

# Bitcoins (Kryptoparty 2014)

Walter Unger

Lehrstuhl für Informatik 1

16:38 Uhr, den 16. Januar 2014

# Möglichkeiten für [elektronisches] Geld

- Staat gibt Papiergeld Geltung

# Möglichkeiten für [elektronisches] Geld

- Staat gibt Papiergeld Geltung
- Elektronische Unterschrift (Identifikation, Elektronischer Schuldschein) 😞

# Möglichkeiten für [elektronisches] Geld

- Staat gibt Papiergeld Geltung
- Elektronische Unterschrift (Identifikation, Elektronischer Schuldschein) 😞
- Blinde elektronische Unterschrift 😊

# Möglichkeiten für [elektronisches] Geld

- Staat gibt Papiergeld Geltung
- Elektronische Unterschrift (Identifikation, Elektronischer Schuldschein) 😞
- Blinde elektronische Unterschrift 😊
- **Verwendete Technik: Public Key Systeme und sichere Hashfunktionen.**

# Möglichkeiten für [elektronisches] Geld

- Staat gibt Papiergeld Geltung
- Elektronische Unterschrift (Identifikation, Elektronischer Schuldschein) 😞
- Blinde elektronische Unterschrift 😊
- Verwendete Technik: Public Key Systeme und sichere Hashfunktionen.
  - Einfacher Umlauf: Bank  $\implies$  Kunde  $\implies$  Geschäft  $\implies$  Bank

# Möglichkeiten für [elektronisches] Geld

- Staat gibt Papiergeld Geltung
- Elektronische Unterschrift (Identifikation, Elektronischer Schuldschein) 😞
- Blinde elektronische Unterschrift 😊
- Verwendete Technik: Public Key Systeme und sichere Hashfunktionen.
  - Einfacher Umlauf: Bank  $\implies$  Kunde  $\implies$  Geschäft  $\implies$  Bank
  - **Beliebiger Umlauf: Bank  $\implies$  Kunde 1  $\implies$  Kunde 2  $\implies$  ...**

# Möglichkeiten für [elektronisches] Geld

- Staat gibt Papiergeld Geltung
- Elektronische Unterschrift (Identifikation, Elektronischer Schuldschein) 😞
- Blinde elektronische Unterschrift 😊
- Verwendete Technik: Public Key Systeme und sichere Hashfunktionen.
  - Einfacher Umlauf: Bank  $\implies$  Kunde  $\implies$  Geschäft  $\implies$  Bank
  - Beliebiger Umlauf: Bank  $\implies$  Kunde 1  $\implies$  Kunde 2  $\implies$  ...
- Bitcoins



# Möglichkeiten für [elektronisches] Geld

- Staat gibt Papiergeld Geltung
- Elektronische Unterschrift (Identifikation, Elektronischer Schuldschein) 😞
- Blinde elektronische Unterschrift 😊
- Verwendete Technik: Public Key Systeme und sichere Hashfunktionen.
  - Einfacher Umlauf: Bank  $\implies$  Kunde  $\implies$  Geschäft  $\implies$  Bank
  - Beliebiger Umlauf: Bank  $\implies$  Kunde 1  $\implies$  Kunde 2  $\implies$  ...
- Bitcoins
  - **Kein Staat involviert.**

# Möglichkeiten für [elektronisches] Geld

- Staat gibt Papiergeld Geltung
- Elektronische Unterschrift (Identifikation, Elektronischer Schuldschein) 😞
- Blinde elektronische Unterschrift 😊
- Verwendete Technik: Public Key Systeme und sichere Hashfunktionen.
  - Einfacher Umlauf: Bank  $\Rightarrow$  Kunde  $\Rightarrow$  Geschäft  $\Rightarrow$  Bank
  - Beliebiger Umlauf: Bank  $\Rightarrow$  Kunde 1  $\Rightarrow$  Kunde 2  $\Rightarrow$  ...
- Bitcoins
  - Kein Staat involviert.
  - Umlauf: ...  $\Rightarrow$  Kunde 1  $\Rightarrow$  Kunde 2  $\Rightarrow$  Kunde 3  $\Rightarrow$  ...

# Möglichkeiten für [elektronisches] Geld

- Staat gibt Papiergeld Geltung
- Elektronische Unterschrift (Identifikation, Elektronischer Schuldschein) 😞
- Blinde elektronische Unterschrift 😊
- Verwendete Technik: Public Key Systeme und sichere Hashfunktionen.
  - Einfacher Umlauf: Bank  $\Rightarrow$  Kunde  $\Rightarrow$  Geschäft  $\Rightarrow$  Bank
  - Beliebiger Umlauf: Bank  $\Rightarrow$  Kunde 1  $\Rightarrow$  Kunde 2  $\Rightarrow$  ...
- Bitcoins
  - Kein Staat involviert.
  - Umlauf: ...  $\Rightarrow$  Kunde 1  $\Rightarrow$  Kunde 2  $\Rightarrow$  Kunde 3  $\Rightarrow$  ...

