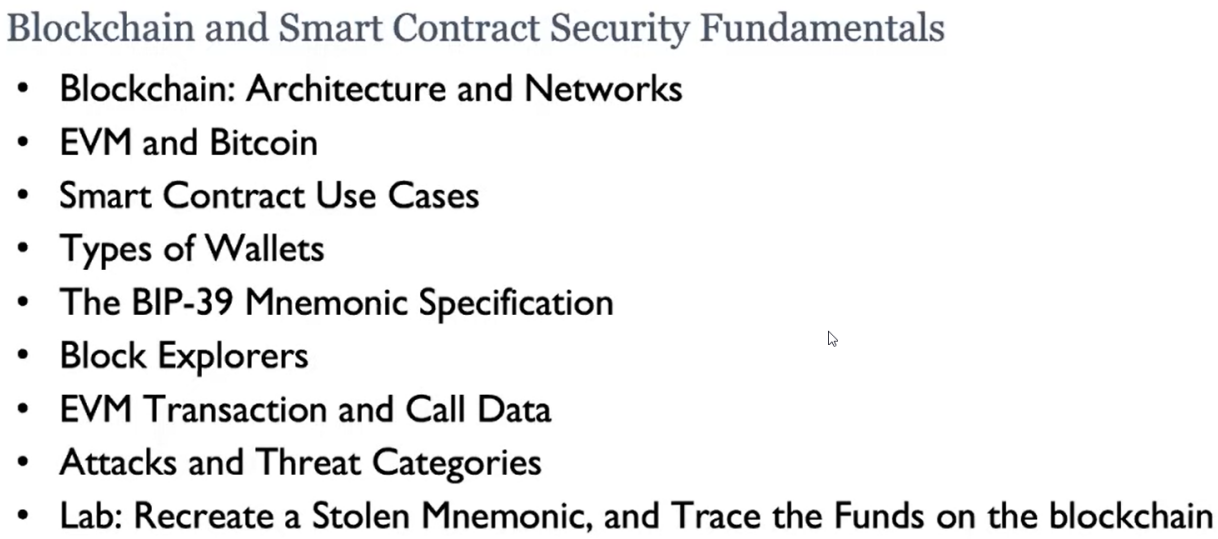
# Basics

So here we start with the basics of blockchain and smart contract



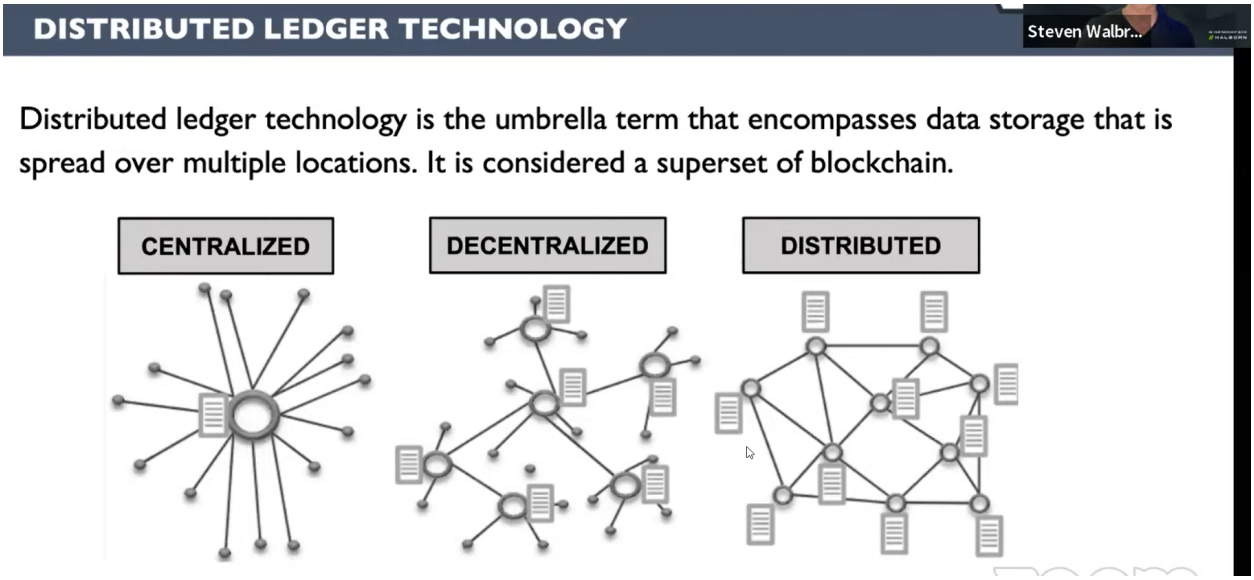

Lets start one by one here

**1) So what is blockchain**

"An open and distributed ledger, which can record transactions between two parties in a permanent way."

- Open: Anyone can access the entire blockchain.
- Distributed: The blockchain does not live in single or multiple data centers. Instead, each miner owns a copy of the entire blockchain.
- Ledger: A sequential log of each transaction.
- Transactions: Destination receives something of value, such as a currency.
- Permanent: To delete a transaction, all subsequent transactions must be removed. Also known as "immutable."

## 2) What is Distributed Ledger Technology




## 3) Blockchain Arch

## BLOCKCHAIN ARCHITECTURE

Steven Walbr...

