Lesson 01 — Introduction to Cryptoeconomics

Study Notes

Cryptoeconomics — BSc Level

Joerg Osterrieder

2025

Contents

Learning Objectives

By the end of this lesson, students should be able to:

1. **Define** cryptoeconomics and explain how it combines cryptography with economic incentive design.
2. **Describe** the historical origins of Bitcoin, key milestones from Hashcash (1997) to the fourth halving (2024), and the identity of Satoshi Nakamoto.
3. **Explain** the double-spending problem and why it required a fundamentally new solution in a digital peer-to-peer context.
4. **Articulate** the Blockchain Trilemma and give concrete examples of systems that prioritise different combinations of its three properties.

Key Definitions

Cryptoeconomics

Cryptoeconomics

Cryptoeconomics is the interdisciplinary field that uses *cryptographic tools* (hash functions, digital signatures, zero-knowledge proofs) combined with *economic mechanisms* (incentives, game theory, mechanism design) to build decentralised systems whose participants behave honestly without requiring mutual trust.

Cryptoeconomics is not simply "economics applied to cryptocurrency". It is a design discipline: engineers specify a protocol, and rational participants respond to its incentives. The goal is to make *desired behaviour* the dominant strategy for self-interested actors.

Blockchain

Blockchain

A **blockchain** is an append-only, distributed ledger structured as an ordered sequence of *blocks*. Each block contains (1) a set of validated transactions, (2) a cryptographic hash of the previous block, and (3) metadata (timestamp, nonce, difficulty target). The chain of hashes creates tamper-evidence: altering any past block invalidates all subsequent hashes.

The key property is **immutability under rewriting**: to alter block k, an attacker must recompute proof-of-work for blocks $k, k+1, \ldots, n$ (the current tip) faster than the honest network adds new blocks. With $\geq 51\%$ honest hashrate this is computationally infeasible.

Consensus

Consensus

In a distributed system, **consensus** is agreement among n nodes (which may include faulty or malicious participants) on a single value or ordering of events. In Bitcoin, consensus determines which chain of blocks is canonical.

Bitcoin achieves probabilistic consensus: with each additional block mined on top of a transaction, the probability of that transaction being reversed decreases exponentially. Six confirmations (≈ 60 minutes) is the conventional threshold for high-value transactions.

Decentralisation

Decentralisation

A system is **decentralised** to the extent that control, computation, and data storage are distributed across many independent participants rather than concentrated in a single entity. Decentralisation is a spectrum, not a binary property.

Trustless

Trustless

A **trustless** system allows parties to transact without needing to trust each other or a central intermediary. Trust is instead placed in the *protocol rules* and *mathematical guarantees*. Neither party can cheat because the protocol makes cheating unprofitable or impossible.

Permissionless

Permissionless

A **permissionless** network allows any participant to join, validate, and transact without prior authorisation. There is no gatekeeper. Compare with *permissioned* blockchains (e.g., Hyperledger Fabric) that require membership approval.

Double-Spending

Double-Spending

Double-spending is the attempt to spend the same unit of digital currency in two or more separate transactions. Unlike physical cash, digital data can be copied trivially, so preventing double-spending in digital payments had historically required a trusted central clearinghouse (a bank).

Bitcoin eliminates the need for a clearinghouse by having the entire network validate every transaction against a shared, ordered ledger. The first broadcast transaction spending a given output is accepted; a later attempt to spend the same output is rejected.

Core Concepts

The Bitcoin Origin Story

The intellectual precursors to Bitcoin emerged through the *cypherpunk* movement of the 1980s–2000s, a loose community of cryptographers and activists who believed cryptographic tools could protect individual privacy and enable censorship-resistant money.

Key precursors include:

- **DigiCash / eCash** (David Chaum, 1989) — first digital cash scheme using blind signatures; required a central mint.
- **Hashcash** (Adam Back, 1997) — proof-of-work scheme for email spam prevention; later became Bitcoin's mining mechanism.
- **b-money** (Wei Dai, 1998) — proposed a distributed ledger maintained by all participants; never implemented.
- **Bit Gold** (Nick Szabo, 1998–2005) — proposed chained proof-of-work puzzles; very close in concept to Bitcoin.

On 31 October 2008, an anonymous author using the pseudonym **Satoshi Nakamoto** published "Bitcoin: A Peer-to-Peer Electronic Cash System" to a cryptography mailing list. The nine-page whitepaper proposed a solution to double-spending without a trusted third party. On 3 January 2009, Satoshi mined the genesis block, embedding the message:

> *"The Times 03/Jan/2009 Chancellor on brink of second bailout for banks."*

This timestamp proved the block was not pre-mined earlier and served as a political commentary on the existing financial system.