# Geschichte

- b-money von Wei Dai [1998]

# Geschichte

- b-money von Wei Dai [1998]
- bit gold von Nick Szabo

# Geschichte

- b-money von Wei Dai [1998]
- bit gold von Nick Szabo
- Bitcoin Konzept von "Satoshi Nakamoto" [2008]

# Geschichte

- b-money von Wei Dai [1998]
- bit gold von Nick Szabo
- Bitcoin Konzept von "Satoshi Nakamoto" [2008]
- 50 erste Bitcoins 3. Januar 2009

# Geschichte

- b-money von Wei Dai [1998]
- bit gold von Nick Szabo
- Bitcoin Konzept von "Satoshi Nakamoto" [2008]
- 50 erste Bitcoins 3. Januar 2009
- Entwickler Gavin Andresen Satoshi Nakamoto, Martti Malmi, Amir Taaki, Pieter Wuille, Nils Schneider, Jeff Garzik und andere



# Geschichte

- b-money von Wei Dai [1998]
- bit gold von Nick Szabo
- Bitcoin Konzept von “Satoshi Nakamoto” [2008]
- 50 erste Bitcoins 3. Januar 2009
- Entwickler Gavin Andresen Satoshi Nakamoto, Martti Malmi, Amir Taaki, Pieter Wuille, Nils Schneider, Jeff Garzik und andere
- Betriebssysteme: Linux, Mac OS X, Windows

# Geschichte

- b-money von Wei Dai [1998]
- bit gold von Nick Szabo
- Bitcoin Konzept von “Satoshi Nakamoto” [2008]
- 50 erste Bitcoins 3. Januar 2009
- Entwickler Gavin Andresen Satoshi Nakamoto, Martti Malmi, Amir Taaki, Pieter Wuille, Nils Schneider, Jeff Garzik und andere
- Betriebssysteme: **Linux**, Mac OS X, **Windows**
- **Programmiersprache: C++**

# Geschichte

- b-money von Wei Dai [1998]
- bit gold von Nick Szabo
- Bitcoin Konzept von “Satoshi Nakamoto” [2008]
- 50 erste Bitcoins 3. Januar 2009
- Entwickler Gavin Andresen Satoshi Nakamoto, Martti Malmi, Amir Taaki, Pieter Wuille, Nils Schneider, Jeff Garzik und andere
- Betriebssysteme: [Linux](#), Mac OS X, [Windows](#)
- Programmiersprache: C++

# Grundlagen

- Es werden elektronische Münzen weitergegeben.

# Grundlagen

- Es werden elektronische Münzen weitergegeben.
- Alle Aufgaben werden von der Gemeinschaft erledigt, wie:

# Grundlagen

- Es werden elektronische Münzen weitergegeben.
- Alle Aufgaben werden von der Gemeinschaft erledigt, wie:
  - **Transaktionen tätigen**

# Grundlagen

- Es werden elektronische Münzen weitergegeben.
- Alle Aufgaben werden von der Gemeinschaft erledigt, wie:
  - Transaktionen tätigen
  - Transaktionen bestätigen (gemeinsame "Zeit" bestimmen)

# Grundlagen

- Es werden elektronische Münzen weitergegeben.
- Alle Aufgaben werden von der Gemeinschaft erledigt, wie:
  - Transaktionen tätigen
  - Transaktionen bestätigen (gemeinsame "Zeit" bestimmen)
- Zahlung durch Kette von Unterschriften.



# Grundlagen

- Es werden elektronische Münzen weitergegeben.
- Alle Aufgaben werden von der Gemeinschaft erledigt, wie:
  - Transaktionen tätigen
  - Transaktionen bestätigen (gemeinsame "Zeit" bestimmen)
- Zahlung durch Kette von Unterschriften.
- Zahlungen gültig, wenn von der Mehrheit der Teilnehmer bestätigt.

# Grundlagen

- Es werden elektronische Münzen weitergegeben.
- Alle Aufgaben werden von der Gemeinschaft erledigt, wie:
  - Transaktionen tätigen
  - Transaktionen bestätigen (gemeinsame "Zeit" bestimmen)
- Zahlung durch Kette von Unterschriften.
- Zahlungen gültig, wenn von der Mehrheit der Teilnehmer bestätigt.
- **Notwendige Grundlagen:**

# Grundlagen

- Es werden elektronische Münzen weitergegeben.
- Alle Aufgaben werden von der Gemeinschaft erledigt, wie:
  - Transaktionen tätigen
  - Transaktionen bestätigen (gemeinsame "Zeit" bestimmen)
- Zahlung durch Kette von Unterschriften.
- Zahlungen gültig, wenn von der Mehrheit der Teilnehmer bestätigt.
- Notwendige Grundlagen:
  - Jeder Teilnehmer hat Public Key Verfahren (persönliches Geheimnis)

# Grundlagen

- Es werden elektronische Münzen weitergegeben.
- Alle Aufgaben werden von der Gemeinschaft erledigt, wie:
  - Transaktionen tätigen
  - Transaktionen bestätigen (gemeinsame "Zeit" bestimmen)
- Zahlung durch Kette von Unterschriften.
- Zahlungen gültig, wenn von der Mehrheit der Teilnehmer bestätigt.
- Notwendige Grundlagen:
  - Jeder Teilnehmer hat Public Key Verfahren (persönliches Geheimnis)
  - Es gibt eine sichere Hashfunktion

