# The Economics of Cryptocurrencies

Jonathan Chiu[1] Thorsten Koeppl[2]

[1]Bank of Canada

[2]Queen's U

Economics of Payments IX

November 2018

# What We Do

1) We formally model a cryptocurrency system according to the Bitcoin protocol.

- A ledger of **digital** balances updated in a **decentralized** fashion.

# What We Do

1) We formally model a cryptocurrency system according to the Bitcoin protocol.

- A ledger of **digital** balances updated in a **decentralized** fashion.

2) We show that cryptocurrencies cannot achieve immediate and final settlement.

- Why? Need to avoid a **double spending problem**.

# What We Do

1) We formally model a cryptocurrency system according to the Bitcoin protocol.

- A ledger of **digital** balances updated in a **decentralized** fashion.

2) We show that cryptocurrencies cannot achieve immediate and final settlement.

- Why? Need to avoid a **double spending problem**.

3) We evaluate the efficiency of a cryptocurrency system.

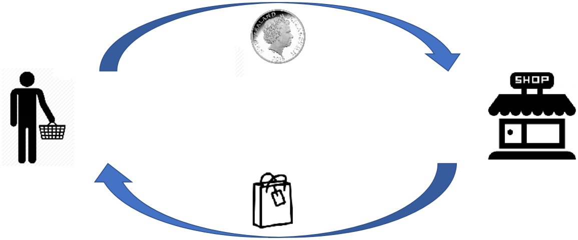
- **Positive inflation is optimal** while transaction fees should be minimized.
- Currently, **welfare loss in BITCOIN** of 1.4% of consumption, but potentially as low as 0.08%.

# Why Cryptocurrency is Special?

# Why Cryptocurrency is Special?

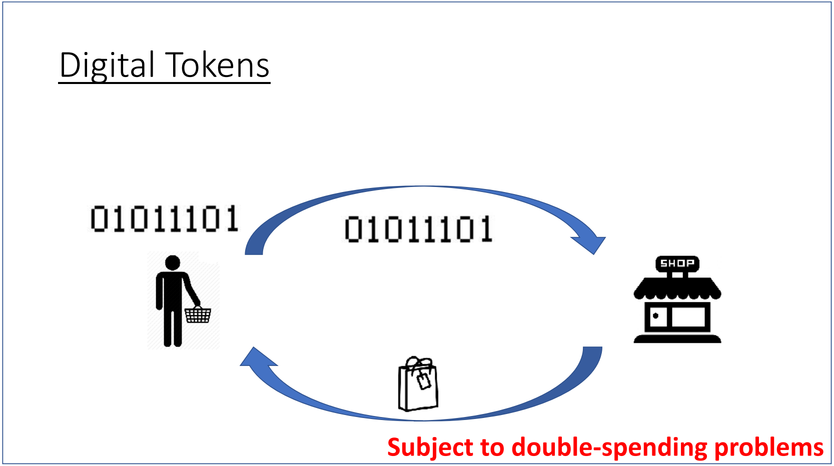
# Why Cryptocurrency is Special?

## Physical Tokens

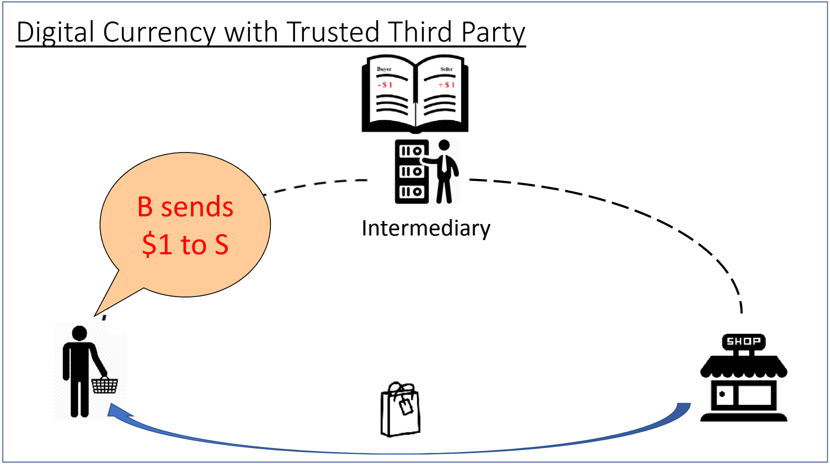



**Immediate and Final Settlement**

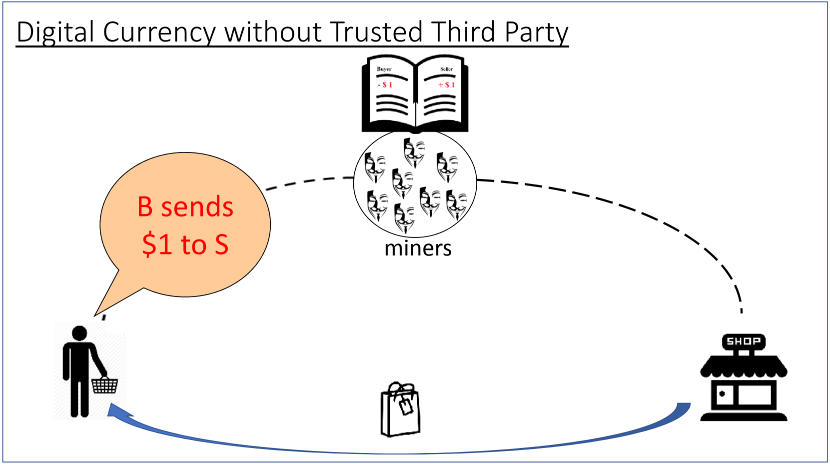
# Why Cryptocurrency is Special?



