# Analysis and Defense mechanism for Blockchain Application Security Issues

Sheikh Mohamad Arsalan
School of
Computer Engineering
University of technology Sydney
Sydney 2000
Email: http://mohamadarsalan.sheikh@uts.edu.au

Dr Farookh Hussain
School of
Computer Engineering
University of technology Sydney

***Abstract*—Recently, many existing blockchain implementations have encountered major bugs and vulnerabilities within the application layer. Although the literature includes a lot of suggestions for protecting blockchain, these methods focus primarily on illustrating the validity or absence from a certain sign of weakness in the transaction, but it can not secure executed (legacy) contracts from attack. Applications running on top of blockchains. Rich technologies and frameworks necessarily contain many security vulnerabilities, which have little equivalents in pure cryptocurrency implementations such as Bitcoin. Since blockchain is a new and emerging technology, it is important to have a detailed and comprehensive perspective on its security from a detailed viewpoint which is inaccessible. The study/survey presented, which can be used as a reference, explores the void to the best of our knowledge. In general, here we are systematizing two Blockchain system security elements: vulnerabilities and defenses. Along with other things, we present insights on the root causes of vulnerability, the consequences of threats and the ability to defend, and shed light on possible directions for research.**

## I. Introduction

The block chain principle was formally adopted in 2008 as the key conceptual method of cryptocurrency called Bitcoin [1], which incorporates a transaction-centered structure such as UTXO. In this kind of model, blockchain is a decentralized and public ledger, which records payment transactions between participants over a peer-to-peer (P2P) network. Besides traditional digital cash systems [2] in which there is a trusted foreign entity (e.g. bank), there's no approved third party in general in a blockchain network, and in Bitcoin specifically. As Blockchain1.0, Bitcoin has often been connected to here since it offers payment services only. The breakthrough of the Bitcoin state has always been its consensus protocol, which enables mutually untrusting nodes in a P2P ecosystem to eventually reach agreement on the outcome after execution of payment processes. With the exception of traditional agreement protocols[3], participants come from an open network and are both driven by both the flow of Bitcoins (or BTCs) that are both "mined" by some kind of tricky cryptographic hash called Proof-of-Work (PoW), an innovation initially proposed as an anti-spam strategy.[4] The aim of this survey is to dig deeper into blockchain protection, which requires a source of formalised treatment in security-related matters. Purposeful detection of found vulnerabilities, not a systematic study of their root causes; this may explain why there are still a number of security holes accessible. A foundation of good practises and core values is expected from a practitioner's perspective. In summing up other best practises, the technology sector has done due diligence, but this can also confuse practitioners. Therefore it is much more useful to have a set of key principles which are easier to establish in practise. From an individual student's point of view who wants to know and understands the protection of the blockchain network, a need for a brief but detailed and organised guide that also offers more specific information reference is required.

### A. Our contributions

We provide a systematic and comprehensive survey on

1) Industry has developed a considerable set of best practices to guide smart contract development. When properly executed, these best practices can indeed prevent several vulnerabilities.

2) There are proactive defenses that can protect against attacks that leverage several vulnerabilities; on the other hand, current reactive defenses can only defend themselves against attacks that exploit a few vulnerabilities.

### B. Paper outline

Section II describes the framework layer briefly, and addresses the methodology for the survey. Section III describes the drawbacks of Blockchain based on both the layers of the various design of the blockchain and a thorough analysis of its root causes. Section IV ends this article.

## II. Application Layer: -

The application layer consists exclusively of end-user services and applications placed on top of blockchains; hence the potential risks are often specific to different types of operations. Nonetheless, there are several categories at application level that are widely used for other high-level applications.

The blockchain security group includes five categories: (1) crypto-tokens and wallets; (2) exchanges; (3) oracles; (4) file systems; (5) identity management; and (6) securitization. While wallets offer safe storage of both the keys needed for encryption in all blockchains, crypto tokens (guarded by

wallets) are usually found in public blockchains, to often increase participation well into the consensus protocol but also for purposes. Application users need to exchange securely with different blockchain-tokens of the same network, or between different blockchains that are the focus of the trade category. Even though blockchain is an autonomous network, the secure access of knowledge from the outside world, which is also the task of both groups of oracles, is crucial for many applications. In addition, several blockchain systems require constant data storage, and in some cases might be extremely storage-consuming, and would therefore be prohibitively expensive on the blockchain for native hardware. Therefore the group of filesystems addresses these issues and also provides options on the fully distributed blockchains themselves for the local information storage. Many blockchain projects include delegated (and very often authenticated) names / identifications of entities having its respective public keys.. It must be noted that although the standard application layer for architecture is layered on top of many other layers, we stress that certain applications also need parts of their features to be integrated into lower layers. For example, decentralised file structures can implement data storage as an application-layer service with both the proof-of-storage algorithm based upon consensus.

## III. Application Attacks:-

Countermeasures and their effectiveness related to the attacks surface of

| APPLICATION LAYER Attacks | Defense | Blockchain | Miners | Mining Pools | Exchanges | Application | Users |
|---|---|---|---|---|---|---|---|
| Blockchain Ingestion and Anonymity | Tumbling/Mixing | X | X | X | X | | X |
| | Conjoin | | | | | | |
| | Zero coin | | | | | | |
| | Coin Witness | | | | | | |
| | Stealth Address | | | | | | |
| | Buy/Sell Bitcoins in Cash | | | | | | |
| | Ring Signature | | | | | | |
| | Zero-knowledge proof | | | | | | |
| | MimbleWimble/Bulletproofs | | | | | | |
| | Dandelion | | | | | | |
| Double-Spending | Transaction verification methodology | | X | | | | X |
| | XMSS | | | | | | |
| | Enhanced observers | | | | | | |
| | Peer Monitoring | | | | | | |
| | Prevent Incoming Connections | | | | | | |
| | Reducing mining difficulty | | | | | | |
| Crypto jacking | Mine guard | | X | | | | X |
| | Message-based crypto jacking | | | | | | |
| | Analyzing code of crypto jacking scripts | | | | | | |
| | Blocking specific URLs | | | | | | |
| | Mining behavior detection | | | | | | |
| | Dynamic opcode/network analysis using Machine Learning Techniques | | | | | | |
| Wallet Theft | BlueWallet | X | X | X | | | |
| | Secret sharing schemes | | | | | | |
| | Time Delay | | | | | | |
| | Encrypt/Backups | | | | | | |
| | Offline Wallet | | | | | | |
| | Multi-Signature Wallets | | | | | | |
| | Wallet insurance | | | | | | |
| Attacks In Smart Contracts | Vulnerability Detection Tools | X | X | | | | |
| | Taint Tracking | | | | | | |
| | Using Different high/low level Languages | | | | | | |
| | Fuzz Testing | | | | | | |
| Replay attacks | Integrating chainID in transactions | | X | | | | |
| | Simultaneous Submission | | | | | | |
| | Utilizing Locktime | | | | | | |
| | Signing Message with Security Certificate | | | | | | |
| | Timestamps Authentication | | | | | | |
| | Account Sequence Number | | | | | | |
| | Identity Binding to the User Account | | | | | | |