# Grundlagen

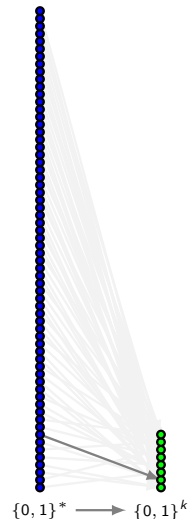
- Es werden elektronische Münzen weitergegeben.
- Alle Aufgaben werden von der Gemeinschaft erledigt, wie:
  - Transaktionen tätigen
  - Transaktionen bestätigen (gemeinsame "Zeit" bestimmen)
- Zahlung durch Kette von Unterschriften.
- Zahlungen gültig, wenn von der Mehrheit der Teilnehmer bestätigt.
- Notwendige Grundlagen:
  - Jeder Teilnehmer hat Public Key Verfahren (persönliches Geheimnis)
  - Es gibt eine sichere Hashfunktion
  - Es gibt ein Netzwerk (Broadcast) oder Bulletin Board

# Grundlagen

- Es werden elektronische Münzen weitergegeben.
- Alle Aufgaben werden von der Gemeinschaft erledigt, wie:
  - Transaktionen tätigen
  - Transaktionen bestätigen (gemeinsame "Zeit" bestimmen)
- Zahlung durch Kette von Unterschriften.
- Zahlungen gültig, wenn von der Mehrheit der Teilnehmer bestätigt.
- Notwendige Grundlagen:
  - Jeder Teilnehmer hat Public Key Verfahren (persönliches Geheimnis)
  - Es gibt eine sichere Hashfunktion
  - Es gibt ein Netzwerk (Broadcast) oder Bulletin Board

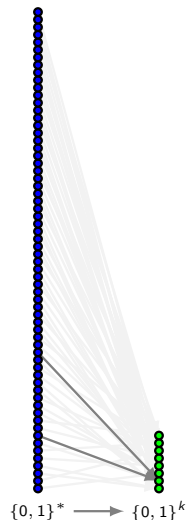
# Sichere Hashfunktion (im Bild)

●  $h : \{0, 1\}^* \mapsto \{0, 1\}^k, k \in \mathbb{N}$



# Sichere Hashfunktion (im Bild)

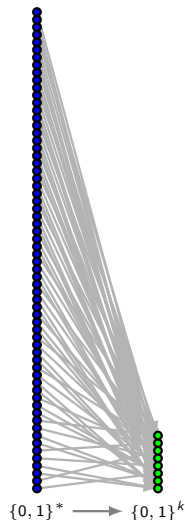
- $h : \{0, 1\}^* \mapsto \{0, 1\}^k, k \in \mathbb{N}$
- $\exists w, w' : h(w) = h(w')$





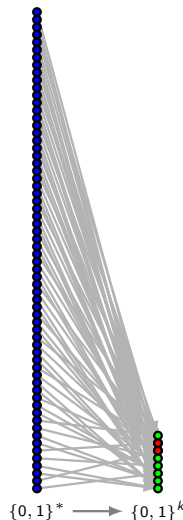
# Sichere Hashfunktion (im Bild)

- $h : \{0, 1\}^* \mapsto \{0, 1\}^k, k \in \mathbb{N}$
- $\exists w, w' : h(w) = h(w')$
- Sind aber schwer zu finden.



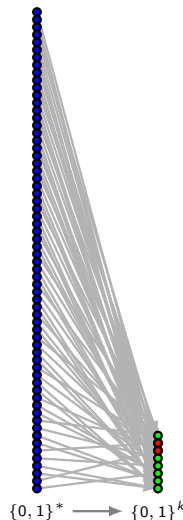
# Sichere Hashfunktion (im Bild)

- $h : \{0, 1\}^* \mapsto \{0, 1\}^k, k \in \mathbb{N}$
- $\exists w, w' : h(w) = h(w')$
- Sind aber schwer zu finden.
- Für einen "kleinen" Wertebereich  $\mathcal{D}$  ist es aufwendig ein  $w$  zu finden mit  $h(w) \in \mathcal{D}$ .



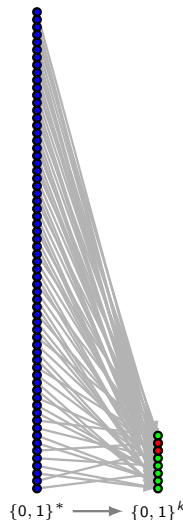
# Sichere Hashfunktion (im Bild) und Unterschrift

- $h : \{0, 1\}^* \mapsto \{0, 1\}^k, k \in \mathbb{N}$
- $\exists w, w' : h(w) = h(w')$
- Sind aber schwer zu finden.
- Für einen “kleinen” Wertebereich  $\mathcal{D}$  ist es aufwendig ein  $w$  zu finden mit  $h(w) \in \mathcal{D}$ .
- Unterschrift von  $A$  unter Text  $T$ :



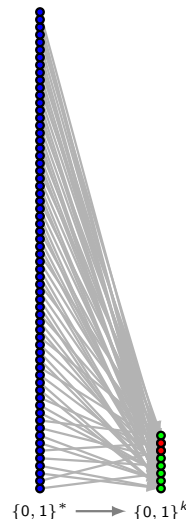
# Sichere Hashfunktion (im Bild) und Unterschrift

