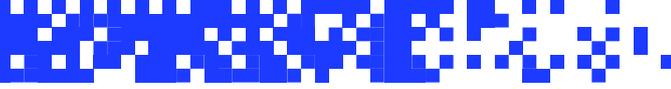


DCIPHER

DCIPHER TOKENOMICS

2025 | RANDAMU



1 Introduction to dcipher network economics

The dcipher network is a generalized decentralized threshold cryptography network. In this document, we break down the cryptoeconomic components that enable signers to signal their availability to participate in general threshold cryptography tasks, as well as the mechanisms through which a user can optimally incentivize a committee of signers.

The dcipher workflow is as follows: The protocol maintains a network of signers that are available to join a threshold cryptography committee upon request.

Once a user chooses a group of available signers to join their committee, they can start requesting general threshold jobs. A given job may require the signer to perform some given computation, then generate and submit a corresponding signature. The user defines the desired threshold, that is, the minimum number of signatures required for the task to be completed successfully.

We will dive into four main pillars of dcipher economics:

1. **Proof of availability (PoA):** Through this mechanism, signers will be able to signal to users that they are available to join a committee and perform threshold cryptography tasks, as well as to signal the conditions under which they will agree to join the committee. Token emissions are used to incentivize the desired amount of signing availability, ensuring that it is always possible for a user to form a committee.
2. **Committee signature pricing mechanism:** Once a user has assembled a committee, the user needs to induce the committee to, on request, submit enough signatures to meet a predefined threshold, without necessarily requiring the entire committee to sign. We discuss the mechanism to properly price and collateralize a signing request such that the user can directly manipulate the probability of having the threshold met.
3. **Service Level Agreement (SLA) Enforcement:** While the committee pricing mechanism ensures that rational signers are incentivized to provide signatures when needed, there is still room for malicious signers to join a committee but avoid submitting signatures, fully relying on others to meet the threshold. We therefore include a mechanism by which anyone can report an underperforming signer.

4. **dcipher macroeconomics:** We elaborate on how these different components tie into self-sustaining tokenomics and how the protocol accrues value as demand for signature committees increases.

With these components in mind, we can briefly summarize the economic workflow.

To become a dcipher signer, one needs to lock a certain amount of collateral to signal signing availability. This collateral can be slashed if the signer claimed they were available, but failed to join a committee when requested.

Different committees have different computational needs, so being an available signer does not mean that one has to accept every possible job that is requested. When a signer signals availability, they list the conditions under which they will be available. These conditions are described by the type of job, their cost per signature, and their maximum frequency of providing signatures.

Suppose that there are many types of job, labeled $j \in J$. A given signer $s \in S$ can signal that they are available to perform the type of job j at a cost of $C_{j,s}$ per signature. Further, they indicate they can complete these signatures at a maximum frequency of $f_{j,s}$. If a signer is requested to join a committee which satisfies their cost and frequency conditions, then they must accept, or their collateral may be slashed.

The protocol also must incentivize signing availability for conditions that are in *actual* high demand. For example, if a signer signals availability with a frequency that is too low to be useful for any committee, then they will not be rewarded for being available since they were available for a job type that did not have any actual demand. Signers are incentivized to honestly indicate their frequency and cost capabilities.

Once a set of signers have entered a committee for a type of job j requested by the user i , the economics incentivizes the signers to provide enough signatures to meet the required threshold, while avoiding waste of resources by providing more signatures than required.

The key element of our signature pricing mechanism is the **probabilistic signature game**. Once a signer enters a committee, there exist positive incentives: A reward is split among all committee members if the threshold is met; and there are negative incentives: The collateral is slashed from all signers if the threshold is not met. These incentives are fine-tuned so that the optimal and utility-maximizing strategy for each signer is to act probabilistically and return signatures when requested only with probability p . The rewards and collateral are calculated using the assumption of cost per signature, $C_{j,i}$, where this cost must be equal to or higher than the signaled cost of every signer in the committee. The probabilistic signature game aligns incentives so that users can obtain their thresholds with high certainty while avoiding wasted resources.



2 Proof of Availability (PoA)

The token emissions are used to encourage a network of nodes to be available to become members of various dcipher committees. Once they accept becoming part of a committee, they are rewarded through user-paid fees.

First, we must define what "to be available" means. Can we target a specific availability ratio: say have 10% more availability than what is currently in use?

To answer these questions, we first must define some quantities, such as how much "signing power" there is in the network.

We provide definitions here of how signing power can be measured and how incentives are used to target a given network availability.

2.1 Signing power

There are J different types of job. The job type is labeled $j = 1, \dots, J$. We assume for now that we want to incentivize the same availability across all types of jobs, but this is not necessary.

For every committee, a cost parameter $C_{j,i}$ where i labels the committee.

The committee consists of N_i signers, of which n_i signatures are needed to meet the threshold.

Each committee also has a frequency $f_{j,i}$ at which signatures are required.

We then define the **signing power** of a given committee as

$$P_{j,i} = n_i C_{j,i} f_{j,i}.$$

This captures the cost of signing incurred by that committee.

We now define the total signing power across all committees performing a job type j as

$$\mathcal{P}_j = \sum_{i \in I_j} P_{j,i},$$

where I_j is the set of committees that perform the job type j .

2.2 Available signing power

Having defined the signing power of the set of active committees, we can compare the amount of signing power that is not in active use but has signaled its availability.

A given signer, labeled s declares that they are available for the job type j . To do this, they must specify two quantities: $C_{j,s}$ their cost per signature for that type of job, and $f_{j,s}$, their maximum signing frequency.

If requested to join a committee with the cost parameter $C_{j,i}$ and frequency $f_{j,i}$, they are required to accept the request, as long as $C_{j,i} > C_{j,s}$ and $f_{j,i} < f_{j,s}$. If they reject becoming part of the committee, they will be penalized.

As long as the signer is available and accepts requests according to their set parameters, they will be rewarded.

We define the amount of signing power that is available but not actually in use as

$$\bar{\mathcal{P}}_j = \sum_{s \in S_{\text{available}}} C_{j,s} f_{j,s}$$

where $S_{\text{available}}$ is the set of signers available for requests, but not currently active.

We now develop the notion of **how much extra availability do we want as a network?** which is defined in the ratio $\bar{\mathcal{P}}_j / \mathcal{P}_j$

2.3 Incentivizing availability

2.3.1 Bare availability incentives

Suppose we use token rewards to incentivize a network of available signers. In a given period of time, a number of tokens B_j are distributed as rewards for a given type of job.

These rewards are divided among all signers who are currently on a committee or are available to become part of a committee. They are distributed according to how much signing power they provide. The available signer s would obtain the amount of tokens

$$B_{j,s} = B_j \frac{C_{j,s} f_{j,s}}{\mathcal{P}_j + \bar{\mathcal{P}}_j}$$

A signer currently active on a committee i would earn

$$B_{j,s} = B_j \frac{C_{j,i} f_{j,i}}{\mathcal{P}_j + \bar{\mathcal{P}}_j}.$$

There are issues with this formulation. A signer is incentivized to declare a very large $C_{j,s}$ to obtain large rewards, while never being asked to participate in a committee, since the cost would be too high. We want to modify this formula so that it encourages availability on higher demand: availability at lower costs and higher frequency.

2.3.2 Demand-adjusted availability incentives

We adjust the previous reward distribution formula by introducing multipliers, which skew rewards towards availability at lower costs and higher frequencies.

We have information on the cost and frequencies of all active committees, that is, $\{C_{j,i}\}, \{f_{j,i}\}$.

If a signer is available at a cost $C_{j,s}$, we can ask how much signing power is currently active at a cost higher than $C_{j,s}$? We define the multiplier

$$x_{j,s}^C = \frac{\mathcal{P}_{j,s}^C}{\mathcal{P}_j},$$

where $\mathcal{P}_{j,s}^C = \sum_{i \in I_s^C} C_{j,i} f_{j,i}$ and I_s^C is the set of all committees that have a cost higher than $C_{j,s}$

Notice, for example, that if the signer sets a cost that is higher than every cost in an active committee, their multiplier will be zero. And if they set a cost that is lower than every cost in an active committee, their multiplier will be one.

Similarly, we define a frequency multiplier,

$$x_{j,s}^f = \frac{\mathcal{P}_{j,s}^f}{\mathcal{P}_j},$$

where $P_{j,s}^f = \sum_{i \in I_s^f} C_{j,i} f_{j,i}$ and I_s^f is the set of all committees that have a frequency *lower than* $f_{j,s}$.

We now propose, using these multipliers, the amount of rewards signer s receives to be modified to,

$$B_{j,s} = B_j \frac{x_{j,s}^C x_{j,s}^f C_{j,s} f_{j,s}}{\mathcal{P}_j^x + \bar{\mathcal{P}}_j^x},$$

where

$$\mathcal{P}_j^x = \sum_{i \in I_j} x_{j,i}^C x_{j,i}^f P_{j,i},$$

$$\bar{\mathcal{P}}_j^x = \sum_{s \in S_{\text{available}}} x_{j,s}^C x_{j,s}^f C_{j,s} f_{j,s}$$

2.4 Incentivizing truthful availability

Signers specify their conditions for availability: their minimum cost per signature, and their maximum signature frequency. Since these quantities are self-reported, we would like measures that incentivize honest reporting of costs and frequencies.

Truthful reporting of availability is encouraged by two mechanisms: Competition and demand-adjusted incentives.

In isolation, a signer may want to signal an artificially high cost per signature. This is because the higher the cost assumption, the higher the fees the user will pay to the committee, and the higher the reward the signer will receive for providing signatures.

Once there is a sufficient supply of signers, competition provides downward pressure on reported costs. If other signers can undercut the cost a signer has reported, then these new signers will be more frequently called to join committees, forcing the original signer to also lower their signaled cost (up to a floor cost that is their actual cost).

Demand-adjusted incentives also provide downward pressure on reported costs. If a signer reports a cost that is higher than the cost per signature of every active committee, they will receive no rewards. The lower they are able to report their costs, the more availability rewards they will receive.

This is an important reason for the network to always incentivize some extra availability in addition to the active signing power: If there is always additional signing availability, this ensures that there will always be competition among available signers. The rewards issued



to ensure signing availability have the positive side effect of lowering costs for users, since they encourage more truthful cost reporting.

2.5 Penalizing fake availability

We described how rewards are issued to incentivize signers to demonstrate honest availability. But how do we ensure signers will actually join committees once they are requested and not simply "walk away" with the availability rewards?

We define **false availability** as signers who signal that they are available at a given minimum cost and maximum frequency, but when asked to join a committee that satisfies their conditions, reject the request.

There is an incentive problem if the signer can collect proof of availability rewards while rejecting actual job requests.

There are two lines of defense against false availability:

Reward escrow: The first line of defense is to ensure that there is no positive utility to be gained from signaling fake availability. This is achieved by ensuring proof-of-availability (PoA) rewards are only received by the signer after joining a committee.

As the signer signals availability, they begin earning rewards; however, these rewards are held in an escrow account and not available immediately to the signer. Once the signer completes a job request corresponding to this availability, the accumulated rewards are then released to the signer.

Fake availability penalties: While the reward escrow ensures that there is no direct financial incentive to signal fake availability, this does not rule out a fake availability attack by malicious signers.

Malicious signers may wish to signal fake availability in order to spam the network and squeeze out real availability. This would result in the network having less than 10% extra availability as intended.

Given this, for signers who continue to signal false availability, the reward escrow is not sufficient. To avoid continued false availability, signers must lock slashable collateral when signaling availability.



The amount of slashable collateral should be one that keeps the total false availability at a minimum. This can be measured by the availability failure rate: when signers are requested to join a committee, how often are they unable to join? We specify a maximum acceptable failure rate, for example: at most 5% requests are allowed to fail.

As soon as the failure rate increases above 5%, the false availability collateral must increase, to return the failure rate to below 5%. As the failure rate continues to increase above 5%, the collateral must increase accordingly. The collateral continues to rise indefinitely as the failure rate approaches 100%

This behavior is captured by a logit function collateral. Defining r_f as the acceptable failure rate, we define:

$$\text{Collateral} = \text{Max} \{0, \text{logit} [t(r)]\},$$

where $t(r)$ is a linear function of the failure rate r , such that $t(r_f) = 0.5$ and $t(1) = 1$.



3 Committee economics

In this section, we dive into the economics of rewarding signers who have already committed to participate in a committee.

Suppose that user i gathers a committee of N_i signers. They establish a threshold of n_i signatures required, to consider the request successful. It is assumed that every signer on the committee has a cost per signature of at most C_i .

Wasteful incentives: Suppose the user wants to ensure that their signature threshold is met. They could ensure that the signers are incentivized to sign, by placing a reward $R_i > C_i N_i$, to be given out if the threshold is met. In this case, it is always in the best interest of each signer to submit their signature.

The problem with this approach is that it can be wasteful, this is paying for N_i signatures, when, in fact, only n_i are needed.

In this section, we will explore the **probabilistic signature game**, where we design incentives such that the optimal strategy for each signer is to sign probabilistically. If each signer signs with a probability $p \geq n_i/N_i$ then on average at least n_i signatures are obtained, while avoiding the waste of paying for N_i signatures.

3.1 Probabilistic signature game

First, let us set out the goals of the probabilistic signature game. The user needs to specify a given amount of reward for signers when the signature threshold is met. The user can also specify an amount of collateral that can be slashed if the signature is not met. The probabilistic signature game is a method of fixing these reward and penalty mechanisms, such that the optimal strategy for a signer to maximize their utility is to sign probabilistically, with a probability $p > n_i/N_i$, but with $p < 1$ to avoid wasting resources.



Rewarding all signers vs. rewarding those who signed: The first relevant question when designing this game is *"How are rewards distributed?"* An option that may seem fair is after the threshold is met, a reward is given only to those signers who provided signatures. After all, the ones who didn't sign didn't do any work, why should they be rewarded?

The problem with this approach is that it cannot satisfy the goals of the probabilistic signature game, as the optimal strategy will be deterministic, leading to waste of resources.

If the reward given per received signature is greater than the cost C_i , then each signer is individually incentivized to return their signature, even if the threshold has already been met. Rewarding only those who sign would then lead to the total reward of $R_i > C_i N_i$, which we have established is wasteful.

In the probabilistic signature game, when the threshold is met, all signers receive a reward, whether they sign or not. As we shall see, this feature leads to the optimal strategy for each signer being probabilistic, which avoids waste of resources.

3.2 Reward, penalty and signer utility

The probabilistic signature game is defined by the following rules:

1. If a threshold of n_i signatures or more is obtained, the user gives him a reward of R that is split equally between all N_i signers, whether they signed or not.
2. If the threshold is not met, each signer gets an amount S of collateral slashed.
3. The user can adjust the parameters R and S , given the inputs n_i , N_i , and C_i , in a way to target a specific behavior of the signer.

With these parameters, we can now build a utility function for the signer s . The signer can perform one of two strategies: return a signature when requested, or do not return a signature. Returning a signature means that the signer will incur a cost C_i , but also increases the chance of obtaining the reward, while decreasing the chance of penalty.

The utility if signer s returns a signature is

$$U^1 = \frac{R}{N_i} P_{n_i-1}^{N_i-1} + \frac{R}{N_i} \sum_{k \in n_i} P_k^{N_i-1} - S \sum_{k=0}^{n_i-2} P_k^{N_i-1} - C,$$

where $P_k^{N_i-1}$ is the probability that the rest of the committee of $N_i - 1$ signers, k signatures

are received. In this case, the signature of the signer s is only relevant in the case where $n_i - 1$ signatures are obtained from the rest of the committee, in which case their signature becomes decisive.

The utility of signer s does not return a signature is,

$$U^0 = -SP_{n_i-1}^{N_i-1} + \frac{R}{N_i} \sum_{k \in n_i}^{N_i-1} P_k^{N_i-1} - S \sum_{k=0}^{n_i-2} P_k^{N_i-1},$$

Given that the probabilistic signature game is defined for the signer s by these utility functions, what is the optimal strategy of the signer?

3.3 Optimal signer strategy

The optimal strategy for each signer depends on how much information they have available about what the other signers will do.

On the one hand, the signers may choose to collaborate and coordinate. The user only needs to get n_i signatures. The signing committee may choose to avoid risk and cost by coordinating with each other and prearranging which of the N_i signers will sign. As long as it is known that n_i of these signers will be signing, other signers do not need to sign for this round and will obtain maximum utility.

On the other hand, we want our design to function robustly, even in the case where there is no coordination among signers.

We therefore elaborate here on the question: *“What is the optimal strategy for a signer who has no information on how the rest of the signers will behave?”*

We have shown the utility of returning a signature U^1 and not returning a signature U^0 . More generally, the signer s may decide to sign probabilistically, with probability p . Their utility function then becomes

$$U^p = \left(p \frac{R}{N_i} - (1-p)S \right) P_{n_i-1}^{N_i-1} + \frac{R}{N_i} \sum_{k \in n_i}^{N_i-1} P_k^{N_i-1} - S \sum_{k=0}^{n_i-2} P_k^{N_i-1} - pC.$$

The assumption that the signer s has no information on how the other signers will behave allows us to further elaborate on what the utility function looks like. **Without further information, signer s 's best guess is to assume that everyone else is signing with the same**

utility maximizing strategy as they are. This means that all other signers also sign with the same probability p . This means that the total number of signatures received follows a binomial distribution:

$$P_k^{N_i} = \binom{N_i}{k} p^k (1-p)^{N_i-k}$$

This can be substituted back into the utility function which becomes

$$\begin{aligned}
 U^p = & \left[p \frac{R}{N_i} - (1-p)S \right] \binom{N_i-1}{n_i-1} p^{n_i-1} (1-p)^{N_i-n_i} \\
 & + \frac{R}{N_i} \sum_{k \in n_i}^{N_i-1} \binom{N_i-1}{k} p^k (1-p)^{N_i-1-k} \\
 & - S \sum_{k=0}^{n_i-2} \binom{N_i-1}{k} p^k (1-p)^{N_i-1-k} - pC.
 \end{aligned}$$

Now, the signer can turn this into an optimization problem. From the above formula, the signer can find what is the optimal response probability, p , that maximizes the expected utility?. The rational signer should then sign with probability

$$p_{\max} = \text{ArgMax} U^p.$$

The user then has to choose the appropriate parameters (n_i, R, S) such that $p_{\max} \geq n_i/N_i$.

3.4 Coordinated signer strategies

The previous section described a signer that has no information on how other signers will act. In that case, your best bet is to assume that everyone else is acting with the same utility maximizing strategy.

This game design ensures that the user expects to get their desired signature threshold **even in the worst-case scenario that there is no coordination between signers.** The threshold guarantees only get stronger if signers coordinate with each other.

If the signers coordinate, they can predetermine which n_i of the N_i signers will return a signature, and the rest $N_i - n_i$ of the signers can skip doing any work, while ensuring that

the threshold is met and waste of resources is avoided.

If the signers have reached such a coordination agreement, There will be n_i signers who gained a utility of

$$U^{n_i} = \left(\frac{R}{N_i} - C \right)$$

and $N_i - n_i$ signers who have received a utility of

$$U^{N_i - n_i} = \frac{R}{N_i}.$$

The average utility per signer is then

$$U^s = \frac{n_i \left(\frac{R}{N_i} - C \right) + (N_i - n_i) \frac{R}{N_i}}{N_i}.$$

In general, this utility is greater than that of the uncoordinated strategy $U^s > U^p$. This means that either signers will make larger profit margins or will be able to pass some of those extra savings to the user, by accepting to be in committees with lower reward rates (lower cost assumptions). Ultimately, coordination can lead to lower costs and higher guarantees.

3.5 The probabilistic signature game service level agreement (SLA)

What guarantees does the dcipher protocol provide to its users? What kind of SLA does the probabilistic signature game result in?

We focus mainly on the "*worst-case SLA*". Within a given committee, every signer has agreed upon a given cost parameter, C , this means that everyone's cost is at most C . This SLA becomes even stronger if the actual costs of the signer are less than C .

As we have discussed, the SLA for the user could even improve if the signers were coordinated.

The worst-case SLA is the case where C reflects the actual costs of all signers, and the signers are completely uncoordinated. How likely is it that the user will have their threshold of signatures met? The number of signatures they will obtain will be random, but how will the total number of signatures be distributed?



As shown, if C reflects real costs and the signers are not coordinated, the optimal strategy for rational signers is to sign with probability p_{\max} . If every signer signs probabilistically in the same way, then the total number of signatures will follow a binomial distribution $B(N_i, p_{\max})$.

We can then ask what the probability is that the threshold of n_i will be met? This can be computed by defining the cumulative binomial distribution function $F(n, N_i, n_i/N_i)$. The probability of the threshold being met is given by $1 - F(n_i, N_i, n_i/N_i)$.

Depending on the SLA required by the user, they may adjust their success rate by adjusting their probabilistic signature game parameters.

For example, if the user wants a higher success rate, they could use a higher threshold parameter, $n'_i > n_i$, in their formulation of the probabilistic signature game. This would lead to a higher fee that must be paid, which in turn shifts the expected number of signatures from n_i to n'_i . This shift leads to a new probabilistic strategy $p_{\max} = n'_i/N_i$. The probability of the threshold being met (the SLA) can then be adjusted as desired by the user, and is given by $1 - F(n_i, N_i, n'_i/N_i)$.



4 SLA guarantees and self reporting

From the incentives of the probabilistic signature game, we expect that the number of signatures the user receives for each request follows a binomial distribution, with a mean that is controlled by setting the reward and slashing parameters.

This binomial distribution should be a worst-case scenario. The number of signatures will be even more consistently above threshold if the signers coordinate or if their operational costs are lower than stated.

While rational signers should return signatures with the appropriate probability, in this section, we consider the case of malicious signers, which may return lower signatures, even if this seems economically irrational.

First, we define what detection of malicious behavior looks like, at the level of a single signer, and at the committee level. Once we have defined how to detect malicious behavior, we propose a self-policing incentive structure, where signers are incentivized to police and report on each other, ensuring negative consequences for signers who behave suboptimally.

4.1 Detection of suboptimal behavior

Since the optimal strategy is probabilistic, at any given signature request, every signer will return a signature or not return a signature. From an isolated event, it is impossible to tell if a signer was acting optimally or not. In the same way, the total number of signatures received can range from 0 to N_i , and it would be impossible to infer from an isolated event whether this was drawn from a binomial distribution or not.

However, after many such requests, it becomes possible, with high certainty, to determine whether a signer, or an entire committee, has been acting optimally or not.

Suppose that a signer's optimal strategy was to return a signature with probability p for each signature request, and there have been M such requests. The total number of signatures this signer will provide on all requests should be drawn from the **binomial**(M, p).

dcipher network keeps track of the activities of each signer, so it is possible for any other signer to see each other's record of signatures. Suppose that it is observed that a signer, over M requests, returned Mp_o signatures.

Anyone with access to these data can perform a simple statistical test to assess the probability that the number of signatures Mp_o could have been drawn from the binomial **binomial** (M, p) .

Any user can simply measure the p-value of the given outcome to see if the null hypothesis, "that the signer was returning signatures with probability of at least p per request", can be rejected. This is given by computing the cumulative distribution $F_{\text{binomial}}(Mp_o, M, p) < \alpha$, where α is a predetermined threshold for statistical significance set by the network.

Similarly, the null hypothesis can be tested in committees. For a committee of N_i signers, with optimal probability of signing p , across M signature requests, the total number of signatures obtained should follow the binary distribution (NMp, p) .

However, for the committee, a more relevant question to ask is: what percentage of times was the threshold not met? Suppose that the required threshold is n_i . For each trial, the probability of the threshold not being met is $F_{\text{binomial}}(n_i, N_i, p)$. Then one can look at the outcome across the M trials and keep the score: 1 point for the threshold being met, 0 points for not meeting the threshold. The total score at the end of the M trials would be drawn from the binomial distribution $(M, 1 - F_{\text{binomial}}(n_i, N_i, p))$.

Now suppose that observing, across M trials, the threshold was actually met m_0 times. We can then test the null hypothesis: "The number of successful full trials follows either the distribution **binary** $(M, 1 - F_{\text{binomial}}(n_i, N_i, p))$, for which we can compute the p-value as $F_{\text{binomial}}(m_0, M, 1 - F_{\text{binomial}}(n_i, N_i, p)) < \alpha$

4.2 Rewards for SLA enforcement, and penalties for malicious strategies

We have defined mechanisms through which it can be measured whether a predefined SLA has been met or whether we can detect the signers were behaving maliciously with a certain level of confidence.

These mechanisms are used as building blocks for an "SLA enforcement" protocol.

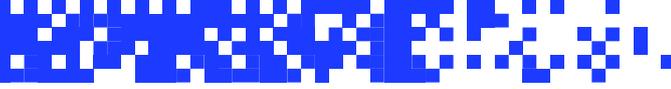
User enforcement: Suppose the user that has paid for a certain SLA wants to have stronger guarantees that the set of signers will actually follow the set incentives. The fees



that the user pays for M signature requests can be held in an escrow account to be paid to the signer committee. If the user can show that with sufficient certainty the committee was acting suboptimally, a portion of the fees may be reimbursed to the user. The percentage of fees reimbursed depends on the p-value, or how certain the user is that the committee behaved maliciously.

Signer mutual enforcement: Similarly, signer a within the committee might notice that the signer b is acting maliciously. The signer a can then submit a report, demonstrating that the signer b has been acting maliciously, with a given level of certainty. When this claim is verified, the signer a is rewarded a percentage of the rewards that were held in escrow for the signer b .

In both cases, the number M of rounds of reward that need to be kept in escrow depends on the level of certainty α required for SLA policing. If a very low alpha is required, then many rounds of signatures are required to reach that level of certainty.



5 Summary of dcipher Macroeconomics

We have described a number of economic incentives that make up the dcipher protocol. How do all of these tie together into a coherent macroeconomic picture for the dcipher token? We summarize here what all these mechanisms mean in terms of an analysis of token sinks and faucets dcipher and supply inflation / deflation.

5.1 dcipher Capped supply and allocation

\$DCP is a fixed supply token, with a total of ten billion \$DCP tokens that will ever be minted. Sixty percent of these tokens have been allocated to the dcipher community, while forty percent have been allocated for maintainers and investors.

5.2 Faucets: Incentivizing demand-driven availability

The main issuance expenditure in dcipher 's cryptoeconomics as we have described is the need to incentivize availability of signers. For this reason, a minimum of 15% of the total token allocation is reserved for Proof of Availability (PoA) rewards.

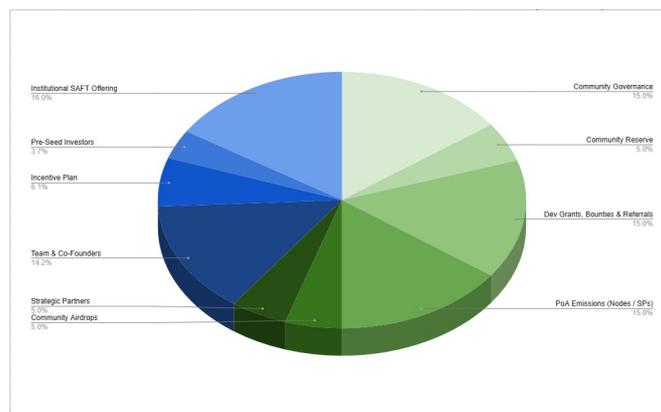


Figure 5.1: DCP Token Allocation



However, how will these tokens be issued over time? What will be the impact on overall inflation?

The fundamental principle governing token issuance is the availability target. For example, dcipher wants to incentivize the availability of 10% more signing power than currently in active use in committees. This target dictates that token issuance be dynamic, the amount of tokens to be minted at a given time is those that ensure that we have 10% extra availability.

An important consequence of this mechanism is that **the total issuance will be based on demand**. As the total signing power of the network increases, the amount of available signing power increases. On the other hand, as we will discuss in the following section, an increase in network usage leads to token value accrual, meaning that a smaller amount of tokens will be needed to incentivize the same level of availability.

Suppose that at a given point in time, the network needs to incentivize an amount $\bar{\mathcal{P}}_j$ of signing power for the type of job j , where this is 10% of the active signing power, so $\bar{\mathcal{P}}_j = 0.1\mathcal{P}_j$. It requires a certain amount of rewards B_j to incentivize this level of availability. The quantity B_j is dynamically set in a way that finds the correct total incentives that lead to the available signing power of the target.

Therefore, B_j will depend on:

1. Network size: More active signing power means more available power is required. To illustrate this, we can denote the total reward as $B_j = 0.1b_j\mathcal{P}_j$
2. Token price: If token becomes more valuable, it requires fewer token emissions to reach the same target.
3. Competition among signers: The price of incentive for availability goes down as signers compete for rewards, accepting lower amounts of rewards to undercut competition.

Suppose that the total network revenue, W_j for the type of job j is proportional to the total signing power for that job, $W_j = w_j\mathcal{P}_j$. At that point in time, the inflation rate is then given by

$$\sum_j (0.1b_j - w_j)\mathcal{P}_j.$$

Notice that since we have established B_j depends on token price, it implicitly depends on network revenue as well, since larger network revenue places upward pressure on token price.

The total supply of tokens being capped means that in the far future, we may not have



enough tokens available to issue the amount B_j necessary to incentivize 10% availability. So, the described issuance mechanism will work in this way until the availability incentive allocation runs out.

5.3 Token sinks and value accrual mechanisms

In this section, we describe the various uses that give intrinsic utility to the dcipher token and which drive network revenue and demand for the token.

First, we have described several forms of slashable collateral at different stages of the protocol.

1. Collateral for incentivizing truthful availability, slashable if the signer turned out to not really be available when requested to join a committee.
2. Collateral within the probabilistic signature game. This is a parameter of the game that can be slashable when the required signature threshold is not met.
3. SLA guarantees. This collateral is not slashed, but is refunded to the user, when they prove the committee was underperforming, or transferred from one underperforming signer to another reporting signer.

Participating in all these aspects of the protocol requires users and signers to acquire and lock up the dcipher token, which intrinsically drives the demand for the token with greater network adoption.

The second token sink comes in the form of network fees that are to be burned. Users derive value from the dcipher service, as it allows users to find and orchestrate a group of signers, and align their behavior through the probabilistic signature game. **Some of that generated value will accrue to the network, by collecting a small fee at every instance of the probabilistic signature game.** This can be done by requiring users with an active committee to pay a fee proportional to their total signing power of the form $W_{j,i} = w_j P_{j,i}$.

The exact fee rate w_j can be adjusted so that a specific inflation rate is targeted. From the previous section, we see that if $w_j = 0.1b_j$ there is zero net inflation.

A higher inflation rate is useful at network launch to bootstrap the formation of a successful signer network. When this network is established, inflation can be lowered to focus on network adoption, leading to token value accrual. We can therefore establish a desirable inflation rate trajectory over time r_t . The network fees can be adjusted such that the

desired inflation rate is achieved,

$$\sum_j (0.1b_j - w_j)\mathcal{P}_j = r_t,$$

or, in the case that we apply the same fee rate to all different types of jobs, (so we define $w = w_j$ not dependent on job type)

$$\sum_j (0.1b_j - w)\mathcal{P}_j = r_t,$$
$$w = \frac{-r_t + \sum_j 0.1b_j\mathcal{P}_j}{\mathcal{P}}.$$



Disclaimer

The present whitepaper and/or any other accompanying documentation (“Document”) only provide educational material about the dcipher Network and its utility token. Please note that the dcipher Network and the token are under active development and are subject to change. The Threshold Association, in formation, may change this Document at any time at its sole discretion without notice. Any documentation is provided for informational purposes only and does not constitute some kind of prospectus, key information document or similar document. No prospectus, key information document or similar document will be provided at any time. There is no guarantee for the completeness of the documentation provided. All numbers and forward-looking statements mentioned within the present document as well as any accompanying documentation reflect mere estimations/indications. They are not guaranteed and may change substantially. Any and all liability of the Threshold Association, in formation, and/or any affiliated legal entity or private individual for the completeness and accuracy of the documentation provided and any damages arising from reliance on such documentation is limited to the fullest extent permitted by any applicable law.

Any dispute related to or arising out of the information provided within the present Document as well as any accompanying documentation shall be submitted to the exclusive jurisdiction of the competent courts of Zug, Switzerland, with the exclusion of any other jurisdiction or arbitration. This disclaimer, the Document as well as any accompanying documentation shall be governed by and construed and interpreted in accordance with the substantive laws of Switzerland, excluding the Swiss conflict of law rules. The United Nations Convention for the International Sales of Goods is excluded.

DCIPHER TOKENOMICS WHITE PAPER V1.0 - APRIL 17, 2025

Questions, comments, and suggestions are welcome at info@randa.mu