Blockchain uses a peer-to-peer network. It has several advantages:

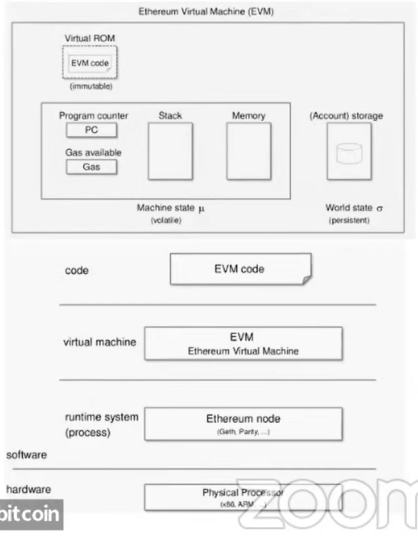
- **Decentralization**: Balance between decentralization and scalability.
- **Resiliency to node outages**: Nodes participate in the network in exchange for the rewards of block creation.
- **Resiliency to network outages**: Distribute transactions and blocks to nodes for block creation and ledger updates.
- **Security:** An attacker can impact the blockchain's operations by filtering a node's view of the state of the blockchain network.

**4)What is EVM?**

## EVM (ETHEREUM VIRTUAL MACHINE)

Steven Walbr...

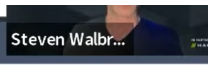
- The EVM is a sandboxed virtual stack embedded within each full Ethereum node, responsible for executing contract bytecode. Contracts are compiled to EVM bytecode.
- Virtual machines create a level of abstraction between the executing code and the executing machine.
- This layer is needed to improve the portability of software, as well as to make sure applications are separated from each other and separated from their host.




Bitcoin is c. Yes, So attacking contracts bitcoin

**5) What is Ethereum Networks?**

## ETHEREUM NETWORKS




**Mainnet**

- The live public Ethereum production blockchain, where actual valued transactions occur on the distributed ledger

**Public testnet(s)**

- Public Ethereum blockchain(s) designed for testing, running on valueless Ether available from "faucets," that mirror the mainnet environment as closely as possible (Examples: Ropsten, Kovan, Rinkeby, Görli)

**Local testnet(s)**

- Local, running on your machine or on a small scale; private Ethereum blockchains (Examples: Ganache, eth-tester, private client network clusters)

ether is a commodity we use to buy gas.

6) use of smart contract

## BLOCKCHAIN AND SMART CONTRACTS USES

| |
|---|
| Decentralized Finance (DeFi) |
| Supply Chain |
| Gambling and Gaming |
| Proof of Ownership and NFTs |
| Metaverse |
| Private Transactions |
| Darknet Markets |
| Consumer Services |
| Authentication and Zero Knowledge Verification |
| Government and Governance |

A) DEFI

## DECENTRALIZED FINANCE (DEFI)

Steven Walbr...

CeFi (centralized finance)

- Driven by large banks and other financial institutions
- No transparency into the flow of money
- Venmo and CashApp give the veneer of decentralization but they, too, are centralized

**vs.**

DeFi

- Programmability: business logic via smart contract
- Immutability: tamper-proof data (AND CODE) available for audits
- Interoperability: built with abstraction layers in mind
- YOU are the bank

## DECENTRALIZED FINANCE (DEFI) - USE CASE (1)

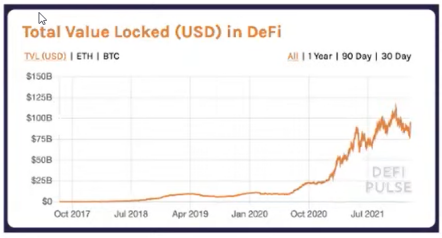
Steven Walbr...

Currently the most popular use for smart contracts:

- Creating monetary banking services, such as issuance of stable coins, providing peer-to-peer or pooled lending and borrowing platforms
- Enabling advanced financial instruments such as DEX, tokenization platforms, derivatives, and predictions markets
- Crowdsourcing of funds via (DAOs)

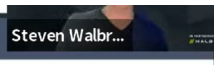
Total Value Locked (USD)

**$96B** as of 01/15 2022




The largest target for security research and exploitation due to immense financial incentive.

## DECENTRALIZED FINANCE (DEFI) - USE CASE (2)




Decentralized finance is the use case to shift traditional financial products to the open- source and decentralized world. Using blockchain and smart contracts, the intention is to remove the need for intermediaries, reduce overall costs for users, provide full transparency, and improve security.

| Use Case | Example | Purpose |
|---|---|---|
| Stable coins | USDC/Tether | Provide cryptocurrencies within the blockchain ecosystem pegged to a fiat currency (i.e., DAI) |
| Lending and borrowing | AAVE, Compound | Provide the ability to lend/borrow cryptocurrencies with interest for use in leveraged trading, investing, or capital growth |
| Exchanges | Uniswap | A decentralized exchange platform that enables users to trade currencies on a market |
| Insurance | Nexus | Insurance against risks like price risk, counterparty default, or technology/security risk |

NFT

## PROOF OF OWNERSHIP AND NFTS - USE CASE

Non-fungible tokens (NFTS) can be used to track and verify ownership (Provenance) and prove authenticity. Ownership of creatively produced materials, land or intellectual property:

- **Artwork**: Authenticity, provenance, information watermarked (i.e., Bored Ape Yacht Club)
- **Music**: Track performance royalties and payment processing of recording usage (i.e., Imogen Heap)
- **Land:** Ownership and claim.