- $h : \{0, 1\}^* \mapsto \{0, 1\}^k, k \in \mathbb{N}$
- $\exists w, w' : h(w) = h(w')$
- Sind aber schwer zu finden.
- Für einen “kleinen” Wertebereich  $\mathcal{D}$  ist es aufwendig ein  $w$  zu finden mit  $h(w) \in \mathcal{D}$ .
- Unterschrift von A unter Text  $T$ :
- $s = \text{Signatur}_{\text{Secret}(A)}(\text{Text})$



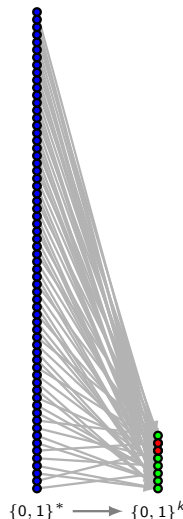
# Sichere Hashfunktion (im Bild) und Unterschrift

- $h : \{0, 1\}^* \mapsto \{0, 1\}^k, k \in \mathbb{N}$
- $\exists w, w' : h(w) = h(w')$
- Sind aber schwer zu finden.
- Für einen “kleinen” Wertebereich  $\mathcal{D}$  ist es aufwendig ein  $w$  zu finden mit  $h(w) \in \mathcal{D}$ .
- Unterschrift von  $A$  unter Text  $T$ :
- $s = \text{Signatur}_{\text{Secret}(A)}(\text{Text})$
- $s = \text{Signatur}_{\text{Secret}(A)}(\text{Hash}(\text{Text}))$



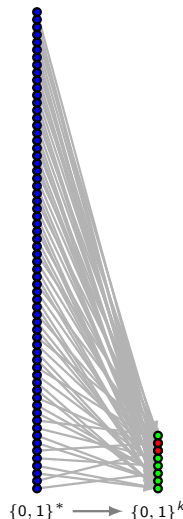
# Sichere Hashfunktion (im Bild) und Unterschrift

- $h : \{0, 1\}^* \mapsto \{0, 1\}^k, k \in \mathbb{N}$
- $\exists w, w' : h(w) = h(w')$
- Sind aber schwer zu finden.
- Für einen “kleinen” Wertebereich  $\mathcal{D}$  ist es aufwendig ein  $w$  zu finden mit  $h(w) \in \mathcal{D}$ .
- Unterschrift von  $A$  unter Text  $T$ :
- $s = \text{Signatur}_{\text{Secret}(A)}(\text{Text})$
- $s = \text{Signatur}_{\text{Secret}(A)}(\text{Hash}(\text{Text}))$
- Geht mit Public Key!



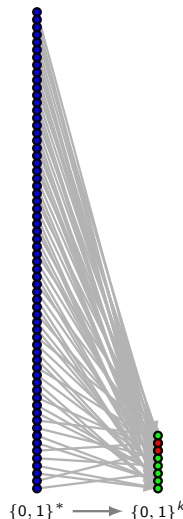
# Sichere Hashfunktion (im Bild) und Unterschrift

- $h : \{0, 1\}^* \mapsto \{0, 1\}^k, k \in \mathbb{N}$
- $\exists w, w' : h(w) = h(w')$
- Sind aber schwer zu finden.
- Für einen “kleinen” Wertebereich  $\mathcal{D}$  ist es aufwendig ein  $w$  zu finden mit  $h(w) \in \mathcal{D}$ .
- Unterschrift von  $A$  unter Text  $T$ :
- $s = \text{Signatur}_{\text{Secret}(A)}(\text{Text})$
- $s = \text{Signatur}_{\text{Secret}(A)}(\text{Hash}(\text{Text}))$
- Geht mit Public Key!
- $1\text{Euro} = \text{Signatur}_{\text{Secret}(\text{Bank})}(\text{Hash}(1.237\dots 8123))$  ☹️



# Sichere Hashfunktion (im Bild) und Unterschrift

- $h : \{0, 1\}^* \mapsto \{0, 1\}^k, k \in \mathbb{N}$
- $\exists w, w' : h(w) = h(w')$
- Sind aber schwer zu finden.
- Für einen “kleinen” Wertebereich  $\mathcal{D}$  ist es aufwendig ein  $w$  zu finden mit  $h(w) \in \mathcal{D}$ .
- Unterschrift von  $A$  unter Text  $T$ :
- $s = \text{Signatur}_{\text{Secret}(A)}(\text{Text})$
- $s = \text{Signatur}_{\text{Secret}(A)}(\text{Hash}(\text{Text}))$
- Geht mit Public Key!
- $1\text{Euro} = \text{Signatur}_{\text{Secret}(\text{Bank})}(\text{Hash}(1.237\dots 8123))$  ☹️





# Münze weiterreichen

- Idee: Sender unterschreibt letzte Transaktion der Münze.

# Münze weiterreichen

- Idee: Sender unterschreibt letzte Transaktion der Münze.
- Dabei wird Hashfunktion  $h$  eingesetzt.

# Münze weiterreichen

- Idee: Sender unterschreibt letzte Transaktion der Münze.
- Dabei wird Hashfunktion  $h$  eingesetzt.
- Beispiel Münze sei 1.343543252 wird von  $A$  an  $B$  gegeben:

# Münze weiterreichen

- Idee: Sender unterschreibt letzte Transaktion der Münze.
- Dabei wird Hashfunktion  $h$  eingesetzt.
- Beispiel Münze sei 1.343543252 wird von  $A$  an  $B$  gegeben:
- $B$  bekommt von  $A$ :  $S_1 = \text{Signatur}_{\text{Secret}(A)}(\text{Hash}(A \mapsto B : 1.343543252))$ .

