Blockchain and Voting – What Could Possibly Go Wrong?

By Julie Baker, Cyber Expert & CISO, retired
jibaker@hotmail.com

Amid growing calls to modernize election infrastructure, blockchain has entered the conversation as a possible solution for securing and streamlining digital voting. Some public figures have floated the idea—whether seriously or speculatively—that blockchain could restore trust in election outcomes. It's a seductive proposition: decentralized, tamper-resistant, and verifiable by design.

Indeed, blockchain has proven transformative in sectors like cryptocurrency, decentralized finance, and real estate, where transparency and immutability are assets. But can the same technology—engineered for financial transactions—hold up under the unique demands of democratic voting?

This article dives into that question by examining the security architecture of blockchain systems, known attack vectors, and the very real trade-offs required to adapt blockchain for elections. At stake isn't just technological feasibility, but whether the cost of implementation is the integrity it promises to uphold.

What is Blockchain?

Blockchain is a peer-to-peer, decentralized, secure digital ledger that records transactions whether they are financial transactions, votes or other items, across many computers. It is made up of a chain of blocks, where each block contains a list of transactions, a timestamp, and a cryptographic hash of the previous block, linking them together. This system enables secure, transparent, and tamper-resistant (immutable) record-keeping, and is used for cryptocurrencies like Bitcoin, smart contracts, supply chain tracking, voting and more.

Benefits of Blockchain

There are many benefits to blockchain, most notably the following:

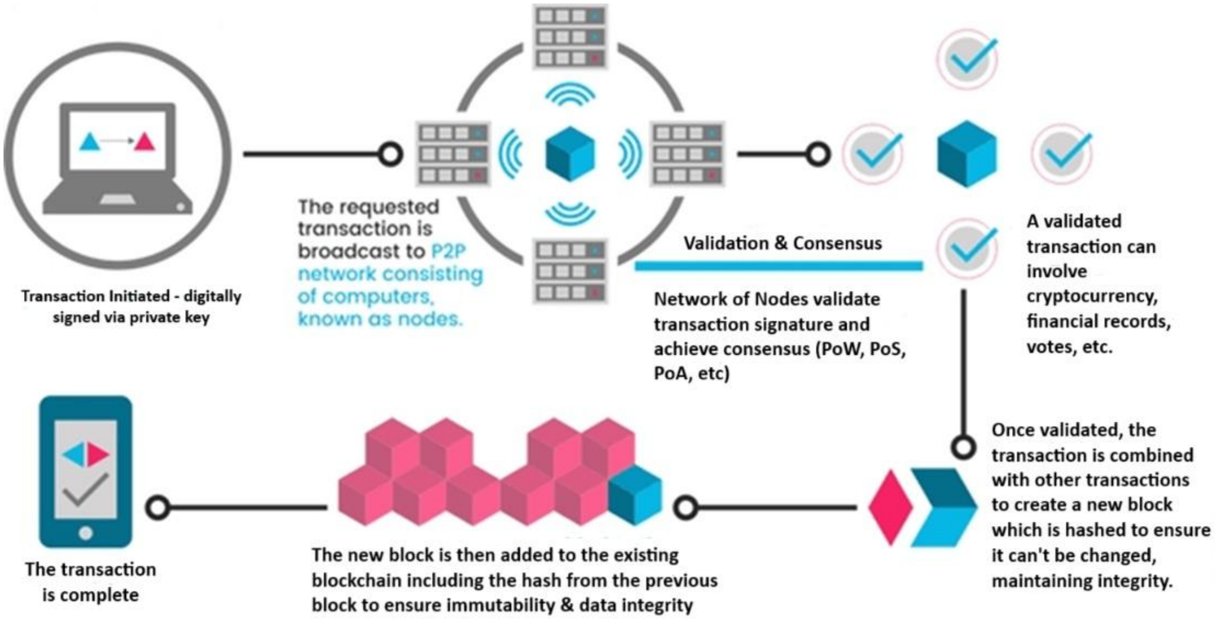
Decentralization: at a high level, copies of the blockchain are stored on multiple computers (nodes) in a network. The nodes must reach a consensus on the blockchain's state. There is no central authority making decisions about what to add or not add to the blockchain; it is a fully distributed, peer-to-peer system.

Immutability: Once a block is added **(**append only), it's nearly impossible to alter it due to the cryptographic links and network consensus, ensuring integrity and security. It is a record, in order, of all transactions (when I write transaction think votes) and can't be changed. There are some caveats to this which we will go into later.

Transparency: Most blockchains are public, so anyone can view the transaction history at any time, though user identities can be protected if required. The latter is very important for banking and voting systems that make use of blockchain technology so they can protect the privacy of the individuals – whether account owners or voters.

It sounds wonderful indeed. These characteristics also make blockchain appealing for election integrity activists. Who wouldn't want full transparency, immutability, decentralization and security for a voting system. These are things we have wanted for a long time. So why wouldn't we push for blockchain for voting? Or is this too good to be true?

How Blockchain Works


Transaction Initiated - digitally signed via private key
The requested transaction is broadcast to P2P network consisting of computers, known as nodes.
Validation & Consensus
Network of Nodes validate transaction signature and achieve consensus (PoW, PoS, PoA, etc)
A validated transaction can involve cryptocurrency, financial records, votes, etc.
Once validated, the transaction is combined with other transactions to create a new block which is hashed to ensure it can't be changed, maintaining integrity.
The new block is then added to the existing blockchain including the hash from the previous block to ensure immutability & data integrity
The transaction is complete

I will start by explaining at a high level how blockchain works. Let's say you as an individual initiate a transaction online and digitally sign it using your private key which is often stored in your digital wallet[i] or some other secure storage mechanism. This key validates that this is your transaction since no one else has your private key.