METAVERSE

## THE METAVERSE - USE CASE

Steven Walbr...

Decentralized social networks and worlds that aim to create a new and trusted social experience, free from bots and fake accounts, and filled with digital representations of real-world items and real estate via NFTs.


| Name | Purpose | Link |
|---|---|---|
| The Sandbox | A game where users can build a virtual world to work, live in, and explore. | https://thesand-box.org/ |
| Decentraland | A virtual world of blockchain assets to collect. | https://docs.decentraland.org |
| META | The company formerly known as Facebook. | https://about.facebook.com/meta/ |

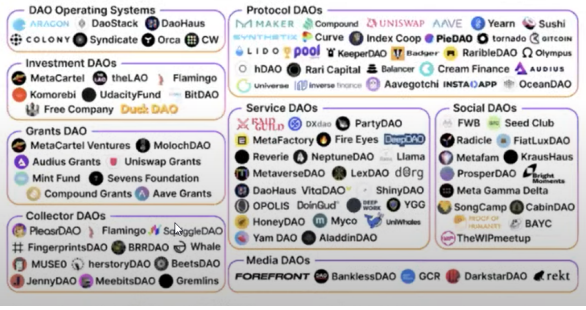
DAO

## DECENTRALIZED GOVERNANCE (DAO) - USE CASE

Steven Walbr...

A DAO (Decentralized Autonomous Organization) uses blockchain to create transparent governing bodies that make decisions by those that participate or own governance tokens. A DAO has no centralized body controlling its decisions. Instead, users who own voting tokens decide on everything, often weighted by the number of tokens owned. Every transaction is published on the public blockchain for everyone to witness.

DAO Lists



## CRYPTOGRAPHY FUNDAMENTALS

Steven Walbr...

Blockchain relies on the same crypto primitives that you are familiar with.

- What are the goals of cryptography? Confidentiality, integrity, and authenticity
  - *Communicate securely over insecure channel*
- Bob sends a message to Alice. Alice wants to check the following:
  - Confidentiality: the plaintext message can only be read by Alice.
  - Integrity: the received message is the same as what Bob sent.
  - Authenticity: the message came from Bob.

## PAPER - WALLET FORMAT

Steven Walbr...

**The least secure wallet type (only recommended for temporary use/small amounts)**

- A private key and address are generated, and the user prints them on a piece of paper.
  - Cannot tell users if they have actually received cryptocurrency unless they check the blockchain itself or scan the QR code to see unspent transactions
  - Exposes private key to user (and any others who see it)
  - Full migration to another wallet type is not possible
  - No seed
- Should be avoided, especially by beginners.
- This type used by Bitcoin ATMs.

## NON-DETERMINISTIC - WALLET TYPE

Steven Walbr...

Each key pair is generated independently.
This seems appealing from a security perspective but creates managerial issues and does not have the flexibility that is provided from deterministic wallets (in the next section).

Disadvantages:

- Backups required: You need to back up your wallet every time you receive a new payment.
- Scalability issues: Keys are fully independent, and do not derive child keys.

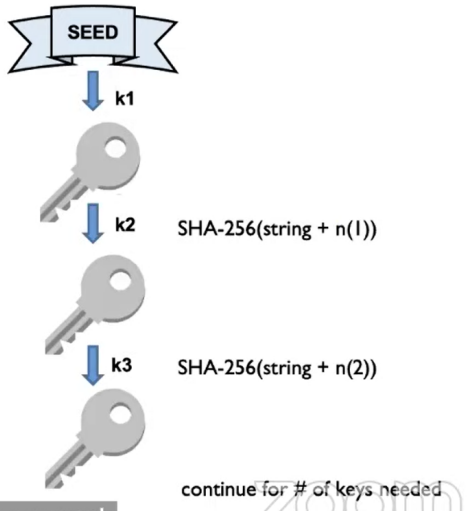
Paper wallets are usually non-deterministic.

## DETERMINISTIC - WALLET TYPE

Steven Walbr...

A single seed is used to generate all subsequent keys.

At a high level, a function is applied to the previous key to generate the next one. This makes it space efficient. Known as "Type 1" Deterministic Wallet.




Advantages:
You only need to remember the seed to restore the backup (such as a mnemonic).

Disadvantages:
Can create a Hierarchy of Parent/Child Keys.

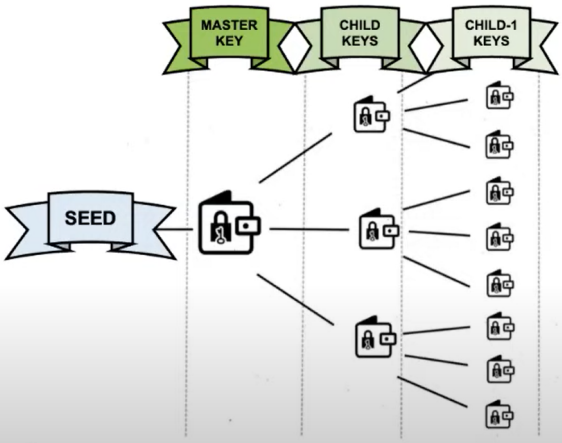
## HIERARCHICAL DETERMINISTIC - WALLET TYPE

Steven Walbr...

In deterministic wallets, key generation could be visualized as a linear chain.

HD wallets enable a tree-like structure. Child keys can generate their own keys.

This unlocks business use cases such as separation of finances between departments.




Most new hardware/software wallets are HD wallets.