# Münze weiterreichen

- Idee: Sender unterschreibt letzte Transaktion der Münze.
- Dabei wird Hashfunktion  $h$  eingesetzt.
- Beispiel Münze sei 1.343543252 wird von  $A$  an  $B$  gegeben:
- $B$  bekommt von  $A$ :  $S_1 = \text{Signatur}_{\text{Secret}(A)}(\text{Hash}(A \mapsto B : 1.343543252))$ .
- $C$  bekommt von  $B$ :  
 $S_2 = \text{Signatur}_{\text{Secret}(B)}(\text{Hash}(A \mapsto B \mapsto C : 1.343543252))$ .

# Münze weiterreichen

- Idee: Sender unterschreibt letzte Transaktion der Münze.
- Dabei wird Hashfunktion  $h$  eingesetzt.
- Beispiel Münze sei 1.343543252 wird von  $A$  an  $B$  gegeben:
- $B$  bekommt von  $A$ :  $S_1 = \text{Signatur}_{\text{Secret}(A)}(\text{Hash}(A \mapsto B : 1.343543252))$ .
- $C$  bekommt von  $B$ :  
 $S_2 = \text{Signatur}_{\text{Secret}(B)}(\text{Hash}(A \mapsto B \mapsto C : 1.343543252))$ .
- $D$  bekommt von  $C$ :  
 $S_3 = \text{Signatur}_{\text{Secret}(C)}(\text{Hash}(B \mapsto C \mapsto D : 1.343543252))$ .

# Münze weiterreichen

- Idee: Sender unterschreibt letzte Transaktion der Münze.
- Dabei wird Hashfunktion  $h$  eingesetzt.
- Beispiel Münze sei 1.343543252 wird von  $A$  an  $B$  gegeben:
- $B$  bekommt von  $A$ :  $S_1 = \text{Signatur}_{\text{Secret}(A)}(\text{Hash}(A \mapsto B : 1.343543252))$ .
- $C$  bekommt von  $B$ :  
 $S_2 = \text{Signatur}_{\text{Secret}(B)}(\text{Hash}(A \mapsto B \mapsto C : 1.343543252))$ .
- $D$  bekommt von  $C$ :  
 $S_3 = \text{Signatur}_{\text{Secret}(C)}(\text{Hash}(B \mapsto C \mapsto D : 1.343543252))$ .
- $B$  bekommt ein Auto von  $C$ .

# Münze weiterreichen

- Idee: Sender unterschreibt letzte Transaktion der Münze.
- Dabei wird Hashfunktion  $h$  eingesetzt.
- Beispiel Münze sei 1.343543252 wird von  $A$  an  $B$  gegeben:
- $B$  bekommt von  $A$ :  $S_1 = \text{Signatur}_{\text{Secret}(A)}(\text{Hash}(A \mapsto B : 1.343543252))$ .
- $C$  bekommt von  $B$ :  
 $S_2 = \text{Signatur}_{\text{Secret}(B)}(\text{Hash}(A \mapsto B \mapsto C : 1.343543252))$ .
- $D$  bekommt von  $C$ :  
 $S_3 = \text{Signatur}_{\text{Secret}(C)}(\text{Hash}(B \mapsto C \mapsto D : 1.343543252))$ .
- $B$  bekommt ein Auto von  $C$ .
- $C'$  bekommt von  $B$ :  
 $S'_2 = \text{Signatur}_{\text{Secret}(B)}(\text{Hash}(A \mapsto B \mapsto C' : 1.343543252))$ .



# Münze weiterreichen

- Idee: Sender unterschreibt letzte Transaktion der Münze.
- Dabei wird Hashfunktion  $h$  eingesetzt.
- Beispiel Münze sei 1.343543252 wird von  $A$  an  $B$  gegeben:
- $B$  bekommt von  $A$ :  $S_1 = \text{Signatur}_{\text{Secret}(A)}(\text{Hash}(A \mapsto B : 1.343543252))$ .
- $C$  bekommt von  $B$ :  
 $S_2 = \text{Signatur}_{\text{Secret}(B)}(\text{Hash}(A \mapsto B \mapsto C : 1.343543252))$ .
- $D$  bekommt von  $C$ :  
 $S_3 = \text{Signatur}_{\text{Secret}(C)}(\text{Hash}(B \mapsto C \mapsto D : 1.343543252))$ .
- $B$  bekommt ein Auto von  $C$ .
- $C'$  bekommt von  $B$ :  
 $S'_2 = \text{Signatur}_{\text{Secret}(B)}(\text{Hash}(A \mapsto B \mapsto C' : 1.343543252))$ .
- $C''$  bekommt von  $B$ :  
 $S''_2 = \text{Signatur}_{\text{Secret}(B)}(\text{Hash}(A \mapsto B \mapsto C'' : 1.343543252))$ .

