# FTEC 5520 – Week 10

# **Agenda** - Advanced Topics in Blockchain – Week 10

1. Blockchain/DLT Security Analysis
2. Security Incidents in Blockchain/DLT
3. Privacy Issues of Blockchain/DLT
4. IT Audit of Blockchain
5. Blockchain Governance
6. Forensics of Blockchain Implementation
7. Trend, Challenges and Opportunity of Blockchain

# Blockchain Security Analysis

# Nature of threats of the blockchain attacks

- Stealing Cryptocurrencies
- Denial of Services
- Double-spending
- Other Compliance Issues and Risk Considerations

# Stealing Cryptocurrencies

Stealing of Encryption key

Theft of user accounts from wallet

Unauthorized access to exchange server

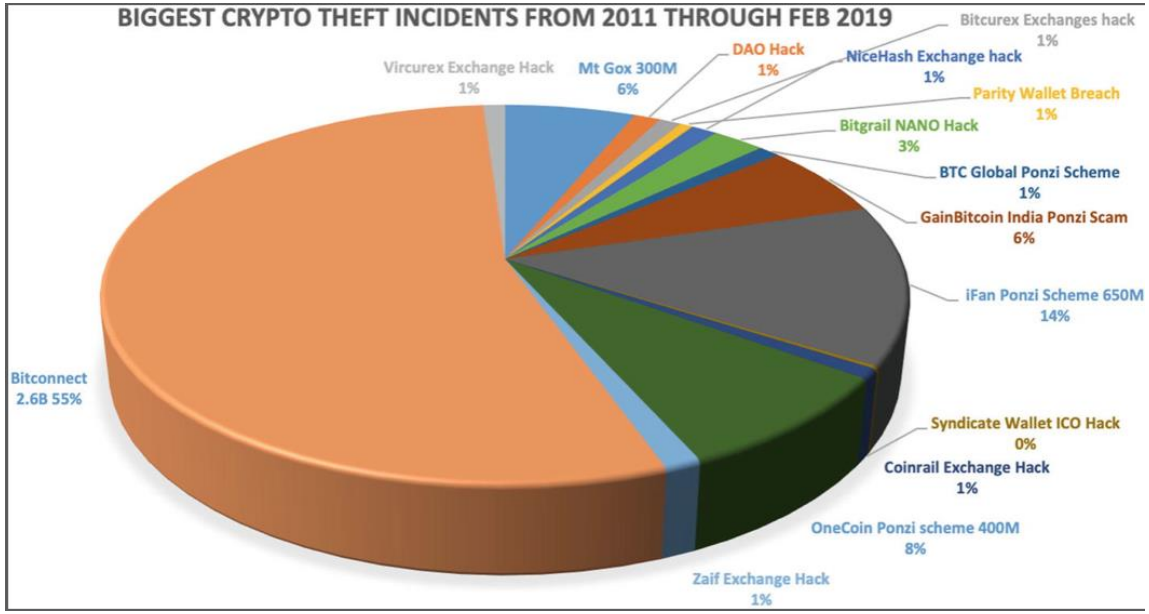
Unauthorized access to separate data storage with user name and password of user

Malicious Program Infection where digital asset control were lost

Capture of Initial Coin Offering via tampering with the collection address through attacks such as domain hijacking, web vulnerabilities, or social engineering, phishing attack

Mining machine attack via stealing of mining servers

# Crypto Theft incidents (2011 – 2019)




Excerpt From: Elad Elrom. “A Practical Guide for Designing, Implementing, Publishing, Testing, and Securing Distributed Blockchain-based Projects”. Apple Books.

# Denial of Services

Distributed denial of service (DDoS) attacks are not confined to Bitcoin or public Blockchain networks. In DDoS, the attacker sends a huge number of requests to overwhelm servers.

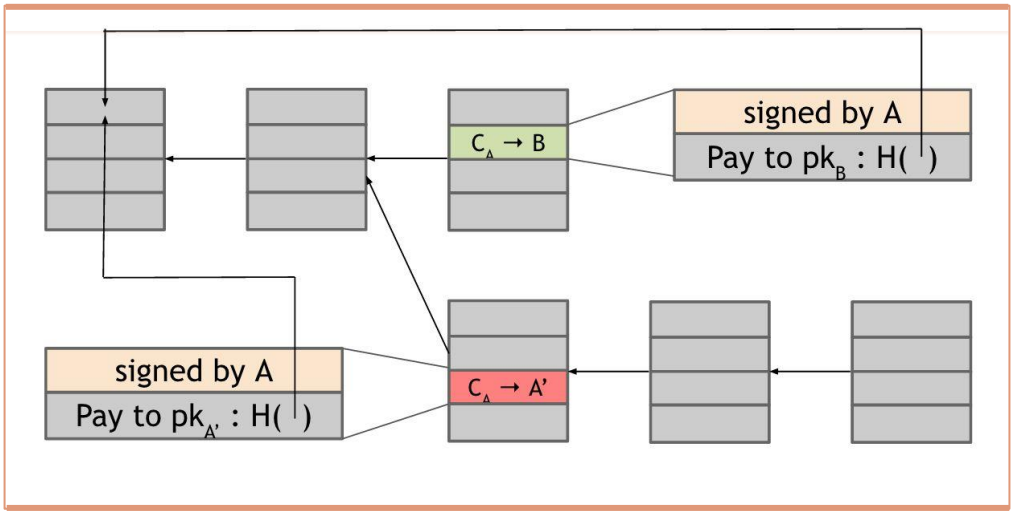
DDoS to Exchange platform. According to report of global DDoS threat landscape Q3 2017 by Incapsula, Bitcoin has become one of the top 10 industries which are most vulnerable to DDoS attacks

BGP hijacking. At present, the security researchers have proved the conceptual feasibility of the attack

A PoW system (or protocol, or function) is an economical measure to deter denial of service attacks and other service abuses such as spam on a network by requiring some work from the service requester, usually meaning processing time by a computer.

# Double Spending Attack

A double spend attempt. Alice creates two transactions: one in which she sends Bob Bitcoins, and a second in which she double spends those Bitcoins by sending them to a different address that she controls.




***51-percent attack.*** Finally, let's consider what would happen if consensus failed and there was in fact a ***51-percent attacker***who controls 51 percent or more of the mining power in the Bitcoin network. We'll consider a variety of possible attacks and see which ones can actually be carried out by such an attacker

# Compliance Issues related to DLT

Anti-money laundering and counter-financing of terrorism issues

Systemic risk

Technology and operational risks




Reporting and transparency

Governance and controls

Cybersecurity

Legal issues

# Risk considerations to DLT

**Operational Risks**

- Malicious validating nodes
- Network problems and attacks

**Identity Theft Risks**

- Identity theft and phishing cases

**Conduct Risks**

- Money laundering
- Sales of illegal drugs and contraband
- Receipt of ransom payments


**Regulatory compliance**

- Comply with industry specific regulations

# Risk considerations to Cryptocurrency

1. The entity chooses to use a cryptocurrency exchange that does not have effective controls over the transactions it enters into on behalf of the entity or over the balances of cryptocurrency maintained in the entity's accounts.

2. The entity has a cryptocurrency wallet that has not been accounted for.

3. The entity loses a private key and therefore can no longer access the related cryptocurrency.

4. An unauthorized party obtains access to the entity's private key and steals the entity's cryptocurrency.

5. The entity misrepresents ownership of a private key and therefore of the related cryptocurrency.

6. The entity sends cryptocurrency to an incorrect address and the cryptocurrency cannot be recovered.

7. The entity enters into and records a cryptocurrency transaction with a related party that cannot be identified because of the anonymity of parties to blockchain transactions.

8. There are significant delays in processing cryptocurrency transactions at the end of a period.

9. Events or conditions make it difficult to determine the value at which a cryptocurrency should be recorded for financial reporting purposes.

# Security Incident of Blockchain

# DeFi blockchain platform attack (2021 – 2022)

Chainalysis said at least $2.2 billion was outright stolen from DeFi protocols in 2021.

Poly Network saw $611 million stolen from their platform in August 2021

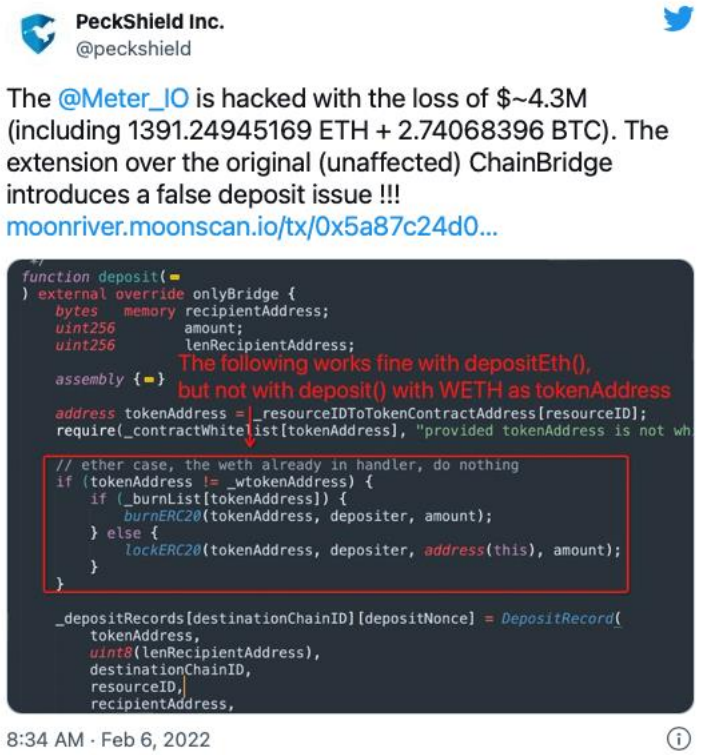
Bitmart lost $196 million in early December 2021

# Attack on blockchain infrastructure Meter (Feb 2022)

Blockchain infrastructure company Meter said $4.4 million was stolen during a cyberattack on the platform that started at around 9 am ET on Saturday morning.

The company said it manages an infrastructure that allows smart contracts to scale and travel through heterogeneous blockchain networks. The Meter network, as well as the Moonriver network, were affected by the hack.

By 6pm, Meter wrote that it stopped all bridge transactions and discovered that the issue related to a bug "introduced in the automatic wrap and wrap of native tokens like BNB and ETH extended by the Meter team."



# Attack on blockchain platform Wormhole (Feb 2022)

Wormhole, a popular blockchain bridge, confirmed on Wednesday evening that hackers stole crypto-assets worth $324 million.

The platform serves as a bridge between different blockchains and allows users to transfer cryptocurrency. The company confirmed in a series of Tweets that 120k wETH was stolen from the platform and the network was down for maintenance as they looked into a potential exploit.

# Crypto.com Hack (2022)

On 17 January 2022, Crypto.com learned that a small number of users had unauthorized crypto withdrawals on their accounts.

Crypto.com promptly suspended withdrawals for all tokens to initiate an investigation and worked around the clock to address the issue. No customers experienced a loss of funds. In the majority of cases we prevented the unauthorized withdrawal, and in all other cases customers were fully reimbursed.

The incident affected 483 Crypto.com users.

Unauthorised withdrawals totalled 4,836.26 ETH, 443.93 BTC and approximately US$66,200 in other cryptocurrencies.

On Monday, 17 January 2022 at approximately 12:46 AM UTC Crypto.com's risk monitoring systems detected unauthorized activity on a small number of user accounts where transactions were being approved without the 2FA authentication control being inputted by the user.

This triggered an  immediate response from multiple teams to assess the impact. All withdrawals on the platform were suspended for the duration of the investigation.

# Advise from Crypto Exchange Platform (JPEX in 2022)

According to the advise from JPEX

- Beware of fake customer support and phishing message
- Fake investment trip
- ICO scam
- Phishing email case


**【以下內容為特約贊助，僅供投資者參考】**

隨著世界慢慢邁向虛擬化，加密貨幣更受市場關注，使越來越多投資者接觸加密貨幣交易，也讓詐騙分子有機可乘。JPEX（JPEX Crypto Platform）在此文提醒各位新手投資者要如何在加密市場中預防詐騙並提高警覺，避免資金的損失。由於區塊鏈具有不可逆轉和匿名性的特質，因此一旦遇上任何形式的詐騙，你的資金都將無法追回！

**1.冒認客服、假官方人員**

詐騙手法：詐騙分子會潛伏在各大社交媒體（Whatapps、Facebook、Telegram 和 instagram等) 中，他們會主動私訊投資者說要幫用戶一對一解決問題或是把投資者拉到假冒的交易所官方群組（群組中存在著不同的共犯，互相聊天營造高熱度假象）。當投資者被拉進詐騙群組後，這些詐騙分子就會以活動名義吸引投資者打幣到某地址，之後再提供獎賞。等到投資者把加密貨幣發送到詐騙分子所指定的地址後便再也拿不回來。

應對方式：

大部份正規交易所其實都一再強調，只要是主動私訊用戶的都大機會是詐騙，或是官方人員絕對不會私訊等等。JPEX指，JPEX客服不會出現在任何的社群裡，只會出現在JPEX官網與JPEX App 中。

# 4 Rising Cyber Threats In 2019

## Taking advantage of AI-generated counterfeit audio and video

- Most of these “deepfakes” are known as a boon to a cyber-terrorist in several ways. Artificial Intelligence-generated “phishing” email messages that attempt to trick people into giving over security passwords and various other delicate information have already been proved to be more efficient than ones made by humans.

## Harming Artificial intelligence protection

- GANs or Generative adversarial networks, which promote a couple of sensory networks against each other, can easily be applied to try to know what formulas defenders are utilizing in their Artificial intelligence models.

2019

## Hacking Smart contracts

- it is still at the outset of the development, and experts are detecting bugs in a few of them. So are online hackers, who’ve taken advantage of imperfections to rob millions of dollars’ worth of digital currencies.

## Breaking shield of encryption using quantum computer systems

- use spectacular phenomena from quantum science to produce rapid leaps in digesting power, could crack encrypted shield that currently helps secure every little thing from health records to e-commerce transactions.

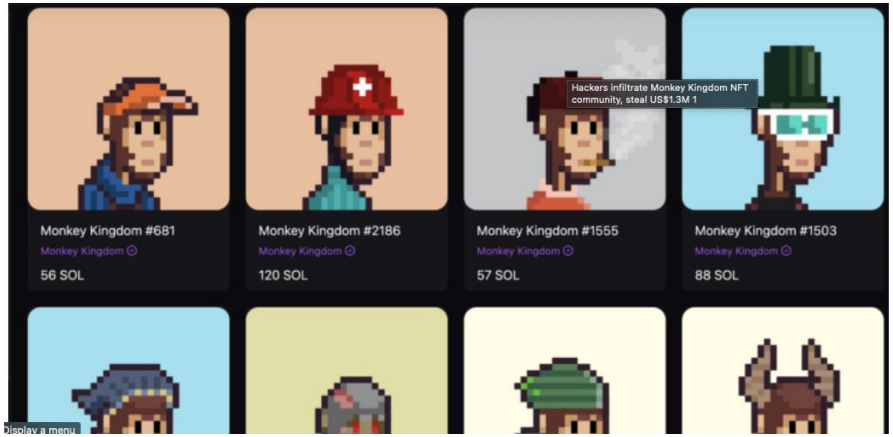
# Hot Hong Kong NFT project Monkey Kingdom loses US$1.3 million in hack (Jan 22)

**Hot Hong Kong NFT project Monkey Kingdom loses US$1.3 million in hack, exposing security concerns**

- A hacker stole an administrator account of the project's group chat on Discord, a popular online instant messaging service
- Monkey Kingdom fraud is the latest in a series of scams seen in the space in recent months as the popularity around NFT reaches fever pitch

**Hackers infiltrate Monkey Kingdom NFT community, steal US$1.3M**

**Hackers scam 7,000 Solana, or US$1.3 million, from would-be buyers of a new NFT collection of the Monkey Kingdom project.**




Popular non-fungible token (NFT) project Monkey Kingdom, founded by entrepreneurs in Hong Kong and promoted by celebrities such as JJ Lin and Steve Aoki, had its group chat hacked on Tuesday, allowing a cyber thief to steal nearly US$1.3 million worth of cryptocurrencies with a phishing link.

A hacker stole an administrator account of the project's group chat on Discord, a popular online instant messaging service, and posted a phishing link in the group chat on Tuesday, just as the project kicked off a new sale in earnest. Buyers lost more than 7,000 Solana, a popular cryptocurrency, to the scam, which amounts to nearly US$1.3 million.

Phishing is a common form of online fraud often used to steal user data, including login credentials and credit card numbers. It occurs when an attacker, masquerading as a trusted entity, dupes a victim into opening an email, instant message, or text message. It is now being used to breach access to users' cryptocurrency wallets.

# Scammers steal $150K worth of crypto from NFT project with Discord hack (Dec 21)

*The attacker posted a fraudulent message from the project's official channel*

By Corin Faife | @corintxt | Dec 21, 2021, 4:23pm EST

SHARE

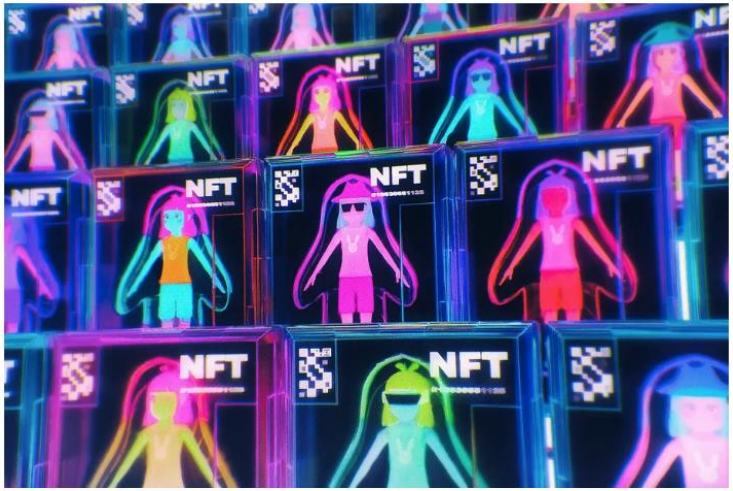



Illustration by Alex Castro / The Verge

Buyers hoping to get a limited-edition NFT from Fractal, a new marketplace for game item NFTs, were given an unpleasant and costly surprise on Tuesday morning when it was revealed that a link sent through the project's official Discord channel was a scam set up to steal crypto.

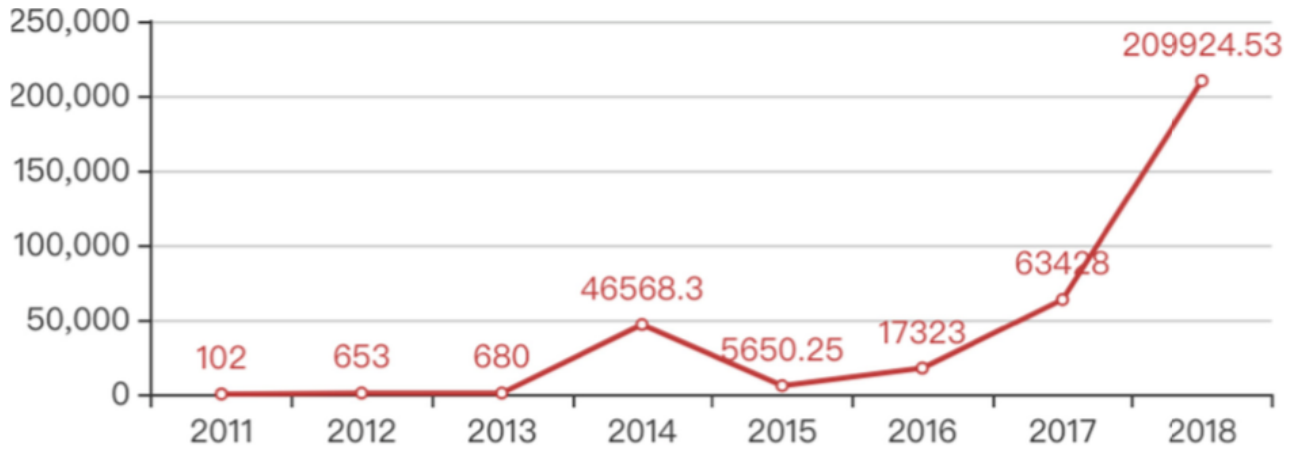
Users who followed the link and connected their crypto wallets, expecting to receive an NFT, instead found that their holdings of Solana (SOL) cryptocurrency were emptied and transferred to the scammer's account. An analysis posted on Medium by Tim Cotten, founder of another NFT gaming project, estimated the value of SOL stolen to be around $150,000.

Fractal is a startup project from Twitch co-founder Justin Kan specializing in the buying and selling of NFTs representing in-game assets. It was announced earlier in December and quickly amassed a following of more than 100,000 users through Discord — making it a target for the kind of scammers that have plagued NFT projects since the beginning.