## MNEMONIC KEYS

Steven Walbr...

We have been talking about a *seed* that forms the basis of key generation.

The most commonly found standard in blockchain and smart contract key generation is the BIP39 mnemonic standard.

The BIP39 standard uses a 12–24-word mnemonic key as the seed.

- Words selected from a bank of 2,048 words
- Corresponds roughly to 128–256 bits of entropy with a built-in checksum

The words are comfortably in our vocabularies and are not too similar to one another.

https://github.com/bitcoin/bips/blob/master/bip-0039/english.txt

## BIP-32 AND BIP-39 STANDARDS

Steven Walbr...

The easiest way that hackers usually get access to your wallet is through stealing the mnemonic seed.

BIP-39, used for a hierarchical deterministic wallet, is the most common mnemonic standard used, which is 12 or 24 words, generated at random, and written down by the owner of the wallet along with a passphrase (not required). This is the standard used for the BIP-32 wallet generation.

Oftentimes, people take a picture of the mnemonic or write it down on unsecured documentation.

Anyone that sees this can steal your funds.


The BIP-39 English word list can be found here (2048):
https://coldbit.com/wp-content/uploads/2019/05/bip-39-wordlist.pdf

Other word lists have been put together for different standards, such as the EFF Diceware list (7,776 words):
https://www.eff.org/files/2016/07/18/eff_large_wordlist.txt

## THE BIP-39 SPECIFICATION (1)

Steven Walbr...

Steps involved in private key creation

1. Source of entropy

The first step is to provide a reliable source of randomness. An example can be a coin flip (heads or tails).

BIP-32 requires entropy length to be between 128 and 256 bits in multiples of 32 bits.

We want to have a 24-word seed, so let's toss the coin 256 times and write heads as "0" and tails as "1." Each coin flip is 1 entropy.

```
10100100101100010011110001111000100111101111011101100100101
00110011001010000101011111010000101111111111110100000100100
00010010101101000101010011001110111001100010111011010010101
11010100111101001001101011111100011001010110010001101000100
00011011000110010111000 1
```

## THE BIP-39 SPECIFICATION (2)

Steven Walbr...

Steps involved in private key creation

1. Source of entropy

2. Split entropy result into groups

Split the entropy binary results into groups.

- 23 groups each 11-bit long
- 24th group has 3 leftover bits:

**10100100101 10001001111 00011110001 00111101111 01110110100 10100110011 00101000010 10111110100 00101111111 11110100000 10010000010 01010110100 01010100110 01110111001 10001011101 10100101011 01010011110 10010011010 11111100011 00101011001 00011010001 00001101100 01100101110 001**

## THE BIP-39 SPECIFICATION (3)

Steven Walbr...

Steps involved in private key creation

1. Source of entropy

2. Split entropy result into groups

3. Encode

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | ABANDON | 147 | BAR | 294 | CAVE | 441 | DAMP |
| 1 | ABILITY | 148 | BARELY | 295 | CEILING | 442 | DANCE |
| 2 | ABLE | 149 | BARGAIN | 296 | CELERY | 443 | DANGER |
| 3 | ABOUT | 150 | BARREL | 297 | CEMENT | 444 | DARING |
| 4 | ABOVE | 151 | BASE | 298 | CENSUS | 445 | DASH |
| 5 | ABSENT | 152 | BASIC | 299 | CENTURY | 446 | DAUGHTER |
| 6 | ABSORB | 153 | BASKET | 300 | CEREAL | 447 | DAWN |
| 7 | ABSTRACT | 154 | BATTLE | 301 | CERTAIN | 448 | DAY |
| 8 | ABSURD | 155 | BEACH | 302 | CHAIR | 449 | DEAL |
| 9 | ABUSE | 156 | BEAN | 303 | CHALK | 450 | DEBATE |
| 10 | ACCESS | 157 | BEAUTY | 304 | CHAMPION | 451 | DEBRIS |
| 11 | ACCIDENT | 158 | BECAUSE | 305 | CHANGE | 452 | DECADE |

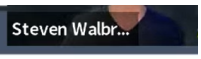
Each 11-bit group represents a number 0–2047 in decimal.

Each of these numbers (0–2047) is an index into the BIP-39 word list. - *https://github.com/bitcoin/bips/blob/master/bip-0039/english.txt*

The first binary number is 10100100101. This binary number converted to decimal is 1317. Do this for each of the 23 eleven-bit groups (all except the last one with three).

**1317, 1103, 241, 495, 948, 1331, 322, 1524, 383, 1952, 1154, 692, 678, 953, 1117, 1323, 670, 1178, 2019, 345, 209, 108, 814**

## THE BIP-39 SPECIFICATION (4)




Steps involved in private key creation

1. Source of entropy
2. Split entropy result into groups
3. Encode
4. Map to 0 indexed word list

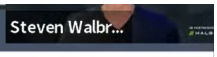
The first word in the index for 1317 is "pigeon."

Do this for each of the numbers derived.

**1317, 1103, 241, 495, 948, 1331, 322, 1524, 383, 1952, 1154, 692, 678, 953, 1117, 1323, 670, 178, 2019, 345, 209, 108, 814**

**pigeon measure bullet digital isolate please choose salon copper village motion final feel jaguar merry pistol fatigue nation wise clinic boss assault grape**

## THE BIP-39 SPECIFICATION (5)




Steps involved in private key creation

1. Source of entropy
2. Split entropy result into groups
3. Encode
4. Map to 0 indexed word list
5. Derive the checksum (24th word)

The purpose of a checksum is to quickly verify if the list of words is correct (valid) or not. Its purpose is to detect errors like using the wrong word, missing a word, or having it in the wrong order.