# Münze weiterreichen

- Idee: Sender unterschreibt letzte Transaktion der Münze.
- Dabei wird Hashfunktion  $h$  eingesetzt.
- Beispiel Münze sei 1.343543252 wird von  $A$  an  $B$  gegeben:
- $B$  bekommt von  $A$ :  $S_1 = \text{Signatur}_{\text{Secret}(A)}(\text{Hash}(A \mapsto B : 1.343543252))$ .
- $C$  bekommt von  $B$ :  
 $S_2 = \text{Signatur}_{\text{Secret}(B)}(\text{Hash}(A \mapsto B \mapsto C : 1.343543252))$ .
- $D$  bekommt von  $C$ :  
 $S_3 = \text{Signatur}_{\text{Secret}(C)}(\text{Hash}(B \mapsto C \mapsto D : 1.343543252))$ .
- $B$  bekommt ein Auto von  $C$ .
- $C'$  bekommt von  $B$ :  
 $S'_2 = \text{Signatur}_{\text{Secret}(B)}(\text{Hash}(A \mapsto B \mapsto C' : 1.343543252))$ .
- $C''$  bekommt von  $B$ :  
 $S''_2 = \text{Signatur}_{\text{Secret}(B)}(\text{Hash}(A \mapsto B \mapsto C'' : 1.343543252))$ .
- $B$  hat drei Autos mit einer Münze bezahlt.

# Münze weiterreichen

- Idee: Sender unterschreibt letzte Transaktion der Münze.
- Dabei wird Hashfunktion  $h$  eingesetzt.
- Beispiel Münze sei 1.343543252 wird von  $A$  an  $B$  gegeben:
- $B$  bekommt von  $A$ :  $S_1 = \text{Signatur}_{\text{Secret}(A)}(\text{Hash}(A \mapsto B : 1.343543252))$ .
- $C$  bekommt von  $B$ :  
 $S_2 = \text{Signatur}_{\text{Secret}(B)}(\text{Hash}(A \mapsto B \mapsto C : 1.343543252))$ .
- $D$  bekommt von  $C$ :  
 $S_3 = \text{Signatur}_{\text{Secret}(C)}(\text{Hash}(B \mapsto C \mapsto D : 1.343543252))$ .
- $B$  bekommt ein Auto von  $C$ .
- $C'$  bekommt von  $B$ :  
 $S'_2 = \text{Signatur}_{\text{Secret}(B)}(\text{Hash}(A \mapsto B \mapsto C' : 1.343543252))$ .
- $C''$  bekommt von  $B$ :  
 $S''_2 = \text{Signatur}_{\text{Secret}(B)}(\text{Hash}(A \mapsto B \mapsto C'' : 1.343543252))$ .
- $B$  hat drei Autos mit einer Münze bezahlt.
- Jemand muss so eine Transaktion bestätigen.

# Münze weiterreichen

- Idee: Sender unterschreibt letzte Transaktion der Münze.
- Dabei wird Hashfunktion  $h$  eingesetzt.
- Beispiel Münze sei 1.343543252 wird von  $A$  an  $B$  gegeben:
- $B$  bekommt von  $A$ :  $S_1 = \text{Signatur}_{\text{Secret}(A)}(\text{Hash}(A \mapsto B : 1.343543252))$ .
- $C$  bekommt von  $B$ :  
 $S_2 = \text{Signatur}_{\text{Secret}(B)}(\text{Hash}(A \mapsto B \mapsto C : 1.343543252))$ .
- $D$  bekommt von  $C$ :  
 $S_3 = \text{Signatur}_{\text{Secret}(C)}(\text{Hash}(B \mapsto C \mapsto D : 1.343543252))$ .
- $B$  bekommt ein Auto von  $C$ .
- $C'$  bekommt von  $B$ :  
 $S'_2 = \text{Signatur}_{\text{Secret}(B)}(\text{Hash}(A \mapsto B \mapsto C' : 1.343543252))$ .
- $C''$  bekommt von  $B$ :  
 $S''_2 = \text{Signatur}_{\text{Secret}(B)}(\text{Hash}(A \mapsto B \mapsto C'' : 1.343543252))$ .
- $B$  hat drei Autos mit einer Münze bezahlt.
- Jemand muss so eine Transaktion bestätigen.

# Münze weiterreichen

- Idee: Sender unterschreibt letzte Transaktion der Münze.
- Dabei wird Hashfunktion  $h$  eingesetzt.
- Beispiel Münze sei 1.343543252 wird von  $A$  an  $B$  gegeben:
- $B$  bekommt von  $A$ :  $S_1 = \text{Signatur}_{\text{Secret}(A)}(\text{Hash}(A \mapsto B : 1.343543252))$ .
- $C$  bekommt von  $B$ :  
 $S_2 = \text{Signatur}_{\text{Secret}(B)}(\text{Hash}(A \mapsto B \mapsto C : 1.343543252))$ .
- $D$  bekommt von  $C$ :  
 $S_3 = \text{Signatur}_{\text{Secret}(C)}(\text{Hash}(B \mapsto C \mapsto D : 1.343543252))$ .
- $B$  bekommt ein Auto von  $C$ .
- $C'$  bekommt von  $B$ :  
 $S'_2 = \text{Signatur}_{\text{Secret}(B)}(\text{Hash}(A \mapsto B \mapsto C' : 1.343543252))$ .
- $C''$  bekommt von  $B$ :  
 $S''_2 = \text{Signatur}_{\text{Secret}(B)}(\text{Hash}(A \mapsto B \mapsto C'' : 1.343543252))$ .
- $B$  hat drei Autos mit einer Münze bezahlt.
- Jemand muss so eine Transaktion bestätigen.