# Why Cryptocurrency is Special?



# Why Cryptocurrency is Special?




**No central authority to keep record**

# How Cryptocurrency works

1. Consensus Protocol

2. Reward Scheme

3. Confirmation Lags

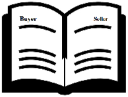
# How Cryptocurrency works

1. Consensus Protocol
   - competition in the form of mining: “miners” compete to update the public ledger (i.e. Blockchain)

2. Reward Scheme

3. Confirmation Lags

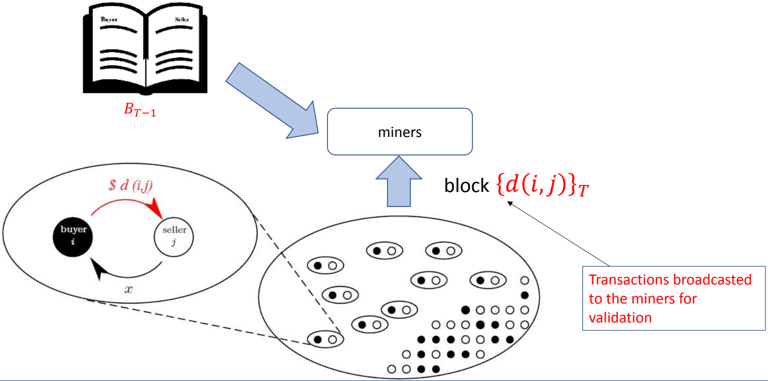
# Blockchain




$B_{T-1}$

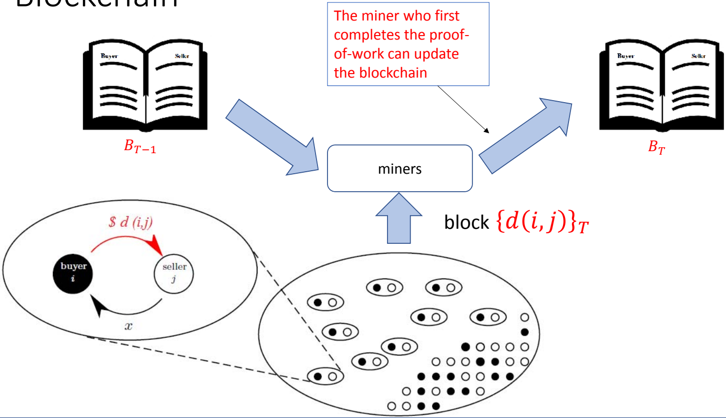
A book containing the ledger of all past transactions

# Blockchain



Buyer
Seller
$B_{T-1}$
miners
block $\{d(i,j)\}_T$
$ d (i,j)
buyer
i
seller
j
x
Transactions broadcasted to the miners for validation

# Blockchain


The miner who first completes the proof-of-work can update the blockchain
Buyer
Seller
$B_{T-1}$
Buyer
Seller
$B_T$
miners
block $\{d(i,j)\}_T$
$ d (i,j)
buyer $i$
seller $j$
$x$

# How Cryptocurrency works

1. Consensus Protocol
   - competition in the form of mining: “miners” compete to update the public ledger (i.e. Blockchain)
   - the prob. of winning is proportional to the fraction of computational power owned by a miner
2. Reward Scheme
3. Confirmation Lags

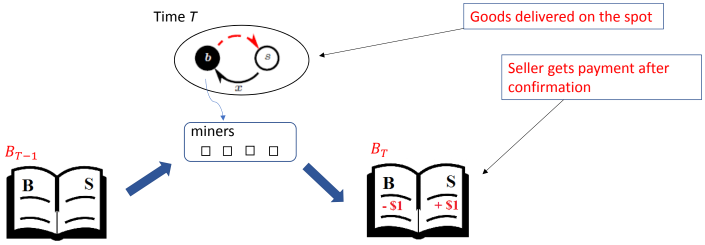
# How Cryptocurrency works

1. Consensus Protocol
   - competition in the form of mining: “miners” compete to update the public ledger (i.e. Blockchain)
   - the prob. of winning is proportional to the fraction of computational power owned by a miner
2. Reward Scheme
   - reward winning miners by seigniorage and transaction fees
3. Confirmation Lags

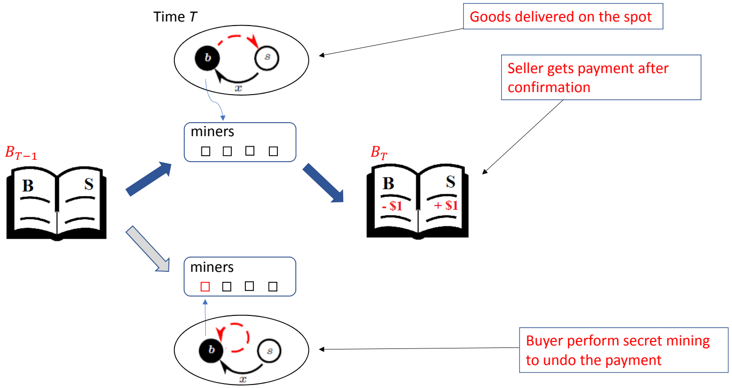
# How Cryptocurrency works

1. Consensus Protocol
   - competition in the form of mining: “miners” compete to update the public ledger (i.e. Blockchain)
   - the prob. of winning is proportional to the fraction of computational power owned by a miner
2. Reward Scheme
   - reward winning miners by seigniorage and transaction fees
3. Confirmation Lags
   - double spending is discouraged by confirmation delay