Everyone on the blockchain will have your public key (stored in their digital wallets) which means they will be able to read your data and know it came from you. This is called asymmetric cryptography (public/private keys) which makes sure only you can control your own money or vote, and everyone can trust the transactions are legitimate. Establishing ownership and identity in the blockchain is critical to ensure that any dollars that change hands, or votes that are cast, are done legitimately and not fraudulently.

The transaction is then broadcast to all the computers, also known as nodes, on the peer-to-peer network. A peer-to-peer network is a decentralized network where each participating computer (node) is "equal" to all the others, directly interacting with each other without relying on a central server or intermediary.

These nodes then validate the transaction – is the sender known and valid, does the sender have the funds in his/her account, and other checks.   The blockchain also has a consensus mechanism which is a way of determining whether or not the new block is valid and should be added to the blockchain. All nodes can participate in consensus, but not all do. The consensus mechanism is used to maintain immutability and make sure only valid transactions are appended to the blockchain. They generally work on a reward/punishment or carrot/stick approach. This is a brief explanation of the some of primary consensus mechanisms:
NOTE: There are many variations on the mechanisms described below and there are proprietary mechanisms used as well. We will just focus the primary mechanisms here.

- Proof of Work (PoW): Miners compete to solve complex mathematical puzzles directly associated with the block of data to validate transactions and add blocks to the blockchain, requiring significant computational power and cost.   This energy-intensive process ensures security through difficulty but can be slow and resource-heavy, as seen in Bitcoin. Not all nodes do the mining because it is so computationally expensive, but other non-mining nodes do contribute by verifying transactions or blocks. Miners are rewarded for this effort with cryptocurrency and punishments for bad behavior are usually in the form of lost resources, or orphan blocks (their transactions are not accepted).
  NOTE: In PoW, miners compete to find a valid hash for a block by solving a cryptographic puzzle. The puzzle involves finding a hash of the block header that is below a specific target value (determined by the network's difficulty) so the hash puzzle is directly tied to the block to be appended.
- Proof of Stake (PoS): Validators are chosen to create new blocks based on the amount of cryptocurrency they hold and "stake" as collateral, reducing energy consumption compared to PoW. It's faster and more scalable, but may favor wealthier participants. These validators can be rewarded with cryptocurrency, transaction fees or some sort of base reward, but this can vary. And there are punishments, called slashing, if a validator behaves badly, where some of their stake can be confiscated.

- Delegated Proof of Stake (DPoS): A derivative of Proof of State above. Token holders vote for a small number of delegates who validate transactions and create blocks. It's faster and more scalable.
- Proof of Authority (PoA): This is more of a niche mechanism – used more by private blockchains. Pre-approved, trusted nodes validate transactions and add blocks, relying on their reputation rather than computational power or stake. It's highly efficient and fast, ideal for private blockchains, but less decentralized due to centralized validator selection. These validators are rewarded usually via transaction fees or some sort of negotiated fixed reward.

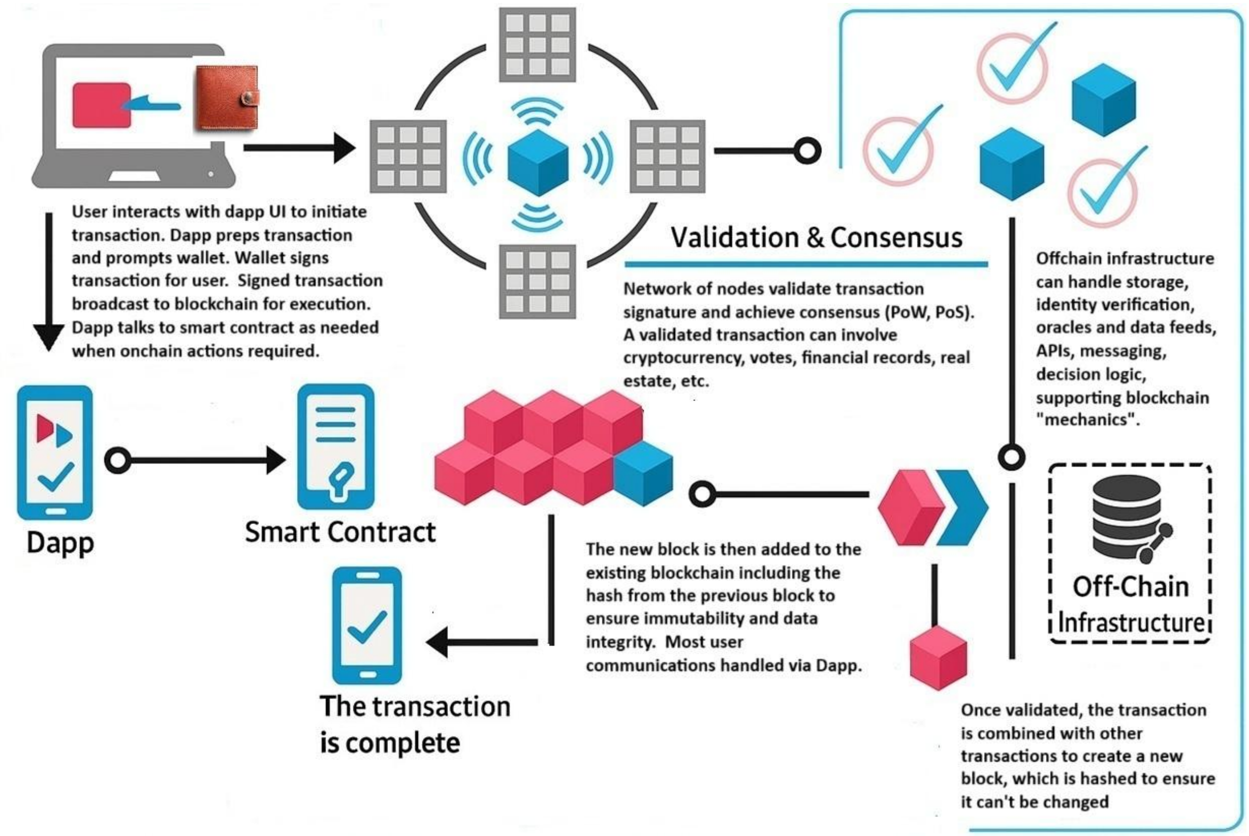
In a nutshell, consensus is about building the blockchain forward (appending new data), while immutability is about locking the past in place (maintaining integrity).