The Blockchain and the associated peer-to-peer network are independent of either of the services that use it. Based on the nature of the Blockchain implementations, they have their very own vulnerabilities and danger surface. As a consequence, we expect a large number of incidents relating to different applications that we address in this section. Our work mainly revolves around innovations such as blockchain and smart contracts. network partitioning, de-anonymization, and attacks on availability. Countermeasures include security of accessibility, naming, routing, data privacy and anonymity.

Core blockchain-oriented services for the network layer are peer management and discovery that rely on the internals of the underlying network, such as domain name resolution (i.e., DNS) or network routing protocols (i.e., IP LAN routing, WAN routing, such as BGP). They address the advantages and disadvantages of private and public networks, and their risks to security affecting blockchain.

We present a taxonomy of the attacks against blockchain's public networks, their origins and their defences.

### *A. Blockchain Ingestion and Anonymity: -*

Digital blockchains have a weak notion of anonymity, and provide open data for digital access. As either, an examination of the public Blockchain can disclose useful information to an opponent. This process is called as blockchain ingestion, and may not be appropriate for Blockchain or its users use. An actual market credit card company, for example, could use data analytics to delve into the public information of the Blockchain and refine its own schemes to compete with the digital currency. Flederet al.[70] used graph analysis to create direct links between Bitcoin Blockchain data and associated wallet user identities to demonstrate potential misuse of the public data. Two attackers exploited the public nature of Bitcoin Blockchain in 2013 for engaging in fraudulent transactions and creating a fake demand for bitcoins at multiple exchanges. Main target for Attack was Mt. Gox; the largest Bitcoin exchange in Japan in 2013. The attackers have conducted a series of fraudulent transactions at Mt. Gox. Since the Blockchain is public, the transaction rate has been found by other exchanges, and the overall demand for the coins was thought to have risen. Consequently, the price of Bitcoin rose from $150USDto$1,000 USD by the end of 2013. The attackers trading at Mt. Gox were not sponsored by the actual coins, which eventually led to bankruptcy on the exchange. Illegal behaviour. Anonymity of Blockchain-based cryptocurrencies offers reprobates valuable opportunities to participate in fraudulent activity.

Accordingly such, cryptocurrencies are become a common source of money transfer for illicit activities linked to the Deep Web[71],[72]. The use of fiat currency leaves traces on Blockchains which can be traced by law enforcement, cryptocurrencies at the other hand preserve the privacy of the consumer. It is a key reason why use of cryptocurrencies has been banned by various countries[73].Blockchains are deceptive, append-only and decentralised; they can not be reversed once a transaction is created. This has resulted in several irreversible online fraud operations, where users are tricked into sending money via BitcoinATMs. Indeed, the lack of a centralized authority makes it harder to disclose fraud, and to expect a refund. Thus, design of Blockchain technology can be misused to facilitate drug trafficking and online fraud.

Defenses: -

(a) Tumbling/Mixing: -

Bitcoin tumbling (combining) is the mechanism through which a third-party provider can be used to break the link between sending coins to a Bitcoin address and the address they are sent to. It's a method of protecting the anonymity of your BTC by combining it with other people's coins, or fresh coins. Since the Bitcoin blockchain is a decentralized ledger that records every transaction, it is necessary to merge coins for someone who doesn't want the whole world to know exactly where they send and store their BTC, or where they get it from. When using Dark Net Markets (DNM) we must exercise caution. New tools are constantly being built to increase capacity of the public, as well as private firms and government agencies, to track the coins through the blockchain and monitor who uses them. Coin tumbling systems effectively stack together some kind of bunch of transactions like a card deck where everyone eventually gets the money back while shambling the origins of the funds. For example, there have been lots of websites using Tor that will mix your coins on the clear network or over the secret network. Many sites provided a mixing service, such as Bitcoin laundry, Bit mix, Bit laundry, and even Blockchain.info inside their app.

You need to know the private key for an address and the transaction ID of an unspent transaction which transfers the coins to that address to complete a bitcoin transference. Just not send someone 1BTC, you send them 1BTC out of the 1.2BTC you got from personX. In several other words, the sum of the unspent transactions is an address balance and every transaction shows where the bitcoins originated.

Through sending coins from you to others and coins from them to you, a tumbler attempts to break the ties between the old address and a new address. Also it randomises the transaction amounts and occasionally adds gaps in transaction time.

Additional Efforts: -

The Tor Browser should be ideal for most users but use secure online operating system like TAILS if you need to make sure you don't leave any trace. Only use those services which do not process cookies or any other personal data. The addresses are personal burner. There are lots of private burner online services in there, since you can receive multiple emails when you register your existing wallets. Delete your nodes: Throughout the method we suggested merging IDs and PGP guarantees into an encrypted note to copy your address details. When you're finished remove the note and the bitcoin goes into your wallet.

Different wallets: If you want to lie low using independent wallets or even multiple accounts. This makes it more difficult to handle your bitcoin but that's the price you have to pay for being invisible at all times. Every time a new e-mail addresses is created with new wallets and accounts. Bitcoin mixing services could be monitored closely by government bodies and others, so any wallets to it or through which you transfer funds can also be tracked. Disable JavaScript: Could be used to detect or delete malware on your system. Double test onion links: Tor secret websites do not have SSL certificates, sometimes called onion pages. Which means you can hardly tell whether you are using a legitimate website or a fraudulent one.

