IB CS Case Study Notes: Blockchain

-

May 2021

Contents

1 Resources

Links:

- Watch this visual demo first: https://www.youtube.com/watch?v=_160oMzblY8
- 3b1b video that goes into some more depth: https://www.youtube.com/watch?v=bBC-nXj3Ng4
- Another in-depth video: https://www.youtube.com/watch?v=Lx9zgZCMqXE
- https://www.khanacademy.org/computing/computer-science/cryptography/modern-crypt/v/the-fundamental-theorem-of-arithmetic-1
- https://www.khanacademy.org/economics-finance-domain/core-finance/money-and-banking/bitcoin/v/bitcoin-what-is-it

2 Terminology

2.1 51% Attack

51% attack refers to an attack on a blockchain - usually bitcoin's, for which such an attack is still hypothetical - by a group of miners controlling more than 50% of the network's mining hashrate, or computing power. The attackers would be able to prevent new transations from gaining confirmations, allowing the to halt payments between some or all users. They would also be able to reverse transactions that were completed while they were in control of the network, meaning they could double-spend coins. They would almost certainly not be able to create a new coins or alter old blocks, so a 51% attack would probably not destroy bitcoin or another blockchain-based currency outright, even if it proved highly damaging.

2.2 Block

Blocks are files where data pertaining to the bitcoin network is permanently recorded. A block records some or all of the most recent bitcoin transactions that have not yet entered any prior blocks. Thus a block is like a page of a ledge or record book. Each time a block is 'completed', it gives way to the next block in the blockchain. A block is thus a permanent store of records which, once written, cannot be altered or removed.

2.3 Blockchain

A blockchain 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).

2.4 Candidate block

In a few words, a candidate block is a block that a mining node (miner) is trying to mine in order to receive the block reward. So a candidate block may described as a temporary block that will be either validated or discarded by the network. Miners compete with each other to validate the next block and add to the blockchain, but first, they have to create a candidate block to participate in the mining competition.

Candidate blocks are created by miners by collecting and organizing multiple unconfirmed transactions from the memory pool. The transactions are then hashed to form a Merkle tree structure, which will eventually produce a Merkle root (or root hash). The merkle root is a single hash that represents all previous hashes of that tree, and therefore, all transactions that were included in that particular block.

2.5 Collision resistance

The collision resistance property requires that two different input messages should not hash to the same output. In other words, $h(x) \neq h(y)$. This property is also known as strong collision resistance.

2.6 Cryptocurrency

Cryptocurrency is a type of digital currency that uses cryptography for security and anti-counterfeiting measures. Public and private keys are often used to transfer cryptocurrency between individuals.

As a counter-culture movement that is often connected to cyberpunks, cryptocurrency is essentially a fiat currency. This means users must reach a consensus about cryptocurrency's value and use it as an exchange medium. However, because it is not tied to a particular country, its value is not controlled by a central bank. With bitcoin, the leading functioning example of cryptocurrency, value is determined by market supply and demand, meaning that it behaves much like precious metals, like silver and gold.

2.7 Cryptographic hash

A cryptographic hash function is a mathematical function used in cryptography. Typical hash functions take inputs of variable length to return outputs of fixed length. A cryptographic hash function combines the message-passing capabilities of hash functions with security properties.

2.8 Determinism

A hash procedure must be deterministic – meaning that for a given input value it must always generate the same hash value. In other words, it must be a function of the data to be hashed, in the mathematical sense of the term. This requirement excludes hash functions that depend on external variable parameters, such as pseudo-random number generators or the time of day. It also excludes functions that depend on the memory address of the object being hashed in cases that the address may change during execution (as may happen on systems that certain methods of garbage collection), although sometimes rehashing of the item is possible.

The determinism is in the context of the reuse of the function. For example, Python adds the feature that hash functions make use of a randomized seed that is generated once when the Python process starts in addition to the input to be hashed. The python hash is still a valid hash function when used within a single run. But if the values are persisted (for example, written to disk) they can no longer be treated as valid hash values, since in the next run the random value might differ.

2.9 Digital signature

A digital signature is a mathematical technique used to validate the authenticity and integrity of a message, software or digital document. The digital equivalent of a handwritten signature or stamped seal, a digital signature offers far more inherent security, and it is intended to solve the problem of tampering and impersonation in digital communications. Digital signatures can provide the added assurances of evidence of origin, identity and status of an electronic document, transaction or message and can acknowledge informed consent by the signer.

2.10 Distributed consensus

Distributed consensus refers to the elaborate, largely mathematically-based game that the members of the bitcoin network use to keep in sync their tens of thousands of individual duplicate copies of the entire set of transactions that ever happened in the blockchain.

Some basic facts you need, in order to understand the general idea:

- Nobody, or rather every member, is in charge of the bitcoin network.
- There are about 24,000 full nodes in the bitcoin network.
- Each 'full node' keeps a complete copy of the entire database of transactions that have ever happened on the bitcoin network. That's called 'the blockchain'.
- The data in the blockchain database is chunked up into groups of transcations. Each group is called a block. The data in each block includes a mathematical dependency on the data in the previous block, which links them together. That's the chain part.
- You can add a transaction to the bitcoin network by just asking any full node to add it. That node sends the transaction out to the rest of the network.