# Zahlungen bestätigen (Zeit verwalten)

- Zeit wird über durchgeführte Transaktionen verwaltet.

# Zahlungen bestätigen (Zeit verwalten)

- Zeit wird über durchgeführte Transaktionen verwaltet.
- Zeit vergeht dabei durch den Einsatz von Rechenleistung.

# Zahlungen bestätigen (Zeit verwalten)

- Zeit wird über durchgeführte Transaktionen verwaltet.
- Zeit vergeht dabei durch den Einsatz von Rechenleistung.
- Dazu werden folgende Schritte gemacht:



# Zahlungen bestätigen (Zeit verwalten)

- Zeit wird über durchgeführte Transaktionen verwaltet.
- Zeit vergeht dabei durch den Einsatz von Rechenleistung.
- Dazu werden folgende Schritte gemacht:
  - ① Transaktionen werden veröffentlicht.

# Zahlungen bestätigen (Zeit verwalten)

- Zeit wird über durchgeführte Transaktionen verwaltet.
- Zeit vergeht dabei durch den Einsatz von Rechenleistung.
- Dazu werden folgende Schritte gemacht:
  - 1 Transaktionen werden veröffentlicht.
  - 2 Jeder Teilnehmer sammelt die anstehenden Transaktionen.

# Zahlungen bestätigen (Zeit verwalten)

- Zeit wird über durchgeführte Transaktionen verwaltet.
- Zeit vergeht dabei durch den Einsatz von Rechenleistung.
- Dazu werden folgende Schritte gemacht:
  - 1 Transaktionen werden veröffentlicht.
  - 2 Jeder Teilnehmer sammelt die anstehenden Transaktionen.
  - 3 Jeder Teilnehmer bestimmt Hashwert von:

# Zahlungen bestätigen (Zeit verwalten)

- Zeit wird über durchgeführte Transaktionen verwaltet.
- Zeit vergeht dabei durch den Einsatz von Rechenleistung.
- Dazu werden folgende Schritte gemacht:
  - ① Transaktionen werden veröffentlicht.
  - ② Jeder Teilnehmer sammelt die anstehenden Transaktionen.
  - ③ Jeder Teilnehmer bestimmt Hashwert von:
    - Liste der anstehenden Transaktionen,

# Zahlungen bestätigen (Zeit verwalten)

- Zeit wird über durchgeführte Transaktionen verwaltet.
- Zeit vergeht dabei durch den Einsatz von Rechenleistung.
- Dazu werden folgende Schritte gemacht:
  - 1 Transaktionen werden veröffentlicht.
  - 2 Jeder Teilnehmer sammelt die anstehenden Transaktionen.
  - 3 Jeder Teilnehmer bestimmt Hashwert von:
    - Liste der anstehenden Transaktionen,
    - Dem vorherigen Hashwert und

# Zahlungen bestätigen (Zeit verwalten)

- Zeit wird über durchgeführte Transaktionen verwaltet.
- Zeit vergeht dabei durch den Einsatz von Rechenleistung.
- Dazu werden folgende Schritte gemacht:
  - 1 Transaktionen werden veröffentlicht.
  - 2 Jeder Teilnehmer sammelt die anstehenden Transaktionen.
  - 3 Jeder Teilnehmer bestimmt Hashwert von:
    - Liste der anstehenden Transaktionen,
    - Dem vorherigen Hashwert und
    - einer Zahl aus einer Zahlenfolge.

# Zahlungen bestätigen (Zeit verwalten)

- Zeit wird über durchgeführte Transaktionen verwaltet.
- Zeit vergeht dabei durch den Einsatz von Rechenleistung.
- Dazu werden folgende Schritte gemacht:
  - 1 Transaktionen werden veröffentlicht.
  - 2 Jeder Teilnehmer sammelt die anstehenden Transaktionen.
  - 3 Jeder Teilnehmer bestimmt Hashwert von:
    - Liste der anstehenden Transaktionen,
    - Dem vorherigen Hashwert und
    - einer Zahl aus einer Zahlenfolge.
  - 4 Das Hashergebnis muss mit einer bestimmten Zahl von Nullen starten.

# Zahlungen bestätigen (Zeit verwalten)

- Zeit wird über durchgeführte Transaktionen verwaltet.
- Zeit vergeht dabei durch den Einsatz von Rechenleistung.
- Dazu werden folgende Schritte gemacht:
  - 1 Transaktionen werden veröffentlicht.
  - 2 Jeder Teilnehmer sammelt die anstehenden Transaktionen.
  - 3 Jeder Teilnehmer bestimmt Hashwert von:
    - Liste der anstehenden Transaktionen,
    - Dem vorherigen Hashwert und
    - einer Zahl aus einer Zahlenfolge.
  - 4 Das Hashergebnis muss mit einer bestimmten Zahl von Nullen starten.
  - 5 D.h. es muss viel Rechenleistung eingesetzt werden.



