Survey of Systems for Distributed Consensus

Outline: We describe the problem of distributed consensus. We define the notion of Open Distributed Systems and Closed Distributed Systems and discuss the security implications of each. We explore a toy example of consensus in a Closed Distributed System and the failure modes.

We overview the Bitorrent protocol and show that the convergence of Bitorrent in the presence of adversarial nodes depends on the existence of a preimage attack resistant compression function.

We overview the Bitcoin protocol. We show that the convergence of the Bitcoin protocol results from defining a total ordering over prospective elements in the decision set.

We examine the security properties of the Bitcoin network. We postulate that a secure system should require multiple dollars in resources to attack, for each dollar in network operating cost (the security ratio). **We calculate the cost to attack the Bitcoin network as a function of the resource cost to operate the network.** We outline a Bitcoin attack scenario and estimate the resources required for a successful attack. We show the cost to attack the Bitcoin network is upper bounded by one half of the operating cost of the network. We outline likely attack scenarios and show that an economically profitable attack on Bitcoin will be large, unexpected and mostly affect Bitcoin institutions rather than the user base.

We examine the incentives motivating an economically profitable attack on the Bitcoin system and we examine the factors which increase and decrease the cost to attack the Bitcoin network

We show:

- the profitability of an attack on Bitcoin is significantly enhanced by the development of a liquid Bitcoin options markets
- Increasing concentration of hashing power into mining pools decreases the resources required for an attack
- KNCminer was able to produce ASICs which obtained majority hash rate for less than 2 million dollars. The emergence of ASICs has decreased the level of resources required for a successful attack.
- cloudhashing decreases resources required for successful attack
- The ability of attacker to buy mining pools significantly reduces resources required for successful attack. Competition between mining pools for users is driving down mining pool fees to zero. Decreased pool profitability make buying an existing mining pool significantly cheaper than buying mining equipment.
- The concentration of hashing power in pools decreases the cost to obtain hashing power for an attack through hacking.
- A successful attack on Bitcoin, would lower Bitcoin price and further decrease the cost of subsequent attacks.

The resource costs to perform the attack are rapidly falling. The profits from a successful attack are increasing. Therefore a catastrophic double spending attack on Bitcoin is increasingly likely in the future.

However, we do not believe such an attack on Bitcoin will have any effect as a practical matter. If an economically significant 51% attack is demonstrated, it will affect dozens of Bitcoin institutions and cause hundreds of millions in damages. The developers would simply issue a patch and checkpoint the chain to deprive the attacker of unjust enrichment.

The very possibility itself however remains a mathematical imperfection in the protocol.

We examine the different notions of Proof of Stake and explore how they increase the resource level for an attack on the network. Proof of stake does not eliminate double spending attacks, but significantly increases the resource requirements compared to Bitcoin.

To obtain a more mathematically perfect system that eliminates the vulnerabilities in Bitcoin, we lay out some principles which we believe an ideal system should have.

We postulate

- The more profitable an attack compared to resources necessary to carry out the attack, the more likely the attack will occur.
- **The only safe dollar is a dollar that costs ten dollar to steal.**
- If a prospective attack yields larger profits than its resource cost, then it will eventually occur
- We postulate that a secure system requires more resources to successfully attack than the value being secured.
- A secure system should eliminate or reduce financial incentive for attacks. Security should therefore focus on precluding attacks with a profit motive over attacks which merely cause temporary network disruption.
- Small failures with limited scope are predictable, have known loses and are things that businesses and users can adapt to (wallet thefts, packages lost in mail, chargebacks, bounced checks). In contrast blackswan catastrophic events, such as resource intensive but highly profitable attacks are system instabilities which can cause billions in losses, the failure of a majority of institutions and permanently throw into question the foundations of trust and security of the system Therefore small failures should be minimized (such as wallet thefts), but catastrophic failures should not be permitted by the mathematical laws governing the system .
- Reduction in probability of attack success coupled with increase in resource commitment required for an attack increases deterrence against attack. An attack which

succeeds with probability 100% is more desirable to an attacker than an attack which succeeds with 50% probability. A high level of resource commitment and uncertainty about payouts from an attack attempt act as a deterrent.

- If resources used in an earlier attack cannot be reused in subsequent attacks, the incentives for an attack are decreased.
- If a successful attack does not reduce the cost of subsequent attacks, then the system is more secure
- Resources required for an attack should be intrinsically scarce.
- Resources required for an attack should have an egalitarian distribution. Higher level of resource concentration decrease the number of agents which must be compromised or in collusion to successfully attack the network. An attack requiring collusion between two hundred agents is more difficult than an attack requiring collusion between three agents.

We show that in Proof of Work system such as Bitcoin

- A successful attack destroys confidence, lowers price and lowers the cost of subsequent attacks. After a successful attack, security level is decreased.
- The hash power used in a single attack can be reused in subsequent attacks
- The 51% attack is an intrinsic weakness of all coins based upon proof of work

To eliminate the 51% attack and move beyond Proof of Work, we create a system based upon a web of trust network which better achieves our mathematical ideal.

The proposed system consists of a graph of nodes with trust relationships with other nodes involved in a negotiation process to determine consensus in a block chain. Social relationship graphs have several interesting properties that make them more secure for Bitcoin like systems than Proof of Work schemes.

- Social relationships and influence in a web of trust graph are intrinsically scarce
- Web of trust graphs mirror real world social and economic relationships

- Peers abusing influence to attack the network cannot use this influence in future attacks because abuse naturally results in destruction of the resources (trust relationships) needed to attack the network.
- The concentration of influence is unequal between nodes, but significantly less concentrated than found in Bitcoin. Instead of requiring collusion between three Mining pools, a successful attack requires collusion among hundreds of community members and stakeholders. The more diffuse power is among institutions and individuals, the greater the effort required to organize an attack.
- Influence is public and self limiting in a way that hash power accumulation is not. Stakeholders can individually and collectively influence network relationships to maintain balance of power between network actors, which is simply not possible in a Proof of Work system.
- The subset of nodes which can successfully attack the network can be calculated. The concentration and distribution of resources needed to attack the network is public.
- Network influence and the decisions of actors are public, allowing greater introspection and public accountability. In theory this allows 51% attacks to be detected and ignored with high probability, even if orchestrated by a vast super majority of nodes in the network

The implementation of the proposed system is called Obelisk and is outlined in another white paper. A coin built upon Obelisk, called Skycoin is also introduced.

Obelisk reduces the ability of a 51% attack in several ways

- increasing the resource cost for a successful attack
- ensuring an egalitarian, distribution of the resources required for an attack
- making attack success rates probabilistic and extremely small, even if attack is originated by a vast super majority of nodes

-- START PAPER --

The Byzantine Generals:

The Byzantine Generals problem comes from a thought experiment on communication protocols and the difficulty of achieving agreement over faulty communication channels. N generals are sieging a city. The generals can only communicate with the overs using messages sent by courier. Every general must attack on the same day. If general A attacks on day 1 and general B attacks on day 2, then the siege fails.

- Messages sent may not arrive
- Messages sent may arrive out of order
- The generals are honest and intend to reach consensus

This is a model of distributed systems composed of unreliable, but benign components. We call these **Closed Distributed Systems**.

In the Adversarial Byzantine Generals some of the generals are working for the enemy. They want the seige to fail and will do everything possible to be disruptive in any way possible. In this case, the objective is get all the **honest generals** to reach the same consensus.

- dishonest generals can collude
- messages may not arrive
- dishonest generals may say one thing to one person and another thing to another
- dishonest generals may lie
- dishonest generals may forge signatures of other generals
- dishonest generals may intercept and modify messages from other generals.

This is a model of systems where the system is actively under attack by its components. We call these **Open Distributed Systems**. This is the appropriate model for Bitcoin and other systems we are interested in.

Consensus in Distributed Systems:

- Distributed System: a network of processes sending messages to each other over a communication channel.
- Process: a computer program, which receives messages, processes messages and sends messages to other processes (state, computation, communication).
- Communication channel: a thing that moves messages between nodes
- Consensus: property in a distributed system where all processes arrive at the same state.

In the standard Byzantine Generals Problem,

- processes are benign
- processes may be running on unreliable hardware (power outages, hardware failures)
- processes may be running on faulty computer equipment (incorrect outputs for a computation)
- communicate on an unreliable communication channel

This is a situation faced by companies such as Google, running distributed programs on their own hardware.

- If Google only has one copy of your Gmail data on a single server and the server fails, the data is lost.
- For redundancy, one copy of the data may be stored on three different servers
- It is impossible to update all copies at the exact moment and the copies of the data will diverge

Real World Example:

- Three servers, A, B, C try to keep up to date copy of your emails

- Email X comes into server A
- Email is forwarded to server B and C, B stores copy
- Server B goes down because a transformer in the data center exploded and does not receive email X
- Server B comes online
- Email Y arrives at server B
- Server B sends copy of email to Server A and C.
- Server C does not receive message because a fishing trawler cut an undersea fiber optical cable which transports a third of the internet bandwidth between North America and Europe.
- Server A's hard drive fails and loses all the messages
- Server C receives copy of Email Y from server B

Result:

- Server A has no messages (hard disc failure)
- Server B has email X and Y
- Server C has email Y but not X
- A user queries their email data and depending on which server the request is routed to, will receive a different response.

If each hard drive fails each day with probability 0.3% and you have 100,000 hard drives then 14 drives will fail each hour. In a large distributed system every failure that can happen, will happen.

Consensus is the property that all processes should eventually arrive at the same state despite these failures.

A simple consensus algorithm is

- each server asks every other server which emails they have
- the server will download any emails it does not have

Closed Distributed System:

- You trust the hardware and the program.
- System has to tolerate failures and unreliability but failures are benign.

Open Distributed System:

- Different people control the hardware and the program. Other nodes cannot be trusted.
- Malicious nodes are actively attempting to exploit the system.
- Malicious nodes are highly intelligent, highly motivated and well financed
- Malicious nodes can send any data to other nodes.
- Malicious nodes can collude.
- Malicious nodes can run any program with any modification
- If network disruption can be caused cheaply, they will do it for fun (or for profit)

A **Closed Distributed System** corresponds to services running on secure trustworthy, owner operated hardware (such as data stored by companies such as Google). Failures are benign.

Consensus in Closed Distributed Systems is tractable and achievable by algorithms such as Paxos.

An **Open Distributed System** corresponds to services such as Bitcoin and Bitorrent, where anyone is able to join and operate the network. Failures are the result of active and purposeful attack by malicious network participants. Malicious network participants will perform any effective attack that is not resource intensive. Malicious network participants will perform any attack which is profitable regardless of resource requirements. Any attack not ruled out by mathematics or deterred resource requirements and incentive, will occur eventually.

Practical algorithms for Consensus in Open Distributed System is a very recent development. Bitorrent and Bitcoin are the two best examples.

Overview of Distributed Consensus in the Bitorrent Protocol:

Consensus Condition:

- At the end of the process each peer should end up with the same file

Assumptions:

- At least one peer has the whole file
- A network of peers have a communication channel between each other
- Each peer has a list of hashes for chunks (segments of the file).
- A malicious peer can send any data

Algorithm:

- peers ask others for what chunks they have
- peers download any chunk they dont have
- peers hash the chunk and ban the peer if the data received does not have the correct hash (indicating corruption or bad data sent by a peer)

The consensus condition is only achievable because inverting hashes is difficult. A user cannot easily produce a chunk that has the correct hash (verification condition). If a user could produce a byte string with a given hash, the user could produce invalid chunks and the network would replicate them. Different peers will have different files at the end of the process.

Convergence rests upon the existence of a pre-image attack resistant compression function.

Overview of Distributed Consensus in the Bitcoin Protocol:

Consensus Condition:

- Each node in network arrives at the same blockchain

Assumptions:

- There are a series of blocks, with each block depending on the previous
- Anyone can create blocks. “minting” or “mining”. There is nothing preventing different people from creating different successor blocks for the same block. Each block can have multiple children (successors) but only one parent. Therefore the blocks form a tree.
- Blocks in middle of a chain cannot be modified without invalidating blocks later in the chain. Each block depends on the hash of the previous block and modifying the block would modify the hash.
- Blocks are difficult to create (require costly hashing operations), A block is only valid if its data hashes to a value less than some value. Only 1 in N random blocks will have a hash less than the number. The quickest way to create a valid block is randomly hashing until a valid block is produced.

Algorithm:

- Each node computes the cost or difficulty necessary to have produced each chain in the tree of blocks. The more hashing operations were required to produce the chain, the highest the difficulty. A chain with more “depth” (more blocks in the sequence) will require more work to produce on average than a chain with fewer blocks.
- Nodes query each other and sequentially download blocks of the “more difficult” or “longest” chain their peers know about.

There is a blocktree and Bitcoin assigns a value or “Difficulty” to each respective chain in the blocktree. Bitcoin says the blockchain with the highest “difficulty” is the consensus chain. The consensus chain propagates between peers until every peer has a copy of the chain.

Bitcoin assigns a numerical value or total ordering to choices in the consensus set and then propagates the choice that maximizes the value assignment. That is the mathematical basis of Bitcoin convergence.

Bitcoin convergence therefore relies on several assumptions

- A total ordering on choices for consensus. Difficulty required to produce ian“honest” signal of work or effort. This is called Proof of Work (total ordering, is costly)
- a new block cannot be produced with the same hash as an existing block (pre-image attack resistant of hashing function)

Bitcoin achieves convergence but has the following properties

- The person who controls 51% of the hashing power determines the consensus chain
- The subgroup of people who together control 51% of the hash power through collusion, determine the consensus
- The next consensus chain does not have to be a longer version of the current consensus chain. The chain can be “forked” at an early block.

This means that Bitcoin becomes more secure the more distributed hashing power is among the user base and less secure the more concentrated hashing power becomes.

- It is much easier for two people controlling 21% and 51% to collude to attack the network, than for five people controlling 10% each to collude.
- Collusion between 51 people with 1% of total hashing power is more difficult than collusion among 5 people with 10%.

A person accumulating 51% of hashing power can easily do so in secret by spreading this over multiple mining pools. A pool that grew too large, faces scrutiny and public backlash, can easily move hashing power so that it does not appear to the public to be associated with the pool.

Security of Bitcoin:

Bitcoin tokens have a direct monetary value and create a unique financial incentive for exploitation that non-monetary systems do not suffer from.

For monetary systems running on an Open Distributed System, we should add the following assumptions

- If stealing $1 costs less than $1 in resources, they will steal it.
- **The only safe dollar is a dollar that costs $10 in resources to steal**

Therefore an ideal system

- The cost to attack system, should be greater than the monetary value being protected
- The cost to attack system, should be significantly greater than financial reward for doing so

In Bitcoin there are several attacks which can be highly profitable if a user can orchestrate majority control of hashing power, even for a short time period

- An attacker can buy put options on Bitcoin and engage in a double spending attack on a large exchange. The resulting price shock would produce extremely large rewards to the attacker. This is becoming an increasingly profitable threat as options markets develop.
- An attacker can deposit money in a bank, withdraw the money and then use the double spending attack to revert the deposit transaction. They now have the withdrawn Bitcoin and the bank does not have the deposited Bitcoin
- An attacker can deposit money in several gambling sites, withdraw the money and then use the double spending attack to revert the deposit transaction.
- An attacker can deposit Bitcoin into an exchange, buy Litecoin and then revert the deposit transaction. The attacker now has the Litecoin and the Bitcoin.

A realistic attack is likely to combine all of the above scenarios at the same time in a single large attack.

In Bitcoin there are several ways a person could obtain the resources necessary for an attack

- The two largest mining pools could collude and obtain majority hash rate
- The largest pool could secretly begin buying smaller pools. There have been a number of instances of mining pools with multiple changes in ownership. The cost to acquire a large pool decreases as pool fees approach 0% as pools compete for users.
- A hacker could compromise several pools and use the resources to orchestrate a double spending attack or a pool participating in attack could merely claim to have been hacked
- KNC Miner sold 200 mining units at $10,000 each, which achieved majority hash rate. Therefore the cost to attack network is less than 2 million dollars using specialized ASICS.
- The largest mining pool could continue its ASIC acquisitions and achieve majority hash rate while hiding the fact from the public.
- A large pool or attacker could rent the resources needed for a double spending attack by the hour from a Cloudhashing services. Renting the hash rate required for an attack by the hour significantly lowers the costs of a 51% attack which requires buying specialized equipment.
- A decrease in Bitcoin price could result in a decrease in mining difficulty, increased availability of second hand mining hardware and availability of cheap rentable hashing power

A realistic attack combines a large pool colluding with other pools, hacker attacks on smaller pools, renting cloud hashing capacity and using purchased ASICs.

The market cap of Bitcoin is approximately 6 billion, but based upon the KNC Miner data, the cost to attack the Bitcoin network is 2 million using commodity mass market hardware. KNC Miner outsources both the design and the production of their ASICs.

The cost to attack the Bitcoin network depends only on the cost to obtain majority hash rate. If the rewards from an attack are greater than the cost of the attack, the network is likely to be attacked.

The security of the Bitcoin network does not depend upon the number of users of Bitcoin, but only on the cost to achieve a majority hash rate. For every transaction processed on the Bitcoin network, miners currently receive a block reward subsidy of approximately $25 in newly created Bitcoin.

If Bitcoin miners spend $100 million/year on mining costs, the cost to achieve majority hashrate on the Bitcoin network (for an outsider with no existing hardware) is $51 million/year and approximately $100 million/year in new Bitcoin will be created to subsidize the miners. This means that the security multiplier for Bitcoin is less than 0.5. Two dollars need to be spent on mining costs to secure the network against one dollar in attack.

This is for a direct attack, using new mining hardware and is overly optimistic. A person does not need to own or buy the full amount of mining hardware required for an attack. It is much cheaper and more cost effective to buy out the management of a large mining pool and rent the remaining hashing power required.

Also, attackers only require majority hash rate for a few hours or days to achieve a successful attack. Miners must mine 24/7 to provide security. The costs for defense must be borne out for a much longer duration of time than the costs for an attack.

Given these considerations, Bitcoin’s security multiplier may be significantly lower than 0.5 . Bitcoin may require multiple dollar in mining cost per year to secure against a single dollar in attack resources.