(b) Coin Join:–

CoinJoin is a[88 ] cryptographically secure way to combine multiple Bitcoin payments from different spenders into one transaction to make it more difficult for third parties to decide which recipient or receiver the spender is paying.

Amy wants to transfer 1 BTC to account A to address Robert wants to transfer 1 BTC from account C to account D CoinJoin allows them to combine their transactions with two (A and C) inputs and two (B and D) outcomes in a single transaction. Anyone who investigates the blockchain is also no longer sure of which of the outputs from Amy, and which one is from Robert. The parties in the joint transaction (there could be more than two of them) need not trust each other because of some cryptography mystery behind the scenes. There's no chance the coins would be stolen compared to traditional Bitcoin mixers. CoinJoin transactions are light weight, and that there are no additional charges (besides standard transaction fees).

(c) Zero coin :-

Zerocoin[89] is a possible extension of the Bitcoin protocol, adding true cryptographic anonymity to transactions with bitcoin. Since Bitcoin transactions are held in a public ledger (in the block chain) they can trace it back the history of any transaction. Zerocoin allows for anonymity by implementing a new cryptocurrency called zerocoin which is contained in the block chain of Bitcoin. Zerocoins are acquired via a zerocoin mint exchange with both the base currency in set demonisation. Later, these zerocoins could be transferred to some other base currency address via a zerocoin spend transaction. Using cryptographic accumulators and proven authenticated guarantees of zero-knowledge, the account used to mint the original zerocoin can not be connected to the address only used to retrieve the zerocoin. (d) Coin Witness: -

In this they write down a small programme for your[ 93]selected verifiable off-chain method which verifies the accuracy of one of those transcripts. The programme requires the last transaction of the transcript to all be special in that it pays to a p2sh / bitcoin scrippubkey. The programme always requires to have the same address as a public input. They call this machine a "witness" as it bears witness to the transcript and agrees if and only if the transcript is correct. Then, the SCIP proof scheme is used to turn the programme to a verification key. If someone wants to create a Bitcoin in an off-chain network they pay the coin to the hash of that verifying key. Users therefore transact in the off-chain network, as they wish. To ensure that perhaps the system works correctly, they will repeat the computationally expensive key generation process to guarantee that it conforms to the transaction rules they expect. If a consumer decides to leave the network (to compact their history, move to another system, spend plain bitcoins, or for any reason), they take a final transaction paying to a Bitcoin address and run the witness on their SCIP transcript and show evidence. We create a Bitcoin

transaction which proves the coin in its script (and not the text, that is kept private), and the Bitcoin network verifies the proof and transaction performance. The public knows nothing about initial transactions, increasing fungibility, unlike other ideas that improve fungibility, this concept has the potential both to enhance Bitcoin's scalability and to safely integrate new and innovative alternative transaction methods, and to expand Bitcoin's zerotrust nature to more transaction types.

(e) Stealth Address:-

A form of encrypted[ 95] address that can be freely shared, where no blockchain analysis can connect any payment to it. A stealth address is a privacy-enhancing device designed to protect the privacy of those receiving cryptocurrencies. Stealth addresses allows the issuer to set random, one-time address for each transaction in favour of the recipient, so that different payments made to the same payee can not be connected. The simplest stealth address procedure was initially created in 2011 by a Bitcoin Community representative named ' ByteCoinwho relies on the protocol DiffieHellman Elliptic Curve (ECDH)

The sender and receiver have private / public key pairs (d, D) and (e, E), where D=d·G and E= e·G and G are the base of an elliptic curve group, respectively. Both the sender and the receiver will use ECDH to measure a shared secret c: c= H(d·e·G)= H(d·E)=H(e·D), where H(·) is a cryptographic hash. The sender simply uses c·G as its ephemeral destination address to transmit the invoice. Unless some transaction has been sent to the supposed c·G destination address, the receiver will actively monitor the blockchain and check it out. If he does have, you can use the corresponding private key c to spend the check.

Dual-Key Stealth Address Protocol (DKSAP)[96][97] dual-key upgrade[98][99], DKSAP, was developed by a programmer such as rynomster / sdcoin for ShadowSend, an open and anonymous decentralised wallet solution. Many cryptocurrency schemes, like Monero, Samourai Wallet, and Token-Pay, have since implemented the DKSAP to just mention a handful.

(f) Buy/Sell Bitcoins in Cash :-

Buy or sell Bitcoins in cash often provides an anonymous form of trade with Bitcoin. Bitcoin not only using exchanges could be purchased anonymously. Alternatively, you could use apps such as Localcryptos that provide a face-to-face anonymous connectivity that incorporates Escrow services.

(g) Ring Signature :-

In cryptography[100] a ring signature is a form of digital signature that a participant of a user team may execute with a cryptographic key. A transaction message authorized with a ring signature is verified by anyone in a specific group of people, without revealing the sender, receiver, full member or any of the other members of the organization's public key (or identity) while maintaining the transaction number. Like Bitcoin, this signature implementation algorithm uses a technique called a hash-based public key + private key. The problem is, incorporating the ring signature technology would combine the transaction sender's public key with other public keys, and only then it sign the data. It gives Monero the ability to remove information about the sender's address, preventing external attackers from targeting the sender. HCASH[101] Bondable ring signatures post-quantum. It is a robust Ring CT protocol, so it contains all of the necessary components including a connectable ring signature (for user anonymity), commitment scheme (for protecting the transaction sum) and range proof (to guarantee the secret value is a valid amount). All parts is in a lattice-based setting, meaning the overall protocol is post-quantum stable.

(h) Zero-knowledge proof :-

The addresses Zcash[102][103] are either private or transparent urls (t-addresses). Z-addresses begin with the "z," and t-addresses begin with a "t." A Z-to-Z transaction happens on a public blockchain, it's known the fee was paid. Nonetheless, all addresses, transaction numbers, and memo area are encrypted and not publicly accessible. Service providers that are neither the sender nor the receiver of a transaction are unable to obtain any data on a transaction being encrypted—also the miner concerned for recording the transaction can not obtain the coded address as well as the transaction amount. Whenever an encrypted transaction is released, the miner can only say "there is an unspent balance and a transaction is produced" and not allow the blocked address and the sum of the transaction to just be reported by itself.