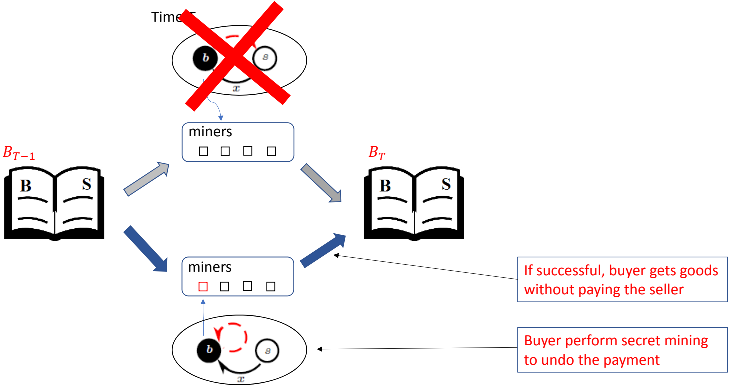
# No confirmation lag (N=0)



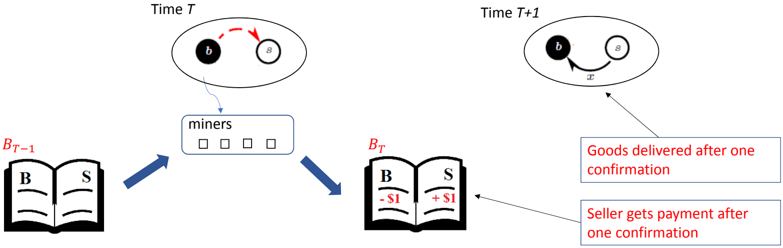
# No confirmation lag (N=0)



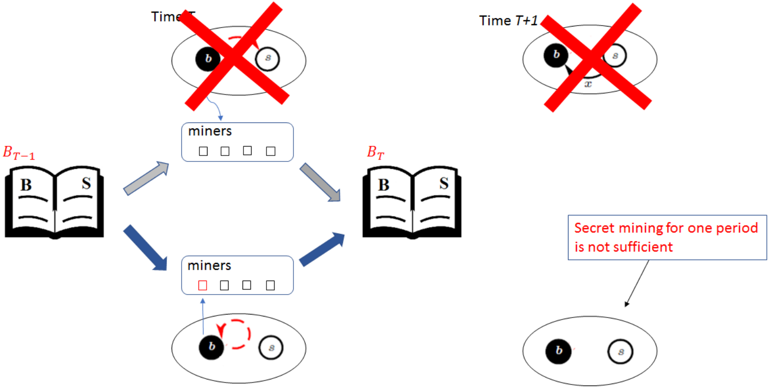
# No confirmation lag (N=0)



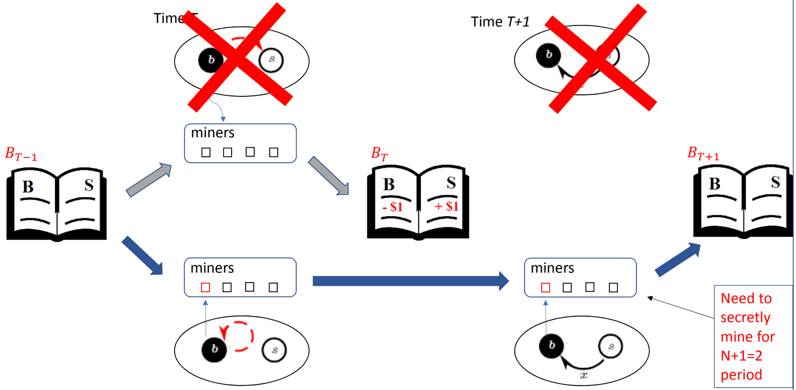
# One confirmation lag (N=1)



# One confirmation lag (N=1)



# One confirmation lag (N=1)



# How Cryptocurrency works

1. Consensus Protocol
   - competition in the form of mining: “miners” compete to update the public ledger (i.e. Blockchain)
   - the prob. of winning is proportional to the fraction of computational power owned by a miner
2. Reward Scheme
   - reward winning miners by seigniorage and transaction fees
3. Confirmation Lags
   - double spending is discouraged by confirmation delay
   - if goods are delivered after $N$ validations are observed, then the buyer needs to win the mining game $N+1$ times to revoke the transaction

# How Cryptocurrency works

1. Consensus Protocol
   - competition in the form of mining: "miners" compete to update the public ledger (i.e. Blockchain)
   - the prob. of winning is proportional to the fraction of computational power owned by a miner
2. Reward Scheme
   - reward winning miners by seigniorage and transaction fees
3. Confirmation Lags
   - double spending is discouraged by confirmation delay
   - if goods are delivered after $N$ validations are observed, then the buyer needs to win the mining game $N+1$ times to revoke the transaction

# Questions

Take as given the design of the cryptocurrency system:

1. How well does it function as a payment system?

2. How to optimally set policy parameters?
   e.g. currency growth, transaction fees

3. How best to use it for different types of transactions?
   e.g. retail vs large value

# Model

# Environment

Based on Lagos and Wright (2005)

Time is discrete: $t = 0, 1, 2, ...$

Three types of agents.

- $B$ buyers
- $\sigma B$ sellers
- $M$ miners

Buyers and seller use balances recorded in a ledger to finance bilateral trade.

Balances in the ledger grow at rate $\mu$ and there are transaction fees $\tau$.

# Proof-of-Work

$M$ miners compete to update the ledger by solving a costly computational task with a random success rate.