Once consensus among the nodes has been achieved, the transaction is combined with other transactions into a new block (most common way of doing this is called a Merkle tree[ii]). This block is hashed using a cryptographic hashing algorithm and that hash is stored in the previous block, linking them together into a chain. You can think of a hash like creating a unique fingerprint for a piece of data. If you change the data in a given block, then the hash from that block will no longer match the original hash and this will break the chain of hashes. This is how the data in a blockchain maintains its integrity; if the data is changed everyone on the blockchain will know and the chain is broken. All the nodes are updated with the new version of the blockchain – just be aware this can take time to propagate to all the nodes in a truly distributed environment.

You can probably see the use case for voting. Your vote is digitally signed (digital fingerprint) by you, proving it is you. It is then stored in a transparent, immutable ledger without any centralized control – no big brother involved. But then there is the privacy issue. With "pure" public blockchains there really is no privacy, but for a voting application, privacy is critical. How you voted, who you voted for is supposed to be private. We will discuss this in a later section – See #1 below for more information.

What Could Possibly Go Wrong?

Blockchain is a great technology – secure, immutable ledger, fully transparent and decentralized. But, it is not usually implemented as the simple blockchain described above. There are lots of other components including wallets, smart contracts, DApps, off-chain infrastructure that all make up a blockchain ecosystem. This image, still very high level, gives a better idea of what a more real world blockchain looks like.



User interacts with dapp UI to initiate transaction. Dapp preps transaction and prompts wallet. Wallet signs transaction for user. Signed transaction broadcast to blockchain for execution. Dapp talks to smart contract as needed when onchain actions required.

Network of nodes validate transaction signature and achieve consensus (PoW, PoS). A validated transaction can involve cryptocurrency, votes, financial records, real estate, etc.

Offchain infrastructure can handle storage, identity verification, oracles and data feeds, APIs, messaging, decision logic, supporting blockchain "mechanics".

The new block is then added to the existing blockchain including the hash from the previous block to ensure immutability and data integrity. Most user communications handled via Dapp.

Once validated, the transaction is combined with other transactions to create a new block, which is hashed to ensure it can't be changed

What are all these other components that make up a blockchain ecosystem?

1. **Wallet or other infra for storing keys** - software or hardware tool that allows users to store, manage, and interact with their digital assets, such as cryptocurrencies or tokens, on a blockchain. It primarily handles the cryptographic keys—private keys for signing transactions and public keys for receiving funds—that enable users to access and control their assets on the blockchain. Digital wallets don’t store the actual coins or tokens (which exist on the blockchain ledger); instead, they store the keys that prove ownership and allow users to send, receive, or manage those assets. Most of the blockchain voting companies use wallets to store cryptographic keys as well.
2. **Smart contracts** - A smart contract is a self-executing program that runs on a blockchain, with the terms of the agreement directly written into its code. Once deployed, it operates autonomously—automatically carrying out actions like transferring funds or verifying conditions—based on predefined rules and triggers. smart contracts are typically considered *on-chain* software because they are deployed and executed directly on a blockchain. The blockchain voting companies also use smart contracts as part of their voting systems.
3. **DApps** - (Decentralized Application) in the context of blockchain is a software application that runs on a decentralized network, typically a blockchain, rather than a

centralized server. It leverages blockchain technology to ensure transparency, security, and immutability, operating without a central authority. The blockchain voting companies also use DApps as part of their voting systems.

4. **Off chain infrastructure - s**ystems, processes, or technologies that operate outside the main blockchain (off-chain) to enhance its functionality, scalability, or efficiency while maintaining a connection to the blockchain for security or final settlement. Blockchain voting companies typically rely on a combination of on-chain and off-chain infrastructure to support their systems, as off-chain components are often necessary to address scalability, privacy, cost, and usability challenges.
5. **Exchanges –** Not on the diagram, but they can be an integral part of the blockchain ecosystem as well. Exchanges serve as marketplaces where users can buy, sell, or trade cryptocurrencies and tokens, bridging the gap between users, blockchain networks, and real-world financial systems. They integrate with multiple components of the blockchain infrastructure to provide liquidity, accessibility, and functionality. There is applicability for the exchange concept with voting as well. For example, an exchange could be used for aggregating and reconciling data or value. In blockchain voting, an "exchange-like" mechanism could refer to the process of collecting, verifying, and tallying votes in a transparent, decentralized manner.
6. **3rd party connectivity –** particularly via API – also not on the diagram, but 3rd party software interacts with many components of the blockchain ecosystem. Blockchain voting companies make use of third-party integrations for off-chain processes like voter authentication, data storage, or user interfaces

So now you have a basic understanding of how blockchain works and some of the moving parts associated with a blockchain ecosystem. Of course what I have explained so far is very high level and there are many permutations. But having a basic understanding of how blockchain ecosystems work will help you to understand some of the security issues. Because, yes, it is now it is time to put your security hat on. As a security person, I look at the blockchain ecosystem and ask a very simple question - “what could possibly go wrong?” We will now start to answer that question.