Satoshi communicated via email and forums until mid-2010, then gradually withdrew. The identity of Satoshi Nakamoto — whether an individual or a group — remains unknown.

Blockchain Trilemma

Vitalik Buterin, co-founder of Ethereum, popularised the **Blockchain Trilemma**: the observation that a blockchain system can at most fully achieve two of the following three properties simultaneously.

Property	Definition and Trade-off
Scalability	The system can handle many transactions per second (TPS). Increasing scalability typically requires fewer validators or larger blocks, which reduces decentralisation or security.
Decentralisation	Many independent nodes participate; no single entity controls the chain. High decentralisation demands low hardware requirements, which limits performance.
Security	The network resists attack; a majority coalition cannot rewrite history. Strong security with high decentralisation implies limited throughput.

Example: Trade-off Comparisons

- **Bitcoin**: maximises Security + Decentralisation. Throughput is ≈7 TPS by design. Visa processes ≈24,000 TPS.
- **EOS / DPOS chains**: maximise Scalability + Security by using 21 elected block producers (sacrifices decentralisation).
- **Lightning Network**: a Layer-2 solution that moves scalability off-chain, preserving Bitcoin's base-layer security.

The trilemma is an empirical observation, not a formal theorem. Active research (sharding, rollups, Ethereum's PoS transition) aims to mitigate these trade-offs, though fundamental limits remain.

Peer-to-Peer Networks

Bitcoin operates as an unstructured P2P network. Nodes connect to peers, propagate transactions and blocks via gossip protocol, and each maintains an independent copy of the blockchain. There is no central server. Key properties:

- **Resilience**: removing any single node does not disrupt the network.
- **Censorship resistance**: no single node can block a transaction from being broadcast.
- **No single point of failure**: the network has survived thousands of nodes going offline simultaneously.

Incentive Design

Bitcoin's security rests on aligning miner incentives with honest behaviour:

- Miners expend real electricity to compute proof-of-work.

- The block reward (newly created BTC + transaction fees) compensates honest miners.
- A miner who produces an invalid block receives no reward — the network simply rejects the block.
- A miner with $> 50\%$ hashrate *could* double-spend, but the opportunity cost exceeds the gain (they sacrifice future block rewards).

This is mechanism design in practice: the protocol transforms individual self-interest into collective security.

Bitcoin Supply and Halving Schedule

Bitcoin's monetary policy is encoded in the protocol and cannot be changed without a hard fork:

Halving	Block Height	Approx. Date	Block Reward (BTC)
Genesis	0	Jan 2009	50.000
1st	210,000	Nov 2012	25.000
2nd	420,000	Jul 2016	12.500
3rd	630,000	May 2020	6.250
4th	840,000	Apr 2024	3.125
⋮	⋮	⋮	⋮
≈32nd	6,720,000	≈2140	$< 10^{-8}$ (effectively 0)

Total supply: $\sum_{k=0}^{\infty} 210{,}000 \times \frac{50}{2^k} = 210{,}000 \times 50 \times 2 = 21{,}000{,}000$ BTC.

After all BTC are mined, miner revenue will consist entirely of transaction fees, raising open questions about long-run security.

The Byzantine Generals Problem

Byzantine Generals Problem (Lamport

n generals must coordinate an attack by communicating only via messengers. Up to f generals are *traitors* who send contradictory messages. **Result**: consensus is achievable if and only if $n \geq 3f + 1$, i.e., fewer than one-third of participants are faulty.

Bitcoin does not directly implement classical BFT algorithms (which require known identities and $O(n^2)$ message complexity). Instead, Proof of Work provides a *Sybil-resistance mechanism*: creating a new identity is costly (requires hashrate), so an attacker controlling $< 50\%$ of hashrate cannot outvote the honest majority.

Worked Examples

Example: Halving Reward Calculation

Question: What is the block reward after the 5th halving?

Solution:

Initial reward: $R_0 = 50$ BTC

After k halvings: $R_k = \dfrac{50}{2^k}$ BTC

At $k = 5$: $R_5 = \dfrac{50}{32} = 1.5625$ BTC

The 5th halving will occur at block height 1,050,000, approximately 2028.

Example: Total Supply Calculation

Question: Show that the total Bitcoin supply is exactly 21,000,000.

Solution: Each halving epoch lasts 210,000 blocks.

$$S = \sum_{k=0}^{\infty} 210{,}000 \times \frac{50}{2^k} = 210{,}000 \times 50 \sum_{k=0}^{\infty} \frac{1}{2^k}$$
$$= 10{,}500{,}000 \times \frac{1}{1 - \frac{1}{2}} = 10{,}500{,}000 \times 2 = 21{,}000{,}000 \text{ BTC}.$$

This uses the geometric series $\sum_{k=0}^{\infty} r^k = \frac{1}{1-r}$ for $|r| < 1$ with $r = \frac{1}{2}$.