News reached Twitter when a tweet from Kan informed followers that the announcements bot on Fractal's Discord server had been hacked. Another tweet from the main Fractal Twitter account confirmed that a fraudulent link had been posted through the channel.

https://www.theverge.com/2021/12/21/22848840/scammers-steal-crypto-nft-project-fractal-discord-hack-solana

# Economic losses caused by Blockchain Security Incidents (2019)



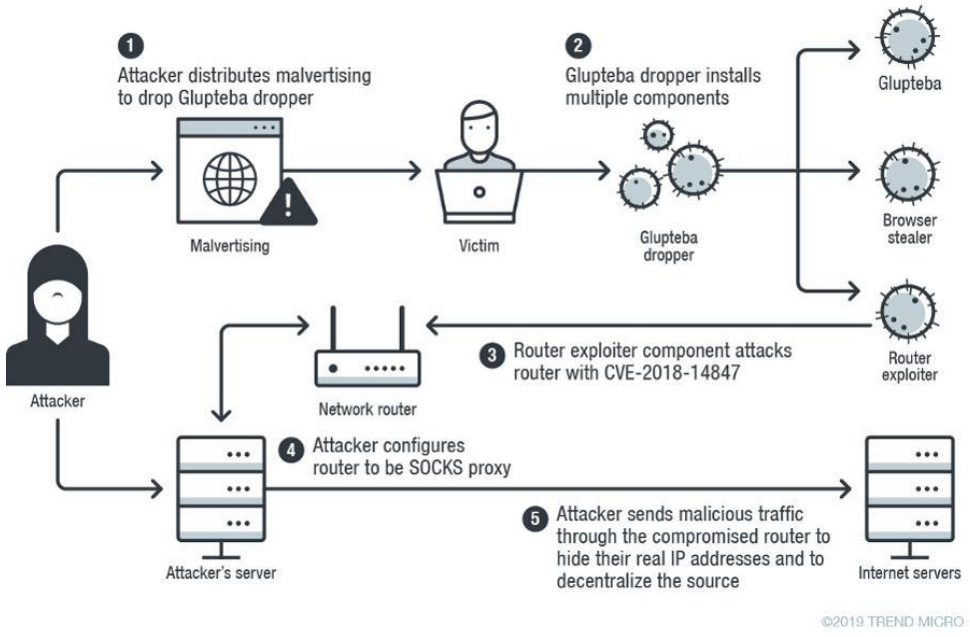

Economic losses caused by blockchain security incidents (ten thousand dollars) according to statistics from BCSEC

# Warning Issued After Malware Is Found To Have Hijacked Bitcoin Blockchain (Sep 2019)

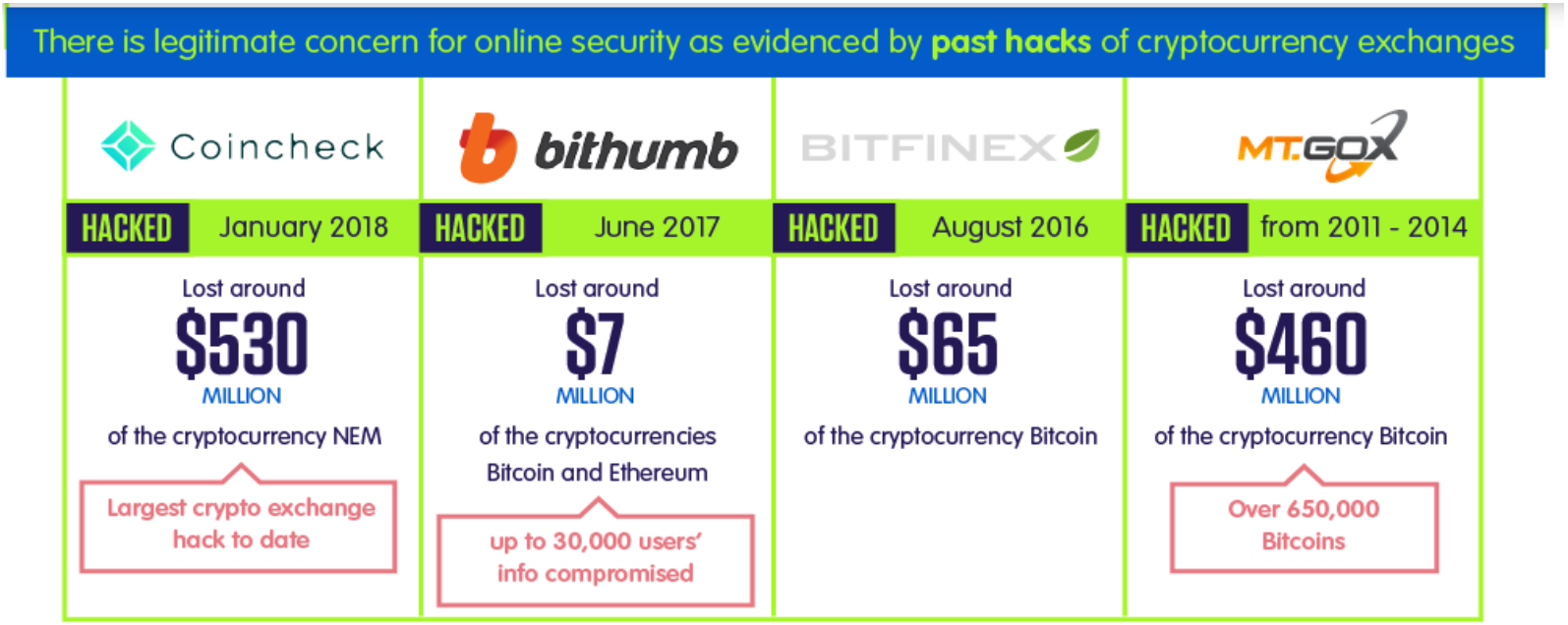
The malware uses the bitcoin blockchain to update, meaning it can continue running even if a device's antivirus software blocks its connection to servers run by the hackers.

The malware uses the Electrum bitcoin wallet to send bitcoin transactions that the attackers use to gain access to systems.

The attacker simply need to add a new bitcoin script and the infected machines obtain a new command and control server by decrypting the script data and reconnecting



# Cryptocurrency Hacks



# Status of Digital Currencies in Hong Kong (Jan - Oct 2018)

## Hong Kong won't ban digital currencies but will educate public on risks via campaign

Yet as an open economy, city won't mean tell investors what to buy and not buy

PUBLISHED : Monday, 29 January, 2018, 8:22pm
UPDATED : Tuesday, 06 February, 2018, 10:51pm

COMMENTS: 3


## Hong Kong needs digital currency regulation, because someone needs to look out for investors

PUBLISHED : Wednesday, 24 October, 2018, 5:01pm
UPDATED : Wednesday, 24 October, 2018, 6:31pm

# 苦主稱誤信回報三代無憂 火燎森被指呃人挖礦 (5 Nov 2018)


有12名事主向民主黨求助，涉及金額達320萬。

民主黨接獲12宗涉及虛擬貨幣陷阱求助，涉及金額由1萬元至過60萬元不等，總金額逾320萬元。苦主大多指控受火燎森誤導，於今年5月透過投資公司「天機科技」或於網上購入HE15雲端挖礦合約，不過，簽約半年，事主獲得的虛擬貨幣卻與火燎森或天機科技描述有極大出入，多名事主均「損手收場」。

# Attack Types and Threats

According to Top 10 Blockchain Attacks Vulnerabilities from Cloud Security Alliance

- Exchange Hack
- DeFi Hack
- Rugpull / Exitscam
- Investment Scam
- High-Profile Doubler Scam
- 51% Attack
- Phishing
- Extortion
- Ransomware
- SIM Swap

Nature of Threats of Blockchain attacks

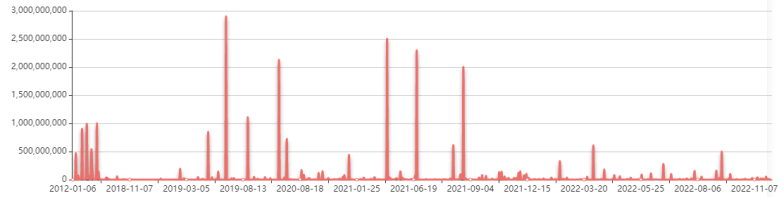
- Stealing Cryptocurrencies
- Double-spending
- Denial of Services
- Other Compliance Issues and Risk Considerations

# Attack Types and Threats (2023)

**[SlowMist Hacked Statistical]:**

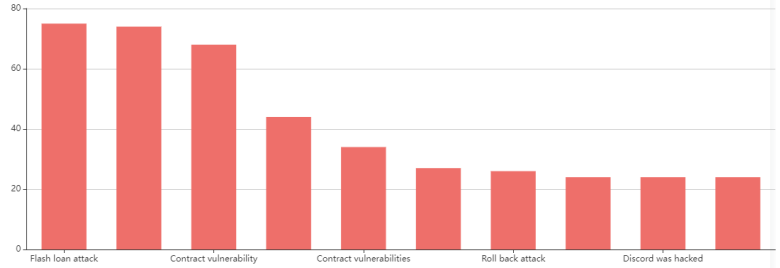
Total hack event(s) 899 ;

The total amount of money lost by blockchain hackers is about $ 28,330,033,325.52 ;




| Category | Hack event(s) | Amount of loss ($) |
|---|---|---|
| Blockchain | 48 | 206,505,567.00 |
| Exchange | 109 | 10,289,201,175.39 |
| Wallet | 31 | 299,741,253.59 |
| ETH Ecosystem | 169 | 2,845,979,975.56 |
| BSC Ecosystem | 124 | 1,422,271,935.55 |
| Tron Ecosystem | 23 | 11,224,334.36 |
| EOS Ecosystem | 119 | 25,927,302.55 |
| Polygon Ecosystem | 11 | 55,185,949.00 |
| HECO Ecosystem | 3 | 8,064,533.00 |
| Fantom Ecosystem | 11 | 86,592,814.00 |
| Solana Ecosystem | 13 | 202,741,994.22 |
| Avalanche Ecosystem | 7 | 118,673,000.00 |
| Polkadot Ecosystem | 9 | 61,545,641.00 |
| NFT | 69 | 190,055,307.00 |
| Bridge | 29 | 1,867,656,543.30 |
| Other | 124 | 10,638,666,000.00 |

**The 10 most common attacks**




| Number | Attack method | Hack event(s) |
|---|---|---|
| 1 | Flash loan attack | 75 |
| 2 | Rug Pull | 74 |
| 3 | Contract vulnerability | 68 |
| 4 | Scam | 44 |
| 5 | Contract vulnerabilities | 34 |
| 6 | Transaction congestion attack | 27 |
| 7 | Roll back attack | 26 |
| 8 | Phishing attack | 24 |
| 9 | Discord was hacked | 24 |
| 10 | 51% attack | 24 |

# Security attack methods on blockchain

- Network-Level attacks
- System-Level attacks
- Smart Contract attacks

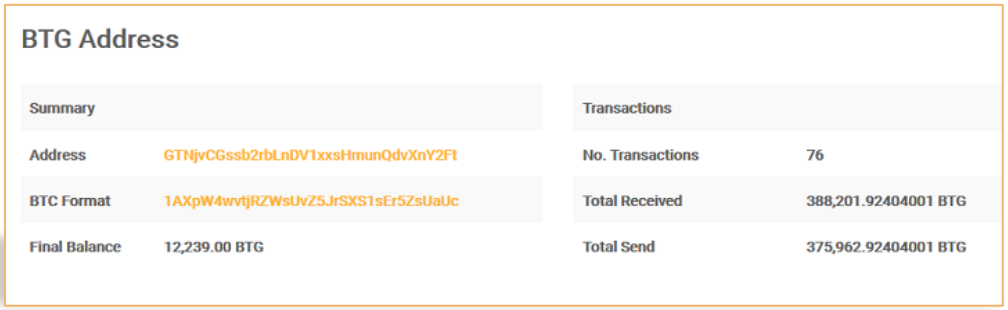
# Network Level Attacks

| | | |
|---|---|---|
| Denial of Service Attack | Replay Attack | Routing Attack |
| Eclipse Attack | 51% Attack | Sybil Attack |

# 51 Percent Attack: Hackers steals $18 Million in Bitcoin Gold (BTG) Tokens

An unknown party with access to very large amounts of hashpower is trying to use '51% attacks' to perform 'double-spend' attacks to steal money from Exchanges. We have been advising all exchanges to increase confirmations and carefully review large deposits.

The image is the Bitcoin Gold address of the suspected hacker. The attacker sends a particular number of BTG tokens to an exchange, trades them for another coin and makes a withdrawal.




Thus, the attacker can spend and hold the same coins at the same time. Looking at the image above, if all 76 transactions were indeed part of the hack, then the hacker has stolen about $18 million based on the current BTG price.

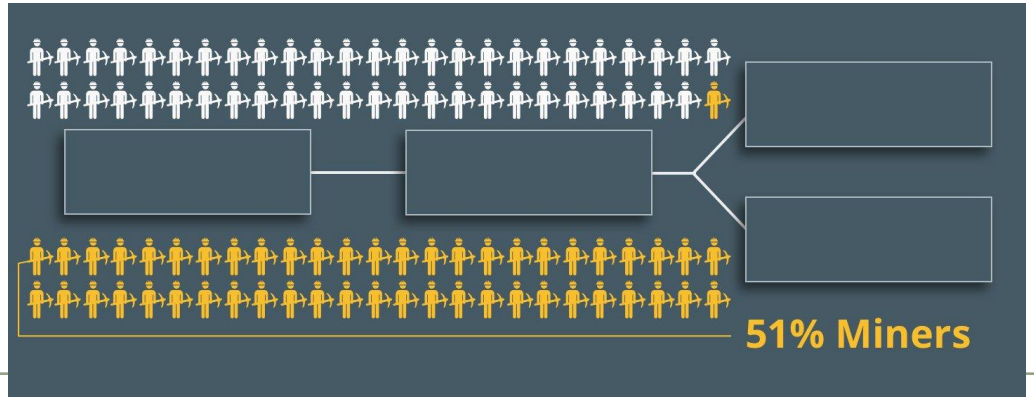
# Flaws in the PoW consensus

A 51 percent attack, or majority attack, is a case when a user or a group of users control the majority of mining power.

The attackers get enough power to control most events in the network.

- Fastest to compute the difficult problem (hash power) can generate the hash in the blockchain
- Longest chain will be kept = majority

They can monopolize generating new blocks and receive rewards since they're able to prevent other miners from completing blocks. (NB: although, it should be noted, that even less than 50% of the hashing power still has a good chance of performing such attacks).



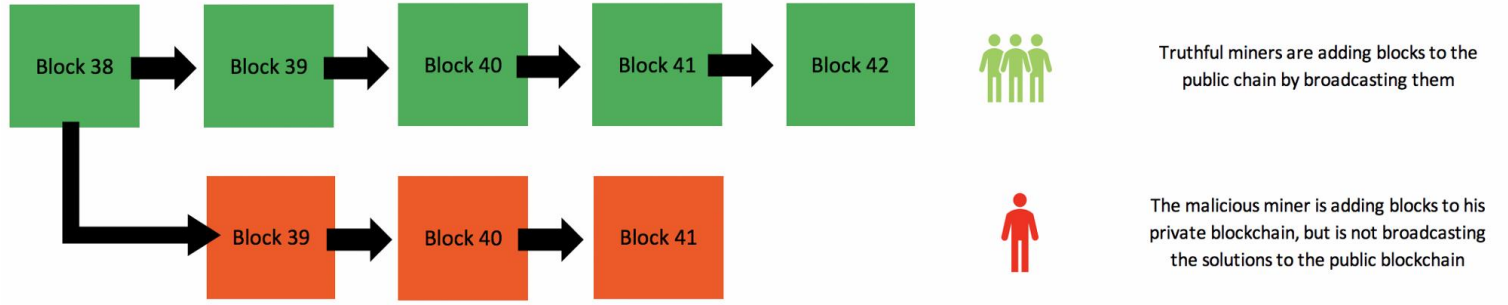
# Security Problems: 51% attacks (more than half)

- Due to the design, in public blockchain:
- Hackers gain **51%** hashing power can gain the control of whole blockchain
- **Vulnerable to 51% attacks**

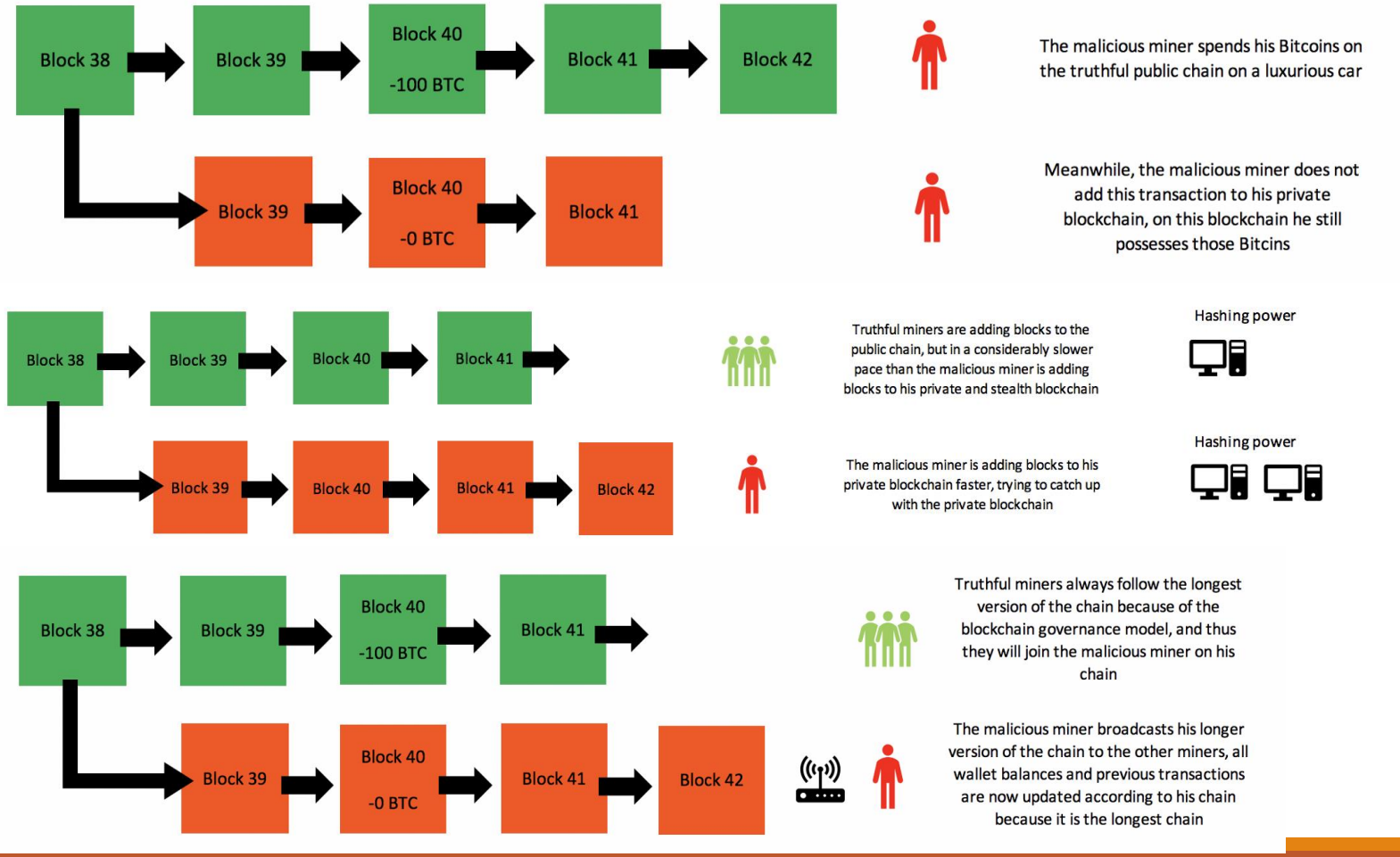
**Double-spend** | **Prevent creation of blocks** | **Stop the transactions**



- Higher risk of 51% attacks in smaller blockchains



# Security Problems: 51% attacks (more than half)



# Security Problems: 51% attacks (more than half)

| Attacker can perform | Attacker cannot perform |
| --- | --- |
| • Reverse transactions that he sends while he's in control<br>• Prevent some or all transactions from gaining any confirmations<br>• Prevent some or all other generators from getting any generations | • Reverse other people's transactions<br>• Prevent transactions from being sent at all (they'll show as 0/unconfirmed)<br>• Change the number of coins generated per block<br>• Create coins out of thin air<br>• Send coins that never belonged to him |

# Selfish Mining attack

A Selfish Mining attack helps an attacker accomplish a 51% attack with less than 51% of the blockchain's resources.

In an ideal world, users would immediately publish the next block once they found a solution to the Proof of Work puzzle.

A selfish miner will hide their solution for a while and immediately begin mining the next block on top of the one that they found.

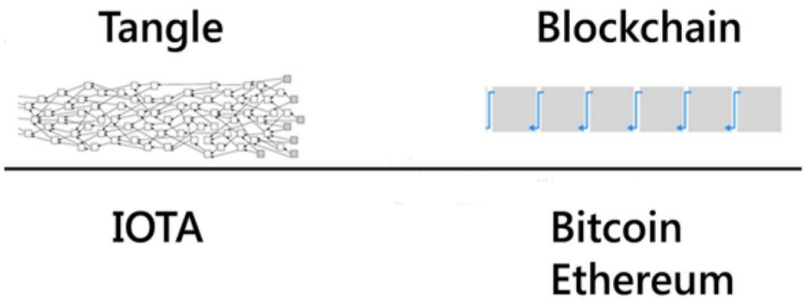
Since other miners can't begin mining until a solution is revealed, this gives them a head start on the race to find a Proof of Work solution for the next block.

If an attacker can cut the power or remove a miner's access to the Internet, they increase their chances of performing a 51% attack, especially if the miner(s) represent a large percentage of the network's computational power.

# Security Problems: 34% attacks (more than 1/3)

In IOTA (cryptocurrency):

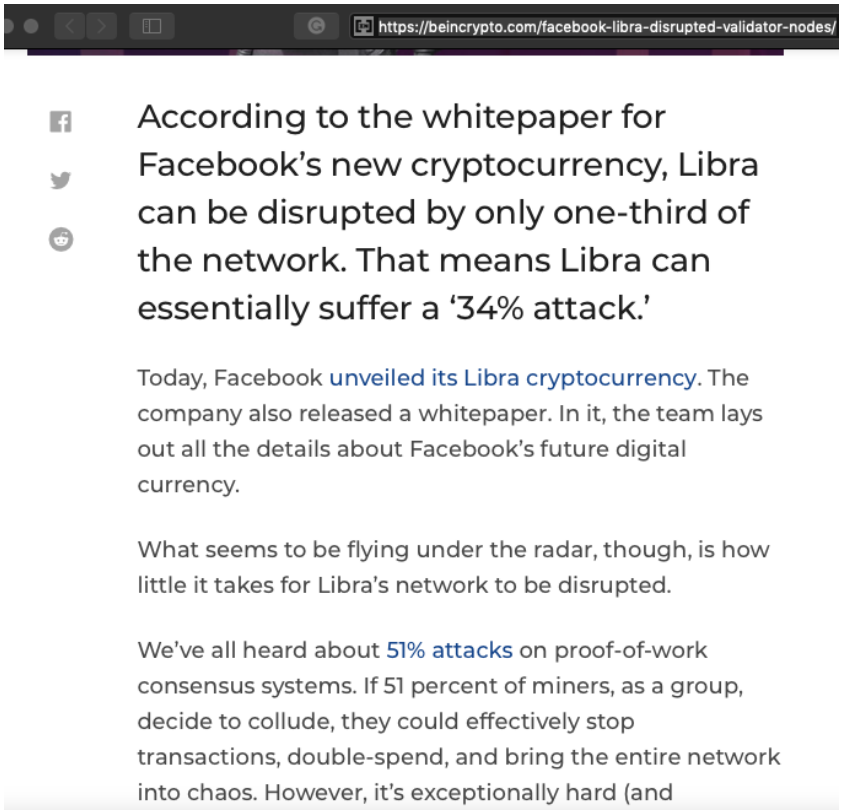
Use Tangle (Directed Acyclic Graph)




Attack reason: Consensus of BFT (require more than 33%)

Solution: Coordinator

Downside:
Coordinators are centralized
Making the blockchain partially centralized




https://beincrypto.com/facebook-libra-disrupted-validator-nodes/

# Consensus algorithms comparison

| Property | PoW | PoS | PBFT | DPOS | Ripple | Tendermint |
|---|---|---|---|---|---|---|
| Node identity management | Open | Open | Permissioned | Open | Open | Permissioned |
| Energy saving | No | Partial | Yes | Partial | Yes | Yes |
| Tolerated power of adversary | <25% computing power | < 51% stake | < 33.3% faulty replicas | < 51% validators | < 20% faulty nodes in UNL | < 33.3% byzantine voting power |
| Example | Bitcoin, Ethereum | Peercoin | Hyperledger Fabric | Bitshares | Ripple | Tendermint |

https://www.smartcontractresearch.org/t/a-brief-introduction-to-blockchain-consensus-algorithms/88

# Sybil Attack

Sybil is synonymous with someone who has a multiple personality disorder.

Sybil Attack is a type of attack seen in peer-to-peer networks in which a node in the network operates multiple identities actively at the same time and undermines the authority/power in reputation systems.

Actually, a Sybil attack is when an attacker creates a large number of accounts on the blockchain network. Create and operate a large number of blockchain accounts

The main aim of this attack is to gain the majority of influence in the network to carry out illegal (with respect to rules and laws set in the network) actions in the system.

A single entity (a computer) has the capability to create and operate multiple identities(user accounts, IP address based accounts).

To outside observers, these multiple fake identities appear to be real unique identities.

A type of sybil attack, called the 51% attack is also practically impossible in the bitcoin network because of so many miners, it is very difficult for a single organization to control 51% of the miners.

Majority Attack = 51% Attack = Sybil Attack = Double-spend Attack

# Eclipse Attack

The attack aims to obscure a participant's view of the peer-to-peer network, in order to cause general disruption, or to prepare for more sophisticated attacks.

Attacker can then controls the nodes' connections to the network.

Successful Eclipse attack allows the attacker to perform

- double-spending attack against the isolated node
- Denial of Service attack
- weakening competing miners

The concept is discussed at length in the 2015 paper Eclipse Attacks on Bitcoin's Peer-to-Peer Network

# Eclipse Attack

How to perform Eclipse Attack

- **Location**. Attacker needs to intercept user's message before user can reach the rest of the blockchain network
- **Power**. Attacker needs to have the power of ISP or similar, so they can control the communications
- **Malware**. May need to use malware to infect other users

How to prevent or minimize the probability of attack

- Increase connections.
- Whitelist to restrict proper connections
- Random Reconnections to reduce the time to detect eclipse attack
- Use of permissioned private blockchain.