I will spend time discussing some of these issues with blockchain and in particular some of the issues with using blockchain for voting. The first set of issues relate more to general IT/architectural and privacy concerns as opposed to security risks although the inflexibility can make it difficult to fix security issues when they are discovered. These demonstrate that not all blockchains are created equal and how it is implemented can have an impact and subvert the utopian vision for how a “pure” blockchain is supposed to operate:

1. **Privacy issues**: One of the hallmarks of blockchain is its transparency. This means all transaction data, user/owner data is public and fully visible for all to see. This makes the use of blockchain for commercial banking applications as well as for voting problematic because this compromises the privacy of the individual. For voting, it would expose who voted for what or whom in any given election. Privacy-preserving solutions like zero-knowledge proofs and homomorphic encryption can help, but they're complex and resource-heavy and the administrators who manage the technology can be risks unto themselves. Privacy requirements, which are absolutely necessary for voting, can start the slippery slope towards private or permissioned blockchains (see #6 below) which leads to more centralization, more power in the hands of a few, and less transparency.
2. **Scalability and Cost:** Peer-to-peer consensus especially for proof of work demands heavy computing resources, making blockchains slow, difficult to scale, and expensive. Proof of Stake and its derivatives is more scalable because it is less resource intensive. This is inherent to how blockchains operate and when you are talking about making this work for a nation's voting, this can be a significant hurdle to overcome. This can drive a move to proprietary or less resource intensive methods of achieving consensus which can in turn lead to more centralization. See 9c and 11 below for more information.
3. **Inflexibility:** Updating blockchain software or modifying protocol rules is inherently difficult, often posing risks to security or requiring centralized coordination, both of which undermine the decentralized ethos of blockchain. Addressing flaws frequently necessitates changes to consensus mechanisms, which can conflict with the principle of immutability. Disseminating updates across nodes requires time-consuming consensus and can be disruptive. Soft forks (backward-compatible updates allowing unpatched nodes to operate) and hard forks (non-compatible changes requiring all nodes to upgrade) may lead to chain splits. Similarly, deployed smart contracts are difficult to modify for the same reasons—immutability, decentralization, and the need for network-wide agreement. This is how blockchain works so it will impact all blockchain implementations, cryptocurrency, financial services, as well as voting.
4. **3rd party risks:** The blockchain space, in general, is a mix of both private and public companies, each playing distinct roles depending on their goals and the type of blockchain they're working with. But for voting companies that claim to be using blockchain, all of them are private companies. These companies often operate with opaque ownership structures and undisclosed investor affiliations, raising questions about accountability and potential conflicts of interest. Sound familiar? While some claim to open-source their platforms, most rely on proprietary components that obscure vulnerabilities and resist independent scrutiny. Additionally, these systems frequently integrate with off-chain infrastructure—such as cloud services, databases,

or identity verification tools—broadening the attack surface beyond the blockchain's cryptographic protections.

5. **Opensource Status:** Open source is a cornerstone of public blockchain ecosystems, offering transparency, community-driven development, and auditability. Projects like Ethereum, Polkadot, and Hyperledger Fabric publish their core codebases under permissive licenses, enabling anyone to inspect, contribute to, or build upon them. This openness fosters innovation, decentralization, and trust—especially in permissionless networks where no central authority guarantees integrity.

   However, open source is not a security guarantee. While peer review and public scrutiny can strengthen code, vulnerabilities still occur, and not all projects receive equal attention in terms of security testing or auditing. Foundations vary in governance and quality control, making security outcomes inconsistent.

   In the voting space, the picture is even murkier. Although several companies claim to use blockchain, most are privately owned and only one has meaningfully open-sourced its code. These platforms often rely on proprietary overlays and third-party integrations, limiting transparency and increasing the risk of hidden vulnerabilities. See Appendix A for more information.

   So out of these **six prominent blockchain voting companies**, **only one—Follow My Vote—has meaningfully open-sourced its code**. The rest either keep their code proprietary or offer limited transparency, which raises concerns about auditability and trust, especially in democratic contexts.

   There is also a dark underbelly to open source. If you are interested in getting more information on this topic, which is beyond the scope of this article, go to Resources on the SCRepublicanAssembly.org website and read my article on this topic.
6. **Architecture and Implementation choices can significantly impact the blockchain:** The way a blockchain is architected—its consensus model, network structure, and permissions—can significantly affect both its security and its adherence to decentralization and transparency. Decisions around consensus (e.g., difficulty of proof of work, validator selection in proof of stake or authority) can be gamed to shift power to a few and undermine trust. Likewise, choosing between public, private, or hybrid models impacts transparency: private or permissioned blockchains (permissionless being the ideal where everyone can write to the blockchain), while faster and more scalable, concentrate control among a select group—creating "hidden centrality" and weakening the peer-to-peer ideal.