Miner $i$ chooses computer power $q_i$ to maximize profits

$$\rho(q_i)R - q_i\alpha$$

where

- $R$ mining reward in real terms
- $\alpha$ price of computer power
- $\rho$ probability of winning given by

$$\rho(q_i) = \frac{q_i}{\sum_{m=1}^{M} q_m}$$

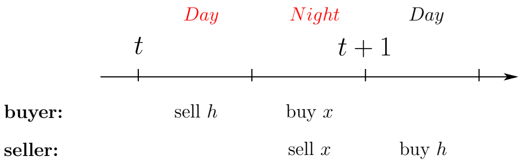
**Results:**

1) Higher $R$ induces higher mining activities $\sum_{m=1}^{M} q_m = MQ$.

2) As $M \to \infty$, all rents $R$ are dissipated.

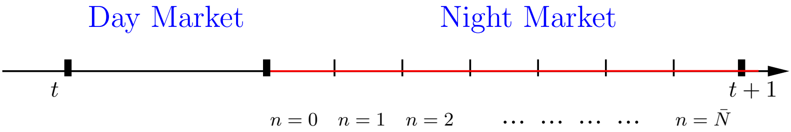
Micro

# Trading




- Preferences
  - Buyer: $\varepsilon u(x_t) - h_t$, where $\varepsilon \sim F$
  - Seller: $-c(x_t) + h_t$
- Trading
  - Day: buyer sells $h$ to acquire balances $z$
  - Night: spends $d \leq z$ to buy $x$ from a seller
  - Next day: the seller uses $d$ to buy $h$

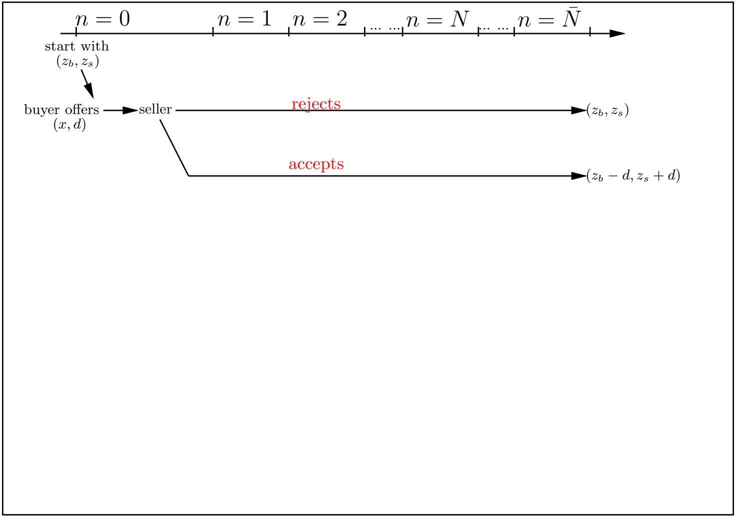
# Night Trading



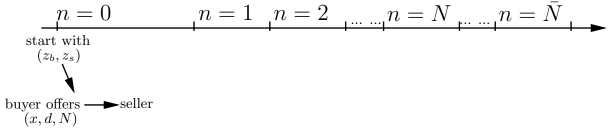

- In session 0, a buyer meets with a seller and makes a take-it-or-leave-it-offer $(x, d, N)$
  - immediate payment $d$ in real balances
  - $x$ goods to be delivered after confirmations of the payment in $N$ consecutive blocks
- After trade, the buyer can attempt to double spend