(i) MimbleWimble/Bulletproofs :-

Bulletproofs[104 ], a recent zeroknowledge claim for knowledge1system that demonstrates that a secret committed value within a given interval. No confident setup is required for the Bulletproofs. We are believed by a discrete logarithm, and made non-interactive using the Fiat-Shamirheuristic

(j) Dandelion :-

Dandelion[105] is a new framework for transaction transmission that reduces the risk of hackers connecting transactions to the source IP. The dandelion transaction is propagated in two phases: first, the step of "running," then the process of "fluffing." The node then relays that transaction during and after stem cycle to a* single* pair. After only a random number of hops across the foundation, the transaction enters the fluff process which works just like ordinary flooding / diffusion.

### *B. Double-Spending :-*

In cryptocurrencies, double-spending is to use a one-time transaction twice or multiple times. Take the following case, via an example to illustrate double-spending. In cryptocurrency processes, a transaction transfers asset rights from the address of a sender to a public address of a recipient, and the signer with a private key identifies the transaction's value. After the contract has been signed.It will be sent to the network on which receiver can validate the transaction. Validation at the recipient's end happens once the receiver checks the unused transaction output of the issuer, verifies the signature of the sender and waits for the transaction to be mined in to a legitimate block. The process will take a couple of minutes, based on the size of both the mempool, the network latency, the transaction priority factor, and the computation time for the cryptocurrency block. For Bitcoin the total mining time

for the block is 10 minutes. They can release the product to the sender in such a swift transaction environment[74],[75] or if a receiver is positive, before the transaction is mined into the Blockchain. As such, this enables the sender to sign the same transaction and pass it to another recipient. This practise of signing and submitting the same transaction to two different receivers using a private key is known as double-spending. In dual spending, two transactions are extracted from the same unclaimed transaction output of the sender and one of them is incorporated into the Blockchain

In Figure 16 we demonstrate how to execute a double-spending attack in a cryptocurrency. Consensus delay in network section V-F, BGP attacks, flood attacks on mempools, or the 51 percent attack section V-B which causes extra latency throughout the verification and propagation process that increases the likelihood of double expenditure of an opponent. In March 2013, due to a soft fork, a lucrative double-spending transaction worth $10,000USDwascarriedoutatBitcoin.$

Defenses: - (a) Transaction verification methodology : -

Examine the means[106] by proposing a viable transaction verification method to solve the above security problem; aim a popular payment platform which integrates various vendor-based digital currencies together. The key elements of transaction cooperation are recognized as currency miners and consumer applications. A scenario-based transaction verification model is developed considering transaction patterns among miners and consumer applications. By building a trustworthy framework between currency miners on the payment platform, the bitcoin-like concept of' trust network' is introduced when verifying transactions, using digital signatures alongside SHA-256 hashing and RSA algorithm. In enhancing the extent of verification, an identification method is defined, correlated with the minimum necessary level of probability. However, a time limit was set for a provided transaction to also be performed with proper verification, based on the requirements of the peer-to-peer network. (b) XMSS:-

One-time signatures[ 110] (or several times) indicating the user's private key if he wishes to double spend. Nonetheless, it needs a change from the current signature algorithms used by applications in Blockchain. These are stately schemes which typically produce shorter signatures, but they have a system of state-keeping (what paths / keys are already being used). The core idea of hash-based signature scheme would be to combine a larger number of one-time key pairs in one structure in order to achieve a more than once (although a limited number of times) usable signing process. It is accomplished using Merkle tree system, including different permutations. A public and private key are generated from both the various public and private keys representing it in the one-time system. (c) Enhanced observers :-

Extends the hybrid approach[107] which integrates key ideas from prior work. They simulated ENHOBS, and demonstrated which we can effectively combat the double-spending attack in fast pay transactions for equal overheads in terms of number of ENHOBS in the network. Hybrid observator and peer-warning network. The ENHOBS will carry out detailed inspections for all transactions received within this programme, and will evaluate all outputs and inputs. Upon discovery of a double-spending attack, the P2P network will submit an error message that includes both transactions as evidence. Once an alert has been issued and reviewed, any transactions that match the same inputs are removed from the memory pool immediately. ENHOBS transmits all incoming transactions and packets, as does every other participant in the Bitcoin P2P network, as well as alert messages in order to distinguish their traffic patterns from other peers in the network. (d) Peer Monitoring : -

When a peer[ 108] detects a new transaction it checks if the transactions were the coins that have not been spent on any transaction that occurs in the block chain and in their memory pool. If so, Bitcoin's current protocol will be followed by peers; peers will link and then forward transaction to their networked memory pool. If, on the other hand, peers find another transaction that spends the same coins in their memory pool on different targets, peers then forward the transaction to their neighbours (without adding the transactions to their memory pools). The core theory behind this strategy is that while A can prevent TRA from being given to V and a subset of V's observers, a considerable number of Bitcoin peers get TRA and TRV both. If most of these peers are honest14, in seconds all transactions will eventually reach V. So, the double-spending of A can be observed before A expects to receive the service from V. (e) Prevent Incoming Connections: -

Neither[ 109] should the merchant find transactions that are coming in. Find two Ta and Tv Transactions To carry out a double-spending attack the attacker must generate two TA and TV transactions. All transactions invest the same output and thus can not be true for both. TV represents the transaction that transfers the amount needed to the merchant, while TA is a transaction that returns the same amount to the attacker. Thereby, the intruder cannot explicitly deliver TV to the merchant. Forcing the attacker to relay it over the network will make sure Tv ends up with those nodes that forward it to the local view. Such nodes use the same inputs to correctly ignore subsequent transactions, e.g. Ta, softening the likelihood of TA winding up in the public ledger. (f) Reducing mining difficulty: -

Reducing the complexity parameter of the Blockchain to allow for fast block mining, which would be a reasonable approach, except it would only promote greedy mining and the stale block rate.