   Even more critically, some blockchains are centrally managed rather than truly distributed, diluting the foundational principle of trust minimization. And as with any digital system, the blockchain's security is only as strong as its weakest link—often found in the supporting infrastructure, not the protocol itself.

Let's look at a real life example. In 2024 Romania used the EU blockchain called the European Blockchain Services Infrastructure (EBSI) for vote monitoring and validation. Romania's blockchain ecosystem, particularly the MultiversX (EGLD) blockchain, might have played a role also but that is unclear. EBSI is a network spanning 27 EU countries and uses a permissioned blockchain (centrally controlled with only trusted nodes with write access to the blockchain) and was most likely selected to assist with scalability and security. It was not used for e-voting, but rather focused on recording and verifying voter turnout data and vote counts in real time.

So for the Romanian election, a centrally managed blockchain was selected whose nodes were located in 27 different countries. Interestingly, there has been penetration testing done on EBSI, but this was done by a Ukrainian company which I find very interesting. The system's reliance on existing IT infrastructure (SIMPV, SICPV) means it's only as secure as its weakest link. Device-level vulnerabilities or human errors at polling stations could still compromise data before it reaches the blockchain.

Do you think it is likely that nation states will implement a truly distributed, transparent and immutable blockchain for voting? Or will they opt for a centralized model that they can control while saying "oh it's blockchain, it's hackproof – trust us"? How you implement and the architecture that is chosen can make all the difference.

Let's move on now to the more security specific issues. These describe specific areas where blockchains or software/hardware associated with blockchains can be attacked. Real life examples are provided as appropriate.

7. **Key Management and Wallets:** Blockchain's security fundamentally depends on cryptography to manage identities and preserve the integrity and immutability of the ledger. While quantum computing and AI advancements raise long-term concerns about the viability of algorithms like SHA-256, no practical compromises have been observed to date. In the near term, however, attackers focus not on breaking cryptographic algorithms, but on exploiting how cryptographic keys are generated, stored, and exchanged—areas that are far more vulnerable.

   Stolen or compromised keys can undermine identify verification, integrity and immutability, enabling unauthorized transactions or fraudulent votes in the case of voting systems, to be appended to the blockchain. Phishing remains a common method for key theft, while wallet compromises—through malware, insecure storage, or social engineering—pose a persistent threat. Although wallets interface with the blockchain, they are external tools and often represent the weakest link in the security chain.

Other key management attack vectors include poor entropy in key generation, insecure key backups, lack of hardware security modules (HSMs), and inadequate multi-signature or threshold schemes. These vulnerabilities are especially critical in high-stakes applications like voting, where a single compromised key could jeopardize the integrity of the entire system.

Confirmed Attacks:

- **Bitfinex, August 2016 - $73M in Bitcoin stolen**: Stolen cryptographic private keys from exchange wallet infrastructure. The hackers were able to steal Bitcoin by exploiting a vulnerability in Bitfinex's multi-signature wallet software, which allowed them to take control of the recovery system for certain accounts. The hackers then moved the funds to separate addresses not associated with Bitfinex, effectively stealing them.[iii]
- **Liquid, August 2021, $97M in various cryptocurrencies stolen:** Compromised hot wallets due to security breach. The hack was most likely caused by the hacker gaining access to the private keys of the warm wallets of the exchange. Warm wallets are similar to hot wallets as they are deployed on an internet-connected endpoint and are used to manage liquidity. [iv]
- **Bybit, February 2025, $1.5B stolen**: The attackers, attributed to North Korea's Lazarus Group, executed a phishing attack against a 3rd party developer associated with Safe Wallet, a widely used multi-signature wallet platform that Bybit relied on for its Ethereum cold wallet. They injected malicious code into the Safe wallet front end and were able to trick the signers (transactions required multiple signers) into approving fraudulent transactions which contained malicious code that transferred control of the wallet to the attackers, allowing them to drain approximately $1.5 billion of cryptocurrency. [v]

8. **Networking and Peer-to-Peer (P2P) Layer Issues:** Blockchain nodes communicate via peer-to-peer networks, and vulnerabilities in the network and how it functions can enable denial-of-service (DoS) attacks (knocking the nodes off the network), eclipse attacks (isolating certain nodes from the network), or Sybil attacks (creating fake nodes). These can lead to double spending (spending the same cryptocurrency or blockchain token more than once**)** and 51% attacks (single entity or group controls

more than 50% of the network's computational power (in proof-of-work systems) or staked tokens (in proof-of-stake systems), allowing them to manipulate the blockchain's consensus mechanism). For voting systems, if one entity could control what transactions/votes are valid, there are obvious issues. Here are some examples:

