

Financial applications of blockchains and distributed ledgers

Master's program in Financial Engineering

Jiahua (Java) Xu

Session 1



Housekeeping

Time and venue

Three sessions: 15:15 – 16:10, 16:25 – 17:20, 17:35 – 18:30

Tuesdays, on Zoom, <https://epfl.zoom.us/j/4897861984>

Preliminary agenda

- (1) 2020-09-15: Blockchain fundamentals
- (2) 2020-09-22: Case study, On-, cross- and off-chain activities
- (3) 2020-09-29: Smart contract with Ethereum, hands-on session
- (4) 2020-10-06: Research seminar
 - ▶ **Andreas Richardson** (Imperial College London): Carbon Trading with Blockchain
 - ▶ **Juan Ignacio Ibañez** (Catholic University of Cordoba): Accounting with Blockchain
 - ▶ **Simon Sive** (UCL): Real estate tokenization
- (5) 2020-10-13: Industry seminar
 - ▶ **Adrien Treccani** ([Metaco](#))
 - ▶ **Ilya Volkov** ([YouHodler](#))
 - ▶ **Gaspard Peduzzi** ([APWine](#))
- (6) 2020-10-20: Group presentation

Lessons learned from previous year

Excellent teacher with a deep understanding of the subject. Interesting format with invited guests every other lecture. Rigorous and demanding final project. Overall excellent class with an implicated teacher which really cares about the students and shows it.

Great teacher

I enjoyed the course, both the teaching assistant and the professor are really nice and helpful people. I enjoyed the guest speakers and I think they were really useful. Probably at the beginning of the course there is already a difference in knowledge amongst students, but despite this fact everyone is able to get something from the lectures and the project is useful to look deeper into a topic.

It is nice to see a teacher that involved in their course! Also the guest lectures are a plus that make the whole class more dynamic.

Java devoted a lot to this course. The course was well organized and the content was rich enough. As for the lectures, she made them appealing and easy to understand. Besides, she is always aside to help students, e.g. answering their questions, providing helpful feedback, etc. Also, the invited guests speaking were interesting and the topics were closely related to the course itself. I would say it's a very good introductory course to the blockchain.

Le cours est intéressant, varié et permet de découvrir le concept de la blockchain. Il faudrait peut-être ajouter une partie sur les applications à la finance. Le projet de recherche est une méthode d'évaluation un peu floue qui demande un travail assez spécial. C'est assez compliqué de faire une recherche conséquente pour les 2 crédits alloués.

One of the best course I had

Overall information of course brings good value. Really enjoyed some guest speakers during class. Also liked building my own smart contract, would be nice to do such applications in more than 1 lecture and explore different topics from a slightly more technical standpoint.

Really nice teacher with a talent for making students interested and involved in class. The project was very enjoyable and we were really free to work on the things that we liked which was motivating and stimulating. I liked that we had intervenants and an apero.

The course given by Jiahua Xu is excellent, thanks to the following points: - I like how the course was set-up: few weeks of introduction to Blockchain technology to get everyone on-board and having the first guest (Adrien) also giving an intro but from another stand-point made things easier to understand. Continuing by having experts that have implemented the technology in their own ideas. And finally presenting your own idea to the class, with excellent feed-back. I think this all makes perfect sense. - We are encouraged to ask questions and we are incentivized to participate. That makes the course super interactive and engaging. - Way of examination is great: first draft of the project presented orally allows to have feed-back and indicates which direction to follow. Then midterm report allows to have more pushed feed-back, before the final report, which is due in beginning of December (which is very welcome since it's before the end-of-semester deadlines from other courses). To sum up: do not change the way of evaluation and the dates. It's fair (for 2ECTS) and convenient. Where the course has room for improvement: - Drawings are essential, in my opinion, to understand blockchain technology. Adrien's drawing of different block chains competing to for consensus was a good example of that. Jiahua could make an effort to have clearer and more thought of drawings. - 3h30-4 hours of lecture with only 1 15min break for a 2 credit course is a bit much. However, since it's only on half of the semester, it makes sense. Therefore, being explicit about it at the beginning of the year would help: "the lecture are going to be 3 to 4h long, the counter part is that they're only going to last have the semester so you have time to focus on your project" - More solidity and Dapp coding would be appreciated, even though I know we were supposed to do more initially. In the end, I have to say, I rarely was so much into a course. Spending time making concrete a vague project and transform that into a reasonable white-paper has been an incredible experience. Thanks for providing us with such an opportunity

The course is well-structured, the lectures as well as the presentations of the speakers are very interesting. Both the Professor and the TA are extremely helpful and always available for clarifications, especially on the project. One small issue is that the initial level of knowledge in the class is very heterogeneous, therefore at the beginning the learning speed differs from one to another.

Very interesting. Lots of interesting viewpoints and guest speakers. Good support provided for project, though workload high for credits awarded.