2.11 Double-spend problem

Double-spending is a problem in which the same digital currency can be spent more than once. In other words, double-spending is an instance in which a transaction uses the same input as another transaction that has already been broadcast on the network. This is a flaw that is unique to digital currencies because digital information is something that can be reproduced rather easily. Digital currencies such as bitcoin, can be thought of as being a digital file. If, for example, Bob has a file that has been saved locally to his computer, there is nothing preventing Bob from simply copying the file as many times as he wants, and sharing the file with multiple individuals. This same principle can be applied to digital currencies. It is not ideal for the same digital currency to be spendable more than once, because it can result in inflation and a loss of trust in that currency, making it effectively worthless.

2.12 Entropy

In cryptography, entropy is a measure of true randomness. An n-bit number chosen uniformly at random with a perfect random number generator has n bits of entropy, and entropy of other things can be computed in comparison to this case. For example, 4 words chosen uniformly at random from a word list of 1024 words has 40 bits of entropy because you can represent each word by 10 bits ($2^10 = 1034$) and stick the 4 groups of 10 bits together to get a 40-bit number chosen uniformly at random. When dealing with things chosen uniformly at random you can also compute the entropy by calculating the base-2 logarithm of the total possible outcomes, e.g. there are 6^{20} possible outcomes when rolling a 6-sided die 20 times and then writing down the results one after another (i.e. not summing or reordering them), so the result has $log_2 6^{20} = 51.7$ bits of entropy. If the result is in any way biased (like the sum of dice rolls, which is very much not uniformly distributed), then you can still calculate the entropy, but it's more difficult.

2.13 Genesis block

A genesis block is the first block of a block chain. Modern versions of bitcoin number it as block 0, though very early versions counted it as block 1. The genesis block is almost always hardcoded into the software of the applications that utilize its block chain. It is a special case in that it does not reference a previous block, and for bitcoin and almost all of its derivatives, it produces an unspendable subsidy.

2.14 Immutable transactions

Immutability – the ability for a blockchain ledger to remain a permanent, indelible, and unalterable history of transactions – is a definitive feature that blockchain evangelists highlight as a key benefit. Immutability has the potential to transform the auditing process into a quick, efficient, cost-effective procedure, and bring more trust and integrity to the data businesses use and share every day.

2.15 Key pair generation

Key generation is the process of generating keys for cryptography. They key is used to encrypt and decrypt data whatever the data is being encrypted or decrypted.

Modern cryptographic systems include symmetric-key algorithms (such as DES and AES) and public-key algorithms (such as RSA). Symmetric-key algorithms use a single shared key; keeping data secret requires keeping this key secret. Public-key algorithms use a public key and a private key. The public key is made available to anyone (often by means of a digital certificate). A sender will encrypt data with the public key; only the holder of the private key can decrypt this data.

2.16 Ledger

Ledge generally refers to the bull of quantities made in accounts.

Same in cryptoworld, it makes sense with the record of transactions being done among bitcoin users.

Also it is a secured database which stores and holds the money of people in the form of bitcoins.

2.17 Merkle proof

Merkle proofs are used to decide upon the following factors:

- If the data belongs in the merkle tree.
- To concisely prove the validity of data being part of a dataset without storing the whole data set.
- To ensure the validity of a certain data set being inclusive in a larger data set without revealing either the complete data set or its subset.

2.18 Merkle tree

A merkle tree is a hash-based data structure that is a generalization of the hash list. It is a tree structure in which each leaf node is a hash of a block of data, and each non-leaf node is a hash of its children. Typically, merkle trees have a branching factor of 2, meaning that each node has up to 2 children.

Merkle trees are used in distributed systems for efficient data verification. They are efficient because they use hashes instead of full files. Hashes are ways of encoding files that are much smaller than the actual file itself. Currently, their main uses are in peer-to-peer networks such as Tor, bitcoin, and git.

2.19 Miner

Miners can be defined as accountants who record every transaction to the blockchain. The concept is simple, a proof of payment is important if you want your payment to be valid. The miners are the ones who keep the records of your payment. Hence they are record keepers who keep the system updated of new payments and existing ones.

2.20 Mining

Bitcoing mining is the process of creating, or rather discovering, bitcoin currency. Unlike real-world money that is printed when more is needed, bitcoin cannot simple be willed into existence, but has to be mined through mathematical processes. Bitcoin maintains a public ledger that contains past transactions, and mining is the process of adding new transactions to this ledger.

2.21 Nonce

A nonce (number only used once) is a number added to a hashed block that, when rehashed, meets the difficulty level restrictions. The nonce is the number that blockchain imners are solving for.