Confirmed Attacks:

a. **Bitcoin Gold (BTG) 51% Attack (2018), $18M stolen -** Attackers created a large number of fake nodes or controlled significant mining power through rented hash rate, effectively dominating the network's hash rate. This Sybil-like control allowed them to execute a 51% attack, where they mined a private chain and used it to double-spend approximately $18 million worth of BTG by reversing transactions on exchanges. While not a pure Sybil attack (as it relied heavily on hash rate control), the ability to spin up multiple nodes to propagate malicious blocks was a contributing factor.[vi]

b. **Verge (XVG) Blockchain Attack (2018**), **$1.75M stolen** - Attackers exploited Verge's multi-algorithm PoW system by creating numerous fake nodes to control a significant portion of the network's hash rate. By overwhelming the network with malicious nodes, the attackers manipulated the blockchain's timestamping mechanism, enabling them to mine blocks faster than legitimate miners (a technique called "timejacking"). This allowed the attackers to execute a 51% attack, reorganizing the blockchain to double-spend millions of XVG tokens.[vii]

c. Solana Network Outage (September 14, 2021), nothing was stolen but had significant market impact - The attack was a Distributed Denial-of-Service (DDoS) attack triggered during the launch of a new project on Solana's blockchain, specifically a decentralized exchange (DEX) Initial DEX Offering (IDO). Bots generated a massive spike in transaction volume, peaking at 400,000 transactions per second, flooding the network's transaction processing queue (the "forwarder queue"). The high volume of resource-intensive transactions overwhelmed validator nodes, causing memory exhaustion and crashes, leading to a 17-hour network outage as nodes struggled to process blocks. The attack exploited Solana's transaction processing mechanism, which lacked sufficient rate-limiting or prioritization to handle such a surge.[viii]

9. **Forking/Upgrade and Node Software Bugs:** Even when blockchain protocols are well-designed, the software that runs them—such as Bitcoin Core or Ethereum's Geth—can contain bugs that compromise integrity, availability, or security. Because blockchain upgrades require decentralized coordination and must preserve immutability, even minor implementation flaws or missteps during forks can introduce serious risks.

When considering voting systems, these can be even more catastrophic. For example, a split chain could lead to conflicting voting records or fragmented consensus where validators can't determine which chain is correct so there could be multiple "official" outcomes. Additionally, votes cast on one chain could be replayed on another – double voting.

Confirmed Flaws or Attacks:

a. Software Bugs in Node Clients

- **Bitcoin BIP42 Bug (2015):** A flaw in Bitcoin's block validation logic could have allowed miners to exceed the 21 million BTC cap. It was resolved via a soft fork.[ix]
- **Ethereum Geth Memory Bug (2021):** This bug, discovered by Guido Vranken on August 18, 2021, could cause nodes to crash or fail to process blocks, potentially disrupting the Ethereum network. It was patched quickly but highlighted the fragility of client software.[x]

b. Protocol Upgrade Vulnerabilities

- **Ethereum Byzantium Fork (2017):** A bug in the state transition logic led some nodes to reject valid blocks, nearly causing a chain split. It was patched preemptively.[xi]

c. Consensus and Coordination Failures

- **Bitcoin Cash Hashrate Bug (2017):** A difficulty adjustment flaw made mining too easy, destabilizing the chain until an emergency patch was deployed.[xii]
- **Bitcoin SV 51% Attack (2021):** Following a contentious fork, BSV's low hash rate enabled a successful 51% attack, allowing double-spends and chain reorganizations.[xiii]
- **Bitcoin Gold 51% attack (2020), $70K stolen:** Bitcoin Gold (BTG) suffered two deep chain reorganizations on January 23–24, 2020, resulting in double-spends totaling over $70,000. The attacker removed and replaced blocks to reverse transactions, exploiting BTG's low mining cost and short confirmation windows on exchanges like Binance.[xiv]
- **Ethereum Classic 51% attack (2020) -** Ethereum Classic (ETC) was hit by three separate 51% attacks in August 2020, including a 7,000+ block reorganization on August 29. Attackers used rented hashpower to

outpace honest miners, enabling double-spends and chain instability. Exchanges like OKEx suffered millions in losses and considered delisting ETC[xv]

These real-world failures illustrate that dominance in consensus—whether through computing power, $$/stake, or validator approval—can allow attackers to rewrite blockchain history, censor transactions, or erode public trust. Even nation-states with large-scale resources could feasibly exploit these dynamics, especially in consensus mechanisms lacking robust decentralization or validator diversity. And I haven't even mentioned Artificial Intelligence and what that could potentially do to undermine the blockchain ecosystem. It's bad enough with just the humans.

d. **Forks and Fragmentation**

- **Bitcoin Cash Split (2020):** Disagreements over protocol governance led to the creation of Bitcoin Cash ABC (BCHA) and Bitcoin Cash Node (BCHN). The split fragmented the mining community, with BCHN receiving the majority of hash power. BCHA was left with minimal support, weakening its network security and adoption.[xvi]

NOTE: Most of the cryptocurrency giants like Ethereum and Bitcoin, have open sourced their code and do make use of bug bounty programs to find vulnerabilities before they go live. This is a way to mitigate the risks, but they can never be eliminated. And if the code is not open sourced and not properly security tested, then this can lead to successful attacks. Since most of the voting companies claiming to use blockchain have not open sourced their codes or been explicit about their security practices, then each of these areas will be significant attack vectors.

10. **Software Vulnerabilities in On and Off Chain Software:** Fundamentally, everything is code. While blockchain's foundational protocols are largely open source and peer-reviewed (but even these have been subject to successful attacks), the broader ecosystem—including smart contracts, dApps, and off-chain infrastructure—remains highly susceptible to software vulnerabilities. Smart contracts are self-executing programs deployed directly on-chain, but they rarely operate in isolation. Most decentralized applications (dApps) combine on-chain logic with off-chain components such as APIs, user interfaces, and databases, all of which expand the attack surface. Poorly written smart contracts can introduce critical bugs—such as reentrancy flaws, logic errors, or integer overflows—that are difficult to patch due to immutability. Off-

chain systems, meanwhile, may lack the transparency and auditability of blockchain code, making them attractive targets for attackers. Oracles, bridges, and APIs that feed data into smart contracts can be manipulated, and interconnected protocols can suffer cascading failures if one component is compromised.