# Zahlungen bestätigen (Zeit verwalten)

- Zeit wird über durchgeführte Transaktionen verwaltet.
- Zeit vergeht dabei durch den Einsatz von Rechenleistung.
- Dazu werden folgende Schritte gemacht:
  - 1 Transaktionen werden veröffentlicht.
  - 2 Jeder Teilnehmer sammelt die anstehenden Transaktionen.
  - 3 Jeder Teilnehmer bestimmt Hashwert von:
    - Liste der anstehenden Transaktionen,
    - Dem vorherigen Hashwert und
    - einer Zahl aus einer Zahlenfolge.
  - 4 Das Hashergebnis muss mit einer bestimmten Zahl von Nullen starten.
  - 5 D.h. es muss viel Rechenleistung eingesetzt werden.
- Erst wenn ausreichend Bestätigungen vorhanden sind, wird Transaktion gültig.

# Zahlungen bestätigen (Zeit verwalten)

- Zeit wird über durchgeführte Transaktionen verwaltet.
- Zeit vergeht dabei durch den Einsatz von Rechenleistung.
- Dazu werden folgende Schritte gemacht:
  - 1 Transaktionen werden veröffentlicht.
  - 2 Jeder Teilnehmer sammelt die anstehenden Transaktionen.
  - 3 Jeder Teilnehmer bestimmt Hashwert von:
    - Liste der anstehenden Transaktionen,
    - Dem vorherigen Hashwert und
    - einer Zahl aus einer Zahlenfolge.
  - 4 Das Hashergebnis muss mit einer bestimmten Zahl von Nullen starten.
  - 5 D.h. es muss viel Rechenleistung eingesetzt werden.
- Erst wenn ausreichend Bestätigungen vorhanden sind, wird Transaktion gültig.

# Zusammenfassung von Bitcoin

- Durch Einsatz von Rechenleistung werden:

# Zusammenfassung von Bitcoin

- Durch Einsatz von Rechenleistung werden:
  - Münzen generiert.

# Zusammenfassung von Bitcoin

- Durch Einsatz von Rechenleistung werden:
  - Münzen generiert.
  - **Transaktionen bestätigt.**

# Zusammenfassung von Bitcoin

- Durch Einsatz von Rechenleistung werden:
  - Münzen generiert.
  - Transaktionen bestätigt.
- Durch die Bestätigung werden geringe Gebühren erhoben.

# Zusammenfassung von Bitcoin

- Durch Einsatz von Rechenleistung werden:
  - Münzen generiert.
  - Transaktionen bestätigt.
- Durch die Bestätigung werden geringe Gebühren erhoben.
- Viele müssen bestätigen.

# Zusammenfassung von Bitcoin

- Durch Einsatz von Rechenleistung werden:
  - Münzen generiert.
  - Transaktionen bestätigt.
- Durch die Bestätigung werden geringe Gebühren erhoben.
- Viele müssen bestätigen.
- Ein Angreifer muss ähnlich viel Rechenleistung aufbringen, wie das Netzwerk.



# Zusammenfassung von Bitcoin

- Durch Einsatz von Rechenleistung werden:
  - Münzen generiert.
  - Transaktionen bestätigt.
- Durch die Bestätigung werden geringe Gebühren erhoben.
- Viele müssen bestätigen.
- Ein Angreifer muss ähnlich viel Rechenleistung aufbringen, wie das Netzwerk.
- Teilnehmer können anonym im Netzwerk auftreten.

# Zusammenfassung von Bitcoin

- Durch Einsatz von Rechenleistung werden:
  - Münzen generiert.
  - Transaktionen bestätigt.
- Durch die Bestätigung werden geringe Gebühren erhoben.
- Viele müssen bestätigen.
- Ein Angreifer muss ähnlich viel Rechenleistung aufbringen, wie das Netzwerk.
- Teilnehmer können anonym im Netzwerk auftreten.

# Ausblick

- Bisherige Angriffe galten den Rechnern, Bitcoins stehlen.

# Ausblick

- Bisherige Angriffe galten den Rechnern, Bitcoins stehlen.
- Anfällig gegen DDoS-Angriffe.

# Ausblick

- Bisherige Angriffe galten den Rechnern, Bitcoins stehlen.
- Anfällig gegen DDoS-Angriffe.
- Wert ist starken Schwankungen unterworfen.

# Ausblick

- Bisherige Angriffe galten den Rechnern, Bitcoins stehlen.
- Anfällig gegen DDoS-Angriffe.
- Wert ist starken Schwankungen unterworfen.
- Seit Mitte Dezember in China verboten.

# Ausblick

- Bisherige Angriffe galten den Rechnern, Bitcoins stehlen.
- Anfällig gegen DDoS-Angriffe.
- Wert ist starken Schwankungen unterworfen.
- Seit Mitte Dezember in China verboten.
- **Sehr interessant: Geld ohne Staat.**

# Ausblick

- Bisherige Angriffe galten den Rechnern, Bitcoins stehlen.
- Anfällig gegen DDoS-Angriffe.
- Wert ist starken Schwankungen unterworfen.
- Seit Mitte Dezember in China verboten.
- Sehr interessant: Geld ohne Staat.
- Mögliche weitere Anwendungen durch "Bestätigungen" von der Allgemeinheit.



# Ausblick

- Bisherige Angriffe galten den Rechnern, Bitcoins stehlen.
- Anfällig gegen DDoS-Angriffe.
- Wert ist starken Schwankungen unterworfen.
- Seit Mitte Dezember in China verboten.
- Sehr interessant: Geld ohne Staat.
- Mögliche weitere Anwendungen durch “Bestätigungen” von der Allgemeinheit.