To calculate the checksum, make a SHA256 digest hash from the 256 entropy bits we started with in Step 1.

We can use "shasum –a 256 -0" on Zion or any Linux CLI.

Our result is:

**5182e8c04b11081decca3b25185744ec85710199de7ac7dba9ffc5554d310fe4**

```
root@ziion:/home/ziion# echo 10100100101100010011110001111000100111101111011011
01001010011001100101000010101111101000010111111111110100000100100000100101011010
00101010011001111011001100010111011010010101101010011110100100110101111110001100
10101100100011010001000011011000110010111 0001 | shasum -a 256 -0
5182e8c04b11081decca3b25185744ec85710199de7ac7dba9ffc5554d310fe4 ^-
```

## THE BIP-39 SPECIFICATION (6)

Steven Walbr...

Steps involved in private key creation

1. Source of entropy
2. Split entropy result into groups
3. Encode
4. Map to 0 indexed word list
5. Derive the checksum (24th word)

5182e8c04b11081decca3b25185744ec85710199de7ac7dba9ffc5554d310fe4

The SHA256 is a hexadecimal format, and for this purpose to derive a checksum, we only need the first 8 leftmost bits (1 byte) from this hash.
We can use any hex to binary converter or online tool.


We get from "51" the binary value "0101 0001"

Finally, we take those 8 binary characters, and add them to the last 3-digit binary from our original entropy in the last (24th) group.

Original 24th binary: **001** + New **01010001** = **00101010001 = 337**

337 BIP Index Word is **clay**

## THE BIP-39 SPECIFICATION (7)

Steven Walbr...

Steps involved in private key creation

1. Source of entropy
2. Split entropy result into groups
3. Encode
4. Map to 0 indexed word list
5. Derive the checksum (24th word)
6. Provide seed phase to Key Derivation Function

The 24-word mnemonic seed is now provided to the BIP-32 function used by deterministic wallets to generate the actual binary master key.

**pigeon measure bullet digital isolate please choose salon copper village motion final feel jaguar merry pistol fatigue nation wise clinic boss assault grape clay**

```
DK = PBKDF2(PRF, Password, Salt, c, dkLen)
```

## ASSUMPTIONS ON BITCOIN'S SECURITY (1)

Steven Walbr...

**Will quantum computing eventually break Bitcoin?**

| Computer | CPU Power |
|---|---|
| **Honeywell's 6 Qubit Quantum Computer** | Quantum volume of 64 qubits. |
| **IBM Q System One** | Quantum volume of 53 qubits. |
| **Google Quantum Computer** | Quantum volume of 54 qubits. |

Announced in 2020, Honeywell released a quantum computer that is two times more powerful than IBM and Google's supercomputers. However, at 64 qubits of power, this still does not even come close to the 3,000 qubits required to break the SHA256 that is used for private keys.

Scientists estimate that it would take **1.1 * 10^57 years** with the best computer today to find the private key that matches a given blockchain account.

## ASSUMPTIONS ON BITCOIN'S SECURITY (2)

Steven Walbr...

**Will quantum computing eventually break Bitcoin?**

NO!

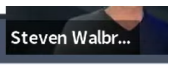
The argument that Bitcoin will be broken with the evolution and implementation of quantum computing is referred to as "quantum supremacy."

SHA-256 produces 256 bits which is 32 bytes.
Each byte has 256 possible values.
Each bit has 2 values (0 or 1), thus 2^256.
Number of Bitcoin Private Keys. **2^256 ~= 10^77**

How big is this?

There are an estimated **10^78** atoms in the known, observable universe. Thus, theoretically, nearly every atom in the entire universe could have its own private key.

## ASSUMPTIONS ON BITCOIN'S SECURITY (3)




**Will updates be needed eventually?** YES!

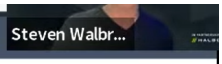
Certain algorithms have been developed specifically for quantum computers. Two of these directly impact blockchain security:

- **Shor's algorithm:** Breaks the security of classical public key cryptography. Post-quantum algorithms are still secure.
- **Grover's algorithm:** Reduces the complexity of brute forcing a hash function from $2^N$ to $2^{(N/2)}$

### Is this a big deal?

At the moment, quantum computers are far from large enough to effectively run Shor's or Grover's algorithms. When this changes, all that is needed is an update to the digital signature and hash algorithms in use.

## ATTACKING BIP-32




So, now that our brain is melted: how hard is this to brute force?

The table to the right outlines levels of attacking/cracking a PBKDF2-HMAC-SHA512-based key.

Each level is increasing in hash power.

Levels 3 and 4 are hypothetical levels, as these specialized ASICs do not exist, but may in the future.

**LEVEL 1 ATTACK** (~10,000 hashes/s):
A standard laptop can run about 500 PBKDF2-HMAC-SHA512 hashes/sec. multiple cores on a high end can get up to 10,000.

**LEVEL 2 ATTACK** (~1,000,000 hashes/s):
GeForce GTX 1080 can run approx. 240,000 hashes/s. You can combine several together with GPU chaining.