Confirmed Attacks:

- **Cetus Protocol (May 2025), $220M stolen:** Hackers exploited a smart contract vulnerability involving fake token spoofing and arithmetic overflow, draining over $223 million from the Sui-based DEX. [xvii]
- **Poly Network (August 2021), $600M stolen:** The attacker exploited a flaw in Poly Network's cross-chain bridge, manipulating smart contracts to redirect funds to their own wallets.[xviii]
- **The DAO Hack (2016), $60M stolen:** A reentrancy bug in Ethereum's first major DAO allowed an attacker to recursively drain $60 million in Ether. The incident led to a controversial hard fork, splitting Ethereum into ETH and ETC. The DAO incident teaches us that while blockchain itself may be robust, vulnerabilities can exist at higher levels of the technology stack, such as in smart contracts and DApps. It also highlights the ethical dilemma of whether or not to intervene in a blockchain to correct a hack, raising questions about the true nature of decentralization and immutability.[xix]
- **dForce Protocol (2020), $25M stolen:** dForce, a DeFi DApp, was exploited via a vulnerability in the ERC-777 token standard, which allowed reentrancy attacks during token transfers. The attacker drained funds from the lending protocol.[xx]
- **Mt Gox (2014), almost $500M stolen.** The Mt. Gox hack, which took place in 2014, was not a breach of the Bitcoin blockchain but of a centralized exchange. Approximately 850,000 Bitcoins were stolen due to security weaknesses in Mt. Gox's systems.[xxi]
- **Ronin Network Breach (2022)** The attacker, after compromising a Sky Mavis employee via spear-phishing, used this lingering API access to extract a validator signature from the Axie DAO node. This API-based backdoor gave the attacker the final key needed to authorize massive withdrawals, bypassing the intended validator threshold. The breach wasn't just about stolen keys—it was about poor API hygiene and failure to revoke access and highlights how off-chain infrastructure, like RPC nodes and API permissions, can become critical attack surfaces in blockchain ecosystems.[xxii]

These incidents underscore a critical truth: blockchain security is only as strong as its weakest component. Securing smart contracts is essential—but so is hardening the off-chain infrastructure, how 3rd parties communicate with the blockchain ecosystem (APIs), exchanges, and DApps. All parts of the blockchain ecosystem and beyond are potential attack vectors.

11. **Consensus Challenges:** Most of the major attack vectors and examples were discussed in #9 Forking/Upgrade and Node Software Bugs, above, but there is another significant area of risk with regard to consensus. Consensus mechanisms are the backbone of blockchain security, yet each model—Proof of Work (PoW), Proof of Stake (PoS), and Proof of Authority (PoA)—has trade-offs that introduce unique risks. PoW is energy-intensive and favors those with access to specialized hardware, while PoS and PoA can centralize control among the wealthy or pre-approved validators. Proprietary consensus algorithms add further opacity, weakening auditability and trust.

    These mechanisms are vulnerable to **hidden centralization**, where a small group gains outsized influence over block validation, transaction ordering, or even protocol upgrades—signaling a shift away from true decentralization. Consensus-level flaws can lead to catastrophic failures such as double-spending, blockchain reorganizations, and censorship. For voting systems, a hidden centralization can be detrimental to the integrity of the system. One of the main selling points for blockchain is its decentralization and if that is undermined then all trust in the system will be undermined as well.

Final thoughts

If a blockchain ecosystem, especially one for voting, needs to sacrifice transparency, immutability, and decentralization in order to function, then it's not progress—it's camouflage. Slapping a fancy digital veneer over vulnerable systems doesn't solve anything; it just shifts the same risks into shinier packaging. Every added layer—be it smart contracts and other software, cryptography, architecture, or third-party API—multiplies complexity, inflates the attack surface, and chips away at trust.

Blockchain-based voting platforms aren't immune to these flaws. They inherit every risk from the systems they aim to replace—and invite new ones. The money in the current blockchain systems has already drawn attackers in droves, and voting systems would be no different. Wherever the votes are, attackers will follow. Complexity won't deter them; it invites them.

A voting process using paper ballots, hand marked, hand counted is the gold standard because it is simple, auditable and difficult to cheat at scale. There is no technological substitute for it because anything electronic is vulnerable to manipulation and attack. Making the decision to retire the vulnerable electronic systems and move to a voting process that is based on the gold standard of paper ballots is the only way to win back the trust of the population.

Appendix A:
Voting Companies' Information

Company Name	Private Company?	Open Sourced Code?	Uses Smart Contracts?	Uses Offchain Infra?	Uses Dapps?	Uses 3rd Parties?	Uses Wallets for Keys?	Which Consensus Mechanism Used?	Used in Gov Elections or Pilots?
Voatz	Yes	No	No	Yes	No	Yes	No	Proprietary (PBFT-inspired)	✅ Used in West VA, Utah, Col
Follow My Vote	Yes	Yes	Yes	Unclear	Yes	Unclear	Yes	Unclear	❌ No confirmed government pilots
Polys	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Ethereum-compatible / Proprietary	✅ Used in Russia and Poland pilot elections
Agora	Yes	Unclear	Yes	Yes	Yes	Yes	Yes	Custom voting-specific consensus	✅ Used in Sierra Leone pres election pilot
Luxoft	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Hyperledger Fabric (PBFT-like)	✅ Used in Zug, Switz municipal pilot
Polyas	Yes	No	No	Yes	No	Yes	No	Proprietary	✅ Used in German party and university elections
DemocracyGuard	Yes	Unclear	Yes	Yes	Yes	Yes	Yes	Unclear	❌ No confirmed government pilots; academic prototype

Note: " ✅ " indicates documented use in official government elections or pilot programs. " ❌ " means no confirmed use in government settings, though some may have academic or private-sector trials. Source: Copilot and not verified by me.