Figure 1: Students' feedback are taken to heart.

Assessment methods

- ▶ 75% Group essay
- ▶ 25% Group presentation
- ▶ *Bonus* In-class and take-home assignments

It is not just any group project ...

Previous students' work

- ▶ **Matthieu Baud, Henry Decléty, Hugo Roussel:** *Jack the Rippler* won [Ripple's Block-Sprint Hackathon](#) (GBP 3,000 award)
- ▶ **Andreas Richardson:** *Carbon Trading with Blockchain* presented at the [International Conference on Mathematical Research for Blockchain Economy](#), to be published by [Springer](#)
- ▶ **Yanan Liu:** *Libra's impact on world economy* published on the Medium Channel of UCL Centre for Blockchain Technologies (CBT)
- ▶ **Lucas Froissart, Ana Frei:** *EUREKAsh* received the highest score of the class

Definition of blcockchian

IBM and Oliver Wyman:

Blockchain is a shared, distributed ledger that facilitates the process of recording transactions and tracking assets in a business network. An asset can be tangible — a house, a car, cash, land — or intangible like intellectual property, such as patents, copyrights, or branding. Virtually anything of value can be tracked and traded on a blockchain network, reducing risk and cutting costs for all involved.

Wikipedia:

A blockchain, originally block chain, is a growing list of records, called blocks, that are linked using cryptography. Each block contains a cryptographic hash of the previous block, a timestamp, and transaction data (generally represented as a Merkle tree).

Oxford:

a system in which a record of transactions made in bitcoin or another cryptocurrency is maintained across several computers that are linked in a peer-to-peer network

What defines “blockchain”?

Properties

- ▶ Trustless?
- ▶ Distributed?
- ▶ Decentralized?
- ▶ Peer-to-peer?
- ▶ Tamper-proof?

Components

- ▶ Cryptographic hash?
- ▶ Timestamp?
- ▶ Block?
- ▶ Chain?

Purpose

- ▶ Record transactions?

Bitcoin: A Peer-to-Peer Electronic Cash System (Nakamoto 2008)

What problem(s) does a blockchain solve?

Why was there centralization in the first place?

Centralization of payment system

- ▶ 1990s, standards for protocol-level encryption just emerged
- ▶ Lack of trust
 - ▶ Security
 - ▶ Privacy
- ▶ Birth of the intermediary architecture

Centralization of monetary system

- ▶ “Fiat” currency: trust in the government
 - ▶ IOU from a bank vs IOU from a stranger
 - ▶ Gold standard: 1 oz gold = 35 USD

It made things easier — because you only have to trust one entity.

Is a central authority still *necessary* today?

Possibility to abandon central authority

- ▶ Security mechanisms advanced
- ▶ Payment scheme was lagging behind

But can we just leave it?

Necessity [?] to abandon central authority

Monetary system

- ▶ Abandonment of the gold standard
- ▶ Capital control
- ▶ Hyperinflation: Zimbabwe
- ▶ Financial crisis -The Times 2009-01-03: [Chancellor on Brink of Second Bailout for Banks](#)
 - ▶ New York Times 2020-04-09: [With \\$2.3 Trillion Injection, Fed's Plan Far Exceeds Its 2008 Rescue](#)

Payment system

- ▶ Privacy

What problem does a blockchain cause?

Latency

Double spending

Double spending is a problem native to a distributed network due to latency, but less problematic in a centralized one.

Detecting double spending

Imagine:

- ▶ You have \$100 in your bank account
- ▶ You issue to merchants A and B *each* a \$100 check

What will happen?

Preventing double spending in a centralized system

Law enforcement: You'll be punished if you double spend.

Transparency -> Privacy issues

No real ID needed – only a BTC address

Does pseudonymity suffice for privacy protection?

UTXO (Unspent transaction output)

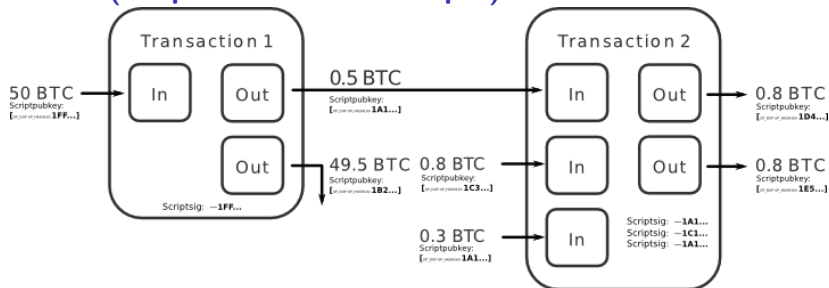


Figure 2: UTXO schema

Hash	597d00408bd38c3e3ccabbdb9263d0ace542e4adc7f0f93a52c... 2011-10-14 13:34	
	<div>13e7yqfcbHYUi5D5mKnc5Erhfq89vB4G7E 35.00000000 BTC</div> <div>➡</div> <div>1LNWw6yCxxUmkhArb2Nf2MPw6vG7u5WG7q 35.00000000 BTC</div>	
Fee	<div>0.00000000 BTC (0.000 sat/B - 0.000 sat/WU - 224 bytes)</div> <div>35.00000000 BTC</div>	

