# HyveMind: A Trustless Election Infrastructure

Abenayo Okeke

**ABSTRACT**
HyveMind is a crypto-system that combines cryptographically-secured protocols and real-world systems which allow for a new and novel level of cooperation among a population with regards to allocating shared resources (taxes, for example) and creating and administrating cooperative solutions to shared problems.

By piggy-backing on the probabilistic properties and guarantees of other well-understood crypto-systems, HyveMind ensures that it cannot be centrally controlled by a motivated minority and cannot be censored unless the Internet is shut down.

We present several examples of using the HyveMind system as a mixed system of representation containing elements of technocracy and direct democracy which, through the use of Proof-of-knowledge (POK) weighting, solves many problems of direct democracy. We call this novel form of government a cryptocracy. This system can be used on multiple scales: from home owner's associations all the way up to nation states.

## 1 INTRODUCTION

Representative forms of government are vulnerable to exploitation both well tested (bribery) and novel (social media propaganda campaigns). HyveMind seeks to use the properties of modern cryptography-powered systems to, in some cases, render various exploits of representative government useless and in other cases, to make exploits exponentially more complex, expensive, and difficult.

Human societies are fluid, complex systems that, once at scale, necessitate protocols and systems for managing common resources and settling disputes. Once human beings created methods and tools capable of doing significant damage to our shared environment, cooperation about how best to responsibly use those methods and tools ceased to be optional; however, because of our societal complexity, there are always competing interests that often have radically different definitions of the word "responsible."

In some cases, a group of people will organize to promote interests that further their personal short-term interests with long-term consequences which are actively detrimental to the continued viability of their host civilizations and, in some extreme cases, the planet itself - our only habitat. The regular and successful propagation of such an organization's interests are, in our judgment, a red flag that the governing system has been successfully compromised (or owned/pwnd in hacker vernacular).

The use of computer security terms may, at first, seem strange; however, when one examines systems of government, one sees that they share many properties with a networked computer system. For example, disproportionate influence of a government by a small subset of the population (oligarchy) rarely starts at the highest level (or scale) of government. It begins with the compromise of smaller client systems: local and then state government.

This is very similar to how a corporate network is breached. A hacker will constantly probe network clients (individual employees, for example) for a weak link using phishing, social engineering, and other means. Once such a link is discovered and compromised, that owned client becomes the springboard to other clients. The goal usually to gain ever high levels of clearance, privileges, access, and/or credibility for each subsequent attack until the entire network is compromised.

In a political science context, the exploits may take the forms of bribery, both legal (lobbying, campaign contributions) and illegal (quid pro quo, influence peddling, and outright bribery). Other exploits include ignored conflicts of interests; for example, an official who is currently in charge of their own election.

We propose a distributed system that can be used to administer elections; A system which cannot be completely controlled or materially compromised by an adversary or any single, centralized party. Using HyveMind it would not be possible for a corrupt official to discard or censor opposition votes. It would not be possible to alter the vote tally. It would not be possible for any individual to vote more than once. It would not be possible to game or otherwise rig the registration process to disenfranchise potential opposition voters.

All these things are true if HyveMind is used to replace the election infrastructure of a representative democracy while keeping the rest of the rules, norms and practices the same; however, if a constituency were so inclined, the novel features that HyveMind introduce, which are not possible or practical using current election infrastructure, allows for a total re-imagining of how a democracy can function.

## 2 SYSTEM STRUCTURE

### 2.1 CORE REQUIREMENTS

*One person, one vote*
For the system to be valid, one of it first and most basic requirement is to be able to ensure that the is no more

than one account per person. The system must be able to perform reliable liveliness testing to protect against fraudulent registration via impersonation, synthesis, or automation.

*No specialized hardware*
Another goal of the system is that it shall not require specialized hardware to run a full node or any version of client software.

## 2.2 ARCHITECTURE

### 2.2.1 PHYSICAL ARCHITECTURE

Blockchain
A digital database containing information (such as records of financial transactions) that can be simultaneously used and shared within a large decentralized, publicly accessible network [2].

For our purposes, the blockchain is secured using Nakamoto Consensus [3].

Full Client
A perpetually running instance of the system software which is designed to process all user interactions with the system including, but not limited to, account registration, vocal sample processing, and vote counting.

Light Client
A instance of the system software which has its functionality limited to submitting transactions to full clients.

### 2.2.2 LOGICAL ARCHITECTURE

Measure
A question with a discrete set of possible answers which is posed to a constituency for the purpose of obtaining a Result (see below).

Result Definition
A data structure containing the following properties:

- Start and end time (in blocks)
- Required endorsements for vote validity (if any)
- Sample selection algorithm (if any)
- Included result(s)
- Included result(s) weighting
- Win/loss definition

Constituent
An actual or potential participant of a Cell (see below) whose input is used to determine the Result (see above) of a Measure (see above).

Cell
A network of constituents that agree to a set of rules defining results, the conditions necessary for winning and losing an election, constraints on participation and many other characteristics which when summed together describe a means of achieving consensus.

Mempool
The mempool holds submitted submitted, unprocessed transactions.

The System
The nature of a Nakamoto Consensus-based network is that the larger the number of running full nodes, the harder that network is to attack and compromise. Because of this, we believe that all cells should use a blockchain secured by the largest amount of full nodes. This means that until such time as a single blockchain is unable to handle the network's transaction volume, all cells should be secured by a global blockchain.

Zero-Knowledge Proofs
There are several system use cases where it is desirable to certain information be verifiable by,but not known to, an untrustworthy 3rd party.

To achieve this, we rely on a technology called Zero-Knowledge Non-Interactive Arguments of Knowledge (ZK-SNARKS).

A detailed discussion of ZK-SNARKS is outside the scope of this paper, but the technology has been in use in the cryptocurrency Z-Cash and, as of the published date of this paper, there have been no known efficient adversaries.

## 2.2 BIOMETRIC REGISTRATION

Since a principal goal of the system is that it should function using non-specialized, commodity computing hardware, biometric techniques such as iris scans and fingerprints are eliminated as possible solutions. The system must be able to prevent fraudulent key registration. This requirement further reduces the possible candidates. Facial recognition becomes non-viable when this constraint is introduced. For example someone with access to a cache of drivers license, college id, or other identification cards would be able to obtain key registration tokens for people without their knowledge or consent. While some web sites perform liveness testing [4] to try ensure the submitted photo was taken by a live subject, they are trivial to circumvent using various presentation attacks and/or image manipulation. Image manipulation detection techniques

such as error level analysis are not reliable enough [5] to be used with our system.

At the time of this writing, the most robust form of presentation attack-resistant biometric identification that can be performed with commodity hardware is voice identification with sonar-based liveliness testing based on the work of Linghan, Sheng and Jies at Florida State University [6]. Further research can be performed on thwarting voice synthesis attacks by requiring subjects to recite a psuedo randomly generated phenom construction proven distinguishable, with high confidence, from synthesized samples.

**2.3 SYSTEM FLOW**

Bob wishes to register as a constituent within a HyveMind cell. In order for the system to function properly, it must ensure that bob can have, at most, key registered request at a time.

Let's say that Bob wants to register a key for a cell governing his local HOA. First Bob must download the mobile light client application. Then he must obtain a registration token. These tokens prevent DDOS attacks on the system as each submission has a financial cost. After opening it, he will hit the "register new key." The app will generate and submit a hashed key. This key is used to determine the phonemes and order they should be spoken. Submitting it to the blockchain ahead of time allows for future third party verification. This submission also creates a timestamp which is used to limit valid submissions to a threshold in the future.

Using a publicly available string of data and the pseudo-random number, the app will deterministically construct a phoneme sequence and prompt Bob to speak the sequence. While Bob is speaking the phonoee sequence, his phone's speaker will be emitting a high frequency sound then recording the sound's reflection then analyzing the Doppler shifts as to determine the liveliness of Bob [7].

The recording and Doppler shift data is submitted to the network. If validation nodes run audio analysis on Bob's submission and determine that Bob's sample was lively, a hash of the sample's identifying characteristics are used as input for submission in a invertible bloom table [8] that is secured by the blockchain. The purpose of the bloom table is to determine if a submission's voice has never been submitted before.

**3 EXAMPLE SYSTEM USES**

We think it may be helpful to illustrate some potential use cases for the system.

Sampling Democracy

In a Sampling Democracy laws are rejected or approved by different random samples from a population. In this system the law making process is separated from the voting process. It also differs from direct democracy as it does not ask the entire population to vote on measures.

For example, a population may elect representatives by traditional means who write laws, vote on laws in committee, and when approved for "the floor" the system will choose n random samples of size p from the population. Each person in the sample is asked to vote up or down on the legislation and if the average "yes" percentage among all n samples is above a threshold, the legislation becomes law.

The selection of the sample, while not truly random, will provide the same level of security as a functional pseudo-random generator - unpredictability. The sample selection is handled autonomously by the network using Randhound and Randherd [9], which is capable or generating randomness in 6 second intervals with a failure probability of <= 0.08% against a byzantine adversary [9].

Because the composition of samples cannot be known ahead of time and the samples themselves are short-lived and ephemeral, bribery could not be targeted and would, by necessity, have to operate on some sort of open market. For example, by posting an ad on a dark web message board offering or soliciting payment for proof of voting a certain way on a piece of legislation.

Epistimocracy

The term Epistimocracy is a combination the greek words for knowledge (epistēmē) and government (dēmokratía). It functions much like a Sampling Democracy except that before a law can be voted on, its knowledge domain must be classified. The knowledge domain specifies what subject matter a potential voter must prove proficiency in before being allowed to vote on a matter in the domain.

Such a system would be heavily front-loaded with debate and consideration about the number of knowledge domains, the classification criteria for each domain, what constitutes domain proficiency, and a host of other issues.

Using Hyvemind, a population could trustlessly select samples, using endorsements (see below) to determine eligibility with respect to knowledge domains, and record votes.

Privacy-Assured Certification/Registration

There are many things, drug registries for example, for which it is a public good to have some sort of record to discourage abuse but, understandably, there is apprehension over the potential misuse of a such a registry that contains identifying information.

Endorsements and the underlying ZK-SNARKS technology allows for a certifying entity to attest to a person's abilities without ever knowing the identity of said person.

For example, if a state wanted to legalize and tax a drug with a high potential for abuse it would make sense for the state to track sales. Using the system, that state would be able to track sales to individuals on an immutable public ledger without that ledger containing any identifying information whatsoever.

For example, the state could require a citizen to possess a state-issued token before a dispensary could legally sell a substance to that citizen. The state could then associate that token with citizen account that submitted the token request, thereby keeping track of how much of the substance has been distributed.

Using ZK-SNARKS, a distributor could verify that the token being submitted with a purchase request is in fact valid without knowing anything about the token or its owner.

This ability to record sensitive information about an individual without recording identifying information can prove useful in a number of other contexts.

## 4 PROBLEMS

### 4.1 VERSION OF THE TRUSTED CLIENT PROBLEM

Ensuring that an individuals ability to register a key and, thus vote, is globally unique is the the most important of the system's core requirements. As a corollary, the system must also ensure that it is technically impossible or at least problematically infeasible that an adversary could fraudulently register and control the keys of multiple persons, allowing them to control multiple votes.

These requirements present a number of technological challenges which we will address here.

### 4.2 GLOBALLY UNIQUE KET REGISTRATION

Biometric identification is the process of using one of the many globally unique physical features to differentiate and thereby identify individuals. There are many features that could be used; from commonly used one's like fingerprints, facial recognition, and retinal scans to more exotic traits such as ear lobes, vein patterns and cardiac rhythms. For our purposes, however, we are constrained by one of the system's core requirements: the ability to function end-to-end using only commodity computing hardware. This requirement eliminates most of the candidate biometric features. Ultimately voice print recognition was chosen. Facial recognition was a close second but the additional constraint imposed while satisfying the next requirement would eliminate facial recognition from consideration.

### 4.3 FRAUDULENT KEY REGISTRATION

With biometric identification the best defense against presentation attack-based fraud is liveliness testing. Liveliness testing is ensuring the biometric sample provided is coming from a living,breathing human being.

One of the reasons facial recognition was eliminated is because the size of the sample payload, along with the required processing power, becomes large when we introduce liveliness testing. This is due to the need to submit and process video.

Linghan, Sheng and Jies research shows it's possible to robustly perform sophisticated liveliness detection using standard smartphone hardware with 95.4% accuracy in the worst case and 99%+ in the best case.

By combining the above liveliness testing with a variation of reCatchpa, we believe we can push the worst case performance percentage closer to the best case percentage with the goal of total convergence. This is achieved by constructing non-words out of phonemes and asking the registrant to read them as part of their liveliness verification. The set of possible phoneme combinations is derived by experimentation and machine learning with a goal of picking the combinations that when processed by state-of-the-art speech synthesis software is most distinguishable from natural human speech.

The system must be able to defend against registration using completely artificially-generated voices. If the system were unable to distinguish a sample created by voice synthesis software from that of a sample spoken from the mouth of a human, then we would run afoul of our one vote, one person requirement.

We believe that the doppler-based articulation detection described by Linghan, Sheng and Jies [6] allow the system to best meet it's various requirements. This allows the registrant to submit voice samples using their smart phones in a natural manner. Those samples can then be submitted to the network for processing.

**4.4 DISTRIBUTED BIOMETRIC DATA PROCESSING**

Another core requirement of the system is that it function in a trustless environment without the aid or dependency on any central authority to resolve disputed or determine canon.

One of the problems introduced by depending on a network of computers which may contain adversaries or byzatine actors is that a node can ignore the biometric sample input and attest that the results of processing the sample are different then that of a honest node. While network size is > 3, a single byzantine node cannot threaten the integrity of the system. As the protocol does not rely on byzantine consensus and can therefore withstand > N/3 byzantine nodes. However, as is the case with Nakamoto consensus, the integrity of the system will come into doubt if the number of byzantine actors ever becomes > N/2.

The next problem to be solved is how to ensure that nodes which are running the current, canonical version of the system's full node implementation agree on the identifying information extracted from the biometric sample data (BSD) before that output is used to modify the global bloom table and the updated bloom table is immutably committed in the blockchain. This problem has two components. First, is how can a distributed network with no central authority tell whether or not node output is being produced by the official full node software? Second, is how to reach consensus among nodes as to which output submitted by other nodes is genuine and should be committed to the blockchain. Third, is how do we prevent byzantine node from censoring the propagation of messages by honest nodes?

To deal with the first problem, we propose the use of a repudiation-resistent voting scheme combined with a limit (as opposed to no-limit) betting system. At a high level, each node constantly queries the mempool for BSD. Processing of the BSD's is a form of transaction conducted on the blockchain. A node will publish a hash of it processing output for a regristrant's BSD. This node will also bet a fixed amount of the system's native token against the outcome of more nodes voting for a different processing output for the same BSD. The bets are also keyed to the node's client software version so a node won't be punished for running an outdated with the exception of the transaction fee, which is always non-refundable. This small penalty is, among other things, intended as incentive for nodes to run the same client version as a majority its peers.

The next problem component is creating a consensus protocol. At a high level, consensus is achieved on a given <OutputsubBSD, version> tuple. The network will consider the <OutputsubBSD, version> tuple with the most votes in the last N blocks to be the canonical processing for <OutputsubBSD, version>. The first instance of a given <OutputsubBSD, version> tuple which is mined in a block is referred to as a “candidate registration.” or CR

To prevent censorship each vote must hashed before submission and submitted with a nonce. A good faith effort must be made to use a unique nonce on a network-wide scope per round (ie, nonces should not be reused).The payload of a CR transaction is E^public_key(hash_algo(output^BSD||nonce), version). After N blocks the submitter of the CR (the first one to have their candidate transaction mined) will publish the preimage of the hash mentioned above. Any vote which is equal to the concatenation of the CR preimage and their submitted nonce is added to that candidate registration’s tally.

Bets for losing candidates are divided among the voters of the winning candidate. This scheme is designed to make it more profitable to bet on the output of your software if you know you‘re running a legitimate version. This as opposed to, for example, accepting a preimage shared by a candidate submitter who pays you to artificially inflate the vote count on their CR. Voting for a dishonest node’s CR, or being the dishonest node yourself, is risky because you are betting against the honesty of the entire network. Considering that only the winning CR gets paid out and all others lose their transaction fees and bets, unless a dishonest node is certain that they have compromised a plurality of the network (the size of which fluctuates), they are essentially throwing their money away trying to influence the network into being dishonest.

**4.5 DETERMINING LEGITIMATE VERSIONS**

Consensus is achieved by totaling the bets for each version. The version with
the most bets is paid out a reward. All others have their bets returned minus the betting fee. To prevent other nodes from censoring votes for version other than their own votes should be

encrypted using the public key provided by the candidate. For consideration: the previous censorship resistance scheme has the disadvantage of having to deal with the outcome of the candidate node being unavailable to provide the private key, leaving other nodes unable to decrypt, and therefore verify candidate votes.

**4.7 MANAGING KEY SECURITY**

One of the main weaknesses of the system described is that the system itself cannot ensure the security of constituent's private keys from loss, theft, or personal computer compromise.While we may suggest best security practices in the creation, storage and utilization of private keys, ultimately true security can only be achieved through enforceable mechanisms - not best-practice suggestions.

For these reasons, the system (by default) states that a key may be used to cast no more than 1 vote. This parameter may be adjusted by a sub-network wishing to add more flexibility to their instantiation.

**4.8 ENSURING PUBLIC CONFIDENCE**

Much like a conventional government, this system must derive its legitimacy from the consent and confidence of the governed; however, the highly technical nature of the system presents a challenge - how do you get people to have confidence in what they do not understand? As a corollary, if a majority of people do not understand the cryptographic primitives and principles which ensure the system's integrity, how can the system's integrity be defended when attacked? For example, a common and effective misinformation technique would be say "how can anyone be sure that most of these votes aren't being fraudulently cast?" Although the question asks the defender to do the impossible (prove a negative), this is an incredibly effective technique for undermining confidence in any system that a sufficient number of people do not fully understand.

The solution is to create a bounty tax. This tax creates an ever-increasing pool of funds that are used to reward anyone who can successfully demonstrate a compromise of the system. For example, a constituency may decide to allocate 25% of their bounty fund to anyone who can demonstrate a fraudulent registration. As time goes on and the bounty grows, the constituency's confidence that most, if not all, of the votes cast into their system are legitimate grows with it. In this way, a person who may not fully understand the workings of the system may still be able to feel confident about its integrity.

**4.9 PARTICIPANT ANONYMITY**

One of the unique aspects of the system is that it provides complete anonymity to its account holders (constituents) while maintaining cryptographic provability and auditability. This is meant to address a critical vulnerability in current representative democracy systems. In a traditional representative system, there is an incentive to influence votes of representatives by private interests which may run counter to the public good. Because representatives are known, there exists the ability to do just that. Our system allows for constituent's anonymity, even to nodes responsible for registering constituent's public keys.

To protect against de-anonymization using IP addresses, the system makes use of onion routing[10] to hide the IP address used to submit transactions to the network.

**5 MANAGING VOTER LOCALITY**

With any constituency there is the issue of proving membership and preventing outsiders from exerting undue influence on the governance process. While one of the core requirements is that users should be able to register using nothing other than commodity hardware, preferably just a smartphone. This present an issue when we want to restrict voting to a certain locality. Because of the trusted client problem, there is no purely technical solution to the locality problem.

What we can provide is a way for groups with an interest in introducing locality-based restraints to be able to do so in a cryptographically secure manner. Being a completely decentralized system means that there is no central authority to obtain or deny permission to aspects of the system. The protocol is the only arbiter of what is and isn't possible within the system. However, a constituency may very well come to an agreement as to what constitutes valid actions in the context of using the system to manage its decision making. To aid in this, the system provides the abstraction of endorsements. Endorsements are arbitrary attributes that may be added to voter accounts. the idea is that a constituency may designate some administrator to handle the validation of voter eligibility. Through endorsements, the system provides an administrator a cryptographically-secure means of capturing voter attributes

**5.1 ENDORSEMENT EXAMPLE**

The standard subdivision home owner's association (SSHOA) wishes to use the system to manage decision making; however, it wishes to restrict participation to

homeowners under its jurisdiction. To achieve this the SSHOA can require that valid votes can only come from accounts with the "SSHOA" endorsement.

To implement this the SSHOA will chose a person to administer the endorsement process. This person will submit a "createEndorsement" transactions with the requisite fee along with the human-readible name of the endorsement, and a set of cryptographic hashes with the number of elements equal to the number of households eligible to participate in the SSHOA. The administrator then uses the postal system to mail a different preimage of one of the set elements submitted with the “createEndorsement” transaction to each eligible household. For convenience the preimage data is encoded as a QR code, printed on a piece of paper and then mailed.

When the resident receives the mailer, it will instruct the homeowner to scan the code using the system's client application software and submit it with an "obtainEndorsement" transaction. When the transaction has been mined and the submitted data is verified as a preimage to one of the hashes submitted with the "createEndorsement" transaction, the resident's account (public key) will now be associated with it.

## 6 PROOF-OF-KNOWLEDGE ENDORSEMENTS

In order to support an epistimocracy as described earlier, a constituency would need some means of testing for proficiency in a knowledge domain and a way of recording this information in the system. The system can natively support the latter via endorsements, the former will likely need to depend on an authority, such as a proctored testing facility on a college campus.

We would eventually like to see Proof-of-Knowledge testing done in a trustless manner, without the use a proctor or any other authority. Work on solving this problem is ongoing.

## 7 CONCLUSION

This paper lays out a high level view of a system that allows for an election infrastructure that removes many of the costs and barriers to participation of current election systems.It is highly resistant to tampering, rigging, or post-election manipulation by any central authority or other adversarial party.

Using this system, representative democracy itself can be completely re-imagined to address and fix many of the weaknesses that are the cause of low public confidence afflicting many democracies around the world.

Finally, it is our hope that this paper will spark discussion about the viability of the proposed system, if ever one is constructed. We hope that with constructive critique and contribution by the cryptocurrency and computer science community at-large that this system can eventually become a reality and change the trajectory of human history.

## REFERENCES


[1] 2017. Hearing Your Voice is Not Enough https://acmccs.github.io/papers/p57-zhangA.pdf (2017)

[2] https://www.merriam-webster.com/dictionary/blockchain

[3] https://bitcoin.org/bitcoin.pdf

[4] https://www.intechopen.com/books/advanced-biometric-technologies/liveness-detection-in-biometrics

[5] https://en.wikipedia.org/wiki/Error_level_analysis#controversy

[6] 2017 Hearing Your Voice is Not Enough http://ww2.cs.fsu.edu/~tan/paper/ccs2017.pdf (2017)

[7] 2016 VoiceLive: A Phoneme Localization based Liveness Detection for Voice Authentication on Smartphones http://ww2.cs.fsu.edu/~tan/paper/ccs2016.pdf (2016)

[8] 2011. Invertible Bloom Lookup Tables https://arxiv.org/pdf/1101.2245

[9] 2016. Scalable Bias-Resistant Distributed Randomness https://eprint.iacr.org/2016/1067.pdf

[10] Onion Routing:Executive Summary https://www.onion-router.net/Summary.html