# Incentives to Double Spend

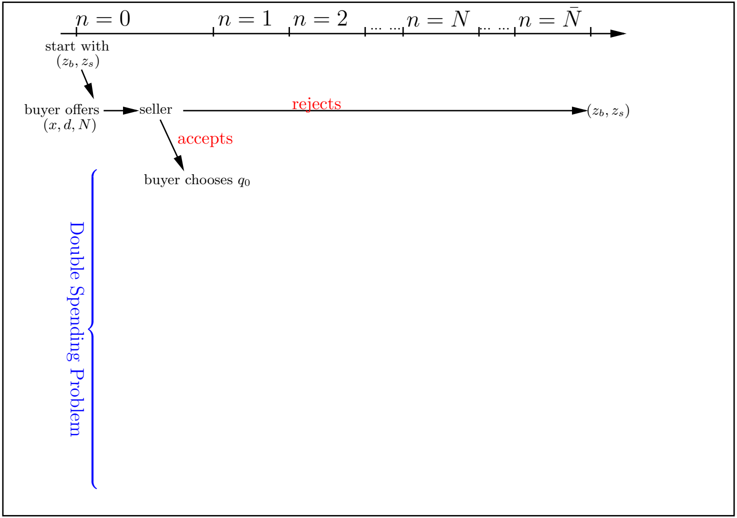
# Transactions in Lagos-Wright



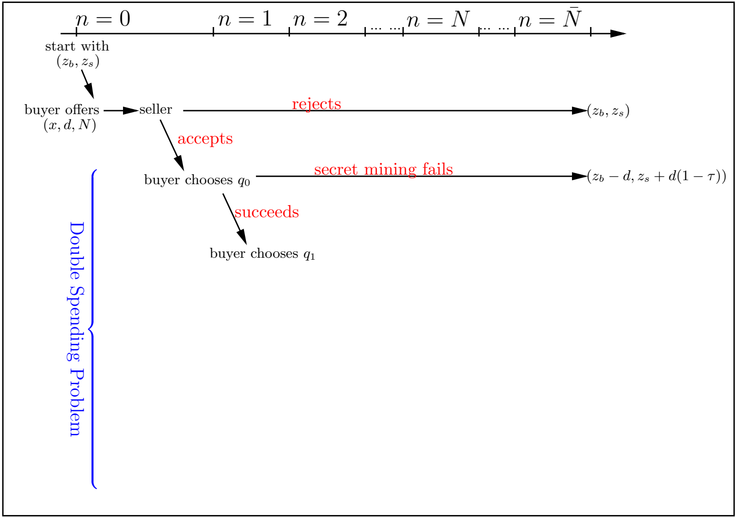
# Double-Spending Problem



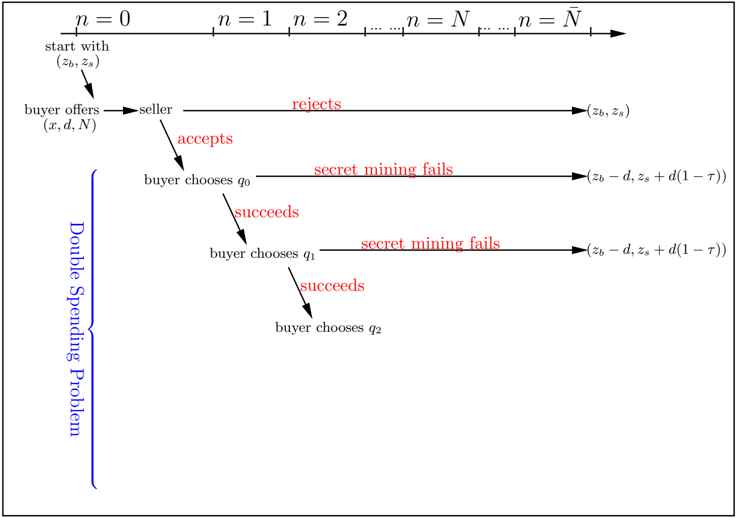
# Double-Spending Problem



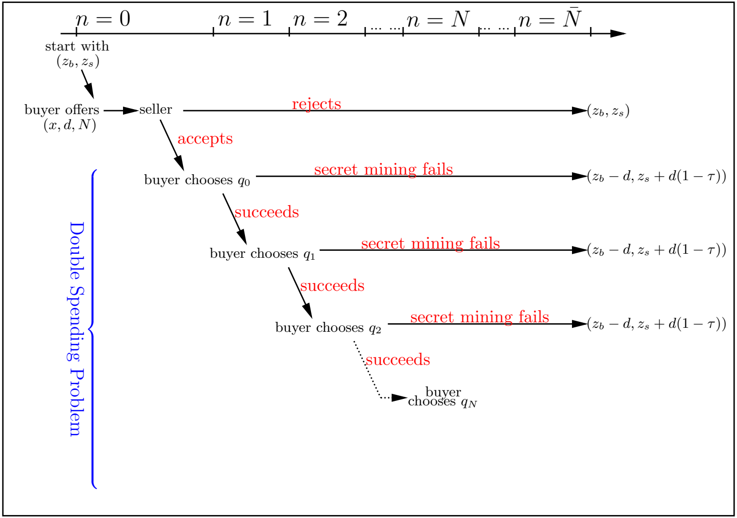
# Double-Spending Problem



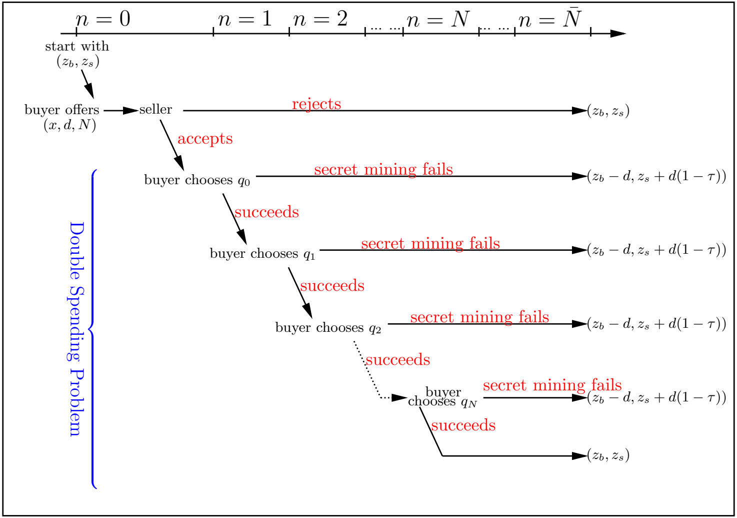
# Double-Spending Problem



# Double-Spending Problem



# Double-Spending Problem



# No Double Spending Constraint

For any contract $(x, d, N)$, the expected payoff from a DS attempt is

$$D_0(d, N) = \max_{\{q_n\}_{n=0}^{N}} P\frac{\beta}{\mu}[d + R(1 + N)] - \sum_{n=0}^{N} \left( \prod_{t=0}^{n-1} \frac{q_n}{QM + q_n} \right) \alpha q_n$$

where

$$P = \prod_{n=0}^{N} \left( \frac{q_n}{QM + q_n} \right) \text{ is the prob. of success}$$

$$R = \frac{Z(\mu - 1) + D\tau}{\bar{N} + 1} \text{ are the rewards form mining}$$

**Lemma**

If $D_0(d, N) = 0$, then the contract $(x, d, N)$ is double-spending proof.