# Technical Attacks

## REPLAY ATTACK

Transactions on blockchain can be resent and replayed

In a replay attack, an attacker can take an existing transaction and resubmits it to the blockchain like it is a new transaction.

To protect against this attack, a blockchain should implement nonces, or unique value in each transaction.

## ROUTING ATTACK

Focus on attacking the underlying communications network used by the blockchain for peer-to-peer communications

Attacker may be able to isolate network into multiple segments. That could be used for supporting the

- Denial of Service attacks
- 51% attack
- Double spending attack

Can be prevented using

- Multi-home nodes
- Intelligent neighbor selection algorithm
- Known route selection
- Encrypted Authenticated communications

# Technical Attacks – Denial of Service attacks

In a Denial of Service attack, an attacker attempts to degrade a service's operations or make it completely non-functional.

Example of DoS attack

- **Transaction flooding**. Flood the network with transactions to increase the queue for transaction
- **Difficulty increases**. Temporarily increase the computational power of Proof of Work blockchain to push up the difficulty then remove the increased resources
- **Block forger DoS**. Prevent the block from being added to the chain by preventing the communication of next block creator to the blockchain network (for Proof of Stake)
- **DoS of Permissioned blockchain MSP**. Deny users access to the blockchain

# Technical Attacks – Denial of Service attacks

Mitigation solutions

- Intentionally create blocks to clear flooded transactions from queue
- Set difficulty increase interval to minimize attack impact
- Implement traditional Anti-DoS and DDoS protection for nodes

# Network Attack Summary

| Attack Name | Purpose of attack | Facilitate Attack | Target of Attack | Defense Mechanism |
|---|---|---|---|---|
| 51% Attacks | Take control of the blockchain | Double-spending | Blockchain nodes | Checkpointing |
| Denial of Service Attack | Introduce service failure, degrade services, take down services, flooding, forger DoS, MSP DoS | | Attack at bottlenecks of the system<br>Attack at MSP | Anti-DDoS at Single Point of Failure |
| Eclipse Attack | Controls of all blockchain nodes and isolated nodes | Double-spending<br>Consensus hijacking<br>Denial of Service Attack | Blockchain nodes, isolated nodes | Increase Connections<br>Whitelists<br>Random Reconnections<br>Permissioned, Private blockchain |
| Replay Attacks | Replay of the existing transaction and resubmit it to the blockchain | | Replay of transaction | Add nonces or unique value in transaction |
| Routing Attack | Routing attacks target underlying communications network, useful in performing 51% attacks, DoS attack and double-spending attacks | 51% attack<br>Double-spending<br>Denial of Service Attack | Target underlying communication network | Multi-homed Nodes<br>Intelligent Neighbor Section<br>Known Route Selection<br>Network Statistics Monitoring<br>Encrypted Authenticated Communications |
| Sybil Attack | Dominate the network by managing more nodes (through Botnet, Virtualization technology, Blockchain-focused malware) | Eclipse Attack<br>Routing Attack<br>Attack at Proof-of-Stake | Create and operate more blockchain accounts | Use Permissioned or Private blockchain |

# System Level Attacks

System Vulnerabilities and Exploits

- Overflow or underflow vulnerability (e.g. Bitcoin Hack, ERC20 Smart Contract)
- Cryptographic signature vulnerability
- Address Space Layout Randomization (ASLR) bypass vulnerability

Flaws in programming

- Complexity - The Verge Hack
- Program logic issues - The Lisk Vulnerability
- Transaction mutability vulnerability

# Bitcoin Vulnerabilities (2010)

Jeff Garzik first spotted and commented the “quite strange” transaction outputs, which stood at 92233720368.54 BTC apiece.

Bitcoin had already endured at least four major bugs or vulnerabilities prior to the integer overflow bug that led to 184 billion BTC being created out of thin air. Previously, around 40 bugs were identified.

CVE-2010-5139, however, was unlike anything the Bitcoin community had ever encountered in 2010 — or has seen since. An integer overflow had caused a negative total transaction value.

An unknown attacker had discovered the bug and used it to generate a ridiculously high number of bitcoins. It is possible their exploit might have lain undiscovered for longer than the 90 minutes it took for the scheme to be spotted. The 184 billion BTC transaction was purged from block 74638.


There was something unusual about Bitcoin block 74638.

A new version of the client was published within five hours of the discovery that contained a soft forking change to the consensus rules that rejected output value overflow transactions (as well as any transaction that paid more than 21 million bitcoins in an output for any reason)

# Bitcoin Vulnerabilities

Bitcoin overflow (CVE-2010-5139)

- Sum of output values overflowed the integer variable
- Soft fork de facto invalidated the transaction

Author | Topic: Version 0.3.10 - block 74638 overflow PATCH! (Read 5059 times)

satoshi
Founder
Sr. Member

Activity: 364

Ignore

**Version 0.3.10 - block 74638 overflow PATCH!**
August 15, 2010, 11:48:22 PM

quote #1

Version 0.3.10 patches the block 74638 overflow bug. http://bitcointalk.org/index.php?topic=823

The Linux version includes tcatm's 4-way SSE2 SHA-256 that makes generating faster on i5, i7 (with hyperthreading) and AMD CPU's. Try the "-4way" switch to enable it and check if it's faster for you.

Download from sourceforge:
http://sourceforge.net/projects/bitcoin/files/Bitcoin/bitcoin-0.3.10/

SHA1 16645ec5fcdb35bc54bc7195309a1a81105242bb bitcoin-0.3.10-win32-setup.exe
SHA1 4f35ad7711a38fe8c880c6c9beab430824c426d3 bitcoin-0.3.10-win32.zip
SHA1 e3fda1ddb31b0d5c35156cacd80dee6ea6ae6423 bitcoin-0.3.10-linux.tar.gz
SHA1 b812ccff4881778b9090f7c0b0255bcba7b078ac bitcoin-0.3.10-macosx.zip

It is no longer necessary to delete blk*.dat. The good block chain has overtaken the bad block chain, so you can just upgrade and it'll automatically reorg away the bad block chain.

# Bitcoin Vulnerabilities (2021)

| CVE | Announced | Affects | Severity | Attack is... | Flaw | Net |
|---|---|---|---|---|---|---|
| Pre-BIP protocol changes | n/a | All Bitcoin clients | Netsplit[1] | Implicit[2] | Various hardforks and softforks | 100% |
| CVE-2010-5137 | 2010-07-28 | wxBitcoin and bitcoind | DoS[3] | Easy | OP_LSHIFT crash | 100% |
| CVE-2010-5141 | 2010-07-28 | wxBitcoin and bitcoind | Theft[4] | Easy | OP_RETURN could be used to spend any output. | 100% |
| CVE-2010-5138 | 2010-07-29 | wxBitcoin and bitcoind | DoS[3] | Easy | Unlimited SigOp DoS | 100% |
| **CVE-2010-5139** | 2010-08-15 | wxBitcoin and bitcoind | Inflation[5] | Easy | Combined output overflow | 100% |
| CVE-2010-5140 | 2010-09-29 | wxBitcoin and bitcoind | DoS[3] | Easy | Never confirming transactions | 100% |
| CVE-2011-4447 | 2011-11-11 | wxBitcoin and bitcoind | Exposure[6] | Hard | Wallet non-encryption | 100% |
| CVE-2012-1909 | 2012-03-07 | Bitcoin protocol and all clients | Netsplit[1] | Very hard | Transaction overwriting | 100% |
| CVE-2012-1910 | 2012-03-17 | bitcoind & Bitcoin-Qt for Windows | Unknown[7] | Hard | Non-thread safe MingW exceptions | 100% |
| BIP 0016 | 2012-04-01 | All Bitcoin clients | Fake Conf[8] | Miners[9] | Softfork: P2SH | 100% |
| CVE-2012-2459 | 2012-05-14 | bitcoind and Bitcoin-Qt | Netsplit[1] | Easy | Block hash collision (via merkle root) | 100% |
| **CVE-2012-3789** | 2012-06-20 | bitcoind and Bitcoin-Qt | DoS[3] | Easy | (Lack of) orphan txn resource limits | 100% |
| CVE-2012-4682 | | bitcoind and Bitcoin-Qt | DoS[3] | | | 100% |
| **CVE-2012-4683** | 2012-08-23 | bitcoind and Bitcoin-Qt | DoS[3] | Easy | Targeted DoS by CPU exhaustion using alerts | 100% |
| **CVE-2012-4684** | 2012-08-24 | bitcoind and Bitcoin-Qt | DoS[3] | Easy | Network-wide DoS using malleable signatures in alerts | 100% |
| CVE-2013-2272 | 2013-01-11 | bitcoind and Bitcoin-Qt | Exposure[6] | Easy | Remote discovery of node's wallet addresses | 99.99% |
| CVE-2013-2273 | 2013-01-30 | bitcoind and Bitcoin-Qt | Exposure[6] | Easy | Predictable change output | 99.99% |
| CVE-2013-2292 | 2013-01-30 | bitcoind and Bitcoin-Qt | DoS[3] | Hard | A transaction that takes at least 3 minutes to verify | 0% |
| **CVE-2013-2293** | 2013-02-14 | bitcoind and Bitcoin-Qt | DoS[3] | Easy | Continuous hard disk seek | 99.99% |
| CVE-2013-3219 | 2013-03-11 | bitcoind and Bitcoin-Qt 0.8.0 | Fake Conf[8] | Miners[9] | Unenforced block protocol rule | 100% |
| CVE-2013-3220 | 2013-03-11 | bitcoind and Bitcoin-Qt | Netsplit[1] | Hard | Inconsistent BDB lock limit interactions | 99.99% |
| BIP 0034 | 2013-03-25 | All Bitcoin clients | Fake Conf[8] | Miners[9] | Softfork: Height in coinbase | 100% |
| BIP 0050 | 2013-05-15 | All Bitcoin clients | Netsplit[1] | Implicit[2] | Hard fork to remove txid limit protocol rule | 99.99% |

| | | | | | | |
|---|---|---|---|---|---|---|
| CVE-2020-14198 | | Bitcoin Core 0.20.0 | DoS[3] | Easy | Remote DoS | 93% |
| CVE-2021-3401 | 2021-02-01 | Bitcoin Core GUI prior to 0.19.0<br>Bitcoin Knots GUI prior to 0.18.1 | Theft | Hard | Qt5 remote execution | 64% |

# BatchOverflow Bug in Multiple ERC20 Smart Contracts (CVE-2018-10299)

System raised an alarm which is related to an unusual BEC token transaction.

In this particular transaction, someone transferred an extremely large amount of BEC token — 0x8000,0000,0000,0000,0000,0000,0000,0000,0000,0000,0000,0000,0000,0000,0000,0000(63 0's

That batchOverflow is essentially a classic integer overflow issue.

With amount zeroed, an attacker can then pass the sanity checks in lines 258–259 and make the subtraction in line 261 irrelevant.

```
255   function batchTransfer(address[] _receivers, uint256 _value) public whenNotPaused returns (bool) {
256     uint cnt = _receivers.length;
257     uint256 amount = uint256(cnt) * _value;
258     require(cnt > 0 && cnt <= 20);
259     require(_value > 0 && balances[msg.sender] >= amount);
260
261     balances[msg.sender] = balances[msg.sender].sub(amount);
262     for (uint i = 0; i < cnt; i++) {
263         balances[_receivers[i]] = balances[_receivers[i]].add(_value);
264         Transfer(msg.sender, _receivers[i], _value);
265     }
266     return true;
267   }
268 }
```

# ECDSA Vulnerability

There exists a well known ECDSA signature vulnerability

- (also present in the non-elliptic curve signature scheme of ElGamal and its popular variant, DSA)

by which an attacker that observes two signatures of different messages made with the same private key is able to extract the private key if the signer reuses the same k selected

given two ECDSA signatures that have been created using the same k and the same private key,

- sig1(m1) = (r,s1) and sig2(m2) = (r,s2) with m1≠ m2,

an attacker that obtains m1, sig1 and m2, sig2 may derive the private key d

Some Bitcoin wallets adopted deterministic ECDSA after this vulnerability was found to affect some Bitcoin transactions.

# Address Space Layout Randomization (ASLR) bypass vulnerability

EOS (Open-source Smart Contract Platform) software needs to be able to parse and execute smart contract files.

A buffer out of bounds writing vulnerability in the parsing function allowed malicious smart contracts to exploit the EOS blockchain software.

Researchers at Qihoo 360 identified and demonstrated that they could bypass Address Space Layout Randomization (ASLR) and get a remote shell from the attacked node.

Vulnerable smart contract platforms can allow a malicious smart contract to exploit the node running the blockchain software.

# Complexity – The Verge Hack

Verge cryptocurrency was hacked through combination of built-in features weaknesses

- Flexible timestamps – lack of a synchronized time server on the blockchain
- Difficulty updates – uses Dark Gravity Wave algorithm
- Consensus algorithm complexity – supports 5 algorithms

By falsifying the timestamp of every other block to an hour in the past, the attacker made it appear that blocks were being created at a much slower rate than expected (hourly instead of every thirty seconds).

The Dark Gravity Wave algorithm dropped the difficulty of the mining algorithm. The mining became ridiculously easy, allowing the attacker to perform a 51% attack with less than 10% of the network's computational resources.

# Program Logic Issues – Lisk Vulnerability

Lisk is a cryptocurrency. However, it is vulnerable to an exploit taking advantage of two built-in features. Lisk is a kind of Ethereum where contracts are written in JavaScript—instead of Solidity or Viper—and where the consensus protocol relies on proof-of-stake instead of proof-of-work.

Use last 64 bits of the SHA-256 hash of user's public key as the address of blockchain.

However, it can be attacked because transactions sending value to an account do not tie an address to a public key.

The complexity of finding a public key that can be tied to a given address is at most $2^{64}$ operations (which is doable) and "failures" may allow an attacker to claim a different address

After Lisk developed fixed this vulnerability, another vulnerability arises. The system is vulnerable to race conditions.

# Program Logic Issues - Vulnerability in IOTA

Serious flaws the hash algorithm adopted by IOTA was discovered in July 2017 by an academic organization of Massachusetts Institute of Technology, these flaws threatened the security of digital signature and PoW algorithm of IOTA.

- one-time hash-based signatures, using key = f(seed, index), with incremented index.
- Instead of using a standard hash function, IOTA initially used **a custom hash** (curl).
- For which **collisions** were easy to find (i.e. M1 and M2 such that H(M1)==H(M2))

# Transaction Mutability Vulnerability

In March 2014, criminals exploited a transaction mutability vulnerability in Bitcoin to attack MtGox, causing its collapse with approximately $450M in Bitcoin stolen.

In earlier implementations because it was in 2014, there's no standard for appending data. In other words, when they create a hash, what they do is they would try to put the content of the transaction together with a padding string at the end, a 01, a random 01 at the end in order to create a hash.

Now, because this hash is the transaction ID, so if the attacker is able to change the padding string, then the ID will change and then what they are trying to do is they claim that the transaction is being lost.

In fact, the transaction has been transferred into their account but they changed the padding string so the ID changes so the owner cannot trace the transaction where it has been gone.

# Transaction Mutability Vulnerability

Japan-based bitcoin exchange Mt. Gox (now-bankrupt) may have lost only 386 bitcoins ($203,000) due to issues stemming from transaction malleability.

Released on 26th March, the report was authored by Christian Decker and Professor Roger Wattenhofer, both of the university's Distributed Computing Group (DCG).

- They identified potential double spending attacks and the limitations they faced in doing so.
- To trace and dump all transactions from the Bitcoin network, the researchers created specialized nodes, allowing them to detect any double-spending attacks observed by peer nodes.
- While double spending attacks could be determined by associating transactions with the outputs they claim, researchers chose to remove signature script from the transactions, and looked instead at the unique keys produced by the malleability attacks.
- indicates that approximately 29,139 conflict sets were identified over the course of the research and later confirmed by the block chain.

# Transaction Mutability Vulnerability

The defect is known as “transaction malleability” and it allows third parties to alter the hash of a fresh transaction without invalidating the signature.

Since the transaction appears as if it has not proceeded correctly, the bitcoins may be re-sent.

This means that an individual could request bitcoins from an exchange or wallet service, alter the resulting transaction’s hash before inclusion in the blockchain, then contact the issuing service while claiming the transaction did not proceed. If the alteration fails, the user can simply send the bitcoins back and try again until successful.

# Wallet Vulnerability

Bitcoin Core is a cryptocurrency wallet designed for use with the Bitcoin cryptocurrency

Potential vulnerabilities in the code can lead to double-spending attack

Through the transaction that cause the DoS of the node, then some nodes can cause double-spending attack

Bitcoin Core developers fixed the issue for Bitcoin, but some other fork of Bitcoin core code have to fix the vulnerability too

Pigeoncoin failed to patch the vulnerability in time which leads to exploit of coin.

# Miner Vulnerability

Simplified Payment Verification (SPV) nodes are “lighter” nodes that only download the headers of blocks

SPV nodes needs to get “head start” on other

With this SPV mining, this enables

- Increase the mining rewards
- Can lead to perform 51% attack with a reduced portion of blockchain’s resources

# System Vulnerability Summary

| Attack Name | Purpose of attack | Target of Attack | Defense Mechanism |
|---|---|---|---|
| Bitcoin Hack | Integer overflow vulnerability | An attacker exploited this vulnerability by creating a transaction that sent 184 billion Bitcoin to an account controlled by an attacker. | Program code review, white-box security assessment |
| Verge Hack | Timestamp issue and Consensus algorithm program, complexity | Perform 51% attack with less than 10% of network's computational resources | Decrease timestamp windows from 2 hrs to 15 minutes and |
| EOS Vulnerability | Address Space Layout Randomization (ASLR) vulnerability in program code | Able to execute malicious smart contract to exploit the node running the blockchain software | Program code review |
| Lisk Vulnerability | Attack of shortened public key and then race conditions found | Able to claim different address of the keys | Ties to address Race conditions |

# … has risks: bugs in smart contract code

← Thread

vitalik.eth
@VitalikButerin

The "centralized anything is evil by default, use defi and self-custody" ethos did very well this week, but remember that it too has risks: bugs in smart contract code.

Important to guard against it:

* Keep code simple
* Audits, formal verification, etc
* Defense in depth

10:04 PM · Nov 16, 2022

# Smart Contract Attack

**Unauthorized access attack**

**Smart contract Development Security**

**Bad contract logic**

**Contract cannot be changed after deployed**

# Ethereum Classic Tokens Stolen and Returned, Proof-of-Work Algorithm Manipulated

A cyber criminal has reportedly returned $100,000 worth Ethereum Classic tokens to cryptocurrency exchange Gate.io . If that seems generous to you, it's just around 10 percent of the loot that is still with the cyber attackers who stole over 200,000 Ethereum Classic Tokens, which roughly sums up to around $ 1.1 million.

Trouble was reportedly noticed by the Exchange around 5th Jan 2019, and noteworthy changes were identified in the Blockchain around January 8th, 2019. By January 10th, 2019, the 51% and double spend attack was confirmed to have been used. To fix this, Gate.io has for now raised the bar for Ethereum Classic transactions.

# What is DAO

**The (Decentralized Autonomous Organization) DAO** was a complex dApp that programmed a decentralized venture capital fund to run on Ethereum.

Holders of The DAO would be able to vote on what projects they wanted to support, and if developers raised enough funding from The DAO holders, they would receive the funds necessary to build their projects.

The DAO was borne from an immutable, unstoppable, and irrefutable computer code, operated entirely by its members, and fueled using ETH, the token that represents value on the Ethereum blockchain, which creates DAO tokens.

The DAO leverages smart contracts on the Ethereum blockchain so that anyone, anywhere in the world can be empowered to participate. In exchange for their early help, participants receive DAO tokens

# What is DAO

Original conception of the DAO

- "In the beginning, we created a slock.it specific smart contract and gave token holders voting power about what we— slock.it— should do with the funds received.
- After further consideration, we gave token holders even more power, by giving them full control over the funds, which would be released only after a successful vote on detailed proposals backed by smart contracts. This was already a few steps beyond the Kickstarter model, but we would have been the only recipient of funds in this narrow slock.it-specific DAO.
- We wanted to go even further and create a "true" DAO, one that would be the only and direct recipient of the funds, and would represent the creation of an organization similar to a company, with potentially thousands of Founders"

The core of this automaton code base that would quickly raise $150 million—were relatively simple requests for The DAO's resources

The DAO backs proposals, which it selects for their innovative nature, to be delivered by "Contractors". Some of these Proposals could hold no promise of return whatsoever (in the case of a charity for example), others could involve the building of products or services which The DAO could then use for its own purposes.

# DAO Incident

The DAO was launched on 30 April 2016 at 01:42:58 AM +UTC on Ethereum Block 1428757 with a website and a 28-day crowdsale to fund the organization

This exploit occurred due to a vulnerability in the pattern of coding the Smart Contract. It was known and it was being fixed by its creators, but while they were doing it, the hacker prepared the exploit in order to drain all the funds from the DAO

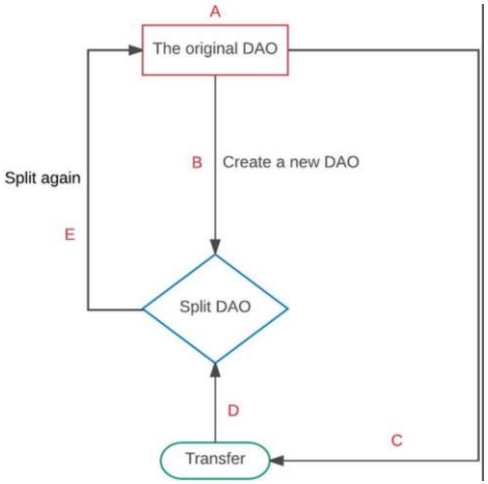
A few days back from Jun 2016, 3,641,694 ETH was split from "theDAO".

The "Attacker" found a loophole in the regular "splitDAO" function so that they could reuse the same DAO tokens over and over again.

Essentially, the "Attacker" found a legal "loophole" in the contract code that allowed him/her to exploit the contract unilaterally.

Someone could recursively split from the DAO, withdrawing amounts equal to their original ETH investment indefinitely, before the record of their withdrawal was ever recorded in the original DAO contract.

Due to flaws in how the smart contract was constructed, an attacker extracted Ether, the cryptocurrency used by Ethereum, resulting in the theft of $50 million




Excerpt From: Vikram Dhillon, David Metcalf and Max Hooper. "Blockchain Enabled Applications". Apple Books.

