I.

1. Monetary Theory Basics

- money: medium of exchange(a), unit of account(b), store of value(c) – without government intervention

a) if other people, do it, you do it too, if not, you don't either

a) examples: jewelry, currency, metal money (gold), cigarettes, shells, millstones on yap(stones as monetary units)

b) measuring prices

b) describe the price in terms of the dominant medium of exchange

c) saving allows you to smoothen your consumption over time

Properties of money: storability, transferability (you don't want stones obviously), divisibility fungibility (identification characteristics), verifiability, scarcity, low price volatility

Monetary value: intrinsic value (material value of the good), promise of payment (subject to issuer risk), liquidity premium (option to trade the monetary unit for goods)

2. Payment Systems

- cash payments: digital cash (not so efficient/ safe, everyone can copy-paste the proof that you own the money), physical cash

- 3rd party (intermediaries) that keep money registries

- closed loop system: direct relationship with the payer and the payee, ex: PayPal, Western Union

- open loop system: whenever you pay someone, it goes through multiple intermediaries. Processing time is >, can include fees.

- network ex: credit card networks (Visa, Mastercard), swift

3. Monetary Control Structures

- creation (monopolistic, centralized), transaction processing, representation

- physical representation (physical monetary units are tied to an object)

- virtual representation (virtual monetary units allow the transfer of value without any change in physical control of an object)

- transaction processing: centralized (with intermediaries), decentralized (direct with the payee)

4. Bitcoin Primer

- created competitively, virtual representation, decentralized management

- system is maintained by its participant, no need for 3rd parties.

- bitcoin unit, asymmetric cryptography, bitcoin network, bitcoin protocol, blockchain

- transaction capacity: participant can initiate a transaction without censorship

- transaction legitimacy: ensure transaction authenticity and integrity (that the current owner of the funds was the one who initiated the transaction).

- transaction legitimacy: public-private key pair. Private key is created without an intermediary, can derive public key. Information encrypted with one key can only be decrypted with the other key. Private key MUST be kept SECRET.

- transaction consensus: deciding which legitimate transactions are valid

- transactions are bundled into blocks., that are sequentially linked = Blockchain

- bitcoin miners = network participants that assemble candidate blocks.

II.

1. Peer-to-Peer Networks
   - Can change dynamically
   - If one of the nodes goes down, the network will find new ways compared to a centralized network, where if the source goes down, they all go offline
   - When you want to connect to the network you need some IP address (to know someone who is part of the network) to make the connection. After the connection is established, you exchange certain messages to get a two-way connection(handshake). After this you can share resources.
   - Advantages: key advantages: no single node is critical for a functioning network, no node has special privileges or powers, no centralized infrastructure requirement (no need to rent out service or no need for computing capacity in the cloud).
   - Problems: a node can share malicious or irrelevant resources, limited by the nodes with the weakest hardware.

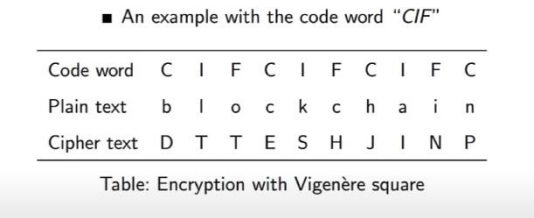
2. Bitcoin Network
   - Is a p2p network, all participants are called nodes
   - After the initial handshake, we requesr a list of IP addresses and connect to additional peers.
   - Full node = fully autonomous node (does not rely on anyone else), core functionality: copy of ledger, verify, relay; optional functionality: wallet, mining
   - Verify new blocks (if transactions are valid)
   - Wallet functionality: 1. Encrypted keystore, 2. Balance (request balance of certain addresses, how much bitcoin you have), 3. GUI (easy way to interact with the blockchain to issue new transactions)
   - Mining functionality: 1. Assemble new blocks (knows the rule set in what structure you must assemble a block, the limitations etc.), 2. Compute (computational resources)
   - Node distribution: 50% in EU, 20% in USA

- Mobile options: it's hard to run a full node on a cellphone bcs of the: storage capacity, bandwidth, battery

III.