**LEVEL 3 ATTACK** (100,000,000 hashes/s): **HYPOTHETICAL**
Specialized ASIC made just for BIP-32 cracking. We assume it will be 10x faster than the GFX card in Level 2.

**LEVEL 4 ATTACK** (1,000,000,000 hashes/s): **HYPOTHETICAL**
A supercluster of ASICs (like a mining pool but specialized only for cracking PBKDF2-HMAC-SHA512)

## ATTACKING BIP-32 - ESTIMATED TIME TO CRACK

Steven Walbr...

Based on the four levels of attack power, these are the estimates to crack the key.

| WORDS | POSSIBILITIES | ENTROPY | LEVEL 1 ATTACK | LEVEL 2 ATTACK | LEVEL 3 ATTACK | LEVEL 4 ATTACK |
|---|---|---|---|---|---|---|
| 2 | 2048^2 | 22 Bits | 9 minutes | 3 seconds | Instant | Instant |
| 3 | 2048^3 | 33 Bits | ~1 week | 2 hours | 1 minute | 8 seconds |
| 4 | 2048^4 | 44 Bits | 55 years | 200 days | 2 days | 4 hours |
| 5 | 2048^5 | 55 Bits | 114 millennia | 1 millennium | 11 years | 1 year |
| 6 | 2048^6 | 66 Bits | Infinity | Infinity | Infinity | 2 millennia |
| 7 | 2048^7 | 77 Bits | Infinity | Infinity | Infinity | Infinity |
| … | … | … | … | … | … | … |

BLOK EXPLORERS

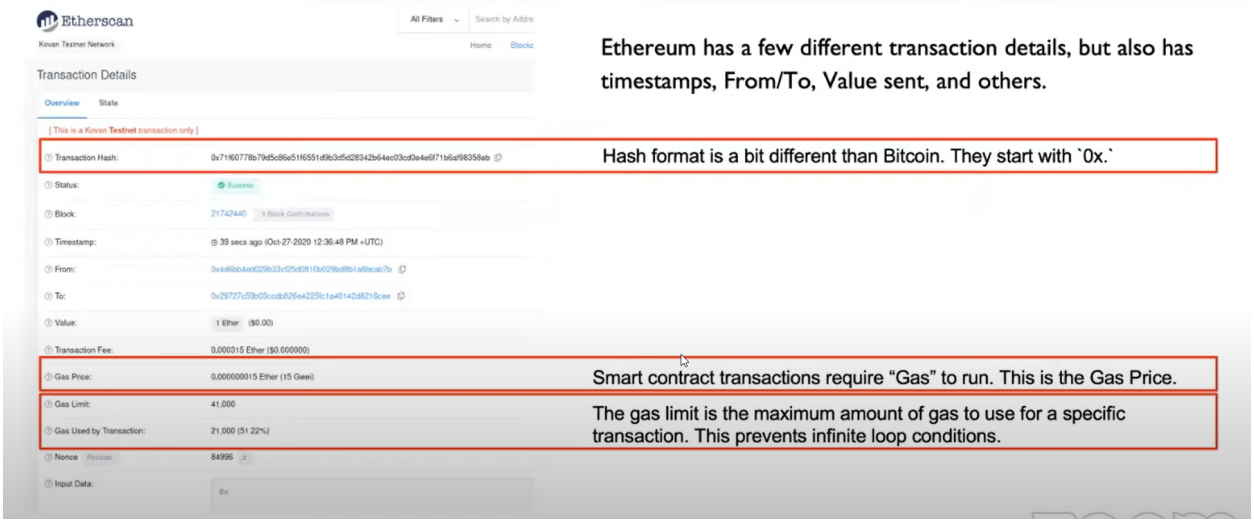
## BLOCK EXPLORERS - ONLINE TRANSACTION ANALYSIS

Steven Walbr...

- Application that allows everyone with an internet connection to track in real time all the transactions made
- Typically, present blocks in tabular form with the following attributes:
  - Block height, hash, time, reward, mined by, size
- Additional information provided regarding each block and the transactions that it contains
- More well-known block explorers (with SegWit support):
  - btc.com
  - blockchair.com
  - Etherscan.io
  - Solscan

## EVM TRANSACTIONS

Steven Walbr...




## EVM TRANSACTIONS - SMART CONTRACT INTERACTION

Steven Walbr...

When a smart contract has been compiled and deployed to a blockchain, it is given a unique 42-character hex consisting of numbers and uppercase and/or lowercase letters (e.g., 0x62a34C55...).
This is exactly the same as addresses and allows users to look up transactions on the ledger that this address was involved with, such as Etherscan.

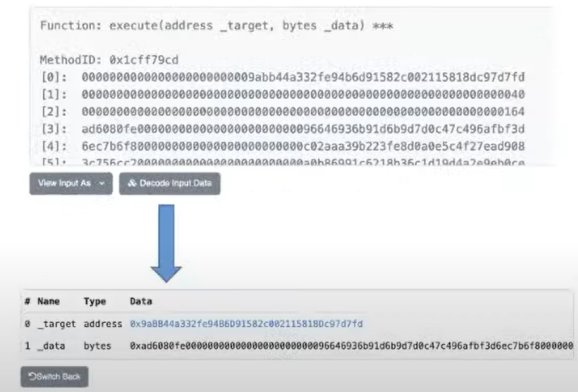
They also expose their ABI to provide methods for interacting. The interactions are similar to REST APIs, since they usually take encoded JSON, and perform similar to GET/POST type HTTP Methods.