Figure 3: UTXO transaction: input -> output

Hash	83132be7f191a0f7e47763ebb30bb36a7d3d940bab7582de71648f3...				2020-09-15 10:27
	1JufTavYWeL2RtWAqVfahKUcjTQ5vcfbGG 18.17349203 BTC		➡	1Q3BjrT4t7GriHXZpiP9Hg4jr7sjrJPaoT 1.87509200 BTC	
	1JufTavYWeL2RtWAqVfahKUcjTQ5vcfbGG 8.42100934 BTC			1JufTavYWeL2RtWAqVfahKUcjTQ5vcfbGG 24.71805737 BTC	
Fee	0.00135200 BTC (363.441 sat/B - 90.860 sat/WU - 372 bytes)				26.59314937 BTC
					3 Confirmations

Figure 4: UTXO transaction: input 1 + input 2 -> output





Hash	557c87f20583fe28d5c14e2eaf0973af35add4e80318044c01ff9... 		2011-10-14 13:34	
	<div>1LNWw6yCxxUmKhArb2Nf2MPw6vG7u5WG7q 35.00000000 BTC </div> ➔ <div>1Eym7pyJcaambv8FG4ZoU8A4xsiL9us2zz 1.00000000 BTC </div> <div>19f5fNB8Qvia3qLmXzUbVQLwJ35xafoTze 34.00000000 BTC </div>			
Fee	0.00000000 BTC (0.000 sat/B - 0.000 sat/WU - 257 bytes)		35.00000000 BTC	

Figure 5: UTXO transaction: input -> output 1 + output 2

Time-stamping with a paper-based ledger

- ▶ Transactions entered one after another in the notebook, with no pages left blank
- ▶ The notebook is then reviewed and stamped on a regular basis by a notary public

Now the order of transactions are guaranteed, what about the content of transactions?

Haber and Stornetta's scheme

Time-stamp the data itself! (Haber and Stornetta 1991)

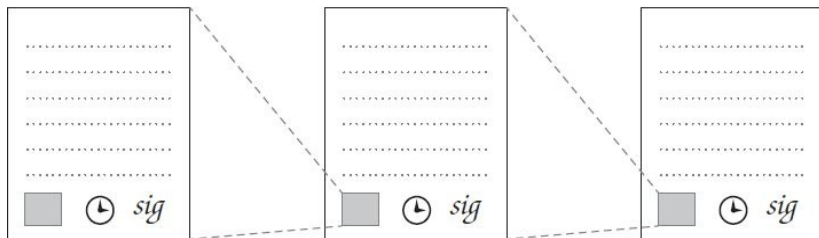


Figure 6: Linked timestamping. To create a certificate for a document, the timestamp server includes a hash pointer to the previous document's certificate and the current time, and it signs these three data elements together.

Why using *blocks*?

- ▶ Haber and Stornetta's original idea was to link documents individually, but changed that into a hybrid method of linking blocks in order to *decrease the time of verifying*.
- ▶ Bitcoin uses multiple transactions in the same block as an optimization.

Trade off between verification and speed.

Digital Signature

Requirement

1. Unforgeability: Only you can make it
2. Verifiability: Everybody else can easily verify it

Scheme

Signer:

Generate key pairs: private (secret) key and public key.

generateKeys is a random function!

$$\textit{generateKeys} : \textit{keysize} \rightarrow (sk, pk)$$

Sign the message:

$$\textit{sign} : (sk, \textit{message}) \rightarrow \textit{sig}$$

Verifier:

The message gets verified:

$$\text{verify} : (pk, message, sig) \rightarrow isValid$$

Note:

$$\text{verify}(pk, message, \text{sign}(sk, message)) == \text{True}$$

What if somebody else gets the same key as me?

The probability of getting the same 256-key is very small so that we don't have to worry about it.

Crucial: Random algorithm used to generate those keys have to be “random”, not predictable!

Unforgeability game:

Attacker:

1. Sees the public key pk^* (but not the private key sk^*), and also functions *generateKeys*, *sign* and *verify*
2. Many messages/documents $\{m_1, m_2, m_3, \dots\}$ of his choice and corresponding $\{sig_1, sig_2, sig_3, \dots\}$
 - ▶ Many: a plausible number, e.g. polynomial function of the key size
3. Now try to sign a new given message M by producing sig'

Verifier:

- ▶ run $verify(pk^*, M, sig')$

A signature scheme is unforgeable iff, no matter what algorithm the attacker uses, his chance of successfully forging a message is extremely small.