### *C. Cryptojacking: -*

Cryptojacking is a form of attack launched on web and cloud-based services to perform PoW illegally for Blockchain-based cryptocurrencies without consent[76],[77]. The most recent and most prevalent form of cryptojacking is in-browser cryptojacking which transforms websites into mining pools[78]. PoW requires intensive mathematical calculations by the processor which usually involve finding a target hash value. As the aggregate hash rate of the cryptocurrency net-

work increases, so does the related complexity of computing a block. Cryptojacking is a type of attack launched on web and cloud-based services to illegally perform PoW for cryptocurrencies based on Blockchain without consent[76],[77]. The latest and most prominent form of cryptojacking is cryptojacking in the browser which translates websites into mining pools[78]. PoW enables the processor to make intensive mathematical computations which typically involve finding a target hash value. As the crypto-currency network's aggregate hash rate increases, so is the associated computational complexity of a node.

a) Cloud-based cryptojacking: In order to compensate for this, malicious miners have found a way to increase their hash power by hijacking remote computer processors for mining. This attack is known as a covert mining or cryptojacking attack and involves hijacking a target system for the attacker to perform PoW calculations. Such attacks were initially launched against cloud service providers, where malicious users performed To counter this, malicious miners have figured out a way to boost their hash power by hijacking external mining computer processors. This attack is classified as a covert mining or cryptojacking attack, which involves hijacking the attacker's target device for conducting calculations of PoW. These very attacks against cloud service providers are initially launched, where malicious users performed covert mining operations on virtual machines and server resources is depleted. This activity was first noticed by Tahiret al.[80], suggesting defensive measures in the type of a software tool named "MineGuard" to effectively detect and avoid covert mining in the cloud.

b) Web Cryptojacking:

It has been brought onto the web and is becoming increasingly popular. Web based cryptojacking is being used by attackers that insert malicious JavaScript code into websites that secretly mine tokens with out the permission of their users. At browser-based cryptojacking the web browser of the client computer executes JavaScript code that establishes a direct connection to WebSocket from the dropzone server. The server then sends the target to the client, which is computing hashes that PoW and transmits back to the server. Throughout this process the machine owner is still unaware about this background activity and proceeds to search the website smoothly . Not only does in-browser cryptojacking pose a major threat to privacy, it also hurts the performance of the visiting computer, since PoW-based hash computations are processor-intensive and can result in unnecessary CPU usage and battery drainage. Such services connect websites to their portal service and make website visitors cryptojacking on computers. Coinhive is by far the most prominent cryptojacking forum on website and is linked to the Monero cryptocurrency . We include the JavaScript cryptojacking code that intruders use to connect victims ' websites to the Coinhive account. The code listing indicates that a WebSocket link with a coin hive server is generated whenever a client loads a coinhive.min.js file, and the victim's key is passed to link to the dropzone. It then gets a goal and sends the respective hashes to the server via the same socket connexion. In-browser cryptojacking is a relatively new attack that is applicable to the PoW-based Blockchain applications, so there is no prior research done that looks at the process and implications of this attack. However, due to the incidents reported in the news, it can be inferred that cryptojacking is becoming popular in recent years. The academic community considers cryptojacking as a growing threat to the security and privacy of Bitcoin systems. Symantec's new Internet SecurityThreat Report (ISTR) shows that website cryptojacking attacks increased by 8500 per cent in 2017. In the community an ethical debate has arisen over whether cryptojacking should serve as a substitute for online advertising. Those of us who advocate the method pointed out that users that provide a mining website with their computing power can use the website without first seeing online advertising.

Defenses: - (a) Mineguard: -

A tool[80] that can detect mining activity in real time through VMs or processes in mining pools, and prevent abuse against an aggressive adversary attempting to circumvent defence. Our system includes hardware-assisted profiling to develop discernible signatures for different mining algorithms and can detect them with negligible overhead for both CPU and GPU-based miners (¡ 0.01 per cent). MineGuard, a simple hypervisor system based on hardware-assisted monitoring of the conduct which reliably detects the signature of a miner. In particular, our system uses Hardware Performance Counters (HPCs) to reliably track low-level mining operations or events with low overhead inside the Processor and GPU. Open source random forest library[112] for implementing the bagged decision tree, and for profiling the CPU and the GPU. (b) Message-based cryptojacking: -

Instead of blocking individual URLs, the plugins[119] will monitor the messages exchanged between user and server during the cryptojacking session. If the messages are obeying the chain of Web frames, the extension can flag them as cryptojacking. This will avoid cryptojacking, even when WebSocket requests are forwarded through a third party. (c) Analyzing code of cryptojacking scripts: -

They discern them and form malicious and benign messages. Furthermore, under normal operating conditions, the performance of the client system during cryptojacking is different from that of the server. Based on these features, an effective protection system can be built which can identify the cryptojacking sites during web browsing. Search engines and web crawlers may use these classifiers to further identify and accurately warn users of cryptojacking websites, or insert them into "protected-browsing" lists. (d) Blocking specific URLs: -

Blocking specific urls can also help stop the mining of coins using your valuable power of calculation. Some of the examples below are: - No Coin [120]: -The extension blocks a set of blacklisted domains in xyz.txt Anti miner-coin minerblock [121]: -Anti miner blocks Javascripts which are mining coins for Monero, Dash and others like coinhive (coin-hive) and jsecoin. The extension clearly blocks No Mining [122] a list of blacklisted domains:-It also blocks unique malicious domains

contained in a registry . (e) Mining behavior detection: -

It seems that there is [123] a particular way to start the mining process which runs through inline script of the website. For example, if the mining script consists of the signature behaviour reference such as form of all this (any name) where the method, string, number or unknown text can be used, it will cause the operation to stop and neutralise the line. If there is a possible mining script the cycle will be killed and completed. If none is present then the process of detection will end immediately. (f) Dynamic opcode/network analysis using Machine Learning Techniques: -

Dynamic opcode tracing is able to detect ransomware at a time comparable to static analysis, without being hampered by obscure tactics. There are two models which use the same technology .

i) This techinque[124 ] is used in the Crypto-Aegis, a Machine Learning (ML) system Distinguishing between cryptomining sites, armed benign sites (e.g. benign sites to filled with crypto-mining code), de-weaponized crypto-mining sites (e.g. crypto-mining sites excluded from thestart()call) and modern world-benign sites. ii) CapJack,[125] a learning-based machine detection framework capable of detecting malicious in-browser cryptocurrency mining activities. This method utilises CapsNet, a machine learning algorithm that imitates biological neural organisation. To implement a host-based solution, CapJack uses system features such as CPU, Ram, Disk and Network.