# Double-Spending Proof Contracts

## Proposition

Suppose $M \to \infty$. A contract $(x, d, N)$ is double-spending proof (i.e. settlement is final) if

$$d < R(N+1)N.$$

Otherwise, the settlement is final only with probability

$$1 - P(d, N) = \frac{N+1}{\sqrt{\frac{d}{R} + (N+1)}}.$$

## Results:

- Settlement cannot be both immediate ($N = 0$) and final ($P = 0$).
- Rewards help discourage double spending and improve finality.
- There is a trade-off between trade size $d$, settlement lag $N$ and finality $1 - P$.

# Key Trade-off

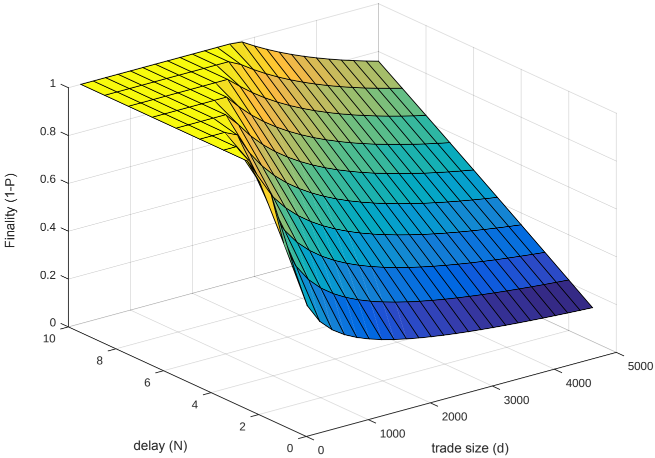



Figure: Trade Size vs. Settlement Lag vs. Finality

# Cryptocurrency Equilibrium

## Definition

A DS-proof cryptocurrency equilibrium with $(\mu, \tau)$ and $M \to \infty$ is given by contracts $(x(\varepsilon), d(\varepsilon), N(\varepsilon))$, money demand $z(\varepsilon)$ and a mining choice $q$ such that

1. the contracts satisfy the No-DS-constraint,
2. the money demand and the offer maximizes a buyer's utility,
3. the mining choice maximizes a miner's utility
4. and markets clear.

## Theorem

▸ Proof

A DS-proof cryptocurrency equilibrium exists for $B$ sufficiently large.

# Optimal Reward Scheme

Define social welfare as

$$\mathcal{W} = \underbrace{B\int[\sigma\delta^{N(\varepsilon)}\varepsilon u(x(\varepsilon)) - x(\varepsilon)]dF_{\varepsilon}(\varepsilon)}_{\text{trade surplus}} - \underbrace{\frac{\beta}{\mu}R(\bar{N}+1)}_{\text{mining costs}}$$

## Proposition

*The optimal reward structure sets transaction fees to zero and only relies on seignorage:* $\tau = 0$ *and* $\mu > 1$*.*

# Optimal Reward Scheme

Define social welfare as

$$\mathcal{W} = \underbrace{B \int [\sigma \delta^{N(\varepsilon)} \varepsilon u(x(\varepsilon)) - x(\varepsilon)] dF_{\varepsilon}(\varepsilon)}_{\text{trade surplus}} - \underbrace{\frac{\beta}{\mu} R(\bar{N}+1)}_{\text{mining costs}}$$

## Proposition

*The optimal reward structure sets transaction fees to zero and only relies on seignorage:* $\tau = 0$ *and* $\mu > 1$*.*

- The reason is that the inflation tax is shared by all buyers while transaction fees are paid only by the active ones who have a high valuation of money.
- ... levying reward costs upfront in terms of inflation allows distortions to be smoothed out across all buyers

# Optimal Reward Scheme

Define social welfare as

$$\mathcal{W} = \underbrace{B \int [\sigma \delta^{N(\varepsilon)} \varepsilon u(x(\varepsilon)) - x(\varepsilon)] dF_{\varepsilon}(\varepsilon)}_{\text{trade surplus}} - \underbrace{\frac{\beta}{\mu} R(\bar{N}+1)}_{\text{mining costs}}$$

## Proposition

*The optimal reward structure sets transaction fees to zero and only relies on seignorage:* $\tau = 0$ *and* $\mu > 1$.

- The reason is that the inflation tax is shared by all buyers while transaction fees are paid only by the active ones who have a high valuation of money.
- ... levying reward costs upfront in terms of inflation allows distortions to be smoothed out across all buyers
- Implication: long-run zero currency growth is suboptimal

# Quantitative Assessment

# Calibration – Basic Parameters

| | values | targets |
|---|---|---|
| $\beta$ | 0.999916 | period length = 1 day |
| $\delta$ | 0.999999 | block time = 10 min |
| $\mu$ | 1.00025 | money growth (9.6% p.a.) |
| $\tau$ | 0.000088 | total fees/vol per block |
| $B$ | 6873428 | max. # of average-sized transactions |
| $\sigma$ | 0.0178 | vol per day/total BTC |
| $\alpha$ | 1 | normalized |

Source: 2015 data from Blockchain.info

- We use log utility.
- We use data on the distribution of transactions.
- Confirmation lags cannot be observed directly.

Details

# 1. Welfare Comparison

| Regime | Welfare Cost as % of consumption |
|---|---|
| Cash (Friedman Rule) | 0% |
| Cash (2% inflation) | 0.003% |
| Bitcoin (benchmark) | 1.410% |
| $\mu - 1 = 9.5\%, \tau = 0.0088\%$ | mining cost: \$359.98 millions |
| Bitcoin (optimal policy) | 0.080% |
| $\mu - 1 = 0.17\%, \tau = 0\%$ | mining cost: \$6.9 millions |

- Welfare loss is currently very large mainly due to the mining cost.
- ... can be reduced substantially by lowering money growth and setting transaction fees to zero.
- Long-run BTC design will bring money growth to 0 and is, thus, inefficient.