# DAO vulnerability

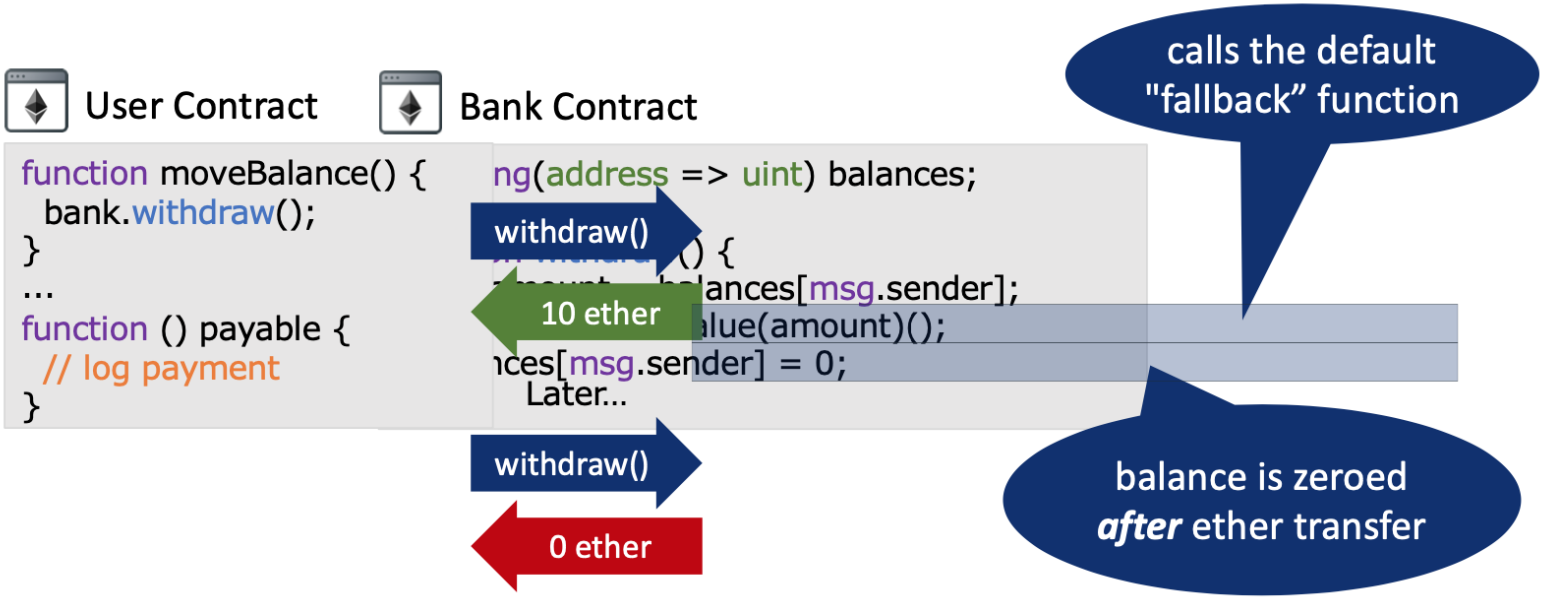
## (i) the contract Bank (vulnerable contract)

```
contract Bank{
/*Contract that stores user balances. This is the vulnerable contract. This contract co
the basic actions necessary to interact with its users, such as: get balance, add to ba
and withdraw balance */

   mapping(address=>uint) userBalances;/*mapping is a variable
   type that saves the relation between the user and the amount contributed to
   this contract. An address (account) is a unique indentifier in the blockchain*/

   function getUserBalance(address user) constant returns(uint) {
     return userBalances[user];
   }/*This function returns the amount (balance) that the user has contributed
   to this contract (this information is saved in the userBalances variable)*/

   function addToBalance() {
     userBalances[msg.sender] = userBalances[msg.sender] + msg.value;
   }/*This function assigns the value sent by the user to the userBalances variable.
   The msg variable is a global variable*/

   function withdrawBalance() {
     uint amountToWithdraw = userBalances[msg.sender];
     if (msg.sender.call.value(amountToWithdraw)() == false) {
         throw;
     }
     userBalances[msg.sender] = 0;
   }
}/*First, this function gets the user's balance and sets it to the amountToWithdraw
variable. Then, the function sends the user the amount set in the
amountToWithdraw variable. If the transaction is successful the userBalances is
set to 0 because all the funds deposited in the balance are sent to the user.
```

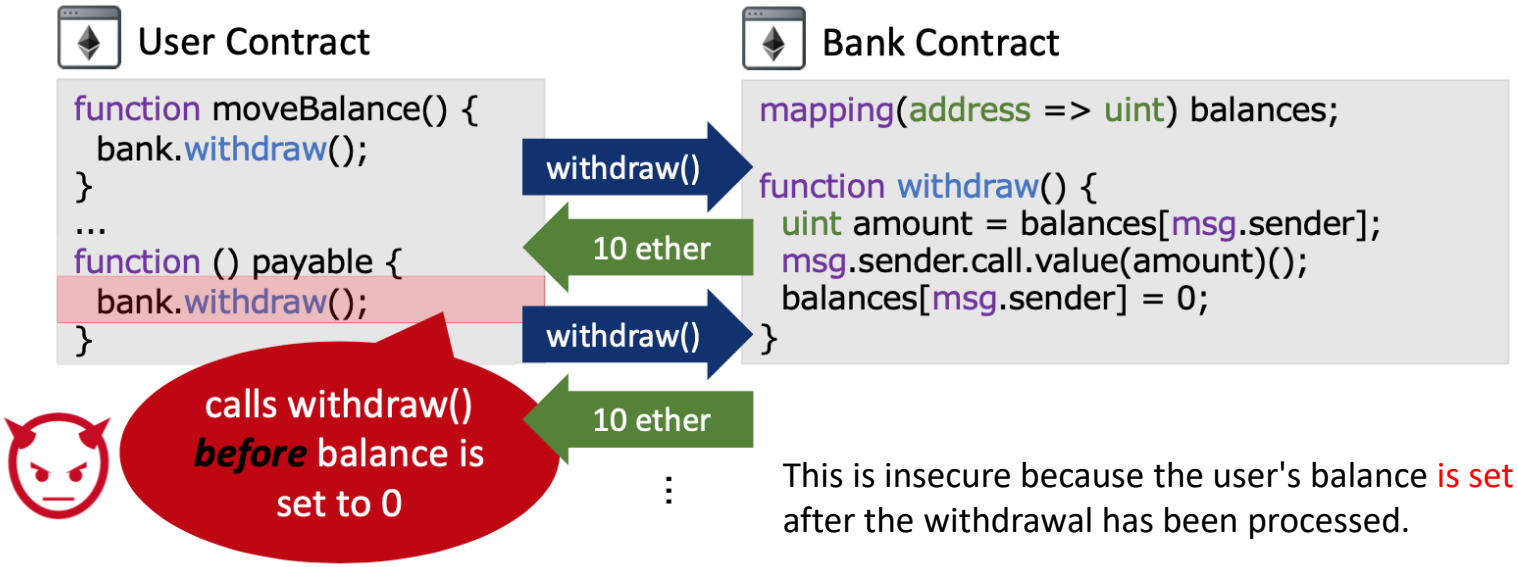
## ii) the contract BankAttacker (malicious contract).

```
contract BankAttacker{
/*This is the malicious contract that implements a double spend attack to the
first contract: contract Bank. This attack can be carried out n times.
For this example, we carried it out only 2 times.*/

   bool is_attack;
   address bankAddress;

   function  BankAttacker(address _bankAddress, bool _is_attack){
       bankAddress=_bankAddress;
       is_attack=_is_attack;
   }/*This function, which is the constructor, sets the address of the contract to be att
   (contract Bank) and enables/disables the double spend attack */

   function() {

       if(is_attack==true)
       {
           is_attack=false;
           if(bankAddress.call(bytes4(sha3("withdrawBalance()")))) {
               throw;
           }
       }
   }/* This is the fallback function that calls the withdrawnBalance function
   when attack flag, previuosly set in the constructor, is enabled. This function
   is triggered because in the withdrawBalance function of the contract Bank a
   send was executed. To avoid infinitive recursive fallbacks, it is necessary
   to set the variable is_attack to false. Otherwise, the gas would run out, the
   throw would execute and the attack would fail */

   function  deposit(){

        if(bankAddress.call.value(2).gas(20764)(bytes4(sha3("addToBalance()")))
        ==false) {
              throw;
         }

   }/*This function makes a deposit in the contract Bank (75 wei) calling the
   addToBalance function of the contract Bank*/

   function  withdraw(){

        if(bankAddress.call(bytes4(sha3("withdrawBalance()")))==false ) {
              throw;
         }

   }/*This function triggers the withdrawBalance function in the contract Bank*/
}
```

# DAO Incident



# DAO Incident




This is insecure because the user's balance is set to 0 only after the withdrawal has been processed.

What is wrong in the code is the fact that the developers did not take into consideration the possibility of a recursive call and the fact that the contract first sent the ETH funds and then updated the internal balance.

# Another Ethereum DApp Hack – Parity Wallet Hack

```
// constructor - just pass on the owner array to the multiowned and  // the limit
function initWallet(address[] _owners, uint _required, uint _daylimit) {
  initDaylimit(_daylimit);
  initMultiowned(_owners, _required);
}
```

The attacker sent two transactions to each of the affected contracts: the first to obtain exclusive ownership of the MultiSig, and the second to move all of its funds.

This function was probably created as a way to extract the wallet's constructor logic into a separate library. This uses a similar idea to the proxy libraries pattern we talked about in the past.

This causes all public functions from the library to be callable by anyone, including initWallet, which can change the contract's owners. Unfortunately, initWallet has no checks to prevent an attacker from calling it after the contract was initialized.

# Another Ethereum DApp Hack – Parity Wallet Hack

The second largest attack to an Ethereum dApp, was the one called the Parity Wallet Hack.

In order to steal 150,000 Ether (ETH), the attacker sent two transactions to each contract with vulnerabilities.

With the first transaction, the hacker gained ownership over the wallet, and with the second he transfers all the victim's funds.

The attacker sent two transactions to each of the affected contracts: the first to obtain exclusive ownership of the MultiSig, and the second to move all of its funds.

```
function() payable {
  // just being sent some cash?
  if (msg.value > 0)
    Deposit(msg.sender, msg.value);
  else if (msg.data.length > 0)
    _walletLibrary.delegatecall(msg.data);
}
```

The wallet contract forwards all unmatched function calls to the library using delegatecall, in line 424 of Wallet

This causes all public functions from the library to be callable by anyone, including initWallet, which can change the contract's owners.

Unfortunately, initWallet has no checks to prevent an attacker from calling it after the contract was initialized.

The recommended pattern is explicitly defining which library functions can be invoked externally on the wallet contract.

More: https://medium.com/coinmonks/the-phenomena-of-smart-contract-honeypots-755c1f943f7b

# Types of vulnerabilities in Smart Contract

## Greedy

- This kind of Smart Contract is capable of lock the funds from their users and then holding them back indefinitely without release them.
- The errors more common in Greedy Smart Contracts are that they accept Ether but they lack the set of instructions that sends the Ether out.

## Prodigal

- The Smart Contracts that can send Ether to an address that is not its owner address, nor one of the address that have sent Ether to the contract before and nor an address of someone that has given some kind of solution to it, are called Prodigal Smart Contracts.

## Suicidal

- In order to let its owner close and kill a Smart Contract under critic circumstances like an attack or because a deprecated function it is often implemented suicidal functions in the Smart Contract.
- When an arbitrary account is capable of invoking these functions, the Smart Contract has a vulnerability and is called a Suicidal Smart Contract.

# BatchOverflow Bug in Multiple ERC20 Smart Contracts (CVE-2018-10299)

System raised an alarm which is related to an unusual BEC token transaction.

In this particular transaction, someone transferred an extremely large amount of BEC token — 0x8000,0000,0000,0000,0000,0000,0000,0000,0000,0000,0000,0000,0000,0000,0000,0000(63 0's

That batchOverflow is essentially a classic integer overflow issue.

With amount zeroed, an attacker can then pass the sanity checks in lines 258–259 and make the subtraction in line 261 irrelevant.

```
255   function batchTransfer(address[] _receivers, uint256 _value) public whenNotPaused returns (bool) {
256     uint cnt = _receivers.length;
257     uint256 amount = uint256(cnt) * _value;
258     require(cnt > 0 && cnt <= 20);
259     require(_value > 0 && balances[msg.sender] >= amount);
260
261     balances[msg.sender] = balances[msg.sender].sub(amount);
262     for (uint i = 0; i < cnt; i++) {
263         balances[_receivers[i]] = balances[_receivers[i]].add(_value);
264         Transfer(msg.sender, _receivers[i], _value);
265     }
266     return true;
267   }
268 }
```

# BatchOverflow Bug in Multiple ERC20 Smart Contracts (CVE-2018-10299)

They advertised it as a legitimate pyramid scheme, and surprisingly its value quickly grew to over a million dollars, and over a thousand Ethereum.

A few hours ago, 866 Ethereum vanished from the contract, due to a flaw in the code.

Vulnerability

- in the part of PoWH's implementation of ERC-20 that allows a person to "approve" another user to transfer tokens on their behalf. The resulting (unsigned) integer underflow would leave the second account with an extremely large balance of PoWH Coins.

# The Parity Wallet Hack Explained (OpenZeppelin)

```
// constructor - just pass on the owner array to the multiowned and  // t
function initWallet(address[] _owners, uint _required, uint _daylimit) {
  initDaylimit(_daylimit);
  initMultiowned(_owners, _required);
}
```

Proxy Library Pattern

```
function() payable {
  // just being sent some cash?
  if (msg.value > 0)
    Deposit(msg.sender, msg.value);
  else if (msg.data.length > 0)
_walletLibrary.delegatecall(msg.data);
}
```

# honeypot1: MultiplicatorX3

```
1  pragma solidity ^0.4.18;
2
3  contract MultiplicatorX3
4  {
5      address public Owner = msg.sender;
6
7      function() public payable{}
8
9      function withdraw()
10     payable
11     public
12     {
13         require(msg.sender == Owner);
14         Owner.transfer(this.balance);
15     }
16
17     function Command(address adr,bytes data)
18     payable
19     public
20     {
21         require(msg.sender == Owner);
22         adr.call.value(msg.value)(data);
23     }
24
25     function multiplicate(address adr)
26     public
27     payable
28     {
29         if(msg.value>=this.balance)
30         {
31             adr.transfer(this.balance+msg.value);
32         }
33     }
34 }
```

At first glance it looks like by transferring more than the current balance of the contract it is possible to withdraw the full balance.

Both statements in line 29 and 31 try to reinforce the idea that this.balance is somehow credited after the function is finished.

This is a trap since

- the this.balance is updated before the multiplicate() function is called and
- so if(msg.value>=this.balance) is never true unless this.balance is initially zero.

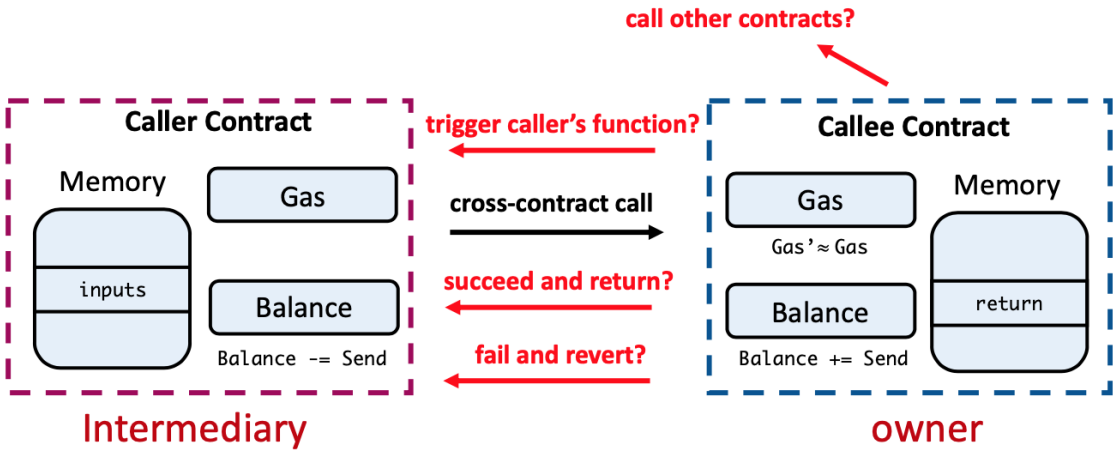
# Payment Statement of Ethereum transactions

```
contract Intermediary {
  uint256 public fee;
  address public seller;
  address public owner;

  function Intermediary() {
    owner = msg.sender;
// seller initialization is omitted
    fee = 10;
  }
```

A sample smart contract.

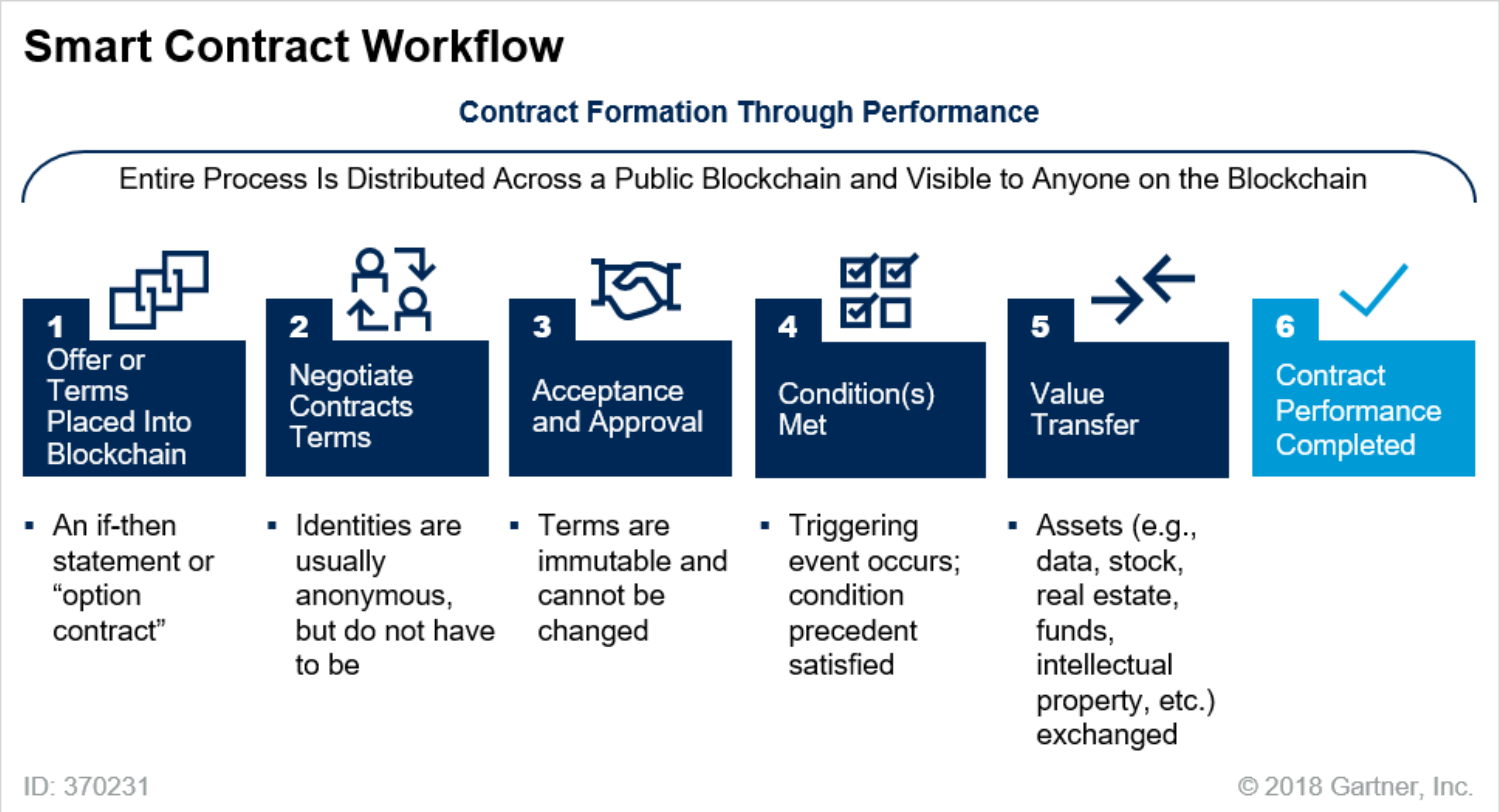
```
  function purchase() {
// msg.value is how much Ether was sent by user
// transfer pays (msg.value-fee) to the seller
    owner.transfer(msg.value - fee);
  }
  function setFe    256 _fee) {
    if (msg.sender    ner)
      fee = _fee;
  }
}
```

A sample smart contract (cont'd).




Everything starts from External Contract Call

# Smart Contract Workflow



# Attacks made to Smart Contract

Reentrancy attack

Arithmetic attack (Integer overflow and underflow)

Unchecked Return Value

Replay attack

Reordering attack

Short address attack

Access Control attack

Race condition attack

Timestamp Dependence

Transaction-Ordering Dependency

Denial of Service

Others

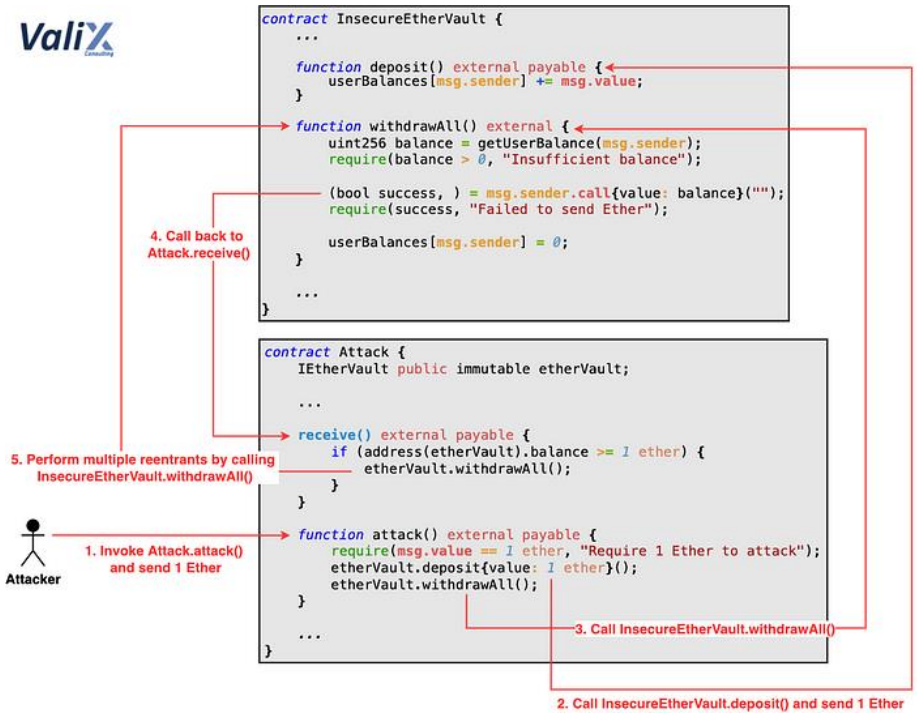
# Attacks made to the contract

Reentrancy attack

- *This attack consists on recursively calling the* call.value() *method in an ERC20 token to extract the ether stored on the contract if the user is not updating the* balance *of the sender before sending the ether*
- this contract uses transfer() instead of call.value() , there's no risk of reentrancy attacks since the transfer function only allows to use 23.000 gas which you can only use for an event to log data and throws on failure.
- Note that the condition to call this function is that the number of bets is bigger or equal the limit of 10 bets but that condition isn't updated until the end of the distributePrizes() function which is risky because someone could theoretically be able to call that function and execute all the logic before updating the state.