1. Hash Functions
   - It is a mapping algorithm
   - Used in: data protection, verification and authentication, proof-of-work, error detection
   - Modular arithmetic: you do a division, but only care about the reminder
   - Checksums: find human errors in data entry (for example IBAN nr, vehicle identification numbers in US and Canada)

2. Symmetric Cryptography
   - It's a branch of secret writing (you have some information that needs to be hidden):steganography, cryptography: transposition (change the position within the information), substitution(using other information packages to substitute from the plain text to the encrypted message): code(each words maps to another), cipher(rule set)
   - Decryption is inverse algorithm of encryption
   - Monoalphabetic substitution: shift in the alphabet by x positions, x is between 1-25. Ex: Principle of the Caesar Cipher, for example if x is 3, A in the plain alphabet gets mapped to D in the cipher alphabet.
   - Polyalphabetic substitution: code word CIF, use repetition of code word as starting point, find length of code word and use frequency analysis on every x-th cipher letter



   - Encryption in the age of the computer: electronics are much faster than mechanical parts, key difference: numbers vs letters (ascii)
   - Data encryption standard: companies need a STANDARDIZED approach.

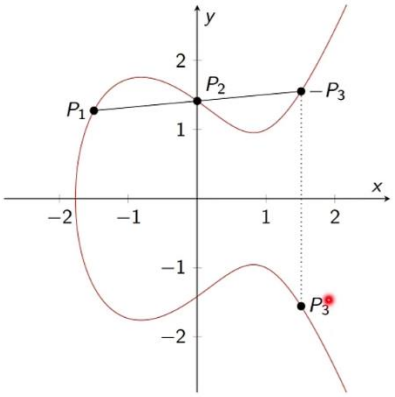
3. Asymmetric Cryptography
   - Asymmetric encryption: key to encrypt and key to decrypt a message are not identical
   - How: use the recipients public key to encrypt the message (the recipient is the only party in possession of the corresponding private key; thus they are the only party that can decrypt it and also it can be used for integrity and authenticity). Encrypt the message with the private key, and everyone else can use the public key to decrypt the message.
   - Steps: 1. Find public key, 2. Encrypt message, 3. Derive private key 4. Decrypt message

4. Elliptic Curves and ECDSA

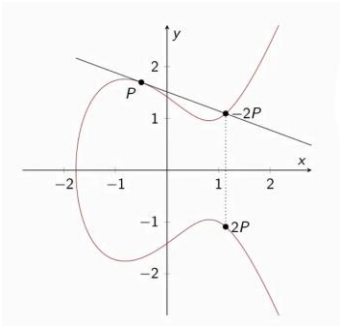
Weierstrass equation:

$$y^2 = x^3 + ax + b$$

- Defined by :
- Adding two points: P1, P2, P3=P1+P2



- Point doubling P + P = 2P



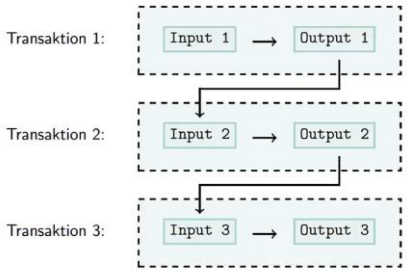
- Modular multiplicative inverse:
  Regular division: 10/4 = 2.5
  Multiplicative inverse: 4/4 = 4 * 4 to the power of -1 = 1
  10 * 4 to the power of -1 = 2.5
  Modular multiplicative inverse: {4 * x} (mod 3) = 1
  For x = 1 because 4 (mod 3) = 1
- ECDSA:

  I had problems understanding the material at this chapter

IV.