Example: Double-Spend Attack Scenario

Scenario: Alice wants to buy a laptop worth 1 BTC. She broadcasts a transaction T_1 paying the merchant. Simultaneously she broadcasts T_2 paying herself (double-spend). How does Bitcoin prevent this?

Solution:

1. Both T_1 and T_2 reference the same UTXO (unspent output).
2. Miners include the first transaction they receive in their candidate block. Suppose T_1 is included in block B.
3. Once B is confirmed and buried under further blocks, the UTXO referenced by T_2 no longer exists in the UTXO set.
4. Any future block including T_2 would be invalid and rejected by the network.

The merchant should wait for ≥ 1 confirmation before delivering high-value goods.

Common Pitfalls

Common Pitfall

Misconception: "Blockchain and Bitcoin are the same thing."
Correction: Bitcoin is one application built on a blockchain. Blockchain is the underlying data structure and protocol design pattern. Many blockchains exist (Ethereum, Litecoin, Solana) and many use cases have nothing to do with currency (supply chain, voting, NFTs).

Common Pitfall

Misconception: "Trustless means there is no trust at all."
Correction: Trustless means trust is not required *between parties*. Trust is still placed in the protocol rules, the cryptographic primitives, and the assumption that a majority of miners are honest. The term is shorthand for "requires no trusted intermediary".

Common Pitfall

Misconception: "Bitcoin transactions are anonymous."
Correction: Bitcoin is *pseudonymous*. All transactions are publicly recorded on the blockchain. Addresses are not directly linked to real identities, but chain analysis can often de-anonymise users, especially when they interact with KYC exchanges.

Common Pitfall

Misconception: "The Blockchain Trilemma means you can only pick two of three properties."
Correction: The trilemma describes a tendency, not a mathematical impossibility. Layer-2 solutions, sharding, and other advances reduce (but do not eliminate) these trade-offs. The trilemma is a useful heuristic for evaluating design choices.

Review Questions

1. **(Recall)** State the year the Bitcoin whitepaper was published and give its full title.
2. **(Recall)** What message did Satoshi embed in the genesis block, and why is it significant?
3. **(Comprehension)** Explain the double-spending problem in your own words. Why is it trivially solved by a bank but hard to solve in a decentralised setting?
4. **(Comprehension)** Distinguish between "trustless" and "no trust required anywhere." What *is* trusted in Bitcoin?
5. **(Application)** Using the halving formula, calculate the block reward after the 6th halving. In what approximate year will this occur?
6. **(Application)** A blockchain designer wants a system with high throughput and strong security guarantees. According to the Blockchain Trilemma, what property must they sacrifice? Give a real-world example of a system that makes this trade-off.

7. **(Analysis)** The Byzantine Generals Problem requires $n \geq 3f + 1$ for consensus with f faulty nodes. Bitcoin uses Proof of Work instead of classical BFT. What assumption does Bitcoin make instead of the $n \geq 3f + 1$ bound? Under what condition does Bitcoin's security guarantee break down?

8. **(Synthesis)** Satoshi chose not to include a Turing-complete scripting language in Bitcoin. Ethereum's Vitalik Buterin did include one. Discuss one advantage and one disadvantage of each approach from a cryptoeconomics perspective.

Further Reading

- Nakamoto, S. (2008). *Bitcoin: A Peer-to-Peer Electronic Cash System*. https://bitcoin.org/bitcoin.pdf
  The original nine-page whitepaper. Read it in full — it is remarkably accessible.

- Antonopoulos, A. M. (2017). *Mastering Bitcoin* (2nd ed.). O'Reilly Media. Chapters 1–2 cover the introduction and how Bitcoin works. Available free online: https://github.com/bitcoinbook/bitcoinbook

- Buterin, V. (2014). *Ethereum Whitepaper*. https://ethereum.org/en/whitepaper/
  Introduces the concept of a Turing-complete blockchain and discusses the limitations of Bitcoin Script.

- Lamport, L., Shostak, R., & Pease, M. (1982). The Byzantine Generals Problem. *ACM Transactions on Programming Languages and Systems*, 4(3), 382–401.
  The foundational paper on fault-tolerant distributed consensus.

- Back, A. (2002). *Hashcash — A Denial of Service Counter-Measure*. http://www.hashcash.org/papers/hashcash.pdf
  Describes the proof-of-work mechanism that Bitcoin's mining is based on.

- Narayanan, A., Bonneau, J., Felten, E., Miller, A., & Goldfeder, S. (2016). *Bitcoin and Cryptocurrency Technologies*. Princeton University Press. Free PDF: https://bitcoinbook.cs.princeton.edu/
  Comprehensive academic textbook; Chapters 1–3 are most relevant to this lesson.