# Attacks made to the contract – Reentrancy attack

```solidity
pragma solidity 0.8.13;

contract InsecureEtherVault {
    mapping (address => uint256) private userBalances;

    function deposit() external payable {
        userBalances[msg.sender] += msg.value;
    }

    function withdrawAll() external {
        uint256 balance = getUserBalance(msg.sender);
        require(balance > 0, "Insufficient balance");

        (bool success, ) = msg.sender.call{value: balance}("");
        require(success, "Failed to send Ether");

        userBalances[msg.sender] = 0;
    }

    function getBalance() external view returns (uint256) {
        return address(this).balance;
    }

    function getUserBalance(address _user) public view returns (uint256) {
        return userBalances[_user];
    }
}
```



# Attacks made to the contract – Reentrancy attack


**Solutions**

- Applying the checks-effects-interactions pattern
- Applying the mutex lock
- Using both solutions #1 and #2

# Attacks made to the contract

Arithmetic attack (Over and under flow)

- *An overflow happens when the limit of the type variable* uint256 *, 2**256, is exceeded. What happens is that the value resets to zero instead of incrementing more*
- On the other hand, an underflow happens when you try to subtract 0 minus a number bigger than 0.
- For example, if you subtract 0 -1 the result will be = 2**256 instead of -1.
- Recommend using a library like the OpenZeppelin's SafeMath.sol.

```
// INSECURE
function transfer(address _to, uint256 _value) {
    /* Check if sender has balance */
    require(balanceOf[msg.sender] >= _value);
    /* Add and subtract new balances */
    balanceOf[msg.sender] -= _value;
    balanceOf[_to] += _value;
}
```

```
// SECURE
function transfer(address _to, uint256 _value) {
    /* Check if sender has balance and for overflows */
    require(balanceOf[msg.sender] >= _value && balanceOf[_to] + _value >= balanceOf[_to]);

    /* Add and subtract new balances */
    balanceOf[msg.sender] -= _value;
    balanceOf[_to] += _value;
}
```

# Attacks made to the contract

Unchecked Return Value attack

- Functions return values were sent back to the main program without proper initialization of the value.
- Return code could be different from the expected return
- “King of the Ether” got an unchecked call to send with only a small amount of gas allocated to it

# Attacks made to the contract

Replay attack

- The replay attack consists on making a transaction on one blockchain like the original Ethereum's blockchain and then repeating it on another blockchain like the Ethereum's classic blockchain.
- its no longer a problem because since the version 1.5.3 of Geth and 1.4.4 of Parity both implement the attack protection EIP 155 by Vitalik Buterin: https://github.com/ethereum/EIPs/blob/master/EIPS/eip-155.md

# Attacks made to the contract

Reordering attack

- This attack consists in that a miner or other party tries to “race” with a smart contract participant by inserting their own information into a list or mapping so the attacker may be lucky in getting their own information stored on the contract.
- When a user places his bet() and the data is saved on the blockchain, anybody will be able to see what number has been bet by simply calling the public mapping playerBetsNumber

# Attacks made to the contract

Short address attack

- This attack affects ERC20 tokens, was discovered by the Golem team and consists of the following:
- A user creates an ethereum wallet with a trailing 0, which is not hard because it's only a digit. For instance: 0xiofa8d97756as7df5sd8f75g8675ds8gsdg0
- Then he buys tokens by removing the last zero:
  - Buy 1000 tokens from account 0xiofa8d97756as7df5sd8f75g8675ds8gsdg
  - If the token contract has enough amount of tokens and the buy function doesn't check the length of the address of the sender, the Ethereum's virtual machine will just add zeroes to the transaction until the address is complete.
  - The virtual machine will return 256,000 for each 1000 tokens bought. This is a bug of the virtual machine that's yet not fixed so whenever you want to buy tokens make sure to check the length of the address.

# Attacks made to the contract

## Access Control attack

- Improperly implemented access control
- No prior checking of the state before function call
- Ownership can be transferred

```
contract OwnableWallet {
  address owner;

  // called by the constructor
  function initWallet(address _owner) {
    owner = _owner; // any user can change owner
    // more setup
  }

  // function that allows the owner to withdraw ether
  function withdraw(uint _amount) {
    if (msg.sender == owner) {
      owner.transfer(_amount);
    }
  }
  // ...
}
```

Ownership can be transferred to anyone

# Attacks made to the contract

## Race conditions Attack

- Risk of calling an external function is that the calling behavior may cause the control flow to be hijacked and accidentally modify the contract data
- Example: Bancor smart contract exchange hack, https://www.apriorit.com/dev-blog/554-bancor-exchange-hack

```
contract Intermediary {
  uint256 public fee;
  address public seller;
  address public owner;

  function Intermediary() {
    owner = msg.sender;
// seller initialization is omitted
    fee = 10;
  }
```

A sample smart contract.

```
  function purchase() {
// msg.value is how much Ether was sent by user
// transfer pays (msg.value-fee) to the seller
    owner.transfer(msg.value - fee);
  }
  function setFee(uint256 _fee) {
      fee = _fee;
  }
}
```

A sample smart contract (cont'd).

# Transaction-Ordering Dependency

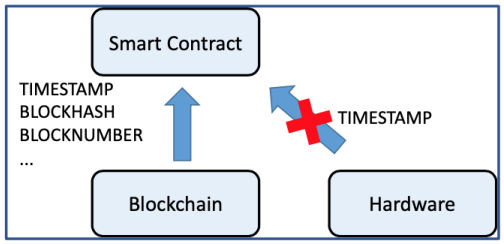
## Transaction-Ordering Dependency

- construct transaction based on the order information contained in the pending transactions, and try to get his transaction to be written into the block before others

```
contract RandomReward {
  uint256 constant private salt = block.timestamp;
  uint256 constant private threshold = 1000;

  function buggy_reward(uint256 bet) public {
      uint256 t = salt * block.timestamp/(salt % 5) ;
      if (t > threshold)
        msg.sender.send.value(bet * 100)().
  }
}
```

Compute random number

- Easy for a unified design and keep consensus across different physical machine.
- But not real random!
  - Another source of nondeterminism since, for instance, system properties can be manipulated by (malicious) miners.




External call to non-determinism and and dependency on other functions

# Timestamp Dependence Attack

## TRANSACTION ORDERING DEPENDENCE (TOD) ATTACK

```
contract TransactionOrdering {
 uint256 price; address owner;

 function estimate(uint256 amount) {
   cost = price * amount
   return cost;
 }

 function setPrice(uint256 _price) {
   // owner can set the price.
   if (msg.sender == owner)
     price = _price;
 }

 function purchase(uint256 money) {
   return money / price;
 }
}
```

## BLOCK STATE DEPENDENCE ATTACK

```
contract RandomReward {
  uint256 constant private salt = block.timestamp;
  uint256 constant private threshold = 1000;

  function buggy_reward(uint bet) public {
      //get the best seed for randomness?
      uint256 t = salt * block.timestamp/(salt % 5)
      if (t > threshold)
        msg.sender.send.value(bet * 100)().
  }
}
```

Malicious miner can control the “timestamp”

The GovernMental smart contract was designed to allow users to bet on whether no-one would place a new bet within 12 hours.

# Attacks made to the contract

## Denial of Service

- make it possible for a smart contract to be rendered non-functional
- created by poor access control and the potential for infinite loops or recursion
- E.g. Ethereum has a built-in maximum gas limit, meaning that the function could be impossible to run if largestWinner becomes too large.

```
1 function selectNextWinners(uint256 _largestWinner) {
2     for (uint256 i = 0; i < largestWinner; i++) {
3         // heavy code
4     }
5     largestWinner = _largestWinner;
6 }
```

# Other issues

## UNEXPECTED REVERT ATTACK

```
address[] private refundAddresses;
mapping (address => uint) public refunds;

function refundAll() public {
  for(uint x; x < refundAddresses.length; x++) {
    // now a single failure on send will hold up all funds
    require(refundAddresses[x].send(refunds[refundAddresses[x]]))
  }
}
```

Failed External Call

## BAD RANDOMNESS

Random numbers are required for smart contracts using a secret value, using embedded secret code.

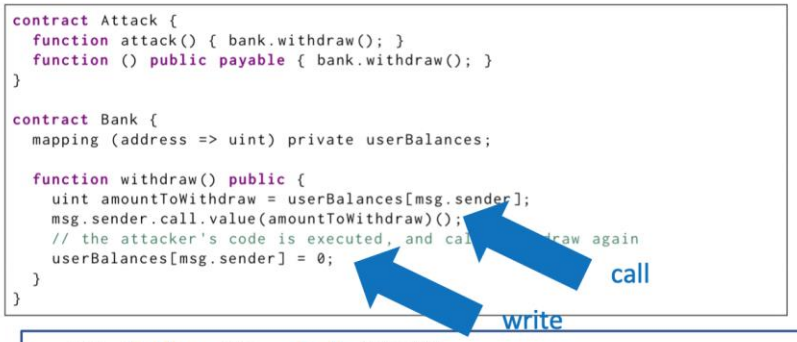
```
1 function play() public payable {
2     require(msg.value >= 1 ether);
3     if (block.blockhash(blockNumber) % 2 == 0) {
4         msg.sender.transfer(this.balance);
5     }
6 }
```

# Other issues

## FUNCTIONALITY DELEGATION

```
contract Wallet {
  // fixed address of the wallet library
  address constant walletLibrary = ...;

  // function that receives ether
  function deposit() payable {
    log(msg.sender, msg.value);
  }

  // function for withdrawing ether
  function withdraw() {
    walletLibrary.delegatecall(msg.data);
  }

  // ...
} // No guaranteed ether transfer
```

## WRITES AFTER CALLS VULNERABILITY




- "No Writes After Calls (NW)"
- "Path condition for the execution before the CALL is executed. We then check if such condition with updated variables"
- ...

Reliance on library for critical functionality

# Smart Contract Vulnerability Summary

| Attack Name | Purpose of attack | Target of Attack |
|---|---|---|
| Re-entrancy | Perform an action before updating state | Logical vulnerability, e.g. DAO breach |
| Access Control | Unauthorized access of data | Bypass application-level permission. E.g. attack Parity wallet |
| Arithmetic | Integer Overflows or Underflows attack | Lead to excessive amount of amount |
| Unchecked Return Value | Return of unchecked value | Gain unchecked value. E.g. gain King of Ether title |
| Denial of Service | Render a smart contract non-functional | Poor Access Control,<br>Potential for infinite recursion or loops. E.g. kill function on the library, causing self-destruct in Parity multisig wallets |
| Bad Randomness | Attack at randomness of data | Attack of guessable data randomness. E.g. SmartBillions lottery (Ethereum based) only keep track of past 256 blocks of number which would repeat after 73 minutes |
| Race Conditions | Use resources to gain better benefit | Allow users to make a profit by getting a trade in first. Can create a block. E.g. Bancor exchange allow users to take advantage of change in price for more benefits. E.g. King of Ether |
| Time Dependence | Attack the contract that depends on time | Attack at contract that depends on time period. Miner can create a block. E.g. GovernMental smart contract |
| Short Addresses | Exploit the arguments to a function that are stored in memory and that Ethereum automatically pads arguments | Attack at Ethereum with padding, larger transaction |

# Security Property to be verified

| Category | Security property |
|---|---|
| *Insecure coding* | Unrestricted write to storage |
| | Unhandled exception |
| | Missing input validation |
| | Unrestricted contract self-destruction |
| | Rounding due to division before multiplication |
| | Unnecessary write to storage |
| *Unsafe transfers* | Locked funds |
| | Rounding affects the amount of transferred funds |
| | Unrestricted transfer of funds |
| *Unsafe inputs* | Unsafe dependence on gas information |
| | Unsafe dependence on user input |
| | Delegatecall dependent on user input |
| | Unsafe call to untrusted contract |
| *Transaction reordering* | Transactions affect the amount of transferred funds |
| | Transactions affect the receiver of funds |
| | Transactions affect the execution of fund transfer |
| *Reentrancy issues* | Write to storage after constant-gas call |
| | Write to storage after call with unrestricted gas |

# Audit Smart Contract

**A Smart Contract audit is：**

- the process investigating carefully a piece of code,
- in this case a Solidity contract to find bugs, vulnerabilities and risks before the code is deployed and
- used in the main Ethereum’s network where it won’t be modifiable.
- It’s just for discussion purposes.

# Audit Smart Contract

**Structure of Audit Report**

- **Disclaimer:** Here you'll say that the audit is not a legally binding document and that it doesn't guarantee anything. That it's just a discussion document.
- **Overview of the audit and nice features:** A quick view of the Smart Contract that will be audited and good practices found.
- **Attacks made to the contract:** In this section you'll talk about the attacks done to the contract and the results. Just to verify that it is, in fact secure.
- **Critical vulnerabilities found in the contract:** Critical issues that could damage heavily the integrity of the contract. Some bug that would allow attackers to steal ether is a critical issue.
- **Medium vulnerabilities found in the contract:** Those vulnerabilities that could damage the contract but with some kind of limitation. Like a bug allowing people to modify a random variable.
- **Low severity vulnerabilities found:** Those are the issues that don't really damage the contract and could exists in the deployed version of the contract.
- **Line by line comments:** In this section you'll analyze the most important lines where you see potential improvements.
- **Summary of the audit:** Your opinion about the contract and final conclusions about the audit.

# Smart Contracts Audit

An audit structure could be:

- Exemption from liability: indicating that it is not a legally binding document, no one can ensure that no bugs will be found in the future that may take effect in existing contracts.
- Overview of the audit, indicating the good practices carried out.
- Attacks to a contract, where all the existing attacks to the contracts of the block- chain and their results, structured in serious, medium and low vulnerabilities will be covered.
- Line-by-line comments indicating possible safety and cost improvements for blockchain maintenance and execution.
- Summary of the audit with conclusions.

It is developed a tool called MAIAN which precisely specify and reasoning about the trace properties of the vulnerabilities in Smart Contracts: Greedy, Prodigal and Suicidal

# Smart Contracts Audit and Review

Analysis of the links between smart contracts and legal contracts.

It also provides a comparative analysis of imperative (procedural based) and declarative (logic-based) languages for smart contracts expression, considered from the context of blockchain systems.

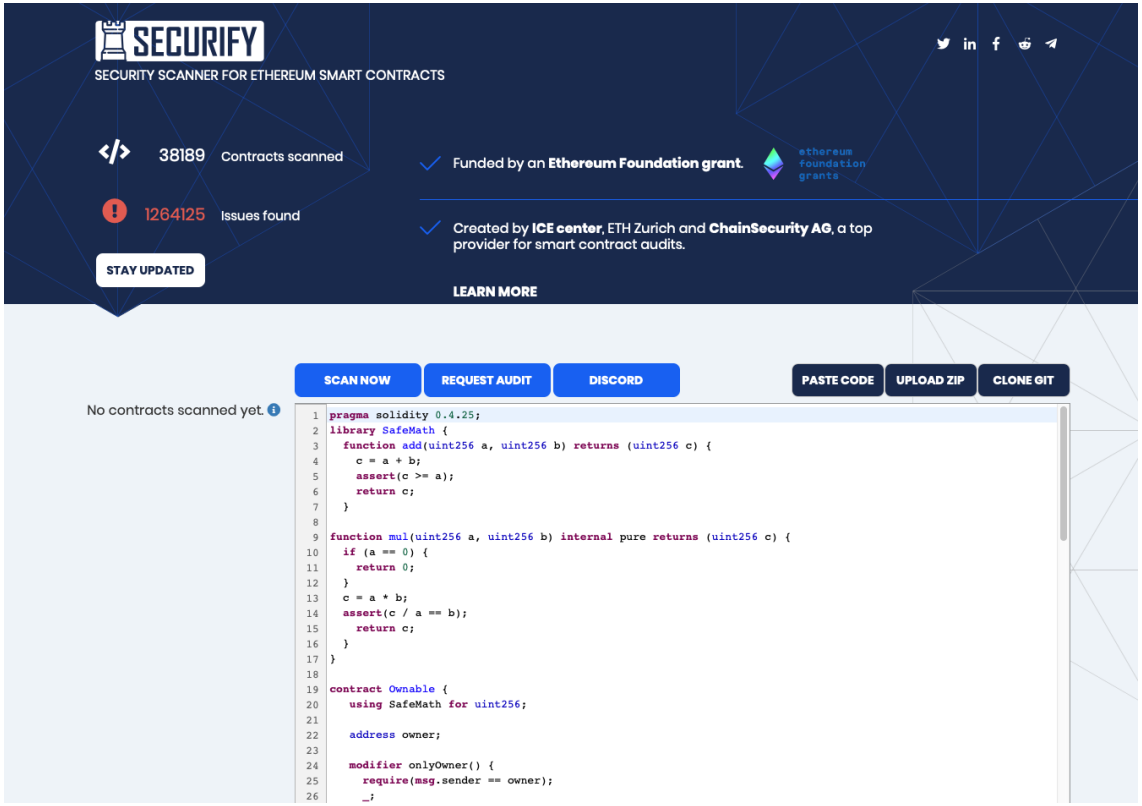
Imperative smart contracts are typically programmed in a procedural or an object-oriented language.

Declarative smart contracts can be written in different declarative languages, such as functional languages and logic-based languages.

Blockchain systems with their integrated digital currencies can facilitate contractual consideration. Misinterpretation of terms may be also reduced if smart contracted are published.

Blockchain systems can also be beneficial to the storage/notarizing and execution of smart contracts, without central control.

# Some tools that can be used




https://securify.ch.

# Another Smart contract security analysis and exploitation

Mythril to find security bugs in Solidity Code and smart contracts deployed on the Ethereum network.

```
$ myth analyze -m ether_thief tokensale.sol

==== Unprotected Ether Withdrawal ====
SWC ID: 105
Severity: High
Contract: TokenSaleChallenge
Function name: sell(uint256)
PC address: 696
Estimated Gas Usage: 6373 - 27034
Anyone can withdraw ETH from the contract account.

Arbitrary senders other than the contract creator can withdraw ETH
from the contract account without previously having sent an
equivalent amount of ETH to it. This is likely to be a
vulnerability.
--------------------
In file: tokensale.sol:25

msg.sender.transfer(numTokens * PRICE_PER_TOKEN)

--------------------
```

# Smart Contract security recommendations

Treat a smart contract like both a legal agreement and application code. Do both a legal review and software application testing before deploying the contract.

Threat model your smart contract to determine how vulnerable components or malicious use could cause unexpected or costly behavior.

Implement termination clauses in code; however, they can be tricky to secure.

Use the well-known techniques around the secure software development life cycle to analyze your smart contract code for known security vulnerabilities and problems.

Get a legal review after you've finished coding, but before you deploy to a SCE. Identify how the terms of the contract will mitigate losses if you have a legal dispute.

Make sure you establish terms for arbitration and for legal elements like choice of law and choice of venue. Because of their complexity and the potential complexity of the participants, without an established set of legal elements beforehand, it may be difficult to establish jurisdiction.

# Other kinds of Attacks

# Underground Dark Web Services

The Dark Web Services

- The two subjects of this case study are BancoPanama, http://bancopanuemswrrz.onion/index.html, a banking site selling anonymous 'offshore banking accounts' for bitcoin, and
- Dark Web UnlockDevices, http://unlockdehrka3cbn.onion/#home, a service that allows you to anonymously unlock phones through bitcoin payment.

# Analysis Tools For Dark Web Onion Sites

Fresh Onions

- http://zlal32teyptf4tvi.onion/
- Fresh Onions is a crawler for websites. It basically tells you any hidden information that you might not see on an onion site and is great for identifying any of these 'digital fingerprints' we are looking for.

Wallet Explorer

- https://www.walletexplorer.com/
- Wallet Explorer is useful as it identifies all Bitcoin addresses owned by one single wallet. When dealing with cryptocurrencies, one wallet may own numerous addresses.

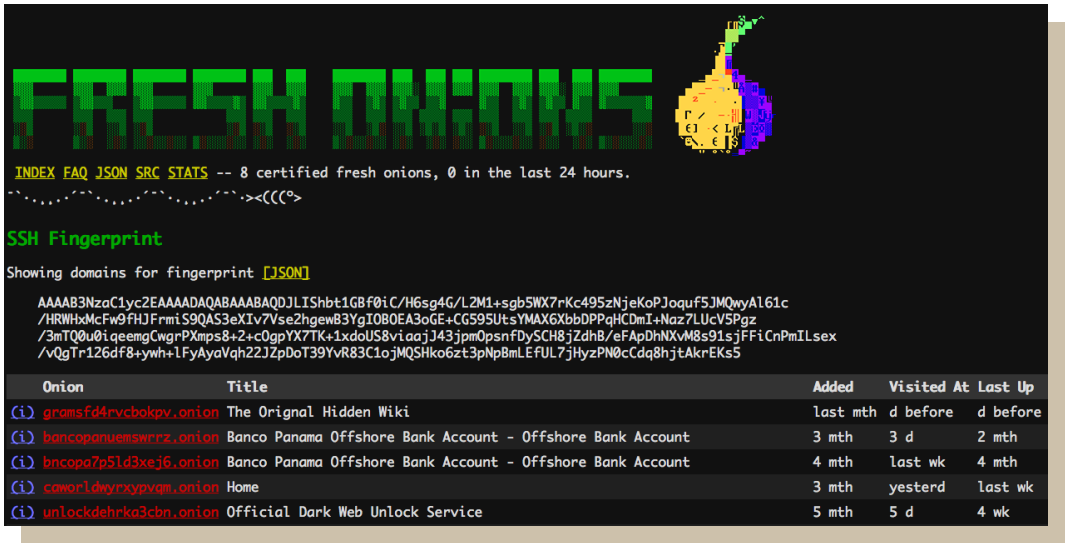
Blockchain Explorer (https://www.blockchain.com/explorer)

Tor Browser (https://www.torproject.org/)

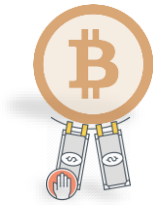
# Analysis Tools For Dark Web Onion Sites

The relationship between these two dark web onion sites can identified using the Fresh Onion crawler.

The relevant details we are looking for is an ‘SSH Fingerprint’.



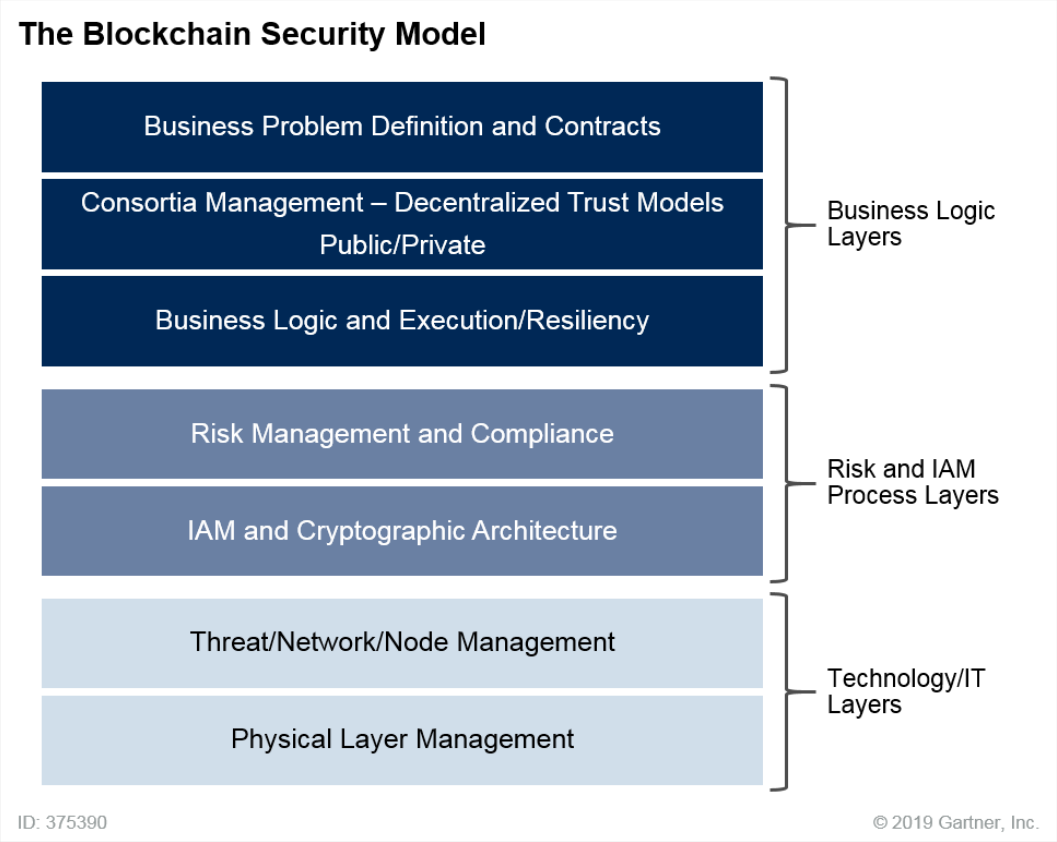
# Bitcoin and AML



The mutual constitution of technology and global governance

Bitcoin, blockchains, and the international anti-money-laundering regime

# Gartner's Blockchain Security Model



# Attack Surface


**Key attack**
- Keylogger malware
- Credential attack
- Server vulnerability
- Wallet attack

**Protocol/Algorithmic attack**
- Sybil attack
- Double Spending or 51% attack
- Faulty code / Faulty smart contract attack (DAO attack)

**Other attacks**
- Miner Ransomware
- DoS or DDoS attack

# Security Features and Configurations

Key Management

Certificate Authority

- Enrollment Certificate Authority
- Transaction Certificate Authority
- TLS Certificate Authority

Identity Management

Data Encryption and Data Hash

# Secure the Wallet

Be careful with online services

Keep small amounts for everyday uses

Backup the wallet

- Backup of entire wallet
- Encrypt online backups
- Use secure locations
- Make regular backups

Encrypt the wallet

- Encrypt
- Use strong password

Keep software up to date

# Privacy in Blockchain

# Anonymity and Privacy

Blockchain networks, because of encryption based on complex and robust cryptographic principles, can frequently be obfuscated quite effectively in a way that your transactions and data cannot be traced back to you.

it is indeed highly possible to make blockchain completely anonymous and private in nature, and this largely depends on what kind of cryptographic algorithm has been used for setting up a certain network.