# 2. Best Usage of Cryptocurrency Technology

| | Retail Payments (US Debit cards) | Large Value Payments (Fedwire) |
|---|---|---|
| avg transaction size | \$38.29 | \$6552236 |
| annual volume | 59539 millions | 135 millions |
| optimal $\mu$ | | |
| optimal $\tau$ | | |
| confirmation lag | | |
| welfare loss | | |
| mining cost (per year) | | |

- DS-proof iff $d < R \cdot N(1 + N)$
  - retail: small trade size, high volume
  - interbank: large trade size, low volume

# 2. Best Usage of Cryptocurrency Technology

| | Retail Payments (US Debit cards) | Large Value Payments (Fedwire) |
|---|---|---|
| avg transaction size<br>annual volume | \$38.29<br>59539 millions | \$6552236<br>135 millions |
| optimal $\mu$<br>optimal $\tau$<br>confirmation lag | | |
| welfare loss<br>mining cost (per year) | 0.00052%<br>\$4.33 millions | 0.0060%<br>\$22.10 billions |

- DS-proof iff $d < R \cdot N(1 + N)$
  - retail: small trade size, high volume
  - interbank: large trade size, low volume
- retial system incurs a lower welfare loss and mining costs

# 2. Best Usage of Cryptocurrency Technology

| | Retail Payments (US Debit cards) | Large Value Payments (Fedwire) |
|---|---|---|
| avg transaction size | \$38.29 | \$6552236 |
| annual volume | 59539 millions | 135 millions |
| optimal $\mu$ | 0.038% | 0.53% |
| optimal $\tau$ | 0% | 0% |
| confirmation lag | 2 mins | 12 mins |
| welfare loss | 0.00052% | 0.0060% |
| mining cost (per year) | \$4.33 millions | \$22.10 billions |

- DS-proof iff $d < R \cdot N(1 + N)$
  - retail: small trade size, high volume
  - interbank: large trade size, low volume
- retial system incurs a lower welfare loss and mining costs
- ... requires smaller rewards
- ... induces shorter confirmation lags

# What to Take Away

**1)** Owing to its digital nature, a cryptocurrency is fundamentally different from cash.

**2)** One can understand the economics of such a system well by looking at the incentives to double-spend.

**3)** BITCOIN is not only really expensive in terms of mining costs, but also inefficient in its long-run design.

**4)** It provides a more efficient payment system when the volume of transactions is large relative to the individual transaction size.

On-going project: Blockchain for security settlement, cross-border payments, ...

**Thanks!**

# Appendix

# Microfoundations for Mining

Investing computing power $q_m$ allows a miner to solve the PoW problem with probability

$$F(t) = 1 - e^{-\mu_m \cdot t}$$

within a time interval $t$, where $1/\mu_m = D/q(m)$ is the expected time to solve the problem.

Hence, $D$ is the difficulty parameter for the PoW problem.

The first solution among miners, $\min(\tau_1, \ldots, \tau_M)$, is thus also exponentially distributed and the probability for any miner to solve it first is given by

$$\rho_n(q_n) = \frac{q_n}{\sum_{m=1}^{M} q_m}.$$

Back

# Oligopolistic Mining Equilibrium

Maximizing profits by miner $j$ yields as a FOC

$$\left(\frac{\sum_{i=1}^{N} q_i - q_j}{\left(\sum_{i=1}^{N} q_i\right)^2}\right)\frac{\beta}{\mu}R = \alpha$$

Imposing symmetry, we obtain for the total mining cost

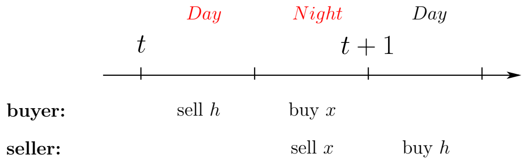
$$C = \alpha M Q = \frac{M-1}{M}\frac{\beta}{\mu}R.$$

For $M \to \infty$ all rents are dissipated and we obtain

$$C = \frac{\beta}{\mu}R$$

Back

# Trading




Two markets

- centralized market in day
- decentralized market at night

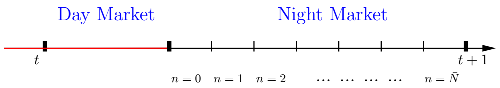
Preferences

- Buyer: $\varepsilon u(x_t) - h_t$, where $\varepsilon \sim F$
- Seller: $-x_t + h_t$

Trading

- Day: buyer sells $h$ to acquire real balances $z$
- Night: spends $d \leq z$ to buy $x$ from a seller
- Next day: the seller uses $d$ to buy $h$

# Day Market




The value of a buyer who draws $\varepsilon$ is

$$\max_{z',h} -h + V(z';\varepsilon)$$

subject to

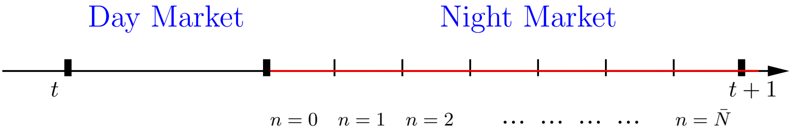
$$h + z \geq z' \geq 0$$

where $z'$ are the real balances carried to the night market.

**Assumption:**

Transactions can be perfectly monitored and there is full liability so that double spending is not a problem.

# Night Market




The night market is divided into $\bar{N}+1$ trading sessions.