### *D. Wallet Theft: -*

If passwords are kept in a digital wallet, such as keys shared with peers on the network, then the "wallet robbery" assault has some device implications. For starters, the wallet is stored unencrypted in Bitcoin by default, allowing a competitor to know the reputation associated with it, and the nature of the transactions it makes. Eventually, with several third-party providers allowing storage of wallets, these systems can also be hacked and the wallets of a rival can be leaked [81].

Key disclosure and Theft :- Exposure and theft of private key in Blockchain-based cryptocurrencies is a well-known problem. When the attacker obtains a user's private key, they can sign and generate a new transaction onto the user part, and therefore can spend the savings on unapproved receivers. Brengelet al.[82] examined key vulnerability in Bitcoin by reviewing the Bitcoin blockchain for reuse of ECDSA nonce. Their results show that ECDSA nonce reuse is abused in Bitcoin to build user-specific transactions. Breitneret al.[83 ] also carried out cryptanalytic assaults on Bitcoin, Ethereum, and Ripple to reveal their private keys. They used a particular algorithm to test private ECDSA keys used in biassed signatures.

Vulnerabilities of software clients:- Open blockchain applications like Bitcoin and Ethereum have clients with open-source software that allow users to connect to the network. New software versions are released regularly, introducing newrules and enhancements. An earlier version also releases an improvement which allows fixing vulnerability. In Bitcoin the Corev0.15 and below are prone to denial-of-service attacks. This vulnerability was fixed in the v0.16 released recently The newly released version, however, fails to support all nodes. An intruder can also exploit the available-source code with ransomware and bugs to release a new version. When the malware is downloaded by a user, it can allow access to the attacker who could perform numerous attacks including DDoS, theft balancing etc. Consequently, the client software must be downloaded from a trusted network. Defenses: - (a) BlueWallet.

This[113] hardware token is used in conjunction with a machine that is connected to the Bitcoin network, just like the user's computer. The machine will prepare a Bitcoin transaction, but will not be able to record it. Members can use BlueWallet to screen and sign the transaction. Instead, the computer must relay the transaction signed to the Bitcoin network. The tightly protected cryptographic signature never leaves the computer and is accessed when the user clicks his PIN correctly.. And it can also be used as both an electronic wallet: the app can be used to render Bitcoin payments directly in conjunction with a point of sale (POS) connected to the Bitcoin Network. BlueWallet provides the consumer with a simple and quick solution to secure bitcoins, while acting as an alternate to cash and credit cards.

(b) Secret sharing schemes: -

Protected exchange mechanisms[114 ] have been used in various Bitcoin offchain and onchain wallets to protect both the crypto holders ' private keys. For example, suppose an organisation needs to hold a single private master key on its bitcoin.

Throughout this scenario, the secret sharing scheme lets many people store the same key. A small example of such a situation was the exchange of a bitcoin wallet key by distributing key shares among three individuals. These individual shares may not convey the information that is really important. Even so, as shown in, any 2 out of 3 people can recreate the key using their shares. Hidden storage schemes can also help blockchain through decentralised processing of secret information to limit access by unauthorised parties. (c) Time Delay: -

Holding[115] bitcoin as something of a single account unit might position this in cold storage, or keep it offline, via code suggesting it can be spent, but not immediately. The owner may set certain delay predetermined on a attempt to expand the coins. The reason it does have an integrated-in delay provides a real owner time to fix a transaction whether it breaches their personal information or if someone tries to steal their crypto. d) Encrypt/Backups :-

Stored in a safe place, a backup of your wallet[116] protects yourself from machine failures and several human errors. When you keep the wallet safe, it also can help you recover the money after your smartphone or computer has been robbed. Encrypting your smartphone or wallet enables you to set a password for anyone who attempts to steal money. It protects from theft, while hardware or software can't defend it from key logging. e) Offline Wallet :-

The offline pocket considered also as cold storage, offers the highest degree of loss security. It involves keeping a wallet in a secure place not connected to the network. That can provide very great protection from computer vulnerabilities if done properly. Combining an offline wallet with backups is indeed a best practice. f) Multi-Signature Wallets:-

Bitcoin has a muti-signature feature that allows you to spend several approvals on a transaction. An association can use this to grant its members access to its funds, allowing removal only if the agreement is signed by three of five members. Most web wallets have multi-signature wallets that hold the user accountable for their money and thus stop a hacker from stealing funds by compromising a single device or server.

g) Wallet insurance: -

Protecting the insuranceusing crypto wallet could be of help or support in fighting wallet theft.

Some of those are as follows: -

i)Curv's[118 ] crypto wallets do not use private keys, a standard technique even for a user to unlock the encrypted information ii) BitGo[ 117] Is a crypto-currency security service and trust company. The company offers a multi-signature bitcoin wallet solution that dispenses risk management keys across a number of owners. BitGo wallets typically have three keys: BitGo owns one, and the wallet owner keeps two. Wallets can be used in both hot and cold conditions, as well as in non-custodial and custodial settings.

### *E. Attacks in Smart Contracts: -*

When applications are placed on top of Blockchain, their own weaknesses together with Blockchain vulnerabilities build an attack surface again. Smart contracts are part of the Blockchain 2.0 generation in this section we'll discuss the possibilities of attacking smart contracts. Most well-known smart contract technology in the digital world is Ethereum, which uses the programming language Solidity for contract coding. Solidity[175] is a contract-oriented language, influenced by Javascript, Python, and C++. Bad programming language, execution environment and coding style can lead to a series of attacks. We display a vulnerable smart contract code in Figure 20 which steals a sender's balance. "The DAO" had a similar weakness in their smart contract which resulted in a loss of US 50 million. Some of the well-known Ethereum smart contract attacks include re-entry attacks, over and under flow attacks, replay attacks, short address attacks and reorder attacks.

Reentrancy attacks: Unless the user does not shift the balance before sending ether in the reentrancy attack, by requesting the call.value, an attacker will steal all the ether stored in the contract (method recursively in such an ERC20 token. Hence, if the lazy client forgets to modify his balance, he may lose his entire balance in the contract.