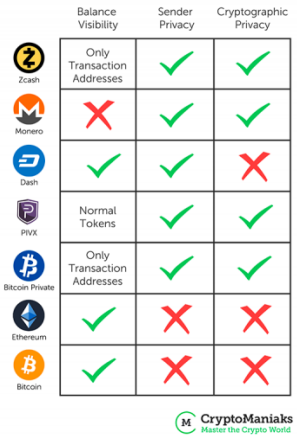
For blockchain networks underlying privacy coins like Monero and DASH, anonymity is key. When you transact Monero actually uses a different secret address each time.

In certain other cases, blockchain can be pseudonymous rather than anonymous, which is the case with the Bitcoin blockchain.

**For Bitcoin**

- Wallet address and transaction details are recorded in the blockchain
- Bitcoin transaction is linked to bitcoin address only.
- It is pseudonymous instead of anonymous cause intelligent tracker can link the address with all the transactions

**Privacy Coins Compared**



# GDPR

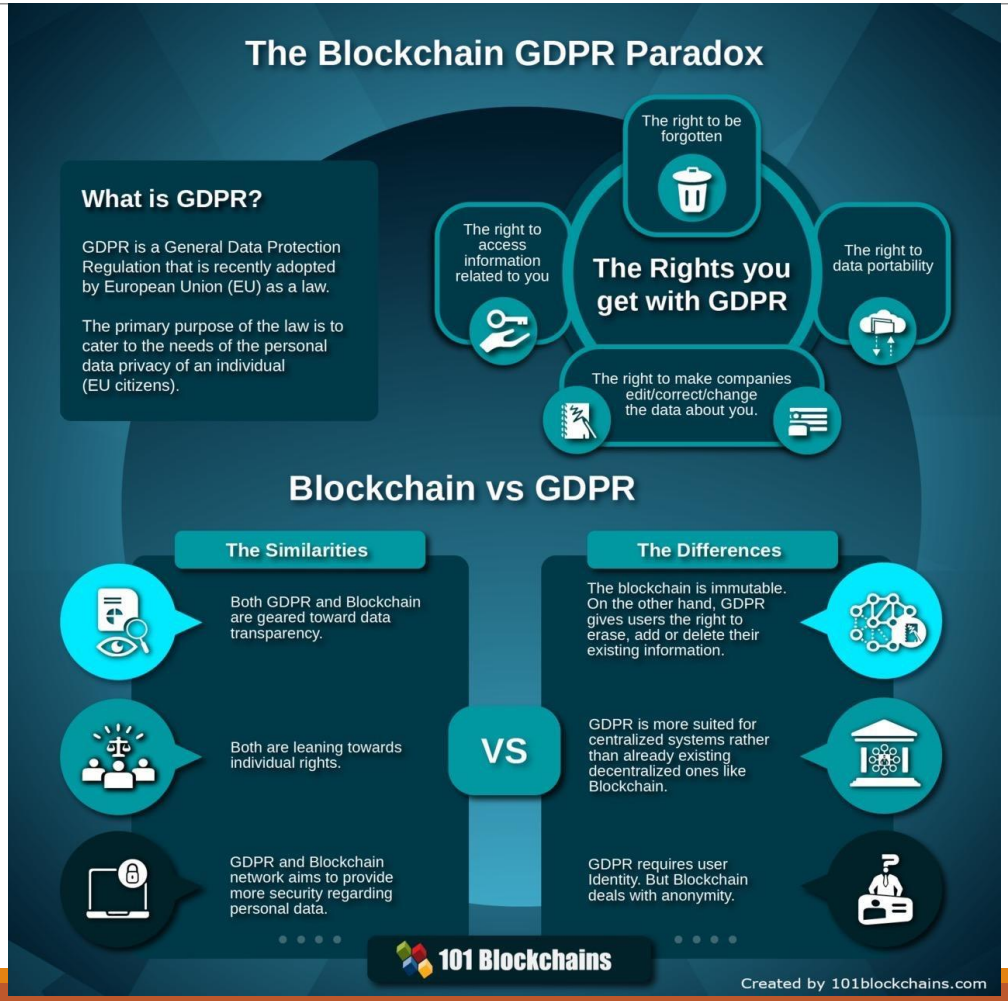
The newest one added to the list is the General Data Protection Regulation ( https://gdpr-info.eu/ ), widely known as GDPR. This is a regulation in EU law on data protection and privacy for citizens within the European Union.

The goal of this act, which has been enforced since 25 May 2018, is to assign the following rights to consumers:

- **The right** to access information related to you
- **The right** to be forgotten
- **The right** to data portability
- **The right** to make companies edit and correct information about you
- **The right** to control usage of personal data within geographic boundaries.

As per the newly devised law, violation of the GDPR protocol may lead to a penalty of “up to €20 million, or 4% of the worldwide annual revenue of the prior financial year, whichever is higher” ( www.gdpreu.org/compliance/fines-and-penalties/ )

# Blockchain GDPR Paradox



# Blockchain and GDPR

New and improving pruning techniques may help solve this problem as this process allows for data to be removed from blockchains when it is no longer wanted.

Best practice is to store all personal data “off-chain” which can then be linked back to the ledger by a hash.

Upon request, the personal data stored off the chain can simply be deleted, rendering the hash key useless

In the financial context, financial institutions have to comply with what is commonly known as the “know-your-customer” rule and must keep records of such transactions, including the personal data of the parties involved in the transaction

# Personal Data (Privacy) Ordinance (PDPO)

The three key characteristics of DLT that need addressing under the PDPO are:

1. the accessibility of some DLT platforms, in which all nodes have equal access to all stored personal data regardless of whether they need to see it;
2. the immutability of stored data, whereby data cannot be amended or erased;
3. the often cross-border nature of DLT, meaning that personal data may be stored outside Hong Kong. The requirement under Section 33 of the PDPO that prohibits the transfer of personal data outside of Hong Kong unless certain conditions are met is not currently in force.

# Personal Data (Privacy) Ordinance (PDPO)

Recommendations

- Avoid storing personal data in the ledger, but rather only keep the hashes of personal data in it.
- Storing personal data off the ledger in more conventional databases while keeping hashes in the ledger
- Discard encryption key when encrypted personal data in a ledger is no longer needed

# Methods for improving privacy

## Avoiding address reuse

- Addresses being used more than once is very damaging to privacy because that links together more blockchain transactions with proof that they were created by the same entity.

## Coin control

- Coin control is a feature of some bitcoin wallets that allow the user to choose which coins are to be spent as inputs in an outgoing transaction.

## Multiple transactions

- Paying someone with more than one on-chain transaction can greatly reduce the power of amount-based privacy attacks such as amount correlation and round numbers

## Change avoidance

- Change avoidance is where transaction inputs and outputs are carefully chosen to not require a change output at all.

# Methods for improving privacy

## Multiple change outputs

- If change avoidance is not an option, then creating more than one change output can improve privacy.

## Centralized mixers

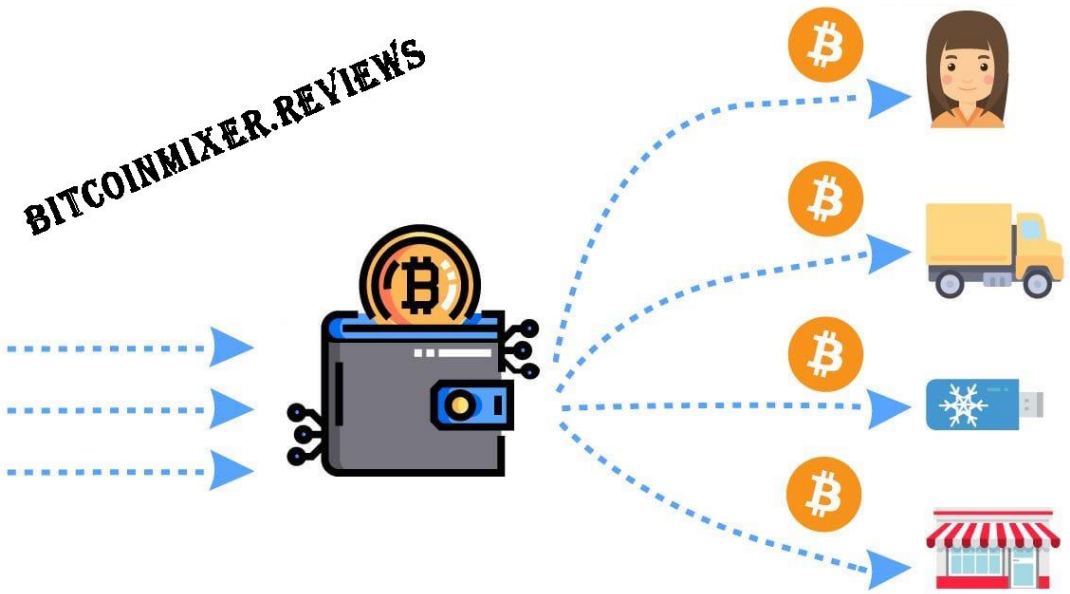
- This is an old method for breaking the transaction graph. Also called "tumblers" or "washers".
- A user would send bitcoins to a mixing service and the service would send different bitcoins back to the user, minus a fee. In theory an adversary observing the blockchain would be unable to link the incoming and outgoing transactions.

# Bitcoin Mixer

Bitcoin mixers or Bitcoin tumblers otherwise called cryptocurrency tumblers, bitcoin blenders are software that separates the exchanges into more modest parts and afterward blends them up with different exchanges and with coins on occasion before they are sent off to their destination.

Mixer

- Bitcoin Laundry
- BitBlender
- Mixtum
- Crypto Mixer
- Bit Cloak
- BitcoinMix
- …



# Bitcoin Mixer

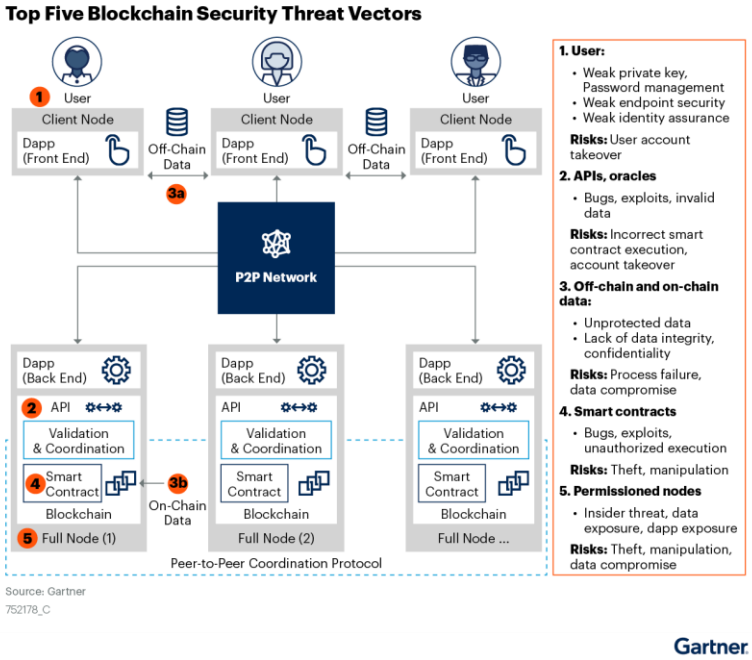
Bitcoin mixers are solutions (software or services) that let users mix their coins with other users, in order to preserve their privacy.

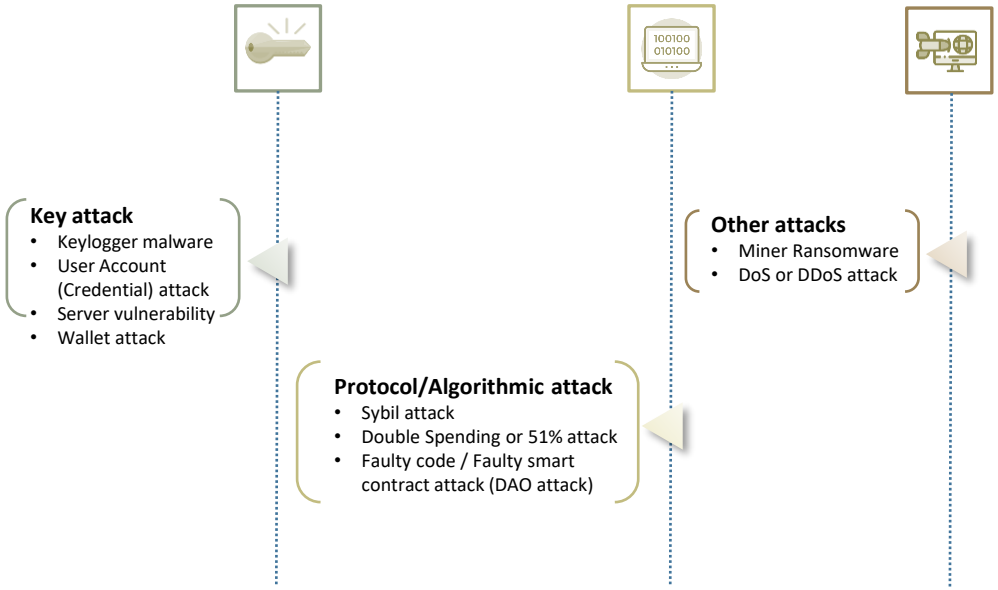
While Bitcoin addresses are “pseudonymous” — meaning, they don’t, in themselves, reveal the identity of their owner — they can often still be linked to real-world identities.

Mixing helps protect privacy and can also be used for money laundering by mixing illegally obtained funds. Mixing large amounts of money may be illegal, being in violation of anti-structuring laws.

# How to conduct FinTech Risk Assessment

# Attack Surface and Threat Vectors



# Conduct Fintech Risk Assessment

## Enterprise Background

- It focuses on consumer and retail payments, peer-to-peer (P2P) mobile payments, mobile wallets, and foreign exchange and remittances.

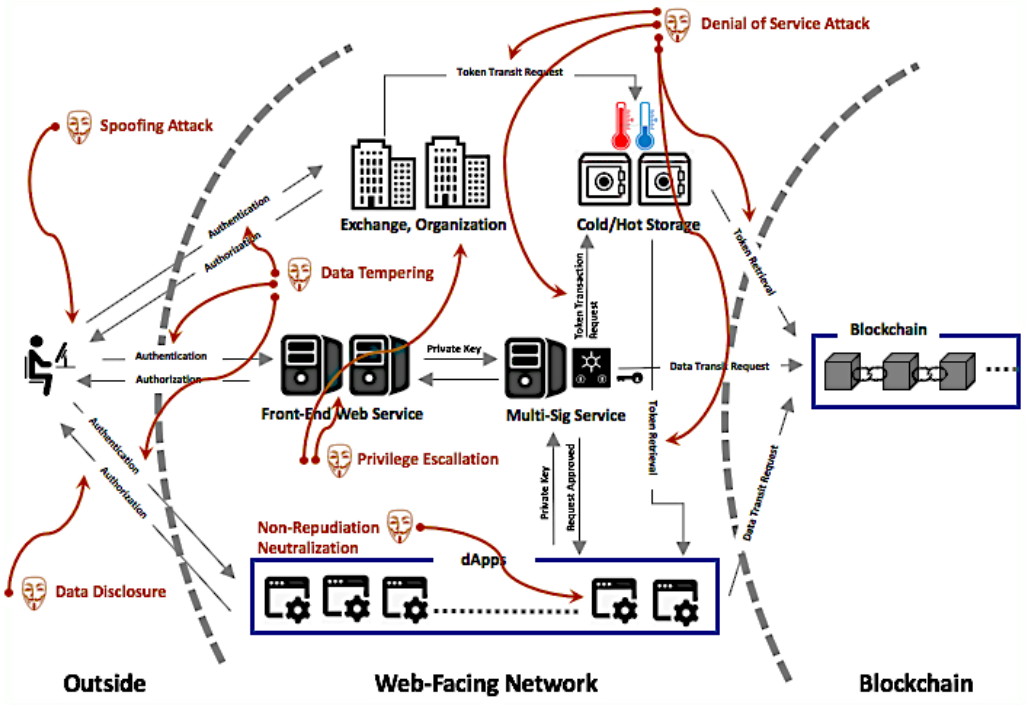
**Figure 3—Fintech Risk Management Framework**

| | | | | | | |
|---|---|---|---|---|---|---|
| **Domain 1:** Fintech Risk Management Phases and Processes | **Fintech Risk Organizing and Planning** | **Fintech Risk Assessment** | | | **Fintech Risk Reaction** | **Fintech Risk Monitor and Control** |
| | Organize and Plan Fintech Risk Effort | Identify Fintech Risk | Perform Qualitative Risk Analysis | Perform Quantitative Risk Analysis | Response to Risk | Monitor and Control Risk |
| **Domain 2:** Fintech Enterprise Risk Area | Investment Management | Customer Management | Regulation | Technology Integration | Security and Privacy | Risk Management Practices |
| **Domain 3:** Fintech Innovative Solutions Risk Area | New Business Models | Applications | Processes | Products | Services | |

# Blockchain Risk Assessment

- Organizing and Planning
- Select Risk Assessment Framework
- Threat Modelling
- Perform Technical Tests (VAPT)
- Perform Smart Contract Review (If needed)
- Conduct Blockchain Audit

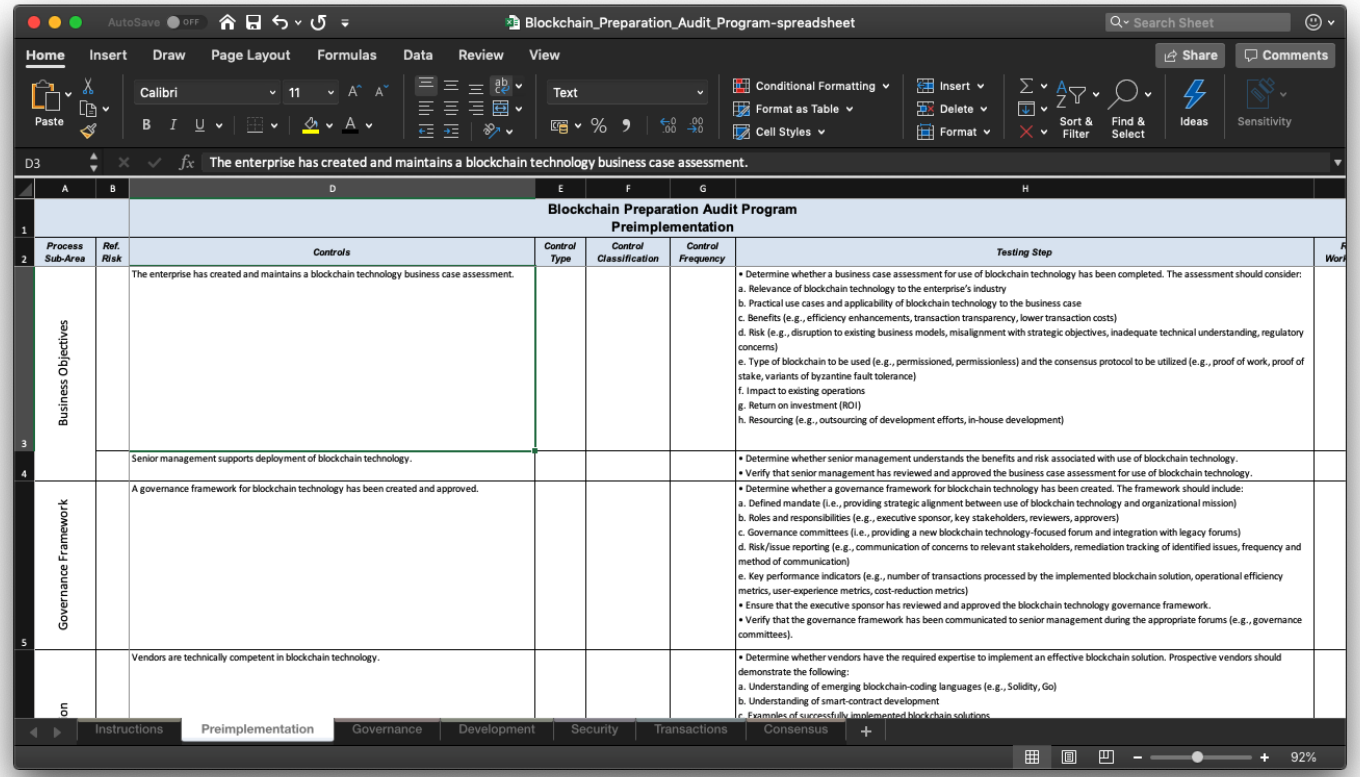
# Threat Modelling of a blockchain system




Lee, Jae Hyung. *Systematic approach to analyzing security and vulnerabilities of blockchain systems*. Diss. Massachusetts Institute of Technology, 2019.

# IT Audit of Blockchain (from ISACA)

# IT Audit of Blockchain (From ISACA)



Blockchain_Preparation_Audit_Program-spreadsheet

D3: The enterprise has created and maintains a blockchain technology business case assessment.