[i] A digital wallet is software or hardware tool that allows users to store, manage, and interact with their digital assets, such as cryptocurrencies or tokens, on a blockchain. It primarily handles the cryptographic keys—private keys for signing transactions and public keys for receiving funds—that enable users to access and control their assets on the blockchain. Digital wallets don't store the actual coins or tokens (which exist on the blockchain ledger); instead, they store the keys that prove ownership and allow users to send, receive, or manage those assets.

[ii] "A Merkle tree, also known as a hash tree, is a structure used in blockchain to efficiently and securely organize large amounts of data. It consists of a root, leaves, and the raw data itself, arranged in a tree-like format." Security Challenges with Blockchain, by Chintan Dave, page 26

[iii] The 2016 Bitfinex Hack: A Comprehensive Analysis, https://senshi.cc/posts/bitfinex/

[iv] Cyber Incident Victim: Liquid, https://www.csidb.net/csidb/incidents/4f5b70f9-8d69-460a-8db1-a360e28de58e/ and Hack Track: Analysis of Liquid Global Security Breach, https://www.merklescience.com/blog/hack-track-analysis-of-liquid-global-security-breach

[v] In-Depth Technical Analysis of the Bybit Hack, https://www.nccgroup.com/us/research-blog/in-depth-technical-analysis-of-the-bybit-hack/

[vi] BTG users lose millions in a 51% attack, https://dn.institute/research/cyberattacks/incidents/2018-05-19-bitcoin-gold/ and Bitcoin Gold Hack Shows 51% Attack Is Real, https://www.investopedia.com/news/bitcoin-gold-hack-shows-51-attack-real/

[vii] Verge Cryptocurrency Network Falls Victim to Same Attack Even After Hard-Fork, https://www.bleepingcomputer.com/news/security/verge-cryptocurrency-network-falls-victim-to-same-attack-even-after-hard-fork/ and Verge suffers a 51% attack in April 2018, https://dn.institute/research/cyberattacks/incidents/2018-04-04-verge/

[viii] How Blockchain DDoS Attacks Work, https://www.halborn.com/blog/post/how-blockchain-ddos-attacks-work

[ix] BIP-42 and Bitcoin's Fixed Monetary Supply, https://ericscrivner.me/2018/07/bip-42-and-bitcoins-fixed-monetary-supply/

[x] High Memory Usage v1.10.12-stable #23907, https://github.com/ethereum/go-ethereum/issues/23907

[xi] Release notes for the patched Geth client that fixed the EIP 211 bug, ensuring compatibility with Byzantium rules, https://github.com/ethereum/go-ethereum/releases/tag/v1.7.2/

[xii] The Bitcoin Cash Difficulty Adjustment Mechanism, https://blog.bitmex.com/wp-content/uploads/2017/11/2017.11.16-The-implications-for-Bitcoin-of-the-new-bitcoin-cash-difficulty-adjustment.pdf

[xiii] Bitcoin SV rocked by three 51% attacks in as many months, https://cointelegraph.com/news/bitcoin-sv-rocked-by-three-51-attacks-in-as-many-months

[xiv] Bitcoin Gold Blockchain Hit by 51% Attack Leading to $70K Double Spend, https://cointelegraph.com/news/bitcoin-gold-blockchain-hit-by-51-attack-leading-to-70k-double-spend

[xv] Hackers Launch Third 51% Attack on Ethereum Classic This Month, https://decrypt.co/40196/hackers-launch-third-51-attack-on-ethereum-classic-this-month

[xvi] Bitcoin Cash Hard Fork: Here's What Happened, https://decrypt.co/48409/bitcoin-cash-hard-fork-heres-what-happened and Bitcoin Cash Split 2020 | BCH Hardfork, https://guarda.com/academy/blockchain/bitcoin-cash-split-2020/

[xvii] How $220M was stolen in minutes: Understanding the Cetus DEX exploit on Sui, https://cointelegraph.com/explained/how-220m-was-stolen-in-minutes-understanding-the-cetus-dex-exploit-on-sui and Cetus Protocol $220M+ Exploit Explained: Token Spoofing & Overflow Attack on Sui, https://www.ccn.com/education/crypto/cetus-220m-fake-token-smart-contract-hack/

[xviii] Strengthening dApp Security: Essential Practices and Real-World Examples, https://auditfirst.io/blog/strengthening-dapp-security-essential-practices-and-examples

[xix] Examining Real-world Cases: Has Blockchain Technology Ever Been Successfully Hacked?, https://techbullion.com/examining-real-world-cases-has-blockchain-technology-ever-been-successfully-hacked/

[xx] Strengthening dApp Security: Essential Practices and Real-World Examples, https://auditfirst.io/blog/strengthening-dapp-security-essential-practices-and-examples

[xxi] Examining Real-world Cases: Has Blockchain Technology Ever Been Successfully Hacked? - TechBullion

[xxii] Back to Building: Ronin Security Breach Postmortem, https://roninchain.com/blog/posts/back-to-building-ronin-security-breach-6513cc78a5edc1001b03c364