2.22 Non-invertibility

Non-invertibility is another feature that's often desirable, depending on the intended usage of the algorithm. This says that it should be impossible, or at least prohibitively difficult, to work out the input that led to any given hash. Ideally, it should be easy to transform data into a hash, and practically impossible to go the other way.

2.23 Non-repudiation

Non-repudiation is a method of guaranteeing message transmission between parties via digital signatures and/or encryption. It is one of the five pillars of information assurance (IA). The other four are availability, integrity, confidentiality and authentication.

Non-repudiation is often used for digital contracts, signatures, and email messages.

By using a data hash, proof of authentic identifying data and data origination can be obtained. Along with digital signatures, public keys can be a problem when it comes to non-repudiation if the message recipient has exposed, either knowingly or unknowingly, their encrypted or secret key.

2.24 One-way function

A hash is designed to act as a one-way function – you can put data into a hashing algorithm and get a unique string, but if you come upon a new hash, you cannot decipher the input data it represents. A unique piece of data will always produce the same hash.

2.25 Proof of work

A proof of work is a piece of data which is difficult (costly, time-consuming) to produce but easy for other to verify and which satisfies certain requirements. Producing a proof of work can be a random process with low probability so that a lot of trial and error is required on average before a valid proof of work is generated. Bitcoin uses the hashcash proof of work system.

2.26 PuTTYgen

PuTTYgen is a key generator. It generates pairs of public and private keys to be used with WinSCP. PuTTYgen generates RSA, DSA, ECDSA, and Ed25519 keys.

2.27 Self-referential data structure

A self-referential class contains a reference member that refers to a class object of the same class type.

Self-referential objects can be linked together to form useful data structures such as lists, queues, stacks and trees.

2.28 SHA256

The Secure Hash Algorithm (SHA) is one of a number of cryptographic hash functions. A cryptographic hash is like a signature for a text or a data file. SHA-

256 algorithm generates an almost-unique, fixed size 256-bit (32-byte) hash. Hash is a one way function - it cannot be decrypted back. This makes it suitable for password validation, challenge hash authentication, anti-tamper, digital signatures, etc.

SHA-256 is one of the successor hash functions to SHA-1, and is one of the strongest hash functions available.

2.29 Takeover attack

Account takeover is a form of identity theft where a fraudster illegally gets access to a victim's bank or online e-commerce account using bots. A successful account takeover attack leads to fraudulent transactions and unauthorized shopping from the victim's compromised account.

2.30 Transaction pool

The transaction pool (or mempool, as it's usually called) is not a network-wide pool. Each node maintains its own mempool. When a node receives your transaction, it will validate and add it to its own mempool, and possible broadcast further on.

3 Other terms

3.1 Asymmetric key cryptography

Asymmetric key cryptography is used when signing a transaction and verifying the identity of the sender. Signing uses a private key which is kept secret, in conjunction with a message (i.e. transaction / ledger entry) to produce a unique signature for that message. Anyone can then use the sender's publicly available verification key (public key) with a verification function. They need to input a public key, the message and the signature, and it outputs whether it was really the sender who signed it with absolute certainty.

3.2 Avalanche effect

A property of good hash functions. When a small change to a message should change the hash value so extensively that the new hash value appears entirely different from the old one.

3.3 Salt

A secret random number that is appended (added to the end) of an input to a hash function. It serves as an additional safeguard against someone breaking it (finding the original input data through trial and error). A salt is used to make common passwords (e.g. "password") generate different keys, but it doesn't have to be unique, but is kept secret, unlike a nonce.

3.4 Generation transaction/coinbase (reward)

Always the first transaction in a block, and contains a predetermined reward for the miner who found the proof of work for the block.

3.5 Rainbow table

It is a precomputed table of inputs for a hash function. It is usually used to crack passwords that have been hashed without using a salt by searching for a hash in it (which is quicker than computing hashes using brute force).

4 Possible Questions

Q. Explain how a cryptocurrency transaction is validated and how the blockchain is involved.

Q. Explain how a program would traverse a [given] self-referential data structure and display the data in each node (using pseudo-code or a flowchart).

Q. Explain how you would add a new node between two [given] nodes in a [given] self-referential data structure (using pseudo-code or a flowchart).

Q. Explain how a self-referential data structure could be used to implement a blockchain.

Q. Explain what a hashing algorithm is.

Q. The essential characteristics of good hashing algorithms are determinism, noninvertibility and collision resistance. Explain the meaning of these terms.

Q. Explain what "asymmetric key cryptography" means in the context of a blockchain.

Q. Digital signatures are used to validate MONS transactions before they are added to the transaction pool. In this context, explain:

- key generation
- creation of a signature
- verification of a signature

Q. Key generation software, such as PuTTYgen, often uses a physical source of entropy to generate key pairs. Explain the use of entropy in this context.

5 Sources

The terminology definitions were taken from: https://quizlet.com/409294126/ib-computer-science-case-study-2020-a-local-economy-driven-by-blockchain-flash-cards

Other terms and possible questions were taken from documents made by strox#4591