DoS Attacks: DoS Attack in Smart Contract allows a malicious attacker to keep the money and power for themselves. Consider a smart contract auction scenario in which a malicious bidder attempts to become the leader of an illustrated auction. The poor contract prevents repayment of the old contract holder, and gives the attacker the new leader. It also cancels all bid) (requests sent by other biders and keeps the attacker as the leader of the auction. In Ethereum smart contract another form of DoS attack includes breaching the gas limit set by the contract. In Ethereum, if the total gas absorbed by the smart contract exceeds the gas cap after execution, the contract transaction can be abused by inserting several repayment conditions.

Overflow attacks: When the size vector value(2256) is exceeded an overflow in a smart contract hap-pens. For eg, if someone sends out a large amount of ether exceeding (2256) in a smart online betting touch the bet value would be set to 0. Although it is impossible to exchange an ether value greater than (2256) it remains a programming vulnerability in smart contracts written by Solidity. Short address attacks: In Ethereum, the short address attack exploits a virtual machine vulnerability to create additional tokens for restricted transactions. ERC20 tokens also contribute to the attack on short emails. For this assault the attacker creates an Ethereum wallet that ends with 0digit. Then, he makes a purchase on the address by removing the last0. If the contract has enough space, then the buy feature will not test the sender's address and Ethereum's virtual machine will add missing 0 to complete the address. Consequently, the system returns 256,000 tokens purchased for foreach 1000 tokens.

Forcible balance transfer:- In unstable smart contract codes, forcible transfer of the balance to the contract may occur, without a fallback feature. This can be used to reach the petrol cap and to reject the final transaction.

Delegate call injection:- To allow reuse of data, EVM offers an opcode, DELEGATE CALL, to insert the bytecode of a street contract into the bytecode of the caller contract as if it were a part of the latter's bytecode[153]. A suspect callee contract will deliberately modify (or manipulate) the variables of the caller contract state as more of a consequence. This vulnerability is caused by the inability to change the state variables of a caller contract by by the byte code of a street contract. The vulnerability can completely be avoided by declaring a contract to be shared as a stateless library through the delegate call[154].

Frozen Ether :- The flaw comes from the users ' willingness to deposit their money into their contract accounts with the unwillingness to spend their money from those accounts essentially freezing their money. The vulnerability is caused by[155]: I contracts that do not provide any purpose to spend money depending on the money-spending position of another contract (as a library) and (ii) accidentally or intentionally break the callee agreement (i.e. library). The risk can be minimised by making sure that mission-critical projects or money-spending positions are not outsourced to another company.

Upgradable Contract :- The flaw is activated by one contract's desire to call another which can be malicious. That vulnerability (i.e. unauthorised communication updating) remains an open issue

Ether Leaking :- The danger exists when any caller may withdraw the funds from a contract, which is neither the owner

of the contract nor an investor who has contributed funds to the agreement. This weakness is triggered by the inability to check a caller's identity whenever the caller invokes a protocol for transferring Ether to an arbitrary address.

Constructor vulnerablity :- This limitation is triggered by Solidity's lack of the special syntax to differentiate a function Object()[native code ] from either a normal function[156 ]. Beginning with version 0.4.22 of Solidity, the error was eliminated by introducing a new keyword constructor[157 ].

Ether lost to unknown address :- Once the transfers have been made[158], Ethereum only verifies if the length of the receiver's address is not more than 160-bit and the recipient's address is not true. When money is sent to a non-existent orphan address, Ethereum can change the transaction's address automatically, rather than terminate it. Since the address is not associated with any EOA or contract account, there is no way to retrieve the transferred money which in effect is lost. The consequence of that limitation is that EVM is not orphan-proof. This vulnerability can be prevented only during mailing, by manually checking the correctness of the receiver's address

Timestamp dependence :- The vulnerability was first discovered at[159]. It happens when a contract uses the block.timestamp as one of the activation conditions for carrying out a critical operation (e.g. money transfer) or as the root of randomness, which a dishonest miner can still manipulate. The drawback is because Ethereum actually strictly adheres to both the fact that even a time mark has to be greater than the time mark of its parent block and it is well into future within 900 seconds of a actual clock.

Defenses: - (a) Vulnerability Detection Tools: -

Slither[131] Is a smart contract technology fixed analytics structure[47]. The security-tracking methods are safe and reliable for possible bugs. Slither can also be used to conduct primary objectives like automatic vulnerability recognition, automated detection of changes, interpretation of code and review of licenced code. Solidity parser creates a Solidity Abstract Syntax Tree (AST) from either the contract's source code, and using the AST as an input to Slither. Throughout much of the initial phase, Slither acquires important contractual knowledge such as the inheritance graph, Control-flow graph (CFG).

MYTHX [133] :- MythX is a security research service that tests smart EVM contracts based on vulnerabilities [49]. It includes various analytical techniques including static, dynamic and symbolic implementation

MYTHRIL [134] :- Mythril, an opensource tool, uses symbolic execution technique to remove code errors. Security fault analysis requires smart contract bytecode execution in such a custom-built EVM.

MANTICORE [135] :- Manticore's main functions include monitoring inputs that terminate a programme, recording instruction-level implementation and giving access to its analytics engine via the Python API. It also has a complex symbolic execution function which analyses both Ethereum's binaries and smart contracts.

SECURIFY [138] :- Securify is an automated tool which can determine whether the con-tract performs based on the specified attributes. Security accepts EVM Bytecode for security analysis. Contracts written in Solidity are allowed as an input, however the code must be translated into EVM bytecode for the entire screening process.

SMARTCHECK [140] :- With proper description and advice, SmartCheck not only highlights the weaknesses in the smart contract code but also clarifies the vulnerability triggers. To detect patterns in vulnerability, SmartCheck was implemented on intermediate representation (IR) queries using XPath[xpa ].

ECHIDNA [142] :- EVM smart fuzzer demands that the proposals for solidity conduct deep bug analysis and provide a simple user interface (UI) to improve performance.

OYENTE [144] :- It not only seeks security vulnerabilities, but also investigates every possible path of execution. This contrasts the new formula with formulas consisting of common bugs

VANDAL [145] :- It includes the declarative word Soufflé. The security experts use the model of the new research in a declarative language

ZEUS [146] :- It embraces the smart contract code and produces the same version in a framework known as XACML. The smart contract code and policy specifications are encoded in LLVM bitcode to enhance the contract behaviour.