- In session 0, a buyer meets with a a seller w.p. $\sigma$ and makes a take-it-or-leave-it-offer $(x, d, N)$.
- There is immediate payment $d$ in real balances.
- $x$ goods are to be delivered after confirmation of the payment in $N$ consecutive blocks.

The offer $(x, d, N)$ determines whether the buyer has an incentive to double spend or not.

Back

# Optimal DS Proof Contracts

At the start of the night market, the buyer with $z$ makes a take-it-or-leave-it offer $(x, d, N)$ to a seller.

The buyer will never carry more real balances than necessary so that $z = d$ and the offer is given by $(x(d), N(d))$.

Requiring the offer to be double spending proof the buyer solves

$$\max_{(x,d,N)} -d + V(d;\varepsilon)$$

subject to

$$V(d;\varepsilon) = \sigma\delta^N \varepsilon u(x) + (1-\sigma)\frac{\beta}{\mu}d$$

$$x \leq \frac{\beta}{\mu}d(1-\tau)$$

$$d \leq R(N+1)N$$

## Sufficient Condition for DS proof

The optimal contract is DS proof if

$$\sigma\left[\delta\varepsilon_{\max}u'(\bar{x})(1-\tau)\mathcal{E}(x)-1\right]<i$$

where

$$\bar{x}=(1-\tau)2R \qquad \text{is the maximum trade size with } N=1$$

$$\mathcal{E}(x)\leq\frac{3}{4} \qquad \text{is the elasticity of } x \text{ w.r.t. } d \text{ at } N=1$$

The reason is that the tightest constraint to avoid DS is a confirmation lag of $N=1$.

This condition is satisfied when

- the opp. cost of carrying balances is high ($i$ is high)
- the matching friction is high ($\sigma$ is low)
- the marginal utility is low ($\varepsilon$ is low)

Back

# Existence Proof

We use Kakutani's Fixed Point Theorem.

Fix $(\mu, \tau)$. The reward $R$ determines the aggregate money supply $S(R)$ which in turn determines total rewards $R'$. Hence, we need to find a fixed point for $R$ given aggregate money demand for a correspondence

$$T(R) = \left( \frac{(\mu - 1) + \sigma\tau}{\bar{N} + 1} \right) S(R).$$

Aggregate money demand can be shown to be u.h.c, convex in $R$ which pins down the aggregate transaction fees and, hence, $R'$.

Furthermore, given $B$ sufficiently large, we can find a lower bound on $R_{\min} > 0$ such that $R > R_{\min}$.

Hence, we can restrict the correspondence to a compact set and show that the correspondence has a closed graph.

Back

# Optimal Contracts

We use data on transactions to recover the implied distribution of $\varepsilon$.

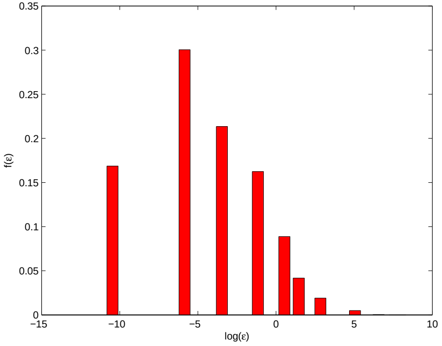



Figure: Implied Distribution of Shocks

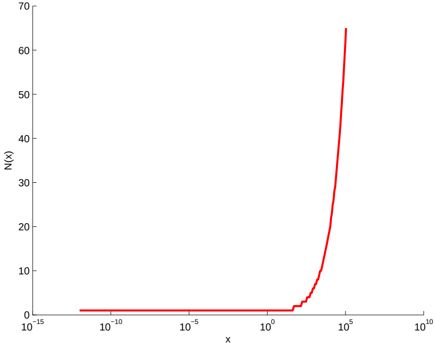
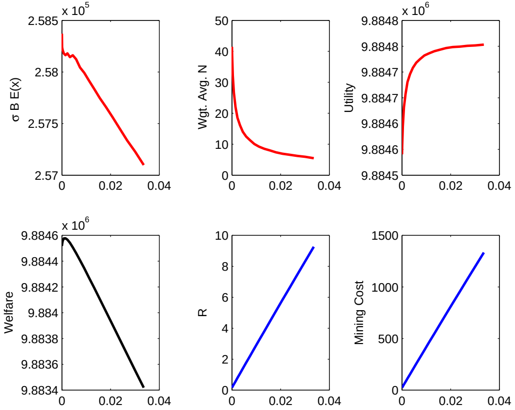



Figure: Optimal Delay

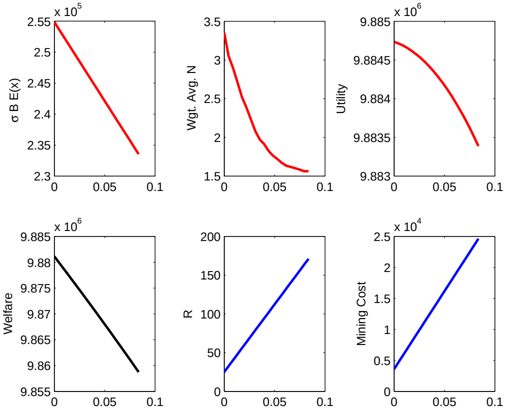
Back

# Optimal Design I – Effects of Money Growth Rate




- Higher inflation implies distortions and higher mining costs ...
- .. but positive inflation is optimal due to lower confirmation lags.

# Optimal Design II – Effects of Transaction Fees




- Same trade-off ...
- ... but zero transaction costs seem to be optimal given $\mu > 0$.