These interactions are the same as sending/receiving Ether but can also be more complex depending on the purpose of the contract.

The most important part of the contract is its “**state**.” This enables a contract to hold Ether. The contract can be loaded with Ether at deployment or sent from one address to another or to an individual user. It all depends on the contract’s business logic.

## EVM TRANSACTION - CALLDATA

Steven Walbr...

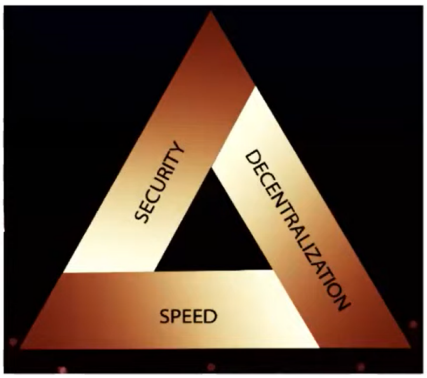



- Smart contract transactions require gas to run and change the state of the EVM chain.
- Sometimes, the transactions are not just sending ether, but are making a "call" to a Smart contract.
- This "CallData" is seen in the Input Data section of the Block Explorer. Call data is encoded/hashed, and usually contains messages to smart contract functions (called methods).
- It can also sometimes contain decodable text that are messages or instructions.

## BLOCKCHAIN TRILEMMA

Steven Walbr...

- Coined by Vitalik Buterin, founder of Ethereum
- Speed, security, decentralization: useful to think about how focusing on one affects the others
  - Decentralization – PoW can ensure decentralized power through consensus, but it doesn't scale well due to resources constraints.
  - Security – PoW can have issues with 51% attack if its not scaled. PoS can be influenced by collusion of majority stakeholders.
  - Speed – PoS or DPoS help scale and speed, but it compromises decentralization.



## TYPES OF VULNERABILITIES TO THE BLOCKCHAIN

Steven Walbr...

Risk categories and subjective liklihood of occurrence.

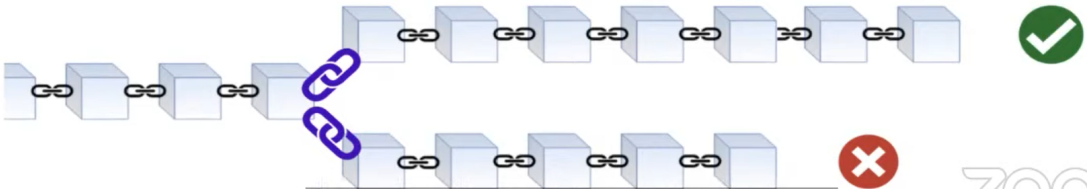
| CATEGORY OF RISK | LIKELIHOOD OF OCCURRENCE |
|---|---|
| Network Attacks | LOW |
| Node Security | MEDIUM |
| Smart Contract Exploits | HIGH |
| Centralization Issues | HIGH |
| User and Personal Security | VERY HIGH |

## NETWORK ATTACKS - 51% ATTACK (1)

Steven Walbr...

### The 51% Attack

- In a Proof of Work blockchain, the chain with the largest block height is considered the "main chain." A blockchain that has a lower level of consensus and hash rate will be more vulnerable to a 51% attack.
- An attacker that controls more than 50% of the network's computing power can mine a longer chain in secret and broadcast it which will overwrite blocks that were mining before it, excluding and/or modifying the ordering of transactions.



## NETWORK ATTACKS - 51% ATTACK (2)

Steven Walbr...

**Things an attacker *can* do in a 51% attack**

- Reverse transactions while in control.
- Double-spend coins.
- Reverse confirmations for any previous transactions while in control.
- Prevent transactions from gaining confirmations.
- Prevent other miners from mining any valid blocks.

**Things the attacker *can't* do in a 51% attack**

- Reverse other user transactions without their cooperation (unless their transaction history has been affected by a double-spend).
- Prevent transactions from being sent at all (they will still show as 0/unconfirmed).
- Change the number of coins generated per block.
- Create new coins.
- Send coins that never belonged the attacker.

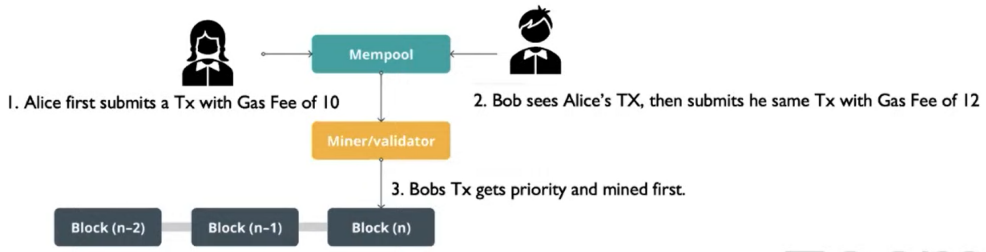
Launching a 51% attack on a large network like Bitcoin is not cheap.
There are electricity costs, mining costs, and hardware and CPU costs to run an attack.

## NETWORK ATTACKS - FRONT RUNNING

Steven Walbr...

Front Running example on the Blockchain:

- Decentralized exchange transactions can be seen in the transaction "Mempool".
- An attacker can have a process monitor the mempool for swap transactions.
- If the process finds a profitable transaction, it can copy the transaction details and submit quickly with it with a higher gas fee.
- The result is that the attacker's transaction is included in the next block ahead of the first transaction.