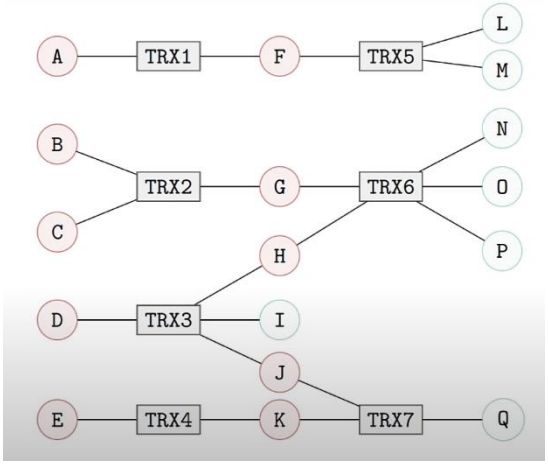
1. Transaction Overview
   - Structure: input(referenced earlier transactions, referencing outputs of earlier transactions, signature-proof that you are the current owner) and output(bitcoin amount, unlocking condition-private key)

     UTXO model



   - (this is how bitcoin works)

- Forward types: 1. Forwarding (input 1 to output 1), aggregating (several inputs, aggregates them as a single output), 3. Splitting (one output, splits them in several outputs), 4. Combined (m to n) (several inputs and several outputs)
- Transaction hierarchy:



- TRX1 – forwarding transaction
- TRX2 – aggregating transaction
- TRX3 – splitting transaction
- TRX4 - forwarding transaction
- TRX5 – splitting transaction
- TRX6 – combined transaction
- TRX7 – aggregating transaction

- If the input value does not correspond to the output value, there are the following possibilities: 1. Output value > input value -> transaction is invalid, 2. Ouput value < input value -> miner gets difference.

- A higher fee results in faster confirmation/ block inclusion.

2. Bitcoin Script and Transaction Types

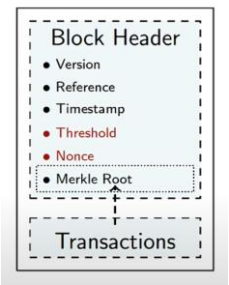
- Unlocking conditions are written and verified in Script (scripting language with predetermined list of commands). Is based on LIFO. Only if all commands run without errors and the result of the stack is 1, the transaction is considered valid.
- Pay-to-public-key links output directly to the public key, scriptSig includes corresponding signature(<sig>), public key(<pubKey>) is added as part of the unlocking condition.
  1. Put the <sig> in the stack
  2. Put the <pubKey> in the stack
  3. Operator “OP_CHECKSIG” takes the elements from the stack and compares whether the signature is valid given a specific private key and if this is okay, it results 1(true).
- Bitcoin address: private key -> public key -hash it> public key hash (can’t go back to public key) - > bitcoin address
- Pay-to-address/pay-to-public-key-hash: output is linked to bitcoin address instead of public key. The person who is providing the solution also must provide the public key with it.
- Pay-to-script-hash: in the scriptPubKey, only the hash value of a script is recorded. In order to reference the transaction output, a scriptSig, whose hash value corresponds to the hash value set in the scriptPubKey, must be provided

- Null Data: used to immortalize arbitrary data up to 40 bytes in the blockchain

3. SigHash Types
   - Network participants exchange signed transaction messages.
   - SigHash types: 1. All outputs (SIGHASH_ALL, default, signing all inputs and outputs), 2. Single outputs (SIGHASH_SINGLE, signing all inputs but only one output),3. no outputs (SIGHASH_NONE, no outputs are signed)
   - SIGHASH_ALL: modified inputs or outpus invalidate the signature
   - SIGHAS_SINGLE: Modified inputs or the modification of the signed output invalidate the signature, other people can change any of the unsigned outputs.
   - SIGHASH_NONE: Anyone can modify the unlocking conditions.
   - SIGHASH_ALL|SIGHAS_ANYONECANPAY: signs only the signer's inputs and all outputs, anyone can add or remove other inputs.
   - SIGHASH_SINGLE|SIGHAS_ANYONECANPAY: signs only the signer's input and one output. Other people are free to add or remove additional inputs and outputs. Any change to the signer's part of the transaction will invalidate the signature.
   - SIGHASH_NONE|SIGHAS_ANYONECANPAY: signs only the signer's inputs and no outputs, anyone can add or remove other inputs and outputs, anyone can use the signed input in any way.