**Blockchain Preparation Audit Program**
**Preimplementation**

| Process Sub-Area | Ref. Risk | Controls | Control Type | Control Classification | Control Frequency | Testing Step |
|---|---|---|---|---|---|---|
| Business Objectives | | The enterprise has created and maintains a blockchain technology business case assessment. | | | | • Determine whether a business case assessment for use of blockchain technology has been completed. The assessment should consider:<br>a. Relevance of blockchain technology to the enterprise's industry<br>b. Practical use cases and applicability of blockchain technology to the business case<br>c. Benefits (e.g., efficiency enhancements, transaction transparency, lower transaction costs)<br>d. Risk (e.g., disruption to existing business models, misalignment with strategic objectives, inadequate technical understanding, regulatory concerns)<br>e. Type of blockchain to be used (e.g., permissioned, permissionless) and the consensus protocol to be utilized (e.g., proof of work, proof of stake, variants of byzantine fault tolerance)<br>f. Impact to existing operations<br>g. Return on investment (ROI)<br>h. Resourcing (e.g., outsourcing of development efforts, in-house development) |
| | | Senior management supports deployment of blockchain technology. | | | | • Determine whether senior management understands the benefits and risk associated with use of blockchain technology.<br>• Verify that senior management has reviewed and approved the business case assessment for use of blockchain technology. |
| Governance Framework | | A governance framework for blockchain technology has been created and approved. | | | | • Determine whether a governance framework for blockchain technology has been created. The framework should include:<br>a. Defined mandate (i.e., providing strategic alignment between use of blockchain technology and organizational mission)<br>b. Roles and responsibilities (e.g., executive sponsor, key stakeholders, reviewers, approvers)<br>c. Governance committees (i.e., providing a new blockchain technology-focused forum and integration with legacy forums)<br>d. Risk/issue reporting (e.g., communication of concerns to relevant stakeholders, remediation tracking of identified issues, frequency and method of communication)<br>e. Key performance indicators (e.g., number of transactions processed by the implemented blockchain solution, operational efficiency metrics, user-experience metrics, cost-reduction metrics)<br>• Ensure that the executive sponsor has reviewed and approved the blockchain technology governance framework.<br>• Verify that the governance framework has been communicated to senior management during the appropriate forums (e.g., governance committees). |
| on | | Vendors are technically competent in blockchain technology. | | | | • Determine whether vendors have the required expertise to implement an effective blockchain solution. Prospective vendors should demonstrate the following:<br>a. Understanding of emerging blockchain-coding languages (e.g., Solidity, Go)<br>b. Understanding of smart-contract development<br>c. Examples of successfully implemented blockchain solutions |

Instructions | Preimplementation | Governance | Development | Security | Transactions | Consensus

# IT Audit of Blockchain (From ISACA)

| Process | • Control Objectives |
|---|---|
| Pre-implementation | • The enterprise's use of blockchain technology aligns with its strategic objectives.<br>• The enterprise has an adequate governance framework to provide oversight for blockchain technology.<br>• The enterprise chooses appropriate vendors to deploy blockchain technology. |
| Governance | • Management oversight provides assurance that the enterprise's strategic objectives are not adversely affected by risk related to blockchain technology (internal or external).<br>• Regulatory risk has been identified and is appropriately mitigated (or accepted and monitored), to ensure that the enterprise's strategic objectives are not adversely affected.<br>• The enterprise's business continuity plan incorporates elements that address the effective operation of blockchain technology.<br>• Vendor contract administration and operational processes ensure ongoing alignment between the enterprise's strategic objectives and blockchain solutions. |

# IT Audit of Blockchain (From ISACA)

| Process | • Control Objectives |
|---|---|
| Development | • The adoption of a blockchain solution does not disrupt operations.<br>• Blockchain technology development supports business requirements and design.<br>• Adequate testing of the blockchain solution prior to deployment minimizes risk of disruption to operations.<br>• Deployment of the blockchain solution is appropriately managed and does not impact operations.<br>• The enterprise's change management program minimizes the impact of change-related incidents on day-to-day operations. |
| Security | • The enterprise has in place a secure approach for private key management in order to mitigate reputational and financial impact.<br>• The enterprise supports secure coding practices for blockchain source code (e.g., smart contracts, distributed ledger networks) in order to mitigate information security risk.<br>• The enterprise effectively manages blockchain network vulnerabilities through monitoring, remediation actions and communication to relevant stakeholders.<br>• Devices using the blockchain solution are properly managed by the enterprise (i.e., the devices are tracked, hardened and addressed if compromised). |

# IT Audit of Blockchain (From ISACA)

| Process | Control Objectives |
| --- | --- |
| Transactions | • The enterprise's blockchain transactions are consistent with the type of blockchain solution that has been implemented.<br>• Blockchain transaction fees are adequately managed |
| Consensus | • The enterprise effectively manages the chosen consensus protocol for the blockchain solution in terms of oversight, communication and risk mitigation.<br>• The infrastructure required for mining activities is consistent with the needs of the implemented blockchain solution. |

# Blockchain governance through COSO 2013 framework

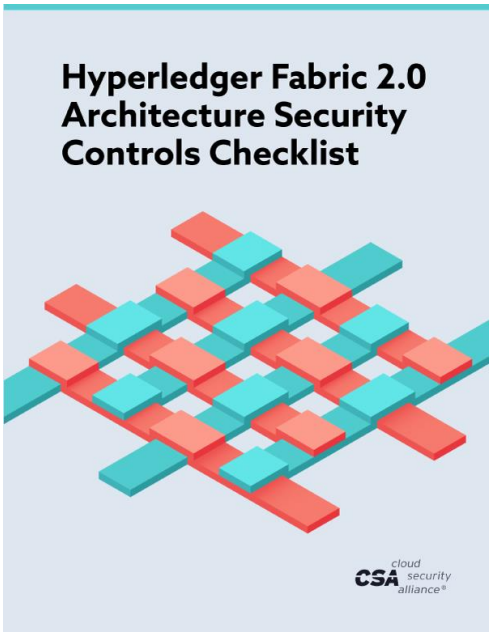

Table 1. **Implications of Blockchain on Five Components**

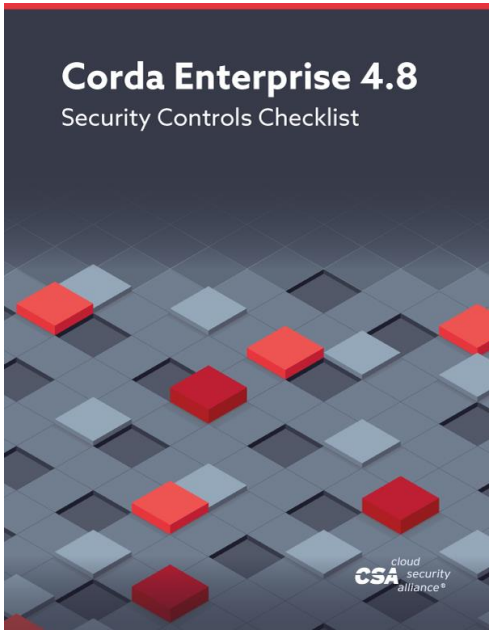
| Component | Implications of Blockchain |
|---|---|
| **Control Environment** | Blockchain may be a tool to help facilitate an effective control environment (e.g., by recording transactions with minimal human intervention). However, many of the principles within this component deal primarily with human behavior, such as management promoting integrity and ethics, which, even with other technologies, blockchain is not able to assess. The greater challenge relates to the intertwining of an entity with other entities or persons participating in a blockchain and how to manage the control environment as a result. |
| **Risk Assessment** | Blockchain creates new risks and simultaneously helps to mitigate extant risks, by promoting accountability, maintaining record integrity, and providing an irrefutable record (i.e., a person or organization cannot deny or contest their role in authorizing/sending a message or record). |
| **Control Activities** | Blockchain can act as a tool to help facilitate control activities. Blockchain and smart contracts can be a powerful means of effectively and efficiently conducting global business (e.g., by minimizing human error and opportunities for fraud). The collaborative aspects of blockchain, however, can introduce additional complexity, particularly when the technology is decentralized and there is no single party accountable for the systems that fall under ICFR. |
| **Information & Communication** | The inherent attributes of blockchain promote enhanced visibility of transactions and availability of data, and can create new avenues for management to communicate financial information to key stakeholders faster and more effectively. One aspect, in particular, for management to consider in applying blockchain is the availability of information to support the financial books and records, and related auditability of information transacted on a blockchain. |
| **Monitoring Activities** | The promise of blockchain to facilitate monitoring more often, on more topics, in more detail, may change practice considerably. The use of smart contracts and standardized business rules, in conjunction with Internet of Things (IoT) devices, may alter how monitoring is performed. |

# COSO 2013 Framework

| Components | Principles |
|---|---|
| Control Environment | 1. Demonstrates commitment to integrity and ethical values<br>2. Exercises oversight responsibility<br>3. Establishes structure, authority, and responsibility<br>4. Demonstrates commitment to competence<br>5. Enforces accountability |
| Risk Assessment | 6. Specifies suitable objectives<br>7. Identifies and analyses risk<br>8. Assesses fraud risk<br>9. Identifies and analyzes significant change |
| Control Activities | 10. Selects and develops control activities<br>11. Selects and develops general controls over technology<br>12. Deploys control activities through policies and procedures |
| Information and Communication | 13. Uses relevant, quality information<br>14. Communicates internally<br>15. Communicates externally |
| Monitoring Activities | 16. Conducts ongoing and/or separate evaluations<br>17. Evaluates and communicates deficiencies |

# Cloud Security Alliance Checklist

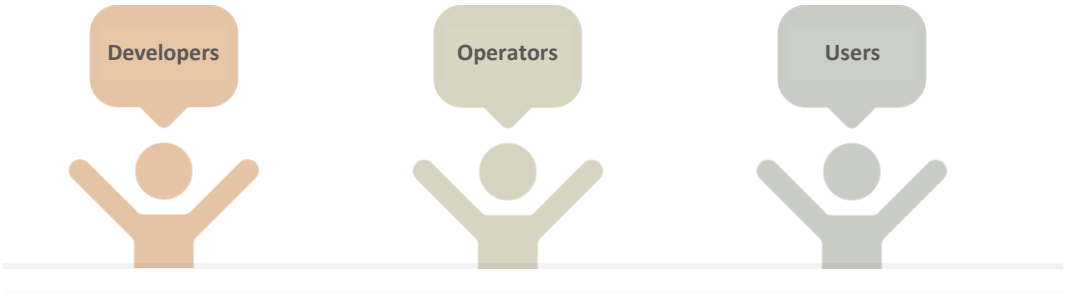




https://cloudsecurityalliance.org/

# Holistic View on Governance

# Governance of DLT/Blockchain

Definition of R&R in the DLT operations




Report and audit

Regulatory compliance

# Governance

Governance Structure of the organizations using DLT

- A consortium-like approach
- A joint venture approach
- A statutory organisation approach

Membership on-boarding and ongoing operation

- Due diligence check
- Security check
- End-user agreement
- User KYC

Technology audit

- Change management of smart contracts
- Administration controls of smart contracts
- Backup, recovery, key protection and revocation
- Verification of data sources

# Governance

Security management

- Data classification
- Hashing of non-sensitive (personal privacy) data

Data Privacy Governance

- Privacy Impact Assessment of the environment
- Use of Tokenisation
- Redacting of data via Merkle tree

Authentication, Access Control and Key Management

- Key management through secure mechanisms
- Key recovery process and mechanisms

Security administration and monitoring

- Access to the DLT environment
- Control of the database

# Governance

Physical security

- Physical access
- Environmental safety

System development and change management

- System development
- Smart contract development
- DApps development
- Testing and security testing
- Portability and compatibility issues
- Program change management

Information processing

- Disaster recovery and resilience management (against network malfunction and compromise of data integrity)
- Business continuity management process
- Define minimum viable number of validating nodes

# Governance

Communication network

- Secure network connectivity behind firewall and other standard security protection

Outsourcing

- Vendor selection
- Financial status of the party
- Technical capability evaluation

Other considerations

- Industry best practices, certification (e.g. ISO27001, CIS security, SANS Critical Security Controls) etc

# Blockchain Forensics

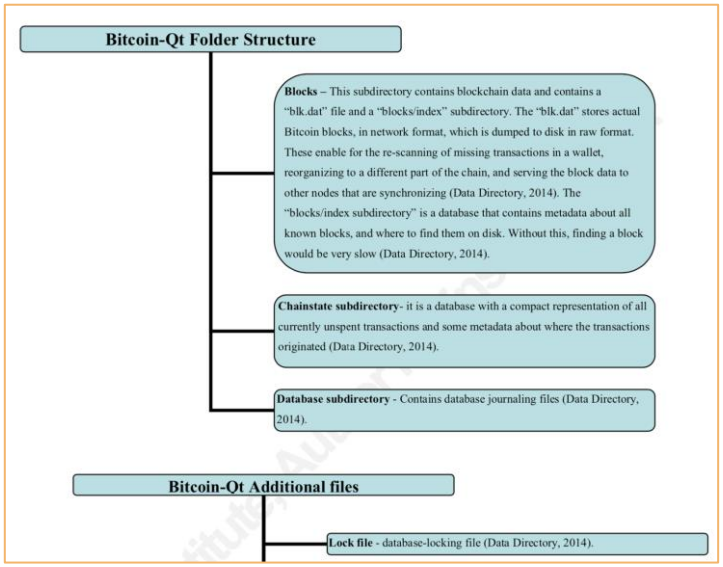
# Purpose of Blockchain Forensics

- Determine whether bitcoin transaction exist
- Determine whether a bitcoin wallet can be recovered
- Extract the ledger about transactions performed
- Analysis of Blockchain traffic

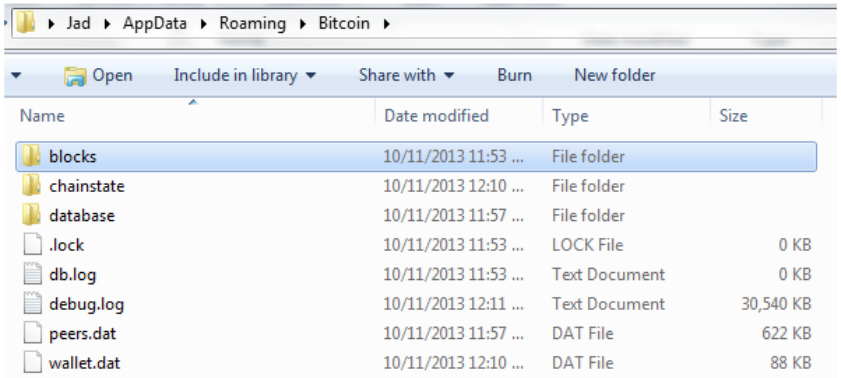
# Bitcoin Forensics

Bitcoin wallet artifacts from human readable public key address

- From software artifacts left behind. E.g.
  - Subdirectories left behind by Bitcoin-Qt software - .lock, db.log, debug.log, peers.dat, wallet.dat, etc




**Bitcoin-Qt file directory (Jad, 2013).**



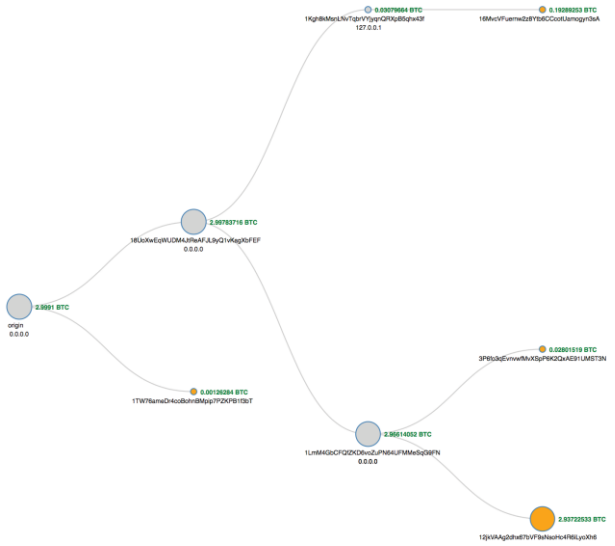
# Extract content from Blockchain about transaction

Extract information from Blockchain (based on Blockchain nature)

Can obtain the following information from one bitcoin address, such as:

- How many transactions have taken place
- Where money has come from and how much
- Where money has been sent and how much
- A historical timeline of transactions
- And other associated bitcoin addresses in that wallet

Based on [Wallet Explorer]



# Sample Forensics analysis

When we access their bitcoin addresses through the ‘purchase’ section of the website, we can identify two main addresses:

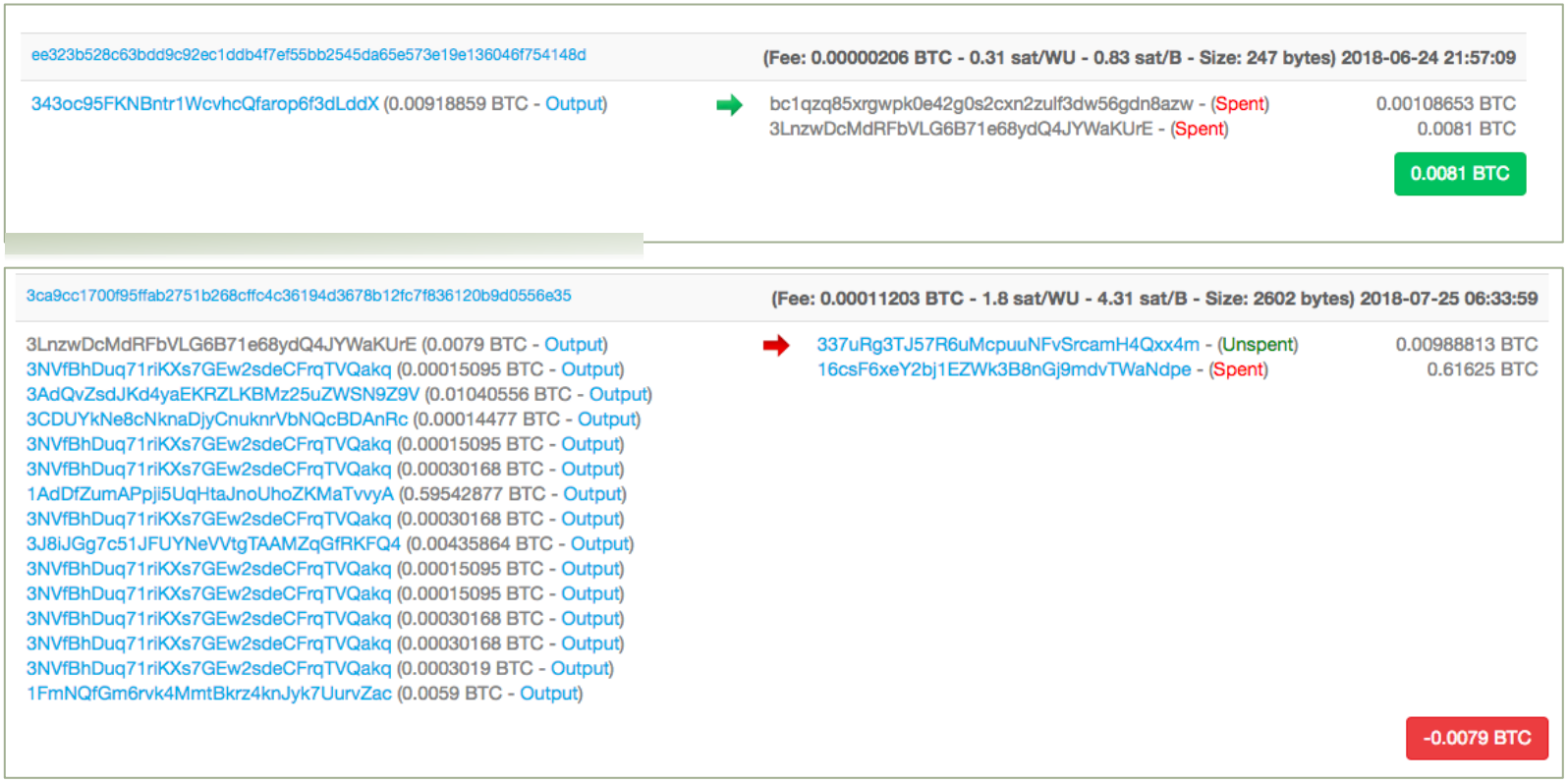
The **address** of the ‘**BancoPanama**’ site ending in **XZ4jo**

And the **address** of the ‘**Dark Web UnlockDevices**’ site ending in **KUrE**

The relationship between these two websites is that they are from the same wallet, meaning that they are owned by the same entity or person.

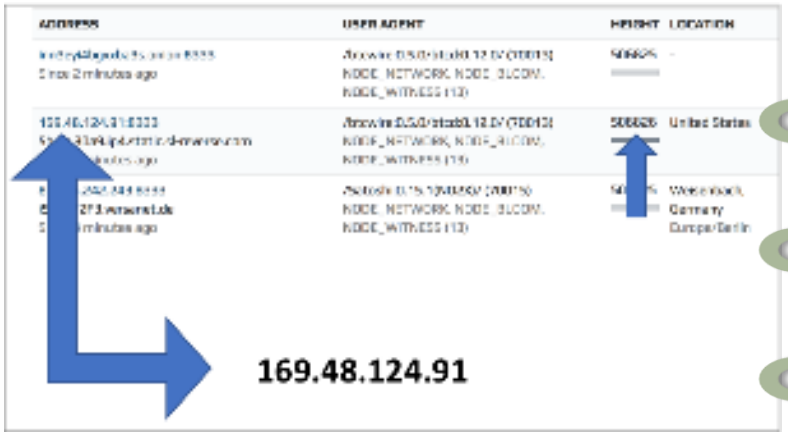
# Tracing payments through exchange

Through the address of blockchain and the wallet, the transaction relationship can be identified and determined.



# Blockchain traffic

**Sample of bitcoin communication traffic**




```
000000  F9 BE B4 D9 74 78 00 00  00 00 00 00 00 00 00 00   ....tx..........
000010  02 01 00 00 E2 93 CD BE  01 00 00 00 01 6D BD DB   .............m..
000020  08 5B 1D 8A F7 51 84 F0  BC 01 FA D5 8D 12 66 E9   .[...Q........f.
000030  B6 3B 50 88 19 90 E4 B4  0D 6A EE 36 29 00 00 00   .;P......j.6)...
```

**Figure 13-8:** Identifying the Bitcoin magic value.

- The next 12 bytes identify the packet type. In Figure 13-9, the value `74 78` with 20 zeros (which is `tx` in ASCII) identifies this as a transaction packet.

```
000000  F9 BE B4 D9 74 78 00 00  00 00 00 00 00 00 00 00   ....tx..........
000010  02 01 00 00 E2 93 CD BE  01 00 00 00 01 6D BD DB   .............m..
000020  08 5B 1D 8A F7 51 84 F0  BC 01 FA D5 8D 12 66 E9   .[...Q........f.
000030  B6 3B 50 88 19 90 E4 B4  0D 6A EE 36 29 00 00 00   .;P......j.6)...
000040  00 8B 48 30 45 02 21 00  F3 58 1E 19 72 AE 8A C7   ..H0E.!..X..r...
```

**Figure 13-9:** The packet type.

- The next 4 bytes are the length of the payload in bytes. This can be found at offset `000010` and are `02 01`. This is formatted in internal byte order, so `01 02` in hex corresponds to a payload of 258 bytes. See Figure 13-10.

```
000000  F9 BE B4 D9 74 78 00 00  00 00 00 00 00 00 00 00   ....tx..........
000010  02 01 00 00 E2 93 CD BE  01 00 00 00 01 6D BD DB   .............m..
000020  08 5B 1D 8A F7 51 84 F0  BC 01 FA D5 8D 12 66 E9   .[...Q........f.
000030  B6 3B 50 88 19 90 E4 B4  0D 6A EE 36 29 00 00 00   .;P......j.6)...
000040  00 8B 48 30 45 02 21 00  F3 58 1E 19 72 AE 8A C7   ..H0E.!..X..r...
```

**Figure 13-10:** The size of the payload in bytes.

# Legal aspects and regulation related to Blockchain and Cryptocurrencies

# Is Bitcoin legal

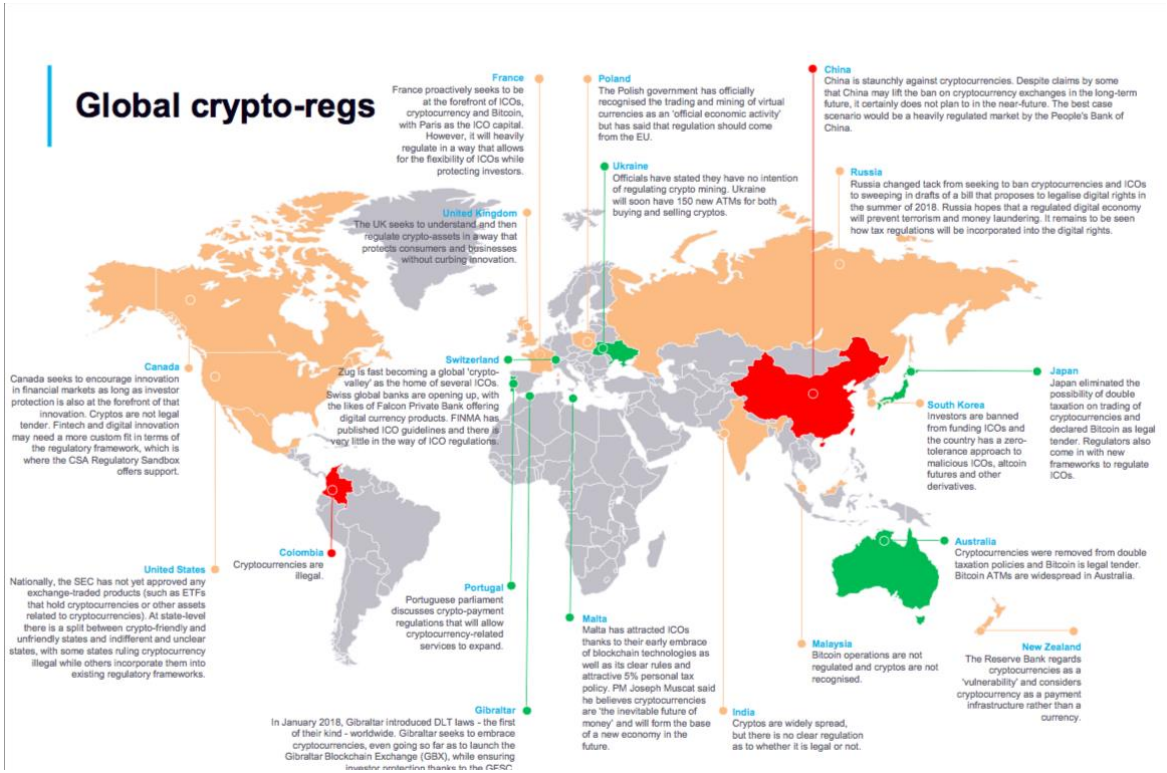
Bitcoin has not been made illegal by legislation in most jurisdictions.