## NETWORK ATTACKS - SYBIL ATTACK (BITCOIN)

Steven Walbr...

The Bitcoin network never keeps count of how many nodes are participating on the network or where they are located. A Sybil attack focuses on isolating honest nodes with many malicious nodes under their control.

With only attacker nodes filling the connectivity, there is a higher chance of executing several exploits, including:

- Refuse to relay blocks and transactions from everyone, effectively disconnecting you from the network.
- Only relay attackers' blocks, effectively putting you on a separate network and leaving you open to double-spending attacks.
- Relying transactions with no confirmations, the attacker can filter out certain transactions to execute double-spending attacks.

*** Note -** Bitcoin makes these attacks more difficult by only making an outbound connection to one IP address per /16 (x.y.0.0).

## NODE SECURITY - CODE BUGS AND CVES

Steven W

The Bitcoin code is open-source and considered a very mature project. The critical sections involving security issues have been reviewed by many computer security experts, and the source code is being updated less frequently over time.

The likelihood of critical vulnerabilities is diminishing over time, but not out of the realm of possibility. There have been, in the past, some vulnerabilities that can be found in a CVE database located here: https://en.bitcoin.it/wiki/Common_Vulnerabilities_and_Exposures

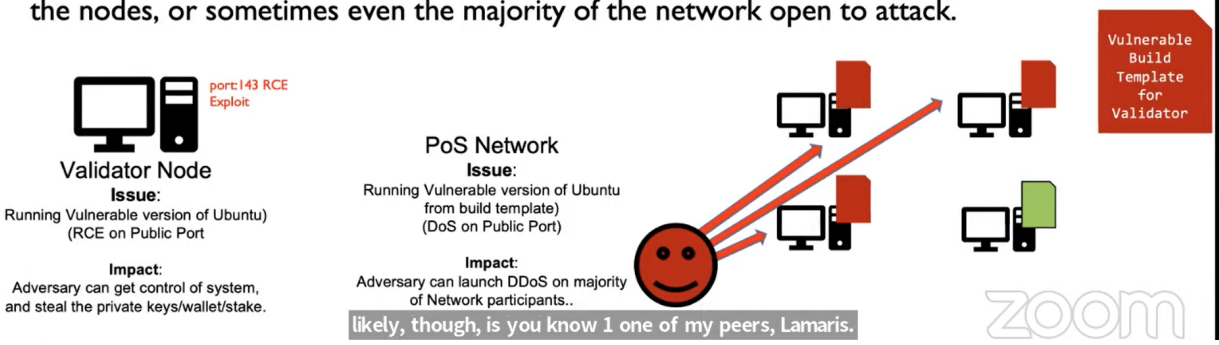
| CVE | Announced | Affects | Severity | Attack is... | Flaw |
|---|---|---|---|---|---|
| Pre-BIP protocol changes | n/a | All Bitcoin clients | Netsplit[1] | Implicit[2] | Various hardforks and softforks |
| CVE-2010-5137 | 2010-07-28 | wxBitcoin and bitcoind | DoS[3] | Easy | OP_LSHIFT crash |
| CVE-2010-5141 | 2010-07-28 | wxBitcoin and bitcoind | Theft[4] | Easy | OP_RETURN could be used to spend any output. |
| CVE-2010-5138 | 2010-07-29 | wxBitcoin and bitcoind | DoS[3] | Easy | Unlimited SigOp DoS |
| CVE-2010-5139 | 2010-08-15 | wxBitcoin and bitcoind | Inflation[5] | Easy | Combined output overflow |
| CVE-2010-5140 | 2010-09-29 | wxBitcoin and bitcoind | DoS[3] | Easy | Never confirming transactions |
| CVE-2011-4447 | 2011-11-11 | wxBitcoin and bitcoind | Exposure[6] | Hard | Wallet non-encryption |
| CVE-2012-1909 | 2012-03-07 | Bitcoin protocol and all clients | Netsplit[1] | Very hard | Transaction overwriting |
| CVE-2012-1910 | 2012-03-17 | bitcoind & Bitcoin-Qt for Windows | Unknown[7] | Hard | Non-thread safe MingW exceptions |

## NODE SECURITY - INFRASTRUCTURE AND CONFIGURATION I

Steven Walbr...

In many Proof of Stake networks or "side chain" systems, there are sometimes a limited number of participants running validator nodes. And oftentimes, the validator nodes being used are built with templates that may not consider security hardening.

If the majority of nodes are built with misconfigured or vulnerable infrastructure, it can leave the nodes, or sometimes even the majority of the network open to attack.




likely, though, is you know 1 one of my peers, Lamaris.

## SMART CONTRACT VULNERABILITIES

Steven Walbr...

| SOME TYPES OF SMART CONTRACT VULNERABILITIES | |
|---|---|
| Reentrancy | Intra- and inter-function loops that allow faulty logic for attacks to re enter into functions. |
| Access Control Bypass | Access to critical private functions or funds via public methods. |
| Arithmetic | Integer overflows and underflows and other miscalculations in mathematics. |
| Logical Errors | Unexpected Contract execution from bugs in the smart contract code or unintended logic. |
| Oracle Manipulation | Manipulation of external data providers. |
| Bad Randomization | Use of insufficient or predictable randomization. |

$1.3 Billion dollars was stolen in 2021 due to Smart Contract Vulnerabilities.

https://swcregistry.io/