V.

1. Block Assembly and Chain Structure
   - Nodes send their transactions to their peers.
   - Each note verifies the received transaction (inputs, outputs, scriptSig)
   - If the transaction is valid, it is either forwarded to respective peers or stored in own transaction mempool.
   - Merkle Root: represents all transactions of the block in the form of a compact 256-bit entry and guarantees that the transaction of a block cannot be modified unnoticed.
     Block header: version, reference (hash value of predecessor block header), timestamp (median timestamp of the previous eleven blocks)
   - Threshold: max hash value a block header may have to be considered valid. Nonce (number we use once): primary source of variation in block creation.



- Chain structure: block identification: block height (Static), block depth(dynamic), block header hash value/ block ID (static)

2. Proof of Work
   - You have to prove that you have expanded a certain amount of resources for having a chance of your block being accepted. Only a small fraction of tries will lead to the desired result.
   - In the bitcoin protocol, Proof of Work is implemented using block header hash values. Trial and error: assemble new candidate blocks and compute block header hash value until it's sufficiently low
   - Mining = the iterative process of creating candidate blocks and checking their block header hash values against the threshold
   - The threshold parameter 'delta' is dynamically adjusted so that the network will produce a valid block on avg every ten minutes. It is calculated based on the expected value and the actual duration.
   - Proof of Work in Bitcoin: the chain must be protected from replication; blocks need to create neither too fast or too slow.

3. Fork Theory
   - Chain with the greatest accumulated difficulty in its block sequence is seen as the most recent version.
     What does "greatest accumulated difficulty" mean?
   - Fork = disagreement on the current state of the ledger that leads to two or more competing versions of the blockchain.
   - Forks may arise for two distinct reasons: 1. The same rules(A=B): process-based forks. 2. Different rules (A != B): protocol based forks.
   - Process-based: 1. unintentional: probabilistic block race (when two blocks are created approximately at the same time). 2. Deliberate: block withholding and forced block race (when the user wants to catch up with a longer version for ex.). All these are temporary.

     Protocol-based: 1. Unintentional: Client Incompatibility (mistake when implementing the consensus rules) 2. Deliberate: rule change (a different consensus protocol)

     Types of Protocol-based forks: soft fork, hard fork, forced fork.

   - Why care about forks? Bcs of: 1. Uncertainty (confirmation status of transactions) 2. Confusion (various competing versions of the asset) 3. Security tokens (competing promises for delivery of one good) 4. Cost driver (tax / legal questions, maintaining compatibility)

4. Incentives and Potential Attacks
   - Blockchain network offers revenues to get participants as compliant CRN and bear the corresponding cost.

- Mining market: competitive due to low entry barriers, profits only through above avg efficiency.
- Underallocation: miners add power to realize more profits. Overallocation: miners remove power to avoid losses.
- Mining pool: successful mining of a block follows a Poisson(?) distribution, short to mid term payouts can deviate significantly, small miners are disproportionally affected.
- CRN operators: 1. rational agents 2. Independent
- Process based forks:

  Probabilistic block race: expected payout drives fast resolution along winning chain

  Block withholding / selfish mining: risk of losing block reward

  Forced block race: longest chain incentives do not discriminate attack chains

  Bitcoin incentives effectively protect consensus in absence of mining power concentration.

- Double spend attack: miner can choose which transaction to include in a block and o influence the consensus chain and deliberately delay a transaction (blocking) or attack a block with conflicting transactions (double spend)
- Double spend scenarios can also be created without any computational resources (not necessary to be a miner)
- Measures to minimize success of attacks: 1. Minimal waiting time between relaying transaction and handing out goods. 2. Maintaining a broad network connection