However, some jurisdictions (such as Argentina and Russia) severely restrict or ban foreign currencies.

Other jurisdictions (such as Thailand) may limit the licensing of certain entities such as Bitcoin exchanges.

The Financial Crimes Enforcement Network (FinCEN), a bureau in the United States Treasury Department, issued non-binding guidance on how it characterizes certain activities involving virtual currencies.

# CryptoCurrency regulations (2018)




*Groups of countries depending on crypto regulation. Source: GreySpark.com*

# More regulations applied in 2022

## US Officials Push Collaboration, AML Controls for Crypto

Treasury, NSC Leaders Look to Curb Russian Sanctions Evasion

Dan Gunderman (dangun127) • March 4, 2022

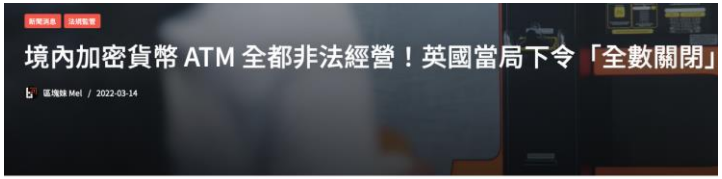
Twitter Facebook LinkedIn Credit Eligible Get Permission

The Russian ruble in an image courtesy of Paolinio via Pixabay

High-ranking U.S. officials say that while it would be nearly impossible for Russia to "flip the switch" and convert to cryptocurrency to stabilize its sanctioned economy, they caution that Russian elites and entities may yet try to skirt the measures by transferring and obfuscating funds across the blockchain.

**See Also:** The Ransomware Files, Episode 3: Critical Infrastructure

In an event hosted by the blockchain analytics firm TRM Labs on Friday, Todd Conklin, counselor to the deputy secretary of the U.S. Treasury Department, and Carole House, director of cybersecurity and secure digital innovation for the White House National Security Council, outlined the unprecedented federal activity over the past week that has hobbled Moscow and aims to choke Russia's economy as it continues its military campaign in Ukraine.




英國金融行為監管局（FCA）日前發出警告稱，該國境內所有的加密貨幣 ATM 都是非法經營，必須全數關閉。

根據 Coin ATM Radar 數據表明，當前英國境內有 81 台比特幣 ATM，分別由 8 家公司運營。FCA 表示將聯繫這些加密貨幣 ATM 的運營商，指示他們關閉機台，否則將面臨進一步行動。

聲明指出，任何在英國提供交易服務的加密貨幣 ATM 都必須事先向 FCA 註冊，並遵守英國《反洗錢條例》（MLR），FCA 續稱：

> 目前已註冊的加密資產公司「均未獲准」提供加密貨幣 ATM 服務，這意味著，任何在英國營運的加密貨幣 ATM 都是非法的，消費者不應使用它們。

自 2020 年 8 月以來，FCA 在《反洗錢條例》（MLR）框架下已向 33 家加密貨幣公司授予註冊批准，其中較為知名的公司有：Gemini Europe Ltd、Kraken 的控股公司 Payward Ltd、Galaxy Digital UK Limited 以及 eToro（英國）有限公司。

FCA 最後在總結表示：「我們經常警告消費者，加密資產不受監管且風險高，這意味著一旦出現問題，人們不太可能獲得任何保護。」

> 如果人們選擇投資加密資產，他們應該做好承擔「損失所有資金」的準備。

## Warning on illegal crypto ATMs operating in the UK

News stories | First published: 11/03/2022 | Last updated: 11/03/2022

Print Page Share page

**We have warned operators of crypto ATMs in the UK to shut their machines down or face enforcement action.**

Crypto ATMs offering cryptoasset exchange services in the UK must be registered with us and comply with UK Money Laundering Regulations (MLR). None of the cryptoasset firms registered with us have been approved to offer crypto ATM services, meaning that any of them operating in the UK are doing so illegally and consumers should not be using them.

The Upper Tribunal recently ruled against Gidiplus, a firm offering crypto ATM services, which wanted to continue trading, pending the Upper Tribunal's determination of its appeal against us refusing its application for registration under the MLRs. The judge concluded that there was a 'lack of evidence as to how Gidiplus would undertake its business in a broadly compliant fashion'.

We are concerned about crypto ATM machines operating in the UK and will therefore be contacting the operators instructing that the machines be shut down or face further action.

Since we published the list of unregistered crypto firms that may have been continuing to conduct business, a recent assessment found that 110 are no longer operational.

We regularly warn consumers that cryptoassets are unregulated and high-risk which means people are very unlikely to have any protection if things go wrong, so people should be prepared to lose all their money if they choose to invest in them.

# Hong Kong, rule of law and the free market

Bitcoin in general is not specifically regulated other existing regulations might apply and the above mentioned realities shape Bitcoin markets and businesses in Hong Kong

Financial Secretary John Tsang mentioned Bitcoin for the first time in a blog post on December 1, 2013

Bitcoin is per se not regulated by any of the financial regulatory bodies, such as the HKMA or the SFC.

Trading activities are controlled by the Customs and Excise Department, including commodities trading.

In March 2015 the Secretary for Financial Services & the Treasury, Prof KC Chan reiterated the government's stance that Bitcoin regulation is "not needed".

In September 2018 the HKMA reiterated their stance that Bitcoin is not money, but instead a type of commodity, in a speech given by its Chief Executive, Norman Chan

On November 1, 2018, the Chief Executive Officer of the SFC, Ashley Alder, announced in a keynote speech at Fintech Week the strict enforcement and clarification of existing rules

# Hong Kong, rule of law and the free market

## Funds

- Whether a fund needs a license from the SFC is independent of whether it holds these ‘virtual assets.’
  - A fund that only invests in cryptocurrencies (and not in futures or securities) does not need a type 9 license (asset management). If it however distributes such a fund in Hong Kong it requires a type 1 license. The SFC however expects that a fund holding cryptocurrencies behaves as if it held securities or futures.
  - All licensed funds that hold more than 10% of their funds in cryptocurrencies are now subject to special Terms and Conditions which may differ from fund to fund.

## Exchanges

- As explained above, trading activity is regulated by the Customs & Excise Department of the Hong Kong Police Force. The SFC only regulates those exchanges that give investment advice, trade futures or on margin.
- Under the new interpretations, the SFC will allow certain cryptocurrency exchanges into a ‘sandbox’ regime, where the exchanges have to behave as if they were license holders without receiving a license.
- Exempt from this sandbox are all currently viable business models, including spot exchanges, Bitcoin ATM providers, OTC exchanges that deal with non-professional investors, margin, derivatives and futures exchanges and those trading ICO tokens.

# Hong Kong, rule of law and the free market

Implications for funds that have more than 10% of their assets in cryptocurrencies and tokens (or state their objective to invest in crypto):

- SFC License necessary
- Fund needs to subscribe to 'Terms and Conditions" imposed by the SFC, which can vary from fund to fund
- Custody arrangements are expected
- Deal mainly with regulated exchanges
- Only interact with institutional investors

# Tokens, Initial Coin Offerings

Some tokens, especially those derived from “Token Creation Events” or “Initial Coin Offerings” are very likely securities, and their offering to unqualified investors in Hong Kong is illegal. Currently seems unlikely that the SFC will go after securities not specifically offered or advertised to Hong Kong investors.

On February 9, 2018 the SFC contacted Hong Kong exchanges urging them to delist tokens deemed securities. ICO issuers were also contacted. They either stopped their ICO or promised to comply with securities regulation.

The SFC clarifies in their statement from March 28, 2019: “Any person who markets and distributes Security Tokens (whether in Hong Kong or targeting Hong Kong investors) is required to be licensed or registered for Type 1 regulated activity. All usual due diligence, information and investor warnings apply.

# New Blockchain-based trade finance platform

In October 30, 2018, the Chairman of the Hong Kong Exchange & Clearing, Charles Li announced a new “blockchain-based” trade finance platform.

HONG KONG (Reuters)

- A new blockchain-based trade finance platform, developed by HSBC, Standard Chartered and 10 other banks, was launched in Hong Kong on Wednesday to boost efficiency in the multi-trillion-dollar funding of international trade.
- eTrade Connect, had allowed the Asia-focused British lender to reduce the time it takes to approve trade loan applications to four hours, compared with the usual one-and-a-half days.
- The use of blockchain technology in the banking industry is expected to reduce the risk of fraud in letters of credit (LoC)

# Hong Kong Regulator Sets Out Rules for Crypto Exchanges to Get Licenses (2019)

In late 2018, the Hong Kong Securities and Futures Commission (SFC) introduced licenses to cryptocurrency funds and related investment managers allowing them to sell digital asset products to their customers.

One year in since the announcement, a number of fund managers have complained that the authorities are frustrating their efforts to carry out their businesses

Industry players say very few fund managers have been approved to invest in crypto-currencies.  Reuters could only identify one: Hong Kong-based Diginex, which operates a cryptocurrency “fund of funds”.

# Hong Kong Regulator Sets Out Rules for Crypto Exchanges to Get Licenses (2019)

Hong Kong's financial regulator published new rules on Wednesday that would allow cryptocurrency exchanges to receive an operating license, a step intended to improve regulation and standards and help prevent fraud.

The new rules, under which exchanges can apply to be regulated from Wednesday, draw on the standards the SFC expects for conventional securities brokers.

They stipulate that an exchange that wants to be licensed must provide services to professional investors only, have an insurance policy to protect clients in case assets are lost or stolen, and use an external market surveillance mechanism.

SFC's Chief Executive Ashley Alder, announced a new crypto regulation framework that covers KYC/AML rules, custodial services and trading of digital assets in the crypto industry in Hong Kong.

In 2022, SFC further issues the Crypto Regulation Circular imposes a number of selling restrictions on the sale of virtual asset related products

# NFT Regulation Circular by SFC (2022)

In June, 2022, the Securities and Futures Commission issued a circular ("NFT Regulation Circular") reminding investors of risks associated with non-fungible tokens ("NFTs").

Where an NFT offers a fractional interest in a digital asset, such as a digital representation of an underlying asset, the NFT will likely be an arrangement in respect of property.

# New AML regulations

The government's proposal to regulate cryptocurrency exchanges could leave ATMs as one of the last remaining avenues for retail investing

In a November consultation, the Financial Services and Treasury Bureau proposed widening the scope of the city's anti-money-laundering and counterterrorist financing ordinance to include cryptocurrency exchanges.

The Hong Kong Monetary Authority (HKMA) has said that bitcoin and other cryptocurrencies are not legal tender but are virtual commodities. Bitcoin does not qualify as a means of payment as it does not have any backing from an issuer, or a physical form.

The price of bitcoin soars to historic highs, hitting US$24,000 on Wednesday after tripling this year. Photo: Getty Images

As Hong Kong's financial regulators tighten their scrutiny of cryptocurrencies, the machines that dispense bitcoin and other digital tokens may soon be among the last remaining avenues for individual retail traders.

A broader set of rules currently being considered would subject the city's virtual currency exchange platforms to licensing requirements by the Securities and Futures Commission, and forbid them from servicing retail investors.

But the bitcoin automatic teller machines (ATMs) could also eventually find themselves off limits if an appeal from the cryptocurrency community for the government to exclude them from the extended regulations designed to tackle money laundering goes unheeded.

There are roughly 60 bitcoin automatic teller machines (ATMs) across the city where users can buy and sell bitcoin and other digital currencies. Located in shopping malls and commercial buildings, they provide a convenient way for members of the public to trade bitcoin, or transfer cryptocurrencies to other users.

The price of bitcoin has soared to historic highs, hitting US$24,000 on

# Your guide to cryptocurrency regulations around the world (2018)

| Regional | View on bitcoin (cryptocurrency) | Policy on exchanges |
|---|---|---|
| Global regulators (G-20) | Legal tender, depending on the country. | No global regulator exists at the moment. |
| Japan | Legal tender as of last April. | Exchanges are legal if they are registered with the Japanese Financial Services Agency. |
| United States | Not legal tender, according to Financial Crimes Enforcement Network.<br><br>With individual State Law on Blockchain, https://blockchainlawguide.com/blockchain/ | Legal, depending on the state. |

# Your guide to cryptocurrency regulations around the world (2018)

Current Status of Cryptocompare (www.cryptocompare.com)

- The U.S. handles the second largest volume of bitcoin, roughly 26 percent, according to Cryptocompare.
- U.S. regulators differ in their definitions of bitcoin and other cryptocurrencies
- The Securities and Exchange Commission has indicated it views digital currency as a security.

# Your guide to cryptocurrency regulations around the world (2018)

| Regional | View on bitcoin (cryptocurrency) | Policy on exchanges |
|---|---|---|
| European Union | No EU member state can introduce its own currency, according to European Central Bank President Mario Draghi. | Legal, depending on the country |
| United Kingdom | Not legal tender. "Only sterling is legal tender in the UK," according to Carney | Legal, and need to register with the Financial Conduct Authority. They are required to meet the same anti-money-laundering counter-terrorism standards as other financial institutions, according to the BOE. |

# Your guide to cryptocurrency regulations around the world (2018)

| Regional | View on bitcoin (cryptocurrency) | Policy on exchanges |
|---|---|---|
| South Korea | Not legal tender. | Legal but use of anonymous bank accounts for virtual coin trading is prohibited.<br>Need to register with South Korea's Financial Services Commission.<br>Trading in South Korea makes up about 4 percent of daily volume of bitcoin. For other cryptocurrency such as XRP, trading in the Korean won commands a premium to U.S. dollars. |
| Switzerland | Legal | Legal, need to register with the Swiss Financial Market Supervisory Authority |
| | | |

# Your guide to cryptocurrency regulations around the world (2018)

Switzerland

- Four in 10 of the biggest proposed initial coin offerings have been based in Switzerland, according to a PwC report. The town of Zug, just south of Zurich, is nicknamed "Crypto Valley" and is home to blockchain companies including the Ethereum Foundation, and cryptocurrency wallet company Cardano.
- The Swiss Financial Market Supervisory Authority has put up clear guidelines for ICOs. https://www.finma.ch/en/~/media/finma/dokumente/dokumentencenter/myfinma/1bewilligung/fintech/wegleitung-ico.pdf?la=en&hash=9CBB35972F3ABCB146FBF7F09C8E88E453CE600C

# Your guide to cryptocurrency regulations around the world (2018)

| Regional | View on bitcoin (cryptocurrency) | Policy on exchanges |
|---|---|---|
| China | Not legal tender | Illegal. In 2017, the government banned ICOs — a way for start-ups to raise funds by selling off new digital currencies — and shut down domestic cryptocurrency exchanges. |
| Singapore | Not legal tender | Legal, may fall under regulatory purview of the Monetary Authority of Singapore. The Singapore dollar makes up 0.02 percent of daily global bitcoin trading volume but the country has emerged as a hub for ICOs. |
| India | Not legal tender, reportedly taking steps to outlaw it. | Legal. The Indian government has issued warnings but does not currently regulate exchanges. |

# Your guide to cryptocurrency regulations around the world (2018)

India

- India is taking steps to make cryptocurrencies illegal to use within its payments system and is looking to appoint a regulator to oversee exchanges.
- The government will "take all measures to eliminate the use of these crypto-assets in financing illegitimate activities or as part of the payment system," India's finance minister told lawmakers in New Delhi in February, according to a transcript by The Hindu newspaper.
- The country's tax department sent notices about cryptocurrency investing to tens of thousands of citizens after a national survey showed more than $3.5 billion worth of transactions have been conducted over a 17-month period.

# Legal Aspects (Risks) related to blockchain

Legal Basis

- Is information defined as property?
- Depend on mutual agreement between participating parties, so such uses should be adequately addressed in the contracts or terms and conditions of any DLT solutions
- The Hong Kong Electronic Transactions Ordinance (ETO) gives the same rights to digital signatures as to handwritten ones for legal documents for government use, and allows private parties to agree on the same.
- However, any assignment, mortgage or legal charge are not managed through ETO. They are within the meaning of the Conveyancing and Property Ordinance.

# Legal Aspects (Risks) related to blockchain

Legal aspects (Risk) related to

- Payment system adopted anti-money laundering (AML)
- Tax evasion
- Cryptocurrency Exchange
- Distributed nature of data
- Transnational
- Pseudonymous nature
- Autonomous System
- Enforceability of Smart Contracts
- Is token, property and ownership?

Distributed nature of data

- Secrecy and Privacy against co-competitors

Transnational

- Jurisdictional boundaries of nodes
- May have to use Alternative Dispute Resolution (ADR) method

Pseudonymous nature

- How can blockchain be compatible with data privacy law
- Usually more than one chain
- Immutable

# Legal Aspects (Risks) related to blockchain

Legal aspects (Risk) related to

- Payment system adopted anti-money laundering (AML)
- Tax evasion
- Cryptocurrency Exchange
- Distributed nature of data
- Transnational
- Pseudonymous nature
- Autonomous System
- Enforceability of Smart Contracts
- Is token, property and ownership?

Autonomous System

- What legal status will attach to a DAO? Organization? Person? Autonomous?

Enforceability of Smart Contracts

- Decentralized authority, who should be responsible?
- What's the role of the contract developer and implementer?

Is token, property and ownership?

- In common law, as a general principle, there is no property right in the information itself
- So can digital assets be considered as property?
- Who should be the owner?

# Legal Aspects and regulations

## Automatic Sovereignty

- Increase use of automation of scheme that verify and perform actions based on actor identity

## Distributed immutable legal support

- Provide distributed legal document support
- If any digital evidence were documented into blockchain, any tampered evidence would be detected immediately through the use of hash pointer

## Smart Contract

- Through the means of decentralized trading autonomous environment for power supply or real-state properties
- But may need to translate into code the clauses of a legal agreement.

## Cross-country transaction

- Contract can be cross-country and executed automatically
- Article 34 of the European Directive on the Information Society and Electronic Commerce promote use of electronic means

## Opportunity

- Use of distributed immutable legal as legally supported mechanism
- Use of smart contract as legal binding contract
- Cross-country transaction

# Legal Aspects (Risks) related to blockchain

The issue of liability associated with participation in a DLT platform, including the use of smart contracts, is a complex one.

Liability arising from harm or losses caused by a failure in the use of DLT (such as data breaches, hacking and non-delivery of assets)

Liability of Autonomous system that no one is in control.

# Technical issues of Blockchain technology

**Limited scalability**

- Low throughput
- Slow transaction times

**Storage constraints**

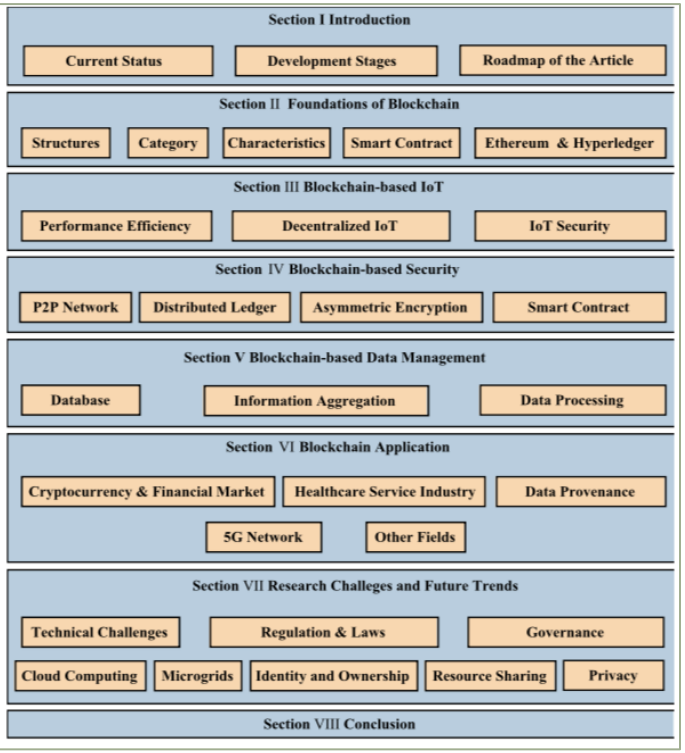
- Limited by nodes

**Unsustainable consensus mechanisms**

- Mining pool centralization
- High energy consumption

**Inadequate Tooling**

**Quantum Computing Threat**




Roadmap of the study

# Challenges and Opportunities (from BSI)

## WHAT ARE THE CHALLENGES FACING DLT/BLOCKCHAIN?

- Lack of clarity on the terminology and perceived immaturity of the technology
- Perceived risks in early adoption and likely disruption to existing industry practices
- Insufficient evidence on business gains and wider economic impact
- Lack of clarity on how the technology is/would be governed
- Uncertainty around regulation
- Multiple non-interoperable implementations and resulting fragmentation
- Maintaining security and privacy of data
- Ensuring integrity of data and strong encryption
- Energy-intensive nature of the technology
- Lack of clarity regarding smart contracts and how to implement them through DLT/Blockchain

## WHAT ARE THE OPPORTUNITIES FOR DLT/BLOCKCHAIN?

- Providing efficiency gains (including cost savings) for businesses and end-users
- Enabling new revenue sources
- Enabling new economic and business models
- Improving resilience and security in transactional systems
- Empowering end-users and improving trust in transactions
- Offering benefits for recording and reporting of data and activities through immutability
- capabilities
- Enabling management of digital identity through public key cryptography
- Providing the underlying mechanism for smart contracts and enabling smart auditing capabilities

# Miscellaneous

Ethernaut, https://ethernaut.openzeppelin.com

CaptureTheEther, https://capturetheether.com

Damn Vulnerable DeFi, https://www.damnvulnerabledefi.xyz

Paradigm CTF, https://github.com/paradigm-operations/paradigm-ctf-2021