Bitcoin uses:

Elliptic Curve Digital Signature Algorithm (ECDSA)

Cryptographic hash functions

Hash function

- ▶ Input: any string of any size.
- ▶ Output: fixed-sized (256-bit)
- ▶ Efficiently computable

To be cryptographically secure, a hash function must have the following 3 properties:

- ▶ collision resistance
- ▶ hiding
- ▶ puzzle friendliness

Collision resistance

A hash function H is said to be collision resistant if it is **infeasible** to find two values, x and y , such that $x \neq y$, yet $H(x) = H(y)$.

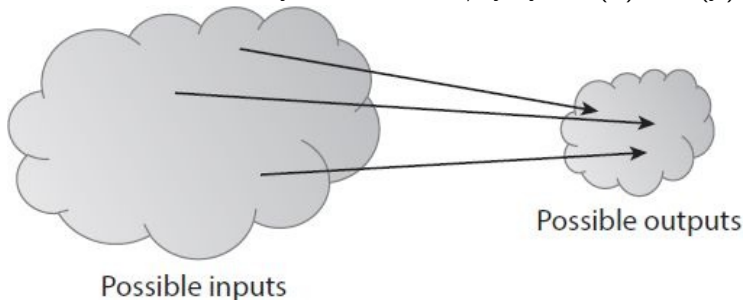


Figure 7: Inevitability of collisions. Because the number of inputs exceeds the number of outputs, we are guaranteed that there must be at least one output to which the hash function maps more than one input.

A lame (not cryptographically secure) hash function:

$$H(x) = x \mod 2^{256}$$

Hiding

A hash function $H()$ is said to be hiding if when a secret value r is chosen from a probability distribution that has high **min-entropy**, then, given $H(r\|x)$, it is infeasible to find x .
where $\|$ is a concatenation operator.

$$\mathcal{H}_{\min(G)} = \min_{a \in \text{Range}(G)} \underbrace{\log_2 \left(\frac{1}{\mathbb{P}[a]} \right)}_{\substack{\text{\# bits to encode most likely password } x \\ \text{\# bits to encode password } x}}$$

Note: r can be our serial number!

Puzzle friendliness

A hash function H is said to be puzzle friendly if for every possible n -bit output value y , if k is chosen from a distribution with high min-entropy, then it is infeasible to find x such that $H(k||x) = y$ in time significantly less than 2^n .

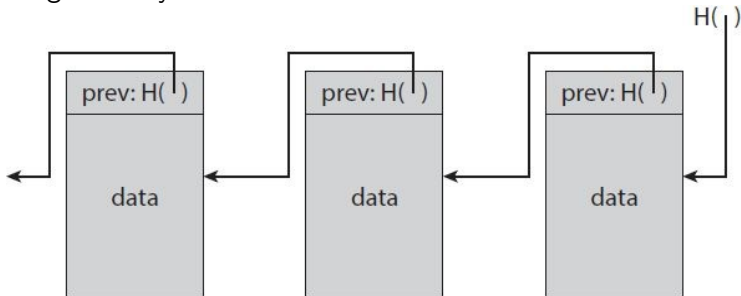


Figure 8: Block chain: a linked list that is built with hash pointers instead of pointers.

Bitcoin uses:

SHA-256 (Secure Hash Algorithm 256)

Distributed consensus protocol

- ▶ n nodes that each has an input value.
- ▶ Some of these nodes are faulty or malicious.
- ▶ A distributed consensus protocol has the following two properties
 1. It must terminate with all honest nodes in agreement on the value.
 2. The value must have been generated by an honest node.

The majority rule?

Sybil attack

Attacker copies of nodes that a malicious (enabled by pseudonymity)

Bitcoin consensus algorithm (simplified)

Proof-of-Work

1. New transactions are broadcast to all nodes.
2. Each node collects new transactions into a block.
3. In each round, a random node gets to broadcast its block.
 - ▶ random: not selected, but through competition (**work**)
4. Other nodes accept the block only if all transactions in it are valid (unspent, valid signatures).
5. Nodes express their acceptance of the block by including its hash in the next block they create.

What do miners compete to solve?

Thank you!

Contact

Jiahua (Java) Xu

Ecole Polytechnique Fédérale de Lausanne (EPFL)

EXTRA 249 (Extranef UNIL)

Quartier UNIL-Dorigny

jiahua.xu@epfl.ch

References I

Haber, Stuart, and W.Scott Stornetta. 1991. "How to time-stamp a digital document." *Journal of Cryptology* 3 (2): 437–55.

<https://doi.org/10.1007/BF00196791>.

Nakamoto, Satoshi. 2008. "Bitcoin: A Peer-to-Peer Electronic Cash System." www.bitcoin.org.