5. Alternative Consensus Protocols
   - Blockchain is a chain of transactions and states, and its rule set is attested by a reliable network of record keeping nodes (other participants)
   - Things you need to reach consensus: 1. explicit and unambiguous rule set (it is important that everyone understands what this rule are). 2. Decision mechanism (how the consensus in the system can be changed, when there are legitimate updates). 3. Incentive system (rewards compliant behavior and penalizes manipulation attacks).
   - Trilemma (goals to have a good consensus): Security, Scalability, Decentralization.
   - Popular Consensus Mechanisms: Proof of Work, Proof of Stake, Proof of Authority
   - Proof of Stake: does not employ any computation resources
   - Proof of Authority resembles a database
   - Proof of Work Trilemma:
     - Scalability: every full node needs to process every transaction. Block creation is very resource intensive.
     - Decentralization: open network with many participants. Mining pools compromise decentralization.
     - Security: Secured by resource allocation. Simplicity increases security.
   - Proof of Stake: staking = to participate in the consensus network, each node needs to deposit and lock native protocol assets.

Block creators are selected at random in proportion to their stake. Another validators (nodes) will attest the validity of the created blocks. Malicious behavior is punished by slashing the staked assets. Validators receive returns on their stake by performing their duties.

Proof of Stake Trilemma:

- Scalability: a subset of validators needs to process each transaction.
- Decentralization: open network with many participants. Potential crowding.
- Security: complex design may introduce new attack vectors.

- Proof of Authority: consensus network consists of a small set of approved nodes, which are called validators.
  Validators create and validate blocks. Other validators will attest the validity of the created blocks. Malicious or unresponsive behavior is punished by exclusion, potential legal actions.
  Proof of Authority Trilemma:
  - Scalability: very small set of validators and simple mechanism.
  - Decentralization: Closed network with risk of collision.
  - Security: avg database.

VI. Bitcoin as Money

1. The History of Digital Money
   - 1982 Digicash: virtual monetary unit that imitates the anonymity of cash
   - 1988 Crypto Manifesto: problems that may arise and cryptography as a potential solution to these problems.
   - 1997 Hashcash: proposed as a mechanism for anti-DoS and spam email.
   - 1998: B-money: existence of an untraceable network, senders and receivers are id-ed only by digital pseudonyms.
   - 2005 RPoW (Reusable Proofs of Work): combines ideas of HashCash and B-money. A client can create a token by providing a proof of work string and signing with his private key. It can be given to another key by signing a transfer order to a public key.
   - 2005 Bit Gold: combination of the proof of work algorithm. Computin power is used to collateralize a public ledger, which grows into a chain of blocks through a reflexive reference.
   - 2008 Bitcoin

2. Pricing Models and Volatility
   - Bitcoins value is solely determined by its liquidity premium.
   - Discounted cash flow idea: you have an asset, and this asset pays dividends or interest -> periodic return. (Basically: calculation these return with a discount, for example: take an interest rate, you discount the future returns of that interest rate)
   - Stock to Flow model: is used for various commodities. The idea: it says you have some stock of the asset and some flow for a given period. With the stock to flow model, you are looking at the relation between the stock and flow. After some calculations you can look at the result as some sort of approximation for price.

- Supply of bitcoin is limited (could have a decreasing supply in the future)

3. CBCDs (Central Bank Digital Currency) and Stablecoins
   - CBDC: established model of central banks as issuer of physical cash and lender of last resort.
     Infrastructure: drop in physical cash use and growing importance of digital payment systems. New payment solutions from private sector outside the banking industry.
     In Terms of transaction processing all CBDC design dimensions are centralized.
     Retail: public has access to it
     Wholesale: pure settlement coin where only a few institutions have access to it
     Token-based model idea: you're tracking the specific token
     Account-based model: regular bank accounts
   - Stablecoins: is a privately issued cryptoasset that is pegged to another asset (tries to maintain a stable exchange rate towards another asset). Goal: to decrease price volatility and create a blockchain based medium of exchange.
     Categories:
     - Off-main collateralized: examples for the US dollar: USTD, TUSD, USDC
       Risks: counterparty risk, regulatory risk
     - On-chain collateralized: examples: DAI, sUSD
       Risks: liquidation risk, oracle risk
     - Algorithmic stablecoin: Risks: flawed economics