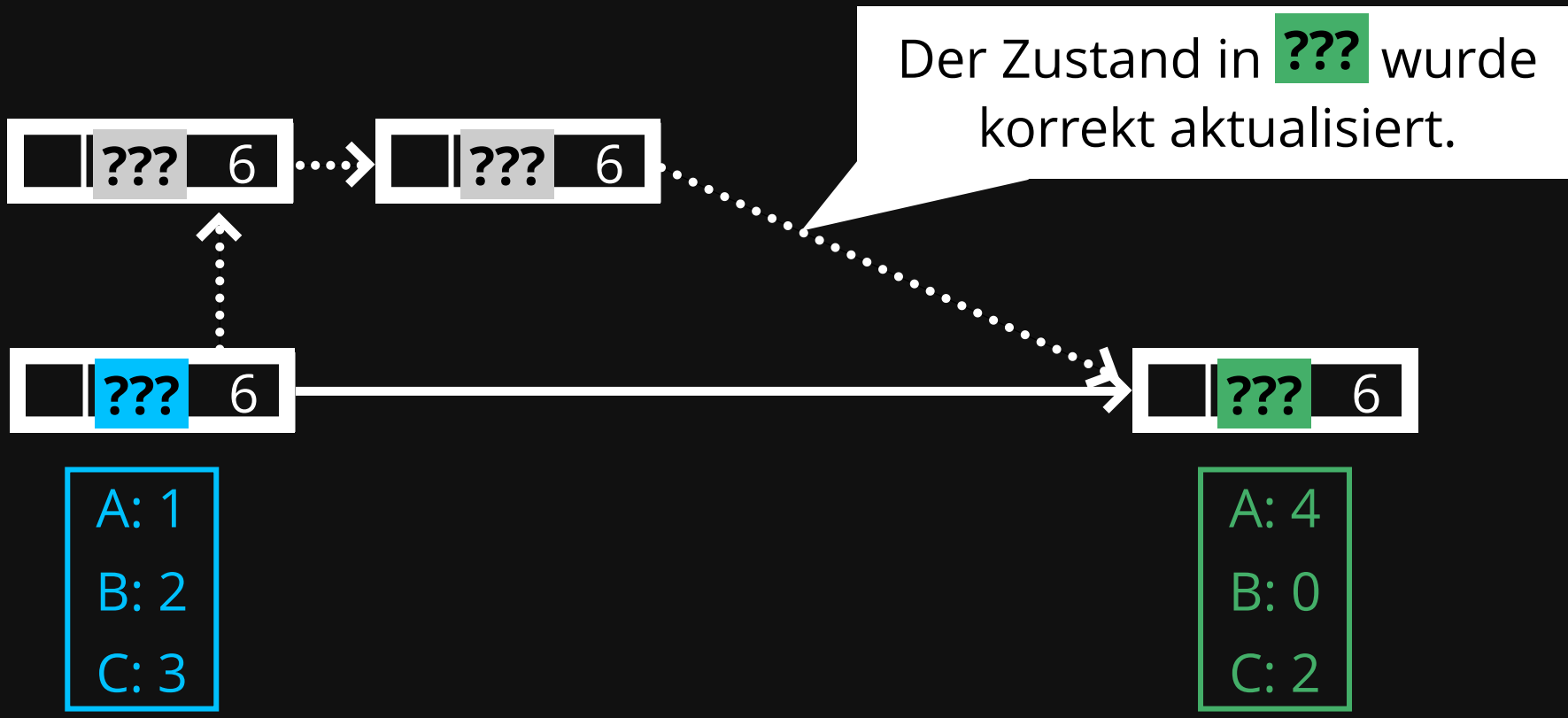
On-Off-Chain Mixer ("Validity Rollup")😕



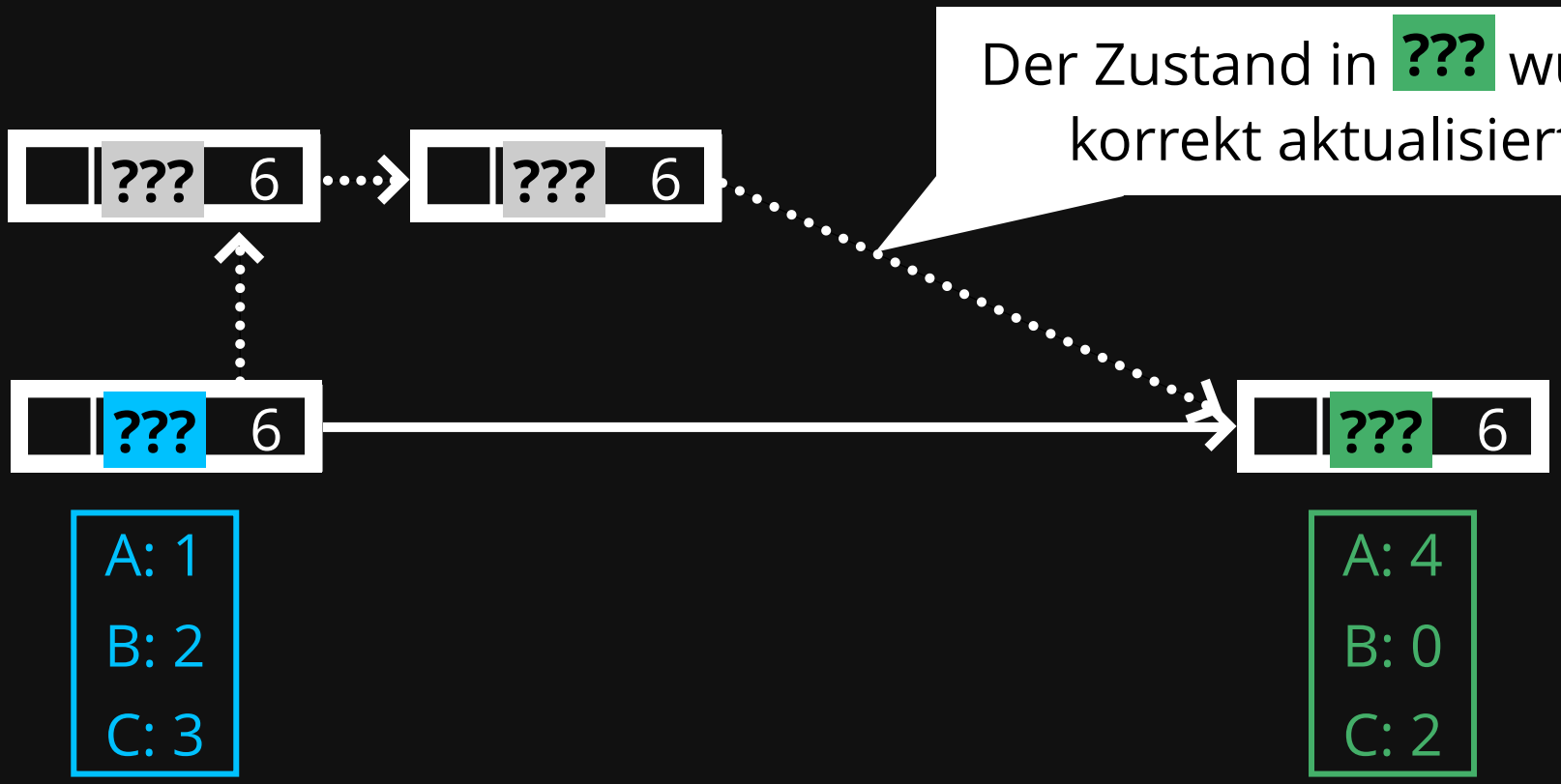
On-Off-Chain Mixer ("Validity Rollup")😕



On-Off-Chain Mixer ("Validity Rollup")😕



On-Off-Chain Mixer ("Validity Rollup")😕



Mit ZK-SNARKs sind interne Transaktionen sehr effizient.

Zusammenfassung

Zusammenfassung

- In Bitcoin gibt es **zahlreichen Anwendungen** für ZK-SNARKs
  - **ZK**: Zero-Knowledge
  - **SNARK**: Succinctness ("Kürze")

Zusammenfassung

- In Bitcoin gibt es **zahlreichen Anwendungen** für ZK-SNARKs
  - **ZK**: Zero-Knowledge
  - **SNARK**: Succinctness ("Kürze")
- ZK-SNARKs sind ein sehr aktives Forschungsfeld aber werden langsam in der **Praxis** nutzbar.

Zusammenfassung

- In Bitcoin gibt es **zahlreichen Anwendungen** für ZK-SNARKs
  - **ZK**: Zero-Knowledge
  - **SNARK**: Succinctness ("Kürze")
- ZK-SNARKs sind ein sehr aktives Forschungsfeld aber werden langsam in der **Praxis** nutzbar.
- Es gibt **off-chain** Anwendungen die heute schon machbar sind (ZKCP), sowie Anwendungen die eine Änderung des Bitcoin Protokolls benötigen würden.



Folien auf nickler.ninja/slides