(b) Taint Tracking: -

Taint monitoring has been a well-known technique for identifying attacks on private data leakage or memory manipulation, which was first used on a smart contract execution platform. In specific, taint analysis for monitoring data flows and control-flow decisions from storage variables. It's used to add write locks forbidding the contract to change storage variables in other declarations from one of the Ethereum transactions proposed under the same contract. Sereum[147] utilizes the same to avoid writing down variables which would make the contract state inconsistent with another re-entered result of the same contract.Sereum even rolls back transactions which cause variables to be write invalid, effectively preventing re-entry attacks.

(c) Using Different high/low level Languages:-

Different construction language used for Ethereum contracts. Makes the return to a state clear and prevents reentrancy issues. Bamboo[148]:- Use polymorphism to reduce the transactional dependency instability Obsidian[149] :- It uses smart contracts as finite state machines and tracks contracts to minimise reentrancy weaknesses Flint[150]:- Uses an asset type to ensure operational atomicity and imposes restrictions on callers ' ability to defend contract functions against unauthorised access, thereby reducing reentrancy vulnerabilities Simplicity[151] :- It provides formal semantics using the proof-assistant to systematically determine the properties of contracts (e.g., safety and liveliness) Scillia[152]:- Distinguishes in-contract calculation from inter-contract contact to differentiate between contract-specific effects.

(d) Fuzz Testing :-

Re-Guard[160 ] (al-30) aims to identify reentrancy(al-1) vulnerabilities in smart contracts by translating smart contracts into contextually similar C++ programmes and creating arbitrary transfers via a flashing tool to check the implementation paths of the C++ scheme.

### *F. Replay attacks:-*

The replay assault involves making one transaction on two separate blockchains. For instance, when a cryptocurrency forks into two different currencies, users hold equal assets on both ledgers. A customer has the option of performing a transaction through either of the two chains. Through replay attacks, the attackers sniff the transaction data into one ledger and repeat it to the other ledger. As such, all chains ' properties are lost by the customer. One basic case could be taken from Ethereum. All Blockchains in Ethereum are true for a transaction signed on one Blockchain. A transaction done on the test network can thus be replicated to transfer funds on the public network.

Defense:-

(a) Integrating chainID in transactions :-

Ethereum has taken countermeasures by incorporating chainID into transactions to prevent replay attacks, and users who do not enable this wallet feature remain vulnerable.

(b) Simultaneous Submission :-

One way of creating replay-protected UTXOs[127] that don't need a coinbase transaction or maybe one of its successors is to make two separate transactions, each sent for you to invest the same UTXO (UTXO calls for unspent transaction output) on yourself.

If everything works out, tx is mined in chain 1 and tx2 mined in chain 2. Tx1 could be replayed on chain 2, for example, however since tx2 was first sent, this will appear as double spending on nodes on chain 2 and nodes are likely to reject it. Kind of like a Chain 1 replay of tx2. This will appear as a double spending on nodes that get tx1 first.

(c) Utilizing Locktime :-

It renders the transaction illegal[127 ] when the Bitcoin blockchain contains certain block number. This is, if they define a block number of 500,000 a block before block number 500,000 thatcontains such a transaction will also be rejected by the netwok.

(d) Signing Message with Security Certificate :-

This [ 128 ] involves the development of a transaction "authorized with a security certificate and containing a [ unique ] security value" (only to be used once), meaning that a computer system blockchain network is eligible to conduct a transaction before it has been forwarded.

(e) Timestamps Authentication :-

Lightweight authentication scheme which can time stamps to prevent attacks from replaying the network. Through node communication is input with a timestamp header which can help to mitigate the attack. Furthermore, timestamps obtained from transactions are added to test the freshness of the messages and minimise replay attacks

(f) Account Sequence Number :-

A series number can defend against replay attacks per transaction. You can't recreate a sequence number that can help avoid replay attacks. As all transactions take place in sequence.

(g) Identity Binding to the User Account :-

An authentication scheme for blockchain identity[129], using a decentralised private key generator (PKG) and preventing single points of failure, but also includes other security features. The identity of the user (such as email address, phone number) is linked to a public key in this identity-based cryptography. Before create a private key for a client the PKG needs to carry a thorough check on the identity of the user. The first is to ensure the identification is uniquely defining the user and the other is to guarantee that perhaps the user is the owner of the identity. A way of linking identity to Ethereum account address.

By approving the random number R with its private Ethereum account address, the client authenticates their identity to the PKG in the key generation process. During the key distribution process, the smart contract decides whether the recipient is the owner of the identity by the Ethereum account address which calls the contract. Thus the scheme will prevent an intruder from maliciously accessing other user private keys by misleading PKG.

Replay Attack misleads the authentication server, an attacker records the contact session, and resends the entire session or part of the session to the authentication server. For such a scheme the challenge number R in a verification process is random, the temporary private key g is also randomised during the signing stage. This is why the results of signatures are different each time, and the authentication information transmitted is special each time. The attacker can not trick authentication server by replaying the captured information.

## IV. Conclusion

We have provided a comprehensive survey on the security of the Blockchain system, including its implementation, data and network layers. The survey considered, when correlating them, two viewpoints, namely attacks and defences. We addressed not only the threat positions but also the root causes thereof. We systematised the attacks against the blockchain system, and the protections for it. We further systematised the industry's current best practises into a small number of guiding principles which might be easier for practitioners to follow. We offer insights into the state of the art and the future directions of science

## References

[1] H. Kopka and P. W. Daly, *A Guide to LaTeX*, 3rd ed. Harlow, England: Addison-Wesley, 1999. [1]. S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008. [2]. D. Chaum, "Blind signatures for untraceable payments," in Advances in Cryptology: Proceedings of CRYPTO '82, Santa Barbara, California, USA, August 23-25, 1982., pp. 199–203, 1982. [3] M. J. Fischer, N. A. Lynch, and M. Paterson, "Impossibility of distributed consensus with one faulty process," J. ACM, vol. 32, no. 2, pp. 374–382, 1985.