ShopChain Distributed Ledger Technology (DLT): a Federated, Staked Consortium

Shopin Team
shopin_team@shopin.com

https://github.com/uniteddata/shopin_public

Abstract*—*ShopChain DLT (Distributed Ledger Technology) is a permissioned, federated network consortium with superior throughput and security, consisting of known staked participants. ShopChain DLT uses a ground-breaking architecture combining distributed hash tables (DHTs) and consensus nodes. Among other innovations, ShopChain DLT leverages BLS signatures to achieve throughput beyond Bitcoin and Ethereum. The BLS signature is an implementation of a digital signature scheme ideated by Boneh, Lynn, and Shacham [1]. BLS signatures have efficient computability and non-degeneracy, which means they are easy to use and difficult to tamper with, therefore secure.

***Index Terms*--shopchain, DHT, DLT**

I. Introduction

We describe a novel approach to a GDPR compatible blockchain architecture using distributed hash tables relying on proof of activity for consensus with reasonable security, survivability, and with provable decentralized governance, decentralized identity management, decentralized naming service, and scalable performance, that operates without degradation regardless of nodes geographic distance, while acting as a decentralized data exchange.

This approach ensures GDPR compliance by taking great care to store all personally identifiable data in a manner that is both mutable and able to be nullified through indexing with a distributed hash table that routes it between nodes that store the aggregated data in persistent storage outside of the immutable ledgers that power the naming service and audit logs.

Through this approach, the immutability of blockchain can provide proof of activity within the network by serving as a shared name service and acting as a log of auditable events where the exchange of data has taken place within the network. It is with the amalgamation of these proven decentralized technologies that we establish an enterprise-grade network capable of empowering users to have complete control of their own data.

II. Overview

This approach provides an enterprise-grade network to transport data between consumers, retailers, and wallet providers. We purposefully avoid inventing technologies whenever possible and instead opt to leverage our expertise in algorithms and networks to create a new type of distributed hash table network to act as an asynchronous data exchange. This technological leap is necessitated by the challenges facing well-established retailers and brands who are suffering from declining business trends and well-publicized data breaches. In the age of limitless online competition, these brands have failed to keep up technologically and have yet to find a strategy that provides sustained growth. Retailers are aware that personalization in their marketing and in the consumer experience yields revenue growth. Few retailers have succeeded in implementing it as they lack access to data on their consumers, due in no small part to the existing barriers to sharing data between retailers.

Our approach enables a consumer to own their data regardless of where it is created on the network. The consumer can elect for their data to be custodied by a third party (e.g. ShopIn.com), another third party on the network (e.g. Microsoft Azure), or to hold it on their own devices (i.e. Microsoft Project Bali) that are peered to the network. We seek to ensure that a consumer can determine where their data is stored, how it is accessed, and if they would like to see it monetized. While the data exchanged must be able to be forgotten, all of the activity that occurs to data stored in the solution will be tracked in our blockchain, creating an immutable audit log.

The decentralization of consumer data provides retailers the ability to create more personalized messaging to consumers. This is accomplished by returning ownership of data to consumers through a decentralized network. Consumers will have true freedom and optionality for where their data is stored, how it is accessed, and how that data is monetized. Retailers are going through an apocalypse. Decentralization enables personalization on a level that exceeds anything possible with a centralized system.

The decentralization of consumer data provides retailers the ability to create more personalized messaging to consumers. This is accomplished by returning ownership of data to consumers through a decentralized network. Consumers will have true freedom and optionality for where their data is stored, how it is accessed, and how that data is monetized.

III. A Federated DLT for Retail

One of the main arguments against the mass adoption of blockchain is its difficulty to achieve scalability. In fact, most distributed ledger platforms encounter great difficulties when trying to provide users with the same transaction processing capacity as legacy processors like VISA. Bitcoin, the most widely used platform for running transactions on blockchain at the time of writing, handles about 7 transactions per second (TPS) in throughput, while VISA handles an average of 1,600.

While VISA relies on a more traditional centralized design, in Bitcoin thousands of nodes are individually engaged to fully validate a single block of transactions. At first glance, it could be argued that one way to speed up such a system may involve decreasing the number of nodes required in the processing of an individual transaction. Usually the cost of fewer nodes is a decrease in security, nevertheless, as Shopin DLT is a permissioned, federated network consisting of known staked participants, we are able to achieve much greater throughput while maintaining the high degree of security necessary for the consortium.

IV. ShopChain DHT

The ShopChain DHT is a modified S/Kademlia DHT [2], which provides the network with additional security guarantees.

A Distributed Hash Table is a decentralized storage system that manages key-value pairs. Each node in the Shopin DHT is roughly responsible for an equal share of the distributed storage layer load. Customer data or even transaction data may be stored across the DHT nodes, with the data easily retrievable by authorized parties as needed.

In order to maintain a high degree of availability of the data in the storage layer, all data is replicated f+1 times (or 8 times), which means even if the network encounters a maximum failure threshold of f, data would still remain intact and accessible. Furthermore, the additional utilization of Erasure Codes can be leveraged in order to enable even greater data integrity, better availability, and quicker retrieval guarantees.

Erasure codes are Forward Error Correction (FED) codes that allow to rebuild given files into larger ones, and share them into a number n of chunks, each containing some pieces of information of the original file, which are distributed to the corresponding number of nodes in the system. To ensure safety, a number m of redundant blocks are also created for each file distributed via erasure coding [3].

V. ShopChain DLT Network System

In this first release of ShopChain (ShopChain DLT), we describe the critical synchronous components of the ShopChain platform, with the ShopChain DHT implementation, consensus, and network security and stability. The full ShopChain platform includes the asynchronous components of the network and will be discussed in future installments.

The Shopin Network consists of 21 professionally managed nodes in a Federated Consortium. In order to join the private network nodes must be invited and are thereafter required to deposit a stake using the Shopin Token. Staking provides additional security to the system, for a malicious node's financial stake can be slashed if caught acting against protocol. Moreover, this staking requirement also establishes a node's identity and public key to the network, which must be known in order to validate BLS aggregate signatures.

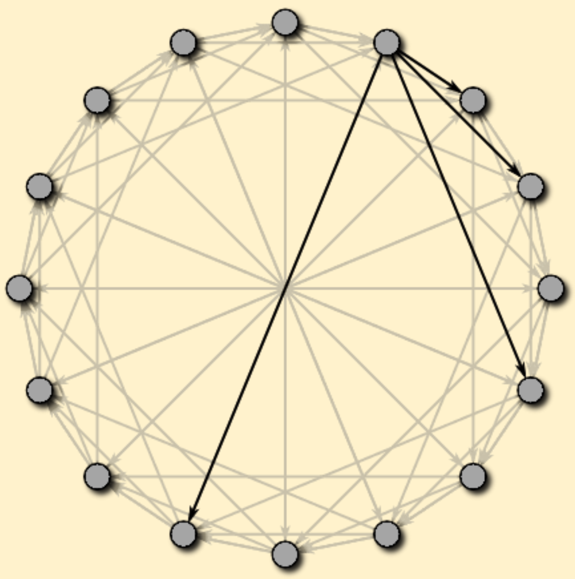
Figure 1. Example of a "Hash Ring" Topology

Nodes in our system are organized via consistent hashing into a ring structure, with each node taking turns in a round-robin fashion to be the next block proposer. Since this is a consortium of known actors, decentralized randomness is not required for the protocol. Instead, nodes take turns, rotating the leader position one block at a time.

The ShopChain DLT network consensus is achieved as follows:

Step 1: The leader node constructs and broadcasts a block of transactions to the network.

Step 2: Each of the nodes validates the block and sign the hash of the block with their unique digital signature and sends it back to the leader.

Step 3: The Leader waits to accumulate 2f+1 signatures for a supermajority and then combines them into a final BLS Aggregate Signature which is broadcasted back to the network.

Step 4: Individually nodes will verify that the aggregate signature is valid, and if it is, they will append the block to their local blockchain, and the protocol moves onto the next leader.

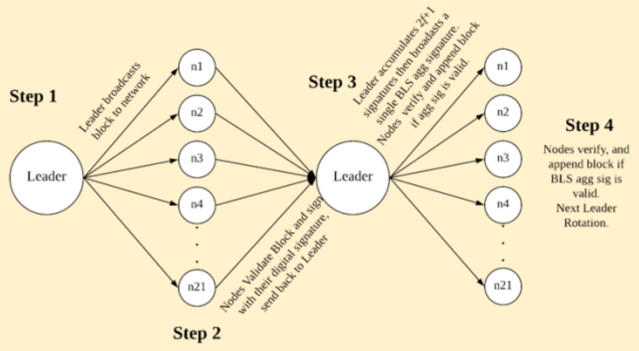
Step 1
Leader broadcasts block to network
Leader
n1
n2
n3
n4
n21
Step 2
Nodes Validate Block and sign with their digital signature, send back to Leader
Step 3
Leader accumulates 2f+1 signatures then broadcasts a single BLS agg signature. Nodes verify and append block if agg sig is valid.
Leader
n1
n2
n3
n4
n21
Step 4
Nodes verify, and append block if BLS agg sig is valid.
Next Leader Rotation.

Figure 2. ShopChain DLT Network Consensus

The proposed block must reference the prior block and nodes shall only sign once per block height. Only blocks that have 2f+1 and satisfy these 2 conditions are considered valid and can be appended to one's blockchain.

VI. A Primer on BLS Digital Signatures

Each node in the Bitcoin network, or similar Proof of Work platforms, individually validate any single block of transactions by means of issuing digital signatures. Bitcoin utilizes ECDSA (Elliptic Curve Digital Signature Algorithm) digital signatures, which are large and not fit for compressed aggregation [1] [4]. This implies that in order to validate a block all issued signatures are individually collected, and as a result, Bitcoin's higher bound throughput is directly affected by the method for collection/disbursement and verification of these digital signatures.
Due to the inconvenient size of these types of digital signatures, over the last decade, new methods have been developed to create digital signature schemes that allow for shorter aggregation of individual signatures, and consequently, faster verification. The most successful innovation at this time of writing is represented by the so-called BLS signature aggregation scheme, which is a recently updated implementation of a digital signature method ideated by Boneh, Lynn, and Shacham [1].
The idea is to shrink the signature size and verification times by compressing all the individual contributions into a new, aggregated output, i.e. one final cumulatively generated signature instead of a vector of individual ones.

The signature scheme is in turn defined by three functions:

- KeyGen() → Choose a random α and set h ß g_1(α) output p_k:=(h) and s_k:=(α)
- Sign(s_k, m) → Output σ ß H(0)m α belonging to G0. The signature is a single group element
- Verify(p_k, m, σ) → if e(g_1, σ)=e(p_k, H0(m)) output «accept», otherwise output «reject»

The main characteristics of a BLS signature are:

- Hash function: H0: M → G0

BLS signatures use an algorithm that hashes directly to the curve. Therefore, not all curves may be utilized for outputting a BLS signature, while only pairing friendly elliptic curves are available.

- Bilinear pairing function: e: G0*G1 → Gt

The bilinear property implies that multipliers may be exchanged without changing results, as follows: e(x*P, Q) = e(P, x*Q).
The above is key to the fast verification of correctness in BLS signatures, which checks that e(P, H(m)) = e(G, S).

BLS signatures owe their success to a number of reasons. First, they are characterized by the properties of efficient computability and non-degeneracy, which in short means they are easy to use but very difficult to tamper with, thus they are reliably secure.

Second, as already mentioned, BLS signatures may be aggregated. The aggregation of signatures in BLS is faster than in other methods (even faster than Schnorr signatures, another popular digital signature scheme) and allows for improved verification speeds as compared to ECDSA signatures [4].

Both the individual signatures and the individual public keys of single validators in a transaction may be easily combined into one, shorter piece of information, just by adding the relevant components up. Given triples (p_k_i, m_i, σ_i) for i = 1 , …, n, anyone can aggregate the signatures σ1,… σ_n belonging to G0 into a short aggregate signature σ by computing:
σ ← σ_1… σ_n ∈ G0

Verifying an aggregate signature reduces to:

e(g_1, σ) = e((p_k)_1, H0(m_1))… = e((p_k)_n, H0(m_n))

VII. Using BLS Aggregate Signatures to Achieve Consensus

As explained by Barbara Liskov and Miguel Castro in their paper “Practical Byzantine Fault Tolerance” from 1999, Byzantine Fault Tolerant (BFT) systems can operate without failure even with the existence of malicious nodes, so long as ⅔ +1 of the total nodes remain honest [5].

Suppose there are n = 3f+1 nodes in total, of which there is a maximum of f nodes that are faulty. In a system of 21 nodes, as long as at least 15 out of 21 nodes are honest (2f+1), the protocol will continue without affecting the required Safety and Liveness guarantees of the BFT protocol. The Safety guarantee promises that after the protocol executes and completes, all honest nodes shall agree on the same block, while the Liveness property guarantees that the protocol shall not stall indefinitely.

As discussed above, BLS signatures are particularly useful for signature aggregation, because they are quick to generate, easy to verify, of a constant size, and as a result, they are superbly suited for our use case.

VIII. Failure Handling and View Changes

In our design, safety is guaranteed because blocks are only appended when any combination of 15 of 21 (2f+1) nodes have cryptographically agreed that the block is valid. Thus, all honest nodes will contain the same data as each other. However, suppose a leader node malfunctions or is otherwise deliberately faulty, how to ensure that the protocol progresses and that the liveness guarantee is not violated?

A common approach that we also will leverage is a timeout mechanism, i.e. if after a specific period of time Δ and if consensus is still not achieved, a view change is triggered, and the protocol will move to the next leader. Block rewards may be generated and provided only to leaders that are able to successfully create a valid block that gets appended. The block rewards as well as a node’s initial stake are mechanics designed to provide the federated system additional security guarantees.

The ShopChain DLT handles view changes when there is a faulty leader as follows:

Step 1: View Change proposer broadcasts request to network seeking 2f+1 agreement.

Step 2: Nodes vote by signing the view change request if they agree or ignoring the request if they don't agree.

Step 3: Proposer generates an aggregate signature out of 2f+1 signatures on the view change request and sends it to the network.

Step 4: Nodes validate the aggregate signature is correct and then increment their view by one. The next leader is determined by a simple function: (block height l+1) mod n, where n is the number of known nodes in the consortium.

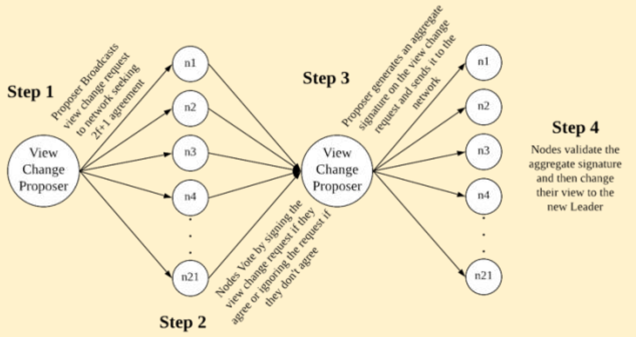
Step 1
Proposer Broadcasts view change request to network seeking 2f+1 agreement
View Change Proposer
n1
n2
n3
n4
n21
Step 2
Nodes Vote by signing the view change request if they agree or ignoring the request if they don't agree
Step 3
Proposer generates an aggregate signature on the view change request and sends it to the network
View Change Proposer
n1
n2
n3
n4
n21
Step 4
Nodes validate the aggregate signature and then change their view to the new Leader

Figure 3. View Change Requests

In the event that a leader is unable to receive 2f+1 signatures on a block that it proposed, the leader will fall back to a resynchronization procedure of asking its peers for the latest information until at least f+1 of peers respond back with the same state and data. The f+1 threshold is important here because it guarantees that the re-synced data proves integrity.

IX. DDoS and Spam Protection

A distributed denial of service attack, also known as DDoS, is the most common type of malicious attack in distributed networks. This attack is based on the triggered exhaustion of information resources of a circuit and causes malicious traffic to slow down or completely prevent any further message requests to arrive to a given predetermined location. In other words, this attack overwhelms resources to a point that servers cannot handle the requests and may shut down.

To a certain extent, a denial of service attack could be imagined as someone inducing a traffic jam on a highway, when there really is not any real congestion due to an uptick of traffic. What happens, in reality, is that an attacker pings a large number of your active servers with requests, often by leveraging a virus or some other form of malware which transforms other computers, servers, or nodes into a botnet under the attacker’s control.

The attacker then targets a given IP address and instructs all the nodes in the botnet to send requests to that destination. This suddenly increased traffic causes a denial of service, as the receiver is overwhelmed with requests and cannot distinguish between normal and attack traffic. As a result, the victim system gets exhausted and unstable with so many incoming requests and becomes unavailable for other legitimate users, whose messages get ignored or never arriving at the preset destination.

The cracker, i.e. the malicious actor, sends the viruses to a number of users who are completely unaware they have been intruded against, thereby turning honest nodes into malicious actors against their will. By the same token, the nodes attacked by the virus are unaware of having become malicious actors, and they involuntarily send out a cascade of requests that end up blocking the target's system.

One of the reasons why this type of attack is particularly severe is that the malicious action comes from a different number of sources, and therefore just blocking the traffic coming from one node will not be suitable to prevent a malicious attack to be carried out.

To a lesser extent, this type of attack can be referred to as spamming. Spamming may in fact not necessarily trigger a denial of service, however, it will result in spreading dust and unwanted information in the network, which will result in a slowing down of transaction processing.

To address this and grant both speed and finality, 9Shopin DHT adopts two methods. The first one, which is common to many distributed technology platforms, relates to the topic of transaction costs. In fact, most distributed ledger platforms attach a small fee to be paid by users in order to process their transaction. This method also makes it "expensive" for nodes to engage in a high number of transactions, and therefore discourages large-scale DDoS attacks. On November 6, 2019, Ethereum average transaction fee was registered to be 0.00067ETH, or $0.127, against a median transaction fee equal to 0.00038ETH, or $0.073. During the same day, Bitcoin's average transaction fee was registered to be 0.00013BTC, or $1.23[6]. However seemingly low, transaction fees apply to each transaction that is sent to run through the platform. Therefore, large-scale DDoS attacks may become expensive for a malicious actor.

The second defense against DDoS is to ignore requests that come in beyond an acceptable standard deviation and require a certain amount of time to pass before another request is accepted. Deploying both of these strategies will substantially limit the ability of a malicious actor to launch an attack.
In general, it is relevant to notice that, in platforms run via Proof of Work consensus algorithms, slashing is a powerful weapon to fight threats of malicious behaviors. Ethereum 2.0 [7], for example, has put in place a method for slashing nodes (i.e. retaining nodes' staked deposits), may these be caught in malicious behavior. The same technique is implemented by Polkadot [8]. In the same way, the Harmony [9] whitepaper claims that "for any misbehaviors detected by the network, a certain number of staked tokens will be slashed".

X. Sybil Attack Prevention

A Sybil attack is defined as a situation in which, in a peer-to-peer network, one individual node begins operating multiple other nodes in the network, therefore creating an increased grouping of seemingly individual malicious nodes.

The name was given to this type of attack in 2002 by Brian Zill, a Microsoft researcher, who noticed the great similarity between this type of event and the story recounted in the book Sybil, within which the main character suffers of a disorder causing her to have multiple dissociative identities.

This attack is extremely threatening for two main reasons. First, because it has the power of radically subverting the balance of the network's reputation system when the attacker achieves operating control of a number of malicious identities in excess of the threshold supported by the chosen consensus algorithm running the distributed ledger. Second, because blockchains and distributed ledger technologies operate by means of pseudonymity, i.e. operating active nodes in the network register on the platform with a pseudonym, which is not directly connected to any other document or tool defining the individual's identity in any other way, for example, a passport or a driver's license. Therefore, it is extremely easy for a single individual to operate multiple nodes concurrently on a multiplicity of devices, by creating one pseudonymous identity per device.

The severity of the threat posed by a Sybil attack differs depending on a number of factors. The first determinant is the consensus algorithm employed by a given blockchain platform. As a matter of fact, Proof of Work (PoW) and some Proof of Stake (PoS) systems are resilient to attacks up to 50% of the network. This implies that, if the user base of a given distributed ledger platform is sufficiently big, the attacker will need to operate concurrently an extremely high number of devices to succeed. Second, both consensus techniques aim at making it expensive to cheat. In fact, while in PoW systems the nodes with the highest computational power will be selected to be block proposers, PoS consensus algorithms privilege those individuals that stake a higher amount of money on the chain. Finally, PoS platforms slash the stake deposited by a given node when the latter is caught incurring in malicious behavior. As aforementioned, great examples of implementation of this technique are Ethereum 2.0 [7], Harmony [9], or Polkadot [8].

While these factors make it extremely expensive for individuals to launch Sybil attacks, they do not prevent them, and past history witnesses this fact.

Some recent examples of Sybil attacks are the following:

- During the latest elections period in the United States, one single identity was found to be interfering by means of a Sybil attack, operating a multiplicity of accounts on Facebook;
- Sybil attacks on the Tor network;
- The 51% attack in Blockchain networks.
- Multiple reviews on platforms such as Amazon left by a single individual operating multiple accounts,

with the purpose of subverting the rating of products and deviate sales.

In the case of networks built on Byzantine Fault Tolerance (BFT) consensus algorithms, the threat posed by Sybil attacks is substantially higher. In fact, such platforms are usually constructed to tolerate up to less than a third of the network being corrupted, which is a lower threshold to be passed by a given malicious actor. As Barbara Liskov and Miguel Castro (1999) explain [5], the Byzantine tolerance threshold supported by BFT consensus algorithms may be set to a number equal to the total amount of nodes active in the system, minus one, divided by three, or f=(n-1)/3, therefore roughly one third of the network. Second, differently from PoS systems, such consensus algorithms do not punish actors caught incurring in malicious behavior by means of slashing their stake deposited. In other words, BFT mechanisms do not make it expensive to cheat.

In order to tackle this problem and create a remedy to the threat posed by Sybil attacks, Shopin DHT, as a BFT-based consensus algorithm, puts in place a number of mechanisms. First, a reward and punishment mechanism together with a reputation tracking system is put in place in order to incentivize correct behavior and disincentivize malicious behavior. Specifically, honest nodes obtain a monetary reward and increase their reputational score every time a correct action is performed. The idea is similar to what implemented by a number of blockchain, which utilize consensus rewards to incentivize action. For example, Harmony [9] rewards nodes in proportion to their stakes. On the other hand, Ethereum [7] and Bitcoin [10] reward miners for solving complicated cryptographic puzzles. The higher the reputational score, the higher the monetary reward cashed in by the node. The reverse is triggered by a malicious action, whereby the node performing it will lose a part of their staked tokens, with its reputational score diminished. If a reputation score ever drops below a tunable system parameter, the node is blocked for a given amount of time. If the node reaches this lower bound multiple times, it is eventually banned from the system.

Second, Shopin DHT is working on a Decentralized Identity layer, in order to allow each individual organization to run at most one node. This is achieved by means of identity registration via innovative methods such as biometrics systems and individual knowledge. We hypothesize his method will eventually completely erase the risk of Sybil attacks.

XI. Churn Impact Reduction

The term churn in common language refers to the abandonment rate in a given context. In other words, it is a measurement that indicates how many individuals, or participants, within the same environment.

When applied to distributed ledger technologies, the term churn rates express the number of validating nodes that leave the system, i.e. go offline during a given amount of time. The reason why churn rate is such a relevant topic in blockchain systems is because of the threats it brings together with it. This is especially true when a Byzantine Fault Tolerance (or BFT) consensus algorithm is adopted.

A consensus algorithm is a formula that, in any distributed ledger technology, determines how blocks get added to the chain. The most common consensus algorithms as of today are Proof of Work (PoW), adopted by Bitcoin and the current version of Ethereum (also known as “Nakamoto Consensus”) and Byzantine Fault Tolerance (BFT) algorithms, which ShopChain will adopt.

BFT algorithms, as the name suggests, are designed to handle up to a given number of malicious actors in the system, while still allowing the transactions to be processed and blocks to be added correctly.

This threshold number is usually set to be the overall population of validating nodes in the system, minus one, divided by three.

This yields the maximum number that a BFT based blockchain design can tolerate and still work correctly. If the amount is exceeded, safety is no longer granted. While the threshold is lower in BFT algorithms, they have some very respectable qualities as compared to other consensus algorithms. For example, in a BFT-like scenario, under the random oracle model, every node has the same probability of being selected to validate a block, while PoS and PoW systems privilege nodes with either a higher amount staked or greater computational power. Furthermore, BFT based designs have the desired quality of instant finality, whereas in Bitcoin for example, finality is only guaranteed to a large degree after approximately 6 blocks or about 60 minutes.

It was mentioned above that the churn rate is a particularly relevant problem to be addressed in blockchain platforms that are built on Byzantine Fault Tolerance consensus algorithms. This is because this family of consensus algorithms is designed to tolerate at most less than one-third of the overall network of validating nodes. If this given number is exceeded, the blockchain in consideration can no longer grant a safety guarantee to its users.

What happens in practice is that in such a consensus mechanism, a number of nodes (or all of them) are sampled to come to a consensus to the validity of a transaction, or a block of transactions, that has been proposed by a leader node. If at least 2f+1, i.e. two times plus one in excess of the equivalent number of malicious nodes that the system could at most tolerate, reach consensus on the validity of a block of transactions, that is added to the chain.

The 1999 iconic “Practical Byzantine Fault Tolerance” by Miguel Castro and Barbara Liskov [5] describes an algorithm that “offers both liveness and safety provided at most [n-1]/3 out of n total of replicas are simultaneously faulty. In this context, therefore, if we assume that the total number of validating nodes on the blockchain platform is 3*f*+1, it is easy to see that when only 2*f*+1 nodes are required to be active and honest, the remaining *f* byzantine nodes can be tolerated by the system and still ensure safety and liveness.

However, given that in practical environments the number of faulty nodes is not stable and may unexpectedly increase over time, at Shopin we grant a higher level of security by increasing the threshold for consensus from *f*+1 to 2*f*+1. Therefore, only blocks of transactions that have accumulated aggregate BLS signatures from 2*f*+1 nodes may be attached to the chain.

When the churn rate increases, or the rate at which nodes leave the network, the number of unresponsive nodes increases. If the churn rate exceeds the given threshold of f, the system is unable to grant liveness and transactions stop being processed and blocks are no longer added to the chain until a sufficient number of validating nodes are alive and active.

ShopChain addresses the threat posed by churn impact seriously. It must be remembered that churn is not always malicious, and it may happen that for any external reason, such as bad connection or sudden shut down of WiFi. Therefore, the churn should not always be punished severely, since it may be unintentional. As ShopChain utilizes a BFT-like consensus algorithm, only the first 2*f*+1 honest and responding nodes are needed for signatures to be generated for safety and liveness be granted. Shopin DHT punishes nodes that are found to be unresponsive, offline, or breaking protocol by adjusting their reputation score and also by slashing their stake. The specifics of this mechanism will have to be outlined in a different document.

XII. Eclipse Attack Mitigation

Eclipse attacks are often confused with Sybil attacks. However, while a Sybil attack creates a situation whereby one individual operates multiple malicious nodes concurrently in the network threatening a subversion of the normal consensus balance and reputation system, an eclipse attack is directed toward honest nodes already active in the system. In other words, while a Sybil attack aims to attacking the whole network concurrently, an eclipse attack has the goal of isolating one or more individual users separately.

As the name may suggest, an eclipse attack, also referred to as information eclipse attack, is a kind of action that is undertaken by a given malicious actor to “eclipse” the normal information path (for example, DHT routing schemes) that a node would pursue in order to receive and send out information. Instead, the malicious actor may re-route the flow of information such that isolated honest nodes may communicate with nodes under its control.

This attack is allowed by the fact that blockchains and other decentralized systems prevent all nodes from simultaneously connecting to each other at the same time, i.e. each node in the network in a decentralized system is usually not connected to all other nodes in the network at the same time. Instead, each active node only connects to a selected subgroup of other nodes. In the case of Bitcoin, the size of this subgroup is 8, while it is 13 in the case of Ethereum. Therefore, for a malicious actor to completely isolate one node it may be enough to take control over only this mentioned subgroup.

The ease with which a malicious actor can carry out this type of attack depends in great part on the routing mechanism that is chosen by the distributed ledger platform. To explain this in further detail, the examples of Kademlia [2] and Chord DHT [11] routing schemes are given.

In the Chord scheme, the routing table, i.e. the guiding mechanism that directs the communication between nodes in a decentralized platform, orders nodes on the basis of their public key (PK). Therefore, a given node will communicate with a subgroup of nodes with the following public keys in numerical order, and with a group of nodes with preceding public keys in numerical order. On the other hand, Kademlia routing scheme directs communication between nodes by having each node communicating with a group of its k-closest nodes, meaning a subgroup of nodes that is chosen on the basis of communication proximity.

As it is clear, Chord [11] routing scheme allows for high predictability of guessing the incoming and outbound connections of any given node. In the opposite way, an attacker may only guess by trial and error which are the nodes communicating with the given node that the malicious actor wishes to target by means of an eclipse attack.

Clearly, this is extremely dangerous. First, because this attack would prevent the honest node from receiving requests and processing them, thus endangering the protocol’s safety and finality. Second, when a malicious actor has attacked a sufficient number of malicious nodes, this attack may even result in the validation of malicious transactions or the forking of the blockchain in favor of the dishonest node’s malicious chain.

Being aware of the severity of the threat posed by an eclipse attack, ShopChain, which is constructed on Kademlia routing scheme, puts in place two main mechanisms for mitigating, if not completely prevent such attacks.

The first prevention method is fairly common in the blockchain panorama and entails periodical re-sharding. In other words, nodes are, once every given period of time, re-positioned in the network and between the shards by means of changing their public key. In order to generate a new public key and rejoin the network, nodes must perform a series of Proof of Work calculations that will ultimately be the basis for their new Node ID and thus their new position in the network.

This implies a periodic change of the routing scheme, and thus lowers the predictability with which attackers may guess the incoming and outbound connections of the targeted nodes.

Second, ShopChain is a federated permissioned DLT. As a result, all Node Identities are clearly known and communications to and from said nodes is limited to only accepted identities. Therefore, the risk of Sybil or Eclipse attacks are virtually null, though it is a serious consideration the team has investigated deeply.

XIII. Configurable Trust Policies

Blockchains and other distributed ledger technologies are largely known for being "trustless". This, in short, means that a user operating on a blockchain, say making a transaction with another party, does not need to trust the integrity of the other party to be sure that the transaction fulfills both the security and finality properties required. As an example, if the sender in the transaction is trying to pretend to send money it does not own, thereby creating coins out of thin air, the cryptographic primitives and transaction validation securities in place in the network prevent the transaction from being finalized, and therefore the receiver from being tricked.

This works independently on who the parties are, as decentralized systems can be anonymous and users do not register by means of their real-life identity, i.e. government-held data. This system is great and innovative not only because it enhances security levels, but also because it increases equality.

By being a pseudonymous system, a decentralized network makes no difference between users depending on their background, provenience, or education level. However, there is one apparent downside to this. As a matter of fact, companies and any other entities deciding to utilize a blockchain for releasing and storing data requires trust. They need to know their users and ensure that documents that are uploaded and released may be verified. So how is trust achieved in a trustless system?

In order to overcome these problems, two solutions are available. The first one is a centralized trust system, which puts at its center a trusted third party that verifies the single pieces of information and ensures trust and security, together with the validity of inbound and outbound information. The second solution, quite the opposite, is a fully decentralized system.

ShopChain tackles the problem of how to handle trust by means of a decentralized trust system. This method does not require a trusted third party to ensure trust and verifiability and therefore allows for the creation of a decentralized platform at 360 degrees. This is achieved by implementing an ecosystem supporting digital identities and configurable trust policies.

The first step toward this goal is achieved by building a solid framework for Decentralized ID (DID). ShopChain is working on a DID implementation that allows user to operate one node exclusively via a structure verifying biometrics and location-based knowledge. This drastically reduces the possibility of individuals of running multiple identities and enhances trust by limiting the probability of Sybil attacks to the network.

Second, ShopChain intends to leverage Zero-Knowledge Proofs (also referred to as ZKPs). Although the practical implementation of ZKPs is noticeably complicated, the idea behind them is very simple. ZKPs are cryptographic methods that allow a user to prove the validity of a given piece of information without having to release the information itself. This clearly allows ShopChain to grant trust and security without releasing data or having to rely on a trusted third party.

XIV. Conclusion

The Shopin Network can be accurately described as a Secure Kademlia DHT that also possesses an overlay consensus layer. The elegance of this design is that the network topology is organized via consistent hashing into a ring structure, thereby when a node joins the consortium its position for leader rotation and data storage is trivial to compute.

Leveraging BLS aggregate signatures eliminates much of the message complexity and communication overhead the network would otherwise require in order to achieve Byzantine Fault Tolerance. By combining DHT and consensus responsibilities into single nodes, the Shopin Network can effectively manage different ledger requirements, such as points-and-rewards systems for partner retailers, in a decentralized, trustworthy manner, while the same nodes can also securely manage private customer data reliably.

In our early tests, we are able to achieve up to 300TPS (transactions per second), while simultaneously saving transaction and customer data among the DHT nodes. Our initial test network does not yet include threshold signatures, which may improve consensus and finality. By the same token, we believe we will inherit additional performance gains once erasure coding and other network optimizations are fully implemented. At this early stage, we hypothesize ShopChain will eventually be able to sustain performance comparable to Ripple, while reliably storing customer data securely simultaneously.

REFERENCES

[1] D. Boneh, B. Lynn, H. Shacham (2001). Short signatures from the Weil pairing. Manuscript.

[2] P. Maymounkov, D. Mazieres (2002). Kademlia: A Peer-to-Peer Information System Based on the XOR Metric

[3] J. S. Plank (2013). Erasure Codes for Storage Systems

[4] D. Boneh, M. Drijvers, G. Neven (2018). Compact Multi-Signatures for Smaller Blockchains

[5] M. Castro, B. Liskov (1999). Practical Byzantine Fault Tolerance

[6] Bitinforcharts website. Accessed November 6, 2019. URL: https://bitinfocharts.com/bitcoin/

[7] V. Buterin (2013). Ethereum: A Next-Generation Smart Contract and Decentralized Application Platform. URL: https://github.com/ethereum/wiki/ wiki/White-Paper.

[8] G. Wood (2017). Polkadot: Vision for a Heterogeneous Multi-Chain Framework

[9] Harmony Team (2018). Harmony

[10] S. Nakamoto (2008). Bitcoin: A Peer-to-Peer Electronic Cash System

[11] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, H. Balakrishnan (2001). Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications