ORI

Technology Overview Series
Consensus System

CLOUD FARMING SCHEDULE POOL

ABSTRACT

This document outlines the basic design principles of the consensus layer (the blockchain) of the ORI network. It is inspired by and similar to the Bitcoin blockchain, which achieves consensus when a majority of the computing power controlled by honest parties. In ORI the resource is not computing power, but individual cloud space. To achieve this, the proofs of work used in Bitcoin are replaced by proofs of cloud space. To get a mining dynamic like in the Bitcoin blockchain, ORI alternates proofs of cloud space with verifiable delay functions. We provide an initial security analysis of the ORI backbone, showing that as long as at least ≈ 61.5% of the space is controlled by honest parties ORI satisfies basic blockchain security properties.

Preface This is intended to be a technical "vision" summary of one possible direction that may be taken in further developing the blockchain paradigm together with some rationale as to why this direction is sensible. It lays out as much detail as possible at this stage of development a system which may give a concrete improvement on a number of aspects of blockchain technology. It is not intended to be a specification, formal or otherwise. It is not intended to be comprehensive nor to be a final design. It is not intended to cover non-core aspects of the framework such as APIs, bindings, languages and usage. This is notably experimental; where parameters are specified, they are likely to change. Mechanisms will be added, refined and removed in response to community ideas and critiques. Large portions of this paper will likely be revised as experimental evidence and prototyping gives us information about what will work and what not. This document includes a core description of the protocol together with ideas for directions that may be taken to improve various aspects. It is envisioned that the core description will be used as the starting point for an initial series of proofs-of-concept. A final version would be based around this refined protocol together with the additional ideas that are proven for the project to reach its goals.Initially, the blockchain was a string of cryptographic data,

The main core action is to distribute the bookkeeping party, and convert string of words into encrypted hash numbers to increase the entire blockchain world . The exchange process is referred to as "Hash Power Mining", which solves the difficulty of mining through the speed of the graphics card, and obtains Bitcoin from it. POW-maintains the interests of miners and expands the blockchain world in the above manner, but we believe that POW/POS alone There are certain limitations for blockchain technology to spread to the world, because the power consumption of mining has exceeded the basic power provided by the country for human needs.

THE HISTORY OF ORI

ORI’s team started in 2012 to develop and optimize peer-to-peer network transactions, mainly private chains
In 2012, the team planned that token can be sent directly to users without gateway or counterparty risk. This is the method of using all currencies (including U.S. dollars) on the Token network. The network may wish to combine the Token with the trust structure network of U.S. dollar payment, for example, to pay transaction fees. We set the supply of Token at a high level of 100 billion.
2013, the ORI team launched the "Private Chain" cryptocurrency with a total circulation of 100 billion, and the block transaction confirmation time is only 3 to 5 seconds.
April 2013, the first round of financing, we received recognition from Google Ventures, Andreessen Horowitz, IDG Capital Partners, FF Angel, Lightspeed Venture Partners, Bitcoin Opportunity Fund and Vast Ventures received 1.5 million US dollars in funding. This is the first round of multiple venture capital, including some of the most famous venture capital companies in the world.

2017, the "encrypted digital currency" successfully entered the Coinbase digital currency ranked TOP 30 places at the end of 2017. Since its establishment in 2020, it has provided services to 120,000 customer services in more than 20 countries and regions on June 2021, our team cooperated with (Phanes Technology) to launch its own unique CFSP "CLOUD FARMING SCHEDULED POOL" service for the public to participate.

Our development team "ORI" reached a cooperation agreement with the cloud computing power mining company "Phanes Technology" and created a unique "CFSP" CLOUD FRAMING SCHEDULED POOL. The green mining model officially opened the "ORI" encrypted digital currency. The ORI peer-to-peer network consists of clients connected through the blockchain network, which can make cloud mining to the blockchain network for everyone. Users perform six active functions: (a) participate data become registry (b) participate in decentralized seed generate. (c) participate in proposed blocks hash lock, the client also observes the block and establish their own view of the chain. (d) Become super node in SPOS to participate in enhancing the ORI community consensus to genarate reward and accelerating TPS (e).Start cloud mining in CFSP focuses on environmentally friendly mining. (f) participate mapping DNS seed data parser to let open cloud link port and multiply node port to your super node.
The network node can be a collection of all clients. On a large scale.
Network, the node will be smaller than the set of all super nodes, and the super node is upgrade to the next round of changes (ie, from one block to another). Through the SPOS mechanism. Select the next super node in the SPOS to create a brand new node for all clients.

ORI'S CONSENSUS MECHANISM HAS SIX LAYERS AS DEPICTED IN FIG

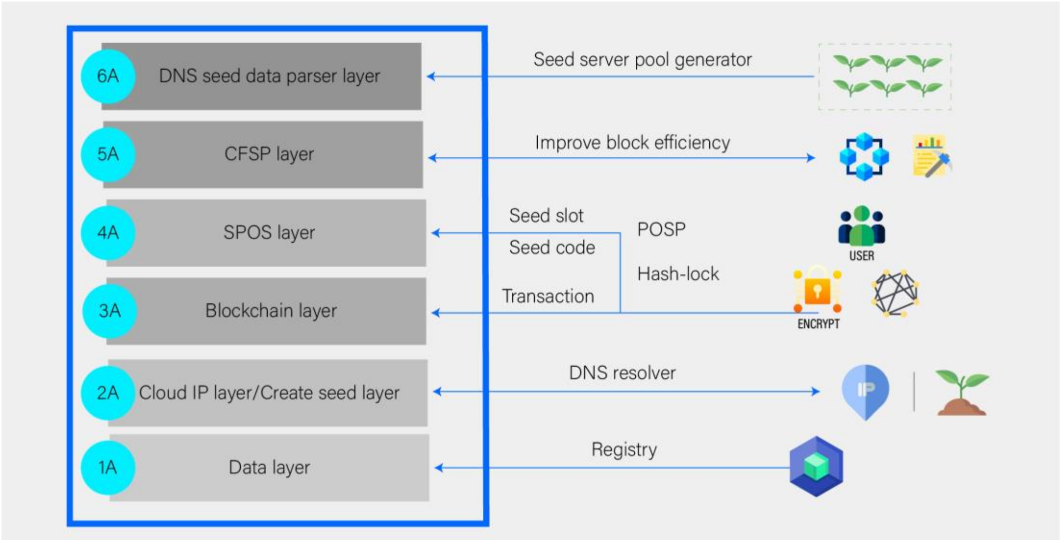


Figure 1: ORI's Introduction consensus mechanism layer.

Data Layer
All customer registrations are digitized.

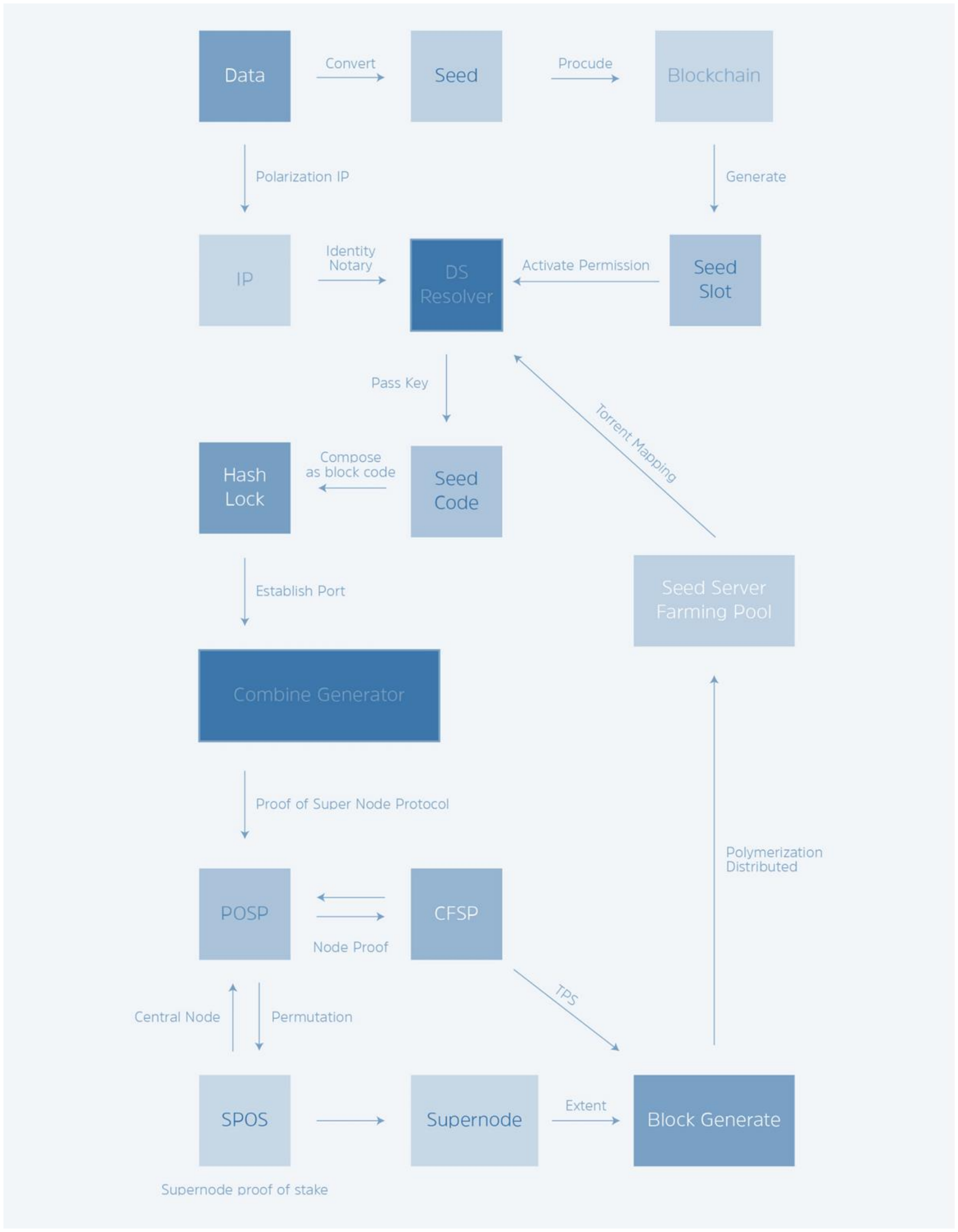
Cloud IP Layer/Create Seed Layer
All higher-level IP converters DNS RESOLVER (DRR); transmit data to create seeds (smart contracts).

Blockchain Layer: Hash Lock /Seed Code to Hash
Verify transactions driven by the probabilistic slot protocol
Generated by a combination of blockchains.

SPOS Layer: Generate Block Link Up with Other Supernode
In SPOS cold mint technology can be understood as a user connected to a supernode in the wallet and can use coins for mining without transferring their coins to the supernode. In terms of security, even if a supernode is attacked by a hacker, users can quickly rent their coins to a new supernode, and it is almost impossible for a hacker to carry out a 51% attack.

CFSP Layer: Cloud Farming Schedule Pool
The resulting CFSP combination will provide rapid certainty
Assure customers and external observers that the seed code is used to generate the mining key, and the program code is linked to the cloud farming scheduling pool to start mining

DNS Seed Data Parser Layer: A Decentralized Seed Server Pool
The seed data is returned to the pool, and the hash value is passed back and forth in the cloud network, The more supernodes, the faster the transmission speed of the overall block.



The First Layer

Identity and registry. Active participants in the ORI network are called cloud data. All clients in ORI have been registered, that is, they have permanent pseudonymous identities. Customer registration has advantages over a typical proof-of-work blockchain. In a typical proof-of-work blockchain, it is impossible to link different blocks to the same miner. For example, if a deposit is required for registration, misbehaving clients will lose all their deposits, while a typical proof-of-work blockchain miner will only give up block rewards when misbehaving.

Therefore, the punishment for misconduct with registered identities may be much greater than the punishment for unregistered identities. This is especially important because the blockchain can track unlimited external value beyond the value of the native token itself. In addition, ORI supports open membership by providing a decomposition program to generate the registration code into a registry, which is the first level of responsibility.

The Second Layer

Protocol communication IP cloud identity becomes a random IP port of identity on the cloud, allowing the IP to get the mark conversion and transfer information in the record section; personal and other key information is hashed again to generate a new SHA1 value as a seed The HASH value of the data, the SHA1 value of the data, such as a 1G capacity, if divided into 1M blocks, it will be divided into 1000 blocks, and the seed will have the fingerprint value of these 1000 data blocks plus the seed program The link generator converts the profile information summarized by the seed into the DS Resolver random beacon seeding. The seed in the second layer is an irreplaceable code, a verifiable random function, and a seed quantization throughput speed, which is generated by connecting multiple seed ports. Every random output is unpredictable by anyone until it is available to everyone before the seed code generated by the DS needs to unlock the Hash Lock. This is a key technology of the ORI system, which relies on a threshold signature verification method with unique and non-interactive characteristics. The DS signature scheme is the only practical solution that can provide these functions, and ORI has a specially optimized DS implementation that uses a threshold mechanism created by randomness to solve the basic "last participant" problem. Any decentralized protocol used to create public randomness without a threshold mechanism will encounter problems, that is, the last participant in the protocol knows the next DS Resolver random value and can decide to terminate the protocol. There are many more Others, such as the Seed slot scheduling of the DS parser, the caching mechanism, the scheduling algorithm of the file partition, the performance improvement of the universal server for tens of millions of users, and so on.

The Third Layer

Blockchain and bifurcation analysis. The third layer deploys the "Probability Slot Protocol" (PSP) in seed slot. The protocol ranks clients according to each height of the chain, and the order is deterministically deduced from the unbiased output of random code of that height. Then assign a weighted seed space to the block proposal according to the proposer’s ranking, which is monitored and filtered by the DS resolver.
The block from the client at the top of the list gets a higher weight. The bifurcation is resolved by biasing towards the "heaviest" chain in terms of cumulative block weight, which is similar to the traditional proof-of-work consensus based on the highest cumulative workload.

The first advantage of the PSP protocol is that rankings are instantly available, which allows predictable and constant block time.
The second advantage is that there is always a client with the highest ranking, which can achieve homogeneous network bandwidth utilization. On the contrary, competition between clients will be conducive to sudden use. The third advantage is space compression and efficient arrangement. The information associated with each seed can be recorded quickly and automatically in the resolution of bifurcation analysis. The excess space is compressed, and the information is divided into multiple seeds and divided into the locked area for absorption. With replication, even if the Tracker is not connected, it will directly use the distributed hash table DHT technology to find like-minded nodes through the network, and always find the seed ports of other node ports. The more sub-distributions will improve the efficiency and Time cost.

The Fourth Layer

ORI gracefully handles the speed generated by each node and the underlying technology of the community. SPOS is a basic structure function created by the community consensus viscosity. After the seed information is unlocked through the code, the arrangement results are calculated by the highest-level Hash Lock.
This can rule out that if two clients start to disagree on average use, then both clients will stop. If there are many evenly distributed clients and one who initially disagrees with all others, then the network may continue and only the isolated clients will be suspended. This is exactly the given expected behavior scenario. Other blockchains cannot handle this situation well. The occurrence of this kind of incident poses a real threat to them. Moreover, the seed link will independently generate anonymous super nodes. The node level is divided into five categories, and the management tasks and tasks of different levels are different, which makes continuous attacks ineffective. The start of super node work is floating. The SPOS level will make the node tasks better. If the attacker terminates any of the nodes in the work bar, this node will immediately switch to the new task bar, and the attacked task will immediately return to Hash Lock, because a single task bar is composed of multiple on the network Anonymous super nodes work, causing the attacker's attack to be invalid. At this level, the speed of super node generation and combination is extremely fast, and there is a seed shell and seed code before the super node is generated. The seed code is forward and backward and cannot be found and verified to change the seed information, it is almost impossible to find the port attack entry or attack the node.
Notarization and near-instant end. A given final transaction implies a system-wide consensus that the given transaction has been executed irreversibly. Although most distributed systems require fast transaction termination, notarization and consensus can hardly coexist, and the existing blockchain technology cannot provide it.

ORI's SPOS layer can increase the throughput of each node and affect the transaction speed. In addition to providing notarized ultimate transactions, it will also increase the consensus level of the community within the foundation.SPOS also uses cold minting technology and does not need to transfer your own coins to super nodes. In terms of security, even after the super node is attacked by hackers, users can quickly rent their coins to the new super node, making it almost impossible for hackers to carry out 51% attacks. Subsequently, the cold mint technology highlights the way that ORI preachers are good at leveraging assets, and safe backing of assets can also allow funds to be leveraged.

The Fifth Layer Protocol

CFSP (Cloud Farming Scheduled Pool) is based on the underlying technology of POSP. It aggregates multiple node mine combinations through the node certification protocol to form a cloud node aggregation pool.
To become a miner, you must have a node to obtain the mining qualification. After the CFSP starts the node mining qualification, you can enter the aggregation pool for mining. The POSP (Proof of Super node Protocol) proves that the P2P random broadband calculation data in the miner node can be used as the basis of computing power. The larger the computing power base, the greater the mining power and the more rewards you will receive. The greater the responsibility of the miner node in the community.
POSP advocates green mining, so it focuses on environmentally friendly mining. Through the infrastructure of NET and Blockchain, it becomes a regular and orderly program. Network application is the foundation, and POSP is a mechanism to maintain the trust and responsibility of miners. The emergence of POSP is to allow every family to use common basic facilities to mine at home. In addition, POSP supports CFSP (Cloud Farming Scheduled-Pool) cloud node aggregation pool green mining.
The work of the combined generator is equivalent to linking the protected seed development port, in which the super node identity verification layer is screened out by level. The secondary port is especially important. The super node enters the POSP and starts to calculate the data to integrate all the calculation power distribution certificates. Distribute and link multiple ports among multiple nodes to perform proof of work in the pool. Once the super node gets a response in SPOS, POSP will join the certified node to enter CFSP for mining.
CFSP breaks the traditional mining system and combines the functions of blockchain and the Internet to create an Internet blockchain mining system (for example, cloud mining we are familiar with). The difference between CFSP is that it promotes green mining and excludes non-environmental mining mode.
The mining model over the years is a code operation program. As long as the code operation program is based on a fixed system, it can replace the physical mining machine and greatly reduce the problem of natural energy consumption.

The Sixth Layer Protocol

DNS seed data parser layer DNS (Decentralized Nous System) infrastructure is to allow the super nodes released by SPOS to create a lot of impetus through mining and provide the DNS resolver for the client to be associated after the super node verification. Seeds, in other words, they convert human-readable addresses (such as user block codes) into machine-readable nodes. When users try to navigate to truncation, their operating system sends a request to the DNS seed data resolver. The DNS seed data parser responds with IP data and seed to DRR, and the cloud link port will allow the working seed breeding independent network to fall back to the DNS seed data parser layer, for example: when the super node response falls back to the DNS seed data parser layer, DRR will make the seeds generate independent network connections. When connected to the ports of other seeds, new block nodes will be quickly generated, and will not repeatedly generate discarded nodes. The seed port link can link multiple seeds independently Network, which makes DNS reverberate faster and more efficient. The entire class develops rapidly, and the problem of congestion is eliminated. At present, many decentralized open sources are seriously lacking in this sector.

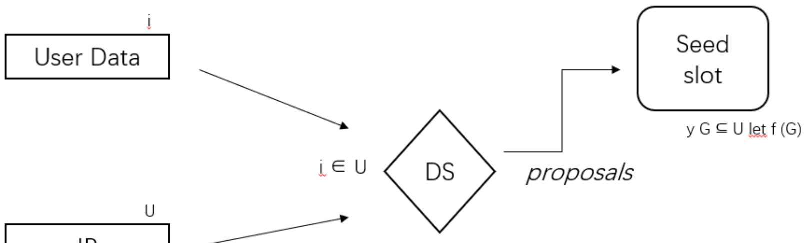
The DNS resolver saves the response to the IP seed query for a period. In this way, the resolver can respond to future queries faster without having to communicate with many of the links involved in the typical DRR resolution process. As long as the specified time-to-live associated with this independent network combination allows the DNS resolver to save the response in its cache.

DS resolver block validator, super node for seeding the egress payloads to some subset of each ORI chain validator group for the next block (and possibly some favored collator(s).

As such, the data pathways per node grow linearly with the overall complexity of the system. While this is reasonable, as the system scales into hundreds or thousands of chains, some communication latency may be absorbed in exchange for a lower complexity growth rate. In this case, a multi-phase routing algorithm may be used in order to reduce the number of instantaneous pathways at a cost of introducing storage buffers and latency.

CRYPTOGRAPHIC PRIMITIVES

Hash function. We assume we have a collision-resistant hash function H with digests of bit-length l where l matches the security parameter κ. We also assume we have a cryptographically secure pseudo-random number generator PRG which turns a seed ξ into a sequence of values PRG(ξ,i) for i = 0, 1, ... 4.3.3 Pseudo-random permutations. The sequence PRG(ξ,i) can be used as input to the Fisher-Yates shuffle [9, Algorithm 3.4.2P] to produce a random permutation of U . The result is an bijective map {1, . . . , |U |} → U which we denote by PermU (ξ). We assume that the adversary is bounded computationally and that the computational DNS-Hellman problem is hard for the elliptic curves with pairings in having seed code/seed slot system



i ∈ U may behave arbitrarily, e.g., it may refuse to participate in the protocol or it may collude with others to perform a coordinated attack the system.

Adversarial strength. For any G ⊆ U let f (G) number of Byzantine replicas in G.

Assumption 1. There is β > 2 such that |U | > β f (U).

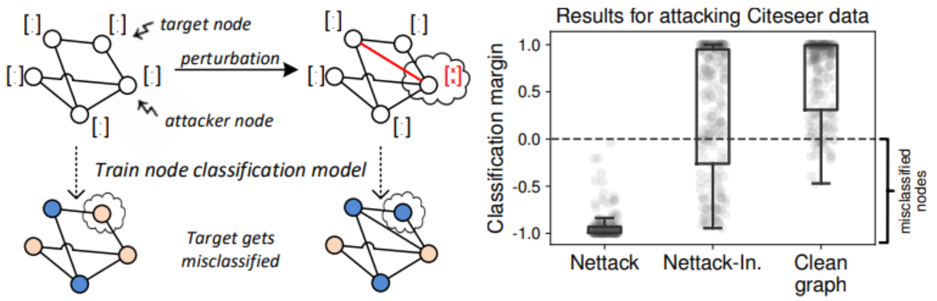
A replica that faithfully follows the protocol is called honest and all other replicas are called Byzantine. A Byzantine i ∈ U may behave arbitrarily, e.g., it may refuse to participate in the protocol or it may collude with others to perform a coordinated attack the system. Adversarial strength. For any G ⊆ U let f (G) denote the number of Byzantine replicas in G. Assumption 1. There is β > 2 such that |U | > β f (U).
The value 1/β is called the adversarial strength. In practice, Assumption 1 is achieved through economic incentives in conjunction with a form of Sybil resistance.3 4.2.3 Honest groups. Let n be the group size. Then a group G is called honest if n > 2f (G). The protocols described in § 5-7 rely on Assumption 2. Each group G used in the system is honest. 4.2.4 Random samples. Given Assumption 1, the universe U itself is honest. Each group G ⊆ U used in the system is a random sample of size n drawn from U.
Given n, the probability Prob [G honest] can be calculated as follows: Proposition 4.1. Let CDFhg (x, n , M, N) denote the cumulative distribution function of the hypergeometric probability distribution where N is the population size, M is the number of successes4 in the population, n is the sample size and x is the maximum number of successes allowed per sample. Then Prob[G honest] = CDFhg (⌈n/2⌉ − 1,n, ⌊|U |/β⌋, |U |).

Given an acceptable failure probability ρ, we can solve (4.3) for the minimal group size n = n(β, ρ, |U |) such that Prob[G honest] > 1 − ρ for each random sample G ⊆ U with |G| = n. The result for the example value |U | = 104 and different values for ρ and β
We consider the task of (semi-supervised) node classification in a single large DRR having binary node features. Formally, let G = (A,X) be an attributed seed, where A ∈ {0, 1} N ×N is the adjacency matrix representing the connections and X ∈ {0, 1} N ×D represents the nodes' features. We denote with xv ∈ {0, 1} D the D-dim. feature vector of node v. we assume the node-ids to be V = {1, . . . , N} and the feature-ids to be F = {1, ...,D}. Given a subset VL ⊆ V of labeled nodes, with class labels from C = {1, 2, . . . ,cK }, the goal of node classification is to learn a function д : V → C which maps each node v ∈ V to one class in C. 2 Since the predictions are done for the given test instances, which are already known before (and also used during) training, this corresponds to a typical transudative learning scenario. In this work, we focus on node classification supernode convolution layers. In particular, we will consider the well-established work . Here, the hidden layer l + 1 is defined as H (l+1) = σ D˜ − 1 2A˜D˜ − 1 2 H (l)W (l) , (1) where A˜ = A + IN is the adjacency matrix of the (undirected) input port G after adding self-loops via the identity matrix IN . W (l) is the trainable weight matrix of layer l, D˜ i i = P j A˜ ij , and σ (·) is an activation function (usually ReLU). In the first layer we have H (0) = X, i.e. using the nodes' features as input. Since the latent representations H are (recursively) relying on the neighboring ones (multiplication withA˜), all instances are coupled together. Following the authors of seed slot, we consider GCNs with a single hidden layer: Z = fθ (A,X) = softmax Aˆ σ AXWˆ (1) W (2) , (2) where Aˆ = D˜ − 1 2A˜D˜ − 1 2 . The output Zvc denotes the probability of assigning node v to class c. Here, we used θ do denote the set of all parameters, i.e. θ = {W (1) ,W (2) }. The optimal parameters θ are then learned in a semi-supervised fashion by minimizing crossentropy on the output of the labeled samples VL, i.e. minimizing L(θ;A,X) = − X v ∈VL lnZv,cv , Z = fθ (A,X) (3) where cv is the given label of v from the training set. After training, Z denotes the class probabilities for every instance in the DRR.

The Result Of Nettack

Given the node classification setting as described above. our goal is to perform small perturbations on the port G (0) = (A (0) ,X (0)), leading to the port G ′ = (A ′ ,X ′), such that the classification performance drops. Changes to A (0) , are called structure attacks, while changes to X (0) are called feature attacks. Target vs. Attackers. Specifically, our goal is to attack a specific target nodev0 ∈ V, i.e. we aim to changev0's prediction. Due to the non-i.i.d. nature of the data, v0's outcome not only depends on the node itself, but also on the other nodes in the port.

Thus, we are not limited to purtubingv0, but we can achieve our aim by changing other nodes as well. Indeed, this reflects real world scenarios much better since it is likely that an attacker has access to a few nodes only, and not to the entire data or v0 itself. Therefore, besides the target node, we introduce the attacker nodes A ⊆ V. The perturbations on G (0) are constrained to these nodes, i.e. it must hold X ′ ui , X (0) ui ⇒ u ∈ A , A ′ uv , A (0) uv ⇒ u ∈ A ∨v ∈ A (4) If the target v0 < A, we call the attack an influencer attack, since v0 gets not manipulated directly, but only indirectly via some influencers. If {v0} = A, we call it a direct attack. To ensure that the attacker cannot modify the port completely, we further limit the number of allowed changes by a budget Δ: X u X i |X (0) ui − X ′ ui | + X u.



```
Algorithm 1: Nettack: Adversarial attacks on port
Input: Port $G^{(0)} \leftarrow (A^{(0)}, X^{(0)})$, $target\ node\ v0,$
    $attacker\ nodes\ A\ , modification\ budget\ \Delta$
Output: Modified Port $G' = (A', X')$
  $Train\ surrogate\ model\ on\ G^{(o)}\ to\ obtain\ W\ /$
  $t \leftarrow 0$ ;
  while $|A^{(t)} - A^{(0)}| + |X^{(t)} - X^{(0)}| < \Delta$ do
    $C_{struct} \leftarrow candidate\_edge\_pertubations(A^{(t)}, A)$;
      $e^* = (u^*, v^*) \leftarrow \underset{e\ \in C_{struct}}{\text{argmax}}\ s_{feat}\ f; G^{(t)}, v_0$ ;
    $C_{feat} \leftarrow$ candidate_feature_perturbations$(X^{(t)}, A)$

      $f^* = (u^*, i^*) \leftarrow \underset{f\ \in C_{struct}}{argmax}\ s_{feat}(f; G^{(t)}, v_0$ ;

     if $s_{struct}\ , e^*, G^{(t)}, v_0) > s_{feat}\ (f^*, G^{(t)}, v_0)$ then
                  $G^{(t+1)} \leftarrow G^{(t)} \pm e^*$;
               else $G^{(t+1)} \leftarrow G^{(t)} \pm f^*$ ;
                          $t \leftarrow t + 1$;
 return : $G^{(t)}$
 // Train final port seed model on the corrupted
 port $G^{(t)}$;
```

we can derive an incremental update – we don't have to recompute the updated [Aˆ2]v0 from scratch.
Theorem given an adjacency matrix A, and its corresponding matrices A˜, Aˆ2, D˜. Denote with A′ the adjacency matrix when adding or removing the element e = (m, n) from A.

$$[\hat{A}^{\prime 2}]_{uv} = \frac{1}{d_u\, d_v} \cdot \overline{\tilde{d}_u\, \tilde{d}_v}\ [\hat{A}^2]_{uv} - \frac{\tilde{a}_{uv}}{\tilde{d}_u} - \frac{a_{uv}}{\tilde{d}_v} + \frac{a'_{uv}}{\tilde{d}_v} + \frac{\tilde{a}'_{uv}}{\tilde{d}_u} -$$

$$\frac{\tilde{a}_{um} a_{mv}}{\widetilde{d_m}} + \frac{\acute{a}'_{um}\, a'_{mv}}{\widetilde{d_m}} - \frac{\tilde{a}_{un} a_{sv}}{\widetilde{d_n}} + \frac{\tilde{a}_{un} a_{nv}}{\widetilde{d_n}}$$

where $\tilde{d}, \acute{a}$, *and* $\check{a}$, *are defined as* (*using the Iversom bracket* I):

$$\tilde{d}_k = \tilde{d}_k + \mathrm{I}[k \in \{m, n\}] \cdot (1 - 2 \cdot a_{mn})$$
$$\tilde{a}_{kl} = \tilde{a}_{ki} + \mathrm{I}[\{k, l\} = \{m, n\}] \cdot (1 - 2 \cdot a_{kl})$$
$$\tilde{a}_{kl} = \tilde{a}_{ki} + \mathrm{I}[\{k, l\} = \{m, n\}] \cdot (1 - 2 \cdot \tilde{a}_{kl})$$

ProoF, Let S and S' be defined *as* $S = \sum_{k=1}^{N} \frac{\tilde{a}_{uk} a_{kv}}{\widetilde{d_k}}$ *and* $S' = \sum_{k=1}^{N} \frac{\tilde{a}_{uk} a_{kv}}{\widetilde{d_k}}$. *We have* $[\hat{A}]_{uv} = , \frac{\tilde{a}_{uv}}{\widetilde{d_v} d_v}$. if u ≠ v , then

$$[\hat{A}^2]_{uv} = \sum_{k=1}^{Xv} [\hat{A}]_{uk} [\hat{A}]_{kv}$$

$$= \frac{\tilde{a}_{uv}}{\tilde{d}_u\, \tilde{d}_u \tilde{d}_v} + \frac{\tilde{a}_{uv}}{\tilde{d}_u\, '\tilde{d}_u \tilde{d}_v} + , \quad \frac{1}{\hat{d}_u\, \hat{d}_v} S 。$$

Having the above equation for $\hat{A}^{\prime 2}$, we get

$$[\hat{A}^{\prime 2}]_{uv} \cdot \overline{\tilde{d}_u\, \widetilde{d_v}} - [\hat{A}^2]_{uv} \cdot \overline{\tilde{d}_u\, \widetilde{d_v}} = \_\_\_\_\_\_\_\_\_\_ \quad \_\_\_\_\_\_\_\_\_\_\_\_$$
$$\frac{\tilde{a}_{uv}}{\widetilde{d_u}} - \frac{a_{uv}}{\widetilde{d_v}} + \frac{a'_{uv}}{\widetilde{d_v}} + \frac{\tilde{a}'_{uv}}{\widetilde{d_v}} + (S' - S).$$

After replacing $S' - S$

The above equation, it is straightforward to derive.Deriving this equation for the case u = v is similar. encompasses both cases. above statement enables us to update the entries in Aˆ2 in constant time; and in a sparse and incremental manner. Remember that all a˜uv ,, auv , and au′ v are either 1 or 0, and their corresponding matrices are sparse. Given this highly efficient update of [Aˆ2]v0 to [Aˆ′2]v0 , the updated log-probabilities and, thus, the final score according to can be easily computed. Feature attacks. The feature attacks are much easier to realize.
Indeed, by fixing the class c ≠ cold with currently largest log-probability score [Aˆ2 XW]v0c , the problem is linear in X and every entry of X acts independently. Thus, to find the best node and feature (u∗, i∗) we only need to compute the gradients and cold wallet in the Spos layer are included

$$\Upsilon_{ui} = \frac{\partial}{\partial X ui}\ [\hat{A}^2 x\ w] v_0 c - [\hat{A}^2 x\ w] \upsilon_0 c_{old}$$
$$= [\hat{A}^2]\upsilon_0 u\ [w] ic - [w] c_{old}$$

and subsequently pick the one with the highest absolute value that points into an allowable direction (e.g. if the feature was 0, the gradient needs to point into the positives). The value of the score function seed for this best element is then simply obtained by adding $|\Upsilon_{ui}|$ to the current value of the loss function:

$$L_s(A,X;v_0) + |\Upsilon ui| \cdot I[(2 \cdot X_{ui} - 1) \cdot \Upsilon_{ui} < 0]$$

All this can be done in Spos layer constant time per feature. The elements where the gradient points outside the allowable direction should not be perturbed since they would only hinder the attack – thus, the old score stays unchanged

After exploring how our attack affects the (fixed) proxy model, We will now find out if our attack was also successful established a deep defense model of the port. To this end, we pursue use the previous method and use a budget of Δ = dv0 + 2, where dv0
Yes, the extent of the target node we are currently attacking. This is due toobserve that high nodes are more difficult to attack than low-degree. Below we always report the score

$$X = Z^* \upsilon_0, c_{old} - max_{c \neq c_{old}} Z^* v_0, c$$

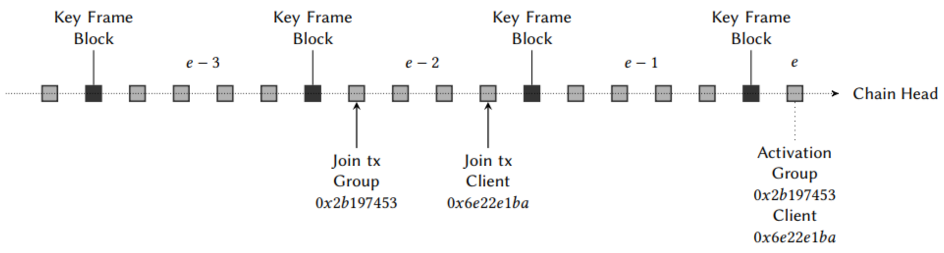
We call X the classification margin. The smaller X, Better for values less than 0, the target will be misclassified. Note that for clean graphics, this can even happen because The classification itself may not be perfect.
Evasion and poisoning attack, we evaluated the performance of network attacks against two types of attacks: evasion attacks, where the model parameters are based on Clean nodes; and poisoning attacks, the model is retrained After the attack (over 10 times on average). Each point represents a target node. It can be seen that direct attacks are very successful.Even in challenging cases of poisoning, almost every target will get classification error.

Therefore, we conclude that our alternative model and the loss is a sufficient approximation of the nonlinear true loss, the model after retraining the disturbance data. Obviously, influencers attacks are harder, but they still works in both cases. Attacks are usually harder due to poisoning. And better match the transduction learning scenario, we report only these results are described below. The damage produced by the comparison results is transferred to different convolution methods: GCN and CLN . The most striking thing is that even unsupervised models are not affected in any way only a structural attack was carried out. Following node classification is performed by training logistic regression learned to embed. In general, we see that direct attacks are much more difficult than the problems caused by influencer attacks. In these plots, we also resolved a comparison with the two baselines Rnd and FGSM, both of which were operating in a direct attack setting.
We evaluate. Not surprisingly, influencer attacks result in lower Compared with direct attacks, the performance is reduced. We see that the performance of FGSM is worse than nettack, we think This is because the gradient method is not optimal For discrete data. we drew the newly generated code lines of Posp and partitioned them.
When changing the elements in A, the actual change in gradient and loss.
Usually, the gradient does not approximate the loss very well in the chain reaction and Direct audit corruption does not even match the logo. Successfully established different defense schemes at the overall level of ORI's consensus. When setting up a threshold signature scheme, we do not want to rely on any trusted third party. Therefore, the group G runs a DKG for BLS to set up the group public key and the secret key shares during the initialization of the blockchain system. The threshold t is a parameter of the setup. Once the DKG has finished successfully, it outputs a public verification vector DS ∈ G t2, and leaves each replica i ∈ G with its secret key share dkG,i . The verification vector DS gets committed to and recorded in the blockchain, for example in the genesis block. Let DS = (d0, . . . ,dt−1). The group public key is dkG = s0 ∈ G2. The secret key dkG corresponding to dkG is not known explicitly to anyone in S but can be used implicitly through dkG,i . The verification vector DS can be used to recover the public key share dkG,i ∈ G2 corresponding to dkG,i via "polynomial" substitution

$$dk_{G,i} = \prod_{k=0}^{t-1} v_k^{i^k} \in \mathbb{G}_2.$$

Hence, all signature shares produced by "i" can be publicly verified against the information DS and i. The group public key dkG can be used to verify the output of Recover signing process. Recall that a replica enters round r upon seeing the first notarization for round r − 1. At the beginning of its round r, replica i ∈ D computes the signature share σr,i = Sign(r ∥ ξr−1,skD,i), where ξr−1 is the random value of round r − 1. To bootstrap, ξ0 has been set to a nothing-up-my-sleeve number, e.g. the hash of the string "ORI". Replica i then broadcasts (i, σr,i). Any replica who receives this data can validate (i, σr,i) against the public information DS as described above. If valid then the replica stores and re-broadcasts (i, σr,i). As soon as a replica has received at least t different valid signature shares, it runs Recover (i1, . . . ,it , σr,i1 , . . . , σr,it) to compute the group signature σD,r . Finally, the random output ξr for round r is computed as the hash of σD,r . We emphasize that the signing process is non-interactive. Any third party can do the recovery after a one-way communication of sufficiently many shares.

Let r be the first round of epoch e. For each j ≤ mmax the j<th candidate group is defined as G = Group(ξr, j). The members of G run a DKG to establish a group public key dkG . If the DKG succeeds then the members create a registration transaction for G which contains the tuple x = (e, j, dkG). After x is signed by a super-majority of G, any member can submit x for inclusion in the blockchain. The validity of the signature under x is publicly verifiable against the information already on the blockchain, i.e., the pool U of active replicas and the random DRR output ξr that defined the group. A registration transaction x is only valid if it is included in a block that lies within the epoch e. If the DKG fails or x fails to get a super-majority signature from G or x is not included in the blockchain within epoch e then G cannot register. An adversary can cause the registration to fail. For example, if the super-majority is defined as a 2/3-majority then an adversary controlling ≥ 1/3 of G can deny the signature under x. However, due to variance, this will happen only to some of the group candidates. For example, an adversary controlling < 1/3 of U will control < 1/3 in at least half of all groups. Groups are de-registered automatically when they expire after a fixed number of epochs defined by a system parameter.



Epochs and Registration. The chain is divided into epochs ORI by the round numbers of the blocks. A client joins by submitting a special transaction into a block which also locks up a stake deposit. A group joins by successfully executing a DKG (distributed key generation) and submitting the result as a join transaction into the blockchain. Clients and groups become actively involved in the protocol only after a gap of at least 1 epoch between their join transaction and their first activity.
Under normal operation r, every transaction included in a block for round r is final after two confirmations plus the maximum network roundtrip time 2Δ. From the perspective of an arbitrary observer, the Main Theorem means the following. Suppose an observer sees a transaction x that has received two confirmations, i.e. a notarized block Br for round r
containing x and another notarized block Br+1 with prv Br+1 = Br. If round r experienced normal operation, then, at time 2Δ after the observer received the notarization for Br+1, the finalization algorithm (Alg. 2) is guaranteed to append Br to the observer's DNS seed date parser. We assume here that T in Alg. 2 is set to 2Δ. The proof of
the Main Theorem will occupy.
We will provide proofs for the synchronous model where an upper bound for the network traversal time Δ is known. We assume that processing times for messages are included in the network traversal time.

This section makes statements about the relative timing of events that happen at different replicas. We do not assume normal operation in any round and therefore have to consider the possibility of multiple POSP zr, z'r , . . . being created and broadcasted for the same round r. We assume that a message broadcasted by an honest replica at time t reaches every honest replica before $t + \Delta$ (i.e. at a time $< t + \Delta$). Since processing times are not in the scope of our analysis, we assume all processing times to be zero.

This applies to the creation as well as to the validation of all messages including block proposals, signatures, DNS receive and return, seed code generate, and POSP outputs. As a consequence, for example, when a replica i receives a SPOS super node output ξ_r at time t then it broadcasts its block proposal for DNS r at the same time t. Or, when a seed code i receives a link to SPOS port z_r for r ,at time t then i broadcasts its super node share for DRR $r + 1$ immediately at the same time t.

$$\underline{T}(A) := i_{honest}^{min}\ T_i(A), \bar{_T}(A)$$
$$:= i_{honest}^{max}\ T_i(A).$$

For example, $\underline{\tau}(r)$ is the time when the first honest replica enters DNS r and $\bar{\tau}(r)$ is the time when the last honest replica enters DNS r. Finally, it is also of interest when an event can first be seen or constructed by the adversary. Therefore, we define:

$$\underline{\tau} * (A) := \min_i T_i(A).$$
$$\underline{\mathrm{I}} *(A) := \min_i T_i(A).$$
$$\underline{^r}(A) + \Delta \leqslant \underline{^r}(r+1) \Rightarrow \bar{_r}(A) \leqslant$$
$$\underline{^r}(A) + \Delta$$

Now let A be any event that falls under the relay policy in seed code port for round r. If τ ¯ (A) + Δ ≤ τ ¯ (r + 1) then the same argument applies. Indeed, the assumption means that all replicas along the broadcast path will still be in round r.

From $\mathrm{T\_}(A) \leqslant \mathrm{T\_}(r)$ + (BlockTime - Δ) we conclude
$$\mathrm{T\_}(A) + \Delta \leqslant \mathrm{T\_}(r) + \text{BlockTime} \overset{(9.5)}{\underset{\leq}{—}} \mathrm{T\_}(r+1).$$
$$\bar{_T}(A) \leqslant \mathrm{T\_}(A) + \Delta \leqslant \mathrm{T\_}(r) + \text{BlockTime}.$$

The interpretation is that if a round-r event is broadcasted by τ (r) +(Block Time−Δ) then it is guaranteed to saturate the network, and this happens by τ¯(r) + Block Time.

(“Fast" Bound). For all torrent mapping we have:

$$L(r) + BlockTime \leqslant L*(r+1).$$
$$\bar{_T}(\mathcal{E}_r) \leqslant \bar{_T}(r) + \Delta$$

We now assume BlockTime $\geq 3\Delta$ for the rest of the subsection. The timing of events with BlockTime $= 3\Delta$ is illustrated

Set $x_0 := \tau_{-}(r) +$ BlockTime. Let B_r be the unique proposal made by i for round r. all honest node receive B_r by time x_0. For each time $x \geq x_0$, we consider the set S_x of valid block proposals for round r that have been received by at least one honest replica. More formally, S_x consists of those valid round-r proposals B that satisfy $\tau_{-}(B) \leq x$. For example, $B_r \in S_{x0}$. We then define

$$f(x) := \ min\{rk\ B|B \in Sx\}$$

For example, $f(x_0) \leq d$ because B_r has rank d. For $x \geq x_0$, the function f has values in the non-negative integers and is monotonically decreasing. Since $f(x_0) \leq d$, it follows that

there is a time $x_0 \leq x_1 \leq x_0 + d\Delta$ such that $f(x_1) = f(x_1 + \Delta)$. (Otherwise, if $f(x + \Delta) \leq f(x) - 1$ for all $x_0 \leq x \leq x_0 + d\Delta$ then $f(x_0 + (d + 1)\Delta) < 0$, a contradiction.)

We claim $\tau(r + 1) \leq x + 2\Delta$.

$$x_1 + 2\Delta \leq x_0 + (d + 2)\Delta$$
$$= T\_(r) + \text{BlockTime} + (d + 2)\Delta$$

If a notarization for a block different from B arrives at any replica before $x + 2\Delta$ then the claim is already proven. We assume w.l.o.g. that this is not the case. Note that under this assumption, the events B and any signature under B.

Let $B \in S_{x1}$ with rk $B = f(x_1)$. Since $B \in S_{x1}$ we have $\tau_{-}(B) \leq x_1$.

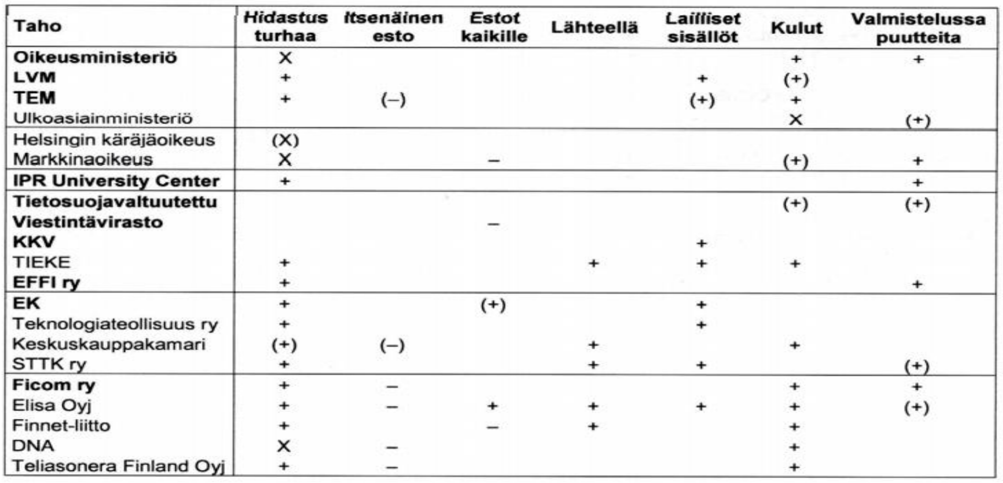
$$\underline{r}(B) \leqslant x_1 + \Delta.$$

The facts τ¯(B) ≤ x1 + Δ and rk B = f (x1 + Δ) together mean that B has minimal rank in every honest replica's view at time x1 + Δ. Also, x1 +Δ ≥ τ ¯ (r)+Block Time+Δ ≥ τ¯(r)+Block Time, i.e. x1 +Δ is past the Block Time waiting period for every honest replica. Thus, all honest replicas have broadcasted a signature for B by x1 + Δ. By (9.4), all honest replicas receive f + 1 signatures by x1 + 2Δ, i.e. τ¯(r + 1) ≤ x1 + 2Δ.

We remark without proof: In the case $d = 0$ the bound can be improved to $\tau_{-}(r + 1) \leq \tau_{-}(r) +$ BlockTime $+ \Delta$. The resulting bounds are then strict for all d.

After the above information is generated, the seed will hash the information in A, as well as the file name in the seed and other key information, and generate a new SHA1 value as the HASH value of the seed, which is the port we often see There is a unique hash value named for this seed, and some can see this value in this kind of magnetic link in the sprouting link, which is the unique mark of the seed.

The above seed code content can be opened by the server, but only garbled characters can be seen. The encoding of this file follows the DRR code encoding rules. But the actual content is mainly the above. Therefore, the seed can be understood as some records of the original data.



Taho	*Hidastus* turhaa	*Itsenäinen* esto	*Estot* kaikille	Lähteellä	*Lailliset* sisällöt	Kulut	Valmistelussa puutteita
Oikeusministeriö	X					+	+
LVM	+				+	(+)	
TEM	+	(–)			(+)	+	
Ulkoasiainministeriö						X	(+)
Helsingin käräjäoikeus	(X)						
Markkinaoikeus	X		–			(+)	+
IPR University Center	+						+
Tietosuojavaltuutettu						(+)	(+)
Viestintävirasto			–				
KKV					+		
TIEKE	+			+	+	+	
EFFI ry	+						+
EK	+		(+)		+		
Teknologiateollisuus ry	+				+		
Keskuskauppakamari	(+)	(–)		+		+	
STTK ry	+			+	+		(+)
Ficom ry	+	–				+	+
Elisa Oyj	+	–	+	+	+	+	(+)
Finnet-liitto	+		–	+		+	
DNA	X	–				+	
Teliasonera Finland Oyj	+	–				+	

The above algorithm to compute trust suffers from the fact that the veracity of the feedback reported from other nodes is unknown. Super nodes could collude and report low trust values for honest nodes and high trust values for dishonest nodes. An improvement is to estimate the credibility of the feedback of other nodes and weight the reported trust values by the credibility score. To do that, common (u, v) is defined for two nodes u and v as the set of nodes with which both nodes have interacted. Given that, a measure for the similarity of the feedback of the two nodes (sim(u, v)) can be calculated:

sim(u, v) = 1 − rP w∈common(u,v) (suw−svw) 2 |common(u,v)| !b if common(u, v) 6= ∅ 0 otherwise where b is a positive integer (the original paper suggests b = 1). Then, feedback credibility can be defined as: fij =P sim(i,j) m sim(i,m) if P m sim(i, m) > 0 0 otherwise and, finally, the matrix L = (lij) can be defined as: lij =P fij cij m fimcim if P m fimcim > 0 0 otherwise which incorporates both the reported trust values and the feedback credibility.

It is now straightforward to define an iteration analog to Equation 9: ~ti =~p if i = 0 (1 − a)L T~ti−1 + a~p otherwise which converges to the left principal eigenvector of the underlying matrix of the power iteration. In additional measures to limit trust propagation between honest and dishonest nodes. We written with file sharing networks in system. In such networks, incomplete data is shared (parts of files) and cannot be checked for validity. Therefore, even honest nodes could distribute malicious data. Nodes in the network always entities in their entirety andverify their integrity before distributing them to
other nodes. CFSP network simulations have shown that the results without the additional trust propagation measures are good enough to mitigate the presence of malicious nodes.

The reported offset is weighted with the importance of the boot account of the node reporting the offset. This is done to prevent Sybil attacks. An attacker that tries to influence the calculated offset by running many nodes with low importances reporting offsets close to the tolerated bound will therefore not have a bigger influence than a single node having the same cumulative importance reporting the same offset. The influence of the attacker will be equal to the influence of the single node on a macro level. Also, the numbers of samples that are available and the cumulative importance of all partner nodes should be incorporated. Each offset is therefore multiplied with a scaling factor. Let Ij be the importance of the node reporting the j-th offset oj and viewSize be the number of samples divided by the number of nodes that were eligible for the last PoI calculation. Then the scaling factor used is

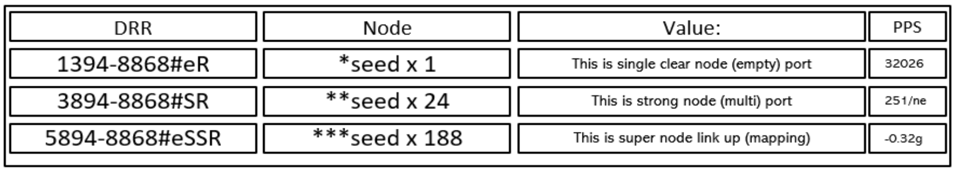
$$scale = \min\left(\frac{1}{\Sigma_j I_j}, \frac{1}{viewSize}\right)$$

This gives the formula for the effective offset o

$$o = scale \sum_j I_j O_j$$

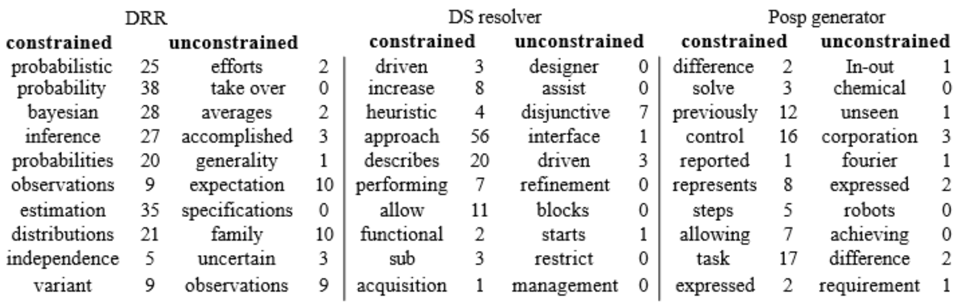
Over time, as a result of the node discovery process, the local node will become aware of more nodes in the network. Eventually, the number of known nodes (including both well-known and other nodes) will be much greater than the number of partner nodes. Periodically, a node recalculates its partner nodes. When this recalculation occurs, all known nodes are eligible to become a partner node, including nodes that have been detected as busy. With default settings, this recalculation will occur more frequently than the recalculation of trust values. By default, trust values are recalculated every 15 minutes.

The known nodes are weighted by their trust values and the partner nodes are randomly selected from among them. Nodes with larger trust values (which are more likely to be good nodes) are more likely to be chosen as partners. Nodes that have been minimally interacted with typically have trust values at or close to 0. In order to give these nodes a chance to prove themselves and build trust, they are effectively given a small boost in trust so that they have the opportunity to be selected and participate in the network. 30% of trust is evenly distributed among nodes with less than 10 interactions. In order to ensure that the network is connected, one adjustment to the random process is made. If the local node is a well-known node, it is additionally connected with all online well-known nodes. If the local node is not a well-known node, it is additionally connected with one random, online, well-known node. This ensures that all nodes are actively partnering with at least one well-known node.



DRR	Node	Value:	PPS
1394-8868#eR	*seed x 1	This is single clear node (empty) port	32026
3894-8868#SR	**seed x 24	This is strong node (multi) port	251/ne
5894-8868#eSSR	***seed x 188	This is super node link up (mapping)	-0.32g

A format for storing attributes and corresponding values within the 'value' field of CFSP to DS resolver records. The format was simply the attribute and the value contained within quotation marks (") and separated by an equal sign (=), such as: "posp=spos"(verify) "spos=cfsp"(farm) "posp=drr"(general) "supernode=block"(stake) etc.
for each random sample DRR. The result for the example for ρ and β is shown in Figure above − log2 ρ n β = 3 β = 4 β = 5 * the result will not show up at any part of the layer, but only PPS code.



DRR				DS resolver				Posp generator			
constrained		**unconstrained**		**constrained**		**unconstrained**		**constrained**		**unconstrained**	
probabilistic	25	efforts	2	driven	3	designer	0	difference	2	In-out	1
probability	38	take over	0	increase	8	assist	0	solve	3	chemical	0
bayesian	28	averages	2	heuristic	4	disjunctive	7	previously	12	unseen	1
inference	27	accomplished	3	approach	56	interface	1	control	16	corporation	3
probabilities	20	generality	1	describes	20	driven	3	reported	1	fourier	1
observations	9	expectation	10	performing	7	refinement	0	represents	8	expressed	2
estimation	35	specifications	0	allow	11	blocks	0	steps	5	robots	0
distributions	21	family	10	functional	2	starts	1	allowing	7	achieving	0
independence	5	uncertain	3	sub	3	restrict	0	task	17	difference	2
variant	9	observations	9	acquisition	1	management	0	expressed	2	requirement	1

The ORI hierarchy is refined into 20 refined parts. (fig above) One of the part

While data availability within open consensus networks is essentially an unsolved problem, there are ways of miti gating the overhead placed on validator nodes. One simple solution is to realise that while validators must shoulder the responsibility for data availability, they need not actually store, communicate or replicate the data themselves. Secondary data silos, possibly related to (or even the very same) collators who compile this data, may manage the task of guaranteeing availability with the validators providing a portion of their interest/income in payment. However, while this might buy some intermediate scalability, it still doesn't help the underlying problem; since adding more chains will in general require additional validators, the ongoing network resource
consumption (particularly in terms of bandwidth) grows with the square of the chains, an untenable property in the long-term. Ultimately, we are likely to
keep bashing our heads against the fundamental limitation which states that for a consensus network to be considered available safe, the ongoing bandwidth requirements are of the order of total validators times total input information. This is due to the inability of an untrusted network to properly distribute the task of data storage across many nodes, which sits apart from the eminently distributable task of processing.

ORI is a server application platform where private and public chains coexist, enabling the public to create functions and deploy them to the edge network of the community.
Unlike other server-less providers or decentralized platforms that only have regional data centers, ORI's edge network is composed of cloud service servers that generate seed codes for every user in the world.
ORI is a powerful platform, and this course provides a good introduction to its functions.
Follow ORI to build a localization engine that presents data based on the edge location closest to the application user.
After completing this course, you will be ready to start experimenting with your